EPSON
EXCEED YOUR VISION

S2R72V18
Technical Manual

SEIKO EPSON CORPORATION

Rev.1.00

NOTICE

No part of this material may be reproduced or duplicated in any form or by any means without the written permission of Seiko Epson. Seiko Epson reserves the right to make changes to this material without notice. Seiko Epson does not assume any liability of any kind arising out of any inaccuracies contained in this material or due to its application or use in any product or circuit and, further, there is no representation that this material is applicable to products requiring high level reliability, such as, medical products. Moreover, no license to any intellectual property rights is granted by implication or otherwise, and there is no representation or warranty that anything made in accordance with this material will be free from any patent or copyright infringement of a third party. This material or portions thereof may contain technology or the subject relating to strategic products under the control of the Foreign Exchange and Foreign Trade Law of Japan and may require an export license from the Ministry of Economy, Trade and Industry or other approval from another government agency.

All other product names mentioned herein are trademarks and/or registered trademarks of their respective companies.

©SEIKO EPSON CORPORATION 2008, All rights reserved.

Scope

This document applies to the S2R72V18 USB 2.0 device/host controller LSI.

Table of Contents

1. Functions 1

1.1 USB Ports 1
1.2 USB Device/Host Selection 2
1.2.1 USB Device/Host Function Selection Procedure 2
1.2.2 USB Port State Change Detection Status 2
1.2.2.1 USB Port State Change Detection Status Usage Examples 2
1.2.2.1.1 Device Port Change Status 2
1.2.2.1.2 Host Port Change Status 3
1.2.2.2 Signal Line Change Status Usage 3
1.3 USB Device Control 6
1.3.1 Endpoints 6
1.3.2 Transactions 8
1.3.2.1 SETUP Transactions 10
1.3.2.2 Bulk/Interrupt OUT Transactions 11
1.3.2.3 Isochronous OUT Transactions 12
1.3.2.4 Bulk/Interrupt IN Transactions 13
1.3.2.5 Isochronous IN Transaction 14
1.3.2.6 PING Transactions 15
1.3.3 Control Transfers 16
1.3.3.1 Setup Stage 17
1.3.3.2 Data Stage/Status Stage 18
1.3.3.3 Automatic Address Setup Function 18
1.3.3.4 Descriptor Reply Function 18
1.3.4 Bulk/Interrupt/Isochronous Transfers 19
1.3.5 Data Flow 19
1.3.5.1 OUT Transfer 19
1.3.5.2 IN Transfer 20
1.3.6 Bulk-Only Support 20
1.3.6.1 CBW Support 21
1.3.6.2 CSW Support 22
1.3.7 Cable Negotiation Function (Auto Negotiator) 23
1.3.7.1 Auto Negotiator 24
1.3.7.1.1 DISABLE 24
1.3.7.1.2 IDLE 24
1.3.7.1.3 WAIT_TIM3US 24

1.3.7.1.4 WAIT_CHIRP 25
1.3.7.1.5 WAIT_RSTEND 25
1.3.7.1.6 DET_SUSPEND 25
1.3.7.1.7 IN_SUSPEND 25
1.3.7.1.8 CHK_EVENT 26
1.3.7.1.9 WAIT_RESTORE 26
1.3.7.1.10 ERR 26
1.3.7.2 Operation for Individual Bus Events 26
1.3.7.2.1 Suspend Detection in HS Mode 26
1.3.7.2.2 Suspend Detection in FS Mode 28
1.3.7.2.3 Reset Detection in HS Mode 30
1.3.7.2.4 Reset Detection in FS Mode 31
1.3.7.2.5 HS Detection Handshake 32
1.3.7.2.5.1 When Connected to the FS Host Downstream Port 33
1.3.7.2.5.2 When Connected to the HS Host Downstream Port 35
1.3.7.2.5.3 If Reset During Sleep 37
1.3.7.2.6 Resume Issue 39
1.3.7.2.7 Resume Detection 41
1.3.7.2.8 Cable Attachment 43
1.4 USB Host Control 45
1.4.1 Channels 45
1.4.1.1 Channel Outline 45
1.4.1.2 Dedicated Control Channel 47
1.4.1.3 General Channels 48
1.4.1.4 Channel Usage Examples 50
1.4.1.4.1 With One Storage Device Connected 50
1.4.1.4.2 With One Communication Device Connected 51
1.4.1.4.3 With One Human Interface Device Connected 52
1.4.1.4.4 With Two Storage Devices Connected Via a Hub 53
1.4.2 Scheduling 54
1.4.3 Transactions 55
1.4.3.1 SETUP Transaction 56
1.4.3.2 Bulk OUT Transaction 57
1.4.3.3 Interrupt OUT Transaction 59
1.4.3.4 Isochronous OUT Transaction 60
1.4.3.5 Bulk IN Transaction 61
1.4.3.6 Interrupt IN Transaction 63
1.4.3.7 Isochronous IN Transaction 65

1.4.3.8 PING Transaction 67
1.4.3.9 Low-speed (LS) Transaction 69
1.4.3.10 Split Transaction 71
1.4.4 Control Transfers 72
1.4.4.1 Setup stage 73
1.4.4.2 Data Stage/Status Stage 73
1.4.4.3 Control Transfer Support Function 74
1.4.5 Bulk/Interrupt/Isochronous Transfers 77
1.4.6 Data Flow 77
1.4.6.1 OUT Transfers 77
1.4.6.2 IN Transfers 78
1.4.7 Zero Length Packet Automatic Issuing Function 79
1.4.7.1 Bulk/Interrupt OUT Transfer Zero-Length Packet Automatic Issuing Function 79
1.4.8 Bulk-Only Support Function 80
1.4.9 Audio Class Assist Function 85
1.4.10 Host State Management Support Function 86
1.4.10.1 Host State 86
1.4.10.1.1 IDLE 88
1.4.10.1.2 WAIT_CONNECT 89
1.4.10.1.3 DISABLED 90
1.4.10.1.4 RESET 90
1.4.10.1.5 OPERATIONAL 92
1.4.10.1.6 SUSPEND 92
1.4.10.1.7 RESUME 93
1.4.10.2 Detection Functions 94
1.4.10.2.1 VBUS Error Detection 94
1.4.10.2.2 Disconnect Detection 96
1.4.10.2.2.1 If HS Device is Disconnected 96
1.4.10.2.2.2 If FS or LS Device is Disconnected 97
1.4.10.2.3 Remote Wakeup Detection 98
1.4.10.2.3.1 If HS Device is Connected 98
1.4.10.2.3.2 If FS Device is Connected 100
1.4.10.2.3.3 If LS Device is Connected 101
1.4.10.2.4 Device Chirp Detection Function 102
1.4.10.2.4.1 If a Correct Device Chirp is Detected 102
1.4.10.2.4.2 If an Error Device Chirp is Detected 104
1.4.10.2.5 Port Error Detection 105
1.4.10.3 Individual Host State Management Support Function Explanations 106

1.4.10.3.1 GoIDLE 106
1.4.10.3.2 GoWAIT_CONNECT 108
1.4.10.3.2.1 If an FS Device is Connected 108
1.4.10.3.2.2 If an LS Device is Connected 110
1.4.10.3.3 GoDISABLED 112
1.4.10.3.3.1 If an HS Device is Connected 112
1.4.10.3.3.2 If an FS Device is Connected 114
1.4.10.3.3.3 If an LS Device is Connected 116
1.4.10.3.4 GoRESET 118
1.4.10.3.4.1 Resetting for HS Device 118
1.4.10.3.4.2 Error Device Chirp Detection 121
1.4.10.3.4.2.1 If Chirp Completion Disable (H_NegoControl_1.DisChirpFinish) Setting is 0 121
1.4.10.3.4.2.2 If Chirp Completion Disable (H_NegoControl_1.DisChirpFinish) Setting is 1 123
1.4.10.3.4.3 Resetting for FS Device 126
1.4.10.3.4.4 Resetting for LS Device 128
1.4.10.3.5 GoOPERATIONAL 129
1.4.10.3.6 GoSUSPEND 130
1.4.10.3.6.1 If HS Device is Connected 130
1.4.10.3.6.2 If FS Device is Connected 132
1.4.10.3.6.3 If LS Device is Connected 134
1.4.10.3.7 GoRESUME 136
1.4.10.3.7.1 If HS Device is Connected 136
1.4.10.3.7.2 If FS Device is Connected 138
1.4.10.3.7.3 If LS Device is Connected 140
1.4.10.3.8 GoWAIT_CONNECTtoDIS 142
1.4.10.3.9 GoWAIT_CONNECTtoOP 143
1.4.10.3.9.1 If HS Device is Connected 143
1.4.10.3.9.2 If FS or LS Device is Connected 145
1.4.10.3.10 GoRESETtoOP 147
1.4.10.3.10.1 If HS Device is Connected 147
1.4.10.3.10.2 If FS or LS Device is Connected 148
1.4.10.3.11 GoSUSPENDtoOP 149
1.4.10.3.12 GoRESUMEtoOP 151
1.5 Power Management Functions 152
1.5.1 SLEEP 153
1.5.2 ACTIVE 153
1.6 FIFO Management 154

1.6.1 FIFO Memory Map 154
1.6.2 Descriptor Area 155
1.6.2.1 Writing Data to Descriptor Area 155
1.6.2.2 Executing Data Stage (IN) with Descriptor Area 156
1.6.3 CBW Area 156
1.6.3.1 CBW Area (for USB device) 156
1.6.3.2 CBW Area (for USB host) 156
1.6.4 CSW Area 157
1.6.4.1 CSW Area (for USB device) 157
1.6.4.2 CSW Area (for USB host) 157
1.6.5 FIFO Access Methods 157
1.6.5.1 RAM Access (RAM_Rd) 157
1.6.5.2 RAM Access (RAM_WrDoor) 158
1.6.5.3 FIFO Access (Register Access) 158
1.6.5.4 FIFO Access (DMA) 158
1.6.5.5 FIFO Access Restrictions 159
1.7 CPUIF 160
1.7.1 CPUIF Mode 160
1.7.2 CPUIF Mode Setup 161
1.7.3 Block Configuration 163
1.7.3.1 REG (Registers) 163
1.7.3.1.1 Synchronous Register Access (Write) 163
1.7.3.1.2 Synchronous Register Access (Read) 163
1.7.3.1.3 FIFO Access (Write) 164
1.7.3.1.4 FIFO Access (Read) 164
1.7.3.1.5 FIFO Access Fractional Number Processing 165
1.7.3.1.6 RAM_Rd Access 168
1.7.3.1.7 Asynchronous Register Access (Writing) 168
1.7.3.1.8 Asynchronous Register Access (Reading) 168
1.7.3.2 DMA0/DMA1(DMA ch.0/ch.1) 169
1.7.3.2.1 Basic Functions 169
1.7.3.2.2 Terminal Setup 171
1.7.3.2.3 Count Mode (Write) 171
1.7.3.2.4 Count Mode (Read) 173
1.7.3.2.5 Free-run Mode (Write) 175
1.7.3.2.6 Free-run Mode (Read) 175
1.7.3.2.7 REQ Assert Count Option (Write) 176
1.7.3.2.8 REQ Assert Count Option (Read) 177

1.7.3.2.9 DMA FIFO Access Fractional Number Processing 177

2. Register Maps 178

2.1 Register Configuration 178
2.2 Initial Register Map 179
2.3 Port 0 Register Maps 180
2.3.1 Device/Host Common Register Map 180
2.3.2 Device Register Map 186
2.3.3 Host Register Map 190
2.4 Port 1 Register Maps 195
2.4.1 Device/Host Common Register Map 195
2.4.2 Host Register Map 201

3. Register Details 206

3.1 Device/Host Common Register Details 206
3.1.1 Port0:000h / Port1:200h *MainIntStat (Main Interrupt Status)* 207
3.1.2 Port0:001h / Port1:N/A *USB_DeviceIntStat (USB Device Interrupt Status)* 209
3.1.3 Port0:002h / Port1:202h *USB_HostIntStat (USB Host Interrupt Status)* 211
3.1.4 Port0:003h / Port1:203h CPU_IntStat (CPU Interrupt Status) 213
3.1.5 Port0:004h / Port1:204h FIFO_IntStat (FIFO Interrupt Status) 214
3.1.6 Port0:006h / Port1:206h *RootIntStat (Root Interrupt Status)* 215
3.1.7 Port0:008h / Port1:208h *MainIntEnb (Main Interrupt Enable)* 216
3.1.8 Port0:009h / Port1:N/A USB_DeviceIntEnb (Device Interrupt Enable) 217
3.1.9 Port0:00Ah / Port1:20Ah *USB_HostIntEnb (Host Interrupt Enable)* 218
3.1.10 Port0:00Bh / Port1:20Bh CPU_IntEnb (CPU Interrupt Enable) 219
3.1.11 Port0:00Ch / Port1:20Ch FIFO_IntEnb (FIFO Interrupt Enable) 220
3.1.12 Port0:010h / Port1:210h *RevisionNum (Revision Number)* 221
3.1.13 Port0:011h / Port1:211h *ChipReset (Chip Reset)* 222
3.1.14 Port0:012h / Port1:212h *PM_Control (Power Management Control)* 223
3.1.15 Port0:014h-015h / Port1:214h-215h *WakeupTim_H,L (Wakeup Time High, Low)* 225
3.1.16 Port0:016h / Port1:216h *H_USB_Control (Host USB Control)* 226
3.1.17 Port0:017h / Port1:217h *H_XcvrControl (Host Xcvr Control)* 227
3.1.18 Port0:018h / Port1:N/A *D_USB_Status (Device USB Status)* 229
3.1.19 Port0:019h / Port1:219h *H_USB_Status (Host USB Status)* 230
3.1.20 Port0:01Bh / Port1 MTM_Config (Multi Transceiver Macro Config) 231
3.1.21 Port0:01Fh / Port1:21Fh *HostDeviceSel (Host Device Select)* 232
3.1.22 Port0:020h / Port1:220h FIFO_Rd_0 (FIFO Read 0) 233
3.1.23 Port0:021h / Port1:221h FIFO_Rd_1 (FIFO Read 1) 233
3.1.24 Port0:022h / Port1:222h FIFO_Wr_0(FIFO Write 0) 234

3.1.25 Port0:023h / Port1:223h FIFO_Wr_1(FIFO Write 1) 234
3.1.26 Port0:024h-025h / Port1:224h-225h FIFO_RdRemain_H,L (FIFO Read Remain High, Low) 235
3.1.27 Port0:026h-027h / Port1:226h-227h FIFO_WrRemain_H,L (FIFO Write Remain High, Low) 236
3.1.28 Port0:028h / Port1:228h FIFO_ByteRd(FIFO Byte Read) 237
3.1.29 Port0:02Ah / Port1:22Ah FIFO_ByteWr(FIFO Byte Write) 238
3.1.30 Port0:030h-031h / Port1:230h-231h RAM_RdAdrs_H,L (RAM Read Address High, Low) 239
3.1.31 Port0:032h / Port1:232h RAM_RdControl (RAM Read Control) 240
3.1.32 Port0:035h / Port1:235h RAM_RdCount (RAM Read Counter) 241
3.1.33 Port0:038h-039h / Port1:238h-239h RAM_WrAdrs_H,L (RAM Write Address High, Low) 242
3.1.34 Port0:03Ah / Port1:23Ah RAM_WrDoor_0 (RAM Write Door 0) 243
3.1.35 Port0:03Bh / Port1:23Bh RAM_WrDoor_1 (RAM Write Door 1) 243
3.1.36 Port0:040h / Port1:240h RAM_Rd_00 (RAM Read 00) 244
3.1.37 Port0:041h / Port1:241h RAM_Rd_01 (RAM Read 01) 244
3.1.38 Port0:042h / Port1:242h RAM_Rd_02 (RAM Read 02) 244
3.1.39 Port0:043h / Port1:243h RAM_Rd_03 (RAM Read 03) 244
3.1.40 Port0:044h / Port1:244h RAM_Rd_04 (RAM Read 04) 244
3.1.41 Port0:045h / Port1:245h RAM_Rd_05 (RAM Read 05) 244
3.1.42 Port0:046h / Port1:246h RAM_Rd_06 (RAM Read 06) 244
3.1.43 Port0:047h / Port1:247h RAM_Rd_07 (RAM Read 07) 244
3.1.44 Port0:048h / Port1:248h RAM_Rd_08 (RAM Read 08) 244
3.1.45 Port0:049h / Port1:249h RAM_Rd_09 (RAM Read 09) 244
3.1.46 Port0:04Ah / Port1:24Ah RAM_Rd_0A (RAM Read 0A) 244
3.1.47 Port0:04Bh / Port1:24Bh RAM_Rd_0B (RAM Read 0B) 244
3.1.48 Port0:04Ch / Port1:24Ch RAM_Rd_0C (RAM Read 0C) 244
3.1.49 Port0:04Dh / Port1:24Dh RAM_Rd_0D (RAM Read 0D) 244
3.1.50 Port0:04Eh / Port1:24Eh RAM_Rd_0E (RAM Read 0E) 244
3.1.51 Port0:04Fh / Port1:24Fh RAM_Rd_0F (RAM Read 0F) 244
3.1.52 Port0:050h / Port1:250h RAM_Rd_10 (RAM Read 10) 244
3.1.53 Port0:051h / Port1:251h RAM_Rd_11 (RAM Read 11) 244
3.1.54 Port0:052h / Port1:252h RAM_Rd_12 (RAM Read 12) 244
3.1.55 Port0:053h / Port1:253h RAM_Rd_13 (RAM Read 13) 244
3.1.56 Port0:054h / Port1:254h RAM_Rd_14 (RAM Read 14) 244
3.1.57 Port0:055h / Port1:255h RAM_Rd_15 (RAM Read 15) 244
3.1.58 Port0:056h / Port1:256h RAM_Rd_16 (RAM Read 16) 244
3.1.59 Port0:057h / Port1:257h RAM_Rd_17 (RAM Read 17) 244

3.1.60 Port0:058h / Port1:258h RAM_Rd_18 (RAM Read 18) 244
3.1.61 Port0:059h / Port1:259h RAM_Rd_19 (RAM Read 19) 244
3.1.62 Port0:05Ah / Port1:25Ah RAM_Rd_1A (RAM Read 1A) 244
3.1.63 Port0:05Bh / Port1:25Bh RAM_Rd_1B (RAM Read 1B) 244
3.1.64 Port0:05Ch / Port1:25Ch RAM_Rd_1C (RAM Read 1C) 244
3.1.65 Port0:05Dh / Port1:25Dh RAM_Rd_1D (RAM Read 1D) 244
3.1.66 Port0:05Eh / Port1:25Eh RAM_Rd_1E (RAM Read 1E) 244
3.1.67 Port0:05Fh / Port1:25Fh RAM_Rd_1F (RAM Read 1F) 244
3.1.68 Port0:061h / Port1:261h DMA_Config (DMA Config) 246
3.1.69 Port0:062h / Port1:262h DMA_Control (DMA Control) 248
3.1.70 Port0:064h-065h / Port1:264h-265h DMA_Remain_H,L (DMA FIFO Remain High, Low) 249
3.1.71 Port0:068h-069h / Port1:268h-269h DMA_Count_HH,HL (DMA Transfer Byte Counter High/High, High/Low) 250
3.1.72 Port0:06Ah-06Bh / Port1:26Ah-26Bh DMA_Count_LH,LL (DMA Transfer Byte Counter Low/High, Low/Low) 250
3.1.73 Port0:06Ch / Port1:26Ch DMA_RdData_0 (DMA Read Data 0) 252
3.1.74 Port0:06Dh / Port1:26Dh DMA_RdData_1 (DMA Read Data 1) 252
3.1.75 Port0:06Eh / Port1:26Eh DMA_WrData_0 (DMA Write Data 0) 253
3.1.76 Port0:06Fh / Port1:26Fh DMA_WrData_1 (DMA Write Data 1) 253
3.1.77 Port0:071h / Port1:271h *ModeProtect (Mode Protection)* 254
3.1.78 Port0:073h / Port1:273h *ClkSelect (Clock Select)* 255
3.1.79 075h *CPU_Config (CPU Configuration)* 256
3.1.80 Port0:07E-07Fh / Port1:27Eh-27Fh *CPUIF_MODE (CPUIF Mode)* 258
3.1.81 Port0:080h-081h / Port1:280h-281h AREA0StartAdrs_H,L (AREA 0 Start Address High, Low) 259
3.1.82 Port0:084h-085h / Port1:284h-285h AREA1StartAdrs_H,L (AREA 1 Start Address High, Low) 259
3.1.83 Port0:088h-089h / Port1:288h-289h AREA2StartAdrs_H,L (AREA 2 Start Address High, Low) 259
3.1.84 Port0:08Ch-08Dh / Port1:28Ch-28Dh AREA3StartAdrs_H,L (AREA 3 Start Address High, Low) 259
3.1.85 Port0:090h-091h / Port1:290h-291h AREA4StartAdrs_H,L (AREA 4 Start Address High, Low) 259
3.1.86 Port0:094h-095h / Port1:294h-295h AREA5StartAdrs_H,L (AREA 5 Start Address High, Low) 259
3.1.87 Port0:082h-083h / Port1:282h-283h AREA0EndAdrs_H,L (AREA 0 End Address High, Low) 261
3.1.88 Port0:086h-087h / Port1:286h-287h AREA1EndAdrs_H,L (AREA 1 End Address High, Low) 261
3.1.89 Port0:08Ah-08Bh / Port1:28Ah-28Bh AREA2EndAdrs_H,L (AREA 2 End Address High, Low) 261

3.1.90 Port0:08Eh-08Fh / Port1:28Eh-28Fh AREA3EndAdrs_H,L (AREA 3 End Address High, Low) 261
3.1.91 Port0:092h-093h / Port1:292h-293h AREA4EndAdrs_H,L (AREA 4 End Address High, Low) 261
3.1.92 Port0:096h-097h / Port1:296h-297h AREA5EndAdrs_H,L (AREA 5 End Address High,Low) 261
3.1.93 Port0:09Fh / Port1:29Fh AREAnFIFO_Clr (AREA n FIFO Clear) 263
3.1.94 Port0:0A0h / Port1:2A0h AREA0Join_0 (AREA 0 Join 0) 264
3.1.95 Port0:0A2h / Port1:2A2h AREA1Join_0 (AREA 1 Join 0) 264
3.1.96 Port0:0A4h / Port1:2A4h AREA2Join_0 (AREA 2 Join 0) 264
3.1.97 Port0:0A6h / Port1:2A6h AREA3Join_0 (AREA 3 Join 0) 264
3.1.98 Port0:0A8h / Port1:2A8h AREA4Join_0 (AREA 4 Join 0) 264
3.1.99 Port0:0AAh / Port1:2Aah AREA5Join_0 (AREA 5 Join 0) 264
3.1.100 Port0:0A1h / Port1:2A1h AREA0Join_1 (AREA 0 Join 1) 265
3.1.101 Port0:0A3h / Port1:2A3h AREA1Join_1 (AREA 1 Join 1) 265
3.1.102 Port0:0A5h / Port1:2A5h AREA2Join_1 (AREA 2 Join 1) 265
3.1.103 Port0:0A7h / Port1:2A7h AREA3Join_1 (AREA 3 Join 1) 265
3.1.104 Port0:0A9h / Port1:2A9h AREA4Join_1 (AREA 4 Join 1) 265
3.1.105 Port0:0ABh / Port1:2Abh AREA5Join_1 (AREA 5 Join 1) 265
3.1.106 Port0:0AEh / Port1:2Aeh ClrAREAnJoin_0 (Clear AREA n Join 0) 267
3.1.107 Port0:0AFh / Port1:2Afh ClrAREAnJoin_1 (Clear AREA n Join 1) 268
3.2 Device Register Details 269
3.2.1 Port0:0B0h / Port1:N/A *D_SIE_IntStat (Device SIE Interrupt Status)* 270
3.2.2 Port0:0B3h / Port1:N/A D_BulkIntStat (Device Bulk Interrupt Status) 273
3.2.3 Port0:0B4h / Port1:N/A D_EPrIntStat (Device EPr Interrupt Status) 274
3.2.4 Port0:0B5h / Port1:N/A D_EP0IntStat (Device EP0 Interrupt Status) 276
3.2.5 Port0:0B6h / Port1:N/A D_EPaIntStat (Device EPa Interrupt Status) 278
3.2.6 Port0:0B7h / Port1:N/A D_EPbIntStat (Device EPb Interrupt Status) 278
3.2.7 Port0:0B8h / Port1:N/A D_EPcIntStat (Device EPc Interrupt Status) 278
3.2.8 Port0:0B9h / Port1:N/A D_EPdIntStat (Device EPd Interrupt Status) 278
3.2.9 Port0:0BAh / Port1:N/A D_EPeIntStat (Device EPe Interrupt Status) 278
3.2.10 Port0:0BCh-0BDh / Port1:N/A D_AlarmIN_IntStat_H,L (Device AlarmIN Interrupt Status High, Low) 280
3.2.11 Port0:0BEh-0BFh / Port1:N/A D_AlarmOUT_IntStat_H,L (Device AlarmOUT Interrupt Status High, Low) 281
3.2.12 Port0:0C0h / Port1:N/A *D_SIE_IntEnb (Device SIE Interrupt Enable)* 282
3.2.13 Port0:0C3h / Port1:N/A D_BulkIntEnb (Device Bulk Interrupt Enable) 283
3.2.14 Port0:0C4h / Port1:N/A D_EPrIntEnb (Device EPr Interrupt Enable) 284
3.2.15 Port0:0C5h / Port1:N/A D_EP0IntEnb (Device EP0 Interrupt Enable) 285
3.2.16 Port0:0C6h / Port1:N/A D_EPaIntEnb (Device EPa Interrupt Enable) 286

3.2.17 Port0:0C7h / Port1:N/A D_EPbIntEnb (Device EPb Interrupt Enable) 286
3.2.18 Port0:0C8h / Port1:N/A D_EPcIntEnb (Device EPc Interrupt Enable)..... 286
3.2.19 Port0:0C9h / Port1:N/A D_EPdIntEnb (Device EPd Interrupt Enable) 286
3.2.20 Port0:0CAh / Port1:N/A D_EPeIntEnb (Device EPe Interrupt Enable)..... 286
3.2.21 Port0:0CCh-0CDh / Port1:N/A D_AlarmIN_IntEnb_H,L (Device AlarmIN Interrupt Enable High, Low)..... 287
3.2.22 Port0:0CEh-0CFh / Port1:N/A D_AlarmOUT_IntEnb_H,L (Device AlarmOUT Interrupt Enable High, Low)..... 288
3.2.23 Port0:0D0h / Port1:N/A D_NegoControl (Device Negotiation Control)..... 289
3.2.24 Port0:0D3h / Port1:N/A D_XcvrControl (Device Xcvr Control) 291
3.2.25 Port0:0D4h / Port1:N/A D_USB_Test (Device USB_Test)..... 292
3.2.26 Port0:0D6h / Port1:N/A D_EPnControl (Device Endpoint Control)..... 294
3.2.27 Port0:0D8h / Port1:N/A D_BulkOnlyControl (Device BulkOnly Control)..... 295
3.2.28 Port0:0D9h / Port1:N/A D_BulkOnlyConfig (Device BulkOnly Configuration) 296
3.2.29 Port0:0E0h / Port1:N/A D_EP0SETUP_0 (Device EP0 SETUP 0) 298
3.2.30 Port0:0E1h / Port1:N/A D_EP0SETUP_1 (Device EP0 SETUP 1) 298
3.2.31 Port0:0E2h / Port1:N/A D_EP0SETUP_2 (Device EP0 SETUP 2) 298
3.2.32 Port0:0E3h / Port1:N/A D_EP0SETUP_3 (Device EP0 SETUP 3) 298
3.2.33 Port0:0E4h / Port1:N/A D_EP0SETUP_4 (Device EP0 SETUP 4) 298
3.2.34 Port0:0E5h / Port1:N/A D_EP0SETUP_5 (Device EP0 SETUP 5) 298
3.2.35 Port0:0E6h / Port1:N/A D_EP0SETUP_6 (Device EP0 SETUP 6) 298
3.2.36 Port0:0E7h / Port1:N/A D_EP0SETUP_7 (Device EP0 SETUP 7) 298
3.2.37 Port0:0E8h / Port1:N/A D_USB_Address (Device USB Address)..... 300
3.2.38 Port0:0EAh / Port1:N/A D_SETUP_Control(Device SETUP Control)..... 301
3.2.39 Port0:0EEh-0EFh / Port1:N/A D_FrameNumber_H,L (Device FrameNumber High, Low)..... 302
3.2.40 Port0:0F0h / Port1:N/A D_EP0MaxSize (Device EP0 Max Packet Size)..... 303
3.2.41 Port0:0F1h / Port1:N/A D_EP0Control (Device EP0 Control) 304
3.2.42 Port0:0F2h / Port1:N/A D_EP0ControlIN (Device EP0 Control IN) 306
3.2.43 Port0:0F3h / Port1:N/A D_EP0ControlOUT (Device EP0 Control OUT) 308
3.2.44 Port0:0F8h-0F9h / Port1:N/A D_EPaMaxSize_H,L (Device EPa Max Packet Size High, Low) 310
3.2.45 Port0:100h-101h / Port1:N/A D_EPbMaxSize_H,L (Device EPb Max Packet Size High, Low) 310
3.2.46 Port0:108h-109h / Port1:N/A D_EPcMaxSize_H,L (Device EPc Max Packet Size High, Low) 310
3.2.47 Port0:110h-111h / Port1:N/A D_EPdMaxSize_H,L (Device EPd Max Packet Size High, Low) 310
3.2.48 Port0:118h-119h / Port1:N/A D_EPeMaxSize_H,L (Device EPe Max Packet Size High, Low) 310
3.2.49 Port0:0FAh / Port1:N/A D_EPaConfig (Device EPa Configuration)..... 312

3.2.50 Port0:102h / Port1:N/A D_EPbConfig (Devie EPb Configuration) 312
3.2.51 Port0:10Ah / Port1:N/A D_EPcConfig (Device EPc Configuration) 312
3.2.52 Port0:112h / Port1:N/A D_EPdConfig (Devie EPd Configuration) 312
3.2.53 Port0:11Ah / Port1:N/A D_EPeConfig (Device EPe Configuration) 312
3.2.54 Port0:0FCh / Port1:N/A D_EPaControl (Device EPa Control) 314
3.2.55 Port0:104h / Port1:N/A D_EPbControl (Device EPb Control) 314
3.2.56 Port0:10Ch / Port1:N/A D_EPcControl (Device EPc Control) 314
3.2.57 Port0:114h / Port1:N/A D_EPdControl (Device EPd Control) 314
3.2.58 Port0:11Ch / Port1:N/A D_EPeControl (Device EPc Control) 314
3.2.59 Port0:120h-121h / Port1:N/A D_DescAdrs_H,L (Device Descriptor Address High, Low) 316
3.2.60 Port0:122h-123h / Port1:N/A D_DescSize_H,L (Device Descriptor Size High, Low) .. 317
3.2.61 Port0:126h / Port1:N/A D_EP_DMA_Ctrl (Device EP DMA Control) 318
3.2.62 Port0:128h-129h / Port1:N/A D_EnEP_IN_H,L (Device Enable Endpoint-IN High, Low) 319
3.2.63 Port0:12Ah-12Bh / Port1:N/A D_EnEP_OUT_H,L (Device Enable Endpoint-OUT High, Low) 320
3.2.64 Port0:12Ch-12Dh / Port1:N/A D_EnEP_IN_ISO_H,L (Device Enable Endpoint-IN Isochronous High, Low) 321
3.2.65 Port0:12Eh-12Fh / Port1:N/A D_EnEP_OUT_ISO_H,L (Device Enable Endpoint-OUT Isochronous High, Low) 322
3.3 Host Register Details 323
3.3.1 Port0:140h / Port1:340h H_SIE_IntStat_0 (Host SIE Interrupt Status 0) 324
3.3.2 Port0:141h / Port1:341h H_SIE_IntStat_1 (SIE Host Interrupt Status 1) 326
3.3.3 Port0:143h / Port1:343h H_FrameIntStat (Host Frame Interrupt Status) 328
3.3.4 Port0:144h / Port1:344h H_CHrIntStat (Host CHr Interrupt Status) 329
3.3.5 Port0:145h / Port1:345h H_CH0IntStat (Host CH0 Interrupt Status) 330
3.3.6 Port0:146h / Port1:346h H_CHaIntStat (Host CHa Interrupt Status) 332
3.3.7 Port0:147h / Port1:347h H_CHbIntStat (Host CHb Interrupt Status) 334
3.3.8 Port0:148h / Port1:348h H_CHcIntStat (Host CHc Interrupt Status) 334
3.3.9 Port0:149h / Port1:349h H_CHdIntStat (Host CHd Interrupt Status) 334
3.3.10 Port0:14Ah / Port1:34Ah H_CHeIntStat (Host CHe Interrupt Status) 334
3.3.11 Port0:150h / Port1:350h H_SIE_IntEnb_0 (Host SIE Interrupt Enable) 336
3.3.12 Port0:151h / Port1:351h H_SIE_IntEnb_1(SIE Host Interrupt Enable 1) 337
3.3.13 Port0:153h / Por1:353h H_FrameIntEnb(Host Frame Interrupt Enable) 338
3.3.14 Port0:154h / Port1:354h H_CHrIntEnb(Host CHr Interrupt Enable) 339
3.3.15 Port0:155h / Port1:355h H_CH0IntEnb(Host CH0 Interrupt Enable) 340
3.3.16 Port0:156h / Port1:356h H_CHaIntEnb (Host CHa Interrupt Enable) 341
3.3.17 Port0:157h / Port1:357h H_CHbIntEnb (Host CHb Interrupt Enable) 342
3.3.18 Port0:158h / Port1:358h H_CHcIntEnb (Host CHc Interrupt Enable) 342

3.3.19 Port0:159h / Port1:359h H_CHdIntEnb (Host CHd Interrupt Enable) 342
3.3.20 Port0:15Ah / Port1:35Ah H_CHeIntEnb (Host CHe Interrupt Enable)..... 342
3.3.21 Port0:160h / Port1:360h H_NegoControl_0 (Host NegoControl 0) 343
3.3.22 Port0:162h / Port1:362h H_NegoControl_1 (Host NegoControl 1) 346
3.3.23 Port0:164h / Por1:364h H_USB_Test (Host USB_Test)..... 347
3.3.24 Port0:170h / Port1:370h H_CH0SETUP_0 (Host CH0 SETUP 0) 349
3.3.25 Port0:171h / Port1:371h H_CH0SETUP_1 (Host CH0 SETUP 1) 349
3.3.26 Port0:172h / Port1:372h H_CH0SETUP_2 (Host CH0 SETUP 2) 349
3.3.27 Port0:173h / Port1:373h H_CH0SETUP_3 (Host CH0 SETUP 3) 349
3.3.28 Port0:174h / Port1:374h H_CH0SETUP_4 (Host CH0 SETUP 4) 349
3.3.29 Port0:175h / Port1:375h H_CH0SETUP_5 (Host CH0 SETUP 5) 349
3.3.30 Port0:176h / Port1:376h H_CH0SETUP_6 (Host CH0 SETUP 6) 349
3.3.31 Port0:177h / Port1:377h H_CH0SETUP_7 (Host CH0 SETUP 7) 349
3.3.32 Port0:17Eh-17Fh / Port1:37Eh-37Fh H_FrameNumber_H,L (Host FrameNumber High, Low) 350
3.3.33 Port0:180h / Port1:380h H_CH0Config_0(Host Channel 0 Configuration0) 351
3.3.34 Port0:181h / Port1:381h H_CH0Config_1(Host Channel 0 Configuration1) 353
3.3.35 Port0:183h / Port1:383h H_CH0MaxPktSize (Host Channel 0 Max Packet Size)..... 354
3.3.36 Port0:186h-187h / Port1:386h-387h H_CH0TotalSize_H,L (Host Channel 0 Total Size High, Low)..... 355
3.3.37 Port0:188h / Port1:388h H_CH0HubAdrs (Host Channel 0 Hub Address)..... 356
3.3.38 Port0:189h / Port1:389h H_CH0FuncAdrs (Host Channel 0 Function Address) 357
3.3.39 Port0:18Bh / Port1:38Bh H_CTL_SupportControl (Host ControlTransfer Support Control) 358
3.3.40 Port0:18Eh / Port1:38Eh H_CH0ConditionCode (Host Channel 0 Condition Code) ... 360
3.3.41 Port0:190h / Port1:390h H_CHaConfig_0(Host Channel a Configuration0) 361
3.3.42 Port0:191h / Port1:391h H_CHaConfig_1(Host Channel a Configuration1) 363
3.3.43 Port0:192h-193h / Port1:392h-393h H_CHaMaxPktSize_H,L (Host Channel a Max Packet Size High, Low)..... 364
3.3.44 Port0:194h-195h / Port1:394h-395h H_CHaTotalSize_HH,HL (Host Channel a Total Size High-High, High-Low)..... 365
3.3.45 Port0:196h-197h / Port1:396h-397h H_CHaTotalSize_LH,LL (Host Channel a Total Size Low-High, Low-Low) 365
3.3.46 Port0:198h / Port1:398h H_CHaHubAdrs (Host Channel a Hub Address)..... 367
3.3.47 Port0:199h / Port1:399h H_CHaFuncAdrs (Host Channel a Function Address) 368
3.3.48 Port0:19Ah / Port1:39Ah H_CHaBO_SupporotCtl (Host CHa Bulk Only Transfer Support Control)..... 369
3.3.49 Port0:19Bh / Port1:39Bh H_CHaBO_CSW_RcvSize (Host CHa Bulk Only Transfer Support CSW Receive Data Size)..... 371
3.3.50 Port0:19Ch / Port1:39Ch H_CHaBO_OUT_EP_Ctl (Host CHa Bulk Only Transfer Support OUT Endpoint Control) 372

3.3.51 Port0:19Dh / Port1:39Dh H_CHaBO_IN_EP_Ctl (Host CHa Bulk Only Transfer Support IN Endpoint Control) 373
3.3.52 Port0:19Eh / Port1:39Eh H_CHaConditionCode (Host Channel a Condition Code) ... 374
3.3.53 Port0:1A0h / Port1:3A0h H_CHbConfig_0(Host Channel b Configuration0) 375
3.3.54 Port0:1B0h / Port1:3B0h H_CHcConfig_0(Host Channel c Configuration0) 375
3.3.55 Port0:1C0h / Port1:3C0h H_CHdConfig_0(Host Channel d Configuration0) 375
3.3.56 Port0:1D0h / Port1:3D0h H_CHeConfig_0(Host Channel e Configuration0) 375
3.3.57 Port0:1A1h / Port1:3A1h H_CHbConfig_1(Host Channel b Configuration1) 377
3.3.58 Port0:1B1h / Port1:3B1h H_CHcConfig_1(Host Channel c Configuration1) 377
3.3.59 Port0:1C1h / Port1:3C1h H_CHdConfig_1(Host Channel d Configuration1) 377
3.3.60 Port0:1D1h / Port1:3D1h H_CHeConfig_1(Host Channel e Configuration1) 377
3.3.61 Port0:1A2h-1A3h / Port1:3A2h-3A3h H_CHbMaxPktSize_H,L (Host Channel b Max Packet Size High, Low) 379
3.3.62 Port0:1B2h-1B3h / Port1:3B2h-3B3h H_CHcMaxPktSize_H,L (Host Channel c Max Packet Size High, Low) 379
3.3.63 Port0:1C2h-1C3h / Port1:3C2h-3C3h H_CHdMaxPktSize_H,L (Host Channel d Max Packet Size High, Low) 379
3.3.64 Port0:1D2h-1D3h / Port1:3D2h-3D3h H_CHeMaxPktSize_H,L (Host Channel e Max Packet Size High, Low) 379
3.3.65 Port0:1A4h-1A5h / Port1:3A4h-3A5h H_CHbTotalSize_HH,HL (Host Channel b Total Size High-High, High-Low) 381
3.3.66 Port0:1A6h-1A7h / Port1:3A6h-3A7h H_CHbTotalSize_LH,LL (Host Channel b Total Size Low-High, Low-Low) 381
3.3.67 Port0:1B4h-1B5h / Port1:3B4h-3B5h H_CHcTotalSize_HH,HL (Host Channel c Total Size High-High, High-Low) 381
3.3.68 Port0:1B6h-1B7h / Port1:3B6h-3B7h H_CHcTotalSize_LH,LL (Host Channel c Total Size Low-High, Low-Low) 381
3.3.69 Port0:1C4h-1C5h / Port1:3C4h-3C5h H_CHdTotalSize_HH,HL (Host Channel d Total Size High-High, High-Low) 381
3.3.70 Port0:1C6h-1C7h / Port1:3C6h-3C7h H_CHdTotalSize_LH,LL (Host Channel d Total Size Low-High, Low-Low) 381
3.3.71 Port0:1D4h-1D5h / Port1:3D4h-3D5h H_CHeTotalSize_HH,HL (Host Channel e Total Size High-High, High-Low) 381
3.3.72 Port0:1D6h-1D7h / Port1:3D6h-3D7h H_CHeTotalSize_LH,LL (Host Channel e Total Size Low-High, Low-Low) 381
3.3.73 Port0:1A8h / Port1:3A8h H_CHbHubAdrs (Host Channel b Hub Address) 383
3.3.74 Port0:1B8h / Port1:3B8h H_CHcHubAdrs (Host Channel c Hub Address) 383
3.3.75 Port0:1C8h / Port1:3C8h H_CHdHubAdrs (Host Channel d Hub Address) 383
3.3.76 Port0:1D8h / Port1:3D8h H_CHeHubAdrs (Host Channel e Hub Address) 383
3.3.77 Port0:1A9h / Port1:3A9h H_CHbFuncAdrs (Host Channel b Function Address) 384
3.3.78 Port0:1B9h / Port1:3B9h H_CHcFuncAdrs (Host Channel c Function Address) 384
3.3.79 Port0:1C9h / Port1:3C9h H_CHdFuncAdrs (Host Channel d Function Address) 384

3.3.80 Port0:1D9h / Port1:3D9h H_CHeFuncAdrs (Host Channel d Function Address) 384
3.3.81 Port0:1AAh-1ABh / Port1:3AAh-3ABh H_CHbInterval_H,L (Host Channel b Interval High, Low) 385
3.3.82 Port0:1BAh-1BBh / Port1:3BAh-3BBh H_CHcInterval_H,L (Host Channel c Interval High, Low) 385
3.3.83 Port0:1CAh-1CBh / Port1:3CAh-3CBh H_CHdInterval_H,L (Host Channel d Interval High, Low) 385
3.3.84 Port0:1DAh-1DBh / Port1:3DAh-3DBh H_CHeInterval_H,L (Host Channel e Interval High, Low) 385
3.3.85 14Ch H_CHbTranPause(Host Channel b Transaction Pause) 386
3.3.86 15Ch H_CHcTranPause(Host Channel c Transaction Pause) 386
3.3.87 16Ch H_CHdTranPause(Host Channel d Transaction Pause) 386
3.3.88 17Ch H_CHeTranPause(Host Channel e Transaction Pause) 386
3.3.89 Port0:1AEh / Port1:3AEh H_CHbConditionCode (Host Channel b Condition Code) .. 387
3.3.90 Port0:1BEh / Port1:3BEh H_CHcConditionCode (Host Channel c Condition Code)... 387
3.3.91 Port0:1CEh / Port1:3CEh H_CHdConditionCode (Host Channel d Condition Code).. 387
3.3.92 Port0:1DEh / Port1:3DEh H_CHeConditionCode (Host Channel e Condition Code).. 387

Appendix A: Connection to Little-endian CPU 389

1. Functions

This section describes various LSI functions.

Register rules are described below.

- Register separation by port

  This LSI has register sets with identical functions for each of the two USB ports.

  (Port 1 lacks device function registers.)

  Where a distinction needs to be made, this is indicated in the form “Register name + (Port number).”

  E.g.: “MainIntStat(0) register”

- Registers for a single address are indicated as follows: “Register name + Register.”

  E.g.: “MainIntStat register”

- Individual bits are indicated as follows: “Register name.Bit name + Bit” or “Bit name + Bit.”

  E.g.: “MainIntStat.CPU_IntStat bit”

- Registers for each device endpoint are indicated as follows: “D_EPx{x=...} Register.”

  E.g.: “D_EPx{x=0,a-e}IntStat register”

- Registers for each host channel are indicated as follows: H_CHx{x=...} Register.

  (e.g., “H_CHx{x=0,a-e}IntStat register”)

1.1 USB Ports

This LSI has two USB ports, designated USB Port 0 and USB Port 1. USB Port 0 can be used for either USB host or USB device functions. USB Port 1 can be used for USB host functions only. Each port has individual USB functions and control interfaces. Chip-specific functions not assigned to either port are controlled by shared registers mirrored at each port control interface.

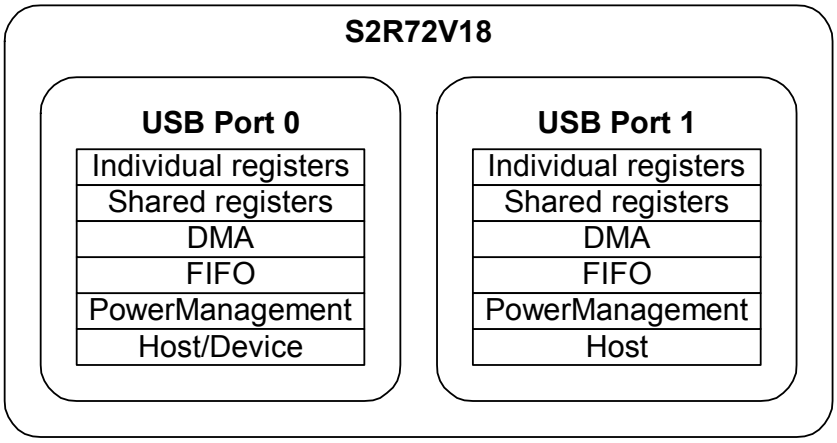


Fig. 1-1 Block image

1.2 USB Device/Host Selection

Either USB device or host functions can be selected when using USB Port 0. For USB Port 1, only USB host function can be selected.

Selecting the USB device function (Device mode) enables shared register and device register bits and functions.

Selecting the USB host function (Host mode) enables shared register and host register bits and functions.

1.2.1 USB Device/Host Function Selection Procedure

Clearing the HostDeviceSel(0).HOSTxDEVICE bit for USB Port 0 enables the device function, and setting this bit enables the host function. It's possible to switch between Host and Device modes when the Port 0 power management state is SLEEP. Table 1-1 shows the USB device/host selection settings.

Table 1-1 USB Device/Host function selection settings

Item	Register/bit	Description
USB device/host selection	HostDeviceSel(0).HOSTxDEVICE	Selects USB Device or Host mode. Allows use of registers and functions corresponding to the mode selected. 1: Host mode 0: Device mode

1.2.2 USB Port State Change Detection Status

This LSI includes a function for detecting USB port states.

This function can be used in both SLEEP and ACTIVE states (refer to "1.5 Power Management Functions").

1.2.2.1 USB Port State Change Detection Status Usage Examples

The examples given here use device port or host port change status.

1.2.2.1.1 Device Port Change Status

This status indicates that a change in the VBUS_0 terminal state for USB Port 0 in Device mode.

Table 1-2 lists registers for device port change status.

Table 1-2 Device port change status registers

Item	Register/bit	Description
VBUS terminal change status	USB_DeviceIntStat(0).VBUS_Changed	Indicates a change in the device port VBUS_0 terminal state.
VBUS terminal change status enable	USB_DeviceIntEnb(0).EnVBUS_Changed	Permits/prevents assertion of the MainIntStat.USB_DeviceIntStat bit by USB_DeviceIntStat.VBUS_Changed.
VBUS terminal state	D_USB_Status(0).VBUS	Indicates the device port VBUS_0 terminal state.

Steps (1), (2), and (4) to (7) are processed by the F/W when using device port change status.

(1) Clear the VBUS terminal change status. Set the MainIntEnb.EnUSB_DeviceIntStat bit. (When using interrupt)

(2) Set the VBUS terminal change status enable.

(3) The VBUS terminal change status is issued when VBUS is supplied.

(4) Check the VBUS terminal change status.

(5) Clear the VBUS terminal change status.

(6) Clear the VBUS terminal change status enable.

(7) Check the VBUS terminal state. VBUS is deemed to be supplied (i.e., a host or hub is connected) if the VBUS terminal state is "1."

1.2.2.1.2 Host Port Change Status

This status indicates the state of the driver controlling the VBUS power supply for USB Port 0 Host mode or USB Port 1.

Table 1-3 lists the registers for host port change status.

Table 1-3 Host port change status registers

Item	Register/bit	Description
VBUS error detection status	USB_HostIntStat.VBUS_Err	Indicates a VBUS error.
VBUS error detection status enable	USB_HostIntEnb.EnVBUS_Err	Permits/prevents assertion of the MainIntStat.USB_HostIntStat bit by USB_HostIntStat.VBUS_Err.
VBUS enable	H_USB_Control.VBUS_Enb	Enables an external USB power switch.
Host port VBUS state	H_USB_Status.VBUS_State	Indicates the host port VBUS state (normal/error).

VBUS enable must be turned off immediately to stop the VBUS drive if a VBUS error detection status is detected.

1.2.2.2 Signal Line Change Status Usage

This status indicates that the DP terminal (DP_0, DP_1) or DM terminal (DM_0, DM_1) state has changed for USB Port 0 Host mode or USB Port 1.

Table 1-4 lists the registers for signal line change status.

Table 1-4 Signal line change status registers

Item	Register/bit	Description
Signal line change status	USB_HostIntStat.LineStateChanged	Indicates a change in the host port DP or DM terminal state.
Signal line change status interrupt enable	USB_HostIntEnb.EnLineStateChanged	Permits/prevents assertion of the MainIntStat.USB_HostIntStat bit by the signal line change status.
VBUS error detection status	USB_HostIntStat.VBUS_Err	Monitors the VBUSFLG terminal (VBUSFLG_0, VBUSFLG_1) and indicates errors in the VBUS power supply.
VBUS error detection status interrupt enable	USB_HostIntEnb.EnVBUS_Err	Permits/prevents assertion of the MainIntStat.USB_HostIntStat bit by the VBUS error detection status.
Operation mode	H_XcvrControl.OpMode[1:0]	Sets the transceiver macro operation mode.
VBUS enable	H_USB_Control.VBUS_Enb	Outputs a signal from the VBUSEN terminal (VBUSEN_0, VBUSEN_1) enabling an external USB power switch.
Host port VBUS state	H_USB_Status.VBUS_State	Indicates the host port VBUSFLG terminal (VBUSFLG_0, VBUSFLG_1) state (normal/error).
Host port signal line state	H_USB_Status.LineState[1:0]	Indicates the signal state on the USB cable.
Host state change execute	H_NegoControl_0.AutoMode[3:0]	Sets the host state change.

The signal line change status differs from the connection detection status when used as a USB host. This special status allows detection of host port signal line changes when not using the USB host function – i.e., in SLEEP state (refer to "1.5 Power Management Functions"). This status is frequently asserted when USB_HostIntEnb.EnLineStateChanged is enabled with the USB host operating. For this reason, USB_HostIntEnb.EnLineStateChanged should be disabled when operating in USB host mode.

VBUS enable should be turned off immediately to stop the VBUS drive if VBUS error detection status is detected, except when operating in USB host mode.

Steps (1) to (4) and (6) to (9) are processed by the firmware when using signal line change status.

(1) Clear the signal line change status and VBUS error detection status.

(2) Set the signal line change status enable and VBUS error detection status enable. Set the MainIntEnb.EnUSB_HostIntStat bit. (When using interrupt)

(3) Set the H_XcvrControl register to 0x93.

(4) Switch on VBUS enable.

(5) The line state changes from SE0. A signal line change status is issued if a device is connected to the host port.

(6) Check the signal line change status.

(7) Clear the signal line change status.

(8) Clear the signal line change status enable.

(9) Check the host port signal line state. A device is deemed to be connected to the host port if the state is “01” or “10.”

The H_XcvrControl register and VBUS_Enb bit retain the values set here when Host mode is subsequently selected. These settings are automatically set by the hardware to suit the host state after the firmware sets appropriate codes in the host state change execution register. For details, refer to “1.4.10 Host State Management Support Function.”

1.3 USB Device Control

This section describes the USB device functions of this LSI.

1.3.1 Endpoints

This LSI features one endpoint (EP0) for control transfers and five general endpoints – EPa, EPb, EPc, EPd, and Epe – for use as bulk, interrupt, or isochronous transfer endpoints. It permits up to 15 IN endpoints and up to 15 OUT endpoints, excluding endpoint EP0, using the alarm endpoint function to generate alarms when a transaction is issued by the USB host. The alarm endpoint function can notify firmware by returning a NAK response for transactions issued to endpoints other than the active endpoints set in EPa, EPb, EPc, EPd, and EPe.

The LSI hardware provides endpoints to control transactions. It does not provide functions for managing USB-defined interfaces, which should be implemented in firmware. The endpoints should be set and combined appropriately to establish USB-defined interfaces in accordance with device-specific descriptor definitions.

Each endpoint includes fixed basic settings determined by the USB-defined interface and variable control items and status for controlling each transfer. The basic settings should be made at the appropriate times – for example, when the chip is initialized or the USB-defined interface switched.

Table 1-5 shows the basic settings for endpoint EP0 (default control pipe).

Endpoint EP0 shares register sets and FIFO areas for the in and out directions. The appropriate data transaction direction should be set by the firmware upon execution for the EP0 data stage and status stage.

The transaction can be executed by linking to a FIFO area described later. Reserve the FIFO area with AREAx{x=0-5}StartAdrs_H,L and AREAx{x=0-5}EndAdrs_H,L, then initialize with AREAnFIFO_Clr before setting AREAx{x=0-5}Join_1.JoinEP0CH0. Data transfers using this FIFO area are not possible until linked to this join process.

Table 1-5 Endpoint EP0 basic settings

Item	Register/bit	Description
Max packet size	D_EP0MaxSize	Sets the maximum packet size to 8, 16, 32, or 64 for FS operation and to 64 for HS operations. Use with endpoint EP0 assigned to FIFO AREA0.
FIFO area	AREAx{x=0-5}StartAdrs_H,L AREAx{x=0-5}EndAdrs_H,L	Specifies the FIFO address corresponding to the AREA0 area. The FIFO area must be at least as large as the max packet size.
FIFO linking	AREAx{x=0-5}Join_1.JoinEP0CH0	Links endpoint EP0 to the FIFO area to allow data transfer for EP0.

Table 1-6 shows the basic settings for the general endpoints (EPa, EPb, EPc, EPd, and EPe). The transaction direction and endpoint number can be set as desired for these endpoints, enabling independent use of up to five endpoints. Make the appropriate settings, given the USB-defined interface definition details, to enable the establishment of a USB-defined interface.

Transactions can be executed by linking endpoints EPa, EPb, EPc, EPd, and EPe to the corresponding FIFO areas. Reserve FIFO areas using the AREAx{x=0-5}StartAdrs_H,L and AREAx{x=0-5}EndAdrs_H,L registers and initialize FIFO areas using AREAnFIFO_Clr before setting the AREAx{x=0-5}Join_1 register as required. Transactions cannot be executed for endpoints for which this join processing has not been performed. Avoid joining endpoints that will not be used.

Table 1-6 General endpoint basic settings

Item	Register/bit	Description
Transaction direction	D_EPx{x=a-e}Config_0.INxOUT	Sets the transfer direction for each endpoint.
Max packet size	D_EPx{x=a-e}MaxSize_H, D_EPx{x=a-e}MaxSize_L	Sets the maximum packet size to 8, 16, 32, 64, or 512 bytes. Endpoints used for bulk transfers should be set to 8, 16, 32, or 64 bytes for FS mode or 512 bytes for HS mode. Endpoints used for isochronous transfers should be set to between 1 and 1,023 bytes for FS mode or between 1 and 1,024 bytes for HS mode.
Endpoint number	D_EPx{x=a-e}Config_0.EndpointNumber	Sets the endpoint numbers as desired within the range 0x1 to 0xF.
Toggle mode	D_EPx{x=a-e}Config_0.IntEP_Mode	Sets the operating mode for interrupt transfers. Endpoints for bulk transfer should be set to “0,” regardless of direction. Set the toggle sequence mode for in-direction endpoints and set to “1” for out-direction endpoints with interrupt transfers.
Isochronous mode	D_EPx{x=a-e}Config_0.ISO	Set to “1” for endpoints used for isochronous transfers.
FIFO area	AREAx{x=0-5}StartAdrs_H,L AREAx{x=0-5}EndAdrs_H,L	Specify and reserve the area address. The areas should be at least as large as the linked endpoint maximum packet sizes. The FIFO size may affect transfer throughput.
FIFO linking	AREAx{x=0-5}Join_1	Links the FIFO areas to endpoints.

Table 1-7 gives the basic settings for alarm endpoints. Alarm endpoints are not assigned at this time to the general endpoints, but provide endpoints defined by the USB-defined interface. Make the appropriate settings, given the USB-defined interface definition details, to enable establishment of a USB-defined interface.

These alarm endpoints do not require FIFO areas.

Table 1-7 Alarm endpoint basic settings

Item	Register/bit	Description
Alarm endpoint enable	D_EnEP_IN_H.EnEPn{n=8-15}IN, D_EnEP_IN_L.EnEPn{n=1-7}IN, D_EnEP_OUT_H.EnEPn{n=8-15}OUT, D_EnEP_OUT_L.EnEPn{n=1-7}OUT	Enables the alarm endpoint.
Isochronous mode	D_EnEP_IN_ISO_H.EnEPn{n=8-15}IN_ISO, D_EnEP_IN_ISO_L.EnEPn{n=1-7}IN_ISO, D_EnEP_OUT_ISO_H.EnEPn{n=8-15}OUT_ISO, D_EnEP_OUT_ISO_L.EnEPn{n=1-7}OUT_ISO	Set to “1” when setting to isochronous transfer mode. Set to “0” for endpoints set for bulk or interrupt transfers.

1.3.2 Transactions

This LSI provides transaction execution functions using the hardware and interfaces for executing transactions with the firmware. The interface with the firmware is installed as an interrupt signal asserted by the control register and status register or status. For details of the settings asserting interrupts using status, refer to “2. Registers.”

This LSI issues status information to the firmware for each transaction. However, the firmware need not control each transaction. The LSI refers to the FIFO when responding to a transaction and automatically performs processing using the data quantity or free space quantity to determine whether data transfers are possible.

For an out endpoint, for example, the firmware reads data from the FIFO using the CPU interface (DMA read or register read) and creates free space in the FIFO to enable continuous and automatic execution of the out transaction. For an in endpoint, the firmware writes data to the FIFO using the CPU interface (DMA write or register write) and creates active data in the FIFO to enable continuous and automatic execution of the in transaction.

Table 1-8 gives the control items and status for endpoint EP0 transaction control.

Table 1-8 Endpoint EP0 control items and status

Item	Register/bit	Description
Transaction direction	D_EP0Control.INxOUT	Sets the transfer direction for the data and status stages.
Descriptor reply enable	D_EP0Control.ReplyDescriptor	Activates automatic descriptor response.
Descriptor reply address	D_DescAdrs_H, DescAdrs_L	Specifies the initial FIFO address for data returned using the automatic descriptor response.
Descriptor size	D_DescSize_H, DescSize_L	Specifies the data size returned using the automatic descriptor response.
Control prohibit	D_SETUP_Control.ProtectEP0	Setting this bit blocks access to the EP0ControlIN and EP0ControlOUT register ForceNAK and ForceSTALL bits. This bit is set by the LSI hardware when the RcvEP0SETUP status is established and can be cleared by CPU register access.
Short packet send enable	D_EP0ControlIN.EnShortPkt	Enables transmission of short packets smaller than the max packet size. Cleared on completion of the in transaction sending the short packet.
Toggle sequence bit	D_EP0ControlIN.ToggleStat, D_EP0ControlOUT.ToggleStat	Indicates the toggle sequence bit status. Automatically initialized by the SETUP stage.
Toggle set	D_EP0ControlIN.ToggleSet, D_EP0ControlOUT.ToggleSet	Sets the toggle sequence bit.
Toggle clear	D_EP0ControlIN.ToggleClr, D_EP0ControlOUT.ToggleClr	Clears the toggle sequence bit.
Forced NAK response	D_EP0ControlIN.ForceNAK, D_EP0ControlOUT.ForceNAK	Returns NAK to in or out transactions (including PING), regardless of FIFO data size or free space size.
STALL response	D_EP0ControlIN.ForceSTALL, D_EP0ControlOUT.ForceSTALL	Returns STALL to in or out transactions (including PING).
Automatic ForceNAK set	D_EP0ControlOUT.AutoForceNAK	Sets the D_EP0ControlOUT.ForceNAK bit after each out transaction is completed.
SETUP receipt status	USB_DeviceIntStat.RcvEP0SETUP	Indicates that a SETUP transaction has been performed.
Transaction status	D_EP0IntStat.OUT_ShortACK, D_EP0IntStat.IN_TranACK, D_EP0IntStat.OUT_TranACK, D_EP0IntStat.IN_TranNAK, D_EP0IntStat.OUT_TranNAK, D_EP0IntStat.IN_TranErr, D_EP0IntStat.OUT_TranErr	Indicates transaction results.
Descriptor reply data stage completion status	D_EP0IntStat.DescriptorCmp	Indicates the completion of an automatic descriptor response data stage.

Table 1-9 shows the control items and status for general endpoint EPa, EPb, EPc, EPd, and EPe transaction processing.

Table 1-9 General endpoint control items and status

Item	Register/bit	Description
Automatic ForceNAK set	D_EPx{x=a-e}Control.AutoForceNAK	Sets the endpoint D_EPx{x=a-e}Control.ForceNAK bit after each out transaction is completed.
Short packet send enable	D_EPx{x=a-e}Control.EnShortPkt	Enables the transmission of short packets smaller than the max packet size for in transactions. Cleared on completion of the in transaction sending the short packet.
Automatic ForceNAK setting prohibition using short packet receipt	D_EPx{x=a-e}Control.DisAF_NAK_Short	Prevents execution of the function (*) setting the endpoint D_EPx{x=a-e}Control.ForceNAK bit automatically on receiving a short packet in the out transaction. *: Active if not prevented by this bit.
Toggle sequence bit	D_EPx{x=a-e}Control.ToggleStat	Indicates the toggle sequence bit status.
Toggle set	D_EPx{x=a-e}Control.ToggleSet	Sets the toggle sequence bit.
Toggle clear	D_EPx{x=a-e}Control.ToggleClr	Clears the toggle sequence bit.
Forced NAK response	D_EPx{x=a-e}Control.ForceNAK	Returns NAK to transactions regardless of FIFO data size or free space size.
STALL response	D_EPx{x=a-e}Control.ForceSTALL	Returns STALL to transactions.
Transaction status	D_EPx{x=a-e}IntStat.OUT_ShortACK, D_EPx{x=a-e}IntStat.IN_TranACK, D_EPx{x=a-e}IntStat.OUT_TranACK, D_EPx{x=a-e}IntStat.IN_TranNAK, D_EPx{x=a-e}IntStat.OUT_TranNAK, D_EPx{x=a-e}IntStat.IN_TranErr, D_EPx{x=a-e}I~tStat.OUT_TranErr	Indicates transaction results.

1.3.2.1 SETUP Transactions

This LSI executes SETUP transactions addressed to endpoint EP0 on its own node regardless of conditions. (The USB function must be enabled by the D_NegoControl.ActiveUSB bit.)

When a SETUP transaction is issued, the contents of all data packets (8 bytes) are placed in registers D_EP0SETUP_0 to D_EP0SETUP_7 and an ACK response returned. RcvEP0SETUP status is issued to the firmware, except for SetAddress() requests.

This LSI does not respond to or issue status information if an error occurs during a SETUP transaction.

On completion of a SETUP transaction, the LSI sets the D_EP0ControlIN and D_EP0ControlOUT register ForceNAK bits and clears the ForceSTALL bit. The ToggleStat bit and D_SETUP_Control.ProtectEP0 bit are also set. When the firmware has completed the endpoint EP0 settings and is ready to proceed to the next stage, the SETUP_Control.ProtectEP0 bit should be cleared and the corresponding direction ForceNAK bits cleared for the D_EP0ControlIN and D_EP0ControlOUT registers.

Fig. 1-2 illustrates the SETUP transaction. (a) The host issues a SETUP token addressed to endpoint 0 on this node. (b) The host then sends an 8-byte data packet. The LSI writes this data to registers D_EP0SETUP_0 to D_EP0SETUP_7. (c) The LSI automatically returns an ACK response. The register set automatically is set, and status information is issued to the firmware.

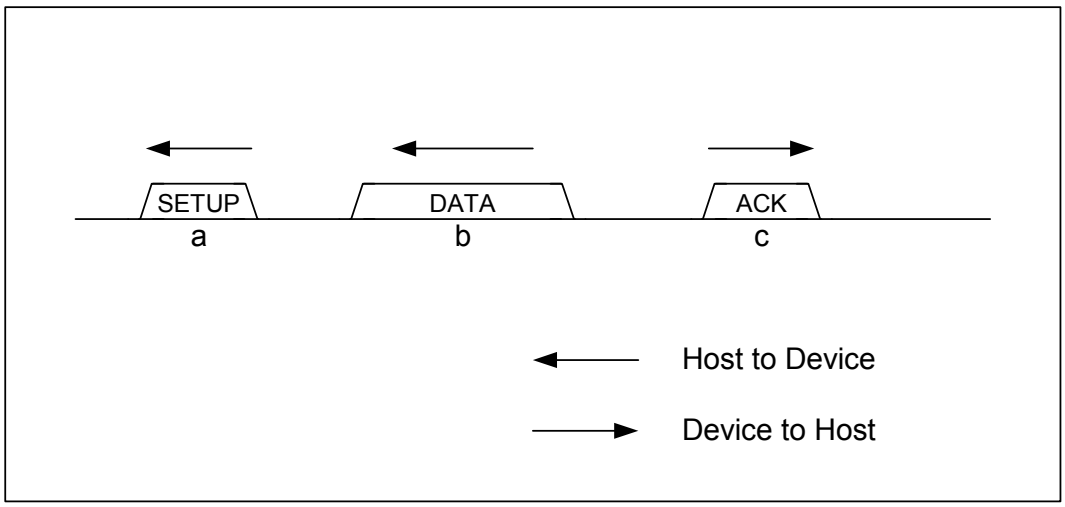


Fig. 1-2 SETUP transaction

1.3.2.2 Bulk/Interrupt OUT Transactions

For bulk or interrupt OUT transactions, data receipt starts, provided the FIFO free space exceeds the max packet size.

The LSI completes the transaction and returns an ACK or NYET response once all data has been correctly received for bulk or interrupt OUT transactions. The corresponding endpoint OUT_TranACK status notification (D_EPx{x=0,a-c}IntStat.OUT_TranACK bit) is then issued to the firmware. The FIFO is updated and area reserved with the data received.

For bulk or interrupt OUT transactions, the OUT_ShortACK status notification (D_EPx{x=0,a-e}IntStat.OUT_ShortACK bit) is also issued on receipt of all short packet data in addition to the transaction completion processing described above. The endpoint D_EPx{x=a-e}Control.ForceNAK bit is set once the D_EPx{x=a-e}Control.DisAF_NAK_Short bit is cleared.

If a toggle mismatch occurs for bulk or interrupt OUT transactions, an ACK response is returned, but no status information is issued. The FIFO is not updated.

If an error occurs for bulk or interrupt OUT transactions, no response is returned for the transaction, and OUT_TranErr status notification (D_EPx{x=0,a-e}IntStat.OUT_TranErr bit) is issued. The FIFO is not updated.

If not all data was received for bulk or interrupt OUT transactions, a NAK response is returned for the transaction. OUT_TranNAK status notification (D_EPx{x=0,a-e}IntStat.OUT_TranNAK bit) is issued. The FIFO is not updated.

Fig. 1-3 illustrates a bulk or interrupt OUT transaction upon completion. (a) The host issues an OUT token addressed to the OUT-direction endpoint in this node. (b) The host then sends data within the max packet size. The LSI writes this data to the corresponding endpoint FIFO. (c) The LSI returns an ACK response automatically on receiving the data. The automatically set register is set and status information issued to the firmware.

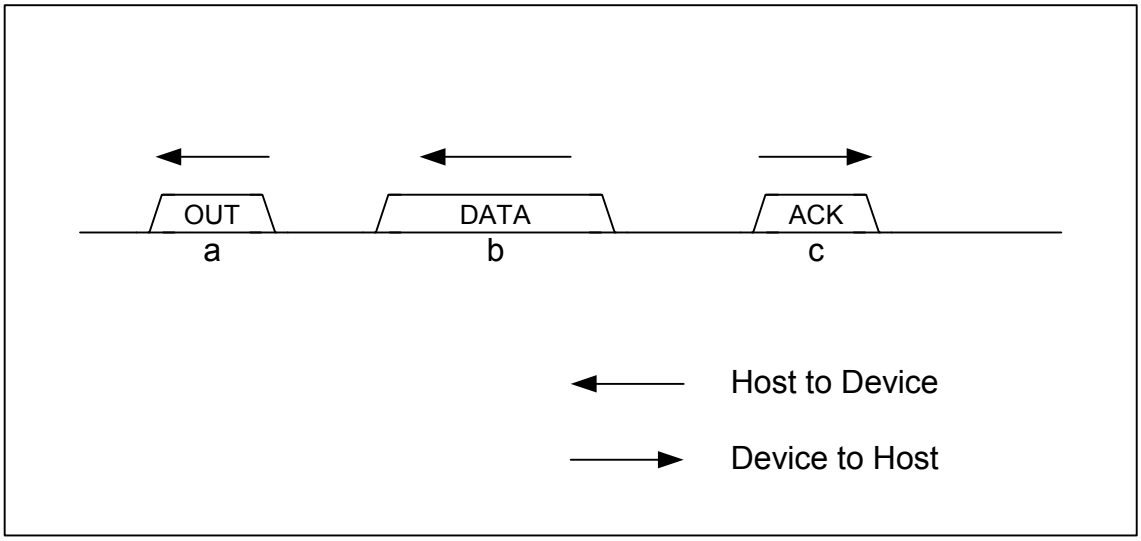


Fig. 1-3 OUT transaction

1.3.2.3 Isochronous OUT Transactions

For isochronous OUT transactions, data receipt starts, provided the FIFO free space exceeds the max packet size. Satisfactory throughput is provided by assigning approximately twice as much FIFO as the max packet size to enable receiving while creating free space by reading FIFO data using register reading or DMA reading via the CPU interface.

The corresponding endpoint OUT_TranACK status notification (EPx{x=a-e}IntStat.OUT_TranACK bit) is issued to the firmware when all data has been correctly received for an isochronous OUT transaction. The FIFO is then updated and the area reserved with the data received.

For isochronous OUT transactions, the OUT_ShortACK status notification (EPx{x=a-e}IntStat.OUT_ShortACK bit) is also issued on receipt of all data smaller than the max packet size, in addition to the transaction completion processing described above. Clearing the EPx{x=a-e}Control.DisAF_NAK_Short bit sets the EPx{x=a-e}ForceNAK bit for that endpoint.

If an error occurs for isochronous OUT transactions, the data is not received, and the FIFO is not updated. OUT_TranErr status notification (EPx{x=a-e}IntStat.OUT_TranErr bit) is issued.

OUT_TranNAK status notification (EPx{x=a-e}IntStat.OUT_TranNAK bit) is issued if the data for one packet is not fully received for isochronous OUT transactions. The FIFO is not updated.

1.3.2.4 Bulk/Interrupt IN Transactions

For IN-direction bulk and interrupt endpoints, a data packet is returned in response to the IN transaction if the FIFO contains data corresponding to the max packet size or if short packet transmission is permitted by the firmware.

Short packet (including data length zero packets) transmission permission is given by setting the D_EP0ControlIN.EnShortPkt bit or D_EPx{x=a-e}Control.EnShortPkt bit. When a short packet is transmitted, new data should not be written to that endpoint FIFO from after the transmission is permitted until the transaction is completed.

The D_EP0ControlIN.ForceNAK bit is set for endpoint EP0 once the IN transaction sending the short packet is completed.

When ACK is received for the IN transaction returning data, the transaction is completed and IN_TranACK status notification (D_EPx{x=0,a-e}IntStat.IN_TranACK bit) is issued. The FIFO is then updated and the area freed with the data transmitted.

If ACK is not received for the IN transaction returning data, the transaction is assumed to have failed, and IN_TranErr status notification (D_EPx{x=0,a-e}IntStat.IN_TranErr bit) is issued. The FIFO is not updated, and the area is not freed.

If the FIFO does not contain data corresponding to the max packet size and short packet transmission is not permitted for bulk and interrupt IN-direction endpoints, a NAK response is returned to the IN transaction and IN_TranNAK status notification (D_EPx{x=0,a-e}IntStat.IN_TranNAK bit) is issued to the firmware. The FIFO is not updated, and the area is not freed.

Fig. 1-4 illustrates a bulk or interrupt IN transaction upon completion. (a) The host issues an IN token addressed to the IN-direction endpoint in this node. (b) The LSI then sends data within the max packet size if a response to this IN transaction is possible. (c) The host returns an ACK response. The LSI sets the register set automatically on receiving the ACK, and issues a status to the firmware.

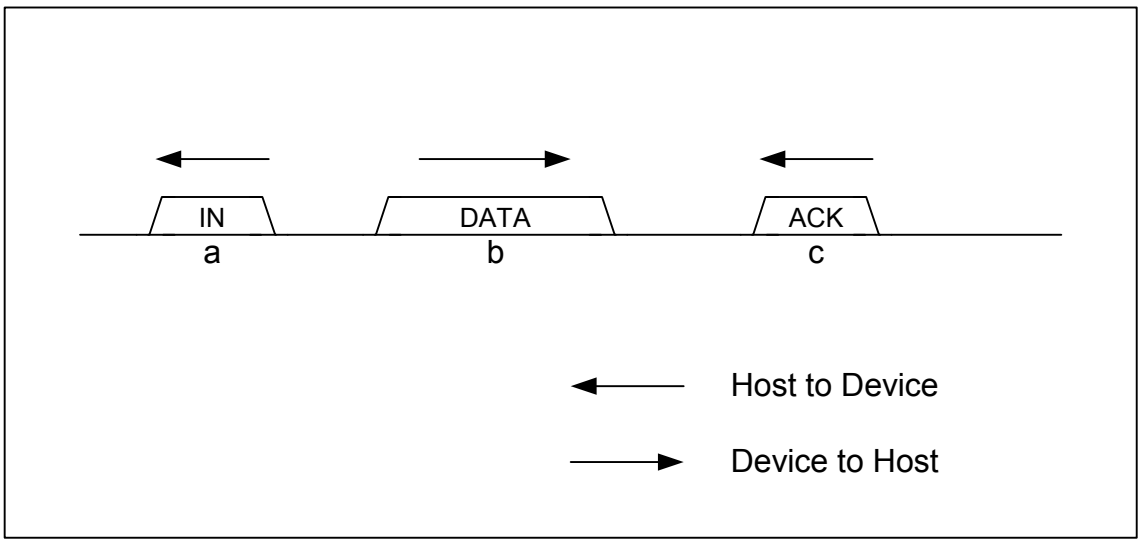


Fig. 1-4 IN transaction

1.3.2.5 Isochronous IN Transaction

For IN-direction isochronous endpoints, a data packet is returned in response to the IN transaction if the FIFO contains data corresponding to the max packet size or if short packet transmission is permitted by the firmware.

Short packet (including data length zero packets) transmission permission is given by setting the EPx{x=a-e}Control.EnShortPkt bit. If a short packet is transmitted, new data should not be written to that endpoint FIFO from after the transmission is permitted until the transaction is completed.

When the data packet is returned to the isochronous IN transaction, the transaction is completed and IN_TranACK status notification (EPx{x=a-e}IntStat.IN_TranACK bit) is issued to the firmware. The FIFO is then updated and the area freed with the data transmitted.

If the FIFO does not contain data corresponding to the max packet size and short packet transmission is not permitted for isochronous IN-direction endpoints, a response is returned to the IN transaction with a zero length data packet, and IN_TranNAK status notification (EPx{x=a-e}IntStat.IN_TranNAK bit) is issued to the firmware. The FIFO is not updated, and the area is not freed.

1.3.2.6 PING Transactions

For bulk OUT-direction endpoints, PING transactions are executed during HS operations.

If the corresponding endpoint FIFO free space exceeds the max packet size, an ACK response is returned to the PING transaction. Status information is not issued to the firmware.

If the corresponding endpoint FIFO free space is less than the max packet size, a NAK response is returned to the PING transaction. OUT_TranNAK status notification (D_EPx{x=0,a-e}IntStat.OUT_TranNAK bit) is not issued to the firmware.

The FIFO is never updated for PING transactions.

Fig. 1-5 illustrates the ACK response to a PING transaction. (a) The host issues a PING token addressed to the OUT-direction endpoint in this node. (b) The LSI returns an ACK response to the PING transaction if FIFO includes free space equivalent to the max packet size. Status information is issued to the firmware.

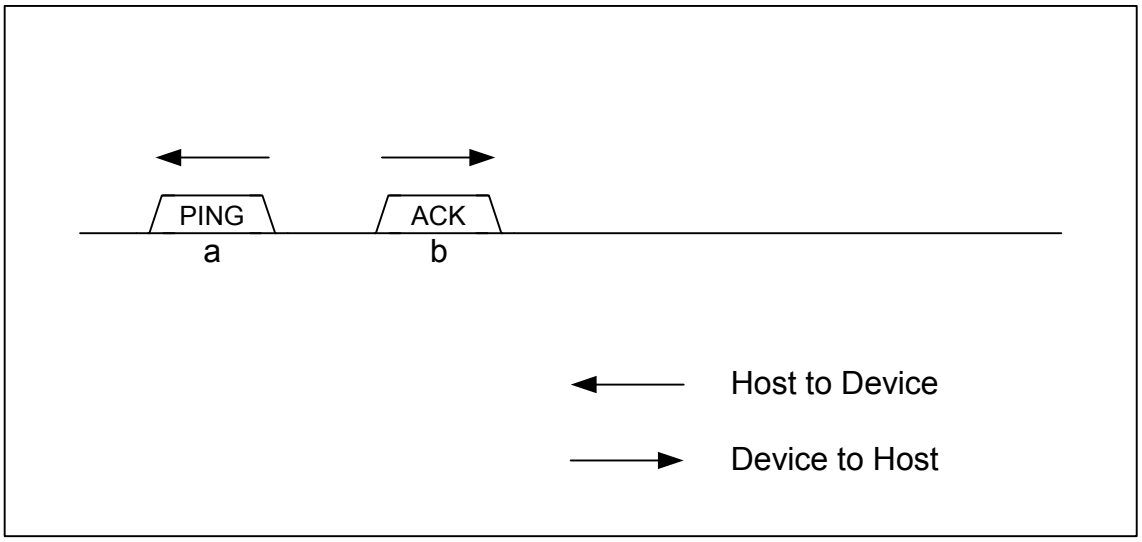


Fig. 1-5 PING transaction

1.3.3 Control Transfers

Control transfers for endpoint EP0 are controlled in conjunction with the individual transactions, with the exception of SetAddress() requests. SetAddress() requests are processed automatically using the automatic address setup function described later.

Fig. 1-6 illustrates a control transfer for when the data stage is in the OUT direction. (a) The host starts the control transfer based on the SETUP transaction. The device firmware analyzes request specifics and prepares to respond to the data stage. (b) The host issues an OUT transaction, the data stage is performed, and the device receives the data. (c) The host issues an IN transaction, the status stage is performed, and the device returns a data length zero packet.

Control transfers without a data stage are performed without the data stage shown in this example.

The transition to the status stage is triggered by the host issuing a transaction in the direction opposite to the data stage. The firmware should be used to monitor the IN_TranNAK status (D_EP0IntStat.IN_TranNAK bit) and serve as the trigger to change from the data stage to status stage.

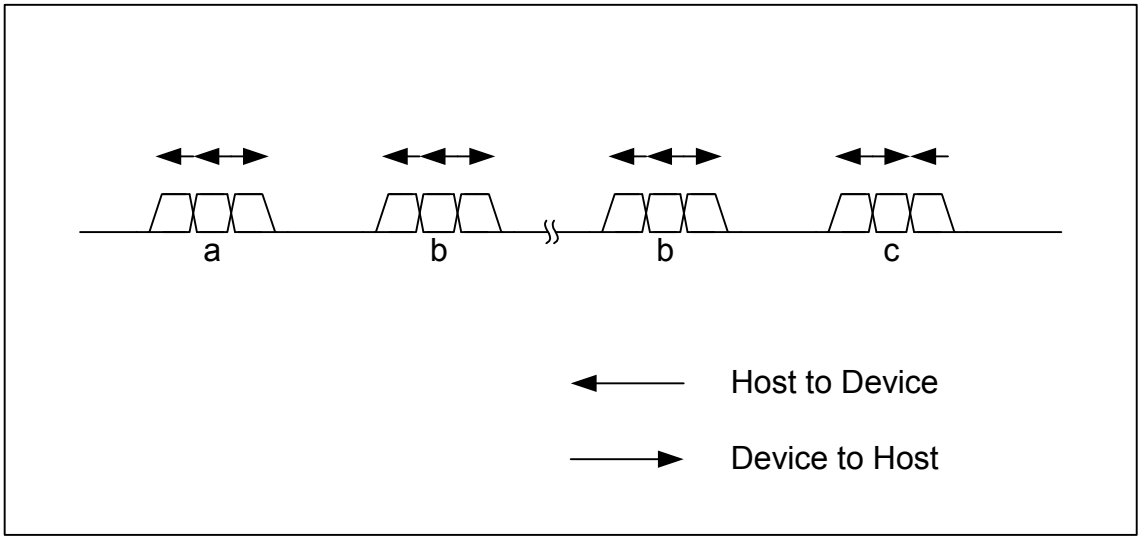


Fig. 1-6 Control transfers when data stage is OUT

Fig. 1-7 illustrates a control transfer for when the data stage is in the IN direction. (a) The host starts the control transfer based on the results of the SETUP transaction. The device firmware analyzes the request specifics and prepares to respond to the data stage. (b) The host issues an IN transaction, the data stage is performed, and the device sends the data. (c) The host issues an OUT transaction, the data stage is performed, and the device returns an ACK response.

The transition to the status stage is triggered by the host issuing a transaction in the direction opposite to the data stage. The firmware should be used to monitor the OUT_TranNAK status (D_EP0IntStat.OUT_TranNAK bit) and serve as the trigger to transition from the data stage to the status stage.

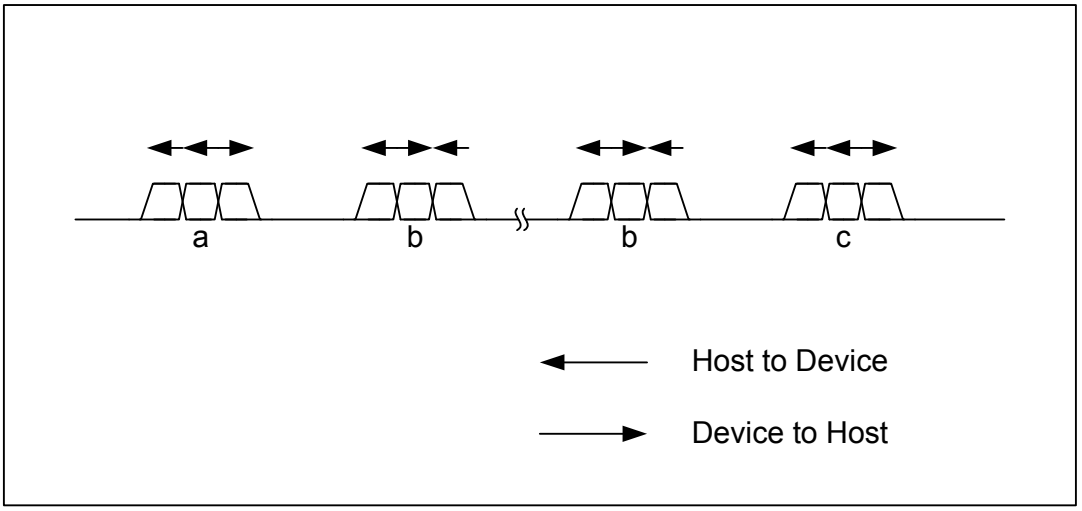


Fig. 1-7 Control transfer when data stage is IN

Since the control transfer data stage and status stage perform normal IN and OUT transactions, flow control is enabled using NAK. The device is permitted to prepare to respond within the specified timeframe.

1.3.3.1 Setup Stage

A setup transaction is automatically executed on receipt of a SETUP token addressed to the same node.

The firmware should monitor the RcvEP0SETUP status and analyze the request from registers D_EP0SETUP_0 to D_EP0SETUP_7 to control transfers.

If the request received includes a data stage in the OUT direction, the D_EP0Control register INxOUT bit should be cleared and endpoint EP0 set to the OUT direction to transition to the data stage.

If the request received includes a data stage in the IN direction, the D_EP0Control register INxOUT bit should be set and endpoint EP0 set to the IN direction to transition to the data stage.

If the request received has no data stage, the D_EP0Control register INxOUT bit should be set and endpoint EP0 set to the IN direction to transition to the status stage.

1.3.3.2 Data Stage/Status Stage

Read registers D_EP0SETUP_0 to D_EP0SETUP_7 and move to the next stage, depending on the details of the request analyzed.

If the stage is in the OUT direction, clear the D_EP0Control register INxOUT and set to the OUT direction, then set the D_EP0ControlOUT register appropriately to control the stage. The ForceNAK bit will be set when the SETUP stage ends. The D_SETUP_Control.ProtectEP0 bit is also set.

If the stage is in the IN direction, set the D_EP0Control register INxOUT to the IN direction, then set the D_EP0ControlIN register appropriately to control the stage. The ForceNAK bit will be set when the SETUP stage ends. The D_SETUP_Control.ProtectEP0 bit is also set.

1.3.3.3 Automatic Address Setup Function

This LSI includes a function to automate SetAddress() request processing for endpoint EP0 control transfers.

The LSI hardware checks request specifics in the registers D_EP0SETUP_0 to D_EP0SETUP_7 and proceeds to the request status stage processing without notifying the firmware if it is a valid SetAddress() request. Once the status stage ends, the D_USB_Address register address is set, and a SetAddressCmp status notification (D_SIE_IntStat.SetAddressCmp bit) is issued to the firmware.

The firmware monitors the SetAddressCmp status, enabling the address to be checked using the D_USB_Address register when it has been issued.

1.3.3.4 Descriptor Reply Function

This LSI includes a descriptor reply function, which is useful for multiple requests for identical data, such as GetDescriptor() for endpoint EP0 control transfers.

The firmware can use this function for IN-transfer data stage requests.

Clear the D_EP0ControlIN.ForceNAK bit, set the first address of the data to be returned within the FIFO descriptor area in the D_DescAdrs_H,L registers and the total number of data bytes to be returned to the D_DescSize_H,L registers before starting the response to the data stage, and then set the D_EP0Control.ReplyDescriptor bit.

The descriptor reply function executes the IN transaction by returning data packets in response to the data stage IN transaction until the set quantity of data has been sent. A NAK response is returned if an IN transaction is issued after the set quantity of data has been sent. Provided the data quantity is smaller than the max packet size, the descriptor reply function sets D_EP0ControlIN.EnShortPkt to allow response to IN transactions until all data has been returned.

If an OUT token is received and a transition to the status stage is detected, the D_EP0Control.ReplyDescriptor bit is cleared, and DescriptorCmp status notification (D_EP0IntStat.DescriptorCmp bit) is issued to the firmware. The firmware should perform the status stage if DescriptorCmp status is detected.

For details of the descriptor area, refer to “1.6 FIFO Management.”

1.3.4 Bulk/Interrupt/Isochronous Transfers

Bulk, interrupt, and isochronous transfers for the general endpoints EPa, EPb, EPc, EPd, and EPe can be controlled as data flows (refer to "1.3.5 Data Flow") or continuous individual transactions (refer to "1.3.2 Transactions").

1.3.5 Data Flow

This section describes the general data flow control for OUT and IN transfers.

1.3.5.1 OUT Transfer

Data received in OUT transfers is written to the FIFO linked to each endpoint. Data can be read from the FIFO by CPU interface register reading or CPU interface DMA reading.

To read FIFO data using CPU interface register reading, select a single FIFO area using the AREAx{x=0-5}Join_0.JoinCPU_Rd bit. The FIFO area selected can be read in the sequence received from the FIFO_Rd or FIFO_ByteRd registers. The FIFO data quantity that can be read out can be obtained from the FIFO_RdRemain_H,L registers. Since empty FIFO areas cannot be read, the data quantity must always be confirmed from the FIFO_RdRemain_H,L registers to ensure the reads do not exceed this quantity.

To read FIFO data using CPU interface DMA reading, select a single FIFO area for each DMA channel using the AREAx{x=0-5}Join_0.JoinDMA bit, and set the DMA_Control.Dir bit to "1." The FIFO area selected can be read out in the sequence received by running the DMA sequence in the CPU interface. The quantity of remaining FIFO data can be checked using the DMA_Remain_H,L registers. The CPU interface will automatically pause the DMA for flow control once the FIFO empties.

If there is free space in the FIFO to receive data packets, a response is automatically returned to the OUT transaction, allowing data to be received. The firmware can perform OUT transfers without controlling individual transactions. However, the endpoint D_EPx{x=a-e}Control.ForceNAK bit is set if a short packet (including data length zero packets) is received when the D_EPx{x=a-e}Control.DisAF_NAK_Short bit is cleared (initial value). The D_EPx{x=a-e}Control.ForceNAK bit should be cleared as soon as preparations are complete for the next data transfers.

1.3.5.2 IN Transfer

Data sent in IN transfers should be written to the FIFO linked to each endpoint. Data can be written to the FIFO by CPU interface register writing or CPU interface DMA writing.

To write data to the FIFO using CPU interface register writing, select a single FIFO area using the AREAx{x=0-5}Join_0.JoinCPU_Wr bit. The FIFO area selected can be written to using the FIFO_Wr register, enabling data packets to be sent in the sequence written. The FIFO empty space can be checked using the FIFO_WrRemain_H,L registers. FIFO areas cannot be written to when full. Always confirm the empty space using the FIFO_WrRemain_H,L registers to ensure writes do not exceed the space available.

To write data to the FIFO using CPU interface DMA writes, select a single FIFO area for each DMA channel using the AREAx{x=0-5}Join.JoinDMA bit and set the DMA_Control.Dir bit to "0." The FIFO area selected can be written to by the CPU interface using the DMA sequence, enabling data packets to be sent in the sequence written. The CPU interface will automatically pause the DMA for flow control once the FIFO fills up.

If the data in the FIFO exceeds the max packet size, a response is automatically returned to the IN transaction, allowing data to be sent. For this reason, IN transfers can be performed by the firmware without controlling individual transactions. However, the EnShortPkt bit must be set if a short packet must be sent at the end of the data transfers. This bit is cleared once the IN transaction is completed for the short packet sent. It can be set once data writing to the FIFO is complete. The endpoint EnShortPkt bit is automatically set in the event of fractional data of less than the max packet size in the FIFO when CPU interface DMA writing ends with the DMA_FIFO_Control.AutoEnShort bit set.

1.3.6 Bulk-Only Support

This LSI includes a bulk-only support function, which assists with Command Block Wrapper (CBW) receipt and Command Status Wrapper (CSW) transmission specific to USB Mass Storage Class (BulkOnly Transport Protocol) for endpoint EPa, EPb, EPc, EPd, and EPe bulk transfers.

Setting the D_BulkOnlyConfig.EPx{x=a-e}BulkOnly bit enables the bulk-only support function for the corresponding endpoint.

While the bulk-only support function is enabled and CBW or CSW support is activated, packets are received (CBW) or sent (CSW) using the area assigned as the CBW or CSW area rather than the FIFO area normally assigned to the endpoint.

1.3.6.1 CBW Support

The firmware can use the CBW support for BulkOnly Transport Protocol command transport. Setting the D_BulkOnlyConfig.EPx{x=a-e}BulkOnly bit enables CBW support for the corresponding OUT endpoint. CBW support should be set to enable it for one endpoint only. Setting the D_BulkOnlyControl.GoCBW_Mode bit while CBW support is enabled will activate CBW support and treat data received in the OUT transaction for the corresponding endpoint as CBW.

If the data packet data length is the 31 bytes expected for CBW, the data is saved in the CBW area and a CBW completion status notification (D_BulkIntStat.CBW_Cmp bit) is issued to the firmware. The D_BulkOnlyControl.GoCBW_Mode bit is automatically cleared, and CBW support terminates. If set, the D_BulkOnlyControl.GoCSW_Mode bit is cleared at the same time.

If the data packet data length is shorter or longer than 31 bytes, the data is not received, and CBW data length error status notification (D_BulkIntStat.CBW_LengthErr bit) is issued to the firmware. The D_BulkOnlyControl.GoCBW_Mode bit is automatically cleared, and CBW support terminates. If set, the D_BulkOnlyControl.GoCSW_Mode bit is cleared at the same time. The issue of a CBW_Err status notification indicates a phase mismatch with the BulkOnly Transport Protocol. The transfer should be reset by firmware (e.g., using STALL) for the endpoint.

If D_EPx{x=a-e}Control.ForceSTALL is set for the corresponding endpoint and a STALL response was returned to the OUT transaction, a CBW error status notification (D_BulkIntStat.CBW_Err bit) is issued to the firmware, the D_BulkOnlyControl.GoCBW_Mode bit is cleared, and CBW support terminates. If set, D_BulkOnlyControl.GoCSW_Mode bit is also cleared at the same time.

If a transaction error such as a CRC error occurs in the OUT transaction, the data is not received, and a CBW transaction error status notification (D_BulkIntStat.CBW_TranErr bit) is issued to the firmware. In this case, the D_BulkOnlyControl.GoCBW_Mode bit is not cleared, and CBW support continues. The D_BulkOnlyControl.GoCSW_Mode bit is not cleared here, even if set.

The data received at the CBW area can be read out using the RAM_Rd function.

1.3.6.2 CSW Support

The firmware can use the CSW support for BulkOnly Transport Protocol status transport. Setting the D_BulkOnlyConfig.EPx{x=a-e}BulkOnly bit enables CSW support for the corresponding IN endpoint. CSW support should be controlled to enable it for one endpoint only. Setting the D_BulkOnlyControl.GoCSW_Mode bit while CSW support is enabled will activate CSW support and send 13 bytes of data from the CSW area, treating the IN transaction for the corresponding endpoint as CSW.

If ACK is received from the host and the transaction is completed after 13 bytes of CSW data have been returned to the host for an IN transaction, a CSW completion status notification (D_BulkIntStat.CSW_Cmp bit) is issued to the firmware. The D_BulkOnlyControl.GoCSW_Mode bit is automatically cleared, and CSW support terminates. The D_BulkOnlyControl.GoCBW_Mode bit is set at the same time and starts CBW support.

If ACK cannot be received from the host after 13 bytes of data have been returned to the host for an IN transaction, a CSW error status notification (D_BulkIntStat.CSW_Err bit) is issued to the firmware. At this time, D_BulkOnlyControl.GoCSW_Mode bit is not cleared and CSW support is activated. The hardware automatically sets the D_BulkOnlyControl.GoCBW_Mode bit at the same time and starts CBW support. In other words, if CSW support is activated here, CBW support is activated concurrently. If an error occurs due to the inability of the host to receive CSW, a CSW retry will be made, but a response is possible since CSW support is activated. Similarly, if an error occurs due to failure of the device to receive the ACK, the next CBW will run, but a response is possible since CBW support is activated. CBW support will terminate CSW support.

Data can be written to the CSW area with the RAM_WrDoor function.

1.3.7 Cable Negotiation Function (Auto Negotiator)

The auto negotiator automatically performs cable negotiation while sequentially checking the USB bus status for bus events such as suspend detection, reset detection, HS Detection Handshake, resume detection. What was actually run by the function can be checked via the individual interrupts (DetectRESET, DetectSUSPEND, ChirpCmp, RestoreCmp). The auto negotiator does not support cable connection/disconnection. It is enabled when the cable is connected and disabled when the cable is disconnected.

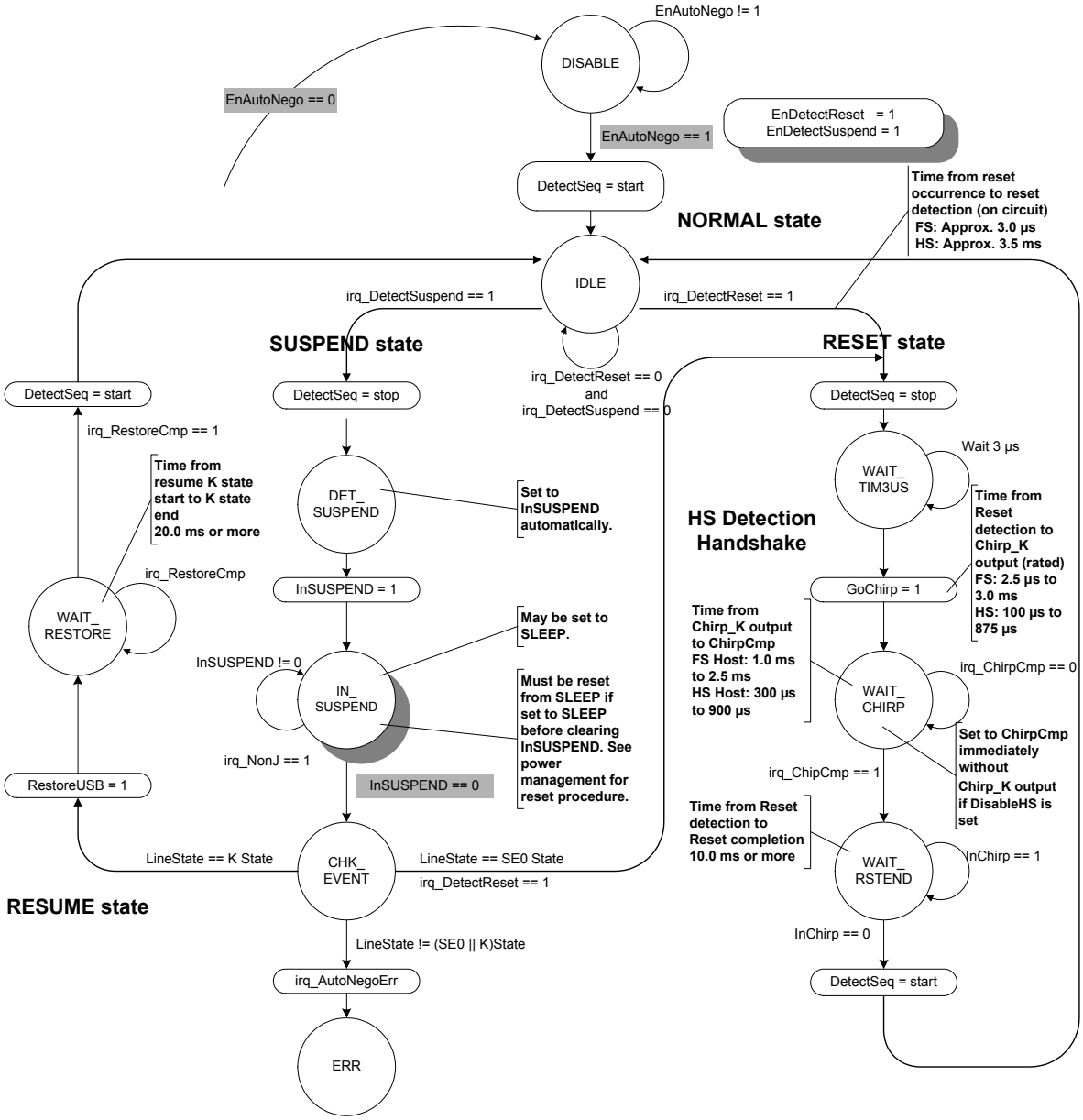


Fig. 1-8 Auto negotiator

1.3.7.1 Auto Negotiator

1.3.7.1.1 DISABLE

This state is assumed when the D_NegoControl.EnAutoNego bit is cleared.

To enable the auto negotiator, set the reset detection interrupt permission bit (D_SIE_IntEnb.EnDetectRESET) and suspend the detection interrupt permission bit (D_SIE_IntEnb.EnDetectSUSPEND). Allow both event detection interrupts before setting the D_NegoControl.EnAutoNego bit.

Enabling the auto negotiator enables the internal event detection function. The D_NegoControl.DisBusDetect bit must never be set when auto negotiator is enabled.

1.3.7.1.2 IDLE

This state awaits reset detection and suspend detection.

If no bus activity is detected on the USB bus for more than 3 ms while the current USB speed is HS, FS termination is enabled. A suspend is determined if FS-J is detected; a reset is determined if SE0 is detected. A reset is determined if SE0 is detected for more than 2.5 µs while the current speed is FS; a suspend is determined if no bus activity is detected for a period exceeding 3 ms. Reset detection or suspend detection interrupts occur at the same time as these judgments, and the D_SIE_IntStat.DetectRESET or D_SIE_IntStat.DetectSUSPEND bits are set.

If a suspend is determined, the event detection function is suspended and the DET_SUSPEND state imposed.

If a reset is determined, the event detection function is suspended and the WAIT_TIM3US state imposed.

1.3.7.1.3 WAIT_TIM3US

This adjusts the time taken to run the HS Detection Handshake after reset detection. WAIT_CHIRP status is imposed after a preset time (approx. 3 µs) has elapsed.

1.3.7.1.4 WAIT_CHIRP

This automatically sets the D_NegoControl.GoChirp bit, and runs the HS Detection Handshake. The Chirp completion interrupt status (D_SIE_IntStat.ChirpCmp) is set once the HS Detection Handshake ends, and WAIT_RSTEND state is imposed. For details of HS Detection Handshake, refer to "1.3.7.2.5 HS Detection Handshake."

If the D_NegoControl.DisableHS bit is set, the Chirp completion interrupt status (D_SIE_IntStat.ChirpCmp) is set without running the HS Detection Handshake, and a WAIT_RSTEND status is imposed.

Operation will be at the transfer speed set in the D_USB_Status.FSxHS bit after this state ends. If it is necessary to detect that the transfer speed has been changed, the D_SIE_IntEnb.EnChirpCmp bit should be set to enable the previously mentioned Chirp completion interrupt.

1.3.7.1.5 WAIT_RSTEND

Waits at this state until the reset period expires. Chirp transmission from the host (receipt by this IC) is used to determine the end of the reset period for HS, and switching from SE0 to J is used for FS.

The event detection function is enabled after the reset period is determined to have expired, and the IDLE state is imposed once again.

1.3.7.1.6 DET_SUSPEND

If a suspend is determined, the D_NegoControl.InSUSPEND bit is automatically set and IN_SUSPEND state imposed. The D_NegoControl.InSUSPEND bit enables the function that detects the bus transition from FS-J, allowing detection of resume or resetting by the host.

Whether the suspend state reduces current consumption depends on the application. This LSI allows lower current consumption during standby using power management. For more information on power management and control methods, refer to "1.5 Power Management Functions."

NonJ interrupt should be permitted to detect the resume (FS-K), which is the suspend end command here.

1.3.7.1.7 IN_SUSPEND

The firmware should clear the D_NegoControl.InSUSPEND bit when the NonJ interrupt status (D_SIE_IntStat.NonJ) is set and a reset from suspend command is determined. Return the state to ACTIVE before clearing InSUSPEND if the state has been switched to SLEEP by power management. The auto negotiator switches to CHK_EVENT state when InSUSPEND is cleared.

To resume from suspend automatically with applications with the remote wakeup function enabled, set the D_NegoControl.SendWakeup bit while in this state and output FS-K for between 1 ms and 15 ms.

1.3.7.1.8 CHK_EVENT

This checks the USB cable and determines a resume if FS-K is detected or a reset if SE0 is detected. If resume is determined, the D_NegoControl.RestoreUSB bit is set and the transfer speed returned to that before the suspend (in accordance with the D_USB_Status.FSxHS value). If a reset is determined, the event detection function is suspended in the same way as for the transition from the IDLE state, and the WAIT_TIM3US state is imposed.

If a state other than FS-K or SE0 is detected, the auto negotiation error interrupt status (D_SIE_IntStat.AutoNegoErr) is set, and the ERR state imposed.

1.3.7.1.9 WAIT_RESTORE

Setting the D_SIE_IntStat.RestoreCmp bit enables event detection and switches to the IDLE state.

1.3.7.1.10 ERR

The only way to reset from this state is to stop the auto negotiator. This state is not specified in the USB standards.

There is no way to determine whether the USB cable has been disconnected in any state. The auto negotiator must be stopped immediately if the USB cable is disconnected.

1.3.7.2 Operation for Individual Bus Events

This section describes methods used to process different bus events. The auto negotiator exerts control via the LSI register interface. This means processing performed automatically by the auto negotiator can also be performed using the firmware.

1.3.7.2.1 Suspend Detection in HS Mode

If a suspend is detected in HS mode, DetectSUSPEND interrupt status is issued and the auto negotiator switches to IN_SUSPEND state. The XINT signal is asserted if the D_SIE_IntEnb.EnDetectSUSPEND bit or USB_DeviceIntEnb.EnD_SIE_IntStat bit is set here and MainIntEnb.EnUSB_DeviceIntStat is set.

The auto negotiator automatically switches to FS mode if no transmission is sent or received (T1) within 3 ms while operating in HS mode (HS termination is disabled and FS termination (Rpu) is enabled). This operation sets DP to “H,” allowing “J” to be checked via the D_USB_Status.LineState[1:0] bit (note that it will be reset (see later) if “SE0” is detected). If “J” is still detected at the subsequent point T2, the D_SIE_IntStat.DetectSUSPEND bit is set and a USB suspend state is determined.

The figure below illustrates the steps involved in running SLEEP during USB suspend.

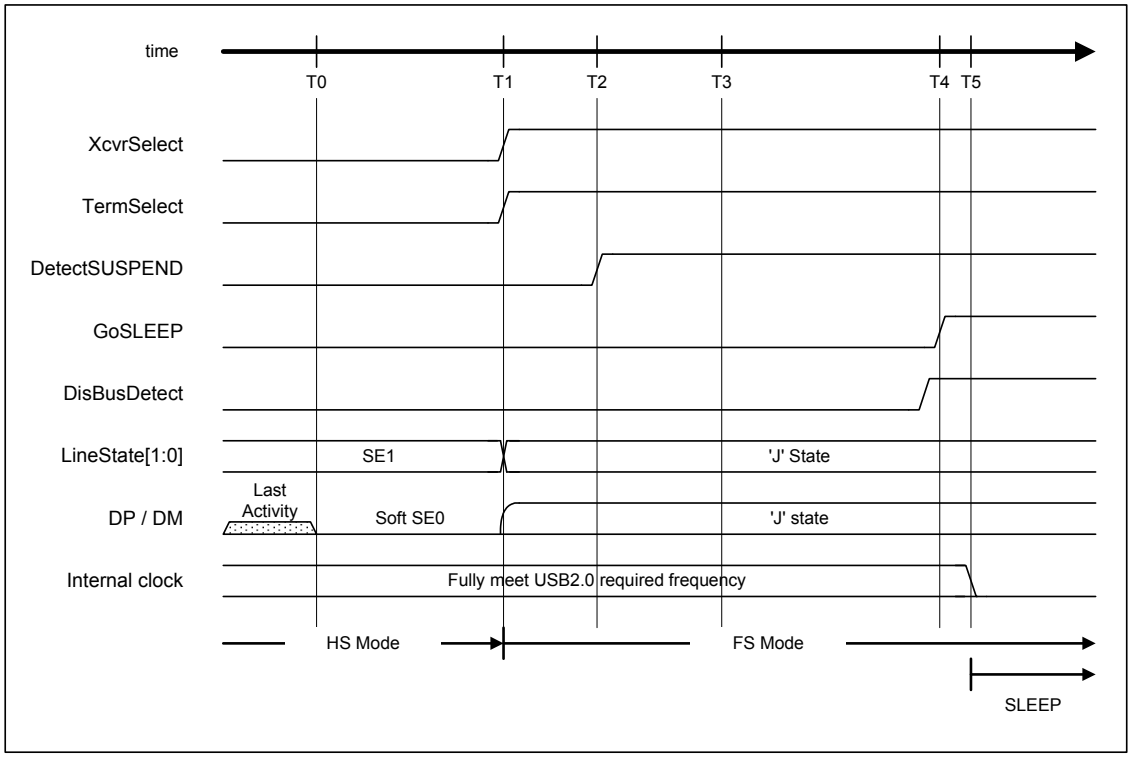


Fig. 1-9 Suspend timing (HS mode)

Table 1-10 Suspend timing values (HS mode)

Timing Parameter	Description	Value
T0	Last bus activity.	0 (reference)
T1	Sets XcvrSelect and TermSelect to "1" and switches from HS mode to FS mode if no activity continues at this point.	HS Reset T0 + 3.0 ms < T1 {T_{WTREV}} < HS Reset T0 + 3.125 ms
T2	Samples LineState[1:0]. If "J" is detected here, DetectSUSPEND is set to "1," and a USB suspend state is determined.	T1 + 100 µs < T2 {$T_{WTWRSTHS}$} < T1 + 875 µs
T3	RESUME must not be issued before this point.	HS Reset T0 + 5 ms {T_{WTRSM}}
T4	Sets GoSLEEP to "1" and moves entirely to SLEEP. Beyond this point, the current drawn from the VBUS cannot exceed the suspend current specified for USB. (Set DisBusDetect to "1" before switching to SLEEP to prevent repeated detection of SUSPEND.)	HS Reset T0 + 10 ms {T_{2SUSP}}
T5	The internal clock stops completely.	T5 < T4 + 10 µs

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.3.7.2.2 Suspend Detection in FS Mode

If a suspend is detected in FS mode, DetectSUSPEND interrupt status is issued and the auto negotiator switches to IN_SUSPEND state. The XINT signal is asserted if the D_SIE_IntEnb.EnDetectSUSPEND bit or USB_DeviceIntEnb.EnD_SIE_IntStat bit is set here and MainIntEnb.EnUSB_DeviceIntStat is set.

The auto negotiator determines USB suspend state and sets the SIE_IntStat.DetectSUSPEND bit if no transmission is sent or received within 3 ms or if "J" is detected continuously in the D_USB_Status.LineState[1:0] bit (T1) and is still detected at point T2.

The figure below illustrates the steps involved in running SLEEP during USB suspend.

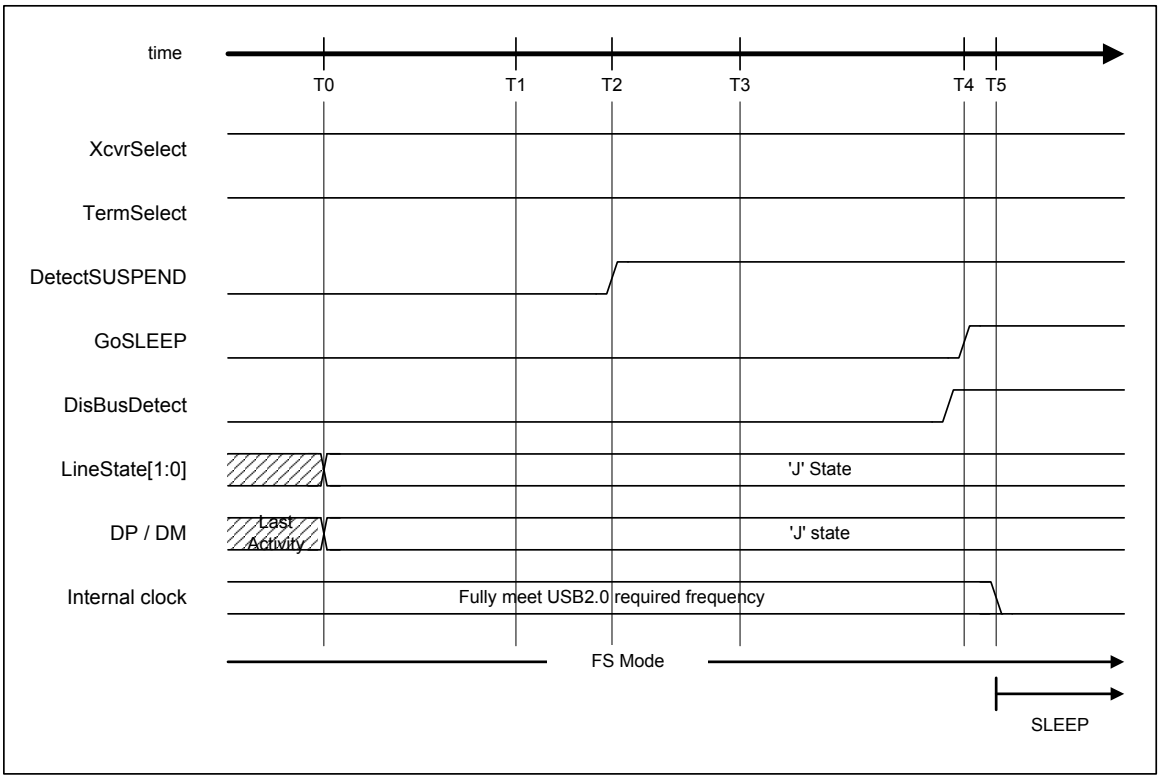


Fig. 1-10 Suspend timing (FS mode)

Table 1-11 Suspend timing values (FS mode)

Timing Parameter	Description	Value
T0	Last bus activity.	0 (reference)
T1	No bus activity at this point.	T0 + 3.0 ms < T1 {T_{WTREV}} < T0 + 3.125 ms
T2	Samples LineState[1:0]. If "J" is detected here, DetectSUSPEND is set to "1," and a USB suspend state is determined.	T1 + 100 µs < T2 {$T_{WTWRSTHS}$} < T1 + 875 µs
T3	RESUME must not be issued before this point.	T0 + 5 ms {T_{WTRSM}}
T4	Sets GoSLEEP to "1" and moves entirely to SLEEP. Beyond this point, the current drawn from the VBUS must not exceed the suspend current specified for USB. (Set DisBusDetect to "1" before switching to SLEEP to prevent repeated detection of SUSPEND.)	T0 + 10 ms {T_{2SUSP}}
T5	The internal clock stops completely.	T5 < T4 + 10 µs

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.3.7.2.3 Reset Detection in HS Mode

If a reset is detected in HS mode, a DetectRESET interrupt status is issued and the auto negotiator performs the reset sequence. The XINT signal is asserted if the D_SIE_IntEnb.EnDetectRESET bit or USB_DeviceIntEnb.EnD_SIE_IntStat bit is set here and MainIntEnb.EnUSB_DeviceIntStat is set.

The auto negotiator switches to FS mode automatically if no transmission is sent or received within 3 ms (HS termination is disabled and FS termination (Rpu) is enabled). If reset, the DP line remains at "L" even when this operation is performed. This allows detection of "SE0" even with the D_USB_Status.LineState[1:0] bit. The D_SIE_IntStat.DetectRESET bit is set if "SE0" is still detected at point T2. The D_NegoControl.DisBusDetect bit is then set before performing the HS Detection Handshake (described later).

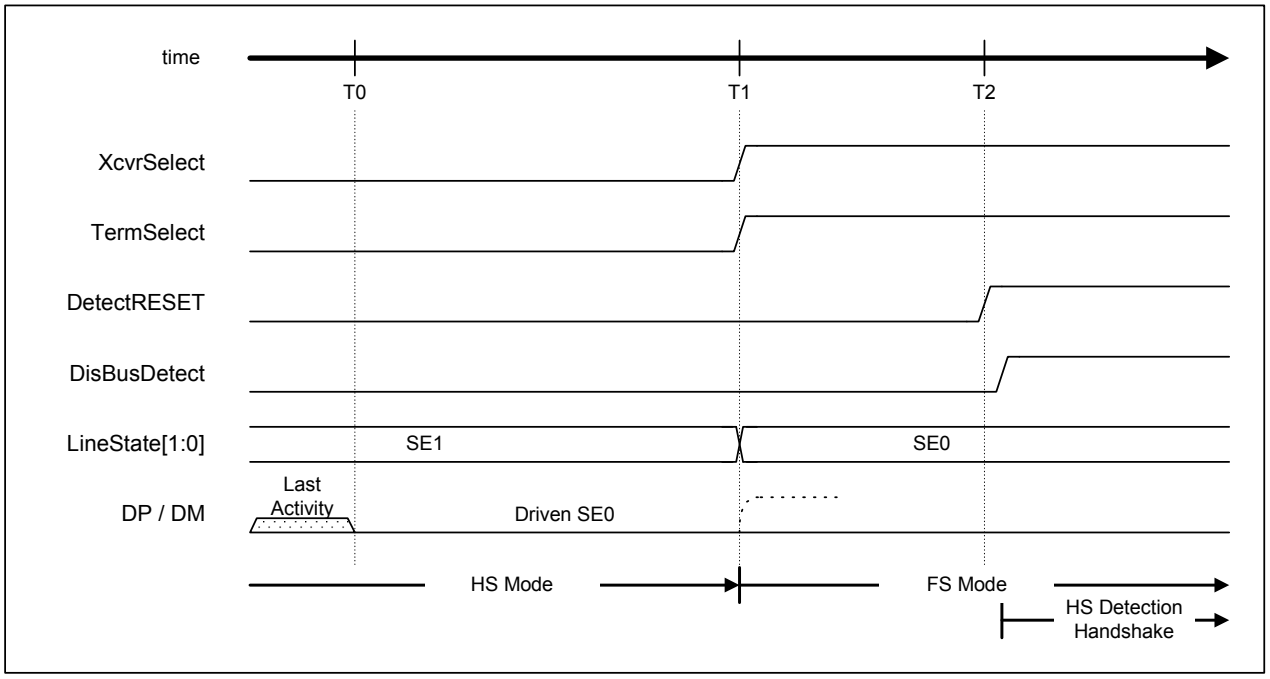


Fig. 1-11 Reset timing (HS mode)

Table 1-12 Reset timing values (HS mode)

Timing Parameter	Description	Value
T0	Last bus activity.	0 (reference)
T1	XcvrSelect and TermSelect are set to "1," and the LSI switches from HS mode to FS mode if there is still no bus activity detected at this point.	HS Reset T0 + 3.0 ms < T1 {T_{WTREV}} < HS Reset T0 + 3.125 ms
T2	Samples LineState[1:0]. If "SE0" is detected here, DetectRESET is set to "1," and the LSI determines a transition to reset. Sets DisBusDetect to "1" after detecting a reset command, then performs the HS Detection Handshake.	T1 + 100 µs < T2 {$T_{WTWRSTHS}$} < T1 + 875 µs

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.3.7.2.4 Reset Detection in FS Mode

If reset is detected in FS mode, a DetectRESET interrupt status is issued and the auto negotiator performs the reset sequence. The XINT signal is asserted if the D_SIE_IntEnb.EnDetectRESET bit or USB_DeviceIntEnb.EnD_SIE_IntStat bit is set here and MainIntEnb.EnUSB_DeviceIntStat is set.

The auto negotiator assumes a reset and sets the D_SIE_IntStat.DetectRESET bit if the D_USB_Status.LineState[1:0] bit is detected as "SE0" continuously for more than 2.5 µs (T1). The D_NegoControl.DisBusDetect bit is then set before performing the HS Detection Handshake (described later).

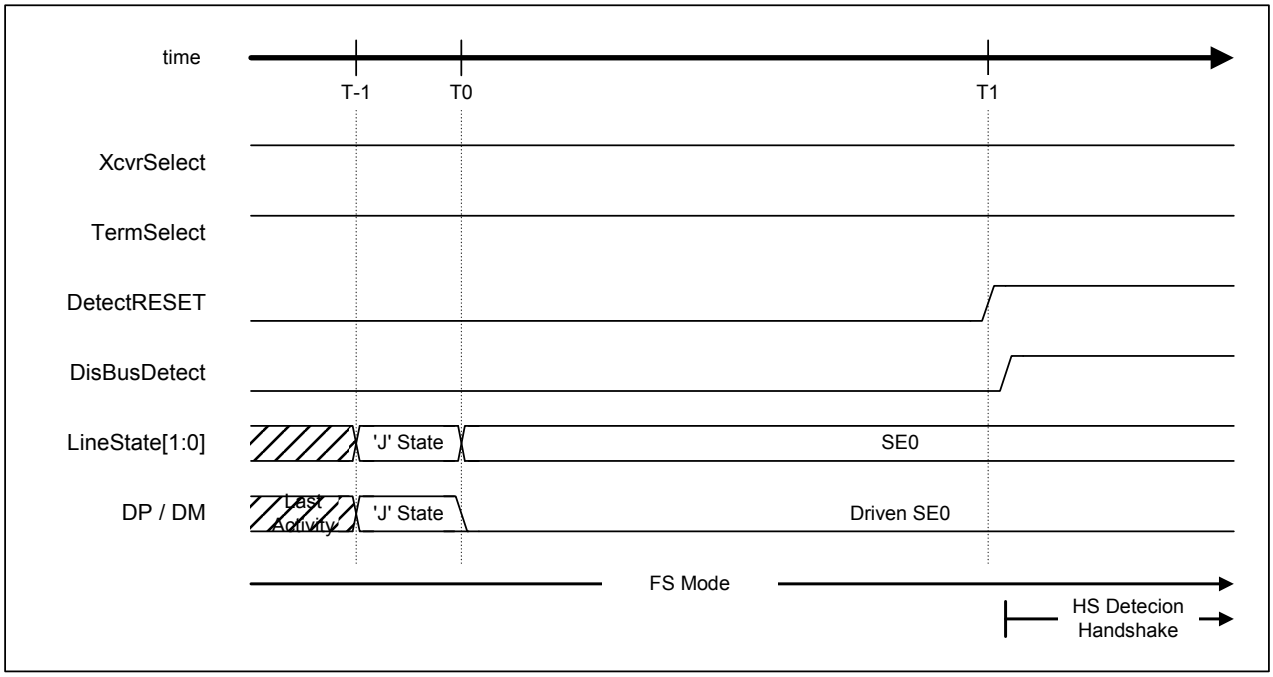


Fig. 1-12 Reset timing (FS mode)

Table 1-13 Reset timing values (FS mode)

Timing Parameter	Description	Value
T-1	Last bus activity.	
T0	Starts the reset command from the host downstream port.	0 (reference)
T1	If "SE0" continues, DetectRESET is set to "1," and the LSI determines a transition to reset. Sets DisBusDetect to "1" after detecting a reset command, then performs an HS Detection Handshake.	HS Reset T0 + 2.5 µs < T1 {T_{WTREV}}

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.3.7.2.5 HS Detection Handshake

An HS Detection Handshake runs when the auto negotiator detects a reset. The D_SIE_IntStat.ChirpCmp interrupt status is set once the HS Detection Handshake ends. The XINT signal is asserted concurrently if the D_SIE_IntEnb.EnChirpCmp bit and USB_DeviceIntEnb.EnD_SIE_IntStat bit are set here and the MainIntEnb.EnUSB_DeviceIntStat is also set.

An HS Detection Handshake is initiated by an "SE0" asserted by the host downstream port during Suspend, FS, or HS states (at the start of resetting from these states). For details, refer to the USB 2.0 standards.

This section describes the procedure for switching to an HS Detection Handshake from the three states above.

In Suspend state, this LSI switches to HS Detection Handshake immediately after detecting "SE0" on the bus.

When operating in FS mode, the LSI switches to HS Detection Handshake after detecting an "SE0" for more than 2.5 μs.

When operating in HS mode, the LSI first switches to FS mode after detecting "SE0" for more than 3.0 ms, since it is necessary to identify the current status (a USB Suspend state or Reset). Both the D_XcvrControl.XcvrSelect and D_XcvrControl.TermSelect bits are switched to FS mode, HS termination is disabled, and FS termination is enabled. The hardware switches from these modes within 3.125 ms, then checks the D_USB_Status.LineState[1:0] bit between 100 μs and 875 μs after switching modes, identifying a USB Suspend state if "J" and a Reset if "SE0." The LSI shifts to an HS Detection Handshake if a Reset is identified here.

A Reset takes at least 10 ms in either case, but the precise duration will vary slightly, depending on the previous state (HS or FS). The operation from "HS Reset T0" is described below, with the Reset start time defined as "HS Reset T0."

The oscillation of the internal clock will be adequately stable during operations, but the internal clock is not output when a Reset is detected if a SLEEP command is issued during Suspend. For this reason, the PM_Control.GoACTIVE bit must always be set to "1" to operate the internal clock for the HS Detection Handshake. For detailed information on this procedure, refer to "1.5 Power Management Functions."

1.3.7.2.5.1 When Connected to the FS Host Downstream Port

The LSI indicates an HS Detection Handshake operation when connected to a host downstream port that does not support HS. Both D_XcvrControl.XcvrSelect and D_XcvrControl.TermSelect bits are in FS mode when the HS Detection Handshake starts (T0) (FS termination, namely DP pull-up resistance (Rpu), is enabled, and HS termination is disabled).

The auto negotiator first sets the D_NegoControl.GoChirp bit. The D_XcvrControl.OpMode[1:0] bit then switches to "Disable Bit Stuffing and NRZI encoding," and data bits composed of "0" values are prepared (T1). This sends an "HS K" (Device Chirp) over the bus. The D_XcvrControl.XcvrSelect bit is set to HS mode concurrently. Setting it to enable transmission allows an "HS K" (Device Chirp) to be sent to the host downstream port. After transmission, the LSI awaits a Host Chirp from the host downstream port (T2). Host downstream ports supporting HS normally send "HS K" and "HS J" continuously after T3 (described later), but the Host Chirp will not be sent even by point T4 if the host downstream port – as in this example – does not support HS. Thus, the D_XcvrControl.XcvrSelect bit automatically switches to FS mode, the D_NegoControl.GoChirp bit is cleared, and the D_USB_Status.FSxHS bit is set. The D_SIE_IntStat.ChirpCmp is also set.

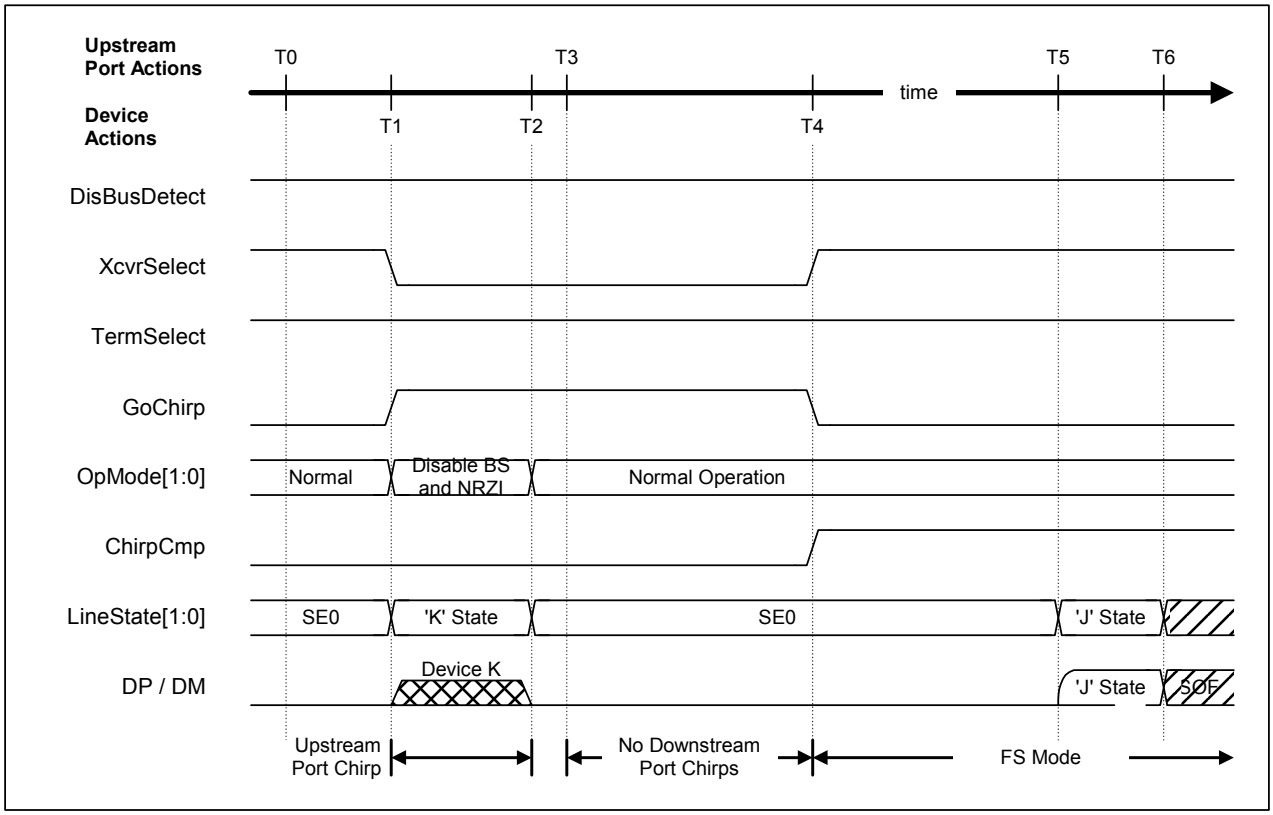


Fig. 1-13 HS detection handshake timing (FS mode)

Table 1-14 HS detection handshake timing values (FS mode)

Timing Parameter	Description	Value
T0	HS detection handshake starts.	0 (reference)
T1	Enables the HS transceiver, sets GoChirp to “1,” and begins sending Chirp K.	T0 < T1 < HS Reset T0 + 6.0 ms
T2	Chirp K transmission ends. Must be sent for a minimum period of 1 ms.	T1 + 1.0 ms {T_{UCH}} < T2 < HS Reset T0 + 7.0 ms {T_{UCHEND}}
T3	Chirp K transmission starts here if the host downstream port supports HS.	T2 < T3 < T2 + 100 µs {T_{WTDCH}}
T4	Returns to FS mode here if no chirp is detected. ChirpCmp is set to “1,” and the LSI awaits the end of the reset sequence.	T2 + 1.0 ms < T4 {T_{WTFS}} < T2 + 2.5 ms
T5	Reset sequence ends.	HS Reset T0 + 10 ms {T_{DRST} (Min)}
T6	Normal operation in FS mode.	T6

Note: Brackets {} indicate names defined in the USB 2.0 standards.

Note: Determine at 66,000 cycles (internal clock: 60 MHz) to generate minimum 1 ms Chirp K.

1.3.7.2.5.2 When Connected to the HS Host Downstream Port

This section discusses the operation of the LSI when connected to a host downstream port that supports HS. Both D_XcvrControl.XcvrSelect and D_XcvrControl.TermSelect bits are in FS mode when the HS Detection Handshake starts (T0) (FS termination, namely DP pull-up resistance (Rpu), is enabled and HS termination disabled).

The auto negotiator first sets the D_NegoControl.GoChirp bit. The D_XcvrControl.OpMode[1:0] bit then switches to "Disable Bit Stuffing and NRZI encoding," and data bits consisting of "0" values are prepared (T1). This sends an "HS K" (Device Chirp) over the bus. The D_XcvrControl.XcvrSelect bit is set to HS mode concurrently; setting it to enable transmission allows an "HS K" (Device Chirp) to be sent to the downstream port. After transmission ends, the LSI awaits a Host Chirp from the downstream port (T2). The downstream port supports HS here, and "HS K" (Chirp K) and "HS J" (Chirp J) are sent continuously in alternation (T3). The XcvrControl.TermSelect bit switches automatically to HS mode (T7) once Chirp K-J-K-J-K-J has been detected at least six times via the USB_Status.LineState[1:0] bit in this state (T6), switching entirely to HS mode. At the same time, the D_NegoControl.GoChirp bit is cleared, the D_NegoStatus.FSxHS bit is cleared, and D_SIE_IntStat.ChirpCmp is also set.

Chirp K and Chirp J from this host downstream port must be recognized as bus activity to prevent the detection of a USB Suspend state. Chirp K and Chirp J are therefore detected sequentially in HS mode and incorporated into the internal Suspend Timer.

The USB_Status.LineState[1:0] bit is used to detect Chirp K-J-K-J-K-J. The D_USB_Status.LineState[1:0] bit can be used since Chirp K and Chirp J are extremely slow compared to normal HS packets. However, loading the bus signal onto the D_USB_Status.LineState[1:0] bit when receiving regular packets is extremely noisy, and so the D_USB_Status.LineState[1:0] bit outputs "J" when bus activity is detected with the D_XcvrControl.TermSelect bit in HS mode, and "SE0" when no bus activity is detected.

In the figure below, the change in the Chirp height from point T6 indicates that device side HS termination is activated by the D_XcvrControl.TermSelect bit. The Chirp is normally approximately 800 mV when the D_XcvrControl.TermSelect bit is in FS mode and approximately 400 mV when the D_XcvrControl.TermSelect bit is in HS mode (same as for normal packets sent and received).

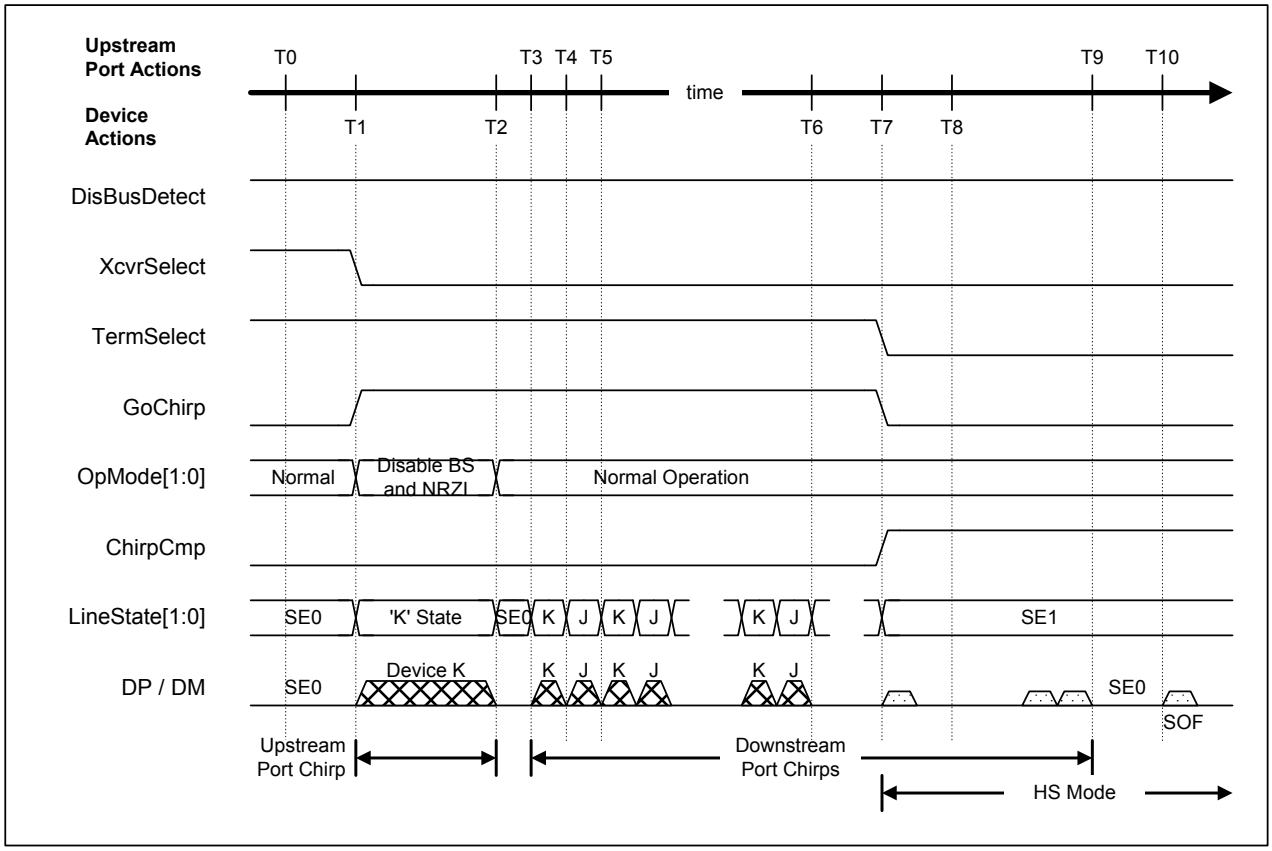


Fig. 1-14 HS detection handshake timing (HS mode)

Table 1-15 HS detection handshake timing values (HS mode)

Timing Parameter	Description	Value
T0	HS detection handshake starts.	0 (reference)
T1	Enables the HS transceiver, sets GoChirp to “1,” and begins sending Chirp K.	T0 < T1 < HS Reset T0 + 6.0 ms
T2	Chirp K transmission ends. Must be sent for a minimum period of 1 ms.	T1 + 1.0 ms {T_{UCH}} < T2 < HS Reset T0 + 7.0 ms {T_{UCHEND}}
T3	Host downstream port sends the first Chirp K to the bus.	T2 < T3 < T2 + 100 µs {T_{WTDCH}}
T4	Host downstream port sends a switch from Chirp K to Chirp J.	T3 + 40 µs {T_{DCHBIT} (Min)} < T4 < T3 + 60 µs {T_{DCHBIT} (Max)}
T5	Host downstream port sends a switch from Chirp J to Chirp K.	T4 + 40 µs {T_{DCHBIT} (Min)} < T5 < T4 + 60 µs {T_{DCHBIT} (Max)}
T6	Detects Chirp K-J-K-J-K-J.	T6
T7	Disables FS termination and enables HS termination on detecting Chirp K-J-K-J-K-J. ChirpCmp is set to “1,” and the LSI awaits the end of the reset sequence.	T6 < T7 < T6 + 500 µs
T8	Recognized as bus activity from Chirp K and Chirp J, but not recognized as packet receipt, since SYNC is not detected.	T8
T9	Chirp K and Chirp J transmission from the host downstream port ends.	T10 - 500 µs {T_{DCHSE0} (Max)} < T9 < T10 - 100 µs {T_{DCHSE0} (Min)}
T10	Reset sequence ends.	HS Reset T0 + 10 ms {T_{DRST} (Min)}

Note: Brackets {} indicate names defined in the USB 2.0 standards.

Note: Determine at 66,000 cycles (internal clock: 60 MHz) to generate minimum 1 ms Chirp K.

1.3.7.2.5.3 If Reset During Sleep

The internal system clock for this LSI does not produce output using PLL in the Sleep state. This section describes resetting from the Sleep state.

The D_SIE_IntStat.NonJ bit is set if a reset is detected in Sleep state (T0). A XINT signal is asserted concurrently if the D_SIE_IntEnb.EnNonJ and USB_DeviceIntEnb.EnD_SIE_IntStat bits are set, and the MainIntEnb.EnUSB_DeviceIntStat bit is set. The PM_Control.GoACTIVE bit should be set to "1" by firmware to enable the LSI to reset immediately from Snooze here and proceed to the reset sequence (T1). After the PLL power-up time has elapsed (T2), PM_Control.PM_State[1:0] switches to "ACTIVE" and internal clock output starts.

The auto negotiator then performs the HS Detection Handshake (described before) after InSUSPEND is cleared.

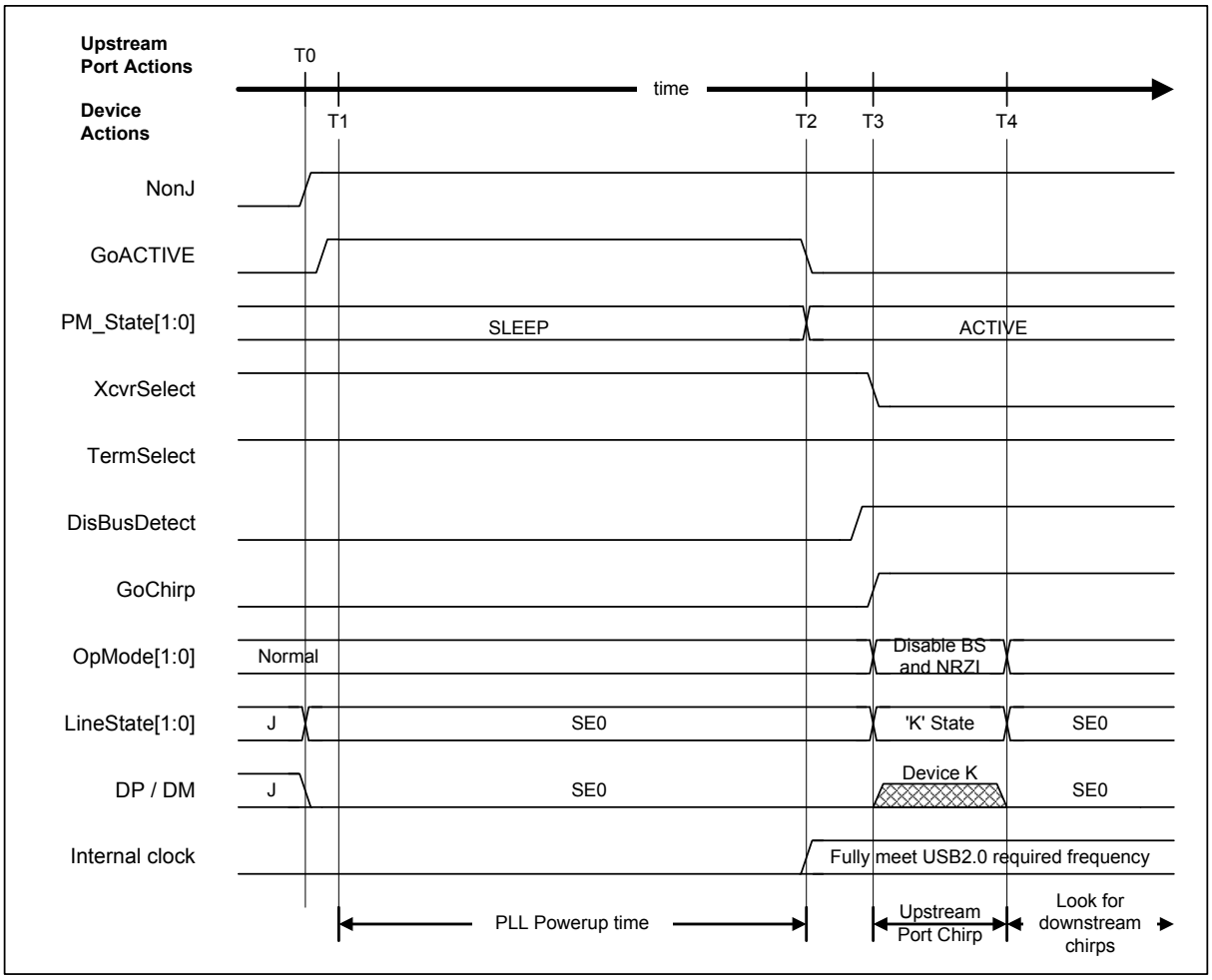


Fig. 1-15 HS detection handshake timing from suspend

Table 1-16 HS detection handshake timing values from suspend

Timing Parameter	Description	Value
T0	Reset during Snooze is detected when NonJ is set to “1” and “LineState[1:0]” is confirmed to be “SE0.”	0 (HS Reset T0)
T1	GoActDevice is set to “1” after a reset is detected.	T1
T2	PM_State switches to “ACT_DEVICE.” The internal clock output stabilizes.	T1 + 250 µs < T2
T3	GoChirp is set to “1,” and Chirp K is sent to the bus. (DisBusDetect is set to “1” before sending Chirp K.)	T2 < T3 < HS Reset T0 + 5.8 ms
T4	Chirp K transmission ends.	T3 + 1.0 ms {T_{UCH}} < T4 < HS Reset T0 + 7.0 ms {T_{UCHEND}}

Note: Brackets {} indicate names defined in the USB 2.0 standards.

Note: Determine at 66,000 cycles (internal clock: 60 MHz) to generate minimum 1 ms Chirp K.

Note: The situation in which the oscillator circuit is also stopped (Sleep state) is described later (PLL and OSC power-up time is required).

1.3.7.2.6 Resume Issue

This section describes methods for resuming automatically when Remote wakeup is supported and Remote wakeup from the host is enabled. Note that Remote wakeup must be permitted only after a minimum of 5 ms has passed since the bus has switched to idle. Also, the current before entering USB Suspend state cannot be drawn from the VBUS until at least 10 ms after the Resume signal has been output.

Remote wakeups must be initiated by firmware, since they are unsupported by the auto negotiator. The sequence will be the same for resuming from normal SUSPEND after the Resume signal is issued. The firmware sets the D_NegoControl.SendWakeup bit and sends the Resume signal to clear InSUSPEND. One ms after transmission starts, the D_NegoControl.SendWakeup bit is cleared, and the Resume signal transmission is stopped. The D_SIE_IntStat.RestoreCmp bit is set once Resume is complete. The XINT signal is asserted concurrently if the D_SIE_IntEnb.EnRestoreCmp or USB_DeviceIntEnb.EnD_SIE_IntStat bits are set and the MainIntEnb.EnUSB_DeviceIntStat bit is set. A Remote wakeup first requires a Reset from the Sleep state. The D_SIE_IntEnb.EnNonJ bit is cleared and the PM_Control.GoACTIVE bit is set (T0). Internal clock output starts once the PM_Control.PM_State[1:0] bit switches to “ACTIVE” after the PLL power-up time elapses (T1).

The D_NegoControl.SendWakeup bit is then set and the Resume signal is sent (T2). D_XcvrControl.OpMode[1:0] is set to “Disable Bit Stuffing and NRZI encoding” by the hardware, “0” is prepared as the data to be sent, the LSI switches to the packet transmission state, and a “K” (Resume signal) is sent. The host downstream port returns a “K” (Resume signal) over the bus when it detects this Resume signal (T3). Clearing the D_NegoControl.SendWakeup bit keeps the Resume signal from being sent over the bus (T4), but the host downstream port keeps the same bus at the Resume signal here.

The auto negotiator detects the Resume signal and sets the D_NegoControl.RestoreUSB bit. The host downstream port stops the Resume signal transmission after a preset period elapses (T5), a 2-bit LS-EOP(2*SE0) is then sent, and the LSI switches to the speed mode prior to the USB Suspend. Both the D_XcvrControl.XcvrSelect and D_XcvrControl.TermSelect bits are switched to the required mode (HS mode here) when this is detected (no longer “K”), and the D_NegoControl.RestoreUSB bit is then cleared.

The speed mode (HS or FS) is stored in the USB_Status.FSxHS bit when USB Suspend starts here, and resetting using Resume returns to the mode indicated by the D_USB_Status.FSxHS bit. The HS Detection Handshake is not performed for each resume here. Note that this explanation addresses only the condition in which the mode before USB Suspend is HS mode. The normal FS mode is used after T5 if the mode before USB Suspend was FS mode, and there are no major sequence differences.

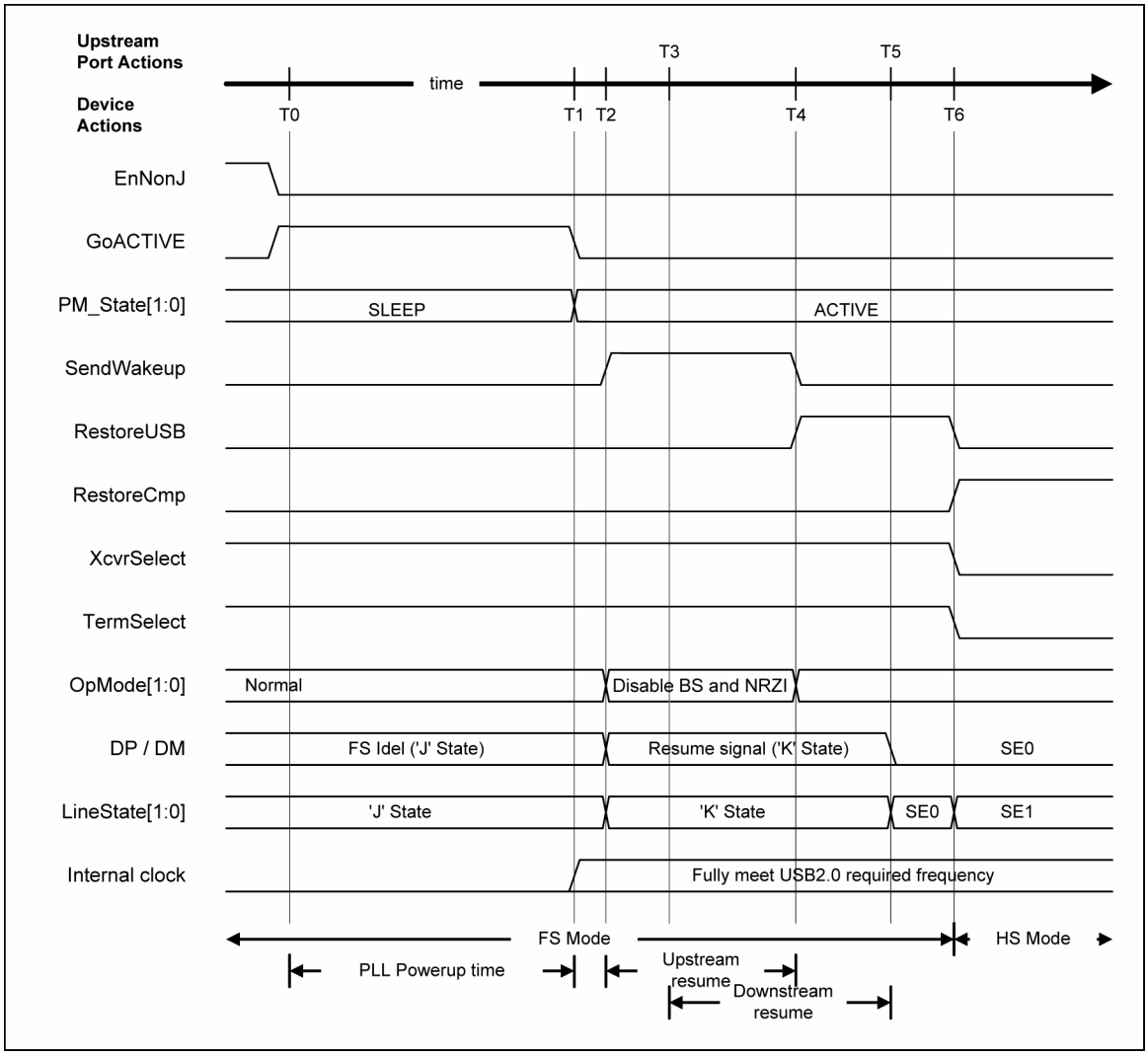


Fig. 1-16 Assert resume timing (HS mode)

Table 1-17 Assert resume timing values (HS mode)

Timing Parameter	Description	Value
T0	Resume starts. GoACTIVE is set to “1.” (EnNonJ must be cleared to “0” before Resume starts.)	0 (reference)
T1	PM_State switches to “ACTIVE.” The internal clock output stabilizes.	T0 + 250 µs < T1
T2	SendWakeup is set to “1,” and FS “K” transmission starts. The current prior to USB Suspend must not be drawn again for at least 10 ms.	T0 < T2 < T0 + 10 ms
T3	The host downstream port returns an FS “K.”	T2 < T3 < T2 + 1.0 ms
T4	SendWakeup is cleared to “0,” and the FS “K” transmission ends. RestoreUSB is set to “1” after confirming “K” using LineState[1:0].	T2 + 1.0 ms {T_{DRSMUP} (Min)} < T4 < T2 + 15 ms {T_{DRSMUP} (Max)}
T5	The host downstream port ends FS “K” transmission.	T2 + 20 ms {T_{DRSMDN}}
T6	RestoreCmp is set to “1.” The LSI automatically switches to HS mode if the mode prior to USB Suspend was HS mode.	T5 + 1.33 µs {2 Low-speed bit times}

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.3.7.2.7 Resume Detection

A NonJ interrupt is issued if Resume is detected during Suspend. The XINT signal is asserted concurrently if the D_SIE_IntEnb.EnNonJ and USB_DeviceIntEnb.EnD_SIE_IntStat bits are set and the MainIntEnb.EnUSB_DeviceIntStat bit is set. If NonJ is detected, the firmware should clear InSUSPEND to reset from the IN_SUSPEND state. The LSI must be returned to ACTIVE before clearing InSUSPEND if set to SLEEP during Suspend. The PM_Control.GoACTIVE bit is set to "1" (T1), and the internal clock output begins when the PM_Control.PM_State[1:0] switches to "ACTIVE" after the PLL power-up time elapses (T2). The D_SIE_IntSTat.RestoreCmp interrupt status is issued once the resume ends. The XINT signal is asserted concurrently if the D_SIE_IntEnb.EnRestoreCmp and USB_DeviceIntEnb.EnD_SIE_IntStat bits are set and the MainIntEnb.EnUSB_DeviceIntStat bit is set.

"J" is observed (D_USB_Status.LineState[1:0] is "J") on the bus during Suspend. Observing "K" on the bus indicates that a wakeup command (Resume command) has been received from the host downstream port (T0). The SIE_IntStat.NonJ bit is set here.

The auto negotiator sets D_NegoControl.RestoreUSB when IN_SUSPEND is reset. The host downstream port stops the Resume signal transmission after a preset period elapses (T3), and the LSI switches to the speed mode prior to the USB Suspend. Both the D_XcvrControl.XcvrSelect and D_XcvrControl.TermSelect bits switch to the required mode (here, HS mode) when this is detected (no longer "K"), and the D_NegoControl.RestoreUSB bit is then cleared and the D_SIE_IntStat.RestoreCmp bit set.
The D_SIE_IntEnb.EnRestoreCmp bit is set here.

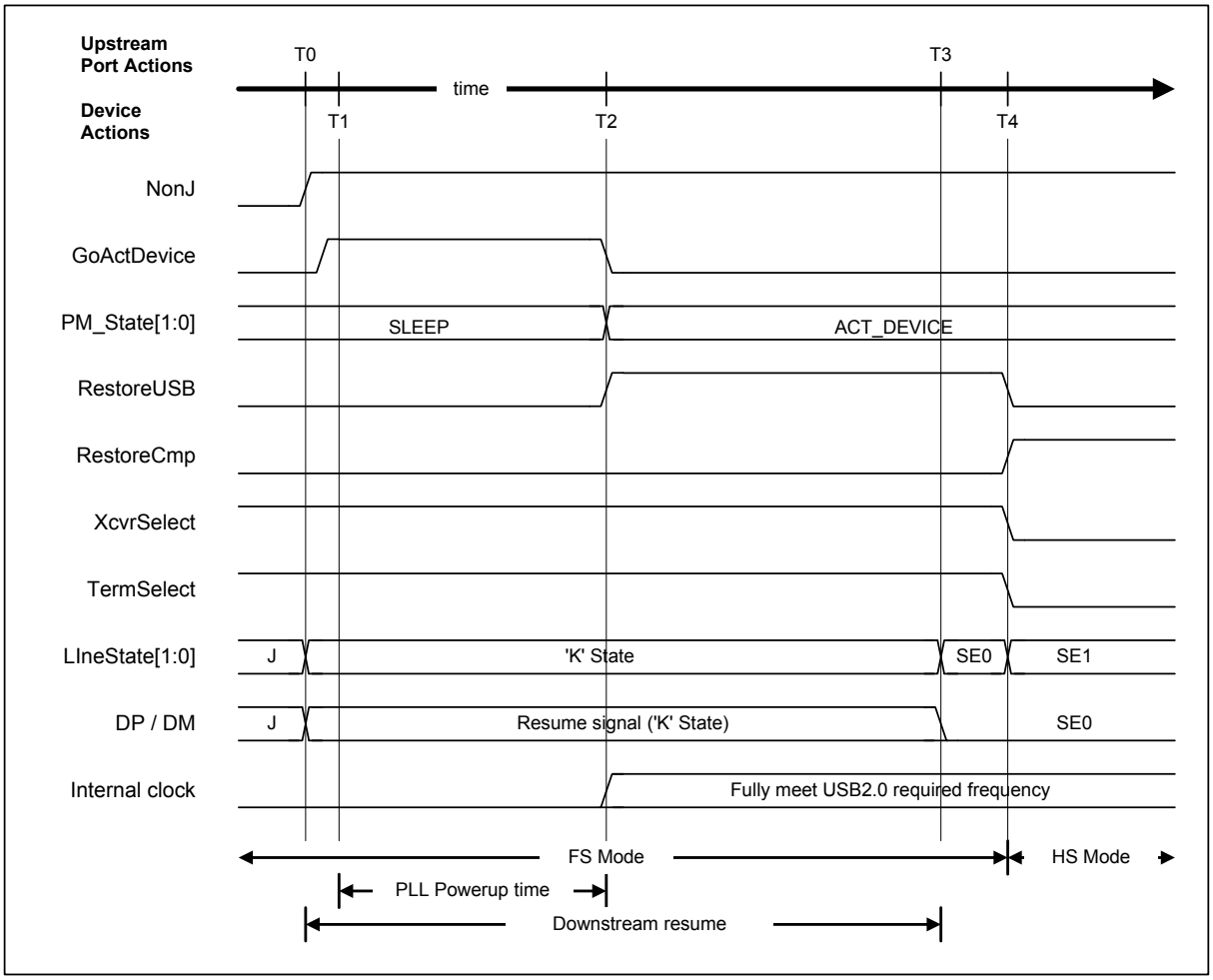


Fig. 1-17 Detect resume timing (HS mode)

Table 1-18 Detect resume timing values (HS mode)

Timing Parameter	Description	Value
T0	The host downstream port outputs FS “K.” NonJ is set to “1.”	0 (reference)
T1	GoACTIVE is set to “1.”	T1
T2	PM_State is set to “ACTIVE.” The internal clock output stabilizes. RestoreUSB is set to “1” after “K” is confirmed in LineState[1:0].	T1 + 250 µs < T2
T3	The host downstream port ends the FS “K” output. At the same time, the host downstream port switches to HS mode prior to USB Suspend.	T2 + 20 ms {T_{DRSMDN}}
T4	The LSI switches automatically to HS mode if the mode prior to USB Suspend was HS mode.	T5 + 1.33 µs {2 Low-speed bit times}

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.3.7.2.8 Cable Attachment

This section describes the situation when the LSI is connected to a hub or host – i.e., when a cable is attached. The auto negotiator does not detect or control cable connections or disconnections. The auto negotiator should be enabled after the cable connection sequence has been completed.

The default values for the D_XcvrControl.XcvrSelect and D_XcvrControl.TermSelect bits should be FS mode and HS mode respectively when the cable is disconnected or not intentionally disconnected.

If the cable is connected when not initially connected (T0), VBUS is set to "H" and the D_USB_Status.VBUS bit is set simultaneously (T1). If the LSI is in Sleep state at this point, for example, the PM_Control.GoACTIVE bit is set to "1" (T2), and the internal clock output begins when the PM_Control.PM_State[1:0] is set to "ACTIVE" after the PLL power-up period has elapsed (T3). The D_XcvrControl.TermSelect bit should be set to FS mode (T4) to switch to FS mode temporarily, since it should be assumed first that an FS device was connected.

The host downstream port then sends a Reset (T5), and the HS Detection Handshake starts.

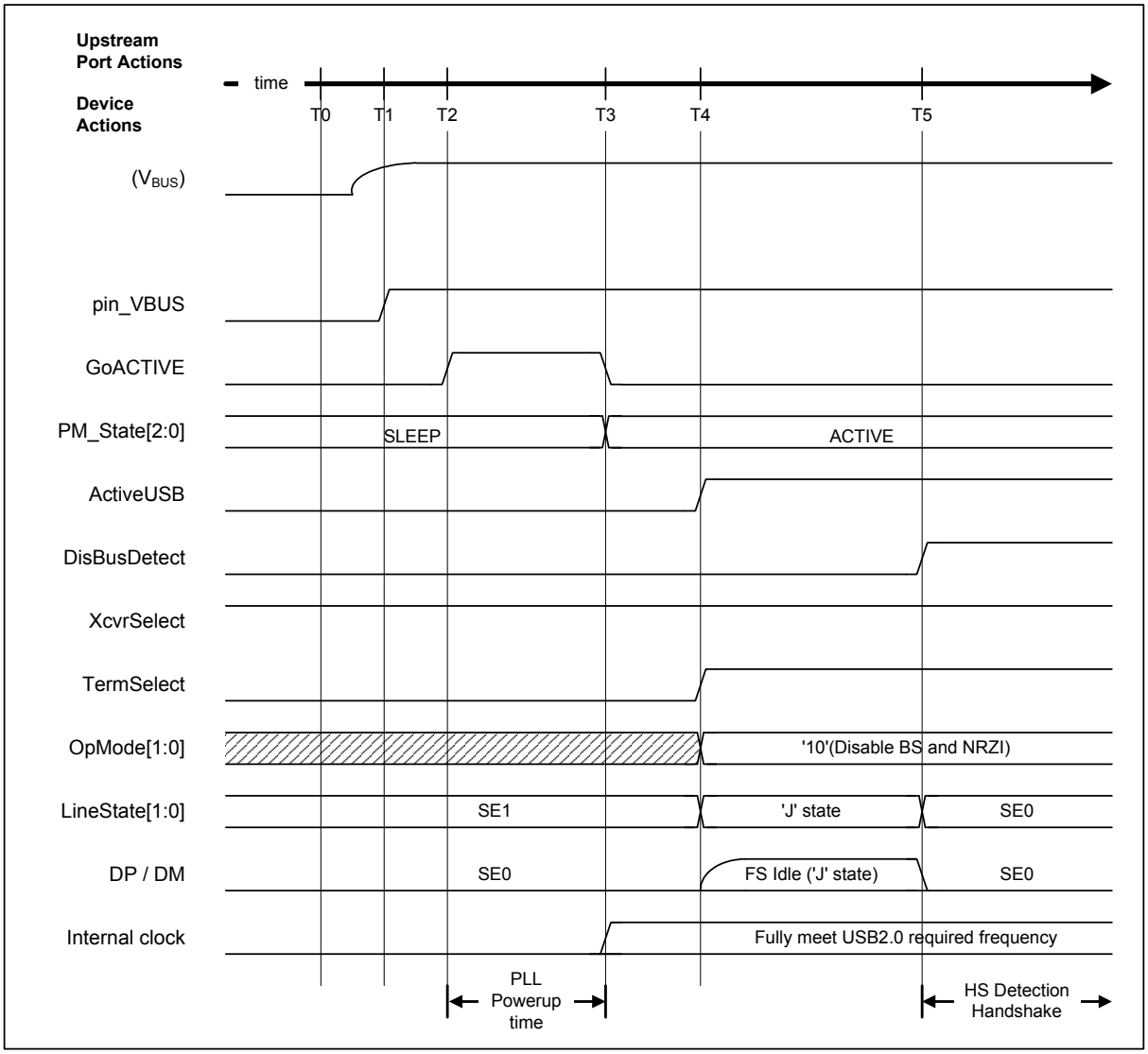


Fig. 1-18 Device attach timing

Table 1-19 Device attach timing values

Timing Parameter	Description	Value
T0	No cable is attached.	0 (reference)
T1	Cable is attached; input pin VBUS_B switches to H.	T1
T2	GoACTIVE is set to “1.”	T2
T3	PM_State is set to “ACTIVE.” The internal clock output stabilizes.	T2 + 250 µs < T3
T4	ActiveUSB is set to “1.” TermSelect is set to “1.” OpMode[1:0] is set to “00.” Switches to FS mode. FS termination is enabled.	T1 + 100 ms {T_{SIGATT}} < T4
T5	A Reset is issued from the host downstream port. DisBusDetect is set to “1.”	T4 + 100 ms {T_{ATTDB}} < T5

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.4 USB Host Control

1.4.1 Channels

1.4.1.1 Channel Outline

This LSI includes channels—host buffers corresponding to individual pipes and the setting registers used for transfers via the buffers.

Transfer information is set to the channels for individual IRPs (I/O Request Packets). FIFO areas are also assigned as buffers for the channels.

The channels automatically divide the IRPs into multiple transactions based on the information set. The channel settings can be switched for individual IRPs, enabling a single channel to support multiple endpoints.

Fig. 1-19 illustrates the channel arrangement.

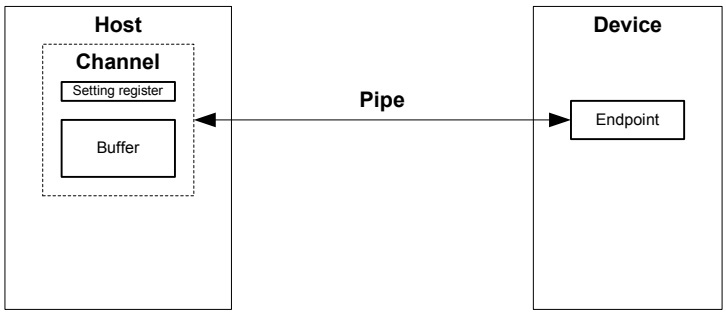


Fig. 1-19 Channel outline

The firmware sets the transfer execution (H_CHx{x=0,a-e}Config_0.TranGo) after setting the buffer and transfer information. After setting the transfer execution, the firmware writes data to the buffer (for OUT transfer) and reads data from the buffer (for IN transfer) until all IRP data has been processed.

Concurrently, the hardware (channel) automatically divides IRPs into multiple transactions. Once transfer is complete, it notifies the firmware using an interrupt.

Fig. 1-20 illustrates the basic procedures involved in transfers.

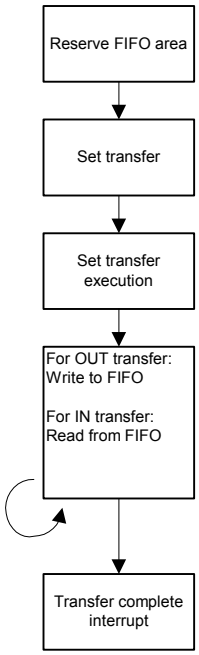


Fig. 1-20 Basic channel transfer procedures

The LSI has a total of six channels, including a channel for control transfer only (CH0), a channel for bulk transfer only (CHa), and four channels for bulk/interrupt/isochronous transfers (CHb, CHc, CHd, CHe). Here, channel CH0 is referred to as the dedicated control channel, while CHa, CHb, CHc, CHd, and CHe are referred to as general channels.

Each channel includes fixed basic settings determined by the USB-defined interface, variable control items for controlling each transfer, and status. The basic settings should be set when initializing the chip or when switching the USB-defined interface.

Up to four interrupt and isochronous transfers can be set at any one time.

Table 1-20 shows the transfer types supported by each channel.

Table 1-20 Supported transfer types

Channel	Supported transfer type	Remarks
CH0	Control transfer	Control transfer support function (1.4.4.3) can be used.
CHa	Bulk transfer	Bulk-only support function (1.4.8) can be used.
CHb, CHc CHd, CHe	Bulk transfer Interrupt transfer Isochronous transfer	Audio class assist function can be used (1.4.9).

1.4.1.2 Dedicated Control Channel

This LSI uses a dedicated control channel (CH0) for control transfers. CH0 is therefore time-multiplexed when used for control transfers to multiple endpoints.

Table 1-21 shows the basic settings for the dedicated control channel (CH0).

Table 1-21 Dedicated control channel basic settings

Item	Register/bit	Description
Transfer speed	H_CH0Config_0.SpeedMode	Sets the transfer speed (HS/FS/LS) for endpoints corresponding to channel CH0.
Toggle sequence bit	H_CH0Config_0.Toggle	Sets the toggle sequence bit initial values when starting a transaction. Also indicates the toggle sequence bit status during or after a transaction.
Transaction type	H_CH0Config_1.TID	Sets the transaction type (SETUP/IN/OUT) issued by channel CH0.
Max packet size	H_CH0MaxPktSize	Sets the maximum packet size to 8 for LS operation and 8, 16, 32, or 64 for FS operation. Sets to 64 for HS operation.
USB address	H_CH0FuncAdrs.FuncAdrs	Sets the USB address forming the destination for the IRP to be executed by channel CH0.
Endpoint number	H_CH0FuncAdrs.EP_Number	Sets the endpoint number forming the destination for the IRP to be executed by channel CH0.
Hub address	H_CH0HubAdrs.HubAdrs	Sets the USB address for the hub used for split transactions.
Port number	H_CH0HubAdrs.Port	Sets the port number for the hub used for split transactions.
IRP data quantity	H_CH0TotalSize_H, H_CH0TotalSize _L	Sets the quantity of data in bytes for the IRP to be executed by channel CH0.
FIFO area	AREAn{n=0-5}StartAdrs_H, AREAn{n=0-5}StartAdrs_L, AREAn{n=0-5}EndAdrs_H, AREAn{n=0-5}EndAdrs_L,	Sets the FIFO address for the area assigned to channel CH0. The FIFO area should be at least as large as the max packet size for channel CH0. The FIFO area size affects data transfer throughput. For detailed information on FIFO area assignments, refer to “1.6 FIFO Management.”
FIFO area join	AREAn{n=0-5}Join_1.JoinEP0CH0	Joins channel CH0 to the FIFO area.
Setup data	H_CH0SETUP_x(x=0-7)	Sets the 8-byte data to be sent by the setup transaction.

1.4.1.3 General Channels

The general channels can be used to set the transaction direction and USB address or endpoint number as required. They can support up to five endpoints simultaneously on a one-to-one basis. Time-multiplexing for each individual IRP in the same way as for the dedicated control channel allows transfers to and from more than five endpoints.

Each channel includes fixed basic settings determined by the USB-defined interface, variable control items for controlling each transfer, and status. The basic settings should be set when initializing the chip or when switching the USB-defined interface.

Table 1-22 shows the basic settings for the general channels. The USB-defined interface should be configured by setting or enabling as appropriate in conjunction with the USB-defined interface definition details.

Table 1-22 Basic settings for general channels

Item	Register/bit	Description
Transfer speed	H_CHx{x=a-e}Config_0.SpeedMode	Sets the transfer speed (HS/FS/LS) for endpoints corresponding to each channel.
Toggle sequence bit	H_CHx{x=a-e}Config_0.Toggle	Sets the toggle sequence bit initial values when starting a transaction. Also indicates the toggle sequence bit status during or after a transaction.
Transaction type	H_CHx{x=a-e}Config_1.TID	Sets the transaction type (IN/OUT) issued by each channel.
Transfer type	H_CHx{x=b-e}Config_1.TranType	Sets the transfer type (bulk/interrupt/isochronous) issued by each channel.
Max packet size	H_CHx{x=a-e}MaxPktSize_H, H_CHx{x=a-e}MaxPktSize_L	Sets the maximum packet size to any value between 1 and 1,024 bytes for transactions executed using each channel.
Hub address	H_CHx{x=a-e}HubAdrs.HubAdrs	Sets the USB address for the hub used for split transactions.
Port number	H_CHx{x=a-e}HubAdrs.Port	Sets the port number for the hub used for split transactions.
USB address	H_CHx{x=a-e}FuncAdrs.FuncAdrs	Sets the USB address forming the destination for the IRP to be executed by each channel.
Endpoint number	H_CHx{x=a-e}FuncAdrs.EP_Number	Sets the endpoint forming the destination for the IRP to be executed by each channel.
IRP data quantity	H_CHx{x=a-e}TotalSize_HH, H_CHx{x=a-e}TotalSize_HL, H_CHx{x=a-e}TotalSize_LH, H_CHx{x=a-e}TotalSize_LL	Sets the quantity of data in bytes for the IRP to be executed by each channel.
Token issue interval	H_CHx{x=b-e}Interval_H, H_CHx{x=b-e}Interval_L	Sets the interval (period) for issuing tokens for interrupt and isochronous transfer.

(continued)

Table 1-22 Basic settings for general channels (continued)

Item	Register/bit	Description
FIFO area	AREAn{n=0-5}StartAdrs_H, AREAn{n=0-5}StartAdrs_L, AREAn{n=0-5}EndAdrs_H, AREAn{n=0-5}EndAdrs_L,	Sets the FIFO address for the area assigned to each channel. The FIFO area should be at least as large as the max packet size for each channel. The FIFO area size affects data transfer throughput. For detailed information on FIFO area assignments, refer to “1.6 FIFO Management.”
FIFO area join	AREAn{n=0-5}Join_1.JoinEPxCHx{x=a-e}	Joins each channel to the FIFO area.

1.4.1.4 Channel Usage Examples

1.4.1.4.1 With One Storage Device Connected

Fig. 1-21 shows a typical channel usage arrangement with a USB Mass Storage Class (BulkOnly Transport Protocol) storage device (e.g., hard disk drive) connected. Devices supporting this class use control, bulk IN, and bulk OUT transfers.

CH0 is used for control transfers.

While general channels are used for bulk IN and bulk OUT transfers, CHa in particular is used when using the bulk-only support function with this LSI.

The bulk-only support function manages USB Mass Storage Class (BulkOnly Transport Protocol) command transport (CBW), data transport, and status transport (CSW) automatically (refer to "1.4.8 Bulk-Only Support Function"), and is included in CHa.

If the bulk-only support function is not used, bulk IN and bulk OUT transfers can be assigned individually to general channels (such as CHb or CHc). In this case, transport must be managed by software.

The hardware executes transactions by scheduling transfers to and from the channels (refer to "1.4.2 Scheduling").

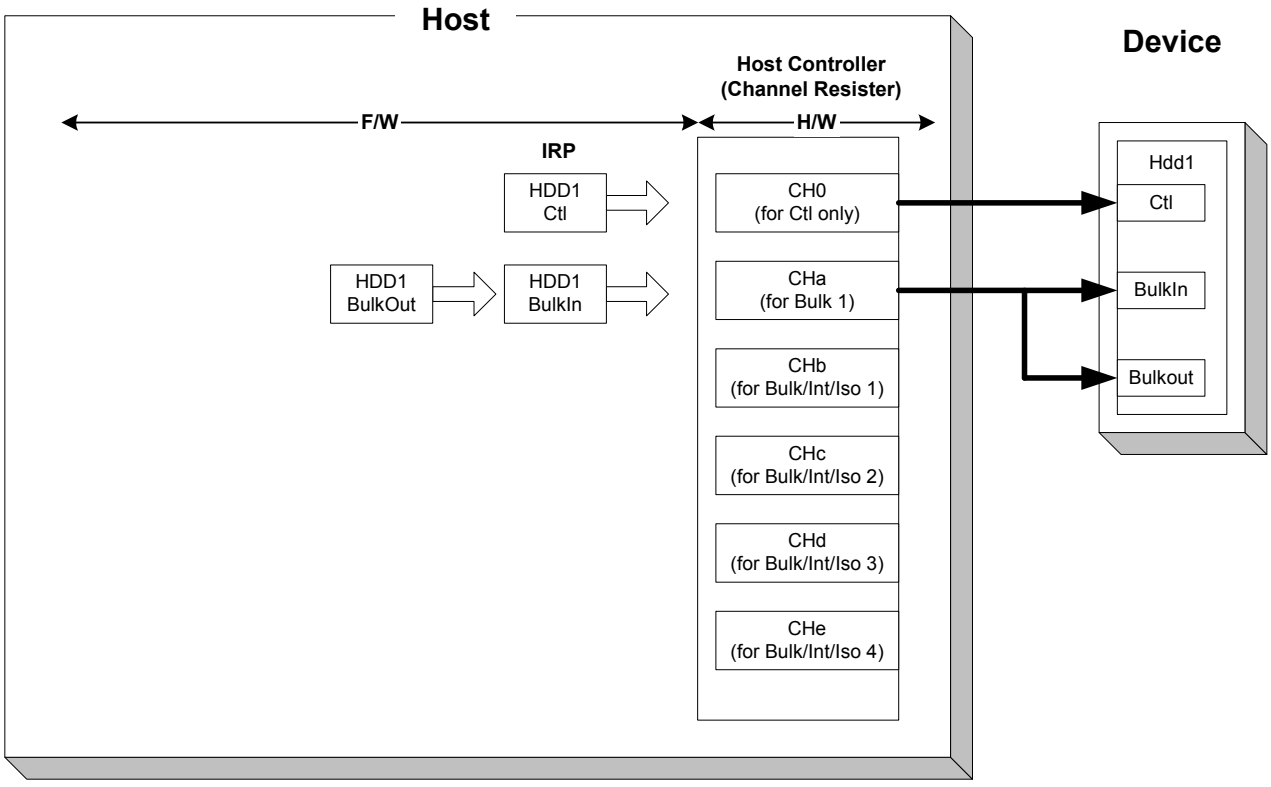


Fig. 1-21 Channel usage example (with one storage device connected)

1.4.1.4.2 With One Communication Device Connected

Fig. 1-22 shows a typical channel usage arrangement with a USB Communication Device Class communication device (e.g., wireless LAN adapter) connected. Devices supporting this class use control, bulk IN, bulk OUT, and interrupt IN transfers.

CH0 is used for control transfers.

General channels (e.g., CHb and CHc) are assigned individually for respective transfers, since bulk IN and bulk OUT transfers are performed in parallel for this class.

A general channel (e.g., CHd) is also assigned individually for interrupt IN transfers in the same way.

The hardware executes transactions by scheduling transfers to and from the channels (refer to "1.4.2 Scheduling").

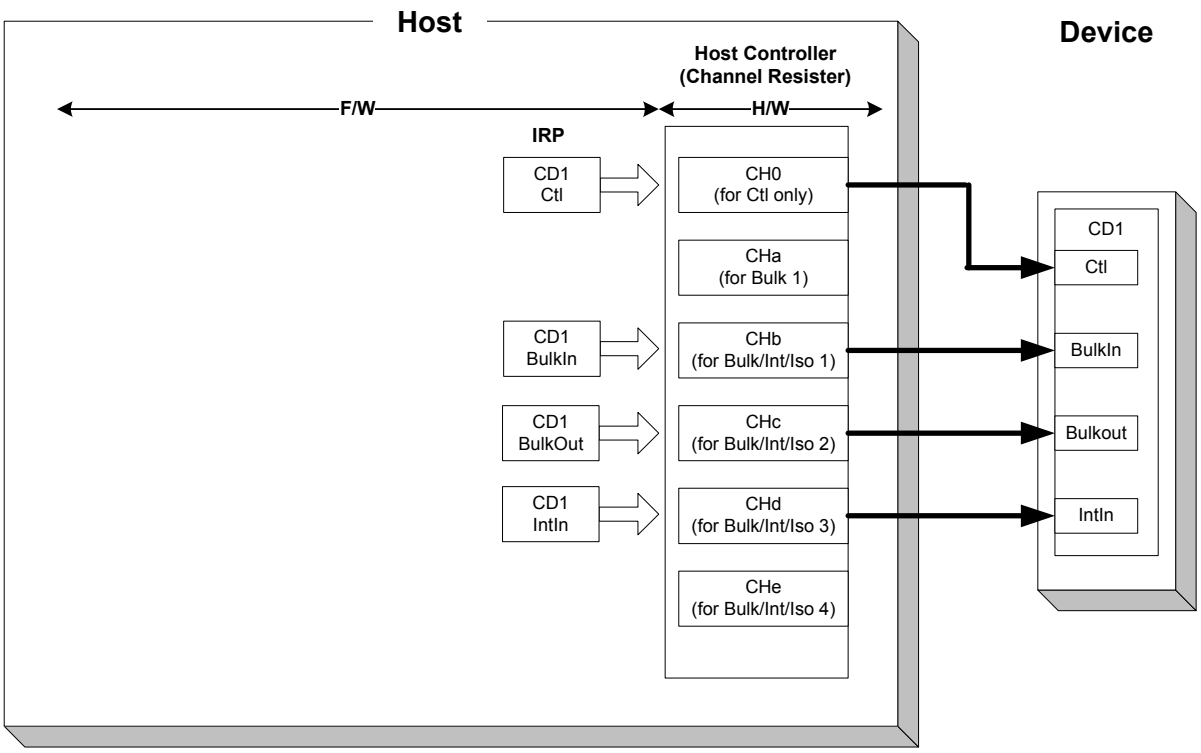


Fig. 1-22 Channel usage example (with one communication device connected)

1.4.1.4.3 With One Human Interface Device Connected

Fig. 1-23 shows a typical channel usage arrangement with a USB Human Interface Device Class device (e.g., mouse) connected. Devices supporting this class use control and interrupt IN transfers.

CH0 is used for control transfers.

A general channel (e.g., CHd) is assigned individually for interrupt IN transfers.

The hardware executes transactions by scheduling transfers to and from the channels (refer to "1.4.2 Scheduling").

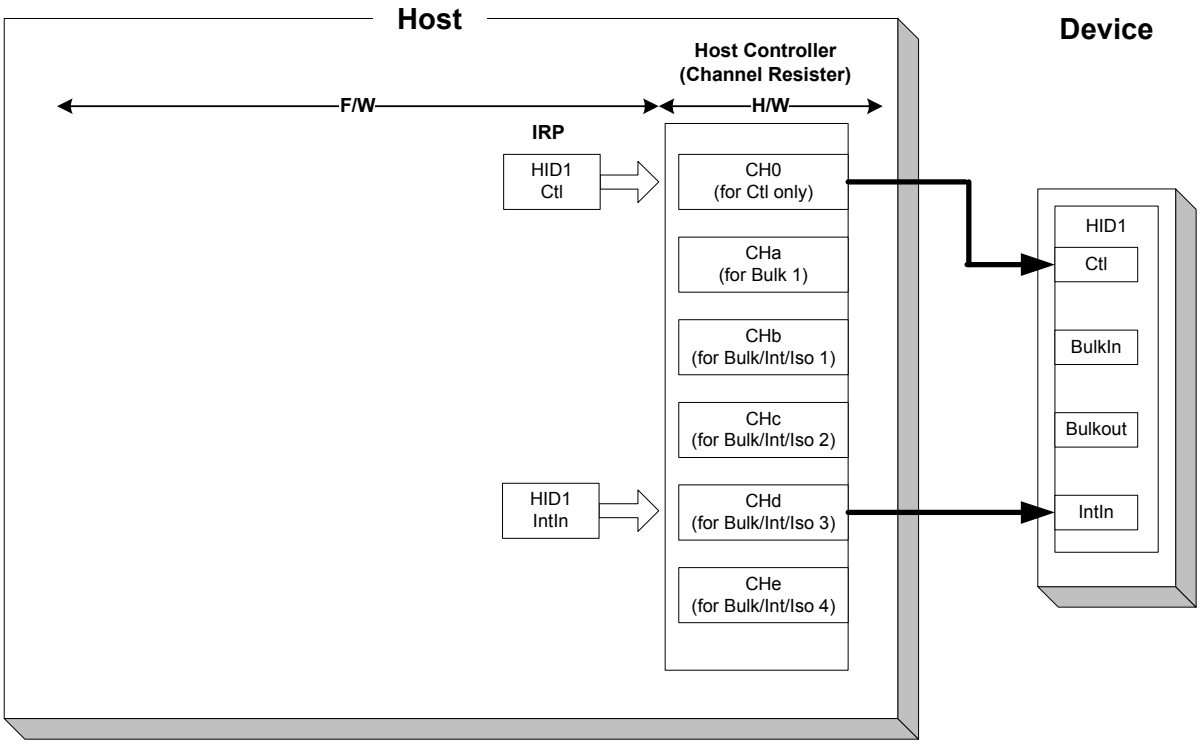


Fig. 1-23 Channel usage example (with one human interface device connected)

1.4.1.4.4 With Two Storage Devices Connected Via a Hub

Fig. 1-24 shows a typical channel usage arrangement with two USB Mass Storage Class (BulkOnly Transport Protocol) storage devices (e.g., hard disk drive or USB memory) connected. USB Mass Storage Class devices use bulk IN and bulk OUT transfers.

Control and interrupt IN transfers are used for the hub.

The USB memory in this example is assumed to have a built-in hub.

CH0 is time-multiplexed for control transfers to and from all devices.

General channels (e.g., CHd and CHe) are individually assigned for hub and USB memory interrupt IN transfers.

CHa is used when using the bulk-only support function with this LSI for bulk IN and bulk OUT transfers to and from a hard disk or USB memory. The bulk-only support function manages USB Mass Storage Class (BulkOnly Transport Protocol) command transport (CBW), data transport, and status transport (CSW) automatically (refer to "1.4.8 Bulk-Only Support Function"), and is included in CHa. If the bulk-only support function is used, the transfer devices (hard disk drive and USB memory in this example) are used by switching individually between command and status transport.

If the bulk-only support function is not used, bulk IN and bulk OUT transfers can be assigned individually to general channels. In this case, transport must be managed by software.

The hardware executes transactions by scheduling transfers to and from the channels (refer to "1.4.2 Scheduling").

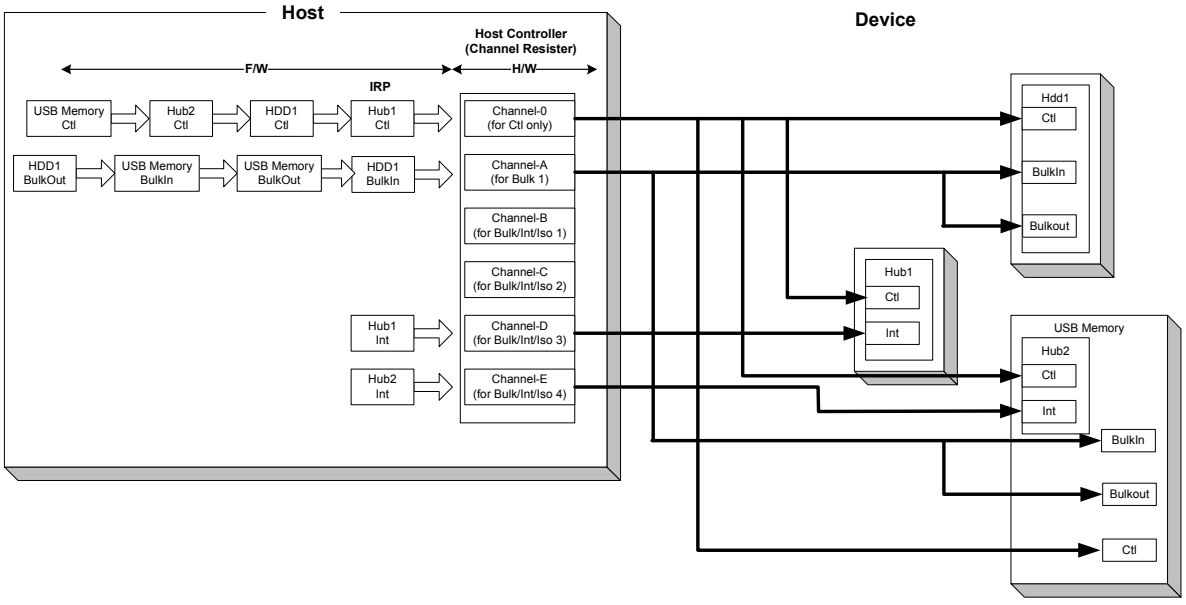


Fig. 1-24 Channel usage example (with two storage devices connected via a hub)

1.4.2 Scheduling

The hardware selects one of the channels for which transfer execution (H_CHx{x=0,a-e}Config_0.TranGo) has been set, then determines whether or not the transfer set for that channel can be executed. If a transfer is determined to be possible, one transaction is executed in accordance with the settings. Once the transaction is completed, the hardware selects another channel, determines whether execution is possible, and executes the transaction in the same way.

The hardware performs transfers with multiple endpoints using multiple channels by repeating this procedure of channel selection, evaluation of execution feasibility, and transaction execution.

Table 1-23 shows the control items for channel CH0 scheduling control.

Table 1-23 Channel CH0 scheduling settings

Item	Register/bit	Description
Execute transfer	H_CH0Config_0.TranGo	Sets transfer execution for channel CH0. Performs the transfer in accordance with channel settings.

Table 1-24 shows the control items for general channel (CHa, CHb, CHc, CHd, CHe) scheduling control.

Table 1-24 General channel scheduling settings

Item	Register/bit	Description
Execute transfer	H_CHx{x=a-e}Config_0.TranGo	Sets transfer execution for each channel. Performs the transfer in accordance with the channel settings.

1.4.3 Transactions

This LSI provides transaction execution functions via the hardware together with interfaces for executing transactions to and from the firmware. The interfaces to firmware consist of control and status registers and interrupt signals asserted by status. For detailed information on settings asserting interrupts by status, refer to "2. Registers."

The hardware selects a channel, determines whether a transfer is possible, then executes the transaction in accordance with the channel settings if a transfer is determined to be possible.

The LSI issues status information to the firmware for each transaction. However, the firmware need not manage each transaction.

For example, the firmware can execute IN transactions automatically and continuously by reading data from the FIFO to create free space in the FIFO using the CPU interface (DMA reading or register reading) if using the IN channel. If using the OUT channel, the firmware can execute OUT transactions automatically and continuously by writing data to the FIFO to create valid data in the FIFO using the CPU interface (DMA writing or register writing).

Table 1-25 shows the control items and status involved in channel CH0 transaction control.

Table 1-25 Channel CH0 control items and status

Item	Register/bit	Description
Transaction status	H_CH0IntStat.TotalSizeCmp, H_CH0IntStat.TranACK, H_CH0IntStat.TranErr, H_CH0IntStat.ChangeCondition	Indicates the transaction results.
Transaction condition code	H_CH0ConditionCode	Indicates transaction result specifics.

Table 1-26 shows the control items and status involved in general channel (CHa, CHb, CHc, CHd, CHe) transaction control.

Table 1-26 General channel control items and status

Item	Register/bit	Description
Transaction status	H_CHx{x=a-e}IntStat.TotalSizeCmp, H_CHx{x=a-e}IntStat.TranACK, H_CHx{x=a-e}IntStat.TranErr, H_CHx{x=a-e}IntStat.ChangeCondition	Indicates the transaction results.
Transaction condition code	H_CHx{x=a-e}ConditionCode	Indicates transaction result specifics.

1.4.3.1 SETUP Transaction

This sets the transaction type (H_CH0Config_1.TID) to "SETUP" for the CH0 basic setting register. The other basic settings are set appropriately, setup data (8 bytes) is written to the H_CH0SETUP_0 to 7 registers, and transfer execution (H_CH0Config_0.TranGo) is set, subjecting channel CH0 to USB transfer scheduling. The remaining frame time is determined when scheduling selects the corresponding channels, and the SETUP transaction is executed.

The data in registers H_CH0SETUP_0 to 7 is used for SETUP transactions. The data packet data length is 8 bytes.

When an ACK is received for the SETUP transaction, ACK status notification (H_CH0IntStat.TranACK bit) is issued to the firmware.

If a normal response is not received for the SETUP transaction, the condition code (H_CH0ConditionCode) is set to "RetryError," and a TranErr status notification (H_CH0IntStat.TranErr bit) is issued to the firmware. Retry processing is performed, but H_CH0Control.TranGo is automatically cleared to end the transfer if three successive errors occur, and then a ChangeCondition status notification (H_CH0IntStat.ChangeCondition bit) is issued.

Fig. 1-25 shows the SETUP transaction arrangement. (a) The LSI issues a SETUP token addressed to endpoint 0 in the destination node. (b) The LSI then sends an 8-byte data packet. (c) The LSI automatically sets the corresponding register on receiving the ACK and issues status information to the firmware.

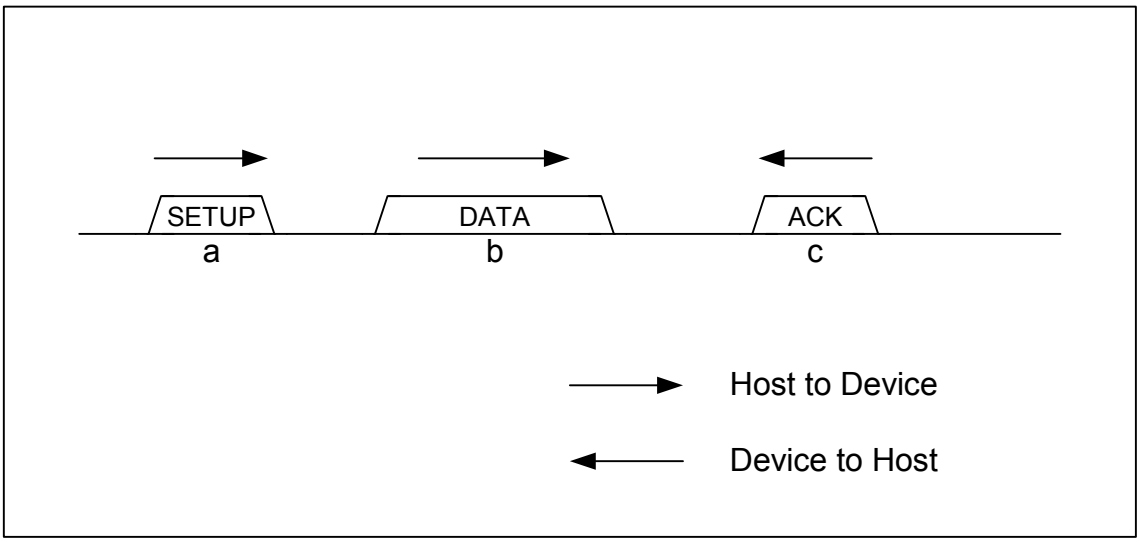


Fig. 1-25 SETUP transaction

1.4.3.2 Bulk OUT Transaction

The transfer type (H_CHx{x=b-e}Config_1.TranType) is set to "Bulk," and the transaction type (H_CHx{x=a-e}Config_1.TID) is set to "OUT" for the CHx{x=a-e} basic setting register. The other basic settings are set appropriately, and transfer execution (H_CHx{x=a-e}Config_0.TranGo) is set, subjecting the channel to hardware-based USB transfer scheduling. The valid FIFO data quantity and remaining frame time are determined when scheduling selects the corresponding channels, and then a bulk OUT transaction is executed. Since Channel CHa is dedicated to bulk transfers, the basic setting registers do not include a transfer type.

The data packet length of the individual data packets is the smaller of H_CHx{x=a-e}MaxPacketSize_H,L and H_CHx{x=a-e}TotalSize_HH,HL,LH,LL.

An ACK status notification (H_CHx{x=a-e}IntStat.TranACK bit) is issued to the firmware on receiving ACK for the bulk OUT transaction. The FIFO is then updated, freeing the area by treating the data sent as already sent.

If a NAK is received for the bulk OUT transaction, the FIFO is not updated, and the area is not freed. Selecting the corresponding channel again executes the same transaction.

If STALL is received for the bulk OUT transaction, H_CHx{x=a-e}Config_0.TranGo is automatically cleared, ending the transfer, and the condition code (H_CHx{x=a-e}ConditionCode) is set to "STALL." A ChangeCondition status notification (H_CHx{x=a-e}IntStat.ChangeCondition bit) is then issued to the firmware. The FIFO is not updated, and the area is not freed.

If a normal response is not received for the bulk OUT transaction, the FIFO is not updated, and the area is not freed. The condition code (H_CHx{x=a-e}ConditionCode) is set to "RetryError," and a TranErr status notification (H_CHx{x=a-e}IntStat.TranErr bit) is issued to the firmware. Retry processing is performed, but H_CHx{x=a-e}Control.TranGo is automatically cleared to end the transfer if three successive errors occur, and a ChangeCondition status notification (H_CHx{x=a-e}IntStat.ChangeCondition bit) is issued to the firmware.

Fig. 1-26 shows the configuration for a bulk OUT transaction when completed. (a) The LSI issues an OUT token addressed to the OUT-direction endpoint at the destination node. (b) The LSI then sends a data packet no larger than the maximum packet size. (c) The LSI automatically sets the corresponding register on receiving ACK and issues status information to the firmware.

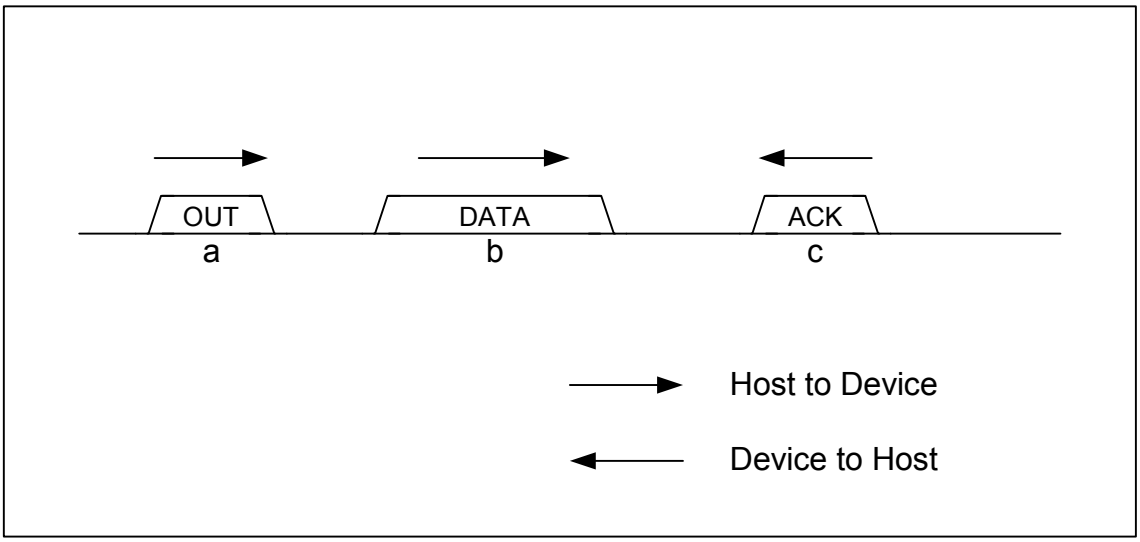


Fig. 1-26 OUT transaction

1.4.3.3 Interrupt OUT Transaction

The transfer type (H_CHx{x=b-e}Config_1.TranType) is set to “Interrupt,” and the transaction type (H_CHx{x=b-e}Config_1.TID) is set to “OUT” for the CHx{x=b-e} basic setting register. The token-issuing interval (H_CHx{x=b-e}Interval_H,L), other basic settings, and then transfer execution (H_CHx{x=b-e}Config_0.TranGo) are set appropriately, subjecting the channel to hardware-based USB transfer scheduling. The token-issuing interval (H_CHx{x=b-e}Interval_H,L), valid FIFO data quantity, and remaining frame time are determined when scheduling selects the corresponding channels, and an interrupt OUT transaction is executed.

The data packet length of the individual data packets is the smaller of H_CHx{x=b-e}MaxPacketSize_H,L and H_CHx{x=b-e}TotalSize_HH,HL,LH,LL.

An ACK status notification (H_CHx{x=b-e}IntStat.TranACK bit) is issued to the firmware on receiving ACK for the interrupt OUT transaction. The FIFO is then updated, freeing the area by treating the data sent as already sent.

If a NAK is received for the interrupt OUT transaction, the FIFO is not updated, and the area is not freed. Selecting the corresponding channel again executes the same transaction.

If STALL is received for the interrupt OUT transaction, H_CHx{x=b-e}Config_0.TranGo is automatically cleared, ending the transfer, and the condition code (H_CHx{x=b-e}ConditionCode) is set to “STALL.” A ChangeCondition status notification (H_CHx{x=b-e}IntStat.ChangeCondition bit) is then issued to the firmware. The FIFO is not updated, and the area is not freed.

If a normal response is not received for the interrupt OUT transaction, the FIFO is not updated, and the area is not freed. The condition code (H_CHx{x=b-e}ConditionCode) is set to “RetryError,” and a TranErr status notification (H_CHx{x=b-e}IntStat.TranErr bit) is issued to the firmware. Although Retry processing is performed, H_CHx{x=b-e}Control.TranGo is automatically cleared to end the transfer if three successive errors occur, and then a ChangeCondition status notification (H_CHx{x=b-e}IntStat.ChangeCondition bit) is issued to the firmware.

1.4.3.4 Isochronous OUT Transaction

The transfer type (H_CHx{x=b-e}Config_1.TranType) is set to “Isochronous,” and the transaction type (H_CHx{x=b-e}Config_1.TID) is set to “OUT” for the CHx{x=b-e} basic setting register. The token-issuing interval (H_CHx{x=b-e}Interval_H,L), other basic settings, and then transfer execution (H_CHx{x=b-e}Config_0.TranGo) are set appropriately, subjecting the channel to hardware-based USB transfer scheduling. The token-issuing interval (H_CHx{x=b-e}Interval_H,L), valid FIFO data quantity, and remaining frame time are determined when scheduling selects the corresponding channels, and an isochronous OUT transaction is executed.

The data packet length is the smaller of H_CHx{x=b-e}MaxPacketSize_H,L and H_CHx{x=b-e}TotalSize_HH,HL,LH,LL.

An ACK status notification (H_CHx{x=b-e}IntStat.TranACK bit) is issued to the firmware once the isochronous OUT transaction ends. The FIFO is then updated, freeing the area by treating the data sent as already sent.

A transaction is not sent if the valid FIFO data quantity is less than the data packet data length. Therefore, the FIFO is not updated, and the area is not freed. TranErr status is issued to the firmware and the condition code (H_CHx{x=b-e}ConditionCode) is set to “BufferUnderrun.”

Fig. 1-27 shows the configuration for an isochronous OUT transaction when completed. (a) The LSI issues an OUT token addressed to the OUT-direction endpoint at the destination node. (b) The LSI then sends a data packet no larger than the maximum packet size. The LSI automatically sets the corresponding register after sending the data packet, and issues status information to the firmware.

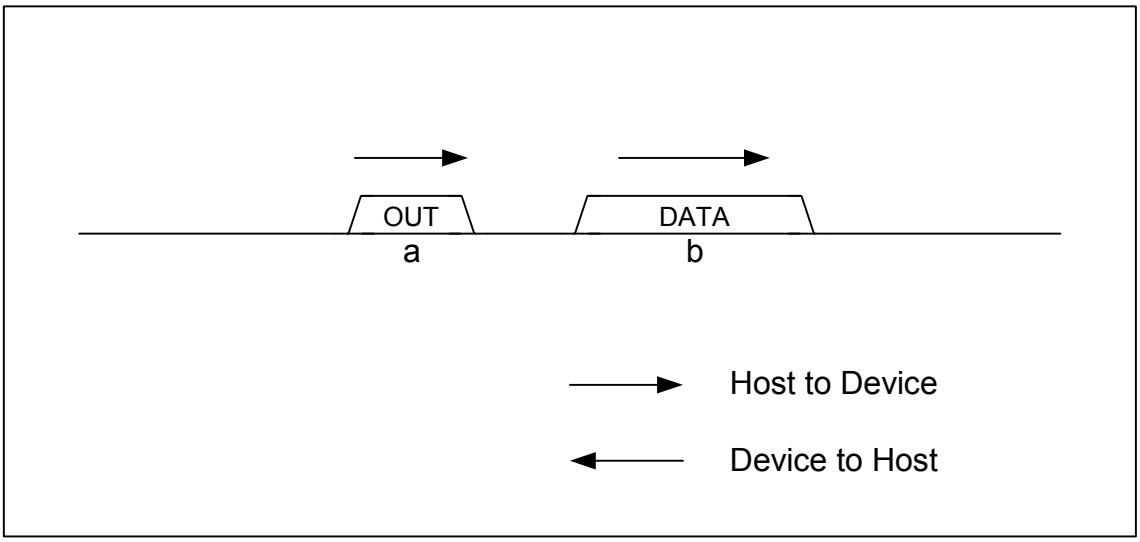


Fig. 1-27 Isochronous OUT transaction

1.4.3.5 Bulk IN Transaction

The transfer type (H_CHx{x=b-e}Config_1.TranType) is set to “Bulk” and the transaction type (H_CHx{x=a-e}Config_1.TID) is set to “IN” for the CHx{x=a-e} basic setting register. The other basic settings are set appropriately, and transfer execution (H_CHx{x=a-e}Config_0.TranGo) is set, subjecting the channel to hardware-based USB transfer scheduling. When the corresponding channel is selected by scheduling, the FIFO free space and remaining frame time is determined, and a bulk IN transaction is executed. Channel CHa is dedicated for bulk transfer, and so the basic setting registers do not include a transfer type.

The data length expected of the data packets to be received is the smaller of H_CHx{x=a-e}MaxPacketSize_H,L and H_CHx{x=a-e}TotalSize_HH,HL,LH,LL.

If all data is received normally in the bulk IN transaction, an ACK response is returned, and the transaction ends. An ACK status notification (H_CHx{x=a-e}IntStat.TranACK bit) is also issued to the firmware. The FIFO is then updated, reserving space by treating the data as already received.

If the data length received is shorter than the data length expected for the bulk IN transaction, H_CHx{x=a-e}Config_0.TranGo is automatically cleared, ending the transfer and returning an ACK response. The condition code (H_CHx{x=a-e}ConditionCode) is set to “DataUnderrun.” A ChangeCondition status notification (H_CHx{x=a-e}IntStat.ChangeCondition bit) is then issued to the firmware. The FIFO is updated and the area is reserved by treating the data as already received.

If a NAK is received for the bulk IN transaction, no status notification is issued, and the FIFO is not updated.

If STALL is received for the bulk IN transaction, H_CHx{x=a-e}Config_0.TranGo is automatically cleared, ending the transfer, and the condition code (H_CHx{x=a-e}ConditionCode) is set to “STALL.” A ChangeCondition status notification (H_CHx{x=a-e}IntStat.ChangeCondition bit) is then issued to the firmware. The FIFO is not updated.

If the data length received is longer than the data length expected for the bulk IN transaction, H_CHx{x=a-e}Config_0.TranGo is automatically cleared, ending the transfer. No response is returned. The condition code (H_CHx{x=a-e}ConditionCode) is set to “DataOverrun.” A ChangeCondition status notification (H_CHx{x=a-e}IntStat.ChangeCondition bit) is then issued to the firmware. The FIFO is not updated.

If a toggle mismatch occurs for the bulk IN transaction, an ACK response is returned. The condition code (H_CHx{x=a-e}ConditionCode) is set to “RetryError,” and a TranErr status notification (H_CHx{x=a-e}IntStat.TranErr bit) is issued to the firmware. The FIFO is not updated.

If a timeout error, CRC error, bit stuffing error, or PID error (including unforeseen PID) occurs for the bulk IN transaction, no response is returned. The condition code (H_CHx{x=a-e}ConditionCode) is set to “RetryError,” and a TranErr status notification (H_CHx{x=a-e}IntStat.TranErr bit) is issued to the firmware. The FIFO is not updated.

If an error occurs for which the condition code (H_CHx{x=a-e}ConditionCode) is set to “RetryError,” retry processing is performed. If three successive errors occur,

H_CHx{x=a-e}Control.TranGo is automatically cleared to end the transfer, and a ChangeCondition status notification (H_CHx{x=a-e}IntStat.ChangeCondition bit) is issued to the firmware.

Fig. 1-28 shows the configuration for a bulk IN transaction when completed. (a) The LSI issues an IN token addressed to the IN-direction endpoint at the destination node. (b) The endpoint sends a data packet not larger than the maximum packet size if a response is possible to the transaction. (c) The LSI returns an ACK response, automatically sets the corresponding register, and issues status information to the firmware.

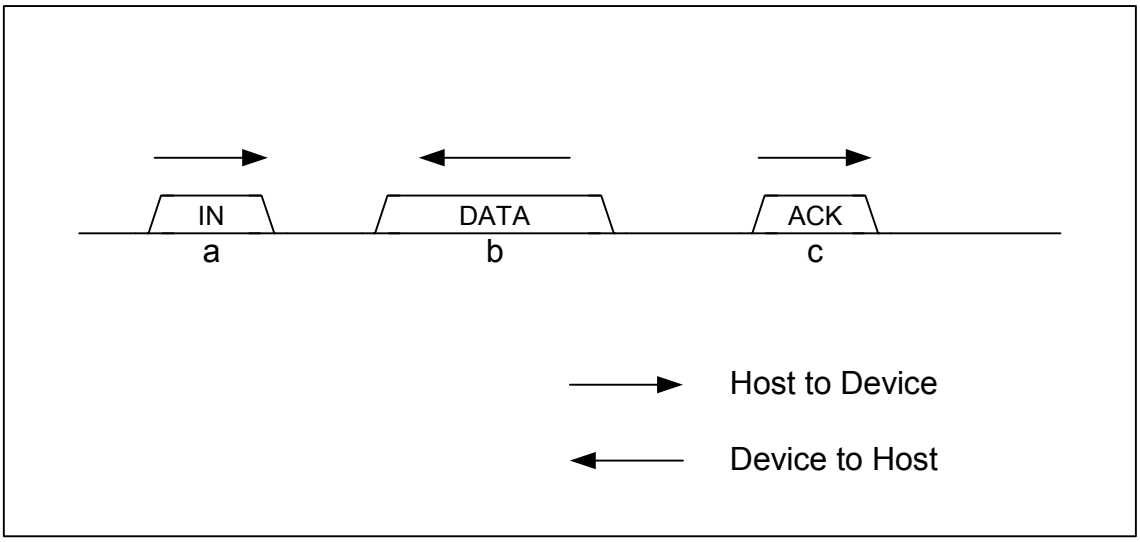


Fig. 1-28 IN transaction

1.4.3.6 Interrupt IN Transaction

The transfer type (H_CHx{x=b-e}Config_1.TranType) is set to “Interrupt” and the transaction type (H_CHx{x=b-e}Config_1.TID) is set to “IN” for the CHx{x=b-e} basic setting register. The token-issuing interval (H_CHx{x=b-e}Interval_H,L), other basic settings, and transfer execution (H_CHx{x=b-e}Config_0.TranGo) are set appropriately, subjecting the channel to hardware-based USB transfer scheduling. The token-issuing interval (H_CHx{x=b-e}Interval_H,L), FIFO free space, and remaining frame time are determined when the corresponding channel is selected by scheduling, and an interrupt IN transaction is executed.

The data length expected of the data packets to be received is the smaller of H_CHx{x=b-e}MaxPacketSize_H,L and H_CHx{x=b-e}TotalSize_HH,HL,LH,LL.

If all data is received normally during the interrupt IN transaction, an ACK response is returned, and the transaction ends. An ACK status notification (H_CHx{x=b-e}IntStat.TranACK bit) is also issued to the firmware. The FIFO is then updated, reserving space by treating the data as already received.

If the data length received is shorter than the data length expected for the interrupt IN transaction, H_CHx{x=b-e}Config_0.TranGo is automatically cleared, ending the transfer and returning an ACK response. The condition code (H_CHx{x=b-e}ConditionCode) is set to “DataUnderrun.” A ChangeCondition status notification (H_CHx{x=b-e}IntStat.ChangeCondition bit) is then issued to the firmware. The FIFO is updated and the area is reserved by treating the data as already received.

If a NAK is received for the interrupt IN transaction, no status notification is issued, and the FIFO is not updated. The next transaction is performed in the next cycle.

If STALL is received for the interrupt IN transaction, H_CHx{x=b-e}Config_0.TranGo is automatically cleared, ending the transfer, and the condition code (H_CHx{x=b-e}ConditionCode) is set to “STALL.” A ChangeCondition status notification (H_CHx{x=b-e}IntStat.ChangeCondition bit) is then issued to the firmware. The FIFO is not updated.

If the data length received is longer than the data length expected for the bulk IN transaction, H_CHx{x=b-e}Config_0.TranGo is automatically cleared, ending the transfer. No response is returned. The condition code (H_CHx{x=b-e}ConditionCode) is set to “DataOverrun.” A ChangeCondition status notification (H_CHx{x=b-e}IntStat.ChangeCondition bit) is then issued to the firmware. The FIFO is not updated.

If a toggle mismatch occurs for the interrupt IN transaction, an ACK response is returned. The condition code (H_CHx{x=b-e}ConditionCode) is set to “RetryError,” and a TranErr status notification (H_CHx{x=b-e}IntStat.TranErr bit) is issued to the firmware. The FIFO is not updated.

If a timeout error, CRC error, bit stuffing error, or PID error (including unforeseen PID) occurs for the interrupt IN transaction, no response is returned. The condition code (H_CHx{x=b-e}ConditionCode) is set to “RetryError,” and a TranErr status notification (H_CHx{x=b-e}IntStat.TranErr bit) is issued to the firmware. The FIFO is not updated.

If an error occurs for which the condition code (H_CHx{x=b-e}ConditionCode) is set to "RetryError," retry processing is performed at the next cycle. If three successive errors occur, H_CHx{x=b-e}Control.TranGo is automatically cleared to end the transfer, and a ChangeCondition status notification (H_CHx{x=b-e}IntStat.ChangeCondition bit) is issued to the firmware.

1.4.3.7 Isochronous IN Transaction

The transfer type (H_CHx{x=b-e}Config_1.TranType) is set to “Isochronous” and the transaction type (H_CHx{x=b-e}Config_1.TID) is set to “IN” for the CHx{x=b-e} basic setting register. The token-issuing interval (H_CHx{x=b-e}Interval_H,L), other basic settings, and transfer execution (H_CHx{x=b-e}Config_0.TranGo) are set appropriately, subjecting the channel to hardware-based USB transfer scheduling. When the corresponding channel is selected by scheduling, the token-issuing interval (H_CHx{x=b-e}Interval_H,L), FIFO free space and remaining frame time are determined, and an isochronous IN transaction is executed.

The data length expected of the data packets to be received is the smaller of H_CHx{x=b-e}MaxPacketSize_H,L and H_CHx{x=b-e}TotalSize_HH,HL,LH,LL.

If all data is received normally in the isochronous IN transaction, the transaction ends. An ACK status notification (H_CHx{x=b-e}IntStat.TranACK bit) is issued to the firmware. The FIFO is then updated, reserving space by treating the data as already received.

If the data length received is shorter than the data length expected for the isochronous IN transaction, an ACK status notification (H_CHx{x=b-e}IntStat.TranACK bit) and a ChangeCondition status notification (H_CHx{x=b-e}IntStat.ChangeCondition bit) are issued to the firmware. The condition code (H_CHx{x=b-e}ConditionCode) is then set to “DataUnderrun.” The FIFO is updated and the area is reserved by treating the data as already received.

If the data length received is longer than the data length expected for the isochronous IN transaction, a ChangeCondition status notification (H_CHx{x=b-e}IntStat.ChangeCondition bit) is then issued to the firmware. The condition code (H_CHx{x=b-e}ConditionCode) is set to “DataOverrun.” The FIFO is not updated.

If a timeout error, CRC error, bit stuffing error, or PID error (including unforeseen PID) occurs for the isochronous IN transaction, a TranErr status notification (H_CHx{x=b-e}IntStat.TranErr bit) is issued to the firmware. The condition code (H_CHx{x=b-e}ConditionCode) is set to “RetryError.” The FIFO is not updated.

Transactions cannot be performed if the FIFO free space is less than the value indicated by H_CHx{x=b-e}MaxPacketSize_H,L. A TranErr status is issued to the firmware, and the condition code (H_CHx{x=b-e}ConditionCode) is set to “BufferOverrun.”

Fig. 1-29 shows the configuration for an isochronous IN transaction when completed. (a) The LSI issues an IN token addressed to the IN-direction endpoint at the destination node. (b) The endpoint sends a data packet not larger than the maximum packet size if a response is possible to this IN transaction. The LSI automatically sets the corresponding register after receiving the data packet, and issues status information to the firmware.

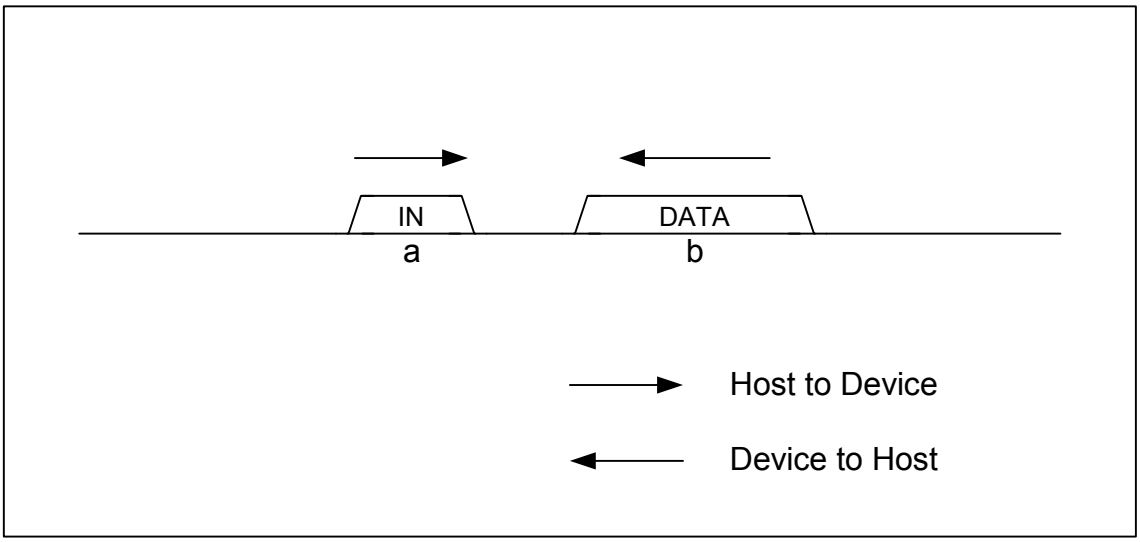


Fig. 1-29 Isochronous IN transaction

1.4.3.8 PING Transaction

The LSI switches to allow execution of PING transactions under the following conditions for HS operations with channels performing bulk OUT or control OUT transactions. When the channel is in PING transaction execution mode, a PING transaction is executed when selected by scheduling.

If an NYET or NAK is received or if no response is received for the OUT transaction, the channel switches to a mode allowing PING transactions.

If a NAK is received for the PING transaction, the channel continues to enable PING transaction execution. No status notification is issued to the firmware.

If an ACK is received for the PING transaction, the channel returns from the mode enabling PING transaction execution to the mode enabling OUT transaction execution. No status notification is issued to the firmware.

If a STALL is received for the PING transaction, H_CHx{x=0,a-e}Config_0.TranGo is automatically cleared to end the transfer, and the condition code (H_CHx{x=0,a-e}ConditionCode) is set to “STALL.” A ChangeCondition status notification (H_CHx{x=0,a-e}IntStat.ChangeCondition bit) is then issued to the firmware.

If a normal response is not received for the PING transaction, the condition code (H_CHx{x=0,a-e}ConditionCode) is set to “RetryError.” A TranErr status notification (H_CHx{x=0,a-e}IntStat.TranErr bit) is issued to the firmware. Retry processing is performed here, but if three successive errors occur, H_CHx{x=0,a-e}Control.TranGo is automatically cleared to end the transfer, and a ChangeCondition status notification (H_CHx{x=0,a-e}IntStat.ChangeCondition bit) is issued to the firmware.

The FIFO is not updated for PING transactions.

Fig. 1-30 shows the configuration for an ACK response to a PING transaction. (a) The LSI issues a PING token addressed to the OUT-direction endpoint existing in the node. (b) The device returns an ACK response to the PING transaction if there is space equivalent to the maximum packet size at the endpoint.

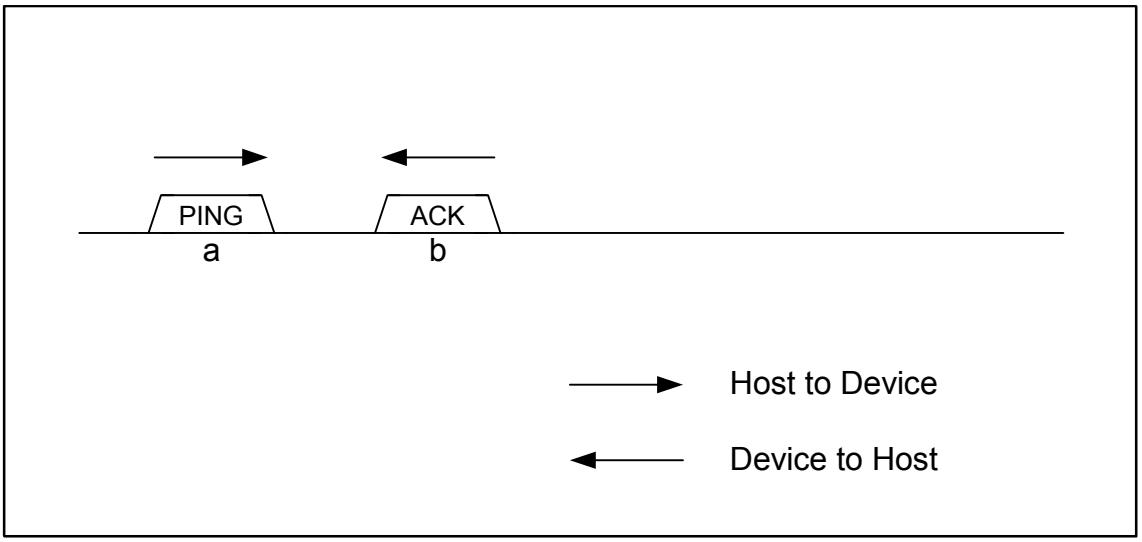


Fig. 1-30 PING transaction

1.4.3.9 Low-speed (LS) Transaction

Transfers to and from LS devices use control and interrupt transfers.

The host operates in LS mode when an LS device is connected to the downstream port. The transfer speed (H_CHx{x=0,a-e}Config_0.SpeedMode) is set to "LS" for the channel used, and the transaction is executed in LS bit time.

If a full-speed (FS) hub is connected to the downstream port and an LS device is connected to the hub's downstream port, the host operates in FS mode. Setting the transfer speed (H_CHx{x=0,a-e}Config_0.SpeedMode) to LS for the channel used sends all downstream packets with preambles attached to the corresponding endpoints. The preamble is sent using FS bit time, and the subsequent downstream packets are sent using LS bit time.

Fig. 1-31 shows the configuration for an interrupt OUT transaction when completed. (a) The LSI issues an OUT token with preamble attached addressed to the OUT-direction endpoint at the destination node. (b) The LSI then sends a data packet not larger than the maximum packet size preceded by a preamble. (c) The LSI automatically sets the corresponding register after receiving the ACK and issues status information to the firmware.

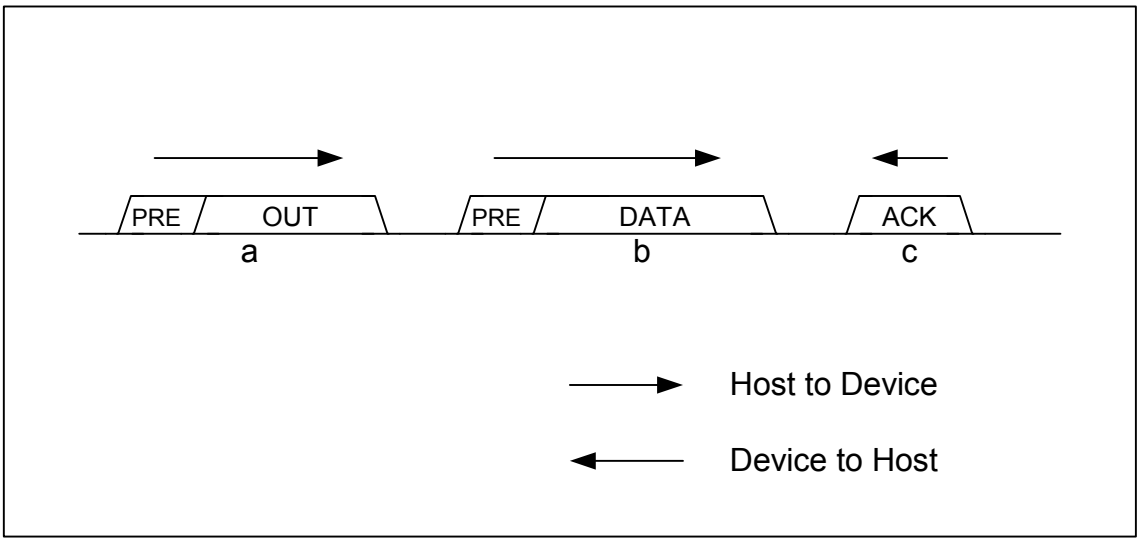


Fig. 1-31 OUT transaction with preamble attached

Fig. 1-32 shows the configuration for an interrupt IN transaction when completed. (a) The LSI issues an IN token preceded by a preamble addressed to the IN-direction endpoint at the destination node. (b) The device sends a data packet not larger than the maximum packet size. The LSI writes this data to the corresponding channel FIFO. (c) The LSI attaches a preamble with an ACK response to the start after receiving the data. It then automatically sets the corresponding register and issues status information to the firmware.

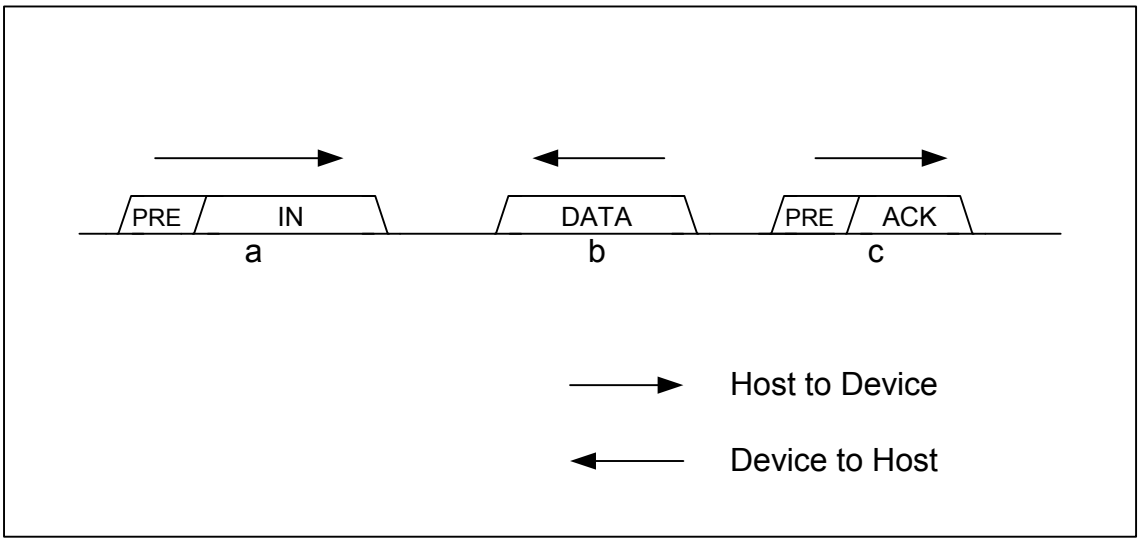


Fig. 1-32 IN transaction with preamble attached

1.4.3.10 Split Transaction

If a high-speed (HS) hub is connected to the downstream port and an FS or LS device is connected to the hub's downstream port, the host operates in HS mode. Setting the transfer speed (H_CHx{x=0,a-e}Config_0.SpeedMode) to FS or LS for the channel used executes the transaction with the hub to the corresponding endpoint using a split transaction.

The hub address (H_CHx{x=0,a-e}HubAdrs.HubAdrs) and port number (H_CHx{x=0,a-e}HubAdrs.Port) are set appropriately for the corresponding channel.

The complete split transaction sequence from the start split transaction is controlled by the hardware for the split transaction. This means the firmware does not need to recognize each transaction of the split transaction.

If the final complete split transaction is completed normally for the complete split transaction sequence from the start split transaction for control, bulk, interrupt, or IN-direction isochronous transfers, an ACK status notification (H_CHx{x=0,a-e}IntStat.TranACK bit) is issued, and the FIFO is updated.

If the final start split transaction is completed normally for OUT-direction isochronous transfer, an ACK status notification (H_CHx{x=b-e}IntStat.TranACK bit) is issued, and the FIFO is updated.

If the start split transaction or complete split transaction is completed normally for transfers other than those described above, no status notification is issued to the firmware.

If an error occurs in individual split transactions for the complete split transaction sequence from the start split transaction, the condition code (H_CHx{x=0,a-e}ConditionCode) is set to "RetryError," and a TranErr status notification (H_CHx{x=0,a-e}IntStat.TranErr bit) is issued to the firmware. The FIFO is not updated. Retry processing is performed. If three successive errors occur for control, bulk, or interrupt transfers, H_CHx{x=0,a-e}Control.TranGo is automatically cleared to end the transfer, and a ChangeCondition status notification (H_CHx{x=0,a-e}IntStat.ChangeCondition bit) is issued to the firmware.

1.4.4 Control Transfers

Control transfers are either controlled as individual transactions for each stage or performed automatically using the control transfer support function described later ("1.4.4.3 Control Transfer Support Function").

Fig. 1-33 shows the configuration for controlling control transfers. The firmware sets the SETUP, DATA, and STATUS stages appropriately, and the hardware executes the control transfer.

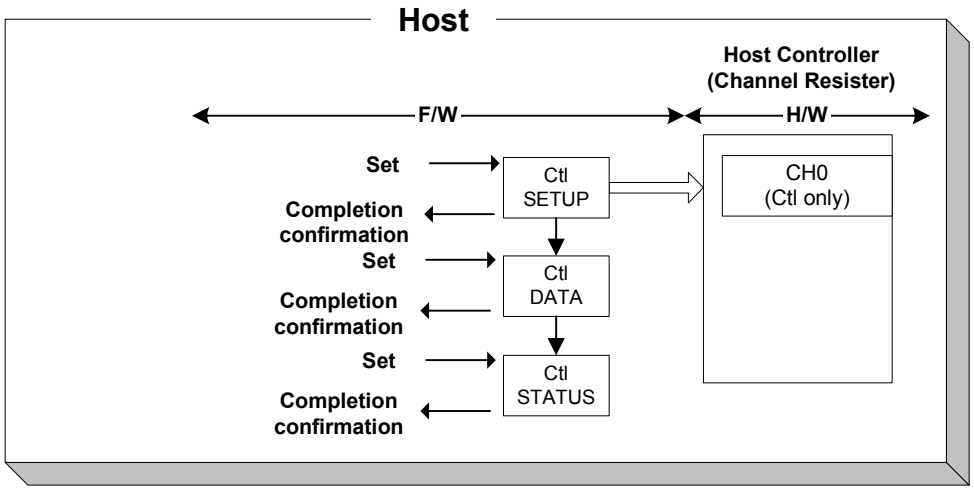


Fig. 1-33 Control transfer control

Fig. 1-34 shows the configuration for control transfers when the data stage is in the OUT direction. (a) The host starts the control transfer using the SETUP transaction. (b) The host issues an OUT transaction and performs the data stage. (c) The host issues an IN transaction and performs the status stage.

Control transfers without a data stage are performed without the data stage shown in this example.

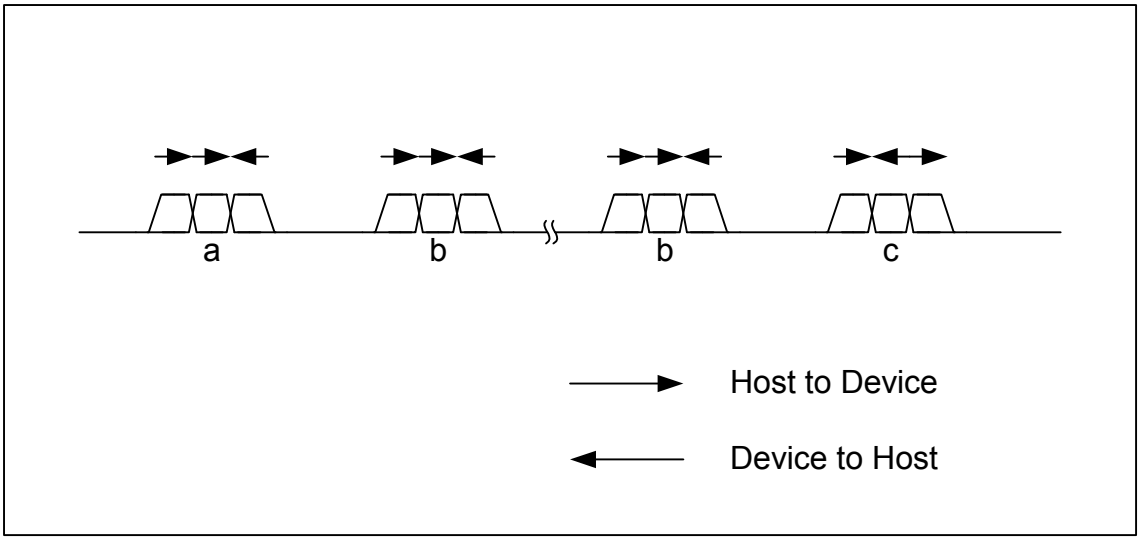


Fig. 1-34 Control transfer with data stage in OUT direction

Fig. 1-35 shows the configuration for control transfers when the data stage is in the IN direction. (a) The host starts the control transfer using the SETUP transaction. (b) The host issues an IN transaction and performs the data stage. (c) The host issues an OUT transaction and performs the status stage.

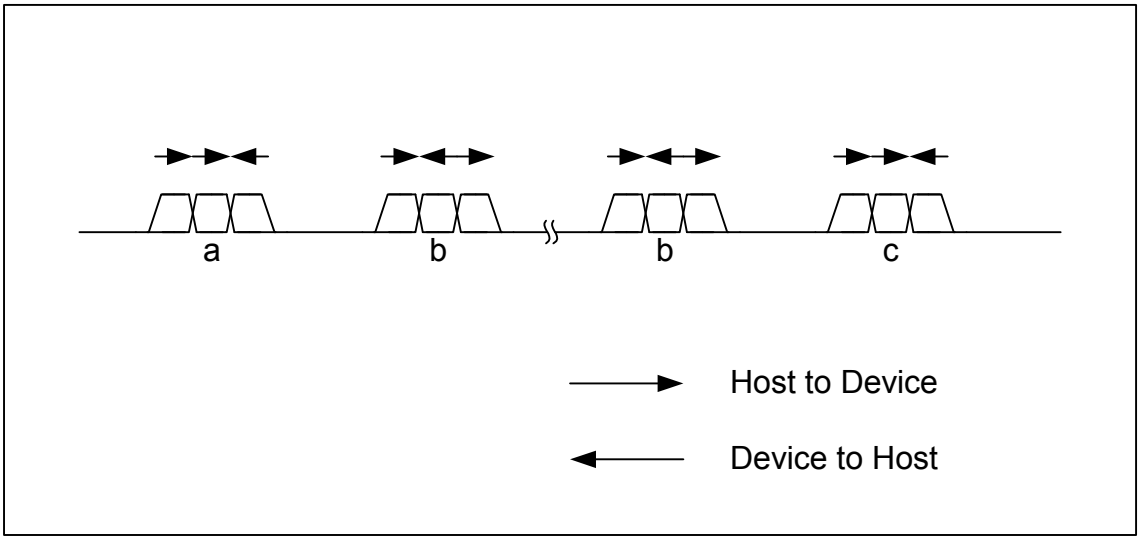


Fig. 1-35 Control transfers with data stage in the IN direction

1.4.4.1 Setup stage

The setup stage is performed using the SETUP transaction. For detailed information on the SETUP transaction, refer to "1.4.3.1 SETUP Transaction."

1.4.4.2 Data Stage/Status Stage

Proceed to the next stage once the setup stage is complete.

If the stage is in the IN direction, set the transaction type (H_CH0Config_1.TID) to "IN," set the other basic setting registers appropriately and execute the transaction.

If the stage is in the OUT direction, set the transaction type (H_CH0Config_1.TID) to "OUT," set the other basic setting registers appropriately and execute the transaction.

For the status stage, the IRP data quantity (H_CH0TotalSize_H,L) should be set to "0x0" before executing the transaction.

1.4.4.3 Control Transfer Support Function

This LSI includes a function for automatic stage management of a series of control transfers. Using this function eliminates the need for the firmware to manage each stage as an individual transaction.

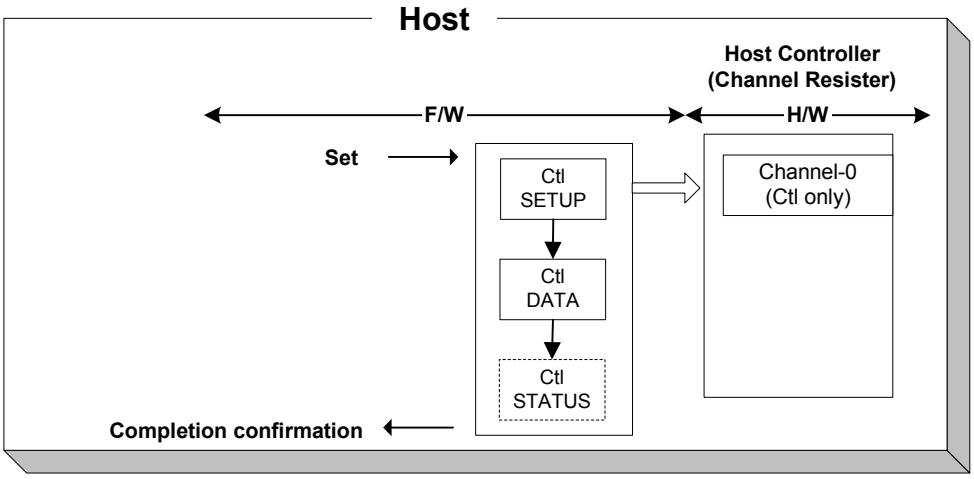


Fig. 1-36 Control transfer support function control

The control transfer support function is enabled for channel CH0. The control transfer using this function is shown below. The firmware performs the processing in steps (1) to (4) and (7).

(1) Set the following basic setting registers appropriately for channel CH0.

Transfer speed (H_CH0Config_0.SpeedMode), maximum packet size (H_CH0MaxPktSize), USB address (H_CH0FuncAdrs.FuncAdrs), endpoint number (H_CH0FuncAdrs.EP_Number), FIFO area (AREAn{n=0-5}StartAdrs_H,L, AREAn{n=0-5}EndAdrs_H,L), FIFO area join (AREAn{n=0-5}Join_1.JoinEP0CH0)

(2) Write the setup data (8 bytes) to the setup registers (H_CH0SETUP_0 to 7).

(3) If the data stage is in the OUT direction, write the data to be sent to the FIFO area joined to CH0. If the data stage is in the IN direction, clear the FIFO area joined to CH0.

(4) Set the control transfer support execute (H_CTL_SupportControl.CTL_SupportGo). The control transfer stage (H_CTL_SupportControl.CTL_SupportState) value is written to the H_CTL_SupportControl register here as “Idle(00b).”

(5) Execute the SETUP transaction using the SETUP register data (8 bytes) (SETUP stage).

(6) Execute the data stage based on the SETUP data.

- If the SETUP data bmRequestType Bit7 is 0, the data in the FIFO area joined to CH0 is sent using an OUT transaction (OUT-direction data stage). The OUT-direction data stage ends when the data quantity indicated by the SETUP data wLength is sent by the OUT transaction.
- If the SETUP data bmRequestType Bit7 is 1, an IN transaction is issued, and the data received is written to the FIFO area joined to CH0 (IN-direction data stage). The IN-direction data stage ends when the data quantity indicated by the SETUP data wLength is received for the IN transaction. The IN-direction data stage also ends when a short packet is received for the IN transaction.
- The data stage is not executed if the SETUP data wLength value is 0x0000.

(7) If the FIFO area joined to CH0 is smaller than the value indicated by SETUP data wLength, the firmware must divide the data stage data for processing.

- If the data stage is in the OUT direction, transactions are no longer issued when there is no more data to be sent in the FIFO area joined to CH0. This means the firmware must write the remaining data to be sent to the FIFO in sequence while checking for FIFO free space.
- If the data stage is in the IN direction, transactions are no longer issued if the FIFO area joined to CH0 has no more free space. This means the firmware must read the data received from the FIFO in sequence while checking the valid data quantity to create FIFO free space.

(8) Execute the status stage based on the SETUP data.

- An IN transaction is issued once the OUT-direction data stage ends (IN-direction status stage).
- A zero-length packet OUT transaction is issued once the IN-direction data stage ends, the received data in the FIFO area joined to CH0 has all been read, and FIFO freed (OUT-direction status stage).

(9) If the control transfer ends normally, a control transfer support execute (H_CTL_SupportControl.CTL_SupportGo) is automatically cleared, and notification of control transfer completion (H_CH0IntStat.CTL_SupportCmp) is issued.

(10) If a transaction error occurs during a control transfer, a control transfer support execute (CTL_SupportControl.CTL_SupportGo) is automatically cleared, aborting the control transfer and issuing notification of control transfer stop status (H_CH0IntStat.CTL_SupportStop). The stage in which the error occurred is indicated by the control transfer stage (H_CTL_SupportControl.CTL_SupportState). The condition code (H_CH0ConditionCode) is set to an appropriate value, and a ChangeCondition status notification (H_CH0IntStat.ChangeCondition bit) is issued.

If control transfer is aborted, control transfer support execution (H_CTL_SupportControl.CTL_SupportGo) is cleared. Status notification is issued once the control transfer abort processing is completed.

If the control transfer is completed up to the status stage here and abort processing is completed, notification of control transfer completion status (H_CH0IntStat.CTL_SupportCmp) is issued.

However, if abort processing is completed but control transfer is not completed up to the status stage, notification of control transfer stop status (H_CH0IntStat.CTL_SupportStop) is issued.

The control transfer stage (H_CTL_SupportControl.CTL_SupportState) indicates the stage for which control transfer was aborted.

If control transfer is resumed from the aborted stage, the control transfer stage (H_CTL_SupportControl.CTL_SupportState) is set in the stage to be resumed (i.e., settings are retained in the stage aborted), and control transfer execution (H_CTL_SupportControl.CTL_SupportGo) is set.

However, if a new control transfer is performed, the control transfer stage (H_CTL_SupportControl.CTL_SupportState) is set to "Idle(00b)," and control transfer support execution (H_CTL_SupportControl.CTL_SupportGo) is set.

The transfer execution bit (H_CH0Config_0.TranGo), toggle sequence bit (H_CH0Config_0.Toggle), transaction type (H_CH0Config_1.TID), and IRP data quantity (H_CH0TotalSize_H,L) are set or updated by the hardware while the control transfer support function is being executed. No data should be written to these bits.

For detailed information on transaction errors, refer to "1.4.3. Transactions."

Table 1-27 shows the settings and status for the control transfer support function.

Table 1-27 Control transfer support function control items and status

Item	Register/bit	Description
Control transfer support execution	H_CTL_SupportControl.CTL_SupportGo	Automatically manages the control transfer stage. For details, refer to "**1.4.4.3 Control Transfer Support Function.**"
Control transfer stage	H_CTL_SupportControl.CTL_SupportState	Indicates the stage being executed for the control transfer support function. Indicates the stage in which the error occurred if a control transfer was aborted due to an error.
Control transfer execution results	H_CH0IntStat.CTL_SupportCmp H_CH0IntStat.CTL_SupportStop	Indicates control transfer execution results using the control transfer support function.
Transaction status	H_CH0IntStat.TotalSizeCmp, H_CH0IntStat.TranACK, H_CH0IntStat.TranErr, H_CH0IntStat.ChangeCondition	Indicates transaction results.
Transaction condition code	H_CH0ConditionCode.ConditionCode	Indicates transaction result specifics.

1.4.5 Bulk/Interrupt/Isochronous Transfers

Bulk transfers for CHa, and bulk, interrupt or isochronous transfers for CHb, CHc, CHd, and CHe can be controlled either as a sequence of individual transactions (see “1.4.3 Transactions”) or as a data flow (see “1.4.6 Data Flow”).

1.4.6 Data Flow

This section describes general data flow control for OUT and IN transfers.

1.4.6.1 OUT Transfers

Set the total OUT transfer data quantity to H_CH0TotalSize_H,L or H_CHx{x=a-e}TotalSize_HH,HL,LH,LL and write the data to be sent by the OUT transfer to the FIFO area joined to each channel. Data can be written to the FIFO using CPU interface register writing or CPU interface DMA writing.

To write data to the FIFO using CPU interface register writing, select one of the FIFO areas joined to the channel by the AREAn{n=0-5}Join_0.JoinCPU_Wr bit. Data can be written to the selected FIFO area by the FIFO_Wr_0,1 or FIFO_ByteWr register and is sent as data packets in the order written. The FIFO free space can be checked using the FIFO_WrRemain_H,L registers. Data cannot be written to the FIFO if the FIFO is full. Always check for free space using the FIFO_WrRemain_H,L registers. Ensure that the quantity of data written does not exceed the free space.

The data packet size sent using OUT transaction is the smaller of H_CH0TotalSize_H,L and H_CH0MaxPktSize for CH0, and the smaller of H_CHx{x=a-e}TotalSize_HH,HL,LH,LL and H_CHx{x=a-e}MaxPktSize_H,L for channel CHx{x=a-e}.

If the data in the FIFO exceeds the data packet data size, an OUT transaction is executed and data is sent. H_CH0TotalSize_H,L or H_CHx{x=a-e}TotalSize_HH,HL,LH,LL are reduced by the amount of the data size sent. When TotalSize reaches zero, H_CHx{x=0,a-e}Config_0.TranGo is automatically cleared to end the transfer, and a TotalSizeCmp status notification (H_CHx{x=0,a-e}IntStat.TotalSizeCmp bit) is issued to the firmware.

OUT transfers can be controlled in this way without controlling individual transactions with firmware.

1.4.6.2 IN Transfers

Set the IN transfer total data quantity to H_CH0TotalSize_H,L or H_CHx{x=a-e}TotalSize_HH,HL,LH,LL.

The expected size for the data packets to be received in the IN transaction is the smaller of H_CH0TotalSize_H,L and H_CH0MaxPktSize for CH0, and the smaller of H_CHx{x=a-e}TotalSize_HH,HL,LH,LL and H_CHx{x=a-e}MaxPktSize_H,L for channel CHx{x=a-e}. If the FIFO free space exceeds the maximum packet size, the IN transaction is executed and data is received. H_CH0TotalSize_H,L or H_CHx{x=0,a-e}TotalSize_HH,HL,LH,LL are reduced by the amount of the data size received. When TotalSize reaches zero, H_CHx{x=0,a-e}Config_0.TranGo is automatically cleared to end the transfer, and a TotalSizeCmp status notification (H_CHx{x=0,a-e}IntStat.TotalSizeCmp bit) is issued to the firmware.

If the received data size exceeds the data length expected, the condition code (H_CHx{x=0,a-e}ConditionCode) is set to "DataOverrun." A ChangeCondition status notification (H_CHx{x=0,a-e}IntStat.ChangeCondition bit) is issued to the firmware. The FIFO is not updated. For control, bulk, or interrupt transfers, H_CHx{x=0,a-e}Config_0.TranGo is automatically cleared to end the transfer for that channel.

If the received data size is smaller than the expected data packet size, the condition code (H_CHx{x=0,a-e}ConditionCode) is set to "DataUnderrun," and a ChangeCondition status notification (H_CHx{x=0,a-e}IntStat.ChangeCondition bit) is issued to the firmware. The FIFO is updated and space reserved by treating the data as already received. For control, bulk, or interrupt transfers, H_CHx{x=0,a-e}Config_0.TranGo is automatically cleared to end the transfer for that channel.

IN transfers can be performed in this way without the need for firmware control of individual transactions.

Data received by IN transfers is written to the FIFO area joined to each channel. FIFO data can be read using CPU interface register reading or CPU interface DMA reading.

To read FIFO data using CPU interface register reading, select one FIFO area joined to the corresponding channel by the AREAn{n=0-5}Join_0.JoinCPU_Rd bit. The selected FIFO area can be read in the order received using the FIFO_Rd_0,1 or FIFO_ByteRd registers. The amount of FIFO data that can be read can be checked using the FIFO_RdRemain_H,L registers. Since data cannot be read from an empty FIFO, the data quantity must always be checked using the FIFO_RdRemain_H,L registers to ensure that reading does not exceed that quantity.

To read FIFO data using CPU interface DMA reading, select one FIFO area for each DMA channel using the AREAn{n=0-5}Join_1.JoinDMA bit. Set the DMA_Control.Dir bit to 1. The selected FIFO area can be read in the order received by executing a DMA sequence for the CPU interface. The amount of FIFO data remaining can be checked using the DMA_Remain_H,L registers. If the FIFO becomes empty, the CPU interface pauses DMA automatically to control the flow.

1.4.7 Zero Length Packet Automatic Issuing Function

Setting the H_CHx{x=a-e}Config_1.AutoZeroLen bit for the OUT transfer channel enables the function that automatically issues zero-length packets.

Table 1-28 shows the settings for the function that automatically issues zero-length packets.

Table 1-28 Settings for the zero-length packet automatic issuing function

Item	Register/bit	Description
Zero-length packet automatic issue	H_CHx{x=a-e}Config_1.AutoZeroLen	Enables the zero-length packet automatic issuing function. This bit is only valid for OUT transfers.

1.4.7.1 Bulk/Interrupt OUT Transfer Zero-Length Packet Automatic Issuing Function

Even when the data size transfer set by the H_CHx{x=a-e}TotalSize_HH,HL,LH,LL registers ends at exactly the Max Packet Size for a channel executing a bulk/interrupt OUT transfer, H_CHx{x=a-e}Config_0.TranGo is not cleared automatically, and the transfer continues. When this channel is scheduled once again, the OUT transaction is executed using a zero-length packet. If this transaction ends normally, H_CHx{x=a-e}Config_0.TranGo is automatically cleared to end the transfer, and a TotalSizeCmp status notification (H_CHx{x=a-e}IntStat.TotalSizeCmp) is issued to the firmware.

1.4.8 Bulk-Only Support Function

This LSI includes a function that automatically manages USB Mass Storage Class (BulkOnly Transport Protocol) command transport (CBW), data transport, and status transport (CSW). This function eliminates the need for firmware control of each transport. Fig. 1-37 illustrates the typical configuration when using the bulk-only support function. Fig. 1-38 illustrates the arrangement when not using the function and controlling transactions as cases of individual transport.

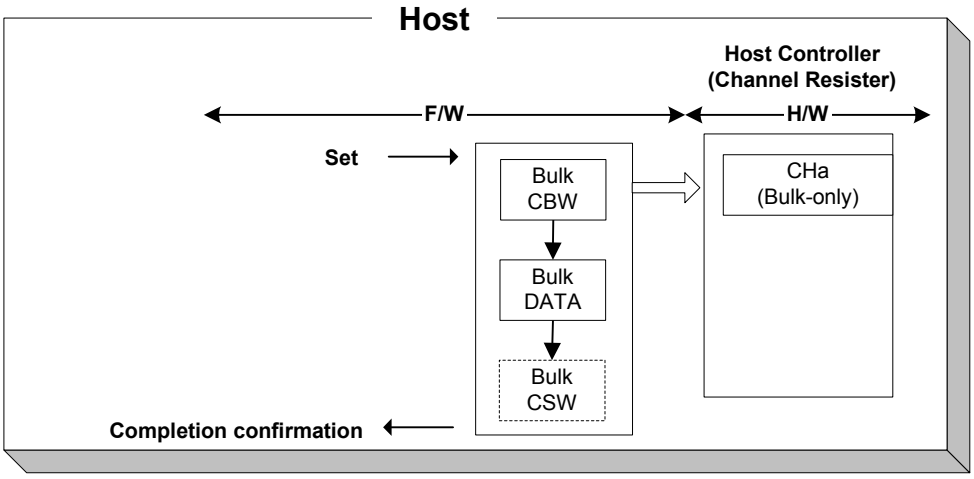


Fig. 1-37 Bulk-only support function control

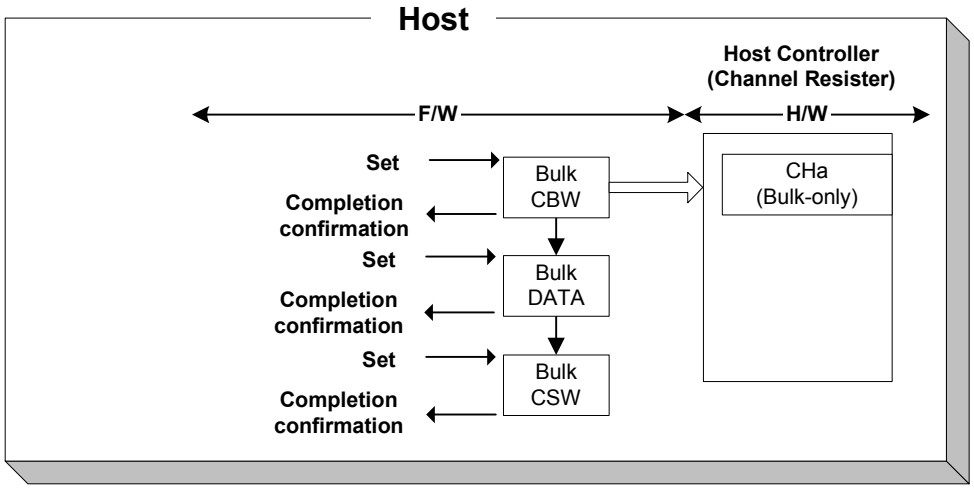


Fig. 1-38 Control when not using bulk-only support function (reference)

The bulk-only support function is valid for channel CHa. The transport processing is as shown below when using this function. The firmware performs steps (1) to (5) and (8).

(1) The following basic settings are set appropriately in channel CHa.
Transfer speed (H_CHaConfig_0.SpeedMode), max packet size (H_CHaMaxPktSize), USB address (H_CHaFuncAdrs.FuncAdrs), FIFO area (AREAn{n=0-5}StartAdrs_H,L, AREAn{n=0-5}EndAdrs_H,L), FIFO area join (AREAn{n=0-5}Join_1.JoinEPaCHa)

(2) The following control registers are set appropriately for the bulk-only support function.
OUT endpoint toggle sequence (H_CHaBO_OUT_EP_Control.OUT_Toggle), OUT endpoint number (H_CHaBO_OUT_EP_Control.OUT_EP_Number), IN endpoint toggle sequence (H_CHaBO_IN_EP_Control.OUT_Toggle), IN endpoint number (H_CHaBO_IN_EP_Control.IN_EP_Number)

(3) Writes CBW data (31 bytes) to the FIFO CBW area.

(4) Sets DMA or PIO settings to the FIFO area joined to CHa.

(5) Sets bulk-only support execution (H_CHaBO_SupportControl.BO_SupportGo).
The transport status (H_CHaBO_SupportControl.BO_TransportState) value is written to the H_CHaBO_SupportControl register as "Idle(00b)."

(6) Sends the CBW area data (31 bytes) to the OUT-direction endpoint indicated by the OUT endpoint number (H_CHaBO_OUT_EP_Control.OUT_EP_Number) using a bulk OUT transaction (command transport).

(7) Executes data transport depending on the CBW data.

- If the CBW data bmCBWFlags Bit7 is 0, the data in the FIFO area joined to CHa is sent to the OUT-direction endpoint indicated by the OUT endpoint number (H_CHaBO_OUT_EP_Control.OUT_EP_Number) by a bulk OUT transaction (OUT-direction data transport). The OUT-direction data transport ends once the data quantity indicated by the CBW data dCBWDataTransferLength has been sent by the OUT transaction.
- If the CBW data bmCBWFlags Bit7 is 1, a bulk IN transaction is issued to the IN-direction endpoint indicated by the IN endpoint number (H_CHaBO_IN_EP_Control.IN_EP_Number), and the data received is written to the FIFO area joined to CHa (IN-direction data transport). The IN-direction data transport ends once the data quantity indicated by the CBW data dCBWDataTransferLength has been received by the IN transaction. The IN-direction data transport also ends when a short packet is received for the IN transaction.
- If the CBW data dCBWDataTransferLength value is 0x00000000, no data transport is performed.

(8) If the FIFO area joined to CHa is smaller than the value indicated by the CBW data wCBWDataTransferLength, the firmware must divide the data transport data before processing it.

- If the data transport is in the OUT direction, transactions are no longer issued when there is no more data to be sent in the FIFO. The firmware must therefore write the remaining data to be sent to the FIFO in sequence while checking for FIFO free space.
- If the data transport is in the IN direction, transactions are no longer issued when there is no remaining free FIFO space. The firmware must read the data from the FIFO in the sequence received to create free space in the FIFO while checking the valid FIFO data quantity.

(9) A bulk IN transaction is issued to the IN-direction endpoint indicated by the IN endpoint number (H_CHaBO_IN_EP_Control.IN_EP_Number), and the data received is written to the FIFO CSW area (status transport). The data quantity received in the status transport reflects in the status transport received data quantity (H_CHaBO_CSW_RcvDataSize.CSW_RcvDataSize).

- Status transport can be executed once the OUT-direction data transport has ended.
- Status transport can be executed once all data received in the FIFO has been read out and the FIFO emptied when the IN-direction data transport has ended.

(10) The CSW data received in the status transport is checked. The details checked are as shown below:

- The received CSW data length is 13 bytes.
- CSW dCSWSignature is 0x53425355.
- CSW dCSWTag matches CBW dCBWTag.
- The BCSWStatus value is 0x00.

If any of the above conditions are not fulfilled, the bulk-only support execute (H_CHaBO_SupportControl.BO_SupportGo) is automatically cleared, and the bulk-only support function is stopped. Bulk-only support stop status notification (H_CHaIntStat.BO_SupportStop) is issued. The data received in the CSW area can be read out using the RAM_Monitor function.

(11) If the status transport ends normally, the bulk-only support execute (H_CHaBO_SupportControl.BO_SupportGo) is automatically cleared and bulk-only support end status notification (H_CHaIntStat.BO_SupportCmp) is issued.

(12) If a transaction error is detected in any of the various transports, the bulk-only support execute (H_CHaBO_SupportControl.BO_SupportGo) is automatically cleared, stopping the bulk-only support function and issuing bulk-only support stop status notification (H_CHaIntStat.BO_SupportStop). The transport for which an error occurred is indicated by the transport state (H_CHaBO_SupportControl.BO_TransportState). The condition code (H_CHaConditionCode) is set appropriately, and a ChangeCondition status notification (H_CHaIntStat.ChangeCondition bit) issued.

The bulk-only support execute (H_CHaBO_SupportControl.BO_SupportGo) is cleared if the bulk-only support function is aborted. Status is issued once abort processing is completed for the bulk-only support function.

If the transport is completed up to status transport before abort processing ends, bulk-only support completion status notification (CHaIntStat.BO_SupportCmp) is issued.

However, if abort processing ends before transport is completed up to status transport, bulk-only support stop status notification (H_CHaIntStat.BO_SupportStop) is issued.

The aborted transport is indicated by the transport state (H_CHaBO_SupportControl.BO_TransportState).

To resume the bulk-only support function from the aborted transport, the transport state (H_CHaBO_SupportControl.BO_TransportState) is set to the transport to be resumed (i.e., the settings are retained for the aborted transport) and the bulk-only support execute (H_CHaBO_SupportControl.BO_SupportGo) is set.

To execute a new bulk-only support function, the transport state (H_CHaBO_SupportControl.BO_TransportState) is set to Idle(00b) and the bulk-only support execute (H_CHaBO_SupportControl.BO_SupportGo) is set.

The transfer execute bit (H_CHaConfig_0.TranGo), toggle sequence bit, (H_CHaConfig_0.Toggle), transaction type (H_CHaConfig_1.TID), total size free bit (H_CHaConfig_1.TotalSizeFree), endpoint number (H_CHaFuncAdrs.EP_Number), and IRP data quantity (H_CHaTotalSize_HH,HL,LH,LL) are set or updated by hardware while the bulk-only support function is being executed. No data should be written to these bits.

For detailed information on transaction errors, refer to “1.4.3 Transactions.”

For detailed information on FIFO CBW and CSW areas, refer to “1.6 FIFO Management.”

For detailed information on DMA, refer to “1.7.3.2 DMA0/DMA1(DMA ch.0 / ch.1).”

Table 1-29 shows the settings and status for the bulk-only support function.

Table 1-29 Bulk-only support function settings and status

Item	Register/bit	Description
Bulk-only support execute	H_CHaBO_SupportControl.BO_SupportGo	Executes the bulk-only support function. For details, refer to "**1.4.8 Bulk-Only Support Function.**"
OUT endpoint toggle sequence	H_CHaBO_OUT_EP_Control.OUT_Toggle	Sets the OUT endpoint toggle sequence bit initial settings. Indicates the OUT endpoint toggle sequence bit state during or after the transaction.
OUT endpoint number	H_CHaBO_OUT_EP_Control.OUT_EP_Number	Sets the OUT endpoint number to a value between 0x0 and 0xF.
IN endpoint toggle sequence	H_CHaBO_IN_EP_Control.IN_Toggle	Sets the IN endpoint toggle sequence bit initial settings. Indicates the IN endpoint toggle sequence bit state during or after the transaction.
IN endpoint number	H_CHaBO_IN_EP_Control.IN_EP_Number	Sets the IN endpoint number to a value between 0x0 and 0xF.
Bulk-only support execute results	H_CHaIntStat.BO_SupportCmp H_CHaIntStat.BO_SupportStop	Indicates the bulk-only support execute results.
Transaction status	H_CHaIntStat.TotalSizeCmp, H_CHaIntStat.TranACK, H_CHaIntStat.TranErr, H_CHaIntStat.ChangeCondition	Indicates the transaction results.
Transaction condition code	H_CHaConditionCode	Indicates the transaction result details.
Transport state	H_CHaBO_SupportControl.BO_TransportState	Indicates the transport state while the bulk-only support function is being executed. Indicates the transport state for which an error occurred if stopped due to an error.
Status transport received data quantity	H_CSW_RcvDataSize	Indicates the data quantity received for the status transport.

1.4.9 Audio Class Assist Function

The audio class assist function is used when 16-bit 2-channel PCM data with a sampling frequency of 44.1 kHz is sent via isochronous transfer with a cycle of 1 ms. When this function is enabled, nine transactions are performed consecutively with a data packet size of 176 bytes, followed by one transaction of 180 bytes. The procedure is then repeated with nine transactions of 176 bytes followed by one transaction of 180 bytes, with the size of data packets varied automatically.

This data packet size change sequence is maintained while the H_CHx{x=b-e}Config_1.Audio441 bit is "1." The H_CHx{x=b-e}Config_1.Audio441 bit should be cleared when initializing this sequence.

Table 1-30 Audio class assist function control settings

Item	Register/bit	Description
Audio class assist function	H_CHx{x=b-e}Config_1.Audio441	Enables the audio class assist function. This function can be used only for OUT-direction isochronous transfers. Do not set this bit to "1" when using other transfers.

1.4.10 Host State Management Support Function

1.4.10.1 Host State

Since the host state must be changed depending on upstream requests or the bus status, the state is managed by firmware. The hardware supports the different settings and negotiations for each state.

Fig. 1-39 illustrates the host state transitions.

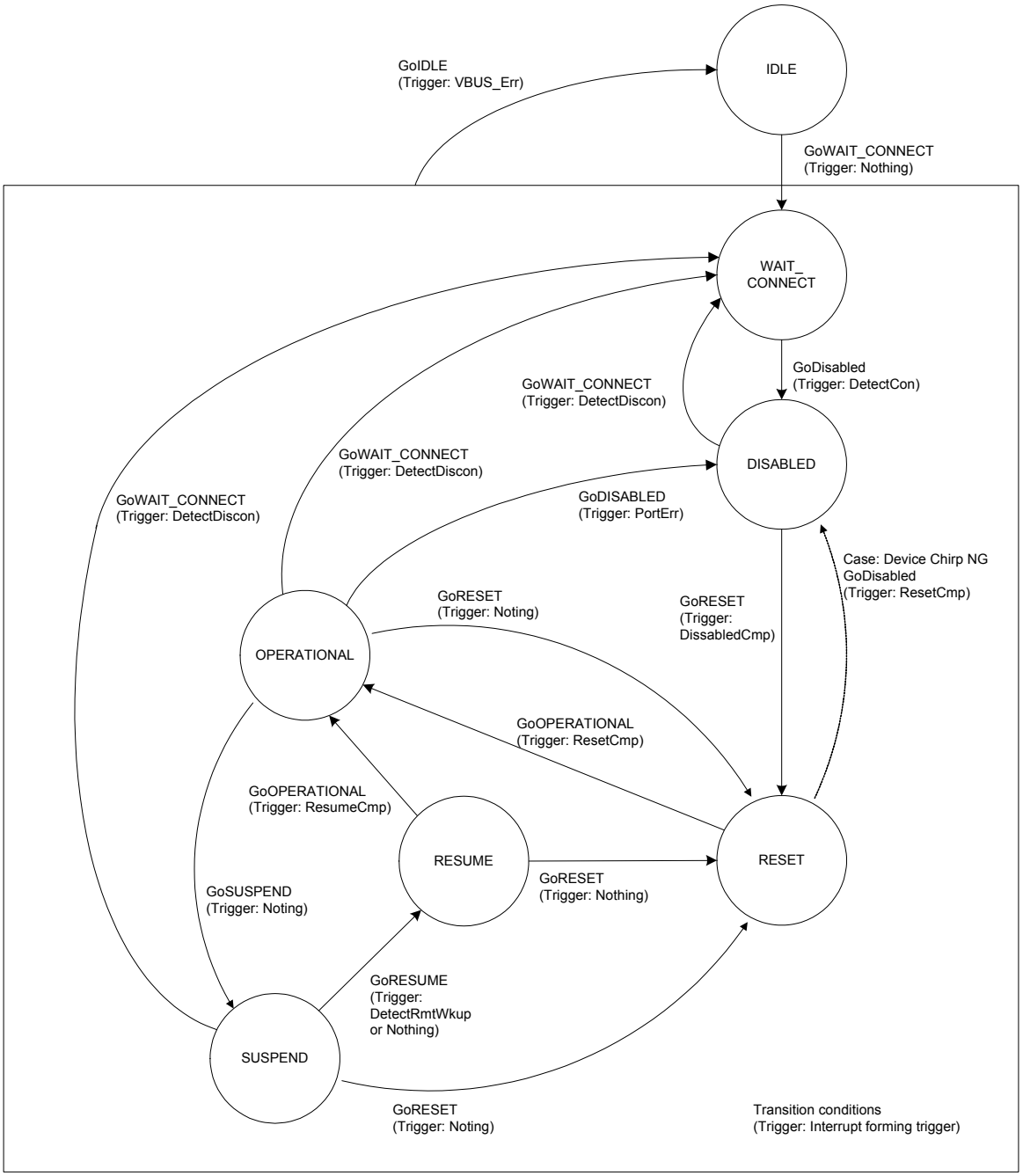


Fig. 1-39 Host state transition diagram

Tables 1-31 and 1-32 show the host state management support function settings and status.

Table 1-31 Host state management support function settings and status

Item	Register/bit	Description
Host state transition execute	H_NegoControl_0.AutoMode	Sets the host state to be changed. Any of the following settings can be set. GoIDLE GoWAIT_CONNECT GoDISABLED GoRESET GoOPERATIONAL GoSUSPEND GoRESUME GoWAIT_CONNECTtoDIS GoWAIT_CONNECTtoOP GoRESETtoOP GoRESUMEtoOP GoSUSPENDtoOP
Host state transition execute cancel	H_NegoControl_0.AutoModeCancel	Stops the current host state processing and remains at that state.
Host state monitor	H_NegoControl_0.HostState	Indicates the current host state (see below). IDLE WAIT_CONNECT DISABLED RESET OPERATIONAL SUSPEND RESUME
VBUS state monitor	H_USB_Status.VBUS_State	Indicates the VBUS state (normal/error).
Remote wakeup receipt enable	H_NegoControl_1.RmtWkupDetEnb	Permits receipt of a remote wakeup.
Chirp completion disable	H_NegoControl_1.DisChirpFinish	Sets the operating mode for when the device chirp fails to terminate within the specified timeframe.
VBUS error detection status	H_SIE_IntStat_0.VBUS_Err	Indicates an error on VBUS.
Connection detection status	H_SIE_IntStat_0.DetectCon	Indicates that a device has been connected to the downstream port.
Disconnection detection status	H_SIE_IntStat_0.DetectDisCon	Indicates that the device has been disconnected from the downstream port.
Remote wakeup detection status	H_SIE_IntStat_0.DetectRmtWkup	Indicates that a remote wakeup signal has been detected from the device.
Device chirp normal detection status	H_SIE_IntStat_0.DetectDevChirpOK	Indicates that the chirp signal from the device is normal.
Device chirp error detection status	H_SIE_IntStat_0.DetectDevChirpNG	Indicates an error in the chirp signal from the device.

(continued)

Table 1-32 Host state management support function settings and status (continued)

Item	Register/bit	Description
Reset completion status	H_SIE_IntStat_1.ResetCmp	Indicates that the USB reset has been completed successfully.
Suspend change completion status	H_SIE_IntStat_1.SuspendCmp	Indicates that the transition to suspend has been completed.
Resume completion status	H_SIE_IntStat_1.ResumeCmp	Indicates that resume has been completed successfully.
Port speed	H_NegoControl_1.PortSpeed	Indicates the downstream port operation speed (HS/FS/LS). Operation of device connected to port.
Line state	H_USB_Status.LineState	Indicates the signal state on the USB cable.
Transceiver selection	H_XcvrControl.XcvrSelect	Selects and enables the HS, FS, or LS transceiver.
Terminal selection	H_XcvrControl.TermSelect	Selects and enables either the HS or FS terminal.
Operating mode	H_XcvrControl.OpMode	Sets the HTM operating mode.

1.4.10.1.1 IDLE

This state initializes the USB host function. It is the default state when the host function is enabled.

In all other states, the state must be changed to this state when VBUS_Err is detected.

To change, write “0x80” to the H_NegoControl_0 register (“1” for H_NegoControl_0.AutoModeCancel and “0x0” for H_NegoControl_0.AutoMode) and halt the state operation during execution. The H_NegoControl_0.AutoModeCancel bit changes to “0” when the stop processing is completed (requires approximately 6 cycles with a 60 MHz clock). This must be checked before setting the register to “0x01” (i.e., setting host state change execute (H_NegoControl0.AutoMode) to “GoIDLE”). Then, the state is changed to this state.

This state sets the following items automatically:

- Immediately stops the USB host transaction execute function.
- Sets the port to FS mode and “NonDriving.”
- Switches off VBUSEN_A.
- Disables all detection functions, including connection detection, disconnection detection, remote wakeup detection, and device chirp detection.

1.4.10.1.2 WAIT_CONNECT

This state waits for a device to be connected to the downstream port.

This state must be switched to wait for device connection following an upstream request in "IDLE" state or device disconnection detection in "DISABLED," "OPERATIONAL" or "SUSPEND" states.

Setting the host state transition execute (H_NegoControl_0.AutoMode) to "GoWAIT_CONNECT" switches to this state.

This state first sets the following items automatically:

- Immediately stops the USB host transaction execute function.
- Sets the port to FS mode and "PowerDown."
- Enables VBUSEN_A.
- Disables all detection functions, including connection detection, disconnection detection, remote wakeup detection, and device chirp detection.

The state then waits for the bus power device internal power supply stabilization time, automatically enables the connection detection function, and waits for device detection. The time taken from switching VBUSEN_A on until device connection detection is not managed by hardware. This time must be managed by firmware.

The line state can be referenced as "J" when an FS or HS device is connected. However, if an LS device is connected, the line state can be referenced as "K." Device connection is detected if either of these states persists for more than 2.5 μs, and the port is switched to LS mode if an LS device connection is detected.

The disconnection detection function is automatically enabled once a connection is detected.

Connection detection status notification (H_SIE_IntStat_0.DetectCon) is issued to the firmware if no disconnection is detected during the debounce interval, and the connection and disconnection detection functions are automatically disabled. However, if a disconnection is detected, the disconnection detection function is automatically disabled and connection detection automatically repeated.

1.4.10.1.3 DISABLED

This state prevents the sending or receiving of bus signals while a device is connected to the downstream port.

The LSI switches to this state when a connection is detected in “WAIT_CONNECT” state, when a chirp is detected from a device with an error in a “RESET” state, or when a port error is detected in the “OPERATIONAL” state.

Setting the host state transition execute (H_NegoControl_0.AutoMode) to “GoDISABLED” switches to this state.

This state sets the following items automatically.

- Waits for the current transaction to end and stops the USB host transaction execute function.
- The port is switched to FS mode if in HS mode when switching to this state, and the port connects in the same mode if in FS or LS mode.
- The port is switched to “PowerDown.”

The following processing is then automatically performed after the end of the disconnection detection disable period.

- Turns on the disconnection detection function.
- Issues disabled transition completion status notification (H_SIE_IntStat_1.DisabledCmp).

1.4.10.1.4 RESET

This state issues a USB reset to the downstream port.

The LSI is switched to this state and issues a USB reset when the disabled change completion status is issued in “DISABLED” state.

This state can be switched to from any USB state (“OPERATIONAL,” “SUSPEND,” or “RESUME”) if requested from upstream.

The LSI is switched to this state if the host state change execute (H_NegoControl_0.AutoMode) is set to “GoRESET.”

The following items are automatically set in this state.

- The USB host transaction execute function is stopped after waiting for the current transaction to be completed.
- The port is set to HS mode and “NormalOperation” (reset signal SE0 is driven for the USB cable signal).
- The connection detection, disconnection detection, and remote wakeup detection functions are disabled.
- The device chirp detection function is enabled.

Chirps from the device are detected using “HS K.” A normal chirp is detected if the line state is detected continuously as “K” for at least 2.5 μs and ends within the specified duration after the USB reset is issued. If it does not end within the specified duration, the chirp is detected as an error.

The following processing is performed automatically based on detection results.

(1) If a normal chirp is detected from the device
“HS K” (Chirp K) and “HS J” (Chirp J) are sent alternately and continuously by the host on completion of the chirp from the device. Reset completion status notification (H_SIE_IntStat_1.ResetCmp) is issued to the firmware once the host has finished sending the chirp.
The port remains in HS mode.

(2) If an error chirp is detected from the device
Device chirp error detection status notification (H_SIE_IntStat_0.DetectDevChirpNG) is issued to the firmware once the specified time has elapsed. One of two subsequent operating modes can be selected using the chirp completion disable (H_NegoControl_1.DisChirpFinish) setting. For details, refer to “1.4.10.2.4.2 If an Error Device Chirp is Detected.”

(3) If the connected device is FS and no chirp is detected from the device
The port is set to FS mode after issuing the USB reset for the specified timeframe. Reset completion status notification (H_SIE_IntStat_1.ResetCmp) is issued to the firmware.

(4) If the connected device is LS
The port is set to LS mode after issuing the USB reset for the specified timeframe. Reset completion status notification (H_SIE_IntStat_1.ResetCmp) is issued to the firmware.

1.4.10.1.5 OPERATIONAL

This state executes the USB transaction.

This state is switched to and the transaction executed after “RESET” or “RESUME” ends.

This state is switched to if the host state change execute (H_NegoControl_0.AutoMode) is set to “GoOPERATIONAL.”

The following items are automatically set in this state:

- The port is set to “NormalOperation.”
- The USB host transaction execute function is enabled.
- The disconnect detection function is enabled.

1.4.10.1.6 SUSPEND

This state suspends the USB.

This state is switched to from “OPERATIONAL” when use of the USB bus is stopped.

This state is switched to if the host state change execute (H_NegoControl_0.AutoMode) is set to “GoSUSPEND.”

The following items are automatically set in this state:

- The disconnect detection and remote wakeup detection functions are disabled.
- The USB host transaction execute function is stopped after waiting for the current transaction to end.
- The port is set to FS mode if currently in HS mode; the port remains in the same mode if currently in FS or LS mode.
- The port is switched to “PowerDown.”

The following processing is then automatically performed after the disconnect or remote wakeup detection disable time period has elapsed.

- The disconnect detection function is enabled.
- The remote wakeup detection function is enabled if the remote wakeup receipt permission (H_NegoControl_1.RmtWkupDetEnb) is enabled.
- Suspend change completion status notification (H_SIE_IntStat_1.SuspendCmp) is issued.

Remote wakeup detection status notification (H_SIE_IntStat_0.DetectRmtWkup) is then issued to the firmware on detection of a remote wakeup signal (“K” for at least 2.5 μs) if the remote wakeup receipt permission (H_NegoControl_1.RmtWkupDetEnb) is enabled.

1.4.10.1.7 RESUME

This state issues a USB resume signal to the downstream port.

This state is switched to from “SUSPEND” to recover the USB device from the suspend state.

This state is entered when the host state change execute (H_NegoControl_0.AutoMode) is set to “GoRESUME.”

The disconnect and remote wakeup detection functions are automatically disabled in this state.

A resume signal (“K”) is issued for the specified duration.

When the resume signal has been issued, the port reverts to the mode setting prior to entering “SUSPEND,” and returns to “NormalOperation.”

Resume completion status notification (H_SIE_IntStat_1.ResumeCmp) is issued to the firmware.

1.4.10.2 Detection Functions

1.4.10.2.1 VBUS Error Detection

A VBUS error is detected in the form of a level change (H to L) at the VBUSFLG_A input terminal. The procedure for detecting a VBUS error is given below. The LSI hardware automatically performs steps (2) below.

(1) The VBUSFLG_A (external USB power switch error occurrence flag) input terminal changes to L (error occurrence) (T0).

(2) VBUS error detection status notification (USB_HostIntStat.VBUS_Err) is issued to the firmware (T0).

VBUS must be switched off immediately when the host detects the VBUS error. The firmware therefore writes 0x80 to the H_NegoControl_0 register ("1" to H_NegoControl_0.AutoModeCancel and "0x0" to H_NegoControl_0.AutoMode) when identifying the VBUS error detection status to stop the state operation currently being performed. The H_NegoControl_0.AutoModeCancel bit changes to "0" when stop processing is completed (requires approximately 6 cycles with a 60 MHz clock). This must be confirmed to be "0" before setting the register to "0x01" (i.e., setting host state change execute (H_NegoControl_0.AutoMode) to "GoIDLE"). This switches to the IDLE state, disables VBUSEN_A terminal logic, and allows the VBUS to be switched off.

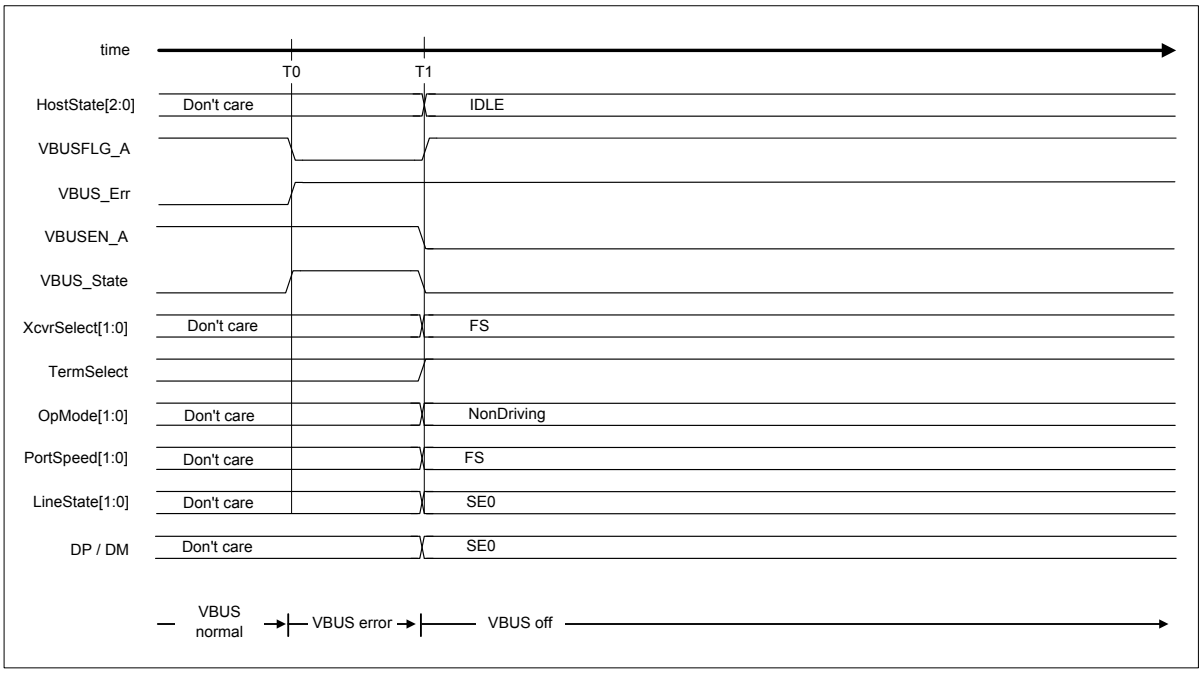


Fig. 1-40 VBUS error detection timing

Table 1-33 VBUS error detection timing values

Timing Parameter	Description	Value
T0	VBUSFLG_A (external USB power switch error occurrence flag) input terminal changes to L (error). VBUS error detection status notification (USB_HostIntStat.VBUS_Err) is issued. (H/W)	0 (reference)
T1 (reference)	Write 0x01 to change to “IDLE” state after writing 0x80 to H_NegoControl_0. (F/W)	T1

1.4.10.2.2 Disconnect Detection

This detects device disconnection while in “DISABLED,” “OPERATIONAL,” or “SUSPEND” state.

When a disconnection is detected, change the host state to “WAIT_CONNECT” to repeat from connect detection without switching off VBUS. Change the host state to “IDLE” to switch off the VBUS.

1.4.10.2.2.1 If HS Device is Disconnected

HS device disconnection is detected in the “OPERATIONAL” state.

The following actions occur if an HS device is disconnected. The LSI hardware automatically performs steps (2) to (3) below.

(1) The device is disconnected (T0).

(2) Disconnection detection is performed for the uSOF(HS_SOF) EOP time period, and the device is determined to be disconnected if identified as disconnected consecutive three times in succession (T1).

(3) Disconnect detection status notification (H_SIE_IntStat_0.DetectDiscon) is issued to the firmware (T1).

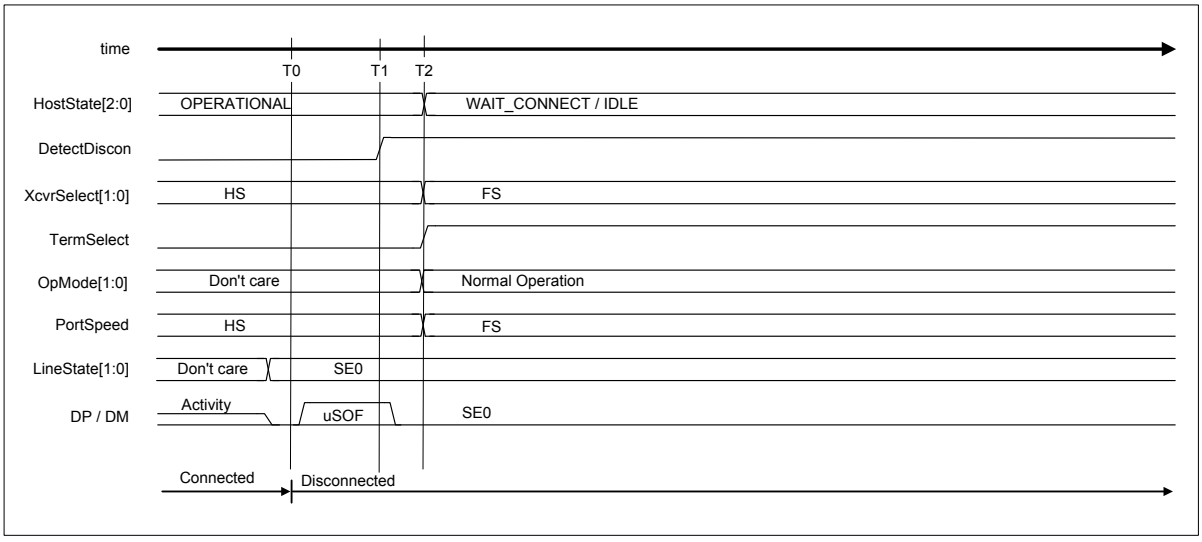


Fig. 1-41 Disconnect detection timing (HS mode)

Table 1-34 Disconnect detection timing values (HS mode)

Timing Parameter	Description	Value
T0	Device is disconnected.	0 (reference)
T1	Disconnect detection status notification (H_SIE_IntStat_0.DetectDiscon) is issued. (H/W)	T1
T2 (reference)	Set the host state change execute (H_NegoControl_0.AutoMode) to “GoWAIT_CONNECT.” (F/W)	T2

1.4.10.2.2.2 If FS or LS Device is Disconnected

FS or LS device disconnection is detected in “DISABLED,” “OPERATIONAL,” and “SUSPEND” states.

The following actions occur if an FS or LS device is disconnected. The LSI hardware automatically performs steps (2) to (3) below.

(1) The device is disconnected (T0).

(2) The disconnection is detected from the signal line state (T1).

(3) Disconnect detection status notification (H_SIE_IntStat_0.DetectDiscon) is issued to the firmware (T1).

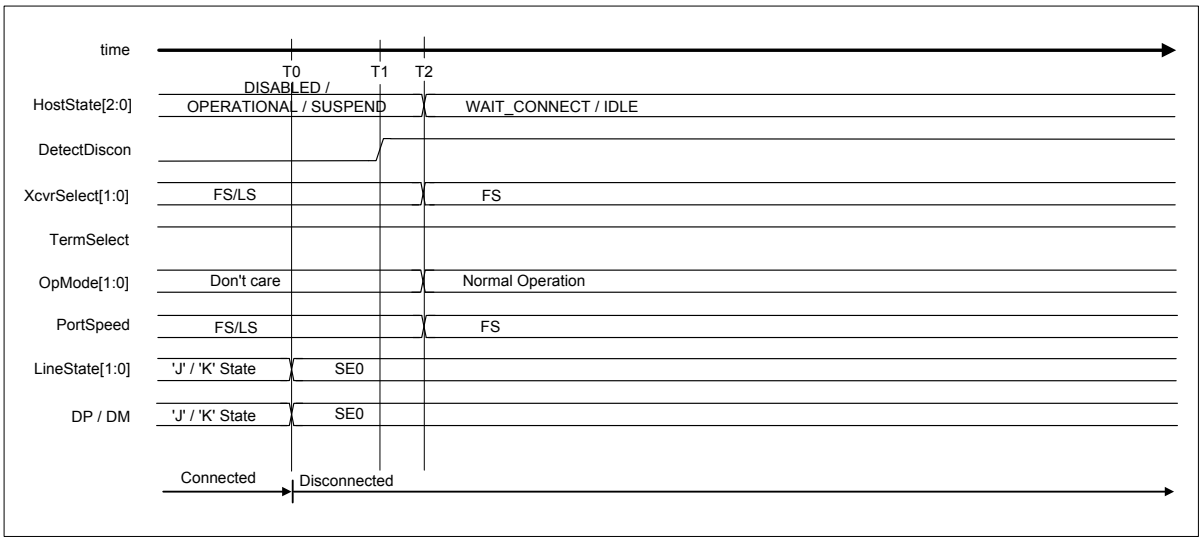


Fig. 1-42 Disconnect detection timing (FS or LS mode)

Table 1-35 Disconnect detection timing values (FS or LS mode)

Timing Parameter	Description	Value
T0	Device is disconnected.	0 (reference)
T1	Disconnect detection status notification (H_SIE_IntStat_0.DetectDiscon) is issued. (H/W)	T0 + 2.5 μs< T1 {T_{DDIS}}
T2 (reference)	Set the host state change execute (H_NegoControl_0.AutoMode) to “GoWAIT_CONNECT.” (F/W)	No specifications

1.4.10.2.3 Remote Wakeup Detection

This detects remote wakeup in “SUSPEND” state if remote wakeup receipt permission (H_NegoControl_1.RmtWkupDetEnb) is enabled.

1.4.10.2.3.1 If HS Device is Connected

The following actions occur if an HS device is connected. The LSI hardware automatically performs steps (2) to (3) below.

(1) The device starts sending a remote wakeup signal (K) (T0).

(2) The host detects the remote wakeup signal (K) (T1).

(3) Remote wakeup detection status notification (H_SIE_IntStat_0.DetectRmtWkup) is issued to the firmware (T1).

Note that the host must issue the resume signal (“K”) within 1 ms of detection of the device remote wakeup. The firmware must recognize the remote wakeup detection status immediately and set the host state change execute (H_NegoControl_0.AutoMode) to “GoRESUME” within 900 µs.

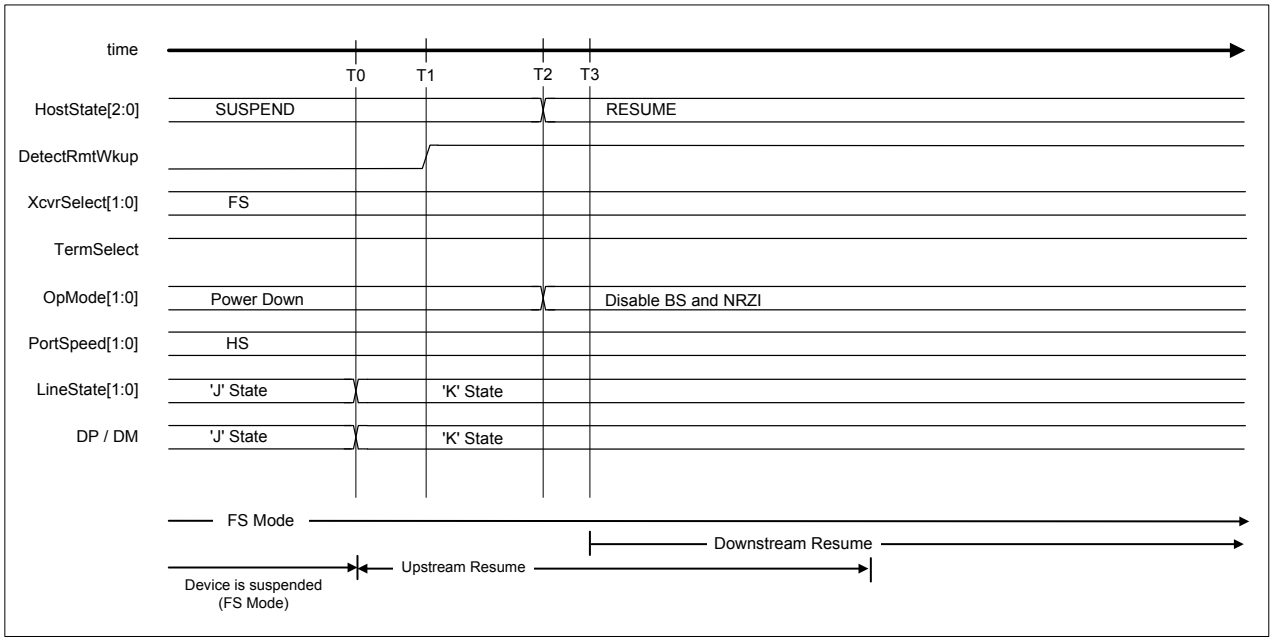


Fig. 1-43 Remote wakeup timing (HS mode)

Table 1-36 Remote wakeup timing values (HS mode)

Timing Parameter	Description	Value
T0	The device starts sending the remote wakeup signal (K).	0 (reference)
T1	The remote wakeup signal (K) is detected. Remote wakeup detection status is issued. (H/W)	T0 + 2.5 µs {T_{URLK}} < T1
T2 (reference)	The host state change execute (H_NegoControl_0.AutoMode) is set to “GoRESUME.” (F/W)	T2 < T1 + 900 µs
T3 (reference)	The host starts sending the resume signal (“K”). (H/W)	T3 < T0 + 1 ms {T_{URSM}}

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.4.10.2.3.2 If FS Device is Connected

The procedures following the connection of an FS device are the same as for an HS device.

For detailed information on the actions taken, refer to "1.4.10.2.3.1 If HS Device is Connected."

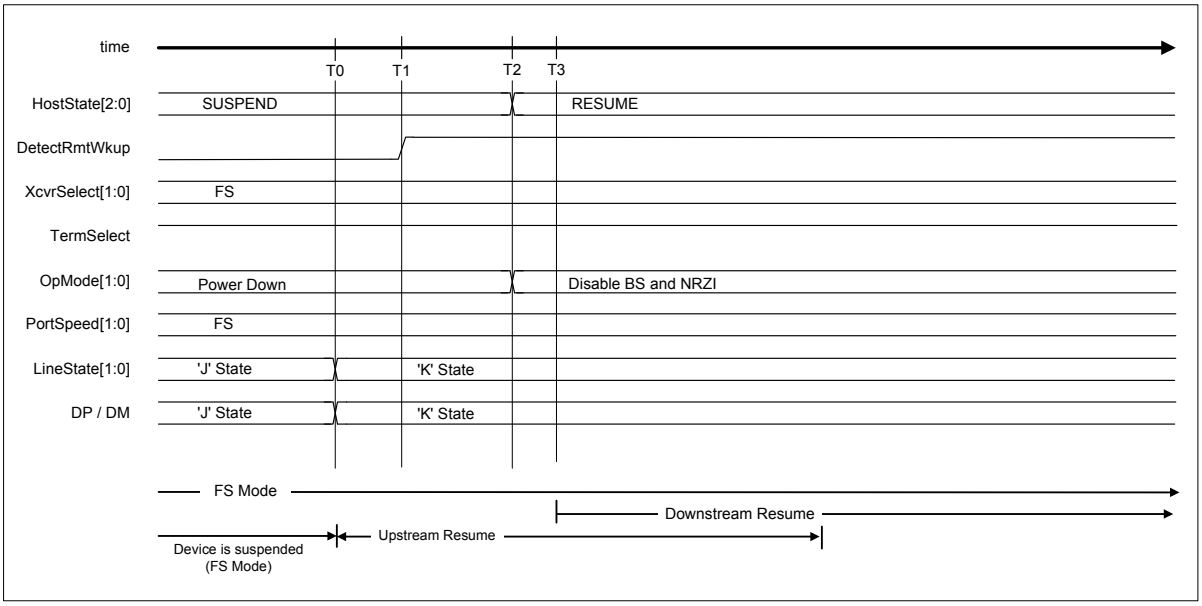


Fig. 1-44 Remote wakeup timing (FS mode)

Table 1-37 Remote wakeup timing values (FS mode)

Timing Parameter	Description	Value
T0	The device starts sending the remote wakeup signal (K).	0 (reference)
T1	The remote wakeup signal (K) is detected. Remote wakeup detection status is issued. (H/W)	T0 + 2.5 µs < T1 {T_{URLK}}
T2 (reference)	The host state change execute (H_NegoControl_0.AutoMode) is set to "GoRESUME." (F/W)	T2 < T1 + 900 µs
T3 (reference)	The host starts sending the resume signal ("K"). (H/W)	T3 < T0 + 1 ms {T_{URSM}}

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.4.10.2.3.3 If LS Device is Connected

The procedures following the connection of an LS device are the same as for an HS device.

For detailed information on the actions taken, refer to "1.4.10.2.3.1 If HS Device is Connected."

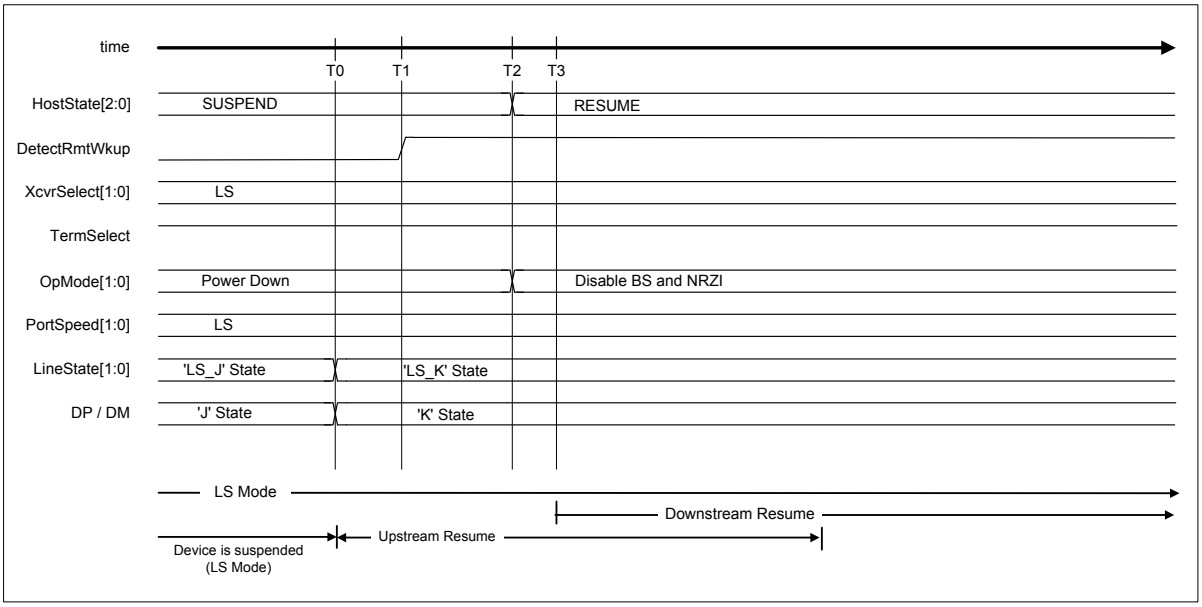


Fig. 1-45 Remote wakeup timing (LS mode)

Table 1-38 Remote wakeup timing values (LS mode)

Timing Parameter	Description	Value
T0	The device starts sending the remote wakeup signal ("K").	0 (reference)
T1	The remote wakeup signal ("K") is detected. Remote wakeup detection status is issued. (H/W)	T0 + 2.5 µs < T1 {T_{URLK}}
T2 (reference)	The host state change execute (H_NegoControl_0.AutoMode) is set to "GoRESUME." (F/W)	T2 < T1 + 900 µs
T3 (reference)	The host starts sending the resume signal ("K"). (H/W)	T3 < T0 + 1 ms {T_{URSM}}

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.4.10.2.4 Device Chirp Detection Function

This detects device chirps.

The device chirp detection function is enabled in a “RESET” state.

1.4.10.2.4.1 If a Correct Device Chirp is Detected

The procedures for device chirp detection are as shown below:

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to “GoRESET” (T0).

(2) The device chirp detection function is enabled (T0).

(3) The device sends a chirp (T1).

(4) The device chirp is recognized if “K” continues for more than the specified duration in the line state (H_USB_Status.LineState[1:0]) (T2).

(5) Device chirp normal detection status notification (H_SIE_IntStat0.DetectDevChirpOK) is issued if the device chirp ends within the specified duration after the reset starts (line state (H_USB_Status.LineState[1:0]) becomes “SE0”) (T3).

(6) The device chirp detection function is disabled if a device chirp is detected (T3).

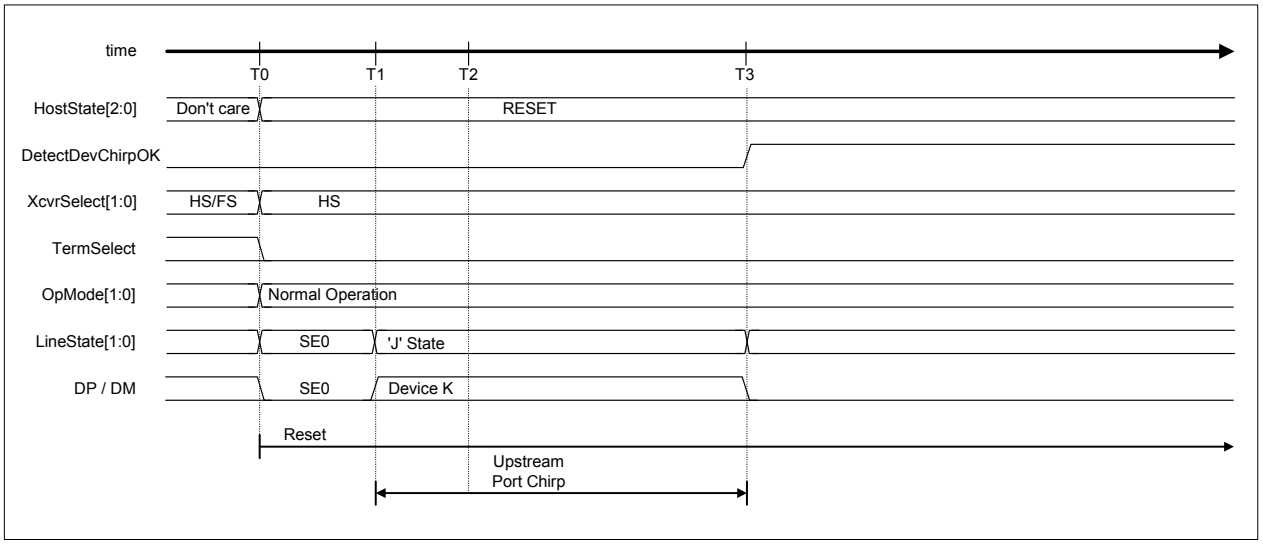


Fig. 1-46 Device chirp timing

Table 1-39 Device chirp timing values

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode is set to “GoRESET.” (F/W)	0 (reference)
T1	The device starts the chirp.	T0 < T1 < T0 + 6.0 ms
T2	The device chirp is recognized. (H/W)	T1 + 2.5 µs {T_{FILT}} < T2
T3	The device ends the chirp. The device chirp detection function is disabled. Device chirp normal detection status notification (DetectDevChirpOK) is issued. (H/W)	T1 + 1.0 ms {T_{UCH}} < T3 < T0 + 7.0 ms {T_{UCHEND}}

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.4.10.2.4.2 If an Error Device Chirp is Detected

If the device chirp fails to terminate within the specified timeframe, the device chirp detection function treats this as an error and issues the appropriate status notification.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to "GoRESET" (T0).

(2) The device chirp detection function is enabled (T0).

(3) The device sends a chirp (T1).

(4) The device chirp is recognized if "K" continues for more than the specified duration in the line state (H_USB_Status.LineState[1:0]) (T2).

(5) Device chirp error detection status notification (H_SIE_IntStat0.DetectDevChirpNG) is issued if the device chirp fails to terminate within the specified duration after the reset starts (T3).

(6) The device chirp detection function is disabled (T3).

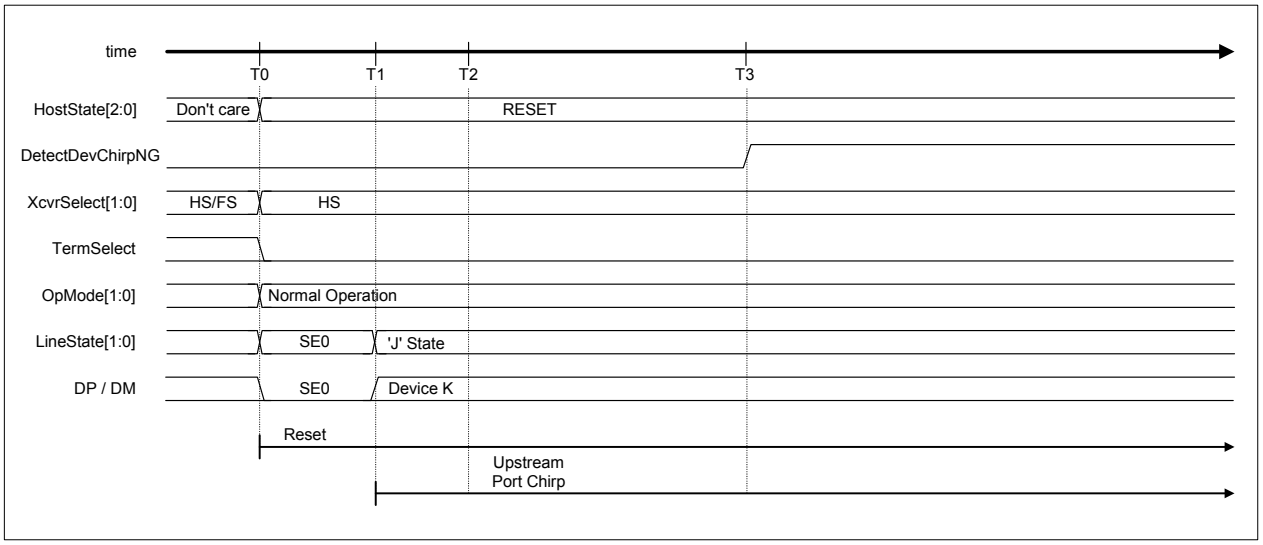


Fig. 1-47 Device chirp timing (NG)

Table 1-40 Device chirp timing (NG) values

Timing Parameter	Description	Value
T0	H_NegoControl_0.AutoMode is set to "GoRESET." (F/W)	0 (reference)
T1	The device starts the chirp.	T0 < T1 < T0 + 6.0 ms
T2	The device chirp is recognized. (H/W)	T1 + 2.5 µs {T_{FILT}} < T2
T3	Device chirp error detection status notification (DetectDevChirpNG) is issued. (H/W)	T0 + 7 ms {T_{UCHEND}} < T3

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.4.10.2.5 Port Error Detection

This detects port errors in the “OPERATIONAL” state.

A port error is assumed if EOP cannot be detected for the packet being received, even after reaching the last (micro) frame.

The host issues port error detection status notification (H_FrameIntStat.PortErr) to the firmware on detecting a port error to stop the transaction immediately. No subsequent transactions are issued, including SOF.

The firmware must perform the following processing if a port error occurs.

(1) Set H_NegoControl_0.AutoMode to “GoDISABLED.”

(2) Set ChipReset.ResetMTM to “1” and reset the transceiver macro.

(3) Set ChipReset.ResetMTM to “0” after at least three cycles have elapsed with a 60 MHz clock and cancel the transceiver macro reset.

1.4.10.3 Individual Host State Management Support Function Explanations

1.4.10.3.1 GoIDLE

The current state is terminated by writing 0x80 to the H_NegoControl_0 register ("1" to H_NegoControl_0.AutoModeCancel and "0x0" to H_NegoControl_0.AutoMode). The H_NegoControl_0.AutoModeCancel bit changes to "0" when the stop processing is completed (requires approximately 6 cycles with a 60 MHz clock). It must be confirmed that the bit has changed to "0" when writing "0x01" to the register (i.e., setting host state change execute (H_NegoControl_0.AutoMode) to "GoIDLE"). The LSI hardware then automatically performs the processing necessary to change to IDLE.

The LSI hardware automatically performs steps (3) to (8) below.

(1) Writes 0x80 to H_NegoControl_0.AutoModeCancel ("1" to H_NegoControl_0.AutoModeCancel and "0x0" to H_NegoControl_0.AutoMode) (T0).

(2) Checks that the H_NegoControl_0.AutoModeCancel bit has changed to "0" and writes "0x01" to H_NegoControl_0 ("0x1" to H_NegoControl_0.AutoMode) (T1).

(3) Sets the host state monitor (H_NegoControl_0.HostState) to "IDLE" (T1).

(4) Switches off VBUSEN_A (T1).

(5) Sets transceiver selection (H_XcvrControl.XcvrSelect) and terminal selection (H_XcvrControl.TermSelect) to FS mode (T1).

(6) Sets the operating mode (H_XcvrControl.OpMode[1:0]) to "NonDriving" (T1).

(7) Immediately stops the USB host transaction execute function (T1).

(8) Disables all detection functions, including connect detection, disconnect detection, remote wakeup detection, and device chirp detection.

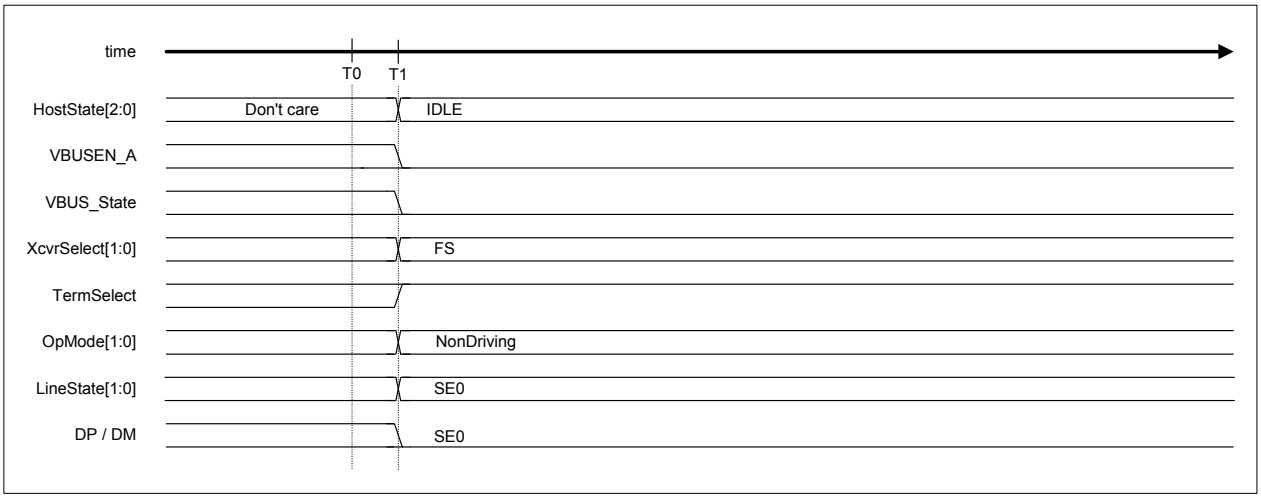


Fig. 1-48 GoIDLE timing

Table 1-41 GoIDLE timing values

Timing Parameter	Description	Value
T0	Stops operation for the state being executed. (F/W)	0 (reference)
T1	Checks that the host state change execute cancel bit is “0,” and sets H_NegoControl_0.AutoMode to “GoIDLE.” (F/W) Switches off VBUSEN_A. Sets the transceiver selection to FS mode. Sets the terminal selection to FS mode. Sets the operating mode to “NonDriving.” Immediately stops the transaction execute function. Disables connect detection, disconnect detection, remote wakeup detection, and device chirp detection. (H/W)	T0 + 5cycle (60 MHz) < T1

1.4.10.3.2 GoWAIT_CONNECT

The LSI hardware automatically performs the processing necessary to change to "WAIT_CONNECT" when "GoWAIT_CONNECT" is set to the host state change execute (H_NegoControl0.AutoMode).

Note that the HS device is connected as an FS device at this point. It operates as an HS device following the reset operation HS Detection Handshake performed later.

1.4.10.3.2.1 If an FS Device is Connected

The following actions are taken if an FS device is connected. The LSI hardware automatically performs steps (2) to (12) below.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to "GoWAIT_CONNECT" (T0).

(2) Sets the host state monitor (H_NegoControl_0.HostState) to "WAIT_CONNECT" (T0).

(3) Enables VBUSEN_A (T0).

(4) Sets transceiver selection (H_XcvrControl.XcvrSelect) and terminal selection (H_XcvrControl.TermSelect) to FS mode (T0).

(5) Sets the operating mode (H_XcvrControl.OpMode[1:0]) to "PowerDown" (T0).

(6) Sets the port speed (H_NegoControl_1.PortSpeed[1:0]) to "FS" (T0).

(7) Waits for the preset duration for the device internal power supply to stabilize, then enables the connect detection function (T1).

(8) The line state (H_USB_Status.LineState[1:0]) indicates "J" when an FS device is connected (T2).

(9) FS device connection is assumed if the line state (H_USB_Status.LineState[1:0]) remains at "J" for at least 2.5 μs (T3).

(10) Enables the disconnect detection function (T3).

(11) Issues connect detection status notification (H_SIE_IntStat_0.DetectCon) if no disconnection is detected during the debounce interval (T4). If a disconnection is detected during this period, the disconnect detection function is disabled and connect detection repeated from step (8). Disconnect detection status notification (H_SIE_IntStat_0.DetectDiscon) is not issued.

(12) Disables the disconnect and connect detection functions (T4).

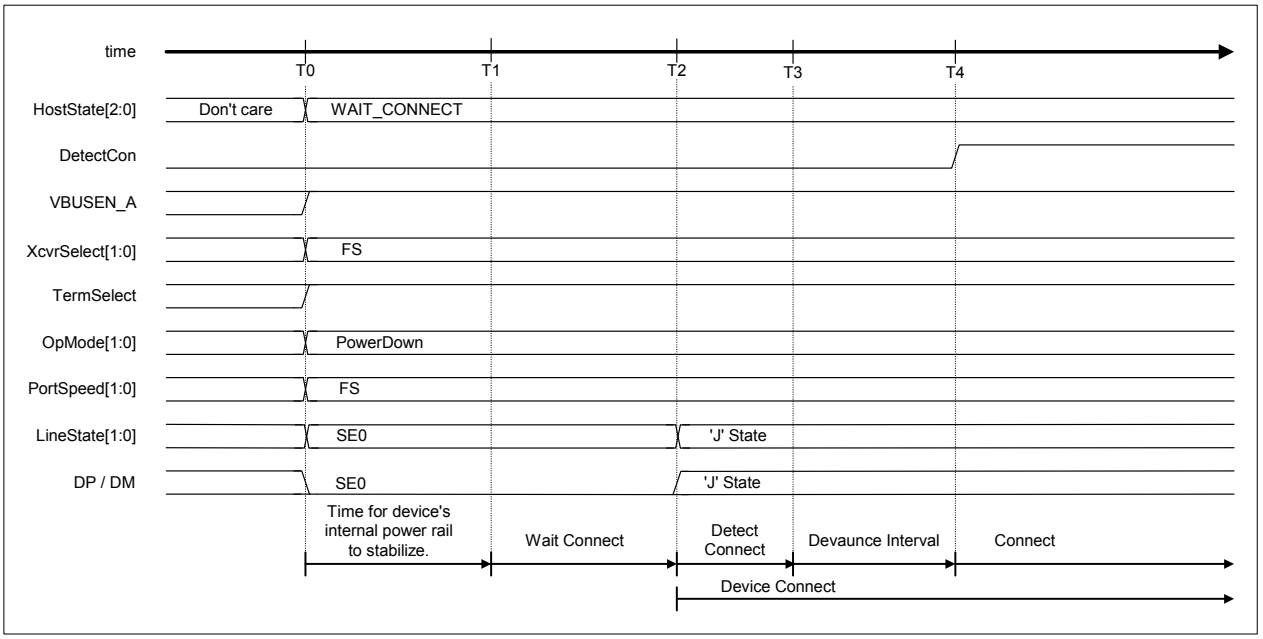


Fig. 1-49 Device attach timing (FS mode)

Table 1-42 Device attach timing values (FS mode)

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to “GoWAIT_CONNECT.” (F/W)	0 (reference)
T1	Enables the connect detection function. (H/W)	T0 + 100 ms {T_{SIGATT}} < T1
T2	The device is connected.	T2
T3	Enables the disconnect detection function. (H/W)	T2 + 2.5 µs {T_{DCNN}} < T3
T4	Issues connect detection status notification (DetectCon). Disables the disconnect and connect detection functions. (H/W)	T3 + 100 ms {T_{ATTDB}} < T4

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.4.10.3.2.2 If an LS Device is Connected

The following actions are taken if an LS device is connected. The LSI hardware automatically performs steps (2) to (14) below.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to “GoWAIT_CONNECT” (T0).

(2) Sets the host state monitor (H_NegoControl_0.HostState) to “WAIT_CONNECT” (T0).

(3) Enables VBUSEN_A (T0).

(4) Sets transceiver selection (H_XcvrControl.XcvrSelect) and terminal selection (H_XcvrControl.TermSelect) to FS mode (T0).

(5) Sets the operating mode (H_XcvrControl.OpMode[1:0]) to “PowerDown” (T0).

(6) Sets the port speed (H_NegoControl_1.PortSpeed[1:0]) to “FS” (T0).

(7) Waits 100 ms for the device internal power supply to stabilize, and enables the connect detection function (T1).

(8) The line state (H_USB_Status.LineState[1:0]) indicates “K” when an LS device is connected (T2).

(9) LS device connection is assumed if the line state (H_USB_Status.LineState[1:0]) remains at “K” for at least 2.5 μs (T3).

(10) Sets transceiver selection (H_XcvrControl.XcvrSelect[1:0]) to “LS” (T3). The line state (H_USB_Status.LineState[1:0]) polarity changes to LS, and the line state (H_USB_Status.LineState[1:0]) indicates “J.”

(11) Sets the port speed (H_NegoControl_1.PortSpeed[1:0]) to “LS” (T3).

(12) Enables the disconnect detection function (T3).

(13) Issues connect detection status notification (H_SIE_IntStat_0.DetectCon) if no disconnection is detected during the debounce interval (T4). If a disconnection is detected during this period, the disconnect detection function is disabled, the transceiver selection (H_XcvrControl.XcvrSelect) and port speed (H_NegoControl_1.PortSpeed[1:0]) are both set to “FS,” and connect detection is repeated from step (8). Disconnect detection status notification (H_SIE_IntStat_0.DetectDiscon) is not issued.

(14) Disables the disconnect and connect detection functions (T4).

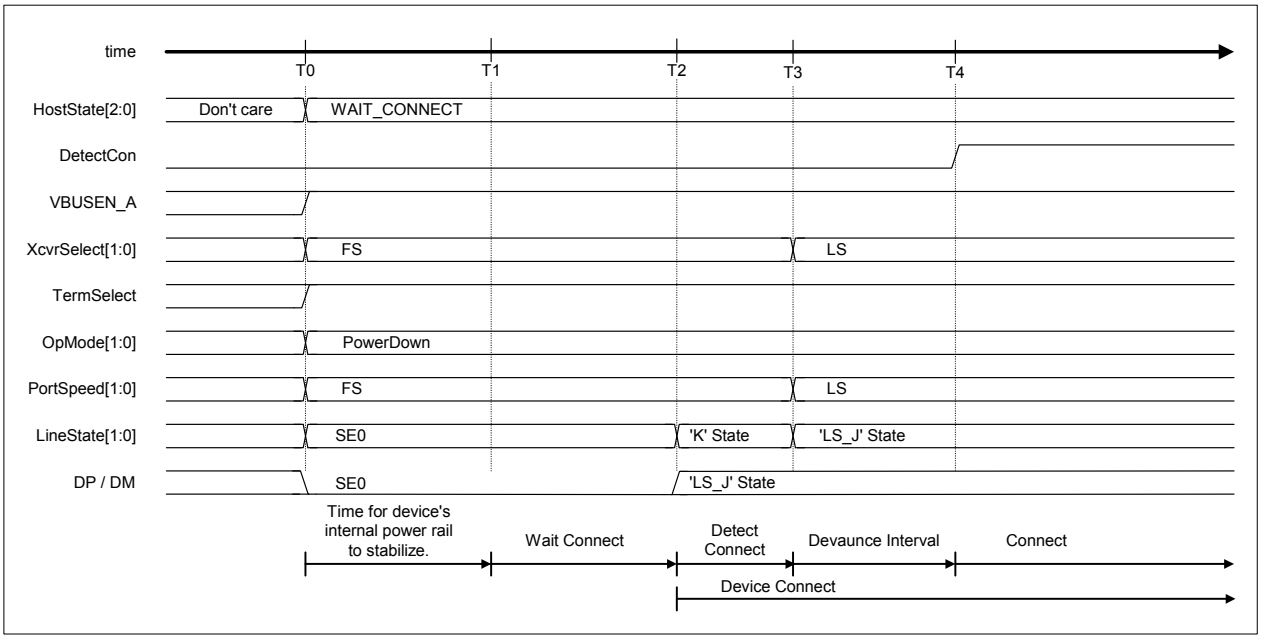


Fig. 1-50 Device attach timing (LS mode)

Table 1-43 Device attach timing values (LS mode)

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to “GoWAIT_CONNECT.” (F/W)	0 (reference)
T1	Enables the connect detection function. (H/W)	T0 + 100 ms {T_{SIGATT}} < T1
T2	The device is connected.	T2
T3	Enables the disconnect detection function. (H/W)	T2 + 2.5 µs {T_{DCNN}} < T3
T4	Issues connect detection status notification (DetectCon). (H/W) Disables the disconnect and connect detection functions. (H/W)	T3 + 100 ms {T_{ATTDB}} < T4

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.4.10.3.3 GoDISABLED

The LSI hardware automatically performs the processing necessary to change to "DISABLED" when "GoDISABLED" is set to the host state change execute (H_NegoControl0.AutoMode).

This state is entered when connection is detected in "WAIT_CONNECT" state, when an error chirp is detected from a device in a "RESET" state, or when a port error is detected in the "OPERATIONAL" state.

1.4.10.3.3.1 If an HS Device is Connected

The following actions are taken if an HS device is connected. The LSI hardware automatically performs steps (2) to (6) below.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to "GoDISABLED" (T0).

(2) Sets the host state monitor (H_NegoControl_0.HostState) to "DISABLED" (T0).

(3) Disables the disconnect detection function (T0).

(4) Waits for the current transaction to end, sets the transceiver selection (H_XcvrControl.XcvrSelect), terminal selection (H_XcvrControl.TermSelect), and port speed (H_NegoControl_1.PortSpeed) to "FS" mode, then sets the operating mode (H_XcvrControl.OpMode[1:0]) to "PowerDown" (T1).

(5) Enables the disconnect detection function (T3).

(6) Issues disabled change completion status notification (H_SIE_IntStat_1.DisabledCmp) to the firmware (T3).

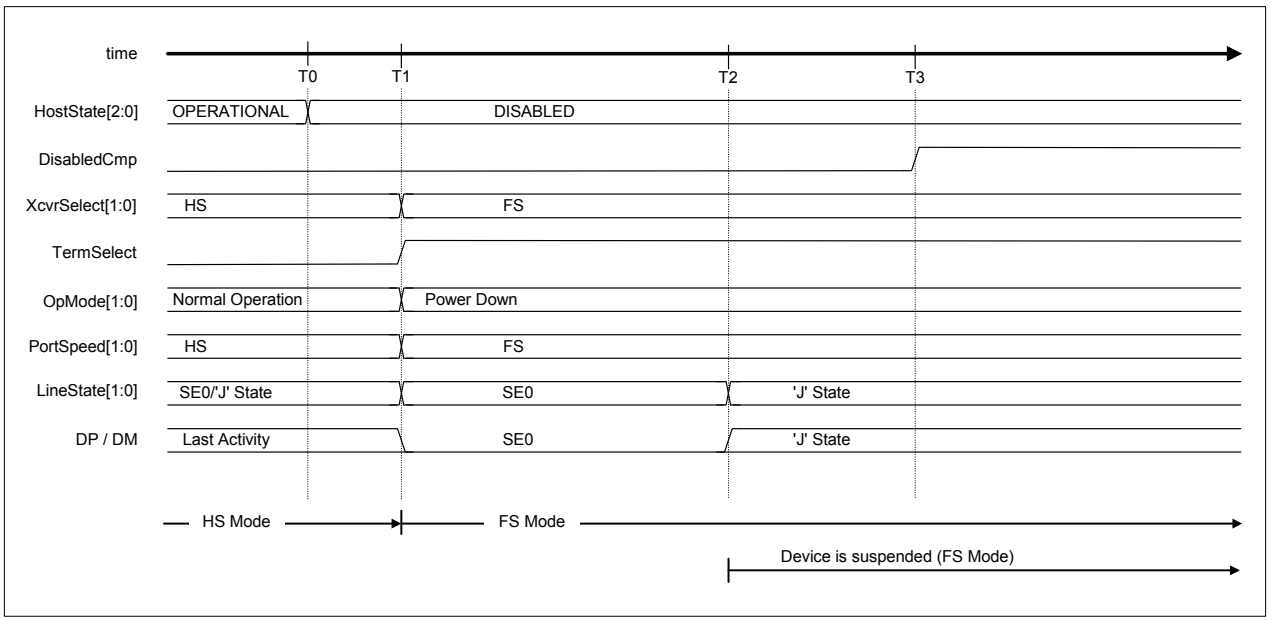


Fig. 1-51 Disabled timing (HS mode)

Table 1-44 Disabled timing values (HS mode)

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to “GoDISABLED.” (F/W) Disables the disconnect detection function. (H/W)	0 (reference)
T1	Sets the port speed for the transceiver selection and terminal selection to “FS” mode after final bus activity ends and sets the operating mode (H_XcvrControl.OpMode[1:0]) to “PowerDown.” (H/W)	T1
T2	The device detects Suspend state and switches to FS mode.	T1 + 3.0 ms < T2 {T_{WTREV}} < T1 + 3.125 ms
T3	Enables the disconnect detection function. (H/W) Issues disabled change completion status. (H/W)	T1 + 4 ms < T3

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.4.10.3.3.2 If an FS Device is Connected

The following actions are taken if an FS device is connected. The LSI hardware automatically performs steps (2) to (6) below.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to “GoDISABLED” (T0).

(2) Sets the host state monitor (H_NegoControl_0.HostState) to “DISABLED” (T0).

(3) Disables the disconnect detection function (T0).

(4) Waits for the current transaction to end, then sets the operating mode (H_XcvrControl.OpMode[1:0]) to “PowerDown” (T1).

(5) Enables the disconnect detection function (T3).

(6) Issues disabled change completion status notification (H_SIE_IntStat_1.DisabledCmp) to the firmware (T3).

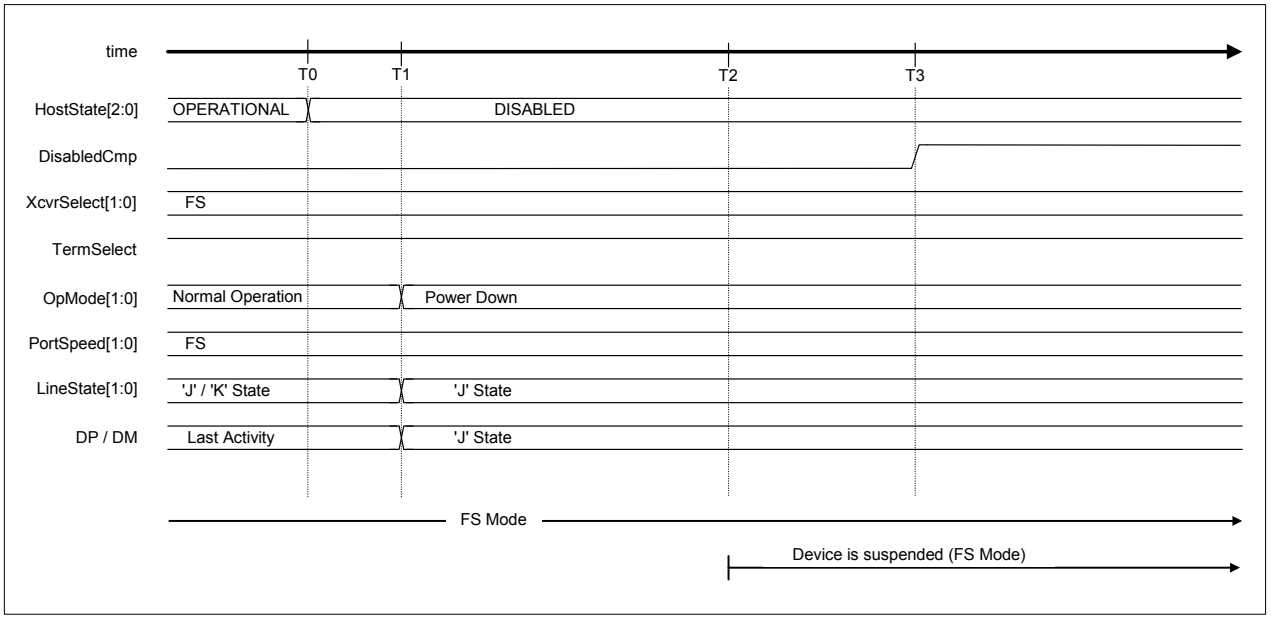


Fig. 1-52 Disabled timing (FS mode)

Table 1-45 Disabled timing values (FS mode)

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to “GoDISABLED.” (F/W) Disables the disconnect detection function. (H/W)	0 (reference)
T1	Sets the operating mode (H_XcvrControl.OpMode[1:0]) to “PowerDown” after final bus activity ends. (H/W)	T1
T2	The device detects Suspend state.	T1 + 3.0 ms < T2 {T_{WTREV}} < T1 + 3.125 ms
T3	Enables the disconnect detection function. Issues disabled change completion status. (H/W)	T1 + 4 ms < T3

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.4.10.3.3.3 If an LS Device is Connected

The following actions are taken if an LS device is connected. The LSI hardware automatically performs steps (2) to (6) below.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to “GoDISABLED” (T0).

(2) Sets the host state monitor (H_NegoControl_0.HostState) to “DISABLED” (T0).

(3) Disables the disconnect detection function (T0).

(4) Waits for the current transaction to end, then sets the operating mode (H_XcvrControl.OpMode[1:0]) to “PowerDown” (T1).

(5) Enables the disconnect detection function (T3).

(6) Issues disabled change completion status notification (H_SIE_IntStat_1.DisabledCmp) to the firmware (T3).

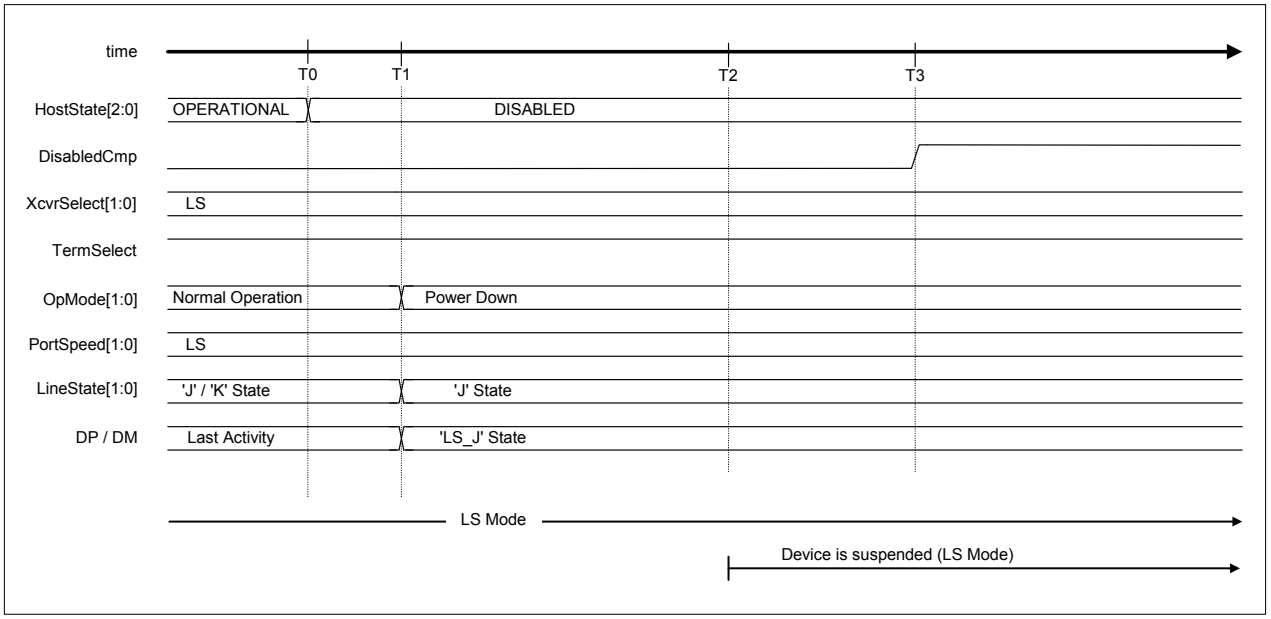


Fig. 1-53 Disabled timing (LS mode)

Table 1-46 Disabled timing values (LS mode)

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to “GoDISABLED.” (F/W) Disables the disconnect detection function. (H/W)	0 (reference)
T1	Sets the operating mode (H_XcvrControl.OpMode[1:0]) to “PowerDown” after final bus activity ends. (H/W)	T1
T2	The device detects Suspend state.	T1 + 3.0 ms < T2 {T_{WTREV}} < T1 + 3.125 ms
T3	Enables the disconnect detection function. Issues disabled change completion status. (H/W)	T1 + 4 ms < T3

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.4.10.3.4 GoRESET

The LSI hardware automatically performs the processing necessary to change to "RESET" when "GoRESET" is set to the host state change execute (H_NegoControl0.AutoMode). When changing to this state from "OPERATIONAL," the hardware waits for the current transaction to end before starting "RESET" processing.

1.4.10.3.4.1 Resetting for HS Device

The following actions are taken to reset an HS device. The LSI hardware automatically performs steps (2) to (14) below.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to "GoRESET" (T0).

(2) Sets the host state monitor (H_NegoControl_0.HostState) to "RESET" (T0).

(3) Sets the transceiver selection (H_XcvrControl.XcvrSelect) and terminal selection (H_XcvrControl.TermSelect) to "HS" mode (T0).

(4) Sets the operating mode (H_XcvrControl.OpMode[1:0]) to "Normal" (T0).

(5) Enables the device chirp detection function (T0).

(6) The device chirp is recognized if activity ("J" state) is detected in the line state (H_USB_Status.LineState[1:0]) for at least 2.5 µs. Device chirp normal detection status notification (H_SIE_IntStat_0.DetectDevChirpOK) is then issued when the device chirp ends within the specified duration after the reset starts (line state (H_USB _Status.LineState[1:0]) changes to no activity (treated as "SE0")) (T2). The hardware determines a normal chirp to have been received from the device if the device chirp normal detection status notification (H_SIE_IntStat_0.DetectDevChirpOK) is set to "1" and the following processing is performed. The device chirp normal detection status notification (H_SIE_IntStat_0.DetectDevChirpOK) must always be cleared to "0" if the HS device is disconnected. If an FS device is connected while the device chirp normal detection status notification (H_SIE_IntStat_0.DetectDevChirpOK) is set to "1" without being cleared, subsequent processing will be performed with the device incorrectly identified as an HS device.

(7) Disables the device chirp detection function (T2).

(8) The host starts to output "Chirp K" after the device chirp ends (T3).

(9) The host starts output of "Chirp J" switching from "Chirp K" (T4).

(10) The host starts output of "Chirp K" switching from "Chirp J" (T5). The host then alternately outputs "Chirp K" and "Chip J."

(11) The device switches to HS mode on detecting the host chirp (T6).
The change in chirp height after point T7 indicates that device HS termination is enabled. The chirp is approximately 800 mV when the device is in FS mode and approximately 400 mV when in HS mode.

(12) The host ends the chirp (T8).

(13) Reset is ended (T9).

(14) Issues reset completion status notification (H_SIE_IntStat_1.ResetCmp) (T9).

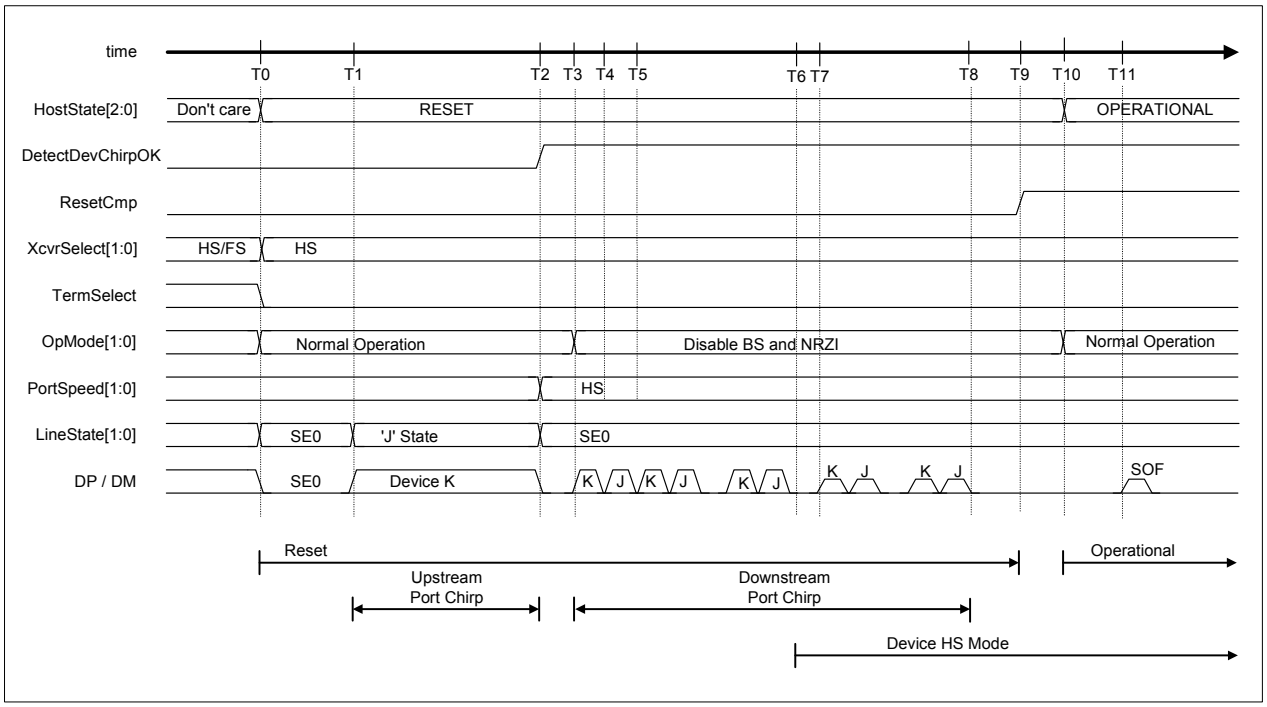


Fig. 1-54 Reset timing (HS mode)

Table 1-47 Reset timing values (HS mode)

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to "GoRESET." (F/W) Enables the device chirp detection function. (H/W)	0 (reference)
T1	The device starts the chirp.	T0 < T1 < T0 + 6.0 ms
T2	The device ends the chirp. Sets the port speed to HS. Disables the device chirp detection function. Issues device chirp normal detection status notification (DetectDevChirpOK). (H/W)	T1 + 1.0 ms {T_{UCH}} < T2 < T0 + 7.0 ms {T_{UCHEND}}
T3	The host outputs the first chirp ("Chirp K"). (H/W)	T2 < T3 < T2 + 100 µs {T_{WTDCH}}
T4	The host starts output of "Chirp J" switching from "Chirp K." (H/W)	T3 + 40 µs {T_{DCHBIT}} < T4 < T3 + 60 µs {T_{DCHBIT}}
T5	The host starts output of "Chirp K" switching from "Chirp J." (H/W)	T4 + 40 µs {T_{DCHBIT}} < T5 < T4 + 60 µs {T_{DCHBIT}}
T6	The device detects the host chirp.	T6
T7	The device switches to HS mode.	T6 < T7 < T6 + 500 µs
T8	The host ends the chirp. (H/W)	T3 + 50 ms {T_{DRSTR}} < T8
T9	Reset is ended. Issues reset completion status notification (ResetCmp). (H/W)	T8 < T9 < T8 + 150 µs
T10 (reference)	Sets H_NegoControl_0.AutoMode to "GoOPERATIONAL." (F/W)	T10 < T9 + 200 µs
T11 (reference)	Sends the first SOF. (H/W)	T10 + 120 µs < T11 < T10 + 130 µs T8 + 100 µs {T_{DCHSE0}} < T11 < T8 + 500 µs {T_{DCHSE0}}

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.4.10.3.4.2 Error Device Chirp Detection

This section discusses operations taken if an error device chirp is detected in the HS Detection Handshake. One of two operating modes can be selected, depending on the chirp completion disable (H_NegoControl_1.DisChirpFinish) setting.

1.4.10.3.4.2.1 If Chirp Completion Disable (H_NegoControl_1.DisChirpFinish) Setting is 0

Host chirping is not performed following error detection. If a device chirp error detection status notification is issued, the firmware awaits reset completion status notification (H_SIE_IntStat_1.ResetCmp), then sets the host state change execute (H_NegoControl_0.AutoMode) to "GoDISABLED" before changing the host state to "DISABLED." The LSI hardware automatically performs steps (2) to (9) below.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to "GoRESET" (T0).

(2) Sets the host state monitor (H_NegoControl_0.HostState) to "RESET" (T0).

(3) Sets the transceiver selection (H_XcvrControl.XcvrSelect) and terminal selection (H_XcvrControl.TermSelect) to HS mode (T0).

(4) Sets the operating mode (H_XcvrControl.OpMode[1:0]) to "Normal" (T0).

(5) Enables the device chirp detection function (T0).

(6) The device chirp is recognized if activity ("J" state) is detected in the line state (H_USB_Status.LineState[1:0]) for at least 2.5 μs. However, an error is assumed if the device chirp fails to terminate within the specified duration after the reset starts, and a device chirp error detection status notification (H_SIE_IntStat_0.DetectDevChirpNG) is issued (T2).

(7) Disables the device chirp detection function (T2).

(8) Reset is ended (T3).

(9) Issues reset completion status notification (H_SIE_IntStat_1.ResetCmp) (T3).

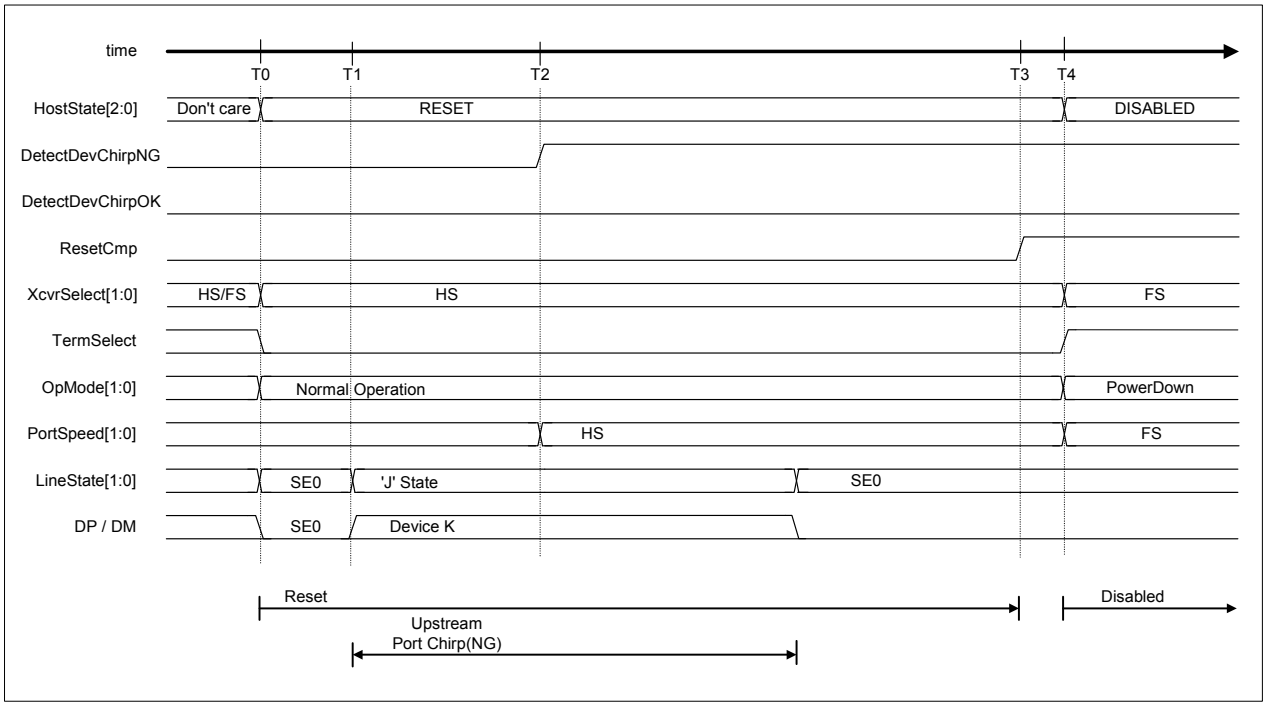


Fig. 1-55 Detect device chirp NG timing (with chirp completion disable set to 0)

Table 1-48 Detect device chirp timing values (with chirp completion disable set to 0)

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to “GoRESET.” (F/W) Enables the device chirp detection function. (H/W)	0 (reference)
T1	The device starts the chirp.	T0 < T1 < T0 + 6.0 ms
T2	Issues device chirp error detection status notification (DetectDevChirpNG). Disables the device chirp detection function. (H/W)	T0 + 7 ms {T_{UCHEND}} < T2
T3	Reset ends. Issues reset completion status notification (ResetCmp). (H/W)	T2 + 50 ms {T_{DRSTR}} < T3
T4 (reference)	Sets H_NegoControl_0.AutoMode to “GoDISABLED.” (F/W)	T4

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.4.10.3.4.2.2 If Chirp Completion Disable (H_NegoControl_1.DisChirpFinish) Setting is 1

Host chirping is performed on completion of device chirping following error detection.

To change the host state to "DISABLED" when using this mode without awaiting reset completion status notification (H_SIE_IntStat_1.ResetCmp), stop the current state by writing 0x80 to the H_NegoControl_0 register ("1" to H_NegoControl_0.AutoModeCancel and "0x0" to H_NegoControl_0.AutoMode). The H_NegoControl_0.AutoModeCancel bit changes to "0" when stop processing is complete (requires approximately 6 cycles with a 60 MHz clock). Confirm that the H_NegoControl_0.AutoModeCancel bit has changed to "0" before writing 0x03 to the register (i.e., set the host state change execute (H_NegoControl_0.AutoMode) to "GoDISABLED").

The LSI hardware automatically performs steps (2) to (15) below.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to "GoRESET" (T0).

(2) Sets the host state monitor (H_NegoControl_0.HostState) to "RESET" (T0).

(3) Sets the transceiver selection (H_XcvrControl.XcvrSelect) and terminal selection (H_XcvrControl.TermSelect) to HS mode (T0).

(4) Sets the operating mode (H_XcvrControl.OpMode[1:0]) to "Normal" (T0).

(5) Enables the device chirp detection function (T0).

(6) The device chirp is recognized if activity ("J" state) is detected in the line state (USB_Host_Status.LineState[1:0]) for at least 2.5 µs. However, an error is assumed if the device chirp fails to terminate within the specified duration after the reset starts, and a device chirp error detection status notification (H_SIE_IntStat_0.DetectDevChirpNG) is issued (T2).

(7) Disables the device chirp detection function (T2).

(8) Device chirp normal detection status notification (H_SIE_IntStat_0.DetectDevChirpOK) is issued when the device chirp ends (i.e., line state (H_USB_Status.LineState[1:0]) changes to no activity (treated as "SE0")) (T3).

(9) The host starts to output "Chirp K" after the device chirp ends (T4).

(10) The host starts output of "Chirp J" switching from "Chirp K" (T5).

(11) The host starts output of "Chirp K" switching from "Chirp J" (T6). The host then alternately outputs "Chirp K" and "Chip J."

(12) The device switches to HS mode on detecting the host chirp (T7). The change in chirp height after point T8 indicates that device HS termination is enabled. The chirp is approximately 800 mV when the device is in FS mode and approximately 400 mV when in HS mode.

(13) The host ends the chirp (T9).

(14) Reset is ended (T10).

(15) Issues reset completion status notification (H_SIE_IntStat_1.ResetCmp) (T10).

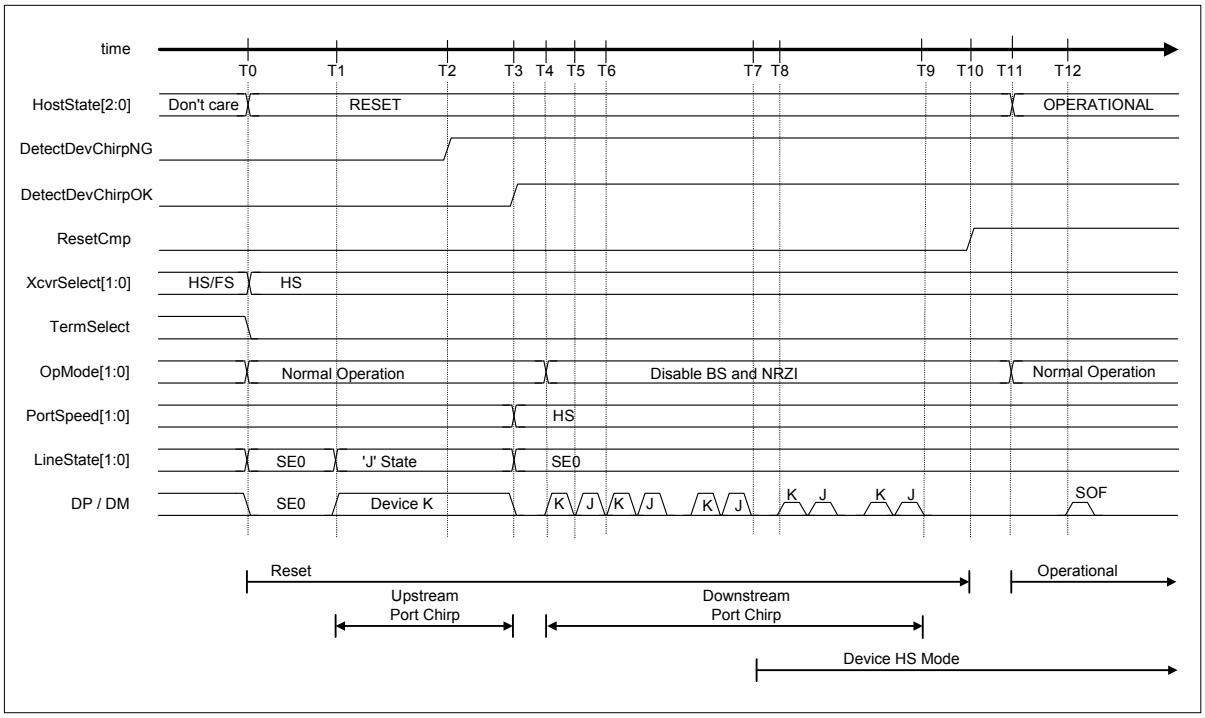


Fig. 1-56 Detect device chirp NG timing (with chirp completion disable set to 1)

Table 1-49 Detect device chirp timing values (with chirp completion disable set to 1)

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to “GoRESET.” (F/W) Enables the device chirp detection function. (H/W)	0 (reference)
T1	The device starts chirping.	T0 < T1 < T0 + 6.0 ms
T2	Issues device chirp error detection status notification (DetectDevChirpNG). Disables the device chirp detection function. (H/W)	T0 + 7 ms {T_{UCHEND}} < T2
T3	The device stops chirping. Sets the port speed to HS. Issues device chirp normal detection status notification (DetectDevChirpOK). (H/W)	T3
T4	The host outputs the first chirp (“Chirp K”). (H/W)	T3 < T4 < T3 + 100 µs {T_{WTDCH}}
T5	The host starts output of “Chirp J” switching from “Chirp K.” (H/W)	T4 + 40 µs {T_{DCHBIT}} < T45 < T4 + 60 µs {T_{DCHBIT}}
T6	The host starts output of “Chirp K” switching from “Chirp J.” (H/W)	T5 + 40 µs {T_{DCHBIT}} < T6 < T5 + 60 µs {T_{DCHBIT}}
T7	The device detects the host chirp.	T7
T8	The device switches to HS mode.	T7 < T8 < T6 + 500 µs
T9	The host ends the chirp. (H/W)	T4 + 50 ms {T_{DRSTR}} < T9
T10	Reset is ended. Issues reset completion status notification (ResetCmp). (H/W)	T9 < T10 < T9 + 150 µs
T11 (reference)	Sets H_NegoControl0.AutoMode to “GoOPERATIONAL.” (F/W)	T11 < T10 + 200 µs
T12 (reference)	Sends the first SOF. (H/W)	T11 + 120 µs < T12 < T11 + 130 µs T9 + 100 µs {T_{DCHSE0}} < T12 < T9 + 500 µs {T_{DCHSE0}}

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.4.10.3.4.3 Resetting for FS Device

Shown below are the procedures for resetting an FS device. The LSI hardware performs steps (2) to (9) automatically.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to “GoRESET” (T0).

(2) Sets the host state monitor (H_NegoControl_0.HostState) to “RESET” (T0).

(3) Sets the transceiver selection (H_XcvrControl.XcvrSelect) and terminal selection (H_XcvrControl.TermSelect) to HS mode (T0).

(4) Sets the operating mode (H_XcvrControl.OpMode[1:0]) to “Normal” (T0).

(5) Enables the device chirp detection function (T0).

(6) The port speed (H_NegoControl_1.PortSpeed[1:0]) is HS/FS, and the connected device is determined to be an FS device if a device chirp is not detected. Transceiver selection (H_XcvrControl.XcvrSelect) and port speed (H_NegoControl_1.PortSpeed[1:0]) are set to “FS” (T1).

(7) Disables the device chirp detection function (T1).

(8) Sets the terminal selection (H_XcvrControl.TermSelect) to “FS” (T2).

(9) Issues reset completion status (H_SIE_IntStat_1.ResetCmp) (T3).

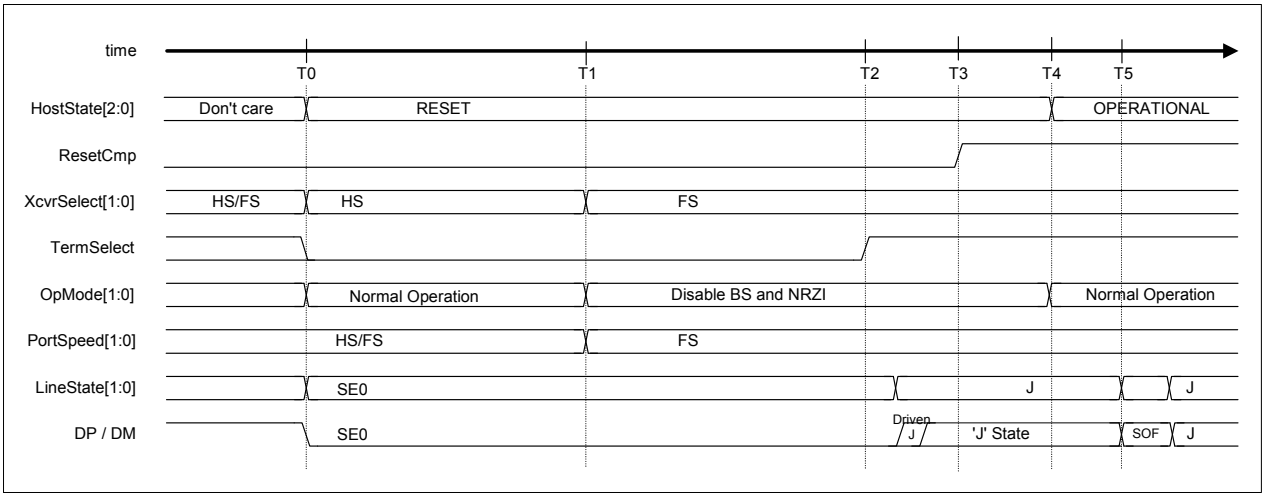


Fig. 1-57 Reset timing (FS mode)

Table 1-50 Reset timing values (FS mode)

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to “GoRESET.” (F/W) Enables the device chirp detection function. (H/W)	0 (reference)
T1	Sets the transceiver selection to “FS.” Sets the port speed to “FS.” Disables the device chirp detection function. (H/W)	T0 + 7.0 ms {T_{UCHEND}} < T1
T2	Sets the terminal selection to “FS.” (H/W)	T0 + 50 ms {T_{DRSTR}} < T2
T3	Issues reset completion status. (H/W)	T2 + 150 µs < T3
T4 (reference)	Sets H_NegoControl_0.AutoMode to “GoOPERATIONAL.” (F/W)	T4
T5 (reference)	Sends the first SOF. (H/W)	T4 + 0.9 ms < T5 < T4 + 1.1 ms (T5 < T2 + 3 ms)

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.4.10.3.4.4 Resetting for LS Device

Shown below are the procedures for resetting an LS device. The LSI hardware performs steps (2) to (7) automatically.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to “GoRESET” (T0).

(2) Sets the host state monitor (H_NegoControl_0.HostState) to “RESET” (T0).

(3) Sets the transceiver selection (H_XcvrControl.XcvrSelect) and terminal selection (H_XcvrControl.TermSelect) to HS mode (T0).

(4) Sets the operating mode (H_XcvrControl.OpMode[1:0]) to “Normal” (T0).

(5) The port speed (H_NegoControl_1.PortSpeed[1:0]) is LS, and so the transceiver selection (H_XcvrControl.XcvrSelect) is set to “LS” (T1).

(6) Sets the terminal selection (H_XcvrControl.TermSelect) to “FS” (T2).

(7) Issues reset completion status (H_SIE_IntStat_1.ResetCmp) (T3).

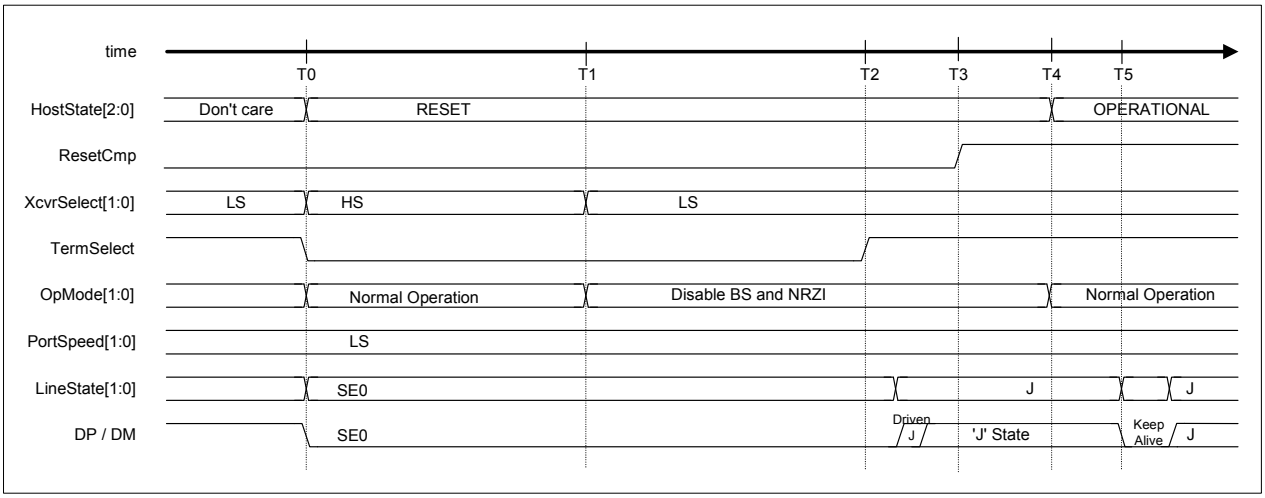


Fig. 1-58 Reset timing (LS mode)

Table 1-51 Reset timing values (LS mode)

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to “GoRESET.” (F/W)	0 (reference)
T1	Sets the transceiver selection to “LS.” (H/W)	T0 + 7.0 ms {T_{UCHEND}} < T1
T2	Sets the terminal selection to “FS.” (H/W)	T0 + 50 ms {T_{DRSTR}} < T2
T3	Issues reset completion status. (H/W)	T2 + 150 µs < T3
T4 (reference)	Sets H_NegoControl_0.AutoMode to “GoOPERATIONAL.” (F/W)	T4
T5 (reference)	Sends the first KeepAlive. (H/W)	T4 + 0.9 ms < T5 < T4 + 1.1 ms (T5 < T2 + 3 ms)

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.4.10.3.5 GoOPERATIONAL

The LSI hardware automatically performs the processing necessary to switch to "OPERATIONAL" when "GoOPERATIONAL" is set to host state change execute (H_NegoControl_0.AutoMode).

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to "GoOPERATIONAL" (T0).

(2) Sets the host state monitor (H_NegoControl_0.HostState) to "OPERATIONAL" (T0).

(3) Sets the operating mode (H_XcvrControl.OpMode[1:0]) to "Normal" to allow USB transactions to be performed (T0).

(4) Enables the disconnect detection function (T0).

(5) Issues the first SOF if the port speed (H_NegoControl_1.PortSpeed[1:0]) is set to "HS" or "FS" (T1). Issues the first KeepAlive if set to "LS" (T1). Subsequent transfers are performed in accordance with the channel settings.

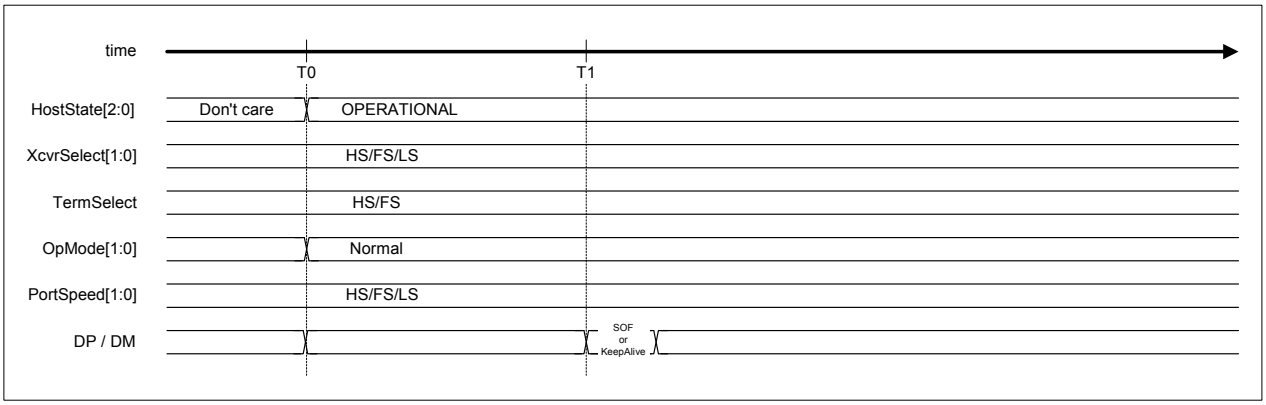


Fig. 1-59 GoOPERATIONAL timing

Table 1-52 GoOPERATIONAL timing values

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to "GoOPERATIONAL." (F/W)	0 (reference)
T1	Sends the first SOF (HS/FS) or KeepAlive (LS).	T0 + 120 µs < T1 (HS) < T0 + 130 µs T0 + 0.9 ms < T1 (FS,LS) < T0 + 1.1 ms

1.4.10.3.6 GoSUSPEND

The LSI hardware automatically performs the processing necessary to switch to “SUSPEND” when “GoSUSPEND” is set to host state change execute (H_NegoControl_0.AutoMode).

1.4.10.3.6.1 If HS Device is Connected

Shown below are the procedures performed when an HS device is connected. The LSI hardware performs steps (2) to (7) automatically.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to “GoSUSPEND” (T0).

(2) Sets the host state monitor (H_NegoControl_0.HostState) to “SUSPEND” (T0).

(3) Disables the disconnect detection function and remote wakeup detection function (T0).

(4) Awaits completion of the transaction currently being executed, sets the transceiver selection (H_XcvrControl.XcvrSelect) and terminal selection (H_XcvrControl.TermSelect) to FS mode, then sets the operating mode (H_XcvrControl.OpMode[1:0]) to “PowerDown” (T1).

(5) Enables the disconnect detection function (T3).

(6) Enables the remote wakeup detection function if remote wakeup receipt permission (H_NegoControl_1.RmtWkupDetEnb) is enabled (T3).

(7) Issues suspend change completion status notification (H_SIE_IntStat_1.SuspendCmp) to the firmware (T3).

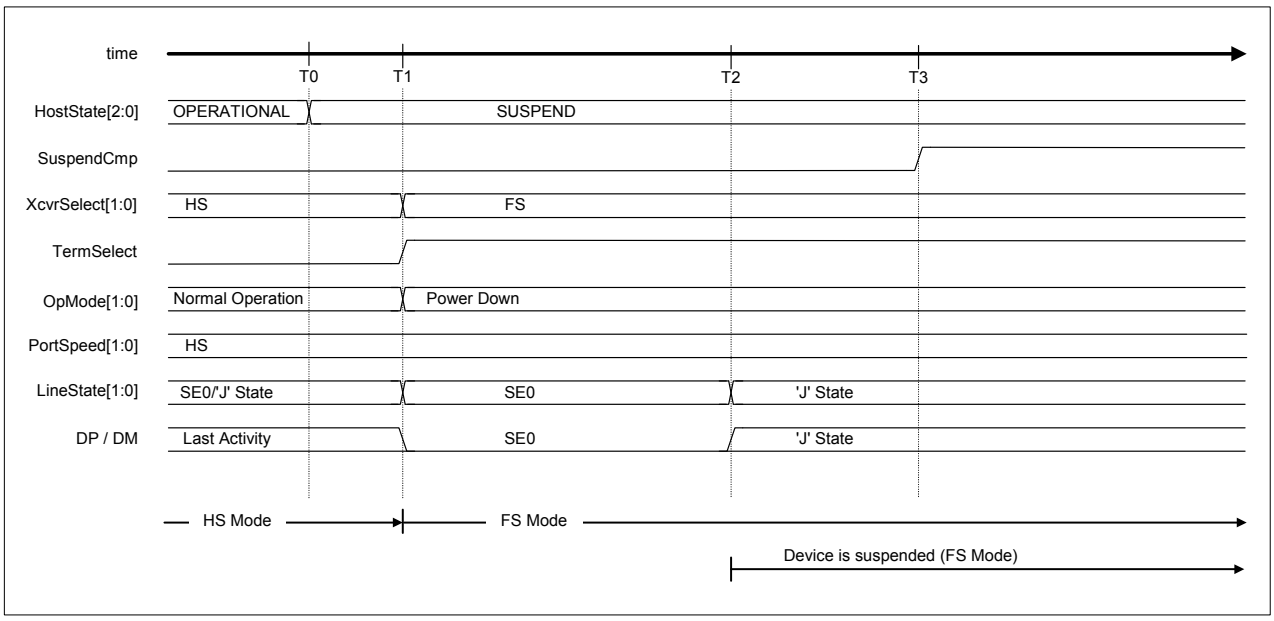


Fig. 1-60 Suspend timing (HS mode)

Table 1-53 Suspend timing values (HS mode)

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to “GoSUSPEND.” (F/W) Disables the disconnect and remote wakeup detection functions. (H/W)	0 (reference)
T1	Sets the transceiver selection and terminal selection to FS mode and sets the operating mode (H_XcvrControl.OpMode[1:0]) to “PowerDown” after final bus activity ends. (H/W)	T1
T2	Detects that the device is in SUSPEND and switches to FS mode.	T1 + 3.0 ms < T2 {T_{WTREV}} < T1 + 3.125 ms
T3	Enables the disconnect detection function. Enables the remote wakeup detection function if remote wakeup receipt permission is enabled. Issues suspend change completion status notification. (H/W)	T1 + 5 ms {T_{WTRSM}} < T3

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.4.10.3.6.2 If FS Device is Connected

Shown below are the procedures performed when an FS device is connected. The LSI hardware performs steps (2) to (7) automatically.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to “GoSUSPEND” (T0).

(2) Sets the host state monitor (H_NegoControl_0.HostState) to “SUSPEND” (T0).

(3) Disables the disconnect detection function and remote wakeup detection function (T0).

(4) Awaits completion of the transaction currently being executed, then sets the operating mode (H_XcvrControl.OpMode[1:0]) to “PowerDown” (T1).

(5) Enables the disconnect detection function (T3).

(6) Enables the remote wakeup detection function if remote wakeup receipt permission (H_NegoControl_1.RmtWkupDetEnb) is enabled (T3).

(7) Issues suspend change completion status notification (H_SIE_IntStat_1.SuspendCmp) to the firmware (T3).

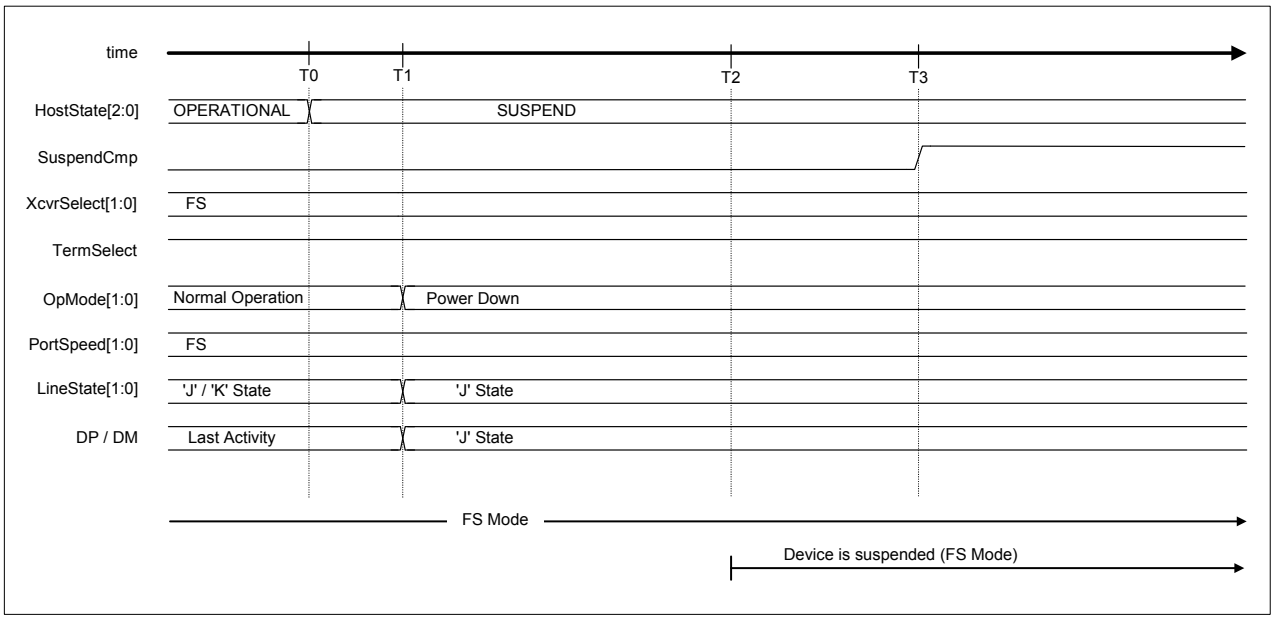


Fig. 1-61 Suspend timing (FS mode)

Table 1-54 Suspend timing values (FS mode)

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to “GoSUSPEND.” (F/W) Disables the disconnect and remote wakeup detection functions. (H/W)	0 (reference)
T1	Sets the operating mode (H_XcvrControl.OpMode[1:0]) to “PowerDown” after final bus activity ends. (H/W)	T1
T2	The device detects Suspend state.	T1 + 3.0 ms < T2 {T_{WTREV}} < T1 + 3.125 ms
T3	Enables the disconnect detection function. Enables the remote wakeup detection function if remote wakeup receipt permission is enabled. Issues suspend change completion status notification. (H/W)	T1 + 5 ms {T_{WTRSM}} < T3

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.4.10.3.6.3 If LS Device is Connected

Shown below are the procedures performed when an LS device is connected. The LSI hardware performs steps (2) to (7) automatically.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to “GoSUSPEND” (T0).

(2) Sets the host state monitor (H_NegoControl_0.HostState) to “SUSPEND” (T0).

(3) Disables the disconnect detection function and remote wakeup detection function (T0).

(4) Awaits completion of the transaction currently being executed, then sets the operating mode (H_XcvrControl.OpMode[1:0]) to “PowerDown” (T1).

(5) Enables the disconnect detection function (T3).

(6) Enables the remote wakeup detection function if remote wakeup receipt permission (H_NegoControl_1.RmtWkupDetEnb) is enabled (T3).

(7) Issues suspend change completion status notification (H_SIE_IntStat_1.SuspendCmp) to the firmware (T3).

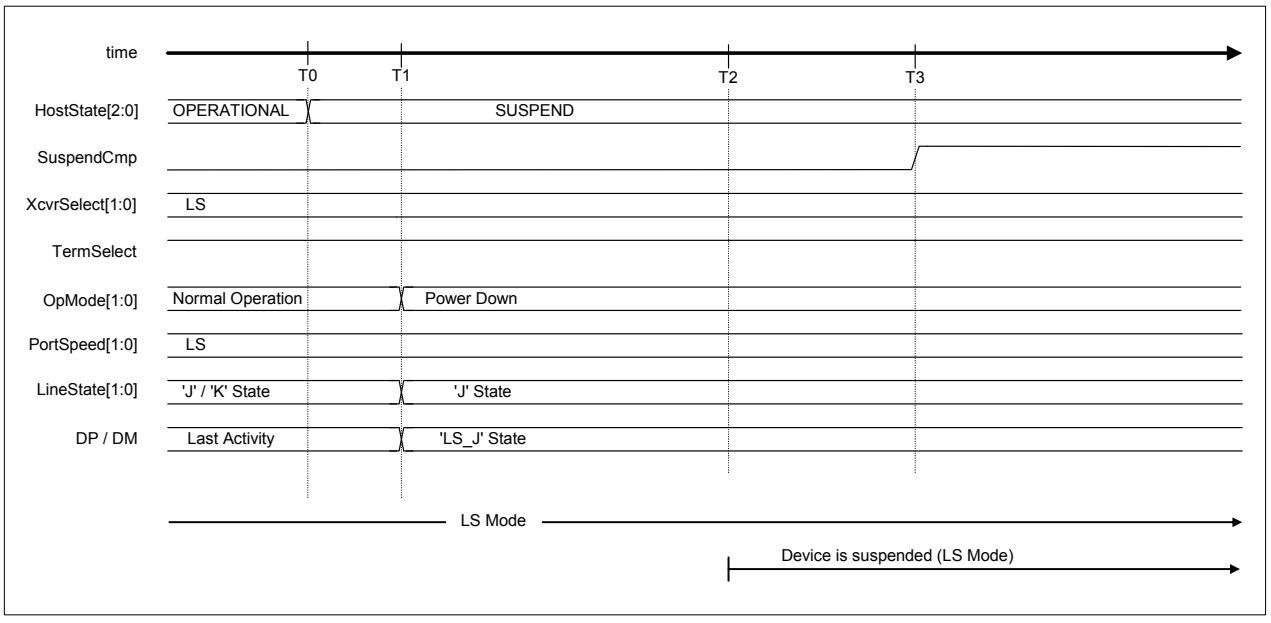


Fig. 1-62 Suspend timing (LS mode)

Table 1-55 Suspend timing values (LS mode)

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to “GoSUSPEND.” (F/W) Disables the disconnect and remote wakeup detection functions. (H/W)	0 (reference)
T1	Sets the operating mode (H_XcvrControl.OpMode[1:0]) to “PowerDown” after final bus activity ends. (H/W)	T1
T2	The device detects Suspend state.	T1 + 3.0 ms < T2 {T_{WTREV}} < T1 + 3.125 ms
T3	Enables the disconnect detection function. Enables the remote wakeup detection function if remote wakeup receipt permission is enabled. Issues suspend change completion status notification. (H/W)	T1 + 5 ms {T_{WTRSM}} < T3

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.4.10.3.7 GoRESUME

The LSI hardware automatically performs the processing necessary to switch to “RESUME” when “GoRESUME” is set to host state change execute (H_NegoControl_0.AutoMode).

1.4.10.3.7.1 If HS Device is Connected

Shown below are the procedures performed when an HS device is connected. The LSI hardware performs steps (2) to (8) automatically.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to “GoRESUME” (T0).

(2) Sets the host state monitor (H_NegoControl_0.HostState) to “RESUME” (T0).

(3) Disables the disconnect detection function and remote wakeup detection function (T0).

(4) Sets the operating mode (H_XcvrControl.OpMode) to “Disable BS and NRZI” and starts issuing the resume “K” signal (T0).

(5) Stops issuing the resume “K” signal (T1).

(6) Sets the terminal selection (H_XcvrControl.TermSelect) to “HS” (T2).

(7) Sets the transceiver selection (H_XcvrControl.XcvrSelect) to “HS” (T3).

(8) Issues resume completion status (H_SIE_IntStat_1.ResumeCmp) to the firmware (T3).

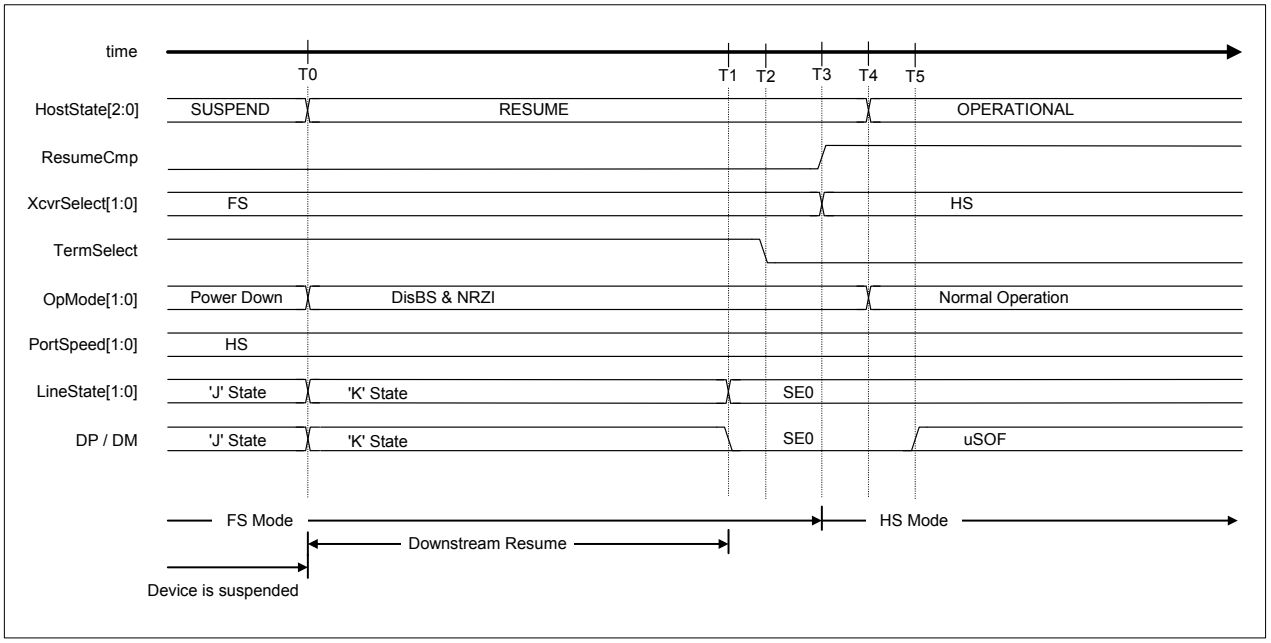


Fig. 1-63 Resume timing (HS mode)

Table 1-56 Resume timing values (HS mode)

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to “GoRESUME.” (F/W) Disables the disconnect and remote wakeup detection functions. Sets the operating mode to “Disable BS and NRZI” and starts issuing the resume “K” signal. (H/W)	0 (reference)
T1	Stops issuing the resume “K” signal. Sets the terminal selection to “HS.” (H/W)	T0 + 20 ms {T_{DRSMDN}} < T1
T2	Sets the transceiver selection to “HS.” (H/W)	T1 + 100 ns < T2 < T1 + 2.0 µs
T3	Issues resume completion status (H_SIE_IntStat_1.ResumeCmp). (H/W)	T1 + 90 µs < T3 < T1 + 110 µs
T4 (reference)	Sets GoOPERATIONAL. (F/W) Sets the operating mode to “NormalOperation.” (H/W)	T4
T5 (reference)	Issues the first micro SOF. (H/W)	T5 < T1 + 3 ms T4 + 120 µs < T5 < T4 + 130 µs

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.4.10.3.7.2 If FS Device is Connected

Shown below are the procedures performed when an FS device is connected. The LSI hardware performs steps (2) to (6) automatically.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to “GoRESUME” (T0).

(2) Sets the host state monitor (H_NegoControl_0.HostState) to “RESUME” (T0).

(3) Disables the disconnect detection function and remote wakeup detection function (T0).

(4) Sets the operating mode (H_XcvrControl.OpMode) to “Disable BS and NRZI” and starts issuing the resume “K” signal (T0).

(5) Stops issuing the resume “K” signal (T1) and finally LS bit time EOP is given.

(6) Issues resume completion status (H_SIE_IntStat_1.ResumeCmp) to the firmware (T2).

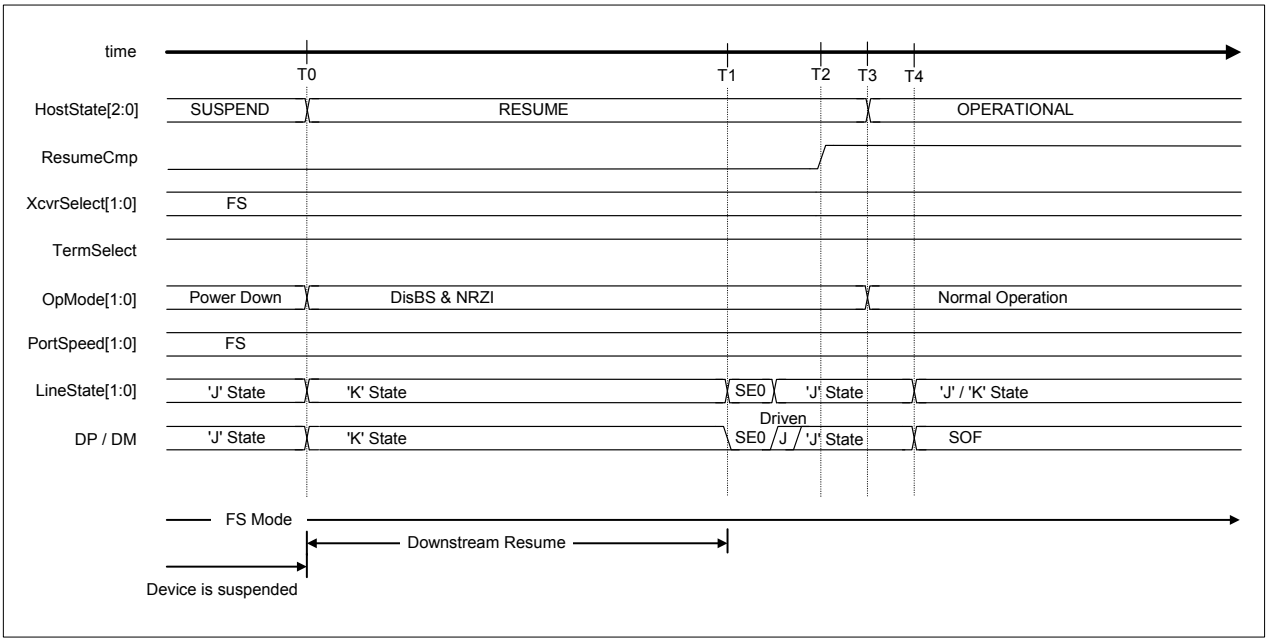


Fig. 1-64 Resume timing (FS mode)

Table 1-57 Resume timing values (FS mode)

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to “GoRESUME.” (F/W) Disables the disconnect and remote wakeup detection functions. Sets the operating mode to “Disable BS and NRZI” and starts issuing the resume “K” signal. (H/W)	0 (reference)
T1	Stops issuing the resume “K” signal and finally the LS bit time EOP is given. (H/W)	T0 + 20 ms {T_{DRSMDN}} < T1
T2	Issues resume completion status (H_SIE_IntStat_1.ResumeCmp). (H/W)	T1 + 90 µs < T2 < T1 + 110 µs
T3 (reference)	Sets GoOPERATIONAL. (F/W) Sets the operating mode to “NormalOperation.” (H/W)	T3
T4 (reference)	Issues the first SOF. (H/W)	T4 < T1 + 3 ms T3 + 0.9 ms < T4 < T3 + 1.1 ms

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.4.10.3.7.3 If LS Device is Connected

Shown below are the procedures performed when an LS device is connected. The LSI hardware performs steps (2) to (6) automatically.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to “GoRESUME” (T0).

(2) Sets the host state monitor (H_NegoControl_0.HostState) to “RESUME” (T0).

(3) Disables the disconnect detection function and remote wakeup detection function (T0).

(4) Sets the operating mode (H_XcvrControl.OpMode) to “Disable BS and NRZI” and starts issuing the resume “K” signal (T0).

(5) Stops issuing the resume “K” signal (T1) and finally LS bit time EOP is given.

(6) Issues resume completion status (H_SIE_IntStat_1.ResumeCmp) to the firmware (T2).

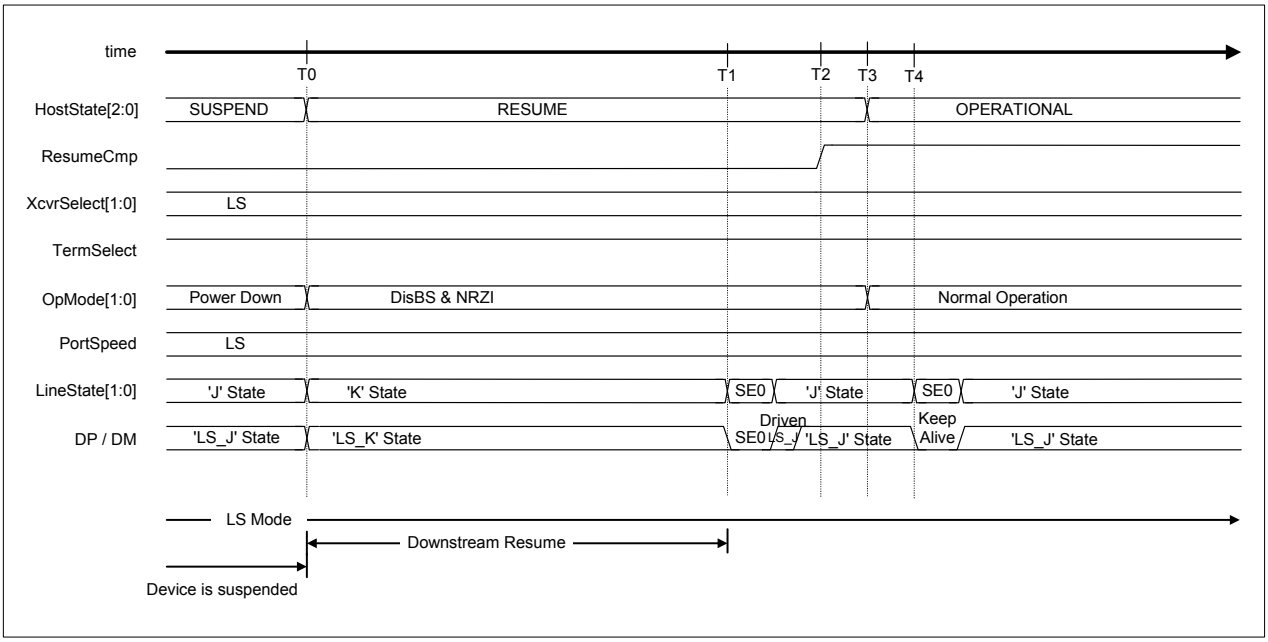


Fig. 1-65 Resume timing (LS mode)

Table 1-58 Resume timing values (LS mode)

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to “GoRESUME.” (F/W) Disables the disconnect and remote wakeup detection functions. Sets the operating mode to “Disable BS and NRZI” and starts issuing the resume “K” signal. (H/W)	0 (reference)
T1	Stops issuing the resume “K” signal and finally the LS bit time EOP is given. (H/W)	T0 + 20 ms {T_{DRSMDN}} < T1
T2	Issues resume completion status (H_SIE_IntStat_1.ResumeCmp). (H/W)	T1 + 90 µs < T2 < T1 + 110 µs
T3 (reference)	Sets GoOPERATIONAL. (F/W) Sets the operating mode to “NormalOperation.” (H/W)	T3
T4 (reference)	Issues the first KeepAlive. (H/W)	T4 < T1 + 3 ms T3 + 0.9 ms < T4 < T3 + 1.1 ms

Note: Brackets {} indicate names defined in the USB 2.0 standards.

1.4.10.3.8 GoWAIT_CONNECTtoDIS

The LSI hardware automatically performs the processing necessary to switch from the "WAIT_CONNECT" state to the "DISABLED" state when "GoWAIT_CONNECTtoDIS" is set to host state change execute (H_NegoControl_0.AutoMode).

The procedures are given below. The LSI hardware performs steps (2) to (5) automatically.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to "GoWAIT_CONNECTtoDIS" (T0).

(2) Performs the same processing performed for "GoWAIT_CONNECT" (T0).

(3) Detects connection and issues connect detection status notification (H_SIE_IntStat_0.DetectCon) (T1).

(4) Performs the same processing performed for "GoDISABLED" (T1).

(5) Issues disabled completion status (H_SIE_IntStat_1.DisabledCmp) (T2).

Note that the timing for each state is the same as when executing "GoWAIT_CONNECT" and "GoDISABLED." For detailed information on timing, refer to "1.4.10.3.2 GoWAIT_CONNECT" and "1.4.10.3.3 GoDISABLED," respectively.

For detailed information on procedures and timing when an error (disconnection or VBUS error) is detected in midcourse, refer to "1.4.10.2.2 Disconnect Detection" and "1.4.10.2.1 VBUS Error Detection."

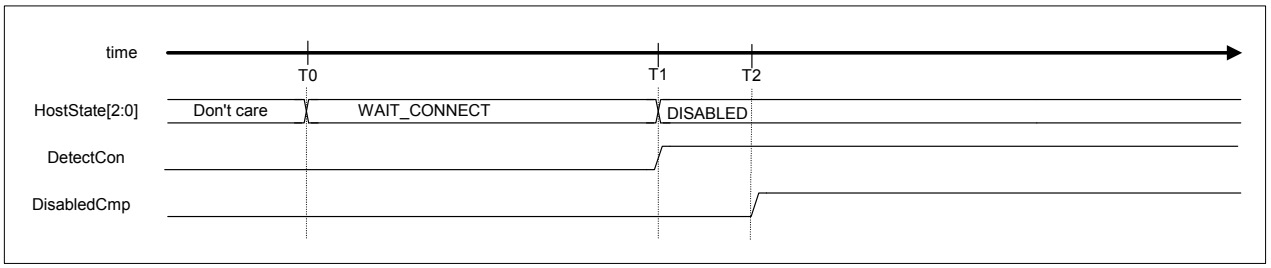


Fig. 1-66 GoWAIT_CONNECTtoDIS timing (HS mode)

Table 1-59 GoWAIT_CONNECTtoDIS timing values (HS mode)

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to "GoWAIT_CONNECTtoDIS." (F/W) Performs the same processing performed for "GoWAIT_CONNECT." (H/W)	0 (reference)
T1	Detects connection and issues connect detection status notification. Performs the same processing performed for "GoDISABLED." (H/W)	T1
T2	Issues disabled completion status. (H/W)	T2

1.4.10.3.9 GoWAIT_CONNECTtoOP

The LSI hardware automatically performs the processing necessary to switch from the "WAIT_CONNECT" state to the "OPERATIONAL" state when "GoWAIT_CONNECTtoOP" is set to host state change execute (H_NegoControl_0.AutoMode).

1.4.10.3.9.1 If HS Device is Connected

Shown below are the procedures performed when an HS device is connected. The LSI hardware performs steps (2) to (9) automatically.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to "GoWAIT_CONNECTtoOP" (T0).

(2) Performs the same processing performed for "GoWAIT_CONNECT" (T0).

(3) Detects connection and issues connection detection status notification (H_SIE_IntStat_0.DetectCon) (T1).

(4) Performs the same processing performed for "GoDISABLED" (T1).

(5) Issues disabled completion status (H_SIE_IntStat_1.DisabledCmp) (T2).

(6) Performs the same processing performed for "GoRESET" (T2).

(7) Detects the device chirp and issues device chirp normal detection status notification (H_SIE_IntStat_0.DetectDevChirpOK) (T3).

(8) Issues reset completion status (H_SIE_IntStat_1.ResetCmp) (T4).

(9) Performs the same processing performed for "GoOPERATIONAL" (T4).

Note that the timing for each state is the same as when executing "GoWAIT_CONNECT," "GoDISABLED," "GoRESET," or "GoOPERATIONAL." For detailed information on timing, refer to "1.4.10.3.2 GoWAIT_CONNECT," "1.4.10.3.3 GoDISABLED," "1.4.10.3.4 GoRESET," and "1.4.10.3.5 GoOPERATIONAL," respectively.

For detailed information on procedures and timing when an error (disconnection, VBUS error, or device chirp error) is detected in midcourse, refer to "1.4.10.2.2 Disconnect Detection" and "1.4.10.2.1 VBUS Error Detection."

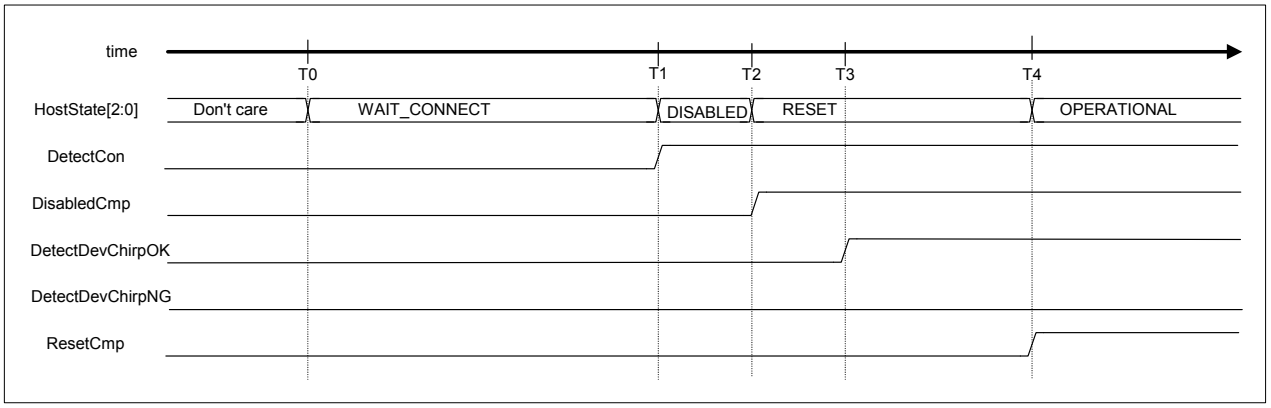


Fig. 1-67 GoWAIT_CONNECTtoOP timing (HS mode)

Table 1-60 GoWAIT_CONNECTtoOP timing values (HS mode)

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to "GoWAIT_CONNECTtoOP." (F/W) Performs the same processing performed for "GoWAIT_CONNECT." (H/W)	0 (reference)
T1	Detects connection and issues connect detection status notification. Performs the same processing performed for "GoDISABLED." (H/W)	T1
T2	Issues disabled completion status. Performs the same processing performed for "GoRESET." (H/W)	T2
T3	Detects the device chirp and issues device chirp normal detection status. (H/W)	T3
T4	Issues reset completion status. Performs the same processing performed for "GoOPERATIONAL." (H/W)	T4

1.4.10.3.9.2 If FS or LS Device is Connected

Shown below are the procedures performed when an FS or LS device is connected. The LSI hardware performs steps (2) to (9) automatically.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to "GoWAIT_CONNECTtoOP" (T0).

(2) Performs the same processing performed for "GoWAIT_CONNECT" (T0).

(3) Detects connection and issues connection detection status notification (H_SIE_IntStat_0.DetectCon) (T1).

(4) Performs the same processing performed for "GoDISABLED" (T1).

(5) Issues disabled completion status (H_SIE_IntStat_1.DisabledCmp) (T2).

(6) Performs the same processing performed for "GoRESET" (T2).

(7) Doesn't detect a device chirp so doesn't issue device chirp normal/error detection status notification (H_SIE_IntStat_0.DetectDevChirpOK/NG) (T3).

(8) Issues reset completion status (H_SIE_IntStat_1.ResetCmp) (T4).

(9) Performs the same processing performed for "GoOPERATIONAL" (T4).

Note that the timing for each state is the same as when executing "GoWAIT_CONNECT," "GoDISABLED," "GoRESET," or "GoOPERATIONAL." For detailed information on timing, refer to "1.4.10.3.2 GoWAIT_CONNECT," "1.4.10.3.3 GoDISABLED," "1.4.10.3.4 GoRESET," and "1.4.10.3.5 GoOPERATIONAL," respectively.

For detailed information on procedures and timing when an error (disconnection or VBUS error) is detected in midcourse, refer to "1.4.10.2.2 Disconnect Detection" and "1.4.10.2.1 VBUS Error Detection," respectively.

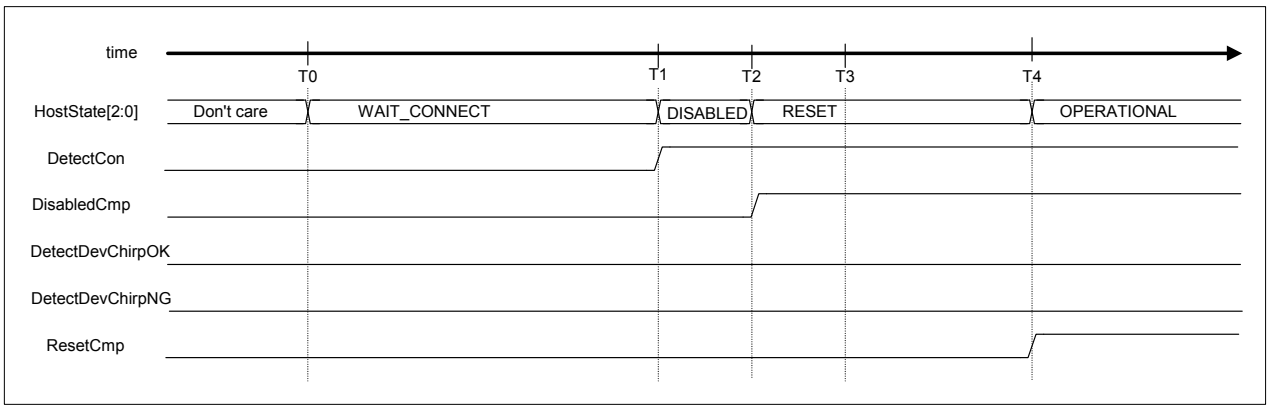


Fig. 1-68 GoWAIT_CONNECTtoOP timing (FS or LS mode)

Table 1-61 GoWAIT_CONNECTtoOP timing values (FS or LS mode)

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to "GoWAIT_CONNECTtoOP." (F/W) Performs the same processing performed for "GoWAIT_CONNECT." (H/W)	0 (reference)
T1	Detects connection and issues connect detection status notification. Performs the same processing performed for "GoDISABLED." (H/W)	T1
T2	Issues disabled completion status. Performs the same processing performed for "GoRESET." (H/W)	T2
T3	No device chirp normal/error detection status notification issued, since no device chirp detected. (H/W)	T3
T4	Issues reset completion status. Performs the same processing performed for "GoOPERATIONAL." (H/W)	T4

1.4.10.3.10 GoRESETtoOP

The LSI hardware automatically performs the processing necessary to switch from the "RESET" state to the "OPERATIONAL" state when "GoRESETtoOP" is set to host state change execute (H_NegoControl_0.AutoMode).

1.4.10.3.10.1 If HS Device is Connected

Shown below are the procedures performed when an HS device is connected. The LSI hardware performs steps (2) to (5) automatically.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to "GoRESETtoOP" (T0).

(2) Performs the same processing performed for "GoRESET" (T0).

(3) Detects the device chirp and issues device chirp normal detection status notification (H_SIE_IntStat_0.DetectDevChirpOK) (T1).

(4) Issues reset completion status (H_SIE_IntStat_1.ResetCmp) (T2).

(5) Performs the same processing performed for "GoOPERATIONAL" (T2).

Note that the timing for each state is the same as when executing "GoRESET" or "GoOPERATIONAL." For detailed information on timing, refer to "1.4.10.3.4 GoRESET" and "1.4.10.3.5 GoOPERATIONAL," respectively.

For detailed information on procedures and timing when an error (VBUS error or device chirp error) is detected in midcourse, refer to "1.4.10.2.2 Disconnect Detection" and "1.4.10.2.1 VBUS Error Detection," respectively.

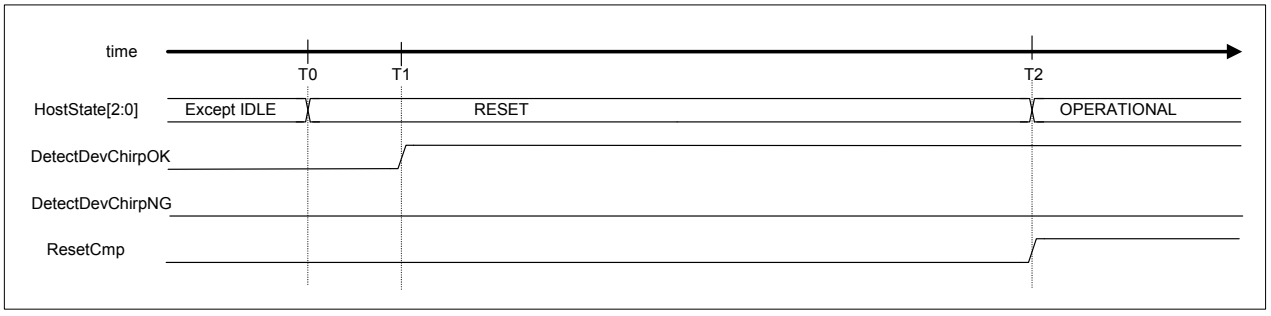


Fig. 1-69 GoRESETtoOP timing (HS mode)

Table 1-62 GoRESETtoOP timing values (HS mode)

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to "GoRESETtoOP." (F/W) Performs the same processing performed for "GoRESET." (H/W)	0 (reference)
T1	Detects the device chirp and issues device chirp normal detection status notification. (H/W)	T1
T2	Issues reset completion status. Performs the same processing performed for "GoOPERATIONAL." (H/W)	T2

1.4.10.3.10.2 If FS or LS Device is Connected

Shown below are the procedures performed when an FS or LS device is connected. The LSI hardware performs steps (2) to (5) automatically.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to "GoRESETtoOP" (T0).

(2) Performs the same processing performed for "GoRESET" (T0).

(3) Doesn't detect a device chirp so doesn't issue device chirp normal/error detection status notification (H_SIE_IntStat_0.DetectDevChirpOK/NG) (T1).

(4) Issues reset completion status (H_SIE_IntStat_1.ResetCmp) (T2).

(5) Performs the same processing performed for "GoOPERATIONAL" (T2).

Note that the timing for each state is the same as when executing "GoRESET" and "GoOPERATIONAL." For detailed information on timing, refer to "1.4.10.3.4 GoRESET" and "1.4.10.3.5 GoOPERATIONAL," respectively.

For detailed information on procedures and timing when an error (VBUS error) is detected in midcourse, refer to "1.4.10.2.1 VBUS Error Detection."

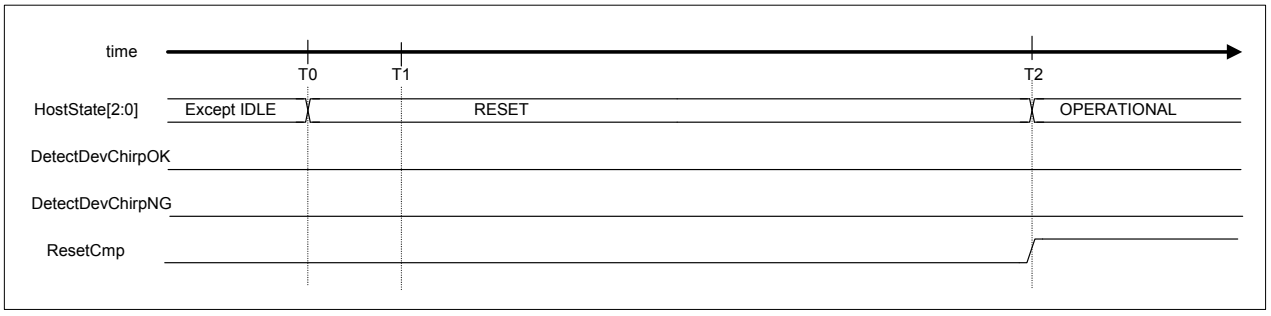


Fig. 1-70 GoRESETtoOP timing (FS or LS mode)

Table 1-63 GoRESETtoOP timing values (FS or LS mode)

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to "GoRESETtoOP." (F/W) Performs the same processing performed for "GoRESET." (H/W)	0 (reference)
T1	No device chirp normal/error detection status notification issued, since no device chirp detected. (H/W)	T1
T2	Issues reset completion status. Performs the same processing performed for "GoOPERATIONAL." (H/W)	T2

1.4.10.3.11 GoSUSPENDtoOP

The LSI hardware automatically performs the processing necessary to switch from the "SUSPEND" state to the "OPERATIONAL" state when "GoSUSPENDtoOP" is set to host state change execute (H_NegoControl_0.AutoMode).

The remote wakeup detection function is automatically enabled or disabled by the hardware when "GoSUSPENDtoOP" is set (although this is not shown by remote wakeup receipt permission (H_NegoControl_1.RmtWkupDetEnb)). There is no need for the firmware to manipulate the remote wakeup receipt permission (H_NegoControl_1.RmtWkupDetEnb).

Avoid using the power management function with this setting.

Shown below are the procedures for this setting. The LSI hardware performs steps (2) to (7) automatically.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to "GoSUSPENDtoOP" (T0).

(2) Performs the same processing performed for "GoSUSPEND" (T0).

(3) Issues suspend change completion status notification (H_SIE_IntStat_1.SuspendCmp) (T1).

(4) Detects the remote wakeup and issues remote wakeup detection status notification (H_SIE_IntStat_0.DetectRmtWkup) (T2).

(5) Performs the same processing performed for "GoRESUME" (T2).

(6) Issues resume completion status (H_SIE_IntStat_1.ResumeCmp) (T3).

(7) Performs the same processing performed for "GoOPERATIONAL" (T3).

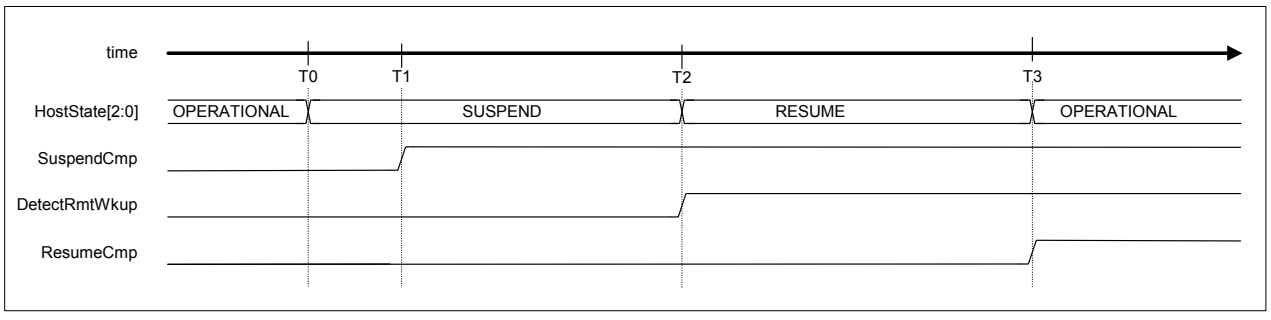


Fig. 1-71 GoSUSPENDtoOP timing

Table 1-64 GoSUSPENDtoOP timing values

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to “GoSUSPENDtoOP.” (F/W) Performs the same processing performed for “GoSUSPEND.” (H/W)	0 (reference)
T1	Issues suspend change completion status notification. (H/W)	T1
T2	Detects the remote wakeup and issues remote wakeup detection status notification. Performs the same processing performed for “GoRESUME.” (H/W)	T2
T3	Issues resume completion status. Performs the same processing performed for “GoOPERATIONAL.” (H/W)	T3

1.4.10.3.12 GoRESUMEtoOP

The LSI hardware automatically performs the processing necessary to switch from the "RESUME" state to the "OPERATIONAL" state when "GoRESUMEtoOP" is set to host state change execute (H_NegoControl_0.AutoMode).

Shown below are the procedures for this setting. The LSI hardware performs steps (2) to (4) automatically.

(1) The firmware sets the host state change execute (H_NegoControl_0.AutoMode) to "GoRESUMEtoOP" (T0).

(2) Performs the same processing performed for "GoRESUME" (T0).

(3) Issues resume completion status (H_SIE_IntStat_1.ResumeCmp) (T1).

(4) Performs the same processing performed for "GoOPERATIONAL" (T1).

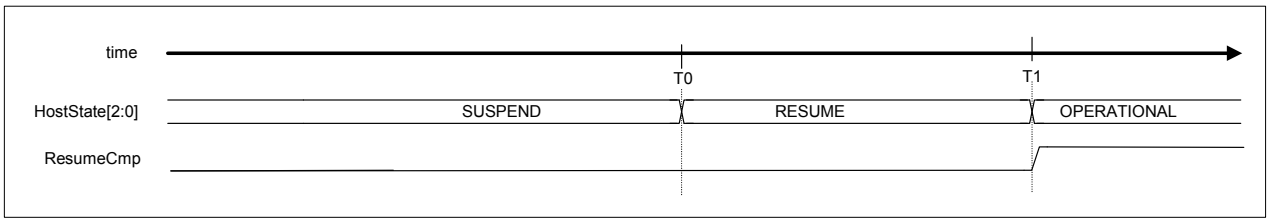


Fig. 1-72 GoRESUMEtoOP

Table 1-65 GoRESUMEtoOP timing values

Timing Parameter	Description	Value
T0	Sets H_NegoControl_0.AutoMode to "GoRESUMEtoOP." (F/W) Performs the same processing performed for "GoRESUME." (H/W)	0 (reference)
T1	Issues resume completion status. Performs the same processing performed for "GoOPERATIONAL." (H/W)	T1

1.5 Power Management Functions

This IC includes independent power management functions for USB Port 0 and Port 1.

The power management functions switch the ports to and from the SLEEP, SNOOZE, and ACTIVE states. Note that SNOOZE state is simply a transient state used between SLEEP and ACTIVE states.

Changing to other states is initiated by setting the PM_Control.GoSLEEP and PM_Control.GoActive bits. Inspect PM_Control.PM_State[3:0] to check the current state. The MainIntStat.FinishedPM event occurs when a change is complete. An interrupt XINT occurs here if the MainIntEnb.EnFinishedPM bit is set.

The individual port power management states are controlled independently of other port states. The LSI switches to SLEEP state, and the MainIntStat.FinishedPM event occurs once the transition is complete if the PM_Control.GoSLEEP bit is set while in ACTIVE state. Similarly, the LSI switches to ACTIVE state and the MainIntStat.FinishedPM event occurs once the transition is complete if the PM_Control.GoActive bit is set while in SLEEP state.

A SNOOZE state exists for hardware between the SLEEP and ACTIVE states, but is not necessary for the recognition by software.

If one port is in the ACTIVE state and the other port is in the SLEEP state, PM_Control.PM_State[3:0] for the port in SLEEP state will indicate “0b01” rather than “0b00.” This indicates that the oscillator is functioning, but should be interpreted as indicating SLEEP state in the same way as for “0b00.”

Note that it is not possible to set a change from the current state to the same state. The PM_Control.GoXXXX bit set will not be cleared in this case. If, for example, PM_Control.GoACTIVE is set while in ACTIVE state, the PM_Control.GoACTIVE bit will remain set without changing states (even though it appears correct). This means operations will no longer be correctly performed.

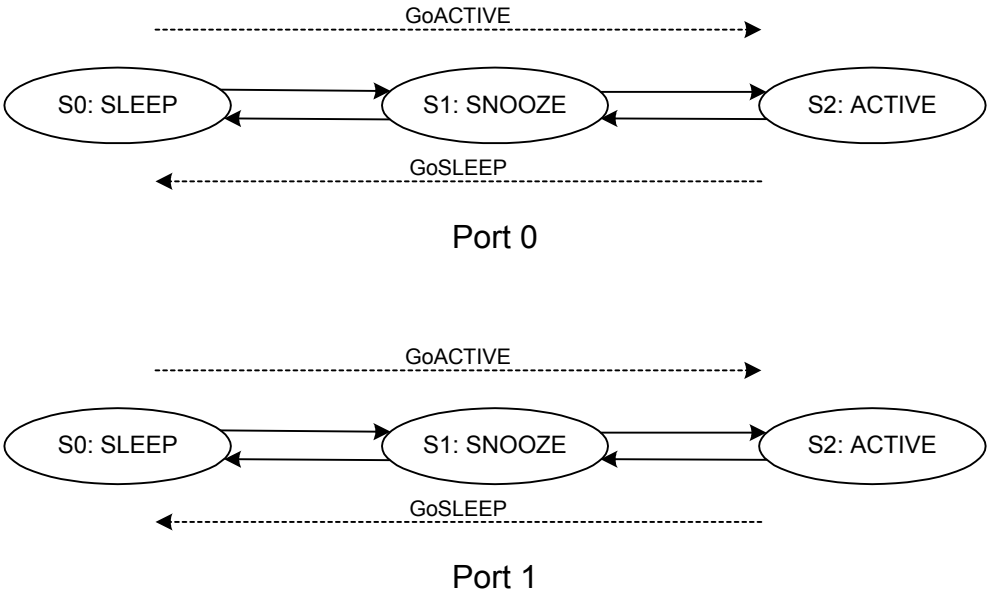


Fig. 1-73 Power management state transitions

1.5.1 SLEEP

Neither the oscillator nor the PLL oscillate in this state. The registers and bits shown in ***bold italic*** in the register map allow reading and writing in SLEEP state.

Setting the PM_Control.GoSLEEP bit while in ACTIVE state changes to SLEEP state via the SNOOZE state. The PLL480 operating here is stopped, followed by the PLL60. The OSCCLK output is then stopped before oscillation stops.

1.5.2 ACTIVE

The oscillator, PLL60, and PLL480 all operate in this state. All registers and bits in the register map allow reading and writing in the ACTIVE state.

Setting the PM_Control.GoACTIVE bit while in SLEEP state changes to ACTIVE state via the SNOOZE state. When switching from SLEEP to SNOOZE states, the internal circuit includes an oscillation stabilization time gate to avoid applying OSCCLK until the oscillator is oscillating stably. This oscillation stabilization time varies with the oscillator cell, oscillator, peripheral circuits, and circuit board and should be set using the WakeUpTim_H,L registers. The WakeUpTim_H,L registers allow asynchronous access and can be read from or written to even in the SLEEP state.

The USB host circuit operates in this state, since it requires a 480 MHz SCLK480.

1.6 FIFO Management

This section describes FIFO management.

1.6.1 FIFO Memory Map

The FIFO memory map is shown below. Ports 0 and 1 have separate FIFO areas.

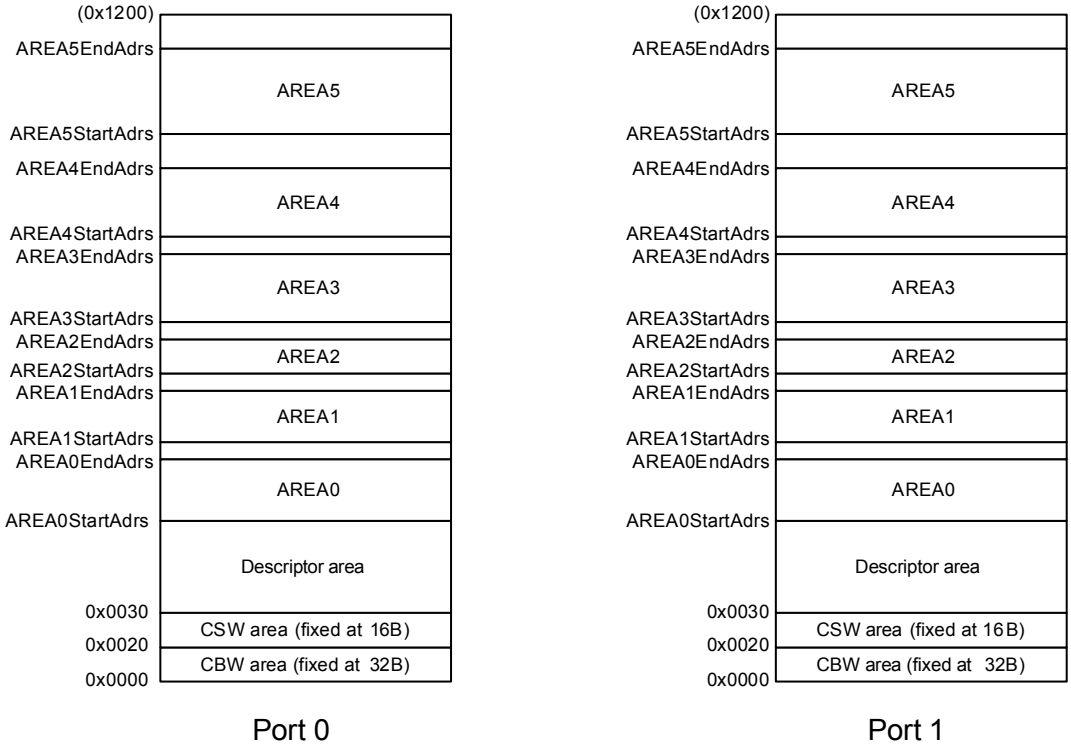


Fig. 1-74 Typical FIFO memory map

FIFO memory can be used after partitioning into CBW, CSW, and descriptor areas, and AREA0, AREA1, AREA2, AREA3, AREA4, and AREA5. The CBW and CSW areas are assigned fixed areas, as shown in Fig. 1-74. The other areas (AREAx{x=0-5}) can be set flexibly using the FIFO area setting registers (AREAx{x=0-5}StartAdrs_H,L and AREAx{x=0-5}EndAdrs_H,L). The descriptor areas can use unused areas as required.

The descriptor area is an area used by the descriptor reply function for USB devices. Any unused FIFO area can be used. For detailed information on actual usage procedures, refer to "1.6.2 Descriptor Area." All FIFO areas can be set for use with the descriptor reply function, but it is recommended to use the areas shown here to prevent interference.

The CBW area is used for bulk-only support function CBW support for USB devices. 32 bytes are reserved, but the 31-byte area starting from address 0x0000 is used. For detailed information on actual usage procedures, refer to "1.6.3.1 CBW Area (for USB device)." This CBW area is also used for the CHa bulk-only support function with USB hosts. For detailed information on actual usage procedures, refer to "1.6.3.2 CBW Area (for USB host)."

The CSW area is used for bulk-only support function CSW support for USB devices. 16 bytes are reserved, but the 13-byte area starting from address 0x0020 is used. For detailed information on actual usage procedures, refer to "1.6.4.1 CSW Area (for USB device)." This CSW area is also used for the CHa bulk-only support function with USB hosts. For detailed information on actual usage procedures, refer to "1.6.4.2 CSW Area (for USB host)."

AREA0, AREA1, AREA2, AREA3, AREA4, and AREA5 are the endpoint areas which can be used by joining to endpoints (EPx{x=0, a-e}) for USB devices. For USB hosts, they can be used by joining to channels (CHx{x=0, a-e}) in the same way. The JoinEPxCHx{x=0,a-e} bit in the AREA join setting register (AREAx{x=0-5}Join_1) is set to join areas. Note that the same endpoint or channel must not be joined to multiple areas.

The AREA0, AREA1, AREA2, AREA3, AREA4, and AREA5 areas are each controlled as FIFO and contain data storage quantities. To clear this stored state, set the AREAnFIFO_Clr.ClearAREAx{x=0-5} bits.

Note that clearing these states only initializes data storage information and does not write or clear data. These bits do not actually clear data on the RAM, and the data stored within the descriptor areas is not removed. There is no need to rewrite data after clearing.

1.6.2 Descriptor Area

The descriptor area is the area used by the descriptor reply function for USB devices. The descriptor reply function can be used for endpoint 0 when the data stage is transferred in the IN direction.

The IN-direction data stage is performed automatically by setting the initial address of the data to be written to the area and the data size to be returned, then executing the descriptor reply function.

Commands can be specified to return the area data on receipt of a request by writing details such as the device descriptor uniquely specified by the device to the area at initialization after turning on power. This allows fast response without writing data to the EP0 FIFO area for each request.

1.6.2.1 Writing Data to Descriptor Area

The RAM_WrDoor function is used to write data to the descriptor area. The initial write address is set to the RAM_WrAdrs_H,L registers, allowing the data to be written to the RAM_WrDoor_0,1 registers. The RAM_WrAdrs_H,L register values are updated by the quantity of data written for each write cycle. This enables continuous writing to the RAM_WrDoor_0,1 registers when writing data to continuous addresses.

Note that the RAM_WrDoor_0,1 registers allow writing only.

1.6.2.2 Executing Data Stage (IN) with Descriptor Area

When using the written data with the descriptor reply function, set the first address of the data to be sent to the data stage in the D_DescAdrs_H,L registers, the data size to be returned in the D_DescSize_H,L registers, then the D_EP0Control.ReplyDescriptor bit to 1. Set the D_EP0Control.INxOUT bit to "1" to allow IN transactions. Be careful to clear the D_EP0Control_IN.ForceNAK bit after clearing D_SETUP_Control.ProtectEP0 to allow data packets to be returned to data stage IN transactions.

After setting, data packets up to the data quantity set by the D_DescSize_H,L registers are returned to the host in response to the IN transaction from the host after automatically dividing them into the max packet size (set by D_EP0MaxSize). Data is automatically sent as short packets if the D_DescSize_H,L register values are smaller than the max packet size or if the data quantity remaining after the division is smaller than the max packet size.

D_EP0Control.ReplyDescriptor is cleared and D_EP0IntStat.DescriptorCmp is set when an OUT transaction is issued by the host. The firmware should switch to status stage processing.

1.6.3 CBW Area

1.6.3.1 CBW Area (for USB device)

The CBW area is the area used by the bulk-only support function CBW support for USB devices. This area can be used to receive data when performing BulkOnly Transport Protocol command transport for Bulk OUT endpoints (endpoints EPa, EPb, EPc, EPd, EPe). This enables only the data received using data transport to be handled by the endpoint FIFO.

With CBW support, data is received at the CBW area if the OUT transaction is performed for the endpoint involved and the data size is 31 bytes. The error status is issued and the data is discarded if the data is not 31 bytes long.

The RAM_Rd function is used to read out data received at the CBW area. Setting the RAM_RdControl.RAM_GoRdCBW_CSW bit reads the CBW area data and copies it to the RAM_Rd_00 to RAM_Rd_1E registers before issuing completion status notification (CPU_IntStat.RAM_RdCmp bit).

1.6.3.2 CBW Area (for USB host)

The CBW area is the area used for the bulk-only support function for USB hosts. CBW data is sent as data packets from these areas when using the BulkOnly Transport Protocol command transport with channel CHa. This enables only the data sent using data transport to be handled by the channel FIFO.

CBW data (31 bytes) should be provided in the CBW area from 0x0000 before sending data packets.

The RAM_WrDoor function is used to write data to the CBW area. The CBW area initial address (0x0000) is written to the RAM_WrAdrs_H,L registers, then 31 bytes of valid data are written via the RAM_WrDoor_0,1 registers. The CBW area holds 32 bytes; there should be no issue with leakage into other areas, even when 32 bytes are written in word access.

1.6.4 CSW Area

1.6.4.1 CSW Area (for USB device)

The CSW area is the area used by the bulk-only support function CSW support for USB devices. This area can be used to send data when performing BulkOnly Transport Protocol status transport for Bulk IN endpoints (endpoints EPa, EPb, EPc, EPd, EPe). This enables only the data sent using data transport to be handled by the endpoint FIFO.

When using CSW support, data is sent from the CSW area if the IN transaction is performed for the endpoint involved and the data size is 13 bytes.

The RAM_WrDoor function is used to write data to the CSW area. The CSW area initial address (0x0020) is written to the RAM_WrAdrs_H,L registers, then 13 bytes of valid data are written via the RAM_WrDoor_0,1 registers. The CSW area holds 16 bytes; there should be no issue with leakage into other areas, even when 14 bytes are written in word access.

1.6.4.2 CSW Area (for USB host)

The CSW area is the area used for the bulk-only support function for USB hosts. CSW data is received in this area when using the BulkOnly Transport Protocol status transport with channel CHa. This enables only the data received using data transport to be handled by the channel FIFO.

The RAM_Rd function is used to read out data received at the CSW area. Setting the RAM_RdControl.RAM_GoRdCBW_CSW bit reads the CSW area data and copies it to the RAM_Rd_00 to RAM_Rd_0C registers before issuing completion status notification (CPU_IntStat.RAM_RdCmp bit).

1.6.5 FIFO Access Methods

FIFO data can be accessed using RAM access or FIFO access. FIFO access parameters include CPU (register), CPU (DMA), and USB.

RAM access allows data to be read from or written to any FIFO area. RAM access does not change the FIFO data quantity. In other words, FIFO areas are not freed, even if FIFO data is read out using the RAM_Rd function. Similarly, FIFO areas are not occupied even if data is written to the FIFO using the RAM_WrDoor function.

1.6.5.1 RAM Access (RAM_Rd)

To read from FIFO using the RAM_Rd register, set the FIFO area initial address and data size to be read in the RAM_RdAdrs_H,L and RAM_RdCount registers before setting the RAM_RdControl.RAM_GoRd bit. Set the CPU_IntStat.RAM_RdCmp bit to “1” once the specified FIFO area data can be read from the RAM_Rd register. Read out the data from registers RAM_Rd_00 to RAM_Rd_1F after checking the RAM_RdCmp bit. The data read out is stored in sequence from RAM_Rd_00. The RAM_Rd register values will be invalid beyond the preset size if the size set in the RAM_RdCount register is smaller than 32.

FIFO data can be read out from the RAM_Rd register as required, regardless of FIFO area settings.

The RAM_RdAdrs_H,L and RAM_RdCount register values are updated in sequence while the RAM_Rd function runs. These registers should not be accessed after the RAM_Rd function starts until the CPU_IntStat.RAM_RdCmp bit has been set. Values read out from these registers are not guaranteed while the RAM_Rd function runs. Writing to these registers may result in malfunctions.

1.6.5.2 RAM Access (RAM_WrDoor)

To write to the FIFO using the RAM_WrDoor_0,1 registers, set the initial write address in the RAM_WrAdrs_H,L registers and write the data using the RAM_WrDoor_0,1 register. The RAM_WrAdrs_H,L registers are incremented automatically by the data quantity written for each access, allowing continuous writing to the RAM_WrDoor_0,1 registers when writing data to continuous addresses.

FIFO data can be written to the FIFO using the RAM_WrDoor_0,1 registers, regardless of FIFO area settings.

1.6.5.3 FIFO Access (Register Access)

To read data from FIFO by register access, set AREAx{x=0-5}Join_0.JoinCPU_Rd to 1 for one area, then read using the FIFO_Rd_0,1 or FIFO_ByteRd registers.

To write data to the FIFO by register access, set AREAx{x=0-5}Join_0.JoinCPU_Wr to 1 for one area, then write using the FIFO_Wr_0,1 or FIFO_ByteWr registers.

The FIFO_RdRemain_H,L registers indicate the remaining data quantity that can be read from FIFO for a single area set using JoinCPU_Rd. Similarly, the FIFO_WrRemain_H,L registers indicate the remaining area for writing to the FIFO for a single area set using JoinCPU_Wr.

Note that data will be read from FIFO during register dumping if either JoinCPU_Rd bit is set for register dumping when debugging the firmware using ICE.

1.6.5.4 FIFO Access (DMA)

To read from FIFO using CPU DMA access, the AREAx{x=0-5}Join_0.JoinDMA bit is set for one area, then the DMA_Control.Dir bit is set to 1, and the data is read out by DMA procedures.

To write to the FIFO using CPU DMA access, the AREAx{x=0-5}Join_0.JoinDMA bit is set for one area for each DMA channel, then the DMA_Control.Dir bit is set to 0, and the data is written by DMA procedures.

The DMA_Remain_H,L registers indicate the remaining data size that can be read from FIFO for the single area selected by the AREAx{x=0-5}Join_0.JoinDMA bit for each DMA channel. It also indicates the remaining space available for writing in the FIFO for the single area selected by the AREAx{x=0-5}Join_0.JoinDMA bit for each DMA channel.

1.6.5.5 FIFO Access Restrictions

The FIFO in this LSI allows simultaneous transfer to and from the USB, register reading and writing from the CPU bus, and reading and writing using DMA. Pre-reading is also used when reading from the CPU bus.

The following exclusive rules exist for accessing (joining) the FIFO in the respective areas:

- Multiple write parameters must not be set simultaneously for the same FIFO area.
- Multiple read parameters must not be set simultaneously for the same FIFO area.
- Only one of JoinCPU_Wr, JoinCPU_Rd, or JoinDMA can be set to a single area.
- JoinCPU_Wr, JoinCPU_Rd, and JoinDMA can be set to only one area at any given time.

The following exceptions are permissible for access parameters from USB (for devices) with duplicated read/write parameters for the same area. For example, JoinCPU_Wr and JoinDMA(DMA_Control.Dir=0) can be set for writing to the OUT endpoint FIFO area. In this case, JoinCPU_Wr or JoinDMA must be set for writing from the CPU after ensuring that OUT transactions are not being performed. Similarly, JoinCPU_Rd and JoinDMA(DMA_Control.Dir=1) can be set for reading from the IN endpoint FIFO area. In this case, JoinCPU_Rd or JoinDMA must be set for reading from the CPU after ensuring that IN transactions are not being performed.

Conditions under which transactions are not performed include the following: ActiveUSB bit is cleared when each endpoint is not joined to FIFO areas and when ForceNAK is not set.

There are no restrictions that apply to multiple ports, since individual FIFO areas are independent.

1.7 CPUIF

1.7.1 CPUIF Mode

The LSI CPUIF supports the two BUS modes, BE mode and Strobe mode. The specific write access method depends on the BUS mode. BE mode allows writes using the write strobe signal indicating the write timing and the higher (CD[15:8]) and lower (CD[7:0]) byte enable signal. Strobe mode allows writes using the individual write strobe signals for the higher and lower bytes (refer to Fig. 1-75). The read access methods are the same for either BUS mode. Both use the read strobe signal.

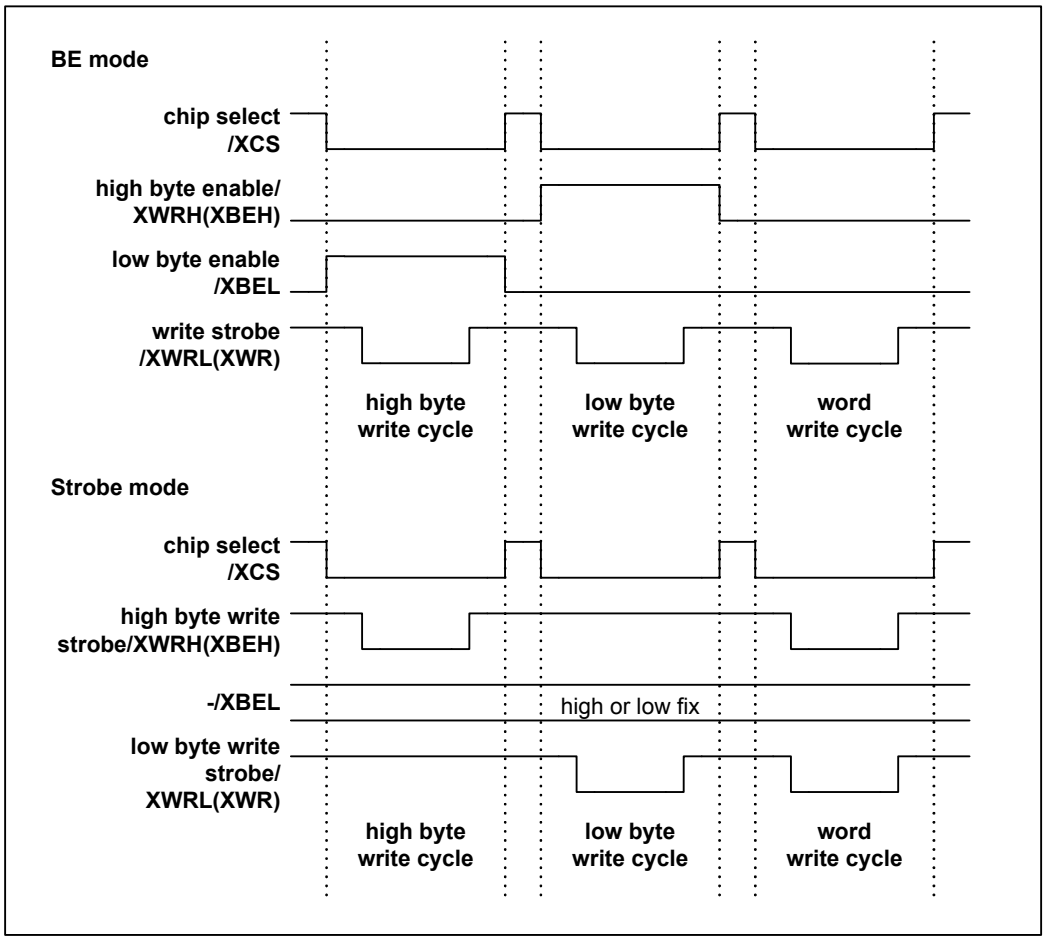


Fig. 1-75 BUS modes

In addition to the BUS modes described above, the LSI CPUIF also supports the two big-endian and little-endian ENDIAN modes to provide CPU endian support. Big-Endian mode is a mode in which the even address register takes the higher position, the odd address register takes the lower position, the first byte data takes the higher position, and the second byte data takes the lower position. Conversely, little-endian mode is a mode in which the even address register takes the lower position, the odd address register takes the higher position, the first byte data takes the lower position, and the second byte data takes the higher position. For detailed information on ENDIAN modes, refer to "Appendix A: Connection to Little-endian CPU."

The LSI should be set to the correct BUS and ENDIAN modes for the CPU before use.

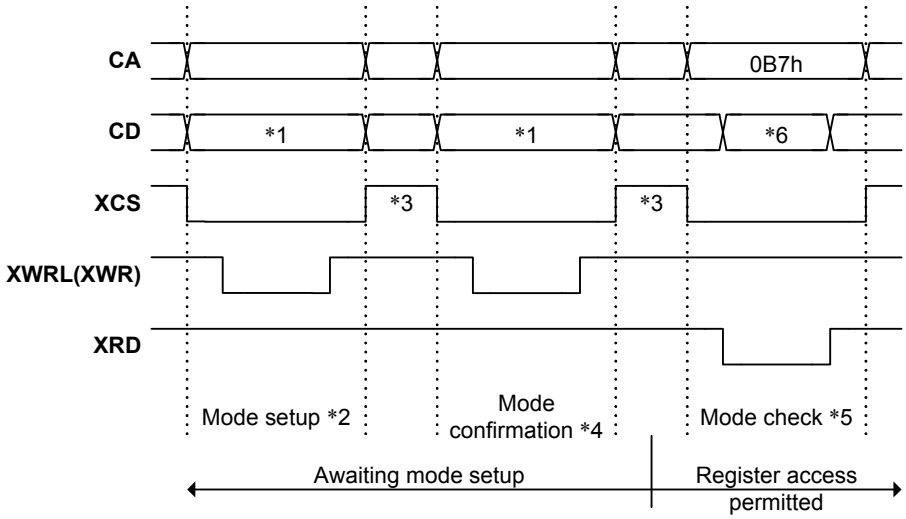
The following sections describe the mode setup procedures.

1.7.2 CPUIF Mode Setup

The LSI awaits CPUIF mode setup (uninitialized period) after hardware resetting. No access is permitted in this state except for mode setup. Register access is permitted once mode setup is complete (initialization period). The mode setup procedures are shown below.

1) Asserts the XCS and XWRL signals and writes mode setup data according to the specified data pattern (mode setup). For example, data is written using words to any address in the memory space mapping the LSI.
2) The XCS signal is temporarily negated. For example, NOP is inserted or memory space is accessed other than that in which the LSI is mapped.
3) The XCS signal is asserted again (mode confirmation). For example, the same operation is performed as in step 1).
4) The XCS signal is temporarily negated. For example, the same operation is performed as in step 2).

Normal register access is possible once this mode setup is complete. Ideally, read the ChipConfig register after setting to check mode setup.



*1:

ENDIAN mode	BUS mode	CD[15:0]
Big endian	Strobe	****_*00*_****_*00*
Big endian	BE	****_*01*_****_*01*
Little endian	Strobe	****_*10*_****_*10*
Little endian	BE	****_*11*_****_*11*

*2: This must be set.

*3: Temporarily negate the chip select signal ("1").

*4: Assert the chip select signal (e.g., same writing as for mode setup).

*5: Checking is recommended (register access is possible even if not checked).

*6: The mode set can be checked.

Fig. 1-76 CPUIF mode setup

Signals other than those included in Fig. 1-76 in mode setup can be either High or Low, provided the AC ratings are satisfied.

The CPUIF mode must always be set after a hard reset, and only while uninitialized period.

The following CPUIF descriptions primarily explain Strobe and Big-endian modes.

1.7.3 Block Configuration

The LSI CPUIF consists of four blocks: REG0, REG1, DMA0, and DMA1.

- REG0: Controls access to the register area
- REG1: Controls access to the register area
- DMA0: Port 0 DMA channel
- DMA1: Port 1 DMA channel

1.7.3.1 REG (Registers)

These control access to the LSI register areas. They include the following functions:

- Synchronous register access
- FIFO access
- RAM_Rd access
- Asynchronous register access

1.7.3.1.1 Synchronous Register Access (Write)

Writes data from an external bus to the register synchronized with the internal clock.

1.7.3.1.2 Synchronous Register Access (Read)

Outputs register data to an external bus with the read (asserted for both XCS and XRD) time as the output enable period.

Significant registers with at least three bytes such as count values in the register read operation include those that maintain the lower byte register value for the highest byte read timing and output the value to an external bus when reading the lower byte to prevent an incorrect count from being read – for example, due to carry-over of count values during the access cycle.

1.7.3.1.3 FIFO Access (Write)

FIFO write access refers to writing to the FIFO_Wr_0,1, FIFO_ByteWr, and RAM_WrDoor_0,1 registers.

FIFO Access (Write) includes the following restrictions:

- Check the amount of data that can be written using the FIFO_WrRemain_H,L registers before accessing after setting the AREAx{x=0-5}Join_0.JoinCPU_Wr bit. The RAM_WrDoor_0,1 registers are not subject to this restriction.
- The FIFO should normally be accessed in word (2-byte) units. Control the strobe signal, taking into account the FIFO byte boundary when writing fractional (odd) bytes. Otherwise, use the FIFO_ByteWr register. For details, refer to "1.7.3.1.5 FIFO Access Fractional Number Processing."
- The FIFO free space cannot be checked accurately if the FIFO_WrRemain_H,L registers are checked immediately after writing to the FIFO_Wr_0,1 or FIFO_ByteWr registers. Always leave an interval of at least one CPU cycle before checking.
- The address cannot be checked accurately if the RAM_WrDoorAdrs_H,L registers are checked immediately after writing to the RAM_WrDoor_0,1 registers. Always leave an interval of at least one CPU cycle before checking.

1.7.3.1.4 FIFO Access (Read)

FIFO read access refers to reading from the FIFO_Rd_0,1 and FIFO_ByteRd registers.

FIFO Access (Read) includes the following restrictions:

- Check the amount of data that can be read using the FIFO_RdRemain_H,L registers and the RdRemainValid bit before accessing after setting the AREAx{x=0-5}Join_0.JoinCPU_Rd bit.
- Use the FIFO_Rd_0,1 registers for word-reading. For byte-reading, use the FIFO_ByteRd register. Use byte-reading if a byte boundary exists. If the FIFO_Rd_0,1 registers are used for word-reading in this case, valid data will only be output on one side. For details, refer to "1.7.3.1.5 FIFO Access Fractional Number Processing."

1.7.3.1.5 FIFO Access Fractional Number Processing

This section describes the correlation between FIFO access and data storage status in the FIFO when handling fractional (odd) data quantities. The actual FIFO has a 4-byte width, but a 2-byte width is used here to simplify the explanation. The operations are the same for either 2-byte or 4-byte.

[Writing]

Writes should normally be performed from a state without byte boundaries.

When setting the AREAnFIFO_Clr.ClrAREAx{x=0-5} bit and word writing from a state without byte boundaries and with odd data quantities, write the last byte (data Z) of the continuous data only to the High side. This situation is shown in Fig. 1-77 (1). Data is output from the USB in the sequence A, B, C, D .. X, Y, Z.

When writing from a state in which byte boundaries exist in the FIFO, first write the data to the Low side (write data K), then clear the byte boundaries before word writing (data L and M). This situation is shown in Fig. 1-77 (2).

These are the normal write operations.

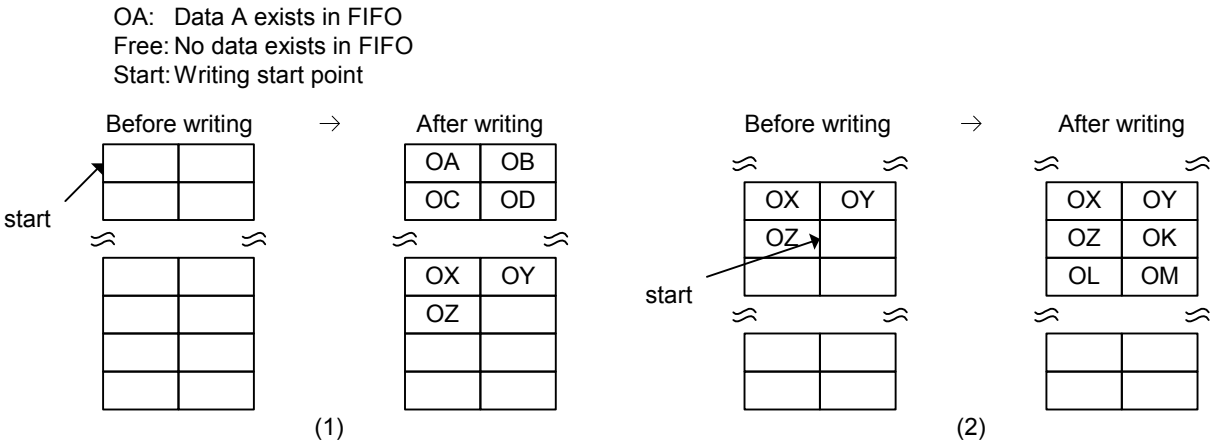


Fig. 1-77 FIFO writing processing (normal operation)

The writing operations below require caution.

If word writing is performed from a state in which byte boundaries exist in the FIFO, writing to High will be ignored. Only writing to Low will be performed (Fig. 1-78 (3)). In other words, the operation is identical to that performed when writing to Low. If writing to High only from a state in which byte boundaries exist in the FIFO, writing will be ignored (Fig. 1-78 (4)).

For cases in which writing occurs only to Low from a state in which byte boundaries do not exist in the FIFO, this writing will be ignored (Fig. 1-78 (5)). If word writing from a state in which byte boundaries do not exist in the FIFO and the quantity that can be written is "1," writing to Low will be ignored and only writing to High will be performed (Fig. 1-78 (6)). In other words, the operation is identical to that performed when writing to High.

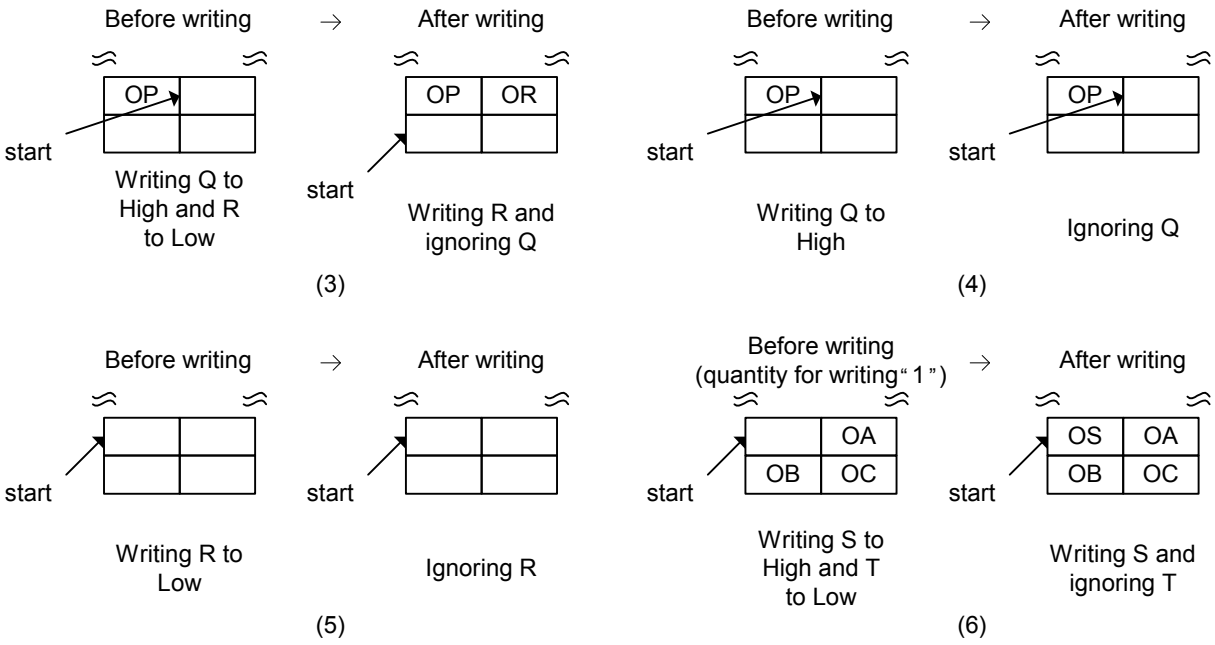


Fig. 1-78 FIFO writing processing (caution required)

[Reading]

If there are no byte boundaries, there should be no problem with word-reading using the FIFO_Rd_0,1 registers or byte-reading using the FIFO_ByteRd register. If there are byte boundaries, read using the FIFO_ByteRd register. There should be no problem with word-reading or byte-reading after the byte boundaries have been temporarily cleared.

Fig. 1-79 (1) shows the situation when word-reading from a state without byte boundaries. Data A and B followed by data C and D are read out for each access. Fig. 1-79 (2) shows the situation when byte-reading. For each access, data is read out in the sequence A, B, C, and D. These are the normal read operations.

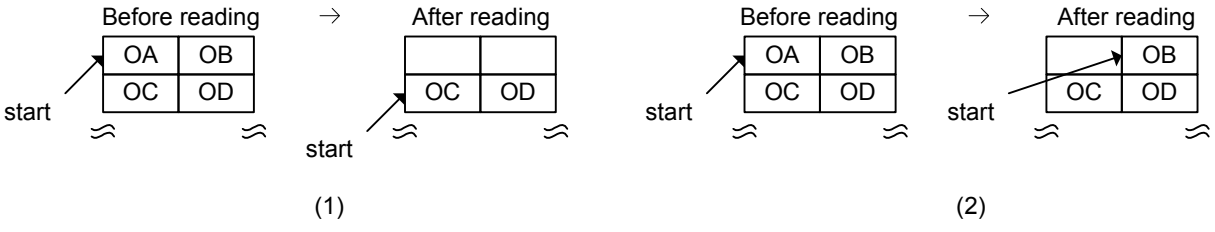


Fig. 1-79 FIFO reading processing (normal operation)

The reading operations below require caution.

Fig. 1-80 (3) shows the situation when word-reading using the FIFO_Rd_0,1 registers from a state having byte boundaries. Indeterminate data is output to High and data J is output to Low. The read pointer increments one byte at a time. Fig. 1-80 (4) shows the operation when word-reading using the FIFO_Rd_0,1 registers from a state in which byte boundaries do not exist but one byte of data remains. Data X is output to High and indeterminate data is output to Low. The read pointer increments one byte at a time.

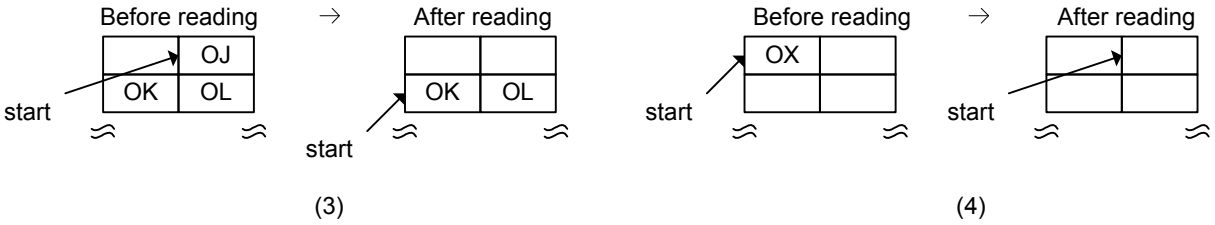


Fig. 1-80 FIFO reading processing (caution required)

Described below is an example of reading processing for fractional number processing, based on the details described above.

1) When data sent from the USB as 64 bytes is read out as 31 to 33 bytes.
   (1) The CPUIF latches the 64-byte ready and initiates the continuous read sequence.
   (2) 30 bytes of data are word-read by the FIFO_Rd_0,1 registers or byte-read by the FIFO_ByteRd register.
   (3) The 31st data byte is byte-read by the FIFO_ByteRd register. Byte boundaries are formed.
   (4) The 32nd data byte is byte-read. Byte-reading using the FIFO_ByteRd register is recommended here. Data is output to Low if it was word-read using the FIFO_Rd_0,1 registers. Byte boundaries are then cleared.
   (5) The remaining 32 bytes of data are word-read by the FIFO_Rd_0,1 registers or byte-read by the FIFO_ByteRd register.

2) When all 64 bytes of the data sent from the USB as 31 and 33 bytes with JoinCPU_Rd set are to be word-read by the FIFO_Rd_0,1 registers.
   (1) The CPUIF latches the 31-byte ready and starts the continuous operation sequence as soon as 31 bytes of data from the USB have been received.
   (2) 30 bytes of data are word-read.
   (3) The join is temporarily severed to clear the cached 31st data byte (byte boundary).
   (4) The join is restored after the 33 bytes of data have been sent from the USB. (1 + 33 bytes)
   (5) The CPUIF latches the 34-byte ready and starts the continuous operation sequence.
   (6) 34 bytes of data are word-read.

1.7.3.1.6 RAM_Rd Access

As with synchronous register reading, data is output to the external bus with the read (asserted for both XCS and XRD) period as the output enable period. For details, refer to "1.6.5.1 RAM Access (RAM_Rd)."

1.7.3.1.7 Asynchronous Register Access (Writing)

A write pulse is created from the external write signal (XCS, XWRL, H), and the external bus data is written to the register.

1.7.3.1.8 Asynchronous Register Access (Reading)

As with synchronous register reading, register data is output to the external bus with the read (asserted for both XCS and XRD) period as the output enable period.

1.7.3.2 DMA0/DMA1(DMA ch.0/ch.1)

1.7.3.2.1 Basic Functions

The basic DMA operations are as shown below.

[Writing]

XDREQ is asserted to permit DMA transfers if the FIFO contains free space for writing.

[Reading]

If the FIFO contains data that can be read and reading is enabled, XDREQ is asserted and DMA transfer enabled.

The three different DMA access methods are described below.

- Access using XDACK
  This access method is used when ChipConfig.CS_Mode="0" and DMA_Config.DMA_Mode="0." Transfers are performed when the XDACK signal is asserted at the logic level set in ChipConfig.DACK_Level and DMA access is identified.
- Access using XDACK&XCS
  This access method is used when ChipConfig.CS_Mode="1" and DMA_Config.DMA_Mode="0." Transfers are performed when the XDACK signal is asserted at the logic level set in ChipConfig.DACK_Level, XCS is asserted ("0"), and DMA access is identified.
- Access using DMA_Mode
  This access method is used when ChipConfig.CS_Mode="0" and DMA_Config.DMA_Mode="1." Transfers are performed with access to the DMA_Rd/WrData_H/L registers identified as DMA access.

In all cases, DMA_Config.ActiveDMA should be set to "1."

DMA has two operating modes and one operating option.

- Count mode
  Performs DMA transfers for the preset number of counts.
  If the internal FIFO contains free space for writing or data for reading and counts remain in the DMA_Count_HH,HL,LH,LL registers, XDREQ is asserted and DMA transfer enabled.
- Free-run mode
  If the internal FIFO contains free space for writing or data for reading, XDREQ is asserted and DMA transfer enabled.

- REQ assert count option
  This option supports CPU burst reads and writes. It can be used in either count or free-run modes. XDREQ is asserted, enabling DMA transfers, if the free space for writing or data quantity for reading in the FIFO is at least equal to the assert count set in the DMA_Config.ReqAssertCount[1:0] bit. Thus, the transfer of the assert count quantity set is basically guaranteed if XDREQ is temporarily asserted. However, XDREQ is asserted, even if the FIFO free space or data quantity is less than the set assert count, if the FIFO free space or data quantity is at least equal to the remaining count when set to count mode. In this case, the guaranteed transfer quantity will be the remaining count.

DMA data processing is primarily performed in word units. Data processing in byte units is possible only in count mode when the remaining count is "1." Shown below are the XDREQ assert conditions and correlations for permissible XDREQ assert transfer quantities for the various operating modes and options.

Table 1-66 Operating modes, options, and transfer start conditions

Count mode: Using ReqAssertCount option

Condition	Count mode (Count > 0)			
	Count ≥ Req		Count < Req	
	Ready ≥ Req	Ready < Req	Ready ≥ Count	Ready < Count
XDREQ	Assert	Negate	Assert	Negate
Transfer quantity	Req	-	Count	-

Free-run mode: Using ReqAssertCount option

Condition	Free-run mode	
	-	
	Ready ≥ Req	Ready < Req
XDREQ	Assert	Negate
Transfer quantity	Req	-

Count mode: Not using ReqAssertCount option

Condition	Count mode (Count > 0)		
	Count ≥ Ready		Count < Ready
	Ready ≥ 2	Ready < 2	Ready ≥ Count
XDREQ	Assert	Negate	Assert
Transfer quantity	Ready (Ready-1 if ready is odd number)	-	Count

Free-run mode: Not using ReqAssertCount option

Condition	Free-run mode	
	-	
	Ready ≥ 2	Ready < 2
XDREQ	Assert	Negate
Transfer quantity	Ready (Ready-1 if ready is odd number)	-

* Req here is the DMA_Config.ReqAssertCount setting, Ready is the FIFO free space and data quantity, and Count is the DMA_Count_HH,HL,LH,LL value.

1.7.3.2.2 Terminal Setup

The XDREQx{x=1,0} and XDACKx{x=1,0} logic levels can be set using the ChipConfig register settings. Unless otherwise stated, the description below applies to negative logic for both XDREQ and XDACK.

1.7.3.2.3 Count Mode (Write)

[Operation start]

Set the DMA_Count_HH,HL,LH,LL registers to the count value, then set the DMA_Control.DMA_Go bit to "1." XDREQ can be asserted and DMA transfer is enabled if the internal FIFO contains at least 2 bytes of free space (DMA_Ready) for writing and there are counts remaining. If a single byte of free space remains in the FIFO, XDREQ is asserted only when set to count mode and the remaining count number is "1."

If byte boundaries are formed in the FIFO when writing an odd number of bytes, use FIFO clear to clear the byte boundaries after transferring the data from the USB, then start the next write operation. For example, to transfer data every 31 bytes from the USB by writing every 31 bytes from the DMA, (1) set the DMA count to 31 and write 31 bytes of data, (2) wait for the 31 bytes of data to be transferred to the USB, (3) clear the FIFO after checking that the 31 bytes of data have been transferred from the USB, then repeat these steps.

"1" can be read in DMA_Control.DMA_Running bit until the operation halts.

[Operation stop]

The following two conditions must be fulfilled for operation stop.

- A DMA transfer must be complete for the count number set in the DMA_Count_HH,HL,LH,LL registers.
- "1" is written to the DMA_Control.DMA_Stop bit.

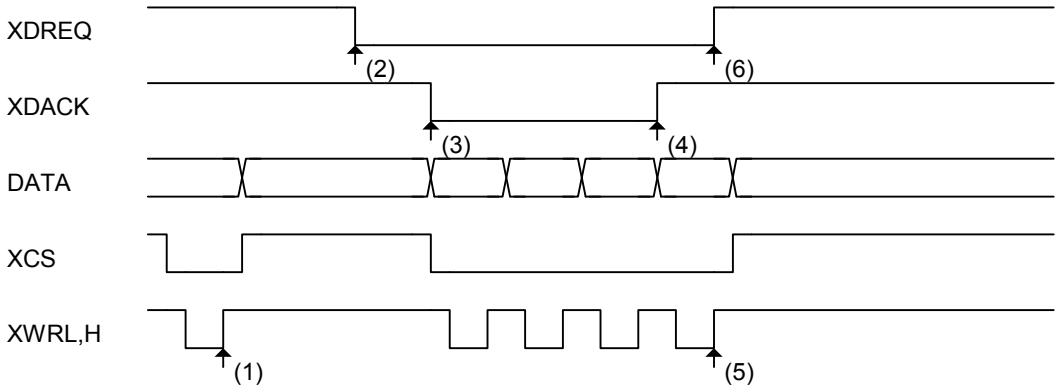
The CPU_IntStat.DMA_Cmp bit is set when DMA operation stops.

XDREQ is negated while the last access strobe is being asserted when the transfer is stopped by the DMA_Count_HH,HL,LH,LL registers.

When transfer is stopped by the DMA_Stop bit, internal chip operations are stopped using the synchronous register access write timing, and XDREQ is negated. To stop the DMA using the DMA_Stop bit, stop the CPU DMAC (master) first.

Fig. 1-81 shows the operational timing for starting transfers in count mode and for stopping transfers using the DMA_Control.DMA_Stop bit before transferring the preset number of counts.

Ex 1: Transfer start conditions: Count (8 bytes) < FIFO free space (16 bytes); Transfer stop conditions: DMA_Stop

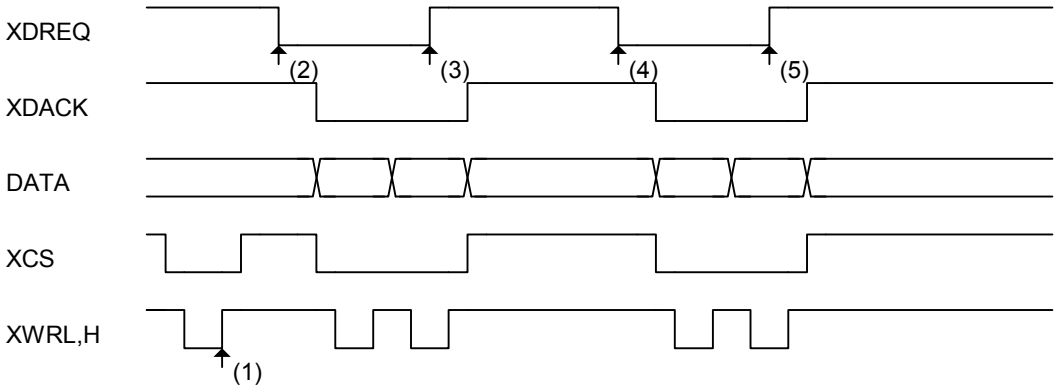


(1) Start DMA circuit operations by writing "1" to the DMA_Control.DMA_Go bit.
(2) Transferring data from the USB creates free space in the FIFO (DMA_Ready). XDREQ is asserted on receipt of DMA_Ready.
(3) XDACK is asserted and DMA transfer starts.
(4) The master stops and XDACK is negated before the count mode transfer quantity is complete.
(5) DMA circuit operation is stopped by writing "1" to the DMA_Control.DMA_Stop bit.
(6) XDREQ is negated on receipt of DMA circuit stop.

Fig. 1-81 Count mode write timing 1

Fig. 1-82 shows the operational timing for starting transfers in count mode and for stopping DMA transfers when the preset number of counts have been transferred.

Ex 2: Transfer start conditions: Count (8 bytes) > FIFO free space (4 bytes); Transfer stop conditions: Count 0



(1) Start DMA circuit operations by writing "1" to the DMA_Control.DMA_Go bit.
(2) Transferring data from the USB creates free space in the FIFO (DMA_Ready). XDREQ is asserted on receipt of DMA_Ready.
(3) XDREQ is negated once DMA_Ready disappears.
(4) Transferring data from the USB creates free space in the FIFO (DMA_Ready). XDREQ is asserted on receipt of DMA_Ready.
(5) XDREQ is negated when the DMA_Count last data is transferred. The DMA circuit is stopped once the DMA_Count quantity has been transferred.

Fig. 1-82 Count mode write timing 2

1.7.3.2.4 Count Mode (Read)

[Operation start]

Set the DMA_Count_HH,HL,LH,LL registers to the count value, then set the DMA_Control.DMA_Go bit to "1." XDREQ is asserted and reading from externally is enabled if the internal FIFO contains at least 2 bytes of data for reading and counts remain. If a single byte of data remains in the FIFO, XDREQ asserted only when count mode is set and the remaining count number is "1."

To describe a typical device operation, the ForceNAK bit is automatically set to "1" to return a NAK response in the count mode reading operation when data exceeding the count number set in the DMA_Count_HH,HL,LH,LL registers remains in the FIFO for the endpoint to which that DMA is connected. Similarly, if a short packet is received from the USB, the ForceNAK bit is automatically set to "1," and the corresponding endpoint NAK response is returned unless the DisAF_NAK_Short bit is set.

If byte boundaries are formed when reading an odd number of bytes, use FIFO clear to clear the byte boundaries before the next transfer. For example, to transfer data every 31 bytes from the USB and read data every 31 bytes from the DMA, (1) receive 31 bytes of data from the USB (ForceNAK is set here and the corresponding endpoint returns a NAK response), (2) read 31 bytes of data from the DMA, (3) clear the FIFO, clear the ForceNAK, permit receipt from the USB, then repeat these steps.

"1" can be read in the DMA_Control.DMA_Running bit until the operation halts.

[Operation stop]

The following two conditions must be met to stop an operation.

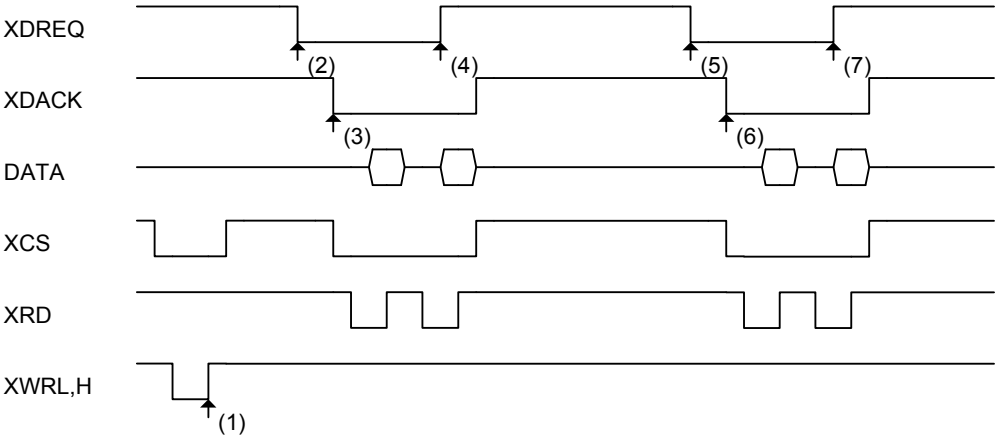
- DMA transfer must be complete for the count number set in the DMA_Count_HH,HL,LH,LL registers.
- "1" is written to the DMA_Control.DMA_Stop bit.

XDREQ is negated while the last access strobe signal is being asserted when the transfer is stopped by the DMA_Count_HH,HL,LH,LL registers.

When a transfer is stopped by the DMA_Stop bit, internal chip operations are stopped using the synchronous register access write timing, and XDREQ is negated. To stop the DMA using the DMA_Stop bit, first stop the CPU DMAC (master).

Fig. 1-83 shows the operational timing for starting transfers in count mode and the DMA transfer ending once the preset number of counts have been transferred.

Ex: Transfer start conditions: Count (8 bytes) > FIFO data (4 bytes); Transfer stop conditions: Count 0



(1) The DMA circuit operation is started by writing "1" to the DMA_Control.DMA_Go bit.
(2) XDREQ is asserted when data is written to the FIFO from the USB, permitting data to be read from externally.
(3) XDACK is asserted and DMA transfer starts.
(4) XDREQ is negated once the FIFO is empty.
(5) XDREQ is asserted when data is written to the FIFO from the USB, permitting data to be read from externally.
(6) XDACK is asserted and DMA transfer starts.
(7) XDREQ is negated when the DMA_Count last data is transferred.

Fig. 1-83 Count mode read timing

1.7.3.2.5 Free-run Mode (Write)

[Operation start]

Set the DMA_Config.FreeRun bit, then write "1" to the DMA_Control.DMA_Go bit. XDREQ is asserted and DMA transfer enabled if there are at least 2 bytes of free space for writing in the internal FIFO. If a single byte of free space remains in the FIFO, XDREQ is not asserted in Free-run mode. To transfer data, refer to "1.7.3.2.3 Count Mode (Write)."

"1" can be read in the DMA_Control.DMA_Running bit until the operation halts.

[Operation stop]

The following condition must be met to stop an operation.

- "1" is written to the DMA_Control.DMA_Stop bit.

When a transfer is stopped by the DMA_Stop bit, internal chip operations are stopped using the synchronous register access write timing, and XDREQ is negated. To stop the DMA using the DMA_Stop bit, stop the CPU DMAC (master) first.

The CPU_IntStat.DMA_Countup bit is set if the DMA_Count_HH,HL,LH,LL register values overflow during DMA transfer in Free-run mode. DMA transfer continues even if this occurs, and DMA_Count_HH,HL,LH,LL also continue and are counted.

The operational timing is the same as for Count mode, except that no restrictions apply due to DMA_Count_HH,HL,LH,LL.

1.7.3.2.6 Free-run Mode (Read)

[Operation start]

Set the DMA_Config.FreeRun bit, then write "1" to the DMA_Control.DMA_Go bit. XDREQ is asserted and reading from externally is enabled if there are at least 2 bytes of data for reading in the internal FIFO. If a single byte of valid data remains in the FIFO, DMA transfer is not started. To transfer data, refer to "1.7.3.2.4 Count Mode (Read)."

"1" can be read in the DMA_Control.DMA_Running bit until the operation halts.

[Operation stop]

The following condition must be met to stop an operation.

- "1" is written to the DMA_Control.DMA_Stop bit.

When a transfer is stopped by the DMA_Stop bit, internal chip operations are stopped using the synchronous register access write timing, and XDREQ is negated. To stop the DMA using the DMA_Stop bit, stop the CPU DMAC (master) first.

The CPU_IntStat.DMA_Countup bit is set if the DMA_Count_HH,HL,LH,LL register values overflow during DMA transfer in Free-run mode. DMA transfer continues even if this occurs, and DMA_Count_HH,HL,LH,LL also continue and are counted.

The operational timing is the same as for Count mode, except that no restrictions apply due to DMA_Count_HH,HL,LH,LL.

1.7.3.2.7 REQ Assert Count Option (Write)

[Operation start]

Set the assert count using the DMA_Config.ReqAssertCount[1:0] bit, then set the DMA_Control.DMA_Go bit to "1." XDREQ is asserted and DMA transfer enabled if free space for writing in the internal FIFO is at least equal to the assert count number. Thus, the transfer of the assert count quantity set is guaranteed if XDREQ is asserted temporarily. However, XDREQ is asserted, even if the free space is less than the assert count, if the free space is at least equal to the remaining count when set to count mode. In this case, the guaranteed transfer quantity will be the remaining count.

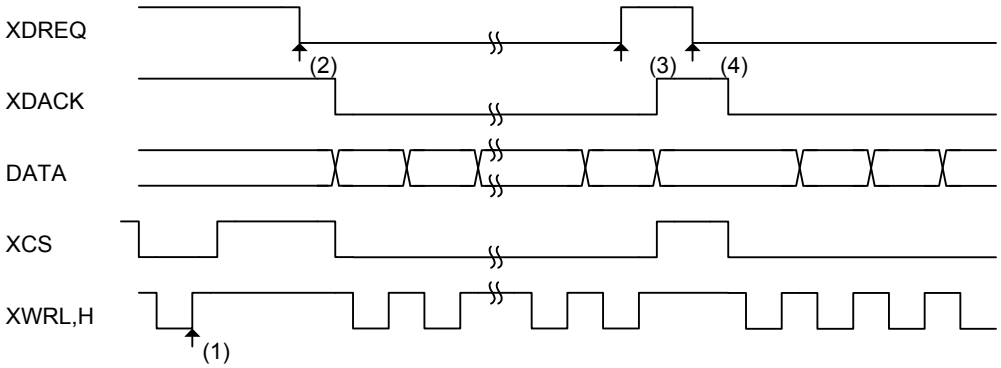
XDREQ is temporarily negated for each transfer quantity set by the ReqAssertCount[1:0] bit in this mode.

"1" can be read in the DMA_Control.DMA_Running bit until the operation halts.

[Operation stop]

For detailed information on operation stop conditions, refer to "1.7.3.2.3 Count Mode (Write)" and "1.7.3.2.5 Free-run Mode (Write)."

Ex: Transfer start: REQ assert count (8-beat: 16-byte)



(1) Start DMA circuit operations by writing "1" to the DMA_Control.DMA_Go bit.
XDREQ is not asserted since the DMA_Ready value is less than the continuous transfer quantity.
(2) Transferring data from the USB creates valid free space in the FIFO (DMA_Ready) exceeding the continuous transfer quantity. XDREQ is asserted on receipt of DMA_Ready.
(3) XDREQ is negated when the continuous transfer quantity (REQ assert count) is complete.
(4) Free space for the next continuous transfer (DMA_Ready) exists once the first continuous transfer is complete. DREQ is asserted on receipt of DMA_Ready.

Fig. 1-84 REQ assert count option write timing

1.7.3.2.8 REQ Assert Count Option (Read)

[Operation start]

Set the assert count using the DMA_Config.ReqAssertCount[1:0] bit, then set the DMA_Control.DMA_Go bit to "1." XDREQ is asserted and DMA transfer enabled if the amount of data for reading in the internal FIFO is at least equal to the assert count number and reading from externally is enabled. Thus, the transfer of the assert count quantity set is guaranteed if XDREQ is asserted temporarily. However, XDREQ is asserted, even if the FIFO data quantity is less than the REQ assert count, if set to Count mode and the data exceeds the count number. In this case, the guaranteed transfer quantity will be the remaining count.

XDREQ is temporarily negated for each transfer quantity set by the ReqAssertCount[1:0] bit in this mode.

"1" can be read in the DMA_Control.DMA_Running bit until the operation halts.

[Operation stop]

For detailed information on operation stop conditions, refer to "1.7.3.2.4 Count Mode (Read)" and "1.7.3.2.6 Free-run Mode (Read)."

For detailed information on operational timing, refer to Figures 1-83 and 1-84.

1.7.3.2.9 DMA FIFO Access Fractional Number Processing

Refer to "1.7.3.1.5 FIFO Access Fractional Number Processing." Note that the DMA lacks an opening for byte-reading or writing.

2. Register Maps

2.1 Register Configuration

Registers are classified as initial registers, device/host common registers for each port, device registers, and host registers. While Port 1 lacks device functions, for the sake of convenience, common classifications apply to device/host common registers.

Port 0 registers range from 0x000 to 0x1FF. Port 1 registers range from 0x200 to 0x3FF. Except where otherwise necessary, the description given below does not distinguish between different ports, since registers other than those with device functions have the same configuration and functions. Common registers are defined for controlling functions shared by both ports.

The register map for this LSI is defined as big-endian. For detailed information on connecting to a little-endian CPU, refer to "Appendix A: Connection to Little-endian CPU."

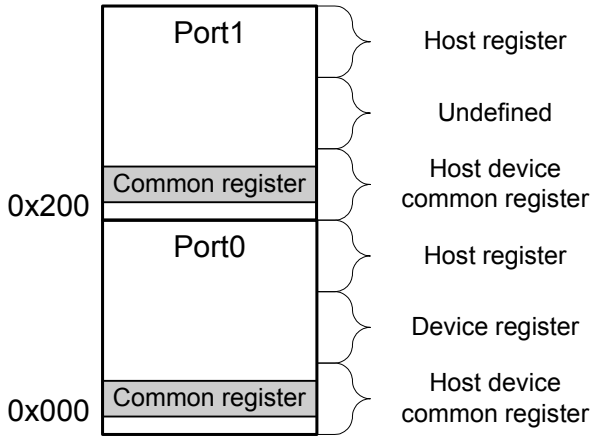


Fig. 2-1 Register map outline

Registers for controlling specific chip functions independent of the different ports are defined as common registers. Common registers are arranged in mirror configurations on the register maps for each port, and accessing them has the same result. Table 2-1 lists the common registers.

Common registers are shaded on the register maps.

Table 2-1 Common register list

Port 0 address	Port 1 address	Register name
0x006	0x206	PortIntStat
0x014, 0x015	0x214, 0x215	WakeupTim_H, WakeupTim_L
0x073	0x273	ClkSelect
0x075	0x275	CPU_Config
0x07E, 0x07F	0x27E, 0x27F	CPUIF_MODE

2.2 Initial Register Map

Only this “Initial register” can be accessed with this LSI from when it is hard-reset until the initial register is set and the setting is enabled (uninitialized period). For details, refer to “1.7.2 CPUIF Mode Setup.”

Word Addr.	Register Name	R/W	Reset	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0x07E	***CPUIF_MODE***	W	0xXXXX						***CPU_Endian***	***BusMode***	
									CPU_Endian	***BusMode***	
0x27E	***CPUIF_MODE***	W	0xXXXX						***CPU_Endian***	***BusMode***	
									CPU_Endian	***BusMode***	

All write accesses to the LSI are treated as write accesses to this register for processing during the uninitialized period. Access to this register is ignored once the initialization period ends.

2.3 Port 0 Register Maps

Port 0 register maps are classified as “Device/host common registers,” “Device registers,” and “Host registers.”

2.3.1 Device/Host Common Register Map

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x000	***MainIntStat***	R/(W)	0x00	***USB_DeviceIntStat***	***USB_HostIntStat***	CPU_IntStat	FIFO_IntStat				***FinishedPM***
0x001	***USB_DeviceIntStat***	R/(W)	0x00	***VBUS_Changed***		***D_SIE_IntStat***	D_BulkIntStat	RcvEP0SETUP		D_EP0IntStat	D_EPrIntStat
0x002	***USB_HostIntStat***	R/(W)	0x00	***VBUS_Err***	***LineStateChanged***	H_SIE_IntStat_1	H_SIE_IntStat_0	H_FrameIntStat		H_CH0IntStat	H_CHrIntStat
0x003	CPU_IntStat	R/(W)	0x00	RAM_RdCmp						DMA_Countup	DMA_Cmp
0x004	FIFO_IntStat	R/(W)	0x00	FIFO_DMA_Cmp					FIFO_NotEmpty	FIFO_Full	FIFO_Empty
0x005			0xXX								
0x006	***RootIntStat***	R	0x00							***Port1IntStat***	***Port0IntStat***
0x007			0xXX								
0x008	***MainIntEnb***	R/W	0x00	***EnUSB_DeviceIntStat***	***EnUSB_HostIntStat***	EnCPU_IntStat	EnFIFO_IntStat				***EnFinishedPM***
0x009	***USB_DeviceIntEnb***	R/W	0x00	***EnVBUS_Changed***		***EnD_SIE_IntStat***	EnD_BulkIntStat	EnRcvEP0SETUP		EnD_EP0IntStat	EnD_EPrIntStat
0x00A	***USB_HostIntEnb***	R/W	0x00	***EnVBUS_Err***	***EnLineStateChaged***	EnH_SIE_IntStat_1	EnH_SIE_IntStat_0	EnH_FrameIntStat		EnH_CH0IntStat	EnH_CHrIntStat
0x00B	CPU_IntEnb	R/W	0x00	EnRAM_RdCmp						EnDMA_Countup	EnDMA_Cmp
0x00C	FIFO_IntEnb	R/W	0x00	EnFIFO_DMA_Cmp					EnFIFO_NotEmpty	EnFIFO_Full	EnFIFO_Empty
0x00D			0xXX								
0x00E			0xXX								
0x00F			0xXX								

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x010	***RevisionNum***	R	0x08	***RevisionNumber***							
0x011	***ChipReset***	R/W	0x80	***ResetMTM***							***AllReset***
0x012	***PM_Control***	R/W	0x00	***GoSLEEP***	***GoACTIVE***	***GoCPU_Cut***				***PM_State[1:0]***	
0x013			0xXX								
0x014	***WakeupTim_H***	R/W	0x00	***WakeupTim [15:8]***							
0x015	***WakeupTim_L***	R/W	0x00	***WakeupTim [7:0]***							
0x016	***H_USB_Control***	R/W	0x00	***VBUS_Enb***							
0x017	***H_XcvrControl***	R/W	0x91	***TermSelect***	***RemoveRPD***	***XcvrSelect[1:0]***				***OpMode[1:0]***	
0x018	***D_USB_Status***	R/W	0xXX	***VBUS***	FSxHS					***LineState[1:0]***	
0x019	***H_USB_Status***	R	0xXX	***VBUS_State***						***LineState[1:0]***	
0x01A			0xXX								
0x01B	MTM_Config	R/W	0x00			MTM_SlopeValue [1:0]				MTM_TermValue [1:0]	
0x01C			0xXX								
0x01D			0xXX								
0x01E			0xXX								
0x01F	***HostDeviceSel***	R/W	0x00								***HOSTxDEVICE***

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x020	FIFO_Rd_0	R	0xXX	FIFO_Rd_0[7:0]							
0x021	FIFO_Rd_1	R	0xXX	FIFO_Rd_1[7:0]							
0x022	FIFO_Wr_0	W	0xXX	FIFO_Wr_0[7:0]							
0x023	FIFO_Wr_1	W	0xXX	FIFO_Wr_1[7:0]							
0x024	FIFO_RdRemain_	R	0x00	RdRemainValid			RdRemain[12:8]				
0x025	FIFO_RdRemain_	R	0x00	RdRemain[7:0]							
0x026	FIFO_WrRemain_	R	0x00				WrRemain[12:8]				
0x027	FIFO_WrRemain_	R	0x00	WrRemain[7:0]							
0x028	FIFO_ByteRd	R	0xXX	FIFO_ByteRd[7:0]							
0x029			0xXX								
0x02A	FIFO_ByteWr	W	0xXX	FIFO_ByteWr[7:0]							
0x02B			0xXX								
0x02C			0xXX								
0x02D			0xXX								
0x02E			0xXX								
0x02F			0xXX								

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x030	RAM_RdAdrs_H	R/W	0x00				RAM_RdAdrs[12:8]				
0x031	RAM_RdAdrs_L	R/W	0x00	RAM_RdAdrs[7:2]							
0x032	RAM_RdControl	R/W	0x00	RAM_GoRdCBW_CSW	RAM_GoRd						
0x033			0xXX								
0x034			0xXX								
0x035	RAM_RdCount	R/W	0x00			RAM_RdCount[5:2]					
0x036			0xXX								
0x037			0xXX								
0x038	RAM_WrAdrs_H	R/W	0x00				RAM_WrAdrs[12:8]				
0x039	RAM_WrAdrs_L	R/W	0x00	RAM_WrAdrs[7:0]							
0x03A	RAM_WrDoor_0	W	0xXX	RAM_WrDoor_0[7:0]							
0x03B	RAM_WrDoor_1	W	0xXX	RAM_WrDoor_1[7:0]							
0x03C			0xXX								
0x03D			0xXX								
0x03E			0xXX								
0x03F			0xXX								

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x040	RAM_Rd_00	R	0x00	RAM_Rd_00[7:0]							
0x041	RAM_Rd_01	R	0x00	RAM_Rd_01[7:0]							
0x042	RAM_Rd_02	R	0x00	RAM_Rd_02[7:0]							
0x043	RAM_Rd_03	R	0x00	RAM_Rd_03[7:0]							
0x044	RAM_Rd_04	R	0x00	RAM_Rd_04[7:0]							
0x045	RAM_Rd_05	R	0x00	RAM_Rd_05[7:0]							
0x046	RAM_Rd_06	R	0x00	RAM_Rd_06[7:0]							
0x047	RAM_Rd_07	R	0x00	RAM_Rd_07[7:0]							
0x048	RAM_Rd_08	R	0x00	RAM_Rd_08[7:0]							
0x049	RAM_Rd_09	R	0x00	RAM_Rd_09[7:0]							
0x04A	RAM_Rd_0A	R	0x00	RAM_Rd_0A[7:0]							
0x04B	RAM_Rd_0B	R	0x00	RAM_Rd_0B[7:0]							
0x04C	RAM_Rd_0C	R	0x00	RAM_Rd_0C[7:0]							
0x04D	RAM_Rd_0D	R	0x00	RAM_Rd_0D[7:0]							
0x04E	RAM_Rd_0E	R	0x00	RAM_Rd_0E[7:0]							
0x04F	RAM_Rd_0F	R	0x00	RAM_Rd_0F[7:0]							

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x050	RAM_Rd_10	R	0x00	RAM_Rd_10[7:0]							
0x051	RAM_Rd_11	R	0x00	RAM_Rd_11[7:0]							
0x052	RAM_Rd_12	R	0x00	RAM_Rd_12[7:0]							
0x053	RAM_Rd_13	R	0x00	RAM_Rd_13[7:0]							
0x054	RAM_Rd_14	R	0x00	RAM_Rd_14[7:0]							
0x055	RAM_Rd_15	R	0x00	RAM_Rd_15[7:0]							
0x056	RAM_Rd_16	R	0x00	RAM_Rd_16[7:0]							
0x057	RAM_Rd_17	R	0x00	RAM_Rd_17[7:0]							
0x058	RAM_Rd_18	R	0x00	RAM_Rd_18[7:0]							
0x059	RAM_Rd_19	R	0x00	RAM_Rd_19[7:0]							
0x05A	RAM_Rd_1A	R	0x00	RAM_Rd_1A[7:0]							
0x05B	RAM_Rd_1B	R	0x00	RAM_Rd_1B[7:0]							
0x05C	RAM_Rd_1C	R	0x00	RAM_Rd_1C[7:0]							
0x05D	RAM_Rd_1D	R	0x00	RAM_Rd_1D[7:0]							
0x05E	RAM_Rd_1E	R	0x00	RAM_Rd_1E[7:0]							
0x05F	RAM_Rd_1F	R	0x00	RAM_Rd_1F[7:0]							

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x060			0xXX								
0x061	DMA_Config	R/W	0x00	FreeRun	DMA_Mode			ActiveDMA		ReqAssertCount [1:0]	
0x062	DMA_Control	R/W	0x00	DMA_Running			CounterClr	Dir		DMA_Stop	DMA_Go
0x063			0xXX								
0x064	DMA_Remain_H	R	0x00				DMA_Remain [12:8]				
0x065	DMA_Remain_L	R	0x00	DMA_Remain [7:0]							
0x066			0xXX								
0x067			0xXX								
0x068	DMA_Count_HH	R/W	0x00	DMA_Count [31:24]							
0x069	DMA_Count_HL	R/W	0x00	DMA_Count [23:16]							
0x06A	DMA_Count_LH	R/W	0x00	DMA_Count [15:8]							
0x06B	DMA_Count_LL	R/W	0x00	DMA_Count [7:0]							
0x06C	DMA_RdData_0	R	0xXX	DMA_RdData_0[7:0]							
0x06D	DMA_RdData_1	R	0xXX	DMA_RdData_1[7:0]							
0x06E	DMA_WrData_0	W	0xXX	DMA_WrData_0[7:0]							
0x06F	DMA_WrData_1	W	0xXX	DMA_WrData_1[7:0]							

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x070			0xXX								
0x071	***ModeProtect***	R/W	0x56	***Protected[7:0](Protected if set to anything other than 56, and reset when set to 0x56)***							
0x072			0xXX								
0x073	***ClkSelect***	R/W	0x00								***ClkSelect***
0x074			0xXX								
0x075	***CPU_Config***	R/W	0x01	***IntLevel***	***IntMode***	***DREQ_Level***	***DACK_Level***	***CS_Mode***	***CPU_Endian***	***BusMode***	***Initialized***
0x076			0xXX								
0x077			0xXX								
0x078			0xXX								
0x079			0xXX								
0x07A			0xXX								
0x07B			0xXX								
0x07C			0xXX								
0x07D			0xXX								
0x07E	***CPUIF_MODE***	W	0xXX	***MergeDMA***					***CPU_Endian***	***BusMode***	
0x07F	***CPUIF_MODE***	W	0xXX	***MergeDMA***					***CPU_Endian***	***BusMode***	

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x080	AREA0StartAdrs_H	R/W	0x00				StartAdrs[12:8]				
0x081	AREA0StartAdrs_L	R/W	0x00	StartAdrs[7:2]							
0x082	AREA0EndAdrs_H	R/W	0x00				EndAdrs[12:8]				
0x083	AREA0EndAdrs_L	R/W	0x00	EndAdrs[7:2]							
0x084	AREA1StartAdrs_H	R/W	0x00				StartAdrs[12:8]				
0x085	AREA1StartAdrs_L	R/W	0x00	StartAdrs[7:2]							
0x086	AREA1EndAdrs_H	R/W	0x00				EndAdrs[12:8]				
0x087	AREA1EndAdrs_L	R/W	0x00	EndAdrs[7:2]							
0x088	AREA2StartAdrs_H	R/W	0x00				StartAdrs[12:8]				
0x089	AREA2StartAdrs_L	R/W	0x00	StartAdrs[7:2]							
0x08A	AREA2EndAdrs_H	R/W	0x00				EndAdrs[12:8]				
0x08B	AREA2EndAdrs_L	R/W	0x00	EndAdrs[7:2]							
0x08C	AREA3StartAdrs_H	R/W	0x00				StartAdrs[12:8]				
0x08D	AREA3StartAdrs_L	R/W	0x00	StartAdrs[7:2]							
0x08E	AREA3EndAdrs_H	R/W	0x00				EndAdrs[12:8]				
0x08F	AREA3EndAdrs_L	R/W	0x00	EndAdrs[7:2]							

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x090	AREA4StartAdrs_H	R/W	0x00				StartAdrs[12:8]				
0x091	AREA4StartAdrs_L	R/W	0x00	StartAdrs[7:2]							
0x092	AREA4EndAdrs_H	R/W	0x00				EndAdrs[12:8]				
0x093	AREA4EndAdrs_L	R/W	0x00	EndAdrs[7:2]							
0x094	AREA5StartAdrs_H	R/W	0x00				StartAdrs[12:8]				
0x095	AREA5StartAdrs_L	R/W	0x00	StartAdrs[7:2]							
0x096	AREA5EndAdrs_H	R/W	0x00				EndAdrs[12:8]				
0x097	AREA5EndAdrs_L	R/W	0x00	EndAdrs[7:2]							
0x098			0xXX								
0x099			0xXX								
0x09A			0xXX								
0x09B			0xXX								
0x09C			0xXX								
0x09D			0xXX								
0x09E			0xXX								
0x09F	AREAnFIFO_Clr	W	0xXX			ClrAREA5	ClrAREA4	ClrAREA3	ClrAREA2	ClrAREA1	ClrAREA0

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x0A0	AREA0Join_0	R/W	0x00	JoinFIFO_Stat					JoinDMA	JoinCPU_Rd	JoinCPU_Wr
0x0A1	AREA0Join_1	R/W	0x00			JoinEPeCHe	JoinEPdCHd	JoinEPcCHc	JoinEPbCHb	JoinEPaCHa	JoinEP0CH0
0x0A2	AREA1Join_0	R/W	0x00	JoinFIFO_Stat					JoinDMA	JoinCPU_Rd	JoinCPU_Wr
0x0A3	AREA1Join_1	R/W	0x00			JoinEPeCHe	JoinEPdCHd	JoinEPcCHc	JoinEPbCHb	JoinEPaCHa	JoinEP0CH0
0x0A4	AREA2Join_0	R/W	0x00	JoinFIFO_Stat					JoinDMA	JoinCPU_Rd	JoinCPU_Wr
0x0A5	AREA2Join_1	R/W	0x00			JoinEPeCHe	JoinEPdCHd	JoinEPcCHc	JoinEPbCHb	JoinEPaCHa	JoinEP0CH0
0x0A6	AREA3Join_0	R/W	0x00	JoinFIFO_Stat					JoinDMA	JoinCPU_Rd	JoinCPU_Wr
0x0A7	AREA3Join_1	R/W	0x00			JoinEPeCHe	JoinEPdCHd	JoinEPcCHc	JoinEPbCHb	JoinEPaCHa	JoinEP0CH0
0x0A8	AREA4Join_0	R/W	0x00	JoinFIFO_Stat					JoinDMA	JoinCPU_Rd	JoinCPU_Wr
0x0A9	AREA4Join_1	R/W	0x00			JoinEPeCHe	JoinEPdCHd	JoinEPcCHc	JoinEPbCHb	JoinEPaCHa	JoinEP0CH0
0x0AA	AREA5Join_0	R/W	0x00	JoinFIFO_Stat					JoinDMA	JoinCPU_Rd	JoinCPU_Wr
0x0AB	AREA5Join_1	R/W	0x00			JoinEPeCHe	JoinEPdCHd	JoinEPcCHc	JoinEPbCHb	JoinEPaCHa	JoinEP0CH0
0x0AC											
0x0AD											
0x0AE	ClrAREAnJoin_0	W	0x00	ClrJoinFIFO_Stat					ClrJoinDMA	ClrJoinCPU_Rd	ClrJoinCPU_Wr
0x0AF	ClrAREAnJoin_1	W	0x00			ClrJoinEPeCHe	ClrJoinEPdCHd	ClrJoinEPcCHc	ClrJoinEPbCHb	ClrJoinEPaCHa	ClrJoinEP0CH0

2.3.2 Device Register Map

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x0B0	***D_SIE_IntStat***	R/(W)	0x00		***NonJ***	RcvSOF	DetectRESET	DetectSUSPEND	ChirpCmp	RestoreCmp	SetAddressCmp
0x0B1			0xXX								
0x0B2		R/(W)	0x00								
0x0B3	D_BulkIntStat	R/(W)	0x00	CBW_Cmp	CBW_LengthErr	CBW_Err		CSW_Cmp	CSW_Err		
0x0B4	D_EPrIntStat	R	0x00	D_AlarmIN_IntStat	D_AlarmOUT_IntStat		D_EPeIntStat	D_EPdIntStat	D_EPcIntStat	D_EPbIntStat	D_EPaIntStat
0x0B5	D_EP0IntStat	R/(W)	0x00	DescriptorCmp	OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
0x0B6	D_EPaIntStat	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
0x0B7	D_EPbIntStat	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
0x0B8	D_EPcIntStat	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
0x0B9	D_EPdIntStat	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
0x0BA	D_EPeIntStat	R/(W)	0x00		OUT_ShortACK	IN_TranACK	OUT_TranACK	IN_TranNAK	OUT_TranNAK	IN_TranErr	OUT_TranErr
0x0BB			0xXX								
0x0BC	D_AlarmIN_IntStat_H	R/(W)	0x00	AlarmEP15IN	AlarmEP14IN	AlarmEP13IN	AlarmEP12IN	AlarmEP11IN	AlarmEP10IN	AlarmEP9IN	AlarmEP8IN
0x0BD	D_AlarmIN_IntStat_L	R/(W)	0x00	AlarmEP7IN	AlarmEP6IN	AlarmEP5IN	AlarmEP4IN	AlarmEP3IN	AlarmEP2IN	AlarmEP1IN	
0x0BE	D_AlarmOUT_IntStat_H	R/(W)	0x00	AlarmEP15OUT	AlarmEP14OUT	AlarmEP13OUT	AlarmEP12OUT	AlarmEP11OUT	AlarmEP10OUT	AlarmEP9OUT	AlarmEP8OUT
0x0BF	D_AlarmOUT_IntStat_L	R/(W)	0x00	AlarmEP7OUT	AlarmEP6OUT	AlarmEP5OUT	AlarmEP4OUT	AlarmEP3OUT	AlarmEP2OUT	AlarmEP1OUT	

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x0C0	***D_SIE_IntEnb***	R/W	0x00		***EnNonJ***	EnRcvSOF	EnDetectRESET	EnDetectSUSPEND	EnChirpCmp	EnRestoreCmp	EnSetAddressCmp
0x0C1			0xXX								
0x0C2		R/W	0x00								
0x0C3	D_BulkIntEnb	R/W	0x00	EnCBW_Cmp	EnCBW_LengthErr	EnCBW_Err		EnCSW_Cmp	EnCSW_Err		
0x0C4	D_EPrIntEnb	R/W	0x00	EnD_AlarmIN_IntStat	EnD_AlarmOUT_IntStat		EnD_EPeIntStat	EnD_EPdIntStat	EnD_EPcIntStat	EnD_EPbIntStat	EnD_EPaIntStat
0x0C5	D_EP0IntEnb	R/W	0x00	EnDescriptorCmp	EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
0x0C6	D_EPaIntEnb	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
0x0C7	D_EPbIntEnb	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
0x0C8	D_EPcIntEnb	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
0x0C9	D_EPdIntEnb	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
0x0CA	D_EPeIntEnb	R/W	0x00		EnOUT_ShortACK	EnIN_TranACK	EnOUT_TranACK	EnIN_TranNAK	EnOUT_TranNAK	EnIN_TranErr	EnOUT_TranErr
0x0CB			0xXX								
0x0CC	D_AlarmIN_IntEnb_H	R/W	0x00	EnAlarmEP15IN	EnAlarmEP14IN	EnAlarmEP13IN	EnAlarmEP12IN	EnAlarmEP11IN	EnAlarmEP10IN	EnAlarmEP9IN	EnAlarmEP8IN
0x0CD	D_AlarmIN_IntEnb_L	R/W	0x00	EnAlarmEP7IN	EnAlarmEP6IN	EnAlarmEP5IN	EnAlarmEP4IN	EnAlarmEP3IN	EnAlarmEP2IN	EnAlarmEP1IN	
0x0CE	D_AlarmOUT_IntEnb_H	R/W	0x00	EnAlarmEP15OUT	EnAlarmEP14OUT	EnAlarmEP13OUT	EnAlarmEP12OUT	EnAlarmEP11OUT	EnAlarmEP10OUT	EnAlarmEP9OUT	EnAlarmEP8OUT
0x0CF	D_AlarmOUT_IntEnb_L	R/W	0x00	EnAlarmEP7OUT	EnAlarmEP6OUT	EnAlarmEP5OUT	EnAlarmEP4OUT	EnAlarmEP3OUT	EnAlarmEP2OUT	EnAlarmEP1OUT	

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x0D0	D_NegoControl	R/W	0x00	DisBusDetect	EnAutoNego	InSUSPEND	DisableHS	SendWakeup	RestoreUSB	GoChirp	ActiveUSB
0x0D1			0xXX								
0x0D2			0xXX								
0x0D3	D_XcvrControl	R/W	0x41	TermSelect	XcvrSelect					OpMode [1:0]	
0x0D4	D_USB_Test	R/W	0x00	EnHS_Test				Test_SE0_NAK	Test_J	Test_K	Test_Packet
0x0D5			0xXX								
0x0D6	D_EPnControl	W	0xXX	AllForceNAK	EPrForceSTALL						
0x0D7			0xXX								
0x0D8	D_BulkOnlyControl	R/W	0x00	AutoForceNAK_CBW					GoCBW_Mode	GoCSW_Mode	
0x0D9	D_BulkOnlyConfig	R/W	0x00				EPeBulkOnly	EPdBulkOnly	EPcBulkOnly	EPbBulkOnly	EPaBulkOnly
0x0DA			0xXX								
0x0DB			0xXX								
0x0DC			0xXX								
0x0DD			0xXX								
0x0DE			0xXX								
0x0DF			0xXX								

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x0E0	D_EP0SETUP_0	R	0x00	SETUP 0[7:0]							
0x0E1	D_EP0SETUP_1	R	0x00	SETUP 1[7:0]							
0x0E2	D_EP0SETUP_2	R	0x00	SETUP 2[7:0]							
0x0E3	D_EP0SETUP_3	R	0x00	SETUP 3[7:0]							
0x0E4	D_EP0SETUP_4	R	0x00	SETUP 4[7:0]							
0x0E5	D_EP0SETUP_5	R	0x00	SETUP 5[7:0]							
0x0E6	D_EP0SETUP_6	R	0x00	SETUP 6[7:0]							
0x0E7	D_EP0SETUP_7	R	0x00	SETUP 7[7:0]							
0x0E8	D_USB_Address	R/(W)	0x00	SetAddress	USB_Address [6:0]						
0x0E9			0xXX								
0x0EA	D_SETUP_Control	R/W	0x00								ProtectEP0
0x0EB			0xXX								
0x0EC			0xXX								
0x0ED			0xXX								
0x0EE	D_FrameNumber_H	R	0x00	Fn_Invalid					FrameNumber [10:8]		
0x0EF	D_FrameNumber_L	R	0x80	FrameNumber [7:0]							

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x0F0	D_EP0MaxSize	R/W	0x00	EP0MaxSize[6:3]							
0x0F1	D_EP0Control	R/W	0x00	INxOUT							ReplyDescriptor
0x0F2	D_EP0ControlIN	R/W	0x00		EnShortPkt		ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
0x0F3	D_EP0ControlOUT	R/W	0x00	AutoForceNAK			ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
0x0F4			0xXX								
0x0F5			0xXX								
0x0F6			0xXX								
0x0F7			0x00								
0x0F8	D_EPaMaxSize_H	R/W	0x00						EPaMaxSize[10:8]		
0x0F9	D_EPaMaxSize_L	R/W	0x00	EPaMaxSize[7:0]							
0x0FA	D_EPaConfig	R/W	0x00	INxOUT	IntEP_Mode	ISO		EndpointNumber[3:0]			
0x0FB			0xXX								
0x0FC	D_EPaControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
0x0FD			0xXX								
0x0FE			0xXX								
0x0FF			0xXX								

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x100	D_EPbMaxSize_H	R/W	0x00						EPbMaxSize[10:8]		
0x101	D_EPbMaxSize_L	R/W	0x00	EPbMaxSize[7:0]							
0x102	D_EPbConfig	R/W	0x00	INxOUT	IntEP_Mode	ISO		EndpointNumber[3:0]			
0x103			0xXX								
0x104	D_EPbControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
0x105			0xXX								
0x106			0xXX								
0x107			0xXX								
0x108	D_EPcMaxSize_H	R/W	0x00						EPcMaxSize[10:8]		
0x109	D_EPcMaxSize_L	R/W	0x00	EPcMaxSize[7:0]							
0x10A	D_EPcConfig	R/W	0x00	INxOUT	IntEP_Mode	ISO		EndpointNumber[3:0]			
0x10B			0xXX								
0x10C	D_EPcControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
0x10D			0xXX								
0x10E			0xXX								
0x10F			0xXX								

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x110	D_EPdMaxSize_H	R/W	0x00						EPdMaxSize[10:8]		
0x111	D_EPdMaxSize_L	R/W	0x00	EPdMaxSize[7:0]							
0x112	D_EPdConfig	R/W	0x00	INxOUT	IntEP_Mode	ISO		EndpointNumber[3:0]			
0x113			0xXX								
0x114	D_EPdControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
0x115			0xXX								
0x116			0xXX								
0x117			0xXX								
0x118	D_EPeMaxSize_H	R/W	0x00						EPeMaxSize[10:8]		
0x119	D_EPeMaxSize_L	R/W	0x00	EPeMaxSize[7:0]							
0x11A	D_EPeConfig	R/W	0x00	INxOUT	IntEP_Mode	ISO		EndpointNumber[3:0]			
0x11B			0xXX								
0x11C	D_EPeControl	R/W	0x00	AutoForceNAK	EnShortPkt	DisAF_NAK_Short	ToggleStat	ToggleSet	ToggleClr	ForceNAK	ForceSTALL
0x11D			0xXX								
0x11E			0xXX								
0x11F			0xXX								

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x120	D_DescAdrs_H	R/W	0x00				DescAdrs[12:8]				
0x121	D_DescAdrs_L	R/W	0x00	DescAdrs [7:0]							
0x122	D_DescSize_H	R/W	0x00							DescSize [9:8]	
0x123	D_DescSize_L	R/W	0x00	DescSize [7:0]							
0x124			0xXX								
0x125			0xXX								
0x126	D_EP_DMA_Ctrl	R/W	0xXX	FIFO_Running	AutoEnShort						
0x127			0xXX								
0x128	D_EnEP_IN_H	R/W	0x00	EnEP15IN	EnEP14IN	EnEP13IN	EnEP12IN	EnEP11IN	EnEP10IN	EnEP9IN	EnEP8IN
0x129	D_EnEP_IN_L	R/W	0x00	EnEP7IN	EnEP6IN	EnEP5IN	EnEP4IN	EnEP3IN	EnEP2IN	EnEP1IN	
0x12A	D_EnEP_OUT_H	R/W	0x00	EnEP15OUT	EnEP14OUT	EnEP13OUT	EnEP12OUT	EnEP11OUT	EnEP10OUT	EnEP9OUT	EnEP8OUT
0x12B	D_EnEP_OUT_L	R/W	0x00	EnEP7OUT	EnEP6OUT	EnEP5OUT	EnEP4OUT	EnEP3OUT	EnEP2OUT	EnEP1OUT	
0x12C	D_EnEP_IN_ISO_H	R/W	0x00	EnEP15IN_ISO	EnEP14IN_ISO	EnEP13IN_ISO	EnEP12IN_ISO	EnEP11IN_ISO	EnEP10IN_ISO	EnEP9IN_ISO	EnEP8IN_ISO
0x12D	D_EnEP_IN_ISO_L	R/W	0x00	EnEP7IN_ISO	EnEP6IN_ISO	EnEP5IN_ISO	EnEP4IN_ISO	EnEP3IN_ISO	EnEP2IN_ISO	EnEP1IN_ISO	
0x12E	D_EnEP_OUT_ISO_H	R/W	0x00	EnEP15OUT_ISO	EnEP14OUT_ISO	EnEP13OUT_ISO	EnEP12OUT_ISO	EnEP11OUT_ISO	EnEP10OUT_ISO	EnEP9OUT_ISO	EnEP8OUT_ISO
0x12F	D_EnEP_OUT_ISO_L	R/W	0x00	EnEP7OUT_ISO	EnEP6OUT_ISO	EnEP5OUT_ISO	EnEP4OUT_ISO	EnEP3OUT_ISO	EnEP2OUT_ISO	EnEP1OUT_ISO	

2.3.3 Host Register Map

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x140	H_SIE_IntStat_0	R/(W)	0x00				DetectCon	DetectDiscon	DetectRmtWkup	DetectDevChirpOK	DetectDevChirpNG
0x141	H_SIE_IntStat_1	R/(W)	0x00					DisabledCmp	ResumeCmp	SuspendCmp	ResetCmp
0x142		R/(W)	0x00								
0x143	H_FrameIntStat	R/(W)	0x00	TriggerFrame					PortErr	FrameNumOver	SOF
0x144	H_CHrIntStat	R	0x00				H_CHeIntStat	H_CHdIntStat	H_CHcIntStat	H_CHbIntStat	H_CHaIntStat
0x145	H_CH0IntStat	R/(W)	0x00	TotalSizeCmp	TranACK	TranErr	ChangeCondition			CTL_SupportCmp	CTL_SupportStop
0x146	H_CHaIntStat	R/(W)	0x00	TotalSizeCmp	TranACK	TranErr	ChangeCondition			BO_SupportCmp	BO_SupportStop
0x147	H_CHbIntStat	R/(W)	0x00	TotalSizeCmp	TranACK	TranErr	ChangeCondition				
0x148	H_CHcIntStat	R/(W)	0x00	TotalSizeCmp	TranACK	TranErr	ChangeCondition				
0x149	H_CHdIntStat	R/(W)	0x00	TotalSizeCmp	TranACK	TranErr	ChangeCondition				
0x14A	H_CHeIntStat	R/(W)	0x00	TotalSizeCmp	TranACK	TranErr	ChangeCondition				
0x14B			0xXX								
0x14C			0xXX								
0x14D			0xXX								
0x14E			0xXX								
0x14F			0xXX								

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x150	H_SIE_IntEnb_0	R/W	0x00				EnDetectCon	EnDetectDiscon	EnDetectRmtWkup	EnDetectDevChirpOK	EnDetectDevChirpNG
0x151	H_SIE_IntEnb_1	R/W	0x00					EnDisabledCmp	EnResumeCmp	EnSuspendCmp	EnResetCmp
0x152		R/W	0x00								
0x153	H_FrameIntEnb	R/W	0x00	EnTriggerFrame					EnPortErr	EnFrameNumOver	EnSOF
0x154	H_CHrIntEnb	R/W	0x00				EnH_CHeIntStat	EnH_CHdIntStat	EnH_CHcIntStat	EnH_CHbIntStat	EnH_CHaIntStat
0x155	H_CH0IntEnb	R/W	0x00	EnTotalSizeCmp	EnTranACK	EnTranErr	EnChangeCondition			EnCTL_SupportCmp	EnCTL_SupportStop
0x156	H_CHaIntEnb	R/W	0x00	EnTotalSizeCmp	EnTranACK	EnTranErr	EnChangeCondition			EnBO_SupportCmp	EnBO_SupportStop
0x157	H_CHbIntEnb	R/W	0x00	EnTotalSizeCmp	EnTranACK	EnTranErr	EnChangeCondition				
0x158	H_CHcIntEnb	R/W	0x00	EnTotalSizeCmp	EnTranACK	EnTranErr	EnChangeCondition				
0x159	H_CHdIntEnb	R/W	0x00	EnTotalSizeCmp	EnTranACK	EnTranErr	EnChangeCondition				
0x15A	H_CHeIntEnb	R/W	0x00	EnTotalSizeCmp	EnTranACK	EnTranErr	EnChangeCondition				
0x15B			0xXX								
0x15C			0xXX								
0x15D			0xXX								
0x15E			0xXX								
0x15F			0xXX								

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x160	H_NegoControl_0	R/W	0x1X	AutoModeCancel	HostState[2:0]			AutoMode[3:0]			
0x161			0xXX								
0x162	H_NegoControl_1	R/W	0x10			PortSpeed[1:0]				DisChirpFinish	RmtWkupDetEnb
0x163			0xXX								
0x164	H_USB_Test	R/W	0x00	EnHS_Test			Test_Force_Enable	Test_SE0_NAK	Test_J	Test_K	Test_Packet
0x165			0xXX								
0x166			0xXX								
0x167			0xXX								
0x168			0xXX								
0x169			0xXX								
0x16A			0xXX								
0x16B			0xXX								
0x16C			0xXX								
0x16D			0xXX								
0x16E			0xXX								
0x16F			0xXX								

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x170	H_CH0SETUP_0	R/W	0x00	SETUP 0[7:0]							
0x171	H_CH0SETUP_1	R/W	0x00	SETUP 1[7:0]							
0x172	H_CH0SETUP_2	R/W	0x00	SETUP 2[7:0]							
0x173	H_CH0SETUP_3	R/W	0x00	SETUP 3[7:0]							
0x174	H_CH0SETUP_4	R/W	0x00	SETUP 4[7:0]							
0x175	H_CH0SETUP_5	R/W	0x00	SETUP 5[7:0]							
0x176	H_CH0SETUP_6	R/W	0x00	SETUP 6[7:0]							
0x177	H_CH0SETUP_7	R/W	0x00	SETUP 7[7:0]							
0x178			0xXX								
0x179			0xXX								
0x17A			0xXX								
0x17B			0xXX								
0x17C			0xXX								
0x17D			0xXX								
0x17E	H_FrameNumber_H	R	0xFF						FrameNumber[10:8]		
0x17F	H_FrameNumber_L	R	0x07	FrameNumber[7:0]							

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x180	H_CH0Config_0	R/W	0x00	ACK_Cnt[3:0]				SpeedMode[1:0]		Toggle	TranGo
0x181	H_CH0Config_1	R/W	0x00	TID[1:0]							
0x182			0xXX								
0x183	H_CH0MaxPktSize	R/W	0x00	MaxPktSize[6:0]							
0x184			0xXX								
0x185			0xXX								
0x186	H_CH0TotalSize_H	R/W	0x00	TotalSize[15:8]							
0x187	H_CH0TotalSize_L	R/W	0x00	TotalSize[7:0]							
0x188	H_CH0HubAdrs	R/W	0x00	HubAdrs[3:0]					Port[2:0]		
0x189	H_CH0FuncAdrs	R/W	0x00	FuncAdrs[3:0]				EP_Number[3:0]			
0x18A			0xXX								
0x18B	H_CTL_SupportControl	R/W	0x00			CTL_SupportState[1:0]					CTL_SupportGo
0x18C			0xXX								
0x18D			0xXX								
0x18E	H_CH0ConditionCode	R	0x00		ConditonCode[2:0]						
0x18F			0xXX								

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x190	H_CHaConfig_0	R/W	0x00	ACK_Cnt[3:0]				SpeedMode[1:0]		Toggle	TranGo
0x191	H_CHaConfig_1	R/W	0x00	TID[1:0]				AutoZerolen			TotalSizeFree
0x192	H_CHaMaxPktSize_H	R/W	0x00						(MaxPktSize[10])	MaxPktSize[9:8]	
0x193	H_CHaMaxPktSize_L	R/W	0x00	MaxPktSize[7:0]							
0x194	H_CHaTotalSize_HH	R/W	0x00	TotalSize[31:24]							
0x195	H_CHaTotalSize_HL	R/W	0x00	TotalSize[23:16]							
0x196	H_CHaTotalSize_LH	R/W	0x00	TotalSize[15:8]							
0x197	H_CHaTotalSize_LL	R/W	0x00	TotalSize[7:0]							
0x198	H_CHaHubAdrs	R/W	0x00	HubAdrs[3:0]					Port[2:0]		
0x199	H_CHaFuncAdrs	R/W	0x00	FuncAdrs[3:0]				EP_Number[3:0]			
0x19A	H_CHaBO_SupportCtl	R/W	0x00			BO_TransportState[1:0]					BO_SupportGo
0x19B	H_CHaBO_CSW_RcvSize	R	0x00					CSW_RcvDataSize[3:0]			
0x19C	H_CHaBO_OUT_EP_Ctl	R/W	0x00				OUT_Toggle	OUT_EP_Number[3:0]			
0x19D	H_CHaBO_IN_EP_Ctl	R/W	0x00				IN_Toggle	IN_EP_Number[3:0]			
0x19E	H_CHaConditionCode	R	0x00		ConditonCode[2:0]						
0x19F			0xXX								

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x1A0	H_CHbConfig_0	R/W	0x00	ACK_Cnt[3:0]				SpeedMode[1:0]		Toggle	TranGo
0x1A1	H_CHbConfig_1	R/W	0x00	TID[1:0]		TranType[1:0]		AutoZerolen	Audio441	TotalSizeFree[1:0]	
0x1A2	H_CHbMaxPktSize_H	R/W	0x00						(MaxPktSize[10])	MaxPktSize[9:8]	
0x1A3	H_CHbMaxPktSize_L	R/W	0x00	MaxPktSize[7:0]							
0x1A4	H_CHbTotalSize_HH	R/W	0x00	TotalSize[31:24]							
0x1A5	H_CHbTotalSize_HL	R/W	0x00	TotalSize[23:16]							
0x1A6	H_CHbTotalSize_LH	R/W	0x00	TotalSize[15:8]							
0x1A7	H_CHbTotalSize_LL	R/W	0x00	TotalSize[7:0]							
0x1A8	H_CHbHubAdrs	R/W	0x00	HubAdrs[3:0]					Port[2:0]		
0x1A9	H_CHbFuncAdrs	R/W	0x00	FuncAdrs[3:0]				EP_Number[3:0]			
0x1AA	H_CHbInterval_H	R/W	0x00						Interval[10:8]		
0x1AB	H_CHbInterval_L	R/W	0x00	Interval[7:0]							
0x1AC	H_CHbTranPause	R/W	0x00							EnTranPause	TranPause
0x1AD			0xXX								
0x1AE	H_CHbConditionCode	R	0x00		ConditonCode[2:0]						
0x1AF			0xXX								

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x1B0	H_CHcConfig_0	R/W	0x00	ACK_Cnt[3:0]				SpeedMode[1:0]		Toggle	TranGo
0x1B1	H_CHcConfig_1	R/W	0x00	TID[1:0]		TranType[1:0]		AutoZerolen	Audio441	TotalSizeFree[1:0]	
0x1B2	H_CHcMaxPktSize_H	R/W	0x00						(MaxPktSize[10])	MaxPktSize[9:8]	
0x1B3	H_CHcMaxPktSize_L	R/W	0x00	MaxPktSize[7:0]							
0x1B4	H_CHcTotalSize_HH	R/W	0x00	TotalSize[31:24]							
0x1B5	H_CHcTotalSize_HL	R/W	0x00	TotalSize[23:16]							
0x1B6	H_CHcTotalSize_LH	R/W	0x00	TotalSize[15:8]							
0x1B7	H_CHcTotalSize_LL	R/W	0x00	TotalSize[7:0]							
0x1B8	H_CHcHubAdrs	R/W	0x00	HubAdrs[3:0]					Port[2:0]		
0x1B9	H_CHcFuncAdrs	R/W	0x00	FuncAdrs[3:0]				EP_Number[3:0]			
0x1BA	H_CHcInterval_H	R/W	0x00						Interval[10:8]		
0x1BB	H_CHcInterval_L	R/W	0x00	Interval[7:0]							
0x1BC	H_CHcTranPause	R/W	0x00							EnTranPause	TranPause
0x1BD			0xXX								
0x1BE	H_CHcConditionCode	R	0x00		ConditonCode[2:0]						
0x1BF			0xXX								

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x1C0	H_CHdConfig_0	R/W	0x00	ACK_Cnt[3:0]				SpeedMode[1:0]		Toggle	TranGo
0x1C1	H_CHdConfig_1	R/W	0x00	TID[1:0]		TranType[1:0]		AutoZerolen	Audio441	TotalSizeFree[1:0]	
0x1C2	H_CHdMaxPktSize_H	R/W	0x00						(MaxPktSize[10])	MaxPktSize[9:8]	
0x1C3	H_CHdMaxPktSize_L	R/W	0x00	MaxPktSize[7:0]							
0x1C4	H_CHdTotalSize_HH	R/W	0x00	TotalSize[31:24]							
0x1C5	H_CHdTotalSize_HL	R/W	0x00	TotalSize[23:16]							
0x1C6	H_CHdTotalSize_LH	R/W	0x00	TotalSize[15:8]							
0x1C7	H_CHdTotalSize_LL	R/W	0x00	TotalSize[7:0]							
0x1C8	H_CHdHubAdrs	R/W	0x00	HubAdrs[3:0]					Port[2:0]		
0x1C9	H_CHdFuncAdrs	R/W	0x00	FuncAdrs[3:0]				EP_Number[3:0]			
0x1CA	H_CHdInterval_H	R/W	0x00						Interval[10:8]		
0x1CB	H_CHdInterval_L	R/W	0x00	Interval[7:0]							
0x1CC	H_CHdTranPause	R/W	0x00							EnTranPause	TranPause
0x1CD			0xXX								
0x1CE	H_CHdConditionCode	R	0x00		ConditonCode[2:0]						
0x1CF			0xXX								

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x1D0	H_CHeConfig_0	R/W	0x00	ACK_Cnt[3:0]				SpeedMode[1:0]		Toggle	TranGo
0x1D1	H_CHeConfig_1	R/W	0x00	TID[1:0]		TranType[1:0]		AutoZerolen	Audio441	TotalSizeFree[1:0]	
0x1D2	H_CHeMaxPktSize_H	R/W	0x00						(MaxPktSize[10])	MaxPktSize[9:8]	
0x1D3	H_CHeMaxPktSize_L	R/W	0x00	MaxPktSize[7:0]							
0x1D4	H_CHeTotalSize_HH	R/W	0x00	TotalSize[31:24]							
0x1D5	H_CHeTotalSize_HL	R/W	0x00	TotalSize[23:16]							
0x1D6	H_CHeTotalSize_LH	R/W	0x00	TotalSize[15:8]							
0x1D7	H_CHeTotalSize_LL	R/W	0x00	TotalSize[7:0]							
0x1D8	H_CHeHubAdrs	R/W	0x00	HubAdrs[3:0]					Port[2:0]		
0x1D9	H_CHeFuncAdrs	R/W	0x00	FuncAdrs[3:0]				EP_Number[3:0]			
0x1DA	H_CHeInterval_H	R/W	0x00						Interval[10:8]		
0x1DB	H_CHeInterval_L	R/W	0x00	Interval[7:0]							
0x1DC	H_CHeTranPause	R/W	0x00							EnTranPause	TranPause
0x1DD			0xXX								
0x1DE	H_CHeConditionCode	R	0x00		ConditonCode[2:0]						
0x1DF			0xXX								

2.4 Port 1 Register Maps

Port 1 register maps are classified as “Device/host common registers” and “Host registers.” Port 1 lacks device functions, but the same naming scheme used for Port 0 is used to ensure uniform terminology.

2.4.1 Device/Host Common Register Map

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x200	MainIntStat	R/(W)	0x00		USB_HostIntStat	CPU_IntStat	FIFO_IntStat				FinishedPM
0x201			0xXX								
0x202	USB_HostIntStat	R/(W)	0x00	VBUS_Err	LineStateChanged	H_SIE_IntStat_1	H_SIE_IntStat_0	H_FrameIntStat		H_CH0IntStat	H_CHrIntStat
0x203	CPU_IntStat	R/(W)	0x00	RAM_RdCmp						DMA_Countup	DMA_Cmp
0x204	FIFO_IntStat	R/(W)	0x00	FIFO_DMA_Cmp					FIFO_NotEmpty	FIFO_Full	FIFO_Empty
0x205			0xXX								
0x206	RootIntStat	R	0x00							Port1IntStat	Port0IntStat
0x207			0xXX								
0x208	MainIntEnb	R/W	0x00	EnUSB_DeviceIntStat	EnUSB_HostIntStat	EnCPU_IntStat	EnFIFO_IntStat				EnFinishedPM
0x209			0xXX								
0x20A	USB_HostIntEnb	R/W	0x00	EnVBUS_Err	EnLineStateChaged	EnH_SIE_IntStat_1	EnH_SIE_IntStat_0	EnH_FrameIntStat		EnH_CH0IntStat	EnH_CHrIntStat
0x20B	CPU_IntEnb	R/W	0x00	EnRAM_RdCmp						EnDMA_Countup	EnDMA_Cmp
0x20C	FIFO_IntEnb	R/W	0x00	EnFIFO_DMA_Cmp					EnFIFO_NotEmpty	EnFIFO_Full	EnFIFO_Empty
0x20D			0xXX								
0x20E			0xXX								
0x20F			0xXX								

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x210	***RevisionNum***	R	0x08	***RevisionNumber***							
0x211	ChipReset	R/W	0x80	ResetMTM							AllReset
0x212	PM_Control	R/W	0x00	GoSLEEP	GoACTIVE	GoCPU_Cut				PM_State[1:0]	
0x213			0xXX								
0x214	WakeupTim_H	R/W	0x00	WakeupTim [15:8]							
0x215	WakeupTim_L	R/W	0x00	WakeupTim [7:0]							
0x216	H_USB_Control	R/W	0x00	VBUS_Enb							
0x217	H_XcvrControl	R/W	0x91	TermSelect	RemoveRPD	XcvrSelect[1:0]				OpMode[1:0]	
0x218			0xXX								
0x219	H_USB_Status	R	0xXX	VBUS_State						LineState[1:0]	
0x21A			0xXX								
0x21B	MTM_Config	R/W	0x00			MTM_SlopeValue [1:0]				MTM_TermValue [1:0]	
0x21C			0xXX								
0x21D			0xXX								
0x21E			0xXX								
0x21F	HostDeviceSel	R	0x00								1'b1

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x220	FIFO_Rd_0	R	0xXX	FIFO_Rd_0[7:0]							
0x221	FIFO_Rd_1	R	0xXX	FIFO_Rd_1[7:0]							
0x222	FIFO_Wr_0	W	0xXX	FIFO_Wr_0[7:0]							
0x223	FIFO_Wr_1	W	0xXX	FIFO_Wr_1[7:0]							
0x224	FIFO_RdRemain_	R	0x00	RdRemainValid			RdRemain[12:8]				
0x225	FIFO_RdRemain_	R	0x00	RdRemain[7:0]							
0x226	FIFO_WrRemain_	R	0x00				WrRemain[12:8]				
0x227	FIFO_WrRemain_	R	0x00	WrRemain[7:0]							
0x228	FIFO_ByteRd	R	0xXX	FIFO_ByteRd[7:0]							
0x229			0xXX								
0x22A	FIFO_ByteWr	W	0xXX	FIFO_ByteWr[7:0]							
0x22B			0xXX								
0x22C			0xXX								
0x22D			0xXX								
0x22E			0xXX								
0x22F			0xXX								

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x230	RAM_RdAdrs_H	R/W	0x00				RAM_RdAdrs[12:8]				
0x231	RAM_RdAdrs_L	R/W	0x00	RAM_RdAdrs[7:2]							
0x232	RAM_RdControl	R/W	0x00	RAM_GoRdCBW_CSW	RAM_GoRd						
0x233			0xXX								
0x234			0xXX								
0x235	RAM_RdCount	R/W	0x00			RAM_RdCount[5:2]					
0x236			0xXX								
0x237			0xXX								
0x238	RAM_WrAdrs_H	R/W	0x00				RAM_WrAdrs[12:8]				
0x239	RAM_WrAdrs_L	R/W	0x00	RAM_WrAdrs[7:0]							
0x23A	RAM_WrDoor_0	W	0xXX	RAM_WrDoor_0[7:0]							
0x23B	RAM_WrDoor_1	W	0xXX	RAM_WrDoor_1[7:0]							
0x23C			0xXX								
0x23D			0xXX								
0x23E			0xXX								
0x23F			0xXX								

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x240	RAM_Rd_00	R	0x00	RAM_Rd_00[7:0]							
0x241	RAM_Rd_01	R	0x00	RAM_Rd_01[7:0]							
0x242	RAM_Rd_02	R	0x00	RAM_Rd_02[7:0]							
0x243	RAM_Rd_03	R	0x00	RAM_Rd_03[7:0]							
0x244	RAM_Rd_04	R	0x00	RAM_Rd_04[7:0]							
0x245	RAM_Rd_05	R	0x00	RAM_Rd_05[7:0]							
0x246	RAM_Rd_06	R	0x00	RAM_Rd_06[7:0]							
0x247	RAM_Rd_07	R	0x00	RAM_Rd_07[7:0]							
0x248	RAM_Rd_08	R	0x00	RAM_Rd_08[7:0]							
0x249	RAM_Rd_09	R	0x00	RAM_Rd_09[7:0]							
0x24A	RAM_Rd_0A	R	0x00	RAM_Rd_0A[7:0]							
0x24B	RAM_Rd_0B	R	0x00	RAM_Rd_0B[7:0]							
0x24C	RAM_Rd_0C	R	0x00	RAM_Rd_0C[7:0]							
0x24D	RAM_Rd_0D	R	0x00	RAM_Rd_0D[7:0]							
0x24E	RAM_Rd_0E	R	0x00	RAM_Rd_0E[7:0]							
0x24F	RAM_Rd_0F	R	0x00	RAM_Rd_0F[7:0]							

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x250	RAM_Rd_10	R	0x00	RAM_Rd_10[7:0]							
0x251	RAM_Rd_11	R	0x00	RAM_Rd_11[7:0]							
0x252	RAM_Rd_12	R	0x00	RAM_Rd_12[7:0]							
0x253	RAM_Rd_13	R	0x00	RAM_Rd_13[7:0]							
0x254	RAM_Rd_14	R	0x00	RAM_Rd_14[7:0]							
0x255	RAM_Rd_15	R	0x00	RAM_Rd_15[7:0]							
0x256	RAM_Rd_16	R	0x00	RAM_Rd_16[7:0]							
0x257	RAM_Rd_17	R	0x00	RAM_Rd_17[7:0]							
0x258	RAM_Rd_18	R	0x00	RAM_Rd_18[7:0]							
0x259	RAM_Rd_19	R	0x00	RAM_Rd_19[7:0]							
0x25A	RAM_Rd_1A	R	0x00	RAM_Rd_1A[7:0]							
0x25B	RAM_Rd_1B	R	0x00	RAM_Rd_1B[7:0]							
0x25C	RAM_Rd_1C	R	0x00	RAM_Rd_1C[7:0]							
0x25D	RAM_Rd_1D	R	0x00	RAM_Rd_1D[7:0]							
0x25E	RAM_Rd_1E	R	0x00	RAM_Rd_1E[7:0]							
0x25F	RAM_Rd_1F	R	0x00	RAM_Rd_1F[7:0]							

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x260			0xXX								
0x261	DMA_Config	R/W	0x00	FreeRun	DMA_Mode			ActiveDMA		ReqAssertCount [1:0]	
0x262	DMA_Control	R/W	0x00	DMA_Running			CounterClr	Dir		DMA_Stop	DMA_Go
0x263			0xXX								
0x264	DMA_Remain_H	R	0x00				DMA_Remain [12:8]				
0x265	DMA_Remain_L	R	0x00	DMA_Remain [7:0]							
0x266			0xXX								
0x267			0xXX								
0x268	DMA_Count_HH	R/W	0x00	DMA_Count [31:24]							
0x269	DMA_Count_HL	R/W	0x00	DMA_Count [23:16]							
0x26A	DMA_Count_LH	R/W	0x00	DMA_Count [15:8]							
0x26B	DMA_Count_LL	R/W	0x00	DMA_Count [7:0]							
0x26C	DMA_RdData_0	R	0xXX	DMA_RdData_0[7:0]							
0x26D	DMA_RdData_1	R	0xXX	DMA_RdData_1[7:0]							
0x26E	DMA_WrData_0	W	0xXX	DMA_WrData_0[7:0]							
0x26F	DMA_WrData_1	W	0xXX	DMA_WrData_1[7:0]							

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x270			0xXX								
0x271	***ModeProtect***	R/W	0x56	***Protected[7:0](Protected if set to anything other than 56, and reset when set to 0x56)***							
0x272			0xXX								
0x273	***ClkSelect***	R/W	0x00	***ClkSource***						***ClkFreq[1:0]***	
0x274			0xXX								
0x275	***CPU_Config***	R/W	0x00	***IntLevel***	***IntMode***	***DREQ_Level***	***DACK_Level***	***CS_Mode***	***CPU_Endian***	***BusMode***	***Bus8x16***
0x276			0xXX								
0x277			0xXX								
0x278			0xXX								
0x279			0xXX								
0x27A			0xXX								
0x27B			0xXX								
0x27C			0xXX								
0x27D			0xXX								
0x27E	***CPUIF_MODE***	W	0xXX	***MergeDMA***					***CPU_Endian***	***BusMode***	
0x27F	***CPUIF_MODE***	W	0xXX	***MergeDMA***					***CPU_Endian***	***BusMode***	

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x280	AREA0StartAdrs_H	R/W	0x00				StartAdrs[12:8]				
0x281	AREA0StartAdrs_L	R/W	0x00	StartAdrs[7:2]							
0x282	AREA0EndAdrs_H	R/W	0x00				EndAdrs[12:8]				
0x283	AREA0EndAdrs_L	R/W	0x00	EndAdrs[7:2]							
0x284	AREA1StartAdrs_H	R/W	0x00				StartAdrs[12:8]				
0x285	AREA1StartAdrs_L	R/W	0x00	StartAdrs[7:2]							
0x286	AREA1EndAdrs_H	R/W	0x00				EndAdrs[12:8]				
0x287	AREA1EndAdrs_L	R/W	0x00	EndAdrs[7:2]							
0x288	AREA2StartAdrs_H	R/W	0x00				StartAdrs[12:8]				
0x289	AREA2StartAdrs_L	R/W	0x00	StartAdrs[7:2]							
0x28A	AREA2EndAdrs_H	R/W	0x00				EndAdrs[12:8]				
0x28B	AREA2EndAdrs_L	R/W	0x00	EndAdrs[7:2]							
0x28C	AREA3StartAdrs_H	R/W	0x00				StartAdrs[12:8]				
0x28D	AREA3StartAdrs_L	R/W	0x00	StartAdrs[7:2]							
0x28E	AREA3EndAdrs_H	R/W	0x00				EndAdrs[12:8]				
0x28F	AREA3EndAdrs_L	R/W	0x00	EndAdrs[7:2]							

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x290	AREA4StartAdrs_H	R/W	0x00				StartAdrs[12:8]				
0x291	AREA4StartAdrs_L	R/W	0x00	StartAdrs[7:2]							
0x292	AREA4EndAdrs_H	R/W	0x00				EndAdrs[12:8]				
0x293	AREA4EndAdrs_L	R/W	0x00	EndAdrs[7:2]							
0x294	AREA5StartAdrs_H	R/W	0x00				StartAdrs[12:8]				
0x295	AREA5StartAdrs_L	R/W	0x00	StartAdrs[7:2]							
0x296	AREA5EndAdrs_H	R/W	0x00				EndAdrs[12:8]				
0x297	AREA5EndAdrs_L	R/W	0x00	EndAdrs[7:2]							
0x298			0xXX								
0x299			0xXX								
0x29A			0xXX								
0x29B			0xXX								
0x29C			0xXX								
0x29D			0xXX								
0x29E			0xXX								
0x29F	AREAnFIFO_Clr	W	0xXX			ClrAREA5	ClrAREA4	ClrAREA3	ClrAREA2	ClrAREA1	ClrAREA0

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x2A0	AREA0Join_0	R/W	0x00	JoinFIFO_Stat					JoinDMA	JoinCPU_Rd	JoinCPU_Wr
0x2A1	AREA0Join_1	R/W	0x00			JoinEPeCHe	JoinEPdCHd	JoinEPcCHc	JoinEPbCHb	JoinEPaCHa	JoinEP0CH0
0x2A2	AREA1Join_0	R/W	0x00	JoinFIFO_Stat					JoinDMA	JoinCPU_Rd	JoinCPU_Wr
0x2A3	AREA1Join_1	R/W	0x00			JoinEPeCHe	JoinEPdCHd	JoinEPcCHc	JoinEPbCHb	JoinEPaCHa	JoinEP0CH0
0x2A4	AREA2Join_0	R/W	0x00	JoinFIFO_Stat					JoinDMA	JoinCPU_Rd	JoinCPU_Wr
0x2A5	AREA2Join_1	R/W	0x00			JoinEPeCHe	JoinEPdCHd	JoinEPcCHc	JoinEPbCHb	JoinEPaCHa	JoinEP0CH0
0x2A6	AREA3Join_0	R/W	0x00	JoinFIFO_Stat					JoinDMA	JoinCPU_Rd	JoinCPU_Wr
0x2A7	AREA3Join_1	R/W	0x00			JoinEPeCHe	JoinEPdCHd	JoinEPcCHc	JoinEPbCHb	JoinEPaCHa	JoinEP0CH0
0x2A8	AREA4Join_0	R/W	0x00	JoinFIFO_Stat					JoinDMA	JoinCPU_Rd	JoinCPU_Wr
0x2A9	AREA4Join_1	R/W	0x00			JoinEPeCHe	JoinEPdCHd	JoinEPcCHc	JoinEPbCHb	JoinEPaCHa	JoinEP0CH0
0x2AA	AREA5Join_0	R/W	0x00	JoinFIFO_Stat					JoinDMA	JoinCPU_Rd	JoinCPU_Wr
0x2AB	AREA5Join_1	R/W	0x00			JoinEPeCHe	JoinEPdCHd	JoinEPcCHc	JoinEPbCHb	JoinEPaCHa	JoinEP0CH0
0x2AC											
0x2AD											
0x2AE	ClrAREAnJoin_0	W	0x00	ClrJoinFIFO_Stat					ClrJoinDMA	ClrJoinCPU_Rd	ClrJoinCPU_Wr
0x2AF	ClrAREAnJoin_1	W	0x00			ClrJoinEPeCHe	ClrJoinEPdCHd	ClrJoinEPcCHc	ClrJoinEPbCHb	ClrJoinEPaCHa	ClrJoinEP0CH0

2.4.2 Host Register Map

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x340	H_SIE_IntStat_0	R/(W)	0x00				DetectCon	DetectDiscon	DetectRmtWkup	DetectDevChirpOK	DetectDevChirpNG
0x341	H_SIE_IntStat_1	R/(W)	0x00					DisabledCmp	ResumeCmp	SuspendCmp	ResetCmp
0x342		R/(W)	0x00								
0x343	H_FrameIntStat	R/(W)	0x00	TriggerFrame					PortErr	FrameNumOver	SOF
0x344	H_CHrIntStat	R	0x00				H_CHeIntStat	H_CHdIntStat	H_CHcIntStat	H_CHbIntStat	H_CHaIntStat
0x345	H_CH0IntStat	R/(W)	0x00	TotalSizeCmp	TranACK	TranErr	ChangeCondition			CTL_SupportCmp	CTL_SupportStop
0x346	H_CHaIntStat	R/(W)	0x00	TotalSizeCmp	TranACK	TranErr	ChangeCondition			BO_SupportCmp	BO_SupportStop
0x347	H_CHbIntStat	R/(W)	0x00	TotalSizeCmp	TranACK	TranErr	ChangeCondition				
0x348	H_CHcIntStat	R/(W)	0x00	TotalSizeCmp	TranACK	TranErr	ChangeCondition				
0x349	H_CHdIntStat	R/(W)	0x00	TotalSizeCmp	TranACK	TranErr	ChangeCondition				
0x34A	H_CHeIntStat	R/(W)	0x00	TotalSizeCmp	TranACK	TranErr	ChangeCondition				
0x34B			0xXX								
0x34C			0xXX								
0x34D			0xXX								
0x34E			0xXX								
0x34F			0xXX								

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x350	H_SIE_IntEnb_0	R/W	0x00				EnDetectCon	EnDetectDiscon	EnDetectRmtWkup	EnDetectDevChirpOK	EnDetectDevChirpNG
0x351	H_SIE_IntEnb_1	R/W	0x00					EnDisabledCmp	EnResumeCmp	EnSuspendCmp	EnResetCmp
0x352		R/W	0x00								
0x353	H_FrameIntEnb	R/W	0x00	EnTriggerFrame					EnPortErr	EnFrameNumOver	EnSOF
0x354	H_CHrIntEnb	R/W	0x00				EnH_CHeIntStat	EnH_CHdIntStat	EnH_CHcIntStat	EnH_CHbIntStat	EnH_CHaIntStat
0x355	H_CH0IntEnb	R/W	0x00	EnTotalSizeCmp	EnTranACK	EnTranErr	EnChangeCondition			EnCTL_SupportCmp	EnCTL_SupportStop
0x356	H_CHaIntEnb	R/W	0x00	EnTotalSizeCmp	EnTranACK	EnTranErr	EnChangeCondition			EnBO_SupportCmp	EnBO_SupportStop
0x357	H_CHbIntEnb	R/W	0x00	EnTotalSizeCmp	EnTranACK	EnTranErr	EnChangeCondition				
0x358	H_CHcIntEnb	R/W	0x00	EnTotalSizeCmp	EnTranACK	EnTranErr	EnChangeCondition				
0x359	H_CHdIntEnb	R/W	0x00	EnTotalSizeCmp	EnTranACK	EnTranErr	EnChangeCondition				
0x35A	H_CHeIntEnb	R/W	0x00	EnTotalSizeCmp	EnTranACK	EnTranErr	EnChangeCondition				
0x35B			0xXX								
0x35C			0xXX								
0x35D			0xXX								
0x35E			0xXX								
0x35F			0xXX								

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x360	H_NegoControl_0	R/W	0x1X	AutoModeCancel	HostState[2:0]			AutoMode[3:0]			
0x361			0xXX								
0x362	H_NegoControl_1	R/W	0x10			PortSpeed[1:0]				DisChirpFinish	RmtWkupDetEnb
0x363			0xXX								
0x364	H_USB_Test	R/W	0x00	EnHS_Test			Test_Force_Enable	Test_SE0_NAK	Test_J	Test_K	Test_Packet
0x365			0xXX								
0x366			0xXX								
0x367			0xXX								
0x368			0xXX								
0x369			0xXX								
0x36A			0xXX								
0x36B			0xXX								
0x36C			0xXX								
0x36D			0xXX								
0x36E			0xXX								
0x36F			0xXX								

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x370	H_CH0SETUP_0	R/W	0x00	SETUP 0[7:0]							
0x371	H_CH0SETUP_1	R/W	0x00	SETUP 1[7:0]							
0x372	H_CH0SETUP_2	R/W	0x00	SETUP 2[7:0]							
0x373	H_CH0SETUP_3	R/W	0x00	SETUP 3[7:0]							
0x374	H_CH0SETUP_4	R/W	0x00	SETUP 4[7:0]							
0x375	H_CH0SETUP_5	R/W	0x00	SETUP 5[7:0]							
0x376	H_CH0SETUP_6	R/W	0x00	SETUP 6[7:0]							
0x377	H_CH0SETUP_7	R/W	0x00	SETUP 7[7:0]							
0x378			0xXX								
0x379			0xXX								
0x37A			0xXX								
0x37B			0xXX								
0x37C			0xXX								
0x37D			0xXX								
0x37E	H_FrameNumber_H	R	0xFF						FrameNumber[10:8]		
0x37F	H_FrameNumber_L	R	0x07	FrameNumber[7:0]							

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x380	H_CH0Config_0	R/W	0x00	ACK_Cnt[3:0]				SpeedMode[1:0]		Toggle	TranGo
0x381	H_CH0Config_1	R/W	0x00	TID[1:0]							
0x382			0xXX								
0x383	H_CH0MaxPktSize	R/W	0x00	MaxPktSize[6:0]							
0x384			0xXX								
0x385			0xXX								
0x386	H_CH0TotalSize_H	R/W	0x00	TotalSize[15:8]							
0x387	H_CH0TotalSize_L	R/W	0x00	TotalSize[7:0]							
0x388	H_CH0HubAdrs	R/W	0x00	HubAdrs[3:0]					Port[2:0]		
0x389	H_CH0FuncAdrs	R/W	0x00	FuncAdrs[3:0]				EP_Number[3:0]			
0x38A			0xXX								
0x38B	H_CTL_SupportControl	R/W	0x00			CTL_SupportState[1:0]					CTL_SupportGo
0x38C			0xXX								
0x38D			0xXX								
0x38E	H_CH0ConditionCode	R	0x00		ConditonCode[2:0]						
0x38F			0xXX								

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x390	H_CHaConfig_0	R/W	0x00	ACK_Cnt[3:0]				SpeedMode[1:0]		Toggle	TranGo
0x391	H_CHaConfig_1	R/W	0x00	TID[1:0]				AutoZerolen			TotalSizeFree
0x392	H_CHaMaxPktSize_H	R/W	0x00						(MaxPktSize[10])	MaxPktSize[9:8]	
0x393	H_CHaMaxPktSize_L	R/W	0x00	MaxPktSize[7:0]							
0x394	H_CHaTotalSize_HH	R/W	0x00	TotalSize[31:24]							
0x395	H_CHaTotalSize_HL	R/W	0x00	TotalSize[23:16]							
0x396	H_CHaTotalSize_LH	R/W	0x00	TotalSize[15:8]							
0x397	H_CHaTotalSize_LL	R/W	0x00	TotalSize[7:0]							
0x398	H_CHaHubAdrs	R/W	0x00	HubAdrs[3:0]					Port[2:0]		
0x399	H_CHaFuncAdrs	R/W	0x00	FuncAdrs[3:0]				EP_Number[3:0]			
0x39A	H_CHaBO_SupportCtl	R/W	0x00			BO_TransportState[1:0]				BO_SupportGo	
0x39B	H_CHaBO_CSW_RcvSize	R	0x00					CSW_RcvDataSize[3:0]			
0x39C	H_CHaBO_OUT_EP_Ctl	R/W	0x00				OUT_Toggle	OUT_EP_Number[3:0]			
0x39D	H_CHaBO_IN_EP_Ctl	R/W	0x00				IN_Toggle	IN_EP_Number[3:0]			
0x39E	H_CHaConditionCode	R	0x00		ConditonCode[2:0]						
0x39F			0xXX								

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x3A0	H_CHbConfig_0	R/W	0x00	ACK_Cnt[3:0]				SpeedMode[1:0]		Toggle	TranGo
0x3A1	H_CHbConfig_1	R/W	0x00	TID[1:0]		TranType[1:0]		AutoZerolen	Audio441	TotalSizeFree[1:0]	
0x3A2	H_CHbMaxPktSize_H	R/W	0x00						(MaxPktSize[10])	MaxPktSize[9:8]	
0x3A3	H_CHbMaxPktSize_L	R/W	0x00	MaxPktSize[7:0]							
0x3A4	H_CHbTotalSize_HH	R/W	0x00	TotalSize[31:24]							
0x3A5	H_CHbTotalSize_HL	R/W	0x00	TotalSize[23:16]							
0x3A6	H_CHbTotalSize_LH	R/W	0x00	TotalSize[15:8]							
0x3A7	H_CHbTotalSize_LL	R/W	0x00	TotalSize[7:0]							
0x3A8	H_CHbHubAdrs	R/W	0x00	HubAdrs[3:0]					Port[2:0]		
0x3A9	H_CHbFuncAdrs	R/W	0x00	FuncAdrs[3:0]				EP_Number[3:0]			
0x3AA	H_CHbInterval_H	R/W	0x00						Interval[10:8]		
0x3AB	H_CHbInterval_L	R/W	0x00	Interval[7:0]							
0x3AC	H_CHbTranPause	R/W	0x00							EnTranPause	TranPause
0x3AD			0xXX								
0x3AE	H_CHbConditionCode	R	0x00		ConditonCode[2:0]						
0x3AF			0xXX								

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x3B0	H_CHcConfig_0	R/W	0x00	ACK_Cnt[3:0]				SpeedMode[1:0]		Toggle	TranGo
0x3B1	H_CHcConfig_1	R/W	0x00	TID[1:0]		TranType[1:0]		AutoZerolen	Audio441	TotalSizeFree[1:0]	
0x3B2	H_CHcMaxPktSize_H	R/W	0x00						(MaxPktSize[10])	MaxPktSize[9:8]	
0x3B3	H_CHcMaxPktSize_L	R/W	0x00	MaxPktSize[7:0]							
0x3B4	H_CHcTotalSize_HH	R/W	0x00	TotalSize[31:24]							
0x3B5	H_CHcTotalSize_HL	R/W	0x00	TotalSize[23:16]							
0x3B6	H_CHcTotalSize_LH	R/W	0x00	TotalSize[15:8]							
0x3B7	H_CHcTotalSize_LL	R/W	0x00	TotalSize[7:0]							
0x3B8	H_CHcHubAdrs	R/W	0x00	HubAdrs[3:0]					Port[2:0]		
0x3B9	H_CHcFuncAdrs	R/W	0x00	FuncAdrs[3:0]				EP_Number[3:0]			
0x3BA	H_CHcInterval_H	R/W	0x00						Interval[10:8]		
0x3BB	H_CHcInterval_L	R/W	0x00	Interval[7:0]							
0x3BC	H_CHcTranPause	R/W	0x00							EnTranPause	TranPause
0x3BD			0xXX								
0x3BE	H_CHcConditionCode	R	0x00		ConditonCode[2:0]						
0x3BF			0xXX								

Registers that can be read from or written to even in the SLEEP state appear in ***bold italic***. All other registers can be read from or written to in the ACTIVE state.

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x3C0	H_CHdConfig_0	R/W	0x00	ACK_Cnt[3:0]				SpeedMode[1:0]		Toggle	TranGo
0x3C1	H_CHdConfig_1	R/W	0x00	TID[1:0]		TranType[1:0]		AutoZerolen	Audio441	TotalSizeFree[1:0]	
0x3C2	H_CHdMaxPktSize_H	R/W	0x00						(MaxPktSize[10])	MaxPktSize[9:8]	
0x3C3	H_CHdMaxPktSize_L	R/W	0x00	MaxPktSize[7:0]							
0x3C4	H_CHdTotalSize_HH	R/W	0x00	TotalSize[31:24]							
0x3C5	H_CHdTotalSize_HL	R/W	0x00	TotalSize[23:16]							
0x3C6	H_CHdTotalSize_LH	R/W	0x00	TotalSize[15:8]							
0x3C7	H_CHdTotalSize_LL	R/W	0x00	TotalSize[7:0]							
0x3C8	H_CHdHubAdrs	R/W	0x00	HubAdrs[3:0]					Port[2:0]		
0x3C9	H_CHdFuncAdrs	R/W	0x00	FuncAdrs[3:0]				EP_Number[3:0]			
0x3CA	H_CHdInterval_H	R/W	0x00						Interval[10:8]		
0x3CB	H_CHdInterval_L	R/W	0x00	Interval[7:0]							
0x3CC	H_CHdTranPause	R/W	0x00							EnTranPause	TranPause
0x3CD			0xXX								
0x3CE	H_CHdConditionCode	R	0x00		ConditonCode[2:0]						
0x3CF			0xXX								

Address	Register Name	R/W	Reset	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x3D0	H_CHeConfig_0	R/W	0x00	ACK_Cnt[3:0]				SpeedMode[1:0]		Toggle	TranGo
0x3D1	H_CHeConfig_1	R/W	0x00	TID[1:0]		TranType[1:0]		AutoZerolen	Audio441	TotalSizeFree[1:0]	
0x3D2	H_CHeMaxPktSize_H	R/W	0x00						(MaxPktSize[10])	MaxPktSize[9:8]	
0x3D3	H_CHeMaxPktSize_L	R/W	0x00	MaxPktSize[7:0]							
0x3D4	H_CHeTotalSize_HH	R/W	0x00	TotalSize[31:24]							
0x3D5	H_CHeTotalSize_HL	R/W	0x00	TotalSize[23:16]							
0x3D6	H_CHeTotalSize_LH	R/W	0x00	TotalSize[15:8]							
0x3D7	H_CHeTotalSize_LL	R/W	0x00	TotalSize[7:0]							
0x3D8	H_CHeHubAdrs	R/W	0x00	HubAdrs[3:0]					Port[2:0]		
0x3D9	H_CHeFuncAdrs	R/W	0x00	FuncAdrs[3:0]				EP_Number[3:0]			
0x3DA	H_CHeInterval_H	R/W	0x00						Interval[10:8]		
0x3DB	H_CHeInterval_L	R/W	0x00	Interval[7:0]							
0x3DC	H_CHeTranPause	R/W	0x00							EnTranPause	TranPause
0x3DD			0xXX								
0x3DE	H_CHeConditionCode	R	0x00		ConditonCode[2:0]						
0x3DF			0xXX								

3. Register Details

3.1 Device/Host Common Register Details

Registers and register bits related to device functions are defined for Port 0 but not for Port 1. All other registers have the same configuration and functions.

The following description does not distinguish between ports for registers with the same functions and configuration.

Addresses are shown as offset addresses from the base address 000h for Port 0 and from the base address 200h for Port 1.

Registers and register bits defined for Port 0 but not for Port 1 are indicated accordingly.

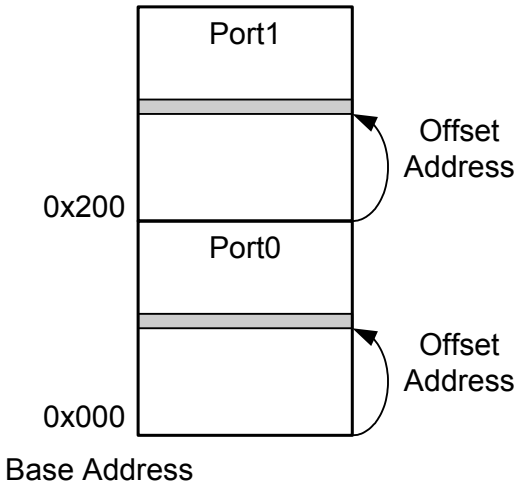


Fig. 3-1 Address notation

3.1.1 Port0:000h / Port1:200h *MainIntStat (Main Interrupt Status)*

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	000h	***MainIntStat***	R	7: ***USB_DeviceIntStat***	0: None	1: USB Device Interrupts	00h
			R	6: ***USB_HostIntStat***	0: None	1: USB Host Interrupts	
			R	5: CPU_IntStat	0: None	1: CPU Interrupts	
			R	4: FIFO_IntStat	0: None	1: FIFO Interrupts	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
			R (W)	0: ***FinishedPM***	0: None	1:Detect FinishedPM	

Indicates the interrupt factors for the LSI.

This register includes bits that directly and indirectly specify the interrupt factors. The bit for indirectly specifying interrupt factors can read the corresponding interrupt status registers to follow the bit for directly specifying interrupt factors. The bit for indirectly specifying interrupt factors is read-only and is automatically cleared by clearing the bit for directly specifying major interrupt factors. The bit for directly specifying interrupt factors can be written to; writing "1" to this bit allows the interrupt factor to be cleared. Setting the interrupt factor to "1" when the corresponding bit interrupt is enabled by the MainIntEnb register asserts the XINT terminal and issues an interrupt to the CPU. Clearing all corresponding interrupt factors negates the XINT terminal.

Bit7 ***USB_DeviceIntStat (*Port0), N/A(Port1)**

Indirectly specifies interrupt factors.

This bit is not defined for Port 1.

It is set to "1" when the USB_DeviceIntStat register includes interrupt factors and the USB_DeviceIntEnb register bit corresponding to the interrupt factors is enabled. This bit allows reading even in SLEEP state.

Bit6 ***USB_HostIntStat***

Indirectly specifies interrupt factors.

It is set to "1" when the USB_HostIntStat register includes interrupt factors and the USB_HostIntEnb register bit corresponding to the interrupt factors is enabled. This bit allows reading even in SLEEP state.

Bit5 **CPU_IntStat**

Indirectly specifies interrupt factors.

It is set to "1" when the CPU_IntStat register includes interrupt factors and the CPU_IntEnb register bit corresponding to the interrupt factors is enabled.

Bit4 **FIFO_IntStat**

Indirectly specifies interrupt factors.

It is set to "1" when the FIFO_IntStat register includes interrupt factors and the FIFO_IntEnb register bit corresponding to the interrupt factors is enabled.

Bit3-1 **Reserved**

Bit0 ***FinishedPM***

Directly specifies interrupt factors.

It is set to "1" on reaching the particular specified state when GoSLEEP or GoACTIVE is set by the PM_Control register. This bit is enabled even in SLEEP state.

3.1.2 Port0:001h / Port1:N/A *USB_DeviceIntStat (USB Device Interrupt Status)*

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	001h	***USB_DeviceIntStat***	R (W)	7: ***VBUS_Changed***	0: None	1: VBUS is Changed	00h
				6:	0:	1:	
			R	5: ***D_SIE_IntStat***	0: None	1: SIE Interrupts	
			R	4: D_BulkIntStat	0: None	1: Bulk Interrupts	
			R (W)	3: RcvEP0SETUP	0: None	1: Receive EP0 SETUP	
				2:	0:	1:	
			R	1: D_EP0IntStat	0: None	1: EP0 Interrupts	
			R	0: D_EPrIntStat	0: None	1: EPr Interrupts	

This register is not defined for Port 1.

Indicates device-related interrupts.

This register includes the bits that directly and indirectly specify the interrupt factors. The bit for indirectly specifying interrupt factors can read the corresponding interrupt status registers to follow the bit for directly specifying interrupt factors. The bit for indirectly specifying interrupt factors is read-only and is automatically cleared by clearing the bit for directly specifying major interrupt factors. The bit for directly specifying interrupt factors can be written to; writing "1" to this bit allows the interrupt factor to be cleared.

Bit7 ***VBUS_Changed***

Directly specifies interrupt factors.

Set to "1" when the VBUS terminal status has changed.

Check the VBUS terminal status using the D_USB_Status register VBUS bit. If VBUS is "0," the cable is disconnected. This bit is enabled even in SLEEP state.

Bit6 **Reserved**

Bit5 ***D_SIE_IntStat***

Indirectly specifies interrupt factors.

Set to "1" when the D_SIE_IntStat register includes interrupt factors and the D_SIE_IntEnb register bit corresponding to the interrupt factors is enabled. This bit allows reading even in SLEEP state.

Bit4 **D_BulkIntStat**

Indirectly specifies interrupt factors.

Set to "1" when D_BulkIntStat register includes interrupt factors and the D_BulkIntEnb register bit corresponding to the interrupt factors is enabled.

Bit3 RcvEP0SETUP

Directly specifies interrupt factors.

Set to "1" when the control transfer setup stage is complete and the received data is stored in registers D_EP0SETUP_0 to D_EP0SETUP_7. The D_EP0ControlIN and D_EP0ControlOUT register ForceSTALL bit is automatically set to "0" at the same time, and the D_EP0ControlIN, D_EP0ControlOUT register ForceNAK, and ToggleStat bits and D_SETUP_Control register ProtectEP0 bit are automatically set to "1." The AutoSetAddress function automatically responds to the SetAddress() request, and this status is not set.

Bit2 Reserved

Bit1 D_EP0IntStat

Indirectly specifies interrupt factors.

Set to "1" when the D_EP0IntStat register includes interrupt factors and the D_EP0IntEnb register bit corresponding to the interrupt factors is enabled.

Bit0 D_EPrIntStat

Indirectly specifies interrupt factors.

Set to "1" when the D_EPrIntStat register includes interrupt factors and the D_EPrIntEnb register bit corresponding to the interrupt factors is enabled.

3.1.3 Port0:002h / Port1:202h *USB_HostIntStat (USB Host Interrupt Status)*

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	002h	***USB_HostIntStat***	R/(W)	7: ***VBUS_Err***	0: None	1: VBUS Error	00h
			R/(W)	6: ***LineStateChanged***	0: None	1: Line Status Changed	
			R	5: H_SIE_IntStat1	0: None	1: SIE Interrupts1	
			R	4: H_SIE_IntStat0	0: None	1: SIE Interrupts0	
			R	3: H_FrameIntStat	0: None	1: Frame Interrupts	
				2:	0:	1:	
			R	1: H_CH0IntStat	0: None	1: CH0 Interrupts	
			R	0: H_CHrIntStat	0: None	1: CHr Interrupts	

Indicates host-related interrupts.

This register includes the bits that directly and indirectly specify the interrupt factors. The bit for indirectly specifying interrupt factors can read the corresponding interrupt status registers to follow the bit for directly specifying interrupt factors. The bit for indirectly specifying interrupt factors is read-only and is automatically cleared by clearing the bit for directly specifying major interrupt factors. The bit for directly specifying interrupt factors can be written to; writing "1" to this bit allows the interrupt factor to be cleared.

Bit7 ***VBUS_Err***

Directly specifies interrupt factors. This bit is enabled even in SLEEP state.

It is set to "1" when a signal is input to the VBUSFLG terminal from the externally connected VBUS power switch indicating a VBUS error (change Edge from High to Low).

Check the VBUSFLG terminal status using the H_USB_Status register VBUS_State bit.

This error signal will depend on the VBUS power switch specifications for the actual external connection. The specifications should be checked beforehand.

Bit6 ***LineStateChanged***

Directly specifies interrupt factors. This bit is enabled even in SLEEP state.

It indicates that the host port DP and DM terminal states have changed.

This interrupt is used to detect signal line changes when the USB host function or USB device function is not being used or when the USB host function is enabled in SUSPEND state.

Bit5 **H_SIE_IntStat1**

Indirectly specifies interrupt factors.

Set to "1" when the H_SIE_IntStat1 register includes interrupt factors and the H_SIE_IntEnb1 register bit corresponding to the interrupt factors is enabled.

Bit4 **H_SIE_IntStat0**

Indirectly specifies interrupt factors.

Set to "1" when the H_SIE_IntStat0 register includes interrupt factors and the H_SIE_IntEnb0 register bit corresponding to the interrupt factors is enabled.

Bit3 **H_FrameIntStat**

Indirectly specifies interrupt factors.

Set to "1" when the H_FrameIntStat register includes interrupt factors and the H_FrameIntEnb register bit corresponding to the interrupt factors is enabled.

Bit2 **Reserved**

Bit1 **H_CH0IntStat**

Indirectly specifies interrupt factors.

Set to "1" when the H_CH0IntStat register includes interrupt factors and the H_CH0IntEnb register bit corresponding to the interrupt factors is enabled.

Bit0 **H_CHrIntStat**

Indirectly specifies interrupt factors.

Set to "1" when the H_CHrIntStat register includes interrupt factors and the H_CHrIntEnb register bit corresponding to the interrupt factors is enabled.

3.1.4 Port0:003h / Port1:203h CPU_IntStat (CPU Interrupt Status)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	003h	CPU_IntStat	R (W)	7: RAM_RdCmp	0: None	1: RAM Read Complete	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
			R (W)	1: DMA_CountUp	0: None	1: DMA Counter Overflow	
			R (W)	0: DMA_Cmp	0: None	1: DMA Complete	

Indicates CPU interface-related interrupts.

All bits can be set to "1" to clear the interrupt factors.

Bit7 RAM_RdCmp

Directly specifies interrupt factors.

Set to "1" when data is read from the RAM by the RAM_Rd function and the RAM_Rd_XX data is enabled.

Bit6-2 Reserved

Bit1 DMA_CountUp

Directly specifies interrupt factors.

Set to "1" when the DMA_Count_HH,HL,LH,LL values overflow while the transfer mode operates in free-run mode. The DMA_Count_HH,HL,LH,LL values return to 0, and DMA operation continues.

Bit0 DMA_Cmp

Directly specifies interrupt factors.

Set to "1" when the DMA transfer is stopped or if the specified transfer quantity has been sent and end processing is complete.

3.1.5 Port0:004h / Port1:204h FIFO_IntStat (FIFO Interrupt Status)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	004h	FIFO_IntStat	R(W)	7: FIFO_DMA_Cmp	0: None	1: DMA Complete on FIFO	00h
				6:	0:	1	
				5:	0:	1	
				4:	0:	1	
				3:	0:	1:	
			R (W)	2: FIFO_NotEmpty	0: None	1: FIFO NotEmpty	
			R (W)	1: FIFO_Full	0: None	1: FIFO Full	
			R (W)	0: FIFO_Empty	0: None	1: FIFO Empty	

Indicates FIFO-related interrupt status.

All bits can be set to "1" to clear the interrupt factors.

Bit7 FIFO_DMA_Cmp

Directly specifies interrupt factors.

Set to "1" if the FIFO becomes empty after DMA transfer ends when the endpoint joined to the DMA is in the IN direction while HostDeviceSel.HOSTxDEVICE=0. Set to "1" if DMA transfer ends when the endpoint joined to the DMA is in the OUT direction.

Set to "1" if the FIFO becomes empty after DMA transfer ends when the channel joined to the DMA is in the OUT direction while HostDeviceSel.HOSTxDEVICE=1. Set to "1" if DMA transfer ends when the channel joined to the DMA is in the IN direction.

Bit6-3 Reserved

Bit2 FIFO_NotEmpty

Directly specifies interrupt factors.

Set to "1" if the FIFO area for the corresponding area contains data when the AREAn{n=0-5}Join_0.JoinFIFO_Stat bit is set to "1."

Bit1 FIFO_Full

Directly specifies interrupt factors.

Set to "1" if the FIFO area for the corresponding area becomes full when the AREAn{n=0-5}Join_0.JoinFIFO_Stat bit is set to "1."

Bit0 FIFO_Empty

Directly specifies interrupt factors.

Set to "1" if the FIFO area for the corresponding area becomes empty when the AREAn{n=0-5}Join_0.JoinFIFO_Stat bit is set to "1."

3.1.6 Port0:006h / Port1:206h *RootIntStat (Root Interrupt Status)*

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	006h	***RootIntStat***		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
			R	1: ***Port1MainIntStat***	0: None	1: MainIntStat on Port1	
			R	0: ***Port0MainIntStat***	0: None	1: MainIntStat on Port0	

This register is a common register mirrored at Port 0 and Port 1.

It indicates the port generating the interrupt when the XINT terminal is asserted.

Bit7-2 Reserved

Bit1 *Port1MainIntStat*

Indirectly specifies interrupt factors. Indicates that Port 1 is the interrupt source.

Bit0 *Port0MainIntStat*

Indirectly specifies interrupt factors. Indicates that Port 0 is the interrupt source.

3.1.7 Port0:008h / Port1:208h *MainIntEnb (Main Interrupt Enable)*

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	008h	***MainIntEnb***	R/W	7: ***EnUSB_DeviceIntStat***	0: Disable	1: Enable	00h
			R/W	6: ***EnUSB_HostIntStat***	0: Disable	1: Enable	
			R/W	5: EnCPU_IntStat	0: Disable	1: Enable	
			R/W	4: EnFIFO_IntStat	0: Disable	1: Enable	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1	
			R/W	0: ***EnFinishedPM***	0: Disable	1: Enable	

This register permits or prohibits interrupt signal (XINT) assertion by MainIntStat register interrupt factors.

Interrupts are permitted if the corresponding bit is set to "1."

The EnUSB_DeviceIntStat, EnUSB_HostIntStat, and EnFinishedPM bits are enabled even in SLEEP state.

The EnUSB_DeviceIntStat bit is not defined for Port 1.

3.1.8 Port0:009h / Port1:N/A USB_DeviceIntEnb (Device Interrupt Enable)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	009h	***USB_DeviceIntEnb***	R/W	7: ***EnVBUS_Changed***	0: Disable	1: Enable	00h
				6:	0:	1:	
			R/W	5: ***EnD_SIE_IntStat***	0: Disable	1: Enable	
			R/W	4: EnD_BulkIntStat	0: Disable	1: Enable	
			R/W	3: EnRcvEP0SETUP	0: Disable	1: Enable	
				2:	0:	1:	
			R/W	1: EnD_EP0IntStat	0: Disable	1: Enable	
			R/W	0: EnD_EPrIntStat	0: Disable	1: Enable	

This register is not defined for Port 1.

Permits or prohibits MainIntStat register USB_DeviceIntStat bit assertion by USB_DeviceIntStat register interrupt factors.

The EnVBUS_Changed and EnD_SIE_IntStat bits are enabled even in SLEEP state.

3.1.9 Port0:00Ah / Port1:20Ah *USB_HostIntEnb (Host Interrupt Enable)*

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	00Ah	***USB_HostIntEnb***	R/W	7: ***EnVBUS_Err***	0: Disable	1: Enable	00h
			R/W	6: ***EnLineStateChanged***	0: Disable	1: Enable	
			R/W	5: EnH_SIE_IntStat1	0: Disable	1: Enable	
			R/W	4: EnH_SIE_IntStat0	0: Disable	1: Enable	
			R/W	3: EnH_FrameIntStat	0: Disable	1: Enable	
			R/W	2:	0:	1:	
			R/W	1: EnH_CH0IntStat	0: Disable	1: Enable	
			R/W	0: EnH_CHrIntStat	0: Disable	1: Enable	

Permits or prohibits MainIntStat register USB_HostIntStat bit assertion by USB_HostIntStat register interrupt factors.

The EnVBUS_Err bit and EnLineStateChanged bit are enabled even in SLEEP state.

3.1.10 Port0:00Bh / Port1:20Bh CPU_IntEnb (CPU Interrupt Enable)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	00Bh	CPU_IntEnb	R/W	7: EnRAM_RdCmp	0: Disable	1: Enable	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
			R/W	1: EnDMA_CountUp	0: Disable	1: Enable	
			R/W	0: EnDMA_Cmp	0: Disable	1: Enable	

Permits or prohibits MainIntStat register CPU_IntStat bit assertion by CPU_IntStat register interrupt factors.

3.1.11 Port0:00Ch / Port1:20Ch FIFO_IntEnb (FIFO Interrupt Enable)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	00Ch	FIFO_IntEnb	R/W	7: EnFIFO_DMA_Cmp	0: Disable	1: Enable	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
			R/W	2: EnFIFO_NotEmpty	0: Disable	1: Enable	
			R/W	1: EnFIFO_Full	0: Disable	1: Enable	
			R/W	0: EnFIFO_Empty	0: Disable	1: Enable	

Permits or prohibits MainIntStat register FIFO_IntStat bit assertion by FIFO_IntStat register interrupt factors.

3.1.12 Port0:010h / Port1:210h *RevisionNum (Revision Number)*

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Device / Host	010h	***RevisionNum***	R	7: ***RevisionNum [7]***	Revision Number	08h
				6: ***RevisionNum [6]***		
				5: ***RevisionNum [5]***		
				4: ***RevisionNum [4]***		
				3: ***RevisionNum [3]***		
				2: ***RevisionNum [2]***		
				1: ***RevisionNum [1]***		
				0: ***RevisionNum [0]***		

Indicates the LSI revision number. This register can be accessed even in SLEEP state.

The revision number for the current specifications is 0x08.

3.1.13 Port0:011h / Port1:211h *ChipReset (Chip Reset)*

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	011h	***ChipReset***	R/W	7: ***ResetMTM***	0: None	1: MTM Reset	80h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
			W	0: ***AllReset***	0: None	1: Reset	

This resets the LSI.

It can be accessed even in SLEEP state.

Bit7 ***ResetMTM***

Setting this bit to "1" initializes the LSI transceiver macro (MTM).

Clear this bit to "0" to cancel the reset.

Bit6-1 **Reserved**

Bit0 ***AllReset***

This initializes all circuits except the CPUIF_MODE register. Soft resetting by this register will not result in a switch to the uninitialized period.

Note that this register should only be written to for resets.

Writing to this register in contravention of the AC spec except for resets will cause malfunctions.

3.1.14 Port0:012h / Port1:212h *PM_Control (Power Management Control)*

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	012h	***PM_Control***	R/W	7: ***GoSLEEP***	0: Do nothing	1: Go to SLEEP	00h
			R/W	6: ***GoACTIVE***	0: Do nothing	1: Go to ACTIVE	
			W	5: ***GoCPU_Cut***	0: Do nothing	1: Go to CPU Cut mode	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
			R	1: ***PM_State[1]***	Power Management State 00: SLEEP, 01: (SNOOZE), 11: ACTIVE		
				0: ***PM_State[0]***			

This sets operations related to LSI power management.

The register is enabled even in SLEEP state.

Bit7 ***GoSLEEP***

This initiates the switch from ACTIVE state to SLEEP state.

Setting this bit to "1" while in ACTIVE state first stops the PLL oscillation before stopping the oscillator oscillation and switching to SLEEP state.

This bit is automatically cleared as soon as the switch is complete, regardless of the state switched from. The MainIntStat.FinishedPM bit is set simultaneously.

Bit6 ***GoACTIVE***

This initiates the switch from SLEEP state to ACTIVE state.

Setting this bit to "1" while in SLEEP state starts oscillator oscillations, starts PLL oscillations after the oscillator oscillation stabilization time elapses (time set in WakeupTim_H, L), then switches to ACTIVE state after the PLL480 oscillation stabilization time (approximately 250 µs) has elapsed.

This bit is automatically cleared as soon as the switch is complete, regardless of the state switched from. The MainIntStat.FinishedPM bit is set simultaneously.

Bit5 ***GoCPU_Cut***

Setting this bit while in SLEEP state switches to CPU_Cut mode, further reducing current consumption.

Setting this bit after switching fully to SLEEP state initially switches off the CPU interface input terminals except for the XCS and CA9 terminal as the CPU writing state ends. This minimizes unnecessary power consumption, since the CPU interface initial driver is off if an attempt is made to operate signal lines other than the XCS and CA9 terminal.

To reset from this mode, dummy-read this register. Note, however, that read data will not be confirmed for data read out here. Since this resetting operation is performed simultaneously with the completion of dummy reading, access should ensure that XCS is negated temporarily ("L" to "H"). Note that this type of operation is performed for regular CPUs when accessing address space other than this LSI.

* Although this LSI is masked to prevent the assertion of a XINT signal in SLEEP state due to interrupt status (synchronous status) unavailable during SLEEP state, the following processing should be performed via the firmware to prevent assertion of the XINT terminal when the SLEEP state is reset.

<Before starting SLEEP>

Process and clear synchronous status (-IntStat).

Disable synchronous status (-IntEnb).

<After clearing SLEEP (after starting ACTIVE)>

Clear synchronous status (-IntStat).

Enable synchronous status (-IntEnb).

Bit4-2 Reserved

Bit1-0 *PM_State [1:0]*

Indicates the power management state.

00: SLEEP state (OSC off, PLL off)

01: SLEEP state* (OSC on, PLL off)

11: ACTIVE state (OSC on, PLL on)

* PM_State will be 0b01 if the referenced port is in SLEEP state and the other ports are in ACTIVE state.

This bit should not be referenced, since it varies in sequence to the corresponding state from when the GoSLEEP or GoACTIVE bits are set until when MainIntStat.FinishedPM interrupt status is set.

3.1.15 Port0:014h-015h / Port1:214h-215h *WakeupTim_H,L (Wakeup Time High, Low)*

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Device / Host	014h -015h	***WakeupTim_H*** ***WakeupTim_L***	R/W	***WakeupTim [15:0]***	Wakeup Time [15:0]	0000h

These registers are common registers mirrored at Port 0 and Port 1.

They set the oscillator stabilization time when reverting from SLEEP to SNOOZE state. This register can be accessed even in SLEEP state.

Writing "1" to the PM_Control.GoACTIVE bit in SLEEP state enables the oscillator cell and starts oscillator oscillation. This WakeupTim_H,L setting is loaded into the counter here, and the OSC rising starts the countdown. Once the countdown ends, the internal OSCCLK gate is opened, and CLK is sent to the PLL and other circuits.

The oscillator stabilization time depends on various factors, including the oscillator, oscillator cell, circuit board, and load capacity. The internal SCLK must be stabilized to 60 MHz ±10% within 5.1 ms after USB RESET detection if dropping to SLEEP state for USB SUSPEND during device operation.

The total time for oscillator stabilization time + PLL stabilization time (within 250 µs) must therefore not exceed 5.1 ms.

3.1.16 Port0:016h / Port1:216h *H_USB_Control (Host USB Control)*

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	016h	***H_USB_Control***	R/W	7: ***VBUS_Enb***	0: Disable	1: Enable	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
				0:	0:	1:	

This sets host-related operations.

This register is enabled even in SLEEP state.

Bit7 ***VBUS_Enb***

Sets the VBUSEN terminal (output) state. The default is Low level.

Bit6-0 **Reserved**

3.1.17 Port0:017h / Port1:217h *H_XcvrControl (Host Xcvr Control)*

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	017h	***H_XcvrControl***	R/W	7: ***TermSelect***	0: HS Termination	1: FS Termination	91h
			R/W	6: ***RemoveRPD***	0: RPD ON	1: RPD OFF	
			R/W	4: ***XcvrSelect[1]***	XcvrSelect[1:0]		
			R/W	4: ***XcvrSelect[0]***			
				3:	0:	1:	
				2:	0:	1:	
			R/W	1: ***OpMode [1]***	OpMode [1:0]		
				0: ***OpMode [0]***			

This sets the host transceiver macro.

This register is enabled even in SLEEP state.

Bit7 ***TermSelect***

Selects and enables either HS or FS termination.

This bit should not be set manually, since it is automatically set by hardware when H_NogoControl_0.AutoMode is set.

Bit6 ***RemoveRPD***

This turns on/off the DP and DM internal pull-down resistors forming the host data line.

0: RPD ON

1: RPD OFF

This bit must be used set to “0” (ON) for USB host operations (including SUSPEND). Note that changing the bit from “0” will alter host data line characteristics and may cause USB malfunctions.

Bit5-4 ***XcvrSelect[1:0]***

Selects and enables the HS, FS, or LS transceiver.

00: High Speed transceiver

01: Full Speed transceiver

10: Reserved

11: Low Speed transceiver

This bit should not be set manually. It is automatically set by the hardware when H_NogoControl_0.AutoMode is set.

Bit3-2 **Reserved**

Bit1-0 ***OpMode[1:0]***

This sets the MTM operating mode.

This bit should not be set manually. It is automatically set by the hardware when H_NogoControl_0.AutoMode is set.

However, to detect signal line change status, refer to "1.2.2.2 Signal Line Change Status Usage" when this bit needs setting.

OpMode		
00	"Normal Operation"	Normal usage state
01	"2Non-Driving"	Non-used state
10	"Disable Bitstuffing and NRZI encoding"	Bitstuffing and NRZI encoding functions disabled in normal usage state
11	"Power-Down"	Only single-end receiver used

3.1.18 Port0:018h / Port1:N/A *D_USB_Status (Device USB Status)*

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	018h	***D_USB_Status***	R	7: ***VBUS***	0: VBUS = L	1: VBUS = H	XXh
			R/W	6: FSxHS	0: HS mode	1: FS mode	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
			R	1: ***LineState [1]***	Line State [1:0]		
				0***: LineState [0]***			

This register is not defined for Port 1.

Indicates the device-related status.

Bit7 ***VBUS***

Indicates the VBUS terminal status. This bit is enabled even in SLEEP state.

Bit6 **FSxHS**

Indicates the current operating mode. It is automatically set when "HS Detection Handshake" (see "Functions") is run using the D_NegoControl.GoChirp bit. Although writing this bit allows forced switching of the operating mode, this bit should be used only when necessary to switch the operating mode without using "HS Detection Handshake," as when running simulations.

Set to "FS(1)" when a cable is connected.

This bit can be read from even in SLEEP state and can be written to in ACTIVE state.

Bit5-2 **Reserved**

Bit1-0 ***LineState [1:0]***

Indicates the USB cable signal status. This bit is enabled even in SLEEP state.

The DP/DM FS receiver received value is indicated if the XcvrSelect bit is "1" (with FS transceiver selected) when the D_XcvrControl register TermSelect bit is "1" (with FS termination selected). The HS receiver received value is indicated if XcvrSelect is "0" (with HS transceiver selected).

The USB bus activity is indicated when TermSelect is "0."

LineState		
TermSelect	**DP / DM**	**LineState [1:0]**
0	Don't Care	Bus activity
1	SE0	0b00
1	J	0b01
1	K	0b10
1	SE1	0b11

3.1.19 Port0:019h / Port1:219h *H_USB_Status (Host USB Status)*

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	019h	***H_USB_Status***	R	7: ***VBUS_State***	0: VBUSFLG = High	1: VBUSFLG= Low	XXh
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
			R	1: ***LineState [1]***	Line State [1:0]		
				0***: LineState [0]***			

Indicates the host-related status.

This register is enabled even in SLEEP state.

Bit7 ***VBUS_State***

Indicates the VBUSFLG terminal status.

Bit6-2 **Reserved**

Bit1-0 ***LineState [1:0]***

Indicates the USB cable signal status.

The DP/DM FS receiver received value is indicated if the H_XcvrControl register XcvrSelect[1:0] is “01” (with FS transceiver selected). The DP/DM LS receiver received value is indicated if it is “11” (with LS transceiver selected).

The USB bus activity is indicated when XcvrSelect[1:0] is “00” (with HS transceiver selected).

LineState		
XcvrSelect[1:0]	**DP / DM**	**LineState [1:0]**
00	Don’t Care	Bus activity Activity present: 0b01 No activity: 0b00
01 or 11	SE0	0b00
01 or 11	J	0b01
01 or 11	K	0b10
01 or 11	SE1	0b11

Note: The XcvrSelect[1:0] = “10” code is reserved and does not guarantee the operation.

3.1.20 Port0:01Bh / Port1 MTM_Config (Multi Transceiver Macro Config)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	01Bh	MTM_Config		7:	0:	1:	XXh
				6:	0:	1:	
			R/W	5: MTM_SlopeValue[1]	MTM Slope Value[1:0]		
			R/W	4: MTM_SlopeValue[0]			
				3:	0:	1:	
				2:	0:	1:	
			R/W	1: MTM_TermValue[1]	MTM Termination Value[1:0]		
			R/W	0: MTM_TermValue[0]			

This register is used to set transceiver macro adjustment values.

Bit7-6 Reserved

Bit5-4 MTM_SlopeValue[1:0]

Sets HS transmitter slew rate to one of four levels.

00: Slow

01: ↑

10: ↓

11: Fast

Bit3-2 Reserved

Bit1-0 MTM_TermValue[1:0]

Sets HS transmission route termination to one of four levels.

00: High

01: ↑

10: ↓

11: Low

3.1.21 Port0:01Fh / Port1:21Fh *HostDeviceSel (Host Device Select)*

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	01Fh	***HostDeviceSel***		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
			R/W	0: ***HOSTxDEVICE***	0: Device mode	1: Host mode	

Sets the USB device mode or host mode.

This register is enabled even in SLEEP mode.

Bit7-1 Reserved

Bit0 *HOSTxDEVICE* (Port0), 1'b1(Port1)

Sets the USB device mode or host mode.

0: USB device mode

1: USB host mode

This bit operates the internal system clock. If HOST x DEVICE is "0," i.e., in device mode, the system clock is fed to the common block and USB device block, and provision of the system clock to the USB host block is suspended. If HOST x DEVICE is "1," i.e., in host mode, the system clock is fed to the common block and USB host block, and provision of the system clock to the USB device block is suspended.

3.1.22 Port0:020h / Port1:220h FIFO_Rd_0 (FIFO Read 0)

3.1.23 Port0:021h / Port1:221h FIFO_Rd_1 (FIFO Read 1)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Device / Host	020h	FIFO_Rd_0	R	7: FIFO_Rd_0 [7]	FIFO Read	XXh
				6: FIFO_Rd_0 [6]		
				5: FIFO_Rd_0 [5]		
				4: FIFO_Rd_0 [4]		
				3: FIFO_Rd_0 [3]		
				2: FIFO_Rd_0 [2]		
				1: FIFO_Rd_0 [1]		
				0: FIFO_Rd_0 [0]		

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Device / Host	021h	FIFO_Rd_1	R	7: FIFO_Rd_1 [7]	FIFO Read	XXh
				6: FIFO_Rd_1 [6]		
				5: FIFO_Rd_1 [5]		
				4: FIFO_Rd_1 [4]		
				3: FIFO_Rd_1 [3]		
				2: FIFO_Rd_1 [2]		
				1: FIFO_Rd_1 [1]		
				0: FIFO_Rd_1 [0]		

020h.Bit7-0 FIFO_Rd_0[7:0]

021h.Bit7-0 FIFO_Rd_1[7:0]

This reads the data from the FIFO set by the AREAn{n=0-5}Join_0.JoinCPU_Rd bit.

If this register is read when the FIFO contains byte boundaries, valid data is output to one side only. For detailed information, refer to “FIFO Access Fractional Number Processing.”

To read out FIFO data using this register, the quantity of data that can be read should first be checked using the FIFO_RdRemain_H,L registers.

3.1.24 Port0:022h / Port1:222h FIFO_Wr_0(FIFO Write 0)

3.1.25 Port0:023h / Port1:223h FIFO_Wr_1(FIFO Write 1)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Device / Host	022h	FIFO_Wr_0	W	7: FIFO_Wr_0 [7]	FIFO Write	XXh
				6: FIFO_Wr_0 [6]		
				5: FIFO_Wr_0 [5]		
				4: FIFO_Wr_0 [4]		
				3: FIFO_Wr_0 [3]		
				2: FIFO_Wr_0 [1]		
				1: FIFO_Wr_0 [1]		
				0: FIFO_Wr_0 [0]		

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Device / Host	023h	FIFO_Wr_1	W	7: FIFO_Wr_1 [7]	FIFO Write	XXh
				6: FIFO_Wr_1 [6]		
				5: FIFO_Wr_1 [5]		
				4: FIFO_Wr_1 [4]		
				3: FIFO_Wr_1 [3]		
				2: FIFO_Wr_1 [2]		
				1: FIFO_Wr_1 [1]		
				0: FIFO_Wr_1 [0]		

022h.Bit7-0 FIFO_Wr_0[7:0]

023h.Bit7-0 FIFO_Wr_1[7:0]

This writes the data to the FIFO set by the AREAn{n=0-5}Join_0.JoinCPU_Wr bit.

If this register is written to when the FIFO contains byte boundaries, data is written to one side only. For detailed information, refer to “FIFO Access Fractional Number Processing.”

To write data to the FIFO using this register, the quantity of data that can be written must first be checked using the FIFO_WrRemain_H,L registers.

3.1.26 Port0:024h-025h / Port1:224h-225h FIFO_RdRemain_H,L (FIFO Read Remain High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	024h-025h	FIFO_RdRemain_H,L	R	15: RdRemainValid	0:None	1: Read Remain Valid	00h
				14:	0:	1:	
				13:	0:	1:	
			R	12: RdRemain [12]	Read Remain		
				11: RdRemain [11]			
				10: RdRemain [10]			
				9: RdRemain [9]			
				8: RdRemain [8]			
				7: RdRemain [7]			
				6: RdRemain [6]			
				5: RdRemain [5]			
				4: RdRemain [4]			
				3: RdRemain [3]			
				2: RdRemain [2]			
				1: RdRemain [1]			
				0: RdRemain [0]			

Bit15 RdRemainValid

Set to “1” if a FIFO is joined to the CPU I/F by the AREAn{n=0-5}Join_0.JoinCPU_Rd bit and the FIFO_RdRemain value is valid.

The RdRemain value is invalid if this bit is cleared.

Bit14-13 Reserved

Bit12 -0 RdRemain[12:0]

Indicates the quantity of readable data within the FIFO connected to the CPU I/F by the AREAn{n=0-5}Join_0.JoinCPU_Rd bit. The FIFO_RdRemain_H and FIFO_RdRemain_L registers must both be accessed as a pair to obtain the FIFO readable data quantity.

3.1.27 Port0:026h-027h / Port1:226h-227h FIFO_WrRemain_H,L (FIFO Write Remain High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	026h-027h	WrRemain_H,L		15:	0:	1:	00h
				14:	0:	1:	
				13:	0:	1:	
			R	12: WrRemain [12]	FIFO Write Remain		
				11: WrRemain [11]			
				10: WrRemain [10]			
				9: WrRemain [9]			
				8: WrRemain [8]			
				7: WrRemain [7]			
				6: WrRemain [6]			
				5: WrRemain [5]			
				4: WrRemain [4]			
				3: WrRemain [3]			
				2: WrRemain [2]			
				1: WrRemain [1]			
				0: WrRemain [0]			

Bit15-13 Reserved

Bit12 -0 WrRemain[12:0]

Indicates the free space in the FIFO connected to the CPU I/F by the AREAn{n=0-5}Join_0.JoinCPU_Wr bit. Note that the FIFO free capacity cannot be accurately checked immediately after writing to the FIFO. Allow at least one CPU cycle before checking the FIFO free space. The FIFO_WrRemain_H and FIFO_WrRemain_L registers must be accessed as a pair to obtain FIFO free capacity.

3.1.28 Port0:028h / Port1:228h FIFO_ByteRd(FIFO Byte Read)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Device / Host	028h	FIFO_ByteRd	R	7: FIFO_ByteRd [7]	FIFO Byte Read	XXh
				6: FIFO_ByteRd [6]		
				5: FIFO_ByteRd [5]		
				4: FIFO_ByteRd [4]		
				3: FIFO_ByteRd [3]		
				2: FIFO_ByteRd [2]		
				1: FIFO_ByteRd [1]		
				0: FIFO_ByteRd [0]		

Bit7-0 FIFO_ByteRd[7:0]

Enables reading of data in byte units from the FIFO set by the AREAn{n=0-5}Join_0.JoinCPU_Rd bit. To read out FIFO data using this register, the quantity of data that can be read must first be checked using the FIFO_RdRemain_H,L registers.

3.1.29 Port0:02Ah / Port1:22Ah FIFO_ByteWr(FIFO Byte Write)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Device / Host	02Ah	FIFO_ByteWr	W	7: FIFO_ByteWr [7]	FIFO Byte Write	XXh
				6: FIFO_ByteWr [6]		
				5: FIFO_ByteWr [5]		
				4: FIFO_ByteWr [4]		
				3: FIFO_ByteWr [3]		
				2: FIFO_ByteWr [2]		
				1: FIFO_ByteWr [1]		
				0: FIFO_ByteWr [0]		

Bit7-0 FIFO_ByteWr[7:0]

Enables writing of data in byte units to the FIFO set by the AREAn{n=0-5}Join_0.JoinCPU_Wr bit. To write data to the FIFO using this register, the quantity of data that can be written must first be checked using the FIFO_WrRemain_H,L registers.

3.1.30 Port0:030h-031h / Port1:230h-231h RAM_RdAdrs_H,L (RAM Read Address High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	030h-031h	RAM_RdAdrs_H,L		15:	0:	1:	00h
				14:	0:	1:	
				13:	0:	1:	
			R/W	12: RAM_RdAdrs [12]	RAM Read Address		
				11: RAM_RdAdrs [11]			
				10: RAM_RdAdrs [10]			
				9: RAM_RdAdrs [9]			
				8: RAM_RdAdrs [8]			
				7: RAM_RdAdrs [7]			
				6: RAM_RdAdrs [6]			
				5: RAM_RdAdrs [5]			
				4: RAM_RdAdrs [4]			
				3: RAM_RdAdrs [3]			
				2: RAM_RdAdrs [2]			
				1:	0:	1:	
				0:	0:	1:	

Bit15-13 Reserved

Bit12-2 RAM_RdAdrs[12:2]

This sets the initial address for RAM_Rd. After setting this register, set the RAM_RdCount register, then set the RAM_RdControl register bit to activate the RAM_Rd function. The register value depends on the internal operation underway while the RAM_Rd function is operating. For this reason, the register value should not be read until the CPU_IntStat.RAM_RdCmp bit is set after the RAM_RdControl register bit has been set and the RAM_Rd function started. Values cannot be guaranteed if this register is read while the RAM_Rd function is operating. Also note that writing to this register while the RAM_Rd function is operating will result in malfunctions.

Bit1-0 Reserved

3.1.31 Port0:032h / Port1:232h RAM_RdControl (RAM Read Control)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	032h	RAM_RdControl	R/W	7: RAM_GoRdCBW_CSW	0: Do nothing	1: RAM Read CBW_CSW start	00h
			R/W	6: RAM_GoRd	0: Do nothing	1: RAM Read start	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
				0:	0:	1:	

Bit7 RAM_GoRdCBW_CSW

This bit starts the RAM_Rd function to read out data received in the CBW area when the device is operating or CSW area during host operation.

Writing "1" to this bit when the device is operating starts the RAM_Rd function and reads data from the CBW area. When the values of registers RAM_Rd_00 to RAM_Rd_1E become valid, the CPU_IntStat.RAM_RdCmp bit is set to "1," and the bit is automatically cleared.

Writing "1" to this bit during host operation starts the RAM_Rd function and reads data from the CSW area. When the values of registers RAM_Rd_00 to RAM_Rd_0C become valid, the CPU_IntStat.RAM_RdCmp bit is set to "1," and the bit is automatically cleared.

In either case, the RAM_RdAdrs_H,L or RAM_RdCount registers need not be set.

The function for this bit takes priority if it is set concurrently with the RAM_GoRd bit.

Bit6 RAM_GoRd

This bit starts the RAM_Rd function.

Setting the RAM_RdCount register and writing "1" to this bit after setting the initial address for RAM_Rd in the RAM_RdAdrs_H,L registers starts the RAM_Rd function. Once the specified count quantity of data has been read from the initial address specified and the RAM_Rd_xx{xx=00 to 1F} register value is enabled, the CPU_IntStat.RAM_RdCmp bit is set to "1," and the bit is automatically cleared.

The function for the RAM_GoRdCBW_CSW bit takes priority if set concurrently with the RAM_GoRdCBW_CSW bit.

Bit5-0 Reserved

3.1.32 Port0:035h / Port1:235h RAM_RdCount (RAM Read Counter)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	035h	RAM_RdCount		7:	0:	1:	00h
				6:	0:	1:	
			R/W	5: RAM_RdCount [5]	RAM Read Counter		
				4: RAM_RdCount [4]			
				3: RAM_RdCount [3]			
				2: RAM_RdCount [2]			
				1:	0:	1:	
				0:	0:	1:	

Bit7-6 Reserved

Bit5-2 RAM_RdCount [5:2]

This sets the quantity of data to be read in the RAM_Rd_xx{xx=00 to 1F} register using the RAM_Rd function. Start the RAM_Rd function by setting the RAM_RdAdrs_H,L registers, setting this register, then setting the RAM_RdControl register bit. The register value depends on the internal operation underway while the RAM_Rd function is operating. For this reason, the register value should not be read until the CPU_IntStat.RAM_RdCmp bit is set after the RAM_RdControl register bit has been set and the RAM_Rd function started. Values cannot be guaranteed if this register is read while the RAM_Rd function is operating. Also note that writing to this register while the RAM_Rd function is operating will result in malfunctions.

Note that this register can be set to a maximum of 32 bytes. Setting a data quantity exceeding 32 bytes will result in malfunctions.

Bit1-0 Reserved

3.1.33 Port0:038h-039h / Port1:238h-239h RAM_WrAdrs_H,L (RAM Write Address High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	038h-039h	RAM_WrAdrs_H,L		15:	0:	1:	00h
				14:	0:	1:	
				13:	0:	1:	
			R/W	12: RAM_WrAdrs [12]	RAM Write Address		
				11: RAM_WrAdrs [11]			
				10: RAM_WrAdrs [10]			
				9: RAM_WrAdrs [9]			
				8: RAM_WrAdrs [8]			
				7: RAM_WrAdrs [7]			
				6: RAM_WrAdrs [6]			
				5: RAM_WrAdrs [5]			
				4: RAM_WrAdrs [4]			
				3: RAM_WrAdrs [3]			
				2: RAM_WrAdrs [2]			
				1: RAM_WrAdrs [1]			
				0: RAM_WrAdrs [0]			

This specifies the address for writing to RAM using the RAM_WrDoor_0,1 registers.

Bit15-13 Reserved

Bit12-0 RAM_WrAdrs[12:0]

This specifies the address for writing to RAM. The address is incremented by the number of bytes written to the RAM_WrDoor_0,1 registers. Note that RAM_WrAdrs cannot be accurately checked immediately after writing to the RAM_WrDoor_0,1 registers. Allow at least one CPU cycle before checking RAM_WrAdrs. For detailed information on writing data, refer to “RAM_WrDoor_0,1 Registers.”

3.1.34 Port0:03Ah / Port1:23Ah RAM_WrDoor_0 (RAM Write Door 0)

3.1.35 Port0:03Bh / Port1:23Bh RAM_WrDoor_1 (RAM Write Door 1)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Device / Host	03Ah	RAM_WrDoor_0	W	7: RAM_WrDoor_0 [7]	RAM Write Door 0	XXh
				6: RAM_WrDoor_0 [6]		
				5: RAM_WrDoor_0 [5]		
				4: RAM_WrDoor_0 [4]		
				3: RAM_WrDoor_0 [3]		
				2: RAM_WrDoor_0 [2]		
				1: RAM_WrDoor_0 [1]		
				0: RAM_WrDoor_0 [0]		

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Device / Host	03Bh	RAM_WrDoor_1	W	7: RAM_WrDoor_1 [7]	RAM Write Door 1	XXh
				6: RAM_WrDoor_1 [6]		
				5: RAM_WrDoor_1 [5]		
				4: RAM_WrDoor_1 [4]		
				3: RAM_WrDoor_1 [3]		
				2: RAM_WrDoor_1 [2]		
				1: RAM_WrDoor_1 [1]		
				0: RAM_WrDoor_1 [0]		

03Ah.Bit7-0 RAM_WrDoor_0[7:0]

03Bh.Bit7-0 RAM_WrDoor_1[7:0]

This write-only register is the access register when writing to RAM.

Set the initial address for writing RAM data to the RAM_WrAdrs_H,L registers before starting to write data. Subsequently writing to this register writes in sequence and automatically increments RAM_WrAdrs_H,L by the number of bytes written.

Data can be written to the descriptor and CSW areas by the RAM_WrDoor_0,1 registers for a USB device. Data written to the descriptor area by the RAM_WrDoor_0,1 registers can be used repeatedly by the ReplyDescriptor function. This means the data cannot be deleted or overwritten by the descriptor reply function. Note, however, that if the area to which descriptor data is written overlaps an area retained by another endpoint, the data can be overwritten.

For a USB host, data can be written to the CBW areas by the RAM_WrDoor_0,1 registers.

3.1.36 Port0:040h / Port1:240h RAM_Rd_00 (RAM Read 00)
3.1.37 Port0:041h / Port1:241h RAM_Rd_01 (RAM Read 01)
3.1.38 Port0:042h / Port1:242h RAM_Rd_02 (RAM Read 02)
3.1.39 Port0:043h / Port1:243h RAM_Rd_03 (RAM Read 03)
3.1.40 Port0:044h / Port1:244h RAM_Rd_04 (RAM Read 04)
3.1.41 Port0:045h / Port1:245h RAM_Rd_05 (RAM Read 05)
3.1.42 Port0:046h / Port1:246h RAM_Rd_06 (RAM Read 06)
3.1.43 Port0:047h / Port1:247h RAM_Rd_07 (RAM Read 07)
3.1.44 Port0:048h / Port1:248h RAM_Rd_08 (RAM Read 08)
3.1.45 Port0:049h / Port1:249h RAM_Rd_09 (RAM Read 09)
3.1.46 Port0:04Ah / Port1:24Ah RAM_Rd_0A (RAM Read 0A)
3.1.47 Port0:04Bh / Port1:24Bh RAM_Rd_0B (RAM Read 0B)
3.1.48 Port0:04Ch / Port1:24Ch RAM_Rd_0C (RAM Read 0C)
3.1.49 Port0:04Dh / Port1:24Dh RAM_Rd_0D (RAM Read 0D)
3.1.50 Port0:04Eh / Port1:24Eh RAM_Rd_0E (RAM Read 0E)
3.1.51 Port0:04Fh / Port1:24Fh RAM_Rd_0F (RAM Read 0F)
3.1.52 Port0:050h / Port1:250h RAM_Rd_10 (RAM Read 10)
3.1.53 Port0:051h / Port1:251h RAM_Rd_11 (RAM Read 11)
3.1.54 Port0:052h / Port1:252h RAM_Rd_12 (RAM Read 12)
3.1.55 Port0:053h / Port1:253h RAM_Rd_13 (RAM Read 13)
3.1.56 Port0:054h / Port1:254h RAM_Rd_14 (RAM Read 14)
3.1.57 Port0:055h / Port1:255h RAM_Rd_15 (RAM Read 15)
3.1.58 Port0:056h / Port1:256h RAM_Rd_16 (RAM Read 16)
3.1.59 Port0:057h / Port1:257h RAM_Rd_17 (RAM Read 17)
3.1.60 Port0:058h / Port1:258h RAM_Rd_18 (RAM Read 18)
3.1.61 Port0:059h / Port1:259h RAM_Rd_19 (RAM Read 19)
3.1.62 Port0:05Ah / Port1:25Ah RAM_Rd_1A (RAM Read 1A)
3.1.63 Port0:05Bh / Port1:25Bh RAM_Rd_1B (RAM Read 1B)
3.1.64 Port0:05Ch / Port1:25Ch RAM_Rd_1C (RAM Read 1C)
3.1.65 Port0:05Dh / Port1:25Dh RAM_Rd_1D (RAM Read 1D)
3.1.66 Port0:05Eh / Port1:25Eh RAM_Rd_1E (RAM Read 1E)
3.1.67 Port0:05Fh / Port1:25Fh RAM_Rd_1F (RAM Read 1F)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Device / Host	040h -05Fh	RAM_Rd_00 ~ RAM_Rd_1F	R	7: RAM_Rd_xx [7]	RAM Read	00h
				6: RAM_Rd_xx [6]		
				5: RAM_Rd_xx [5]		
				4: RAM_Rd_xx [4]		
				3: RAM_Rd_xx [3]		
				2: RAM_Rd_xx [2]		
				1: RAM_Rd_xx [1]		
				0: RAM_Rd_xx [0]		

040h-05Fh.Bit7-0 RAM_Rd_xx[7:0]

This register stores data read from RAM using the RAM_Rd function. Start the RAM_Rd function by setting RAM_RdAdrs_H,L and RAM_RdCount registers and using the RAM_RdControl register bit. The FIFO_IntStat.RAM_RdCmp bit is set to “1” when this register value becomes valid. If the value set in the RAM_RdCount register is less than 32 bytes, the data read from RAM is stored in sequence from RAM_Rd_00. Register values above the count set in the RAM_RdCount register (e.g., RAM_Rd_10 to RAM_Rd_1F when the count setting is “16”) become invalid.

3.1.68 Port0:061h / Port1:261h DMA_Config (DMA Config)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	061h	DMA_Config	R/W	7: FreeRun	0: Count mode	1: FreeRun mode	00h
			R/W	6: DMA_Mode	0: Normal mode	1: Address Decode mode	
				5:	0:	1:	
				4:	0:	1:	
			R/W	3: ActiveDMA	0: DMA Inactive	1: DMA Active	
				2:	0:	1:	
			R/W	1: ReqAssertCount [1]	Request Assert Count		
				0: ReqAssertCount [0]			

This sets the DMA operating mode.

Bit7 FreeRun

Sets the DMA operating mode.

0: Count mode

1: Free-run mode

Bit6 DMA_Mode

Sets the DMA mode.

0: Operates the DMA using the XDACK from the host as an acknowledge.

1: Operates the DMA using the access to the DMA_RdData/DMA_WrData register from the host as an acknowledge.

Bit5-4 Reserved

Bit3 ActiveDMA

Enables DMA DACK.

0: DMA (DACK) disabled

1: DMA (DACK) enabled

Bit2 Reserved

Bit1-0 ReqAssertCount [1:0]

This bit sets the REQ assert count option to support CPU burst-reading and burst-writing.

Sets the XDREQ assert count (number of transfer bytes). XDREQ is asserted if the free space for writing or data for reading in the FIFO is at least as equal to the assert count set. Once the DMA transfer ends for the assert count set, XDREQ is first negated, then asserted again once the free space or data is confirmed as being at least as equal to the assert count. In other words, transfers for the assert count number set are guaranteed for a single XDREQ assert.

However, if count mode is set and the remaining count for DMA_Count_HH,HL,LH,LL is smaller than the assert count set, the DMA_Count_HH,HL,LH,LL count takes priority. XDREQ is asserted if the free space or data in the FIFO is at least equal to the DMA_Count_HH,HL,LH,LL count.

The table below shows the correspondence between DMA_Count_HH,HL,LH,LL (Count in the table), ReqAssertCount (Req in the table), and FIFO free space/data (Ready in the table), with the XDREQ signal and transferable quantity.

The DMA_Count_HH,HL,LH,LL remaining count must be at least "1."

	Count>=Req		**Count<Req**	
	Ready>=Req	**Ready<Req**	**Ready>=Count**	**Ready<Count**
XDREQ	Assert	Negate	Assert	Negate
Transferable quantity	Req	-	Req	-

ReqAssertCount [1:0]	**Mode**	
	16bit mode	**8bit mode**
0b00	Normal	Normal
0b01	16Byte(8Count)	16Byte(16Count)
0b10	32Byte(16Count)	32Byte(32Count)
0b11	64Byte(32Count)	64Byte(64Count)

The REQ assert count option is not used for the 00 (Normal) setting.

3.1.69 Port0:062h / Port1:262h DMA_Control (DMA Control)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	062h	DMA_Control	R	7: DMA_Running	0: DMA is not running	1: DMA is running	00h
				6:	0:	1:	
				5:	0:	1:	
			W	4: CounterClr	0: Do nothing	1: Clear DMA counter	
			R/W	3: Dir	0: CPU-IF → FIFO RAM	1: CPU-IF ← FIFO RAM	
				2:	0:	1:	
			W	1: DMA_Stop	0: Do nothing	1: Finish DMA	
			W	0: DMA_Go	0: Do nothing	1: Start DMA	

This controls DMA and indicates the status.

Bit7 DMA_Running

This bit is set to "1" during DMA transfer. The AREAn{n=0-5}Join_0.JoinDMA bit cannot be changed while this bit is "1."

Bit6-5 Reserved

Bit4 CounterClr

Setting this bit to "1" clears the DMA_Count_HH,HL,LH,LL registers to 0x00. Writing to this bit is ignored if the DMA_Running bit is set to "1."

Bit3 Dir

This sets the DMA transfer direction.

0: CPU-IF → FIFO RAM (DMA writing)

1: CPU-IF ← FIFO RAM (DMA reading)

Bit2 Reserved

Bit1 DMA_Stop

Setting this bit to "1" ends the DMA transfer. Stopping the DMA transfer clears the DMA_Running bit to "0." The CPU_IntStat register DMA_Cmp bit is also set to "1." To restart the DMA transfer, check the DMA_Running bit or DMA_Cmp bit, then wait for the DMA to end.

Bit0 DMA_Go

Setting this bit to "1" starts the DMA transfer.

3.1.70 Port0:064h-065h / Port1:264h-265h DMA_Remain_H,L (DMA FIFO Remain High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	064h-065h	DMA_Remain_H,L		15:	0:	1:	0000h
				14:	0:	1:	
				13:	0:	1:	
			R	12: DMA_Remain [12]	DMA FIFO Remain		
				11: DMA_Remain [11]			
				10: DMA_Remain [10]			
				9: DMA_Remain [9]			
				8: DMA_Remain [8]			
				7: DMA_Remain [7]			
				6: DMA_Remain [6]			
				5: DMA_Remain [5]			
				4: DMA_Remain [4]			
				3: DMA_Remain [3]			
				2: DMA_Remain [2]			
				1: DMA_Remain [1]			
				0: DMA_Remain [0]			

Bit15-13 Reserved

Bit12-0 DMA_Remain[12:0]

For reading, this indicates the quantity of data remaining in the FIFO connected to the DMA by the AREAn{n=0-5}Join_0.JoinDMA bit.

For writing, this indicates the quantity of free space in the FIFO connected to the DMA by the AREAn{n=0-5}Join_0.JoinDMA bit.

The correct FIFO free space cannot be checked using this register immediately after a DMA write. Allow at least one CPU cycle before checking the FIFO free space.

3.1.71 Port0:068h-069h / Port1:268h-269h DMA_Count_HH,HL (DMA Transfer Byte Counter High/High, High/Low)

3.1.72 Port0:06Ah-06Bh / Port1:26Ah-26Bh DMA_Count_LH,LL (DMA Transfer Byte Counter Low/High, Low/Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Device / Host	068h-069h	DMA_Count_HH,HL	R/W	15: DMA_Count [31]	DMA Transfer Byte Counter High	0000h
				14: DMA_Count [30]		
				13: DMA_Count [29]		
				12: DMA_Count [28]		
				11: DMA_Count [27]		
				10: DMA_Count [26]		
				9: DMA_Count [25]		
				8: DMA_Count [24]		
				7: DMA_Count [23]		
				6: DMA_Count [22]		
				5: DMA_Count [21]		
				4: DMA_Count [20]		
				3: DMA_Count [19]		
				2: DMA_Count [18]		
				1: DMA_Count [17]		
				0: DMA_Count [16]		

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Device / Host	06Ah-06Bh	DMA_Count_LH,LL	R/W	15: DMA_Count [15]	DMA Transfer Byte Counter Low	0000h
				14: DMA_Count [14]		
				13: DMA_Count [13]		
				12: DMA_Count [12]		
				11: DMA_Count [11]		
				10: DMA_Count [10]		
				9: DMA_Count [9]		
				8: DMA_Count [8]		
				7: DMA_Count [7]		
				6: DMA_Count [6]		
				5: DMA_Count [5]		
				4: DMA_Count [4]		
				3: DMA_Count [3]		
				2: DMA_Count [2]		
				1: DMA_Count [1]		
				0: DMA_Count [0]		

These set the DMA transfer data length in bytes when in Counter mode. Up to 0xFFFF_FFFF bytes can be set. The value set is used to start the countdown. After setting the transfer quantities for these registers, set the DMA_Control.DMA_Go bit to “1” and start the DMA transfer. The DMA transfer ends when the number of transfer bytes set in these registers has been transferred. In Free-run mode, the value set is used to start the count-up. The CPU_IntStat register DMA_CountUp bit is set to “1” if the DMA_Count_HH,HL,LH,LL register values overflow. The count continues even after the overflow. The DMA transfer quantity can be checked in this mode.

The count cannot be accurately checked using these registers immediately after DMA writing. Allow at least one CPU cycle before checking the count. These registers should be read out in the sequence of DMA_Count_HH,HL followed by DMA_Count_LH,LL.

3.1.73 Port0:06Ch / Port1:26Ch DMA_RdData_0 (DMA Read Data 0)

3.1.74 Port0:06Dh / Port1:26Dh DMA_RdData_1 (DMA Read Data 1)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Device / Host	06Ch	DMA_RdData_0	R	7: DMA_RdData_0 [7]	DMA Read Data 0	XXh
				6: DMA_RdData_0 [6]		
				5: DMA_RdData_0 [5]		
				4: DMA_RdData_0 [4]		
				3: DMA_RdData_0 [3]		
				2: DMA_RdData_0 [2]		
				1: DMA_RdData_0 [1]		
				0: DMA_RdData_0 [0]		

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Device / Host	06Dh	DMA_RdData_1	R	7: DMA_RdData_1 [7]	DMA Read Data 1	XXh
				6: DMA_RdData_1 [6]		
				5: DMA_RdData_1 [5]		
				4: DMA_RdData_1 [4]		
				3: DMA_RdData_1 [3]		
				2: DMA_RdData_1 [2]		
				1: DMA_RdData_1 [1]		
				0: DMA_RdData_1 [0]		

06Ch.Bit7-0 DMA_RdData_0[7:0]

06Dh.Bit7-0 DMA_RdData_1[7:0]

Accessing these registers when the DMA_Config.DMA_Mode bit is set to “1” enables data to be read from the FIFO connected to the DMA by the AREAn{n=0-5}Join_0.JoinDMA bit. Here, the DMA_Control.Dir bit must be set to DMA read.

DMA access is possible in the same way, whether DMA_RdData_0 or DMA_RdData_1 is accessed, when operating in 8-bit mode.

3.1.75 Port0:06Eh / Port1:26Eh DMA_WrData_0 (DMA Write Data 0)

3.1.76 Port0:06Fh / Port1:26Fh DMA_WrData_1 (DMA Write Data 1)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Device / Host	06Eh	DMA_WrData_0	W	7: DMA_WrData_0[7]	DMA Write Data 0	XXh
				6: DMA_WrData_0[6]		
				5: DMA_WrData_0[5]		
				4: DMA_WrData_0[4]		
				3: DMA_WrData_0[3]		
				2: DMA_WrData_0[2]		
				1: DMA_WrData_0[1]		
				0: DMA_WrData_0[0]		

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Device / Host	06Fh	DMA_WrData_1	W	7: DMA_WrData_1[7]	DMA Write Data 1	XXh
				6: DMA_WrData_1[6]		
				5: DMA_WrData_1[5]		
				4: DMA_WrData_1[4]		
				3: DMA_WrData_1[3]		
				2: DMA_WrData_1[2]		
				1: DMA_WrData_1[1]		
				0: DMA_WrData_1[0]		

06Eh.Bit7-0 DMA_WrData _0[7:0]

06Fh.Bit7-0 DMA_WrData _1[7:0]

Accessing these registers when the DMA_Config.DMA_Mode bit is set to “1” enables data to be written to the FIFO connected to the DMA by the AREAn{n=0-5}Join_0.JoinDMA bit. The DMA_Control.Dir bit must be set to DMA write here.

DMA access is possible in the same way, whether DMA_WrData_H or DMA_WrData_L is accessed when operating in 8-bit mode.

3.1.77 Port0:071h / Port1:271h *ModeProtect (Mode Protection)*

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Device / Host	071h	***ModeProtect***	R/W	7: ***ModeProtect [7]***	Mode Protection	56h
				6: ***ModeProtect [6]***		
				5***: ModeProtect [5]***		
				4: ***ModeProtect [4]***		
				3: ***ModeProtect [3]***		
				2: ***ModeProtect [2]***		
				1: ***ModeProtect [1]***		
				0: ***ModeProtect [0]***		

This register is a common register mirrored at Port 0 and Port 1.

Bit7-0 ***ModeProtect [7:0]***

This protects the CPU_Config and ClkSelect register values. Writing 56h to this register enables write access to the CPU_Config and ClkSelect registers.

For normal use, to protect the CPU_Config and ClkSelect register settings, the CPU_Config and ClkSelect registers should be set as required before setting this register to a value other than 56h (e.g. 00h).

This bit can be accessed even in SLEEP state.

3.1.78 Port0:073h / Port1:273h *ClkSelect (Clock Select)*

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	073h	***ClkSelect***	R/W	7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
			R/W	0: ***ClkSelect***	0: 12MHz	1: 24MHz	

This register is a common register mirrored at Port 0 and Port 1.

It sets the initial values for the LSI clock.

This register must be set before operating the LSI.

The register is enabled even in SLEEP state.

Bit7-1 Reserved

Bit1-0 *ClkSelect*

This sets the clock frequency used by the LSI.

0: 12 MHz

1: 24 MHz

3.1.79 075h *CPU_Config (CPU Configuration)*

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	075h	***CPU_Config***	R/W	7: ***IntLevel***	0: Low Active	1: High Active	01h
			R/W	6: ***IntMode***	0: 1/0 mode	1: Hi-z/0 mode	
			R/W	5: ***DREQ_Level***	0: Low Active	1: High Active	
			R/W	4: ***DACK_Level***	0: Low Active	1: High Active	
			R/W	3: ***CS_Mode***	0: DACK mode	1: CS mode	
			R	2: ***CPU_Endian***	0: Do nothing	1: Bus Swap	
			R	1: ***BusMode***	0: XWRH/L mode	1: XBEH/L mode	
			R	0: ***Inisialized***	0: N/A	1: Initialized	

This register is a common register mirrored at Port 0 and Port 1.

It sets the LSI operating mode.

This bit can be accessed even in SLEEP state.

Bit7 ***IntLevel***

This sets the XINT logic level.

0: Negative logic

1: Positive logic

Bit6 ***IntMode***

This sets the XINT output mode.

0: 1/0 mode

1: Hi-z/0 mode

Bit5 ***DREQ_Level***

This sets the XDREQ logic level.

0: Negative logic

1: Positive logic

Bit4 ***DACK_Level***

This sets the XDACK logic level.

0: Negative logic

1: Positive logic

Bit3 ***CS_Mode***

This sets the DMA operating mode.

0: Operates with DMA access enabled when XDACK is asserted.

1: Operates with DMA access enabled when XCS and XDACK are asserted.

Bit2 **CPU*_Endian***

This indicates the CPU bus setting.

0: Sets even addresses first and odd addresses last.

1: Sets even addresses last and odd addresses first.

This bit indicates the value written to the CPUIF_MODE register during the initialization period.

Bit1 ***BusMode***

This indicates the CPU operating mode setting.

0: 16bit Strobe mode

1: 16bit BE mode

This bit indicates the value written to the CPUIF_MODE register during the initialization period.

Bit0 ***Initialized***

This flag indicates that initialization is complete. It is normally read as "1."

3.1.80 Port0:07E-07Fh / Port1:27Eh-27Fh *CPUIF_MODE (CPUIF Mode)*

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	07Eh-07Fh	***CPUIF_MODE***	W	15: ***MergeDMA***	0: none	1: Merge DMA Ports	XXh
				14:	0:	1:	
				13:	0:	1:	
				12:	0:	1:	
				11:	0:	1:	
			W	10: ***CPU_Endian***	0: Big Endian	1: Little Endian	
			W	9: ***BusMode***	0: Strobe mode	1: BE mode	
				8:	0:	1:	
			W	7: ***MergeDMA***	0: none	1: Merge DMA Ports	
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
			W	2: ***CPU_Endian***	0: Big Endian	1: Little Endian	
			W	1: ***BusMode***	0: Strobe mode	1: BE mode	
				0:	0:	1:	

This register can be accessed only during the uninitialized period.

This register is a common register mirrored at Port 0 and Port 1.

This sets the LSI CPUIF operating mode.

This register must be set after hard-resetting the LSI. For detailed information on setting this register, refer to "1.7.2 CPUIF Mode Setup."

The same register bits are defined by odd and even-number bytes but are set if "1" is written to one or both.

Bit15,8 *MergeDMA*

The XDREQ0 and XDACK0 terminals are shared for the Port 0 and Port 1 DMA. The XDREQ1 and XDACK1 terminals cannot be used.

The Port 0 DMA and Port 1 DMA cannot be operated simultaneously, since the terminals are shared. The DMA for this chip should be controlled exclusively by the software to prevent simultaneously launching with Port 0 and Port 1.

Bit10,2 *CPU_Endian*

Sets the CPUIF endian.

Bit9,1 *BusMode*

Sets the CPUIF write access mode.

Bit14-11,9,6-3,0 *Reserved*

3.1.81 Port0:080h-081h / Port1:280h-281h AREA0StartAdrs_H,L (AREA 0 Start Address High, Low)

3.1.82 Port0:084h-085h / Port1:284h-285h AREA1StartAdrs_H,L (AREA 1 Start Address High, Low)

3.1.83 Port0:088h-089h / Port1:288h-289h AREA2StartAdrs_H,L (AREA 2 Start Address High, Low)

3.1.84 Port0:08Ch-08Dh / Port1:28Ch-28Dh AREA3StartAdrs_H,L (AREA 3 Start Address High, Low)

3.1.85 Port0:090h-091h / Port1:290h-291h AREA4StartAdrs_H,L (AREA 4 Start Address High, Low)

3.1.86 Port0:094h-095h / Port1:294h-295h AREA5StartAdrs_H,L (AREA 5 Start Address High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	080h-081h 084h-085h 088h-089h 08Ch-08Dh 090h-091h 094h-095h	AREA0StartAdrs_H,L AREA1StartAdrs_H,L AREA2StartAdrs_H,L AREA3StartAdrs_H,L AREA4StartAdrs_H,L AREA5StartAdrs_H,L		15:	0:	1:	0000h
				14:	0:	1:	
				13:	0:	1:	
			R/W	12: StartAdrs[12]	AREAx{x=0-5} Start Address		
				11: StartAdrs[11]			
				10: StartAdrs[10]			
				9: StartAdrs[9]			
				8: StartAdrs[8]			
				7: StartAdrs[7]			
				6: StartAdrs[6]			
				5: StartAdrs[5]			
				4: StartAdrs[4]			
				3: StartAdrs[3]			
				2: StartAdrs[2]			
				1:			
				0:			

These set the FIFO areas used by AREAx{x=0-5}.

Bit15-13 Reserved

Bit12 -2 StartAdrs[12:2]

This sets the initial address of the FIFO assigned to the FIFO area AREAx{x=0-5}.

The address is specified in 4-byte units, since it is set with 12 to 2 higher bits.

The area assigned to the FIFO area AREAx{x=0-5} extends up to the first byte of the address set by AREAx{x=0-5}EndAdrs.

The AREAnFIFO_Clr register ClrAREAx{x=0-5} bit must always be set to “1” and the FIFO area AREAx{x=0-5} FIFO cleared after setting AREAx{x=0-5}StartAdrs and AREAx{x=0-5}EndAdrs.

Note that the LSI will not operate correctly if the MaxSize for the USB device/host joined exceeds the area set here. The same applies if the FIFO area overlaps another FIFO area.

This LSI has 4.5 kB of internal RAM and supports addresses up to 0x1200.

Bit1-0 Reserved

3.1.87 Port0:082h-083h / Port1:282h-283h AREA0EndAdrs_H,L (AREA 0 End Address High, Low)

3.1.88 Port0:086h-087h / Port1:286h-287h AREA1EndAdrs_H,L (AREA 1 End Address High, Low)

3.1.89 Port0:08Ah-08Bh / Port1:28Ah-28Bh AREA2EndAdrs_H,L (AREA 2 End Address High, Low)

3.1.90 Port0:08Eh-08Fh / Port1:28Eh-28Fh AREA3EndAdrs_H,L (AREA 3 End Address High, Low)

3.1.91 Port0:092h-093h / Port1:292h-293h AREA4EndAdrs_H,L (AREA 4 End Address High, Low)

3.1.92 Port0:096h-097h / Port1:296h-297h AREA5EndAdrs_H,L (AREA 5 End Address High,Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	082h-083h 086h-087h 08Ah-08Bh 08Eh-08Fh 092h-093h 096h-097h	AREA0EndAdrs_H,L AREA1EndAdrs_H,L AREA2EndAdrs_H,L AREA3EndAdrs_H,L AREA4EndAdrs_H,L AREA5EndAdrs_H,L		15:	0:	1:	0000h
				14:	0:	1:	
				13:	0:	1:	
			R/W	12: EndAdrs[12]	AREAx{x=0-5} End Address		
				11: EndAdrs[11]			
				10: EndAdrs[10]			
				9: EndAdrs[9]			
				8: EndAdrs[8]			
				7: EndAdrs[7]			
				6: EndAdrs[6]			
				5: EndAdrs[5]			
				4: EndAdrs[4]			
				3: EndAdrs[3]			
				2: EndAdrs[2]			
				1:			
				0:			

These set the FIFO areas used by AREAx{x=0-5}.

Bit15-13 Reserved

Bit12 -2 EndAdrs[12:2]

This sets the subsequent byte of the final address of the FIFO assigned to the FIFO area AREAx{x=0-5}.

The address is specified in 4-byte units, since it is set with 2 to 12 higher bits.

The area assigned to the FIFO area AREAx{x=0-5} extends up to the first byte of the address set by AREAx{x=0-5}EndAdrs.

The AREAnFIFO_Clr register ClrAREAx{x=0-5} bit must always be set to “1” and the FIFO area AREAx{x=0-5} FIFO cleared after setting AREAx{x=0-5}StartAdrs and AREAx{x=0-5}EndAdrs.

Note that the LSI will not operate correctly if the MaxSize for the USB device/host joined exceeds the area set here. The same applies if the FIFO area overlaps another FIFO area.

This LSI has 4.5 kB of internal RAM and supports addresses up to 0x1200.

Bit1-0 Reserved

3.1.93 Port0:09Fh / Port1:29Fh AREAnFIFO_Clr (AREA n FIFO Clear)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	09Fh	AREAnFIFO_Clr		7:	0:	1:	XXh
				6:	0:	1:	
			W	5: AREA5FIFO_Clr	0: Do nothing	1: Clear AREA5 FIFO	
			W	4: AREA4FIFO_Clr	0: Do nothing	1: Clear AREA4 FIFO	
			W	3: AREA3FIFO_Clr	0: Do nothing	1: Clear AREA3 FIFO	
			W	2: AREA2FIFO_Clr	0: Do nothing	1: Clear AREA2 FIFO	
			W	1: AREA1FIFO_Clr	0: Do nothing	1: Clear AREA1 FIFO	
			W	0: AREA0FIFO_Clr	0: Do nothing	1: Clear AREA0FIFO	

This write-only register clears the corresponding FIFO area AREAn{n=0-5} FIFO.

Setting each bit of this register to “1” only clears the FIFO; it does not retain the values set.

Do not set the corresponding endpoint bit to “1” when the DMA is joined to the FIFO area AREAn{n=0-5} and the corresponding DMA is running (i.e., while the DMA_Running bit is “1”).

This register only initializes the data retention information. It does not write or clear the data itself. Data in the RAM therefore cannot be cleared by this bit.

3.1.94 Port0:0A0h / Port1:2A0h AREA0Join_0 (AREA 0 Join 0)
3.1.95 Port0:0A2h / Port1:2A2h AREA1Join_0 (AREA 1 Join 0)
3.1.96 Port0:0A4h / Port1:2A4h AREA2Join_0 (AREA 2 Join 0)
3.1.97 Port0:0A6h / Port1:2A6h AREA3Join_0 (AREA 3 Join 0)
3.1.98 Port0:0A8h / Port1:2A8h AREA4Join_0 (AREA 4 Join 0)
3.1.99 Port0:0AAh / Port1:2Aah AREA5Join_0 (AREA 5 Join 0)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0A0h	AREA0Join_0	R/W	7: JoinFIFO_Stat	0: Do nothing	1: Join to FIFO Status	00h
/ Host	0A2h	AREA1Join_0		6:	0:	1:	
	0A4h	AREA2Join_0		5:	0:	1:	
	0A6h	AREA3Join_0		4:	0:	1:	
	0A8h	AREA4Join_0		3:	0:	1:	
	0AAh	AREA5Join_0	R/W	2: JoinDMA	0: Do nothing	1: Join to DMA	
			R/W	1: JoinCPU_Rd	0: Do nothing	1: Join to CPU Read	
			R/W	0: JoinCPU_Wr	0: Do nothing	1: Join to CPU Write	

These set the port connected to the FIFO area AREAx{x=0-5}.

Bit7 JoinFIFO_Stat

Allows monitoring of the FIFO area AREAx{x=0-5} FIFO Full and Empty states using FIFO_IntStat.FIFO_NotEmpty, FIFO_IntStat.FIFO_Full and FIFO_IntStat.FIFO_Empty.

Bit6-3 Reserved

Bit2 JoinDMA

Performs DMA transfers using the FIFO area AREAx{x=0-5} FIFO. The transfer direction is set by the DMA_Control.Dir bit.

Bit1 JoinCPU_Rd

Performs CPU register access read transfers using the FIFO area AREAx{x=0-5} FIFO. In other words, data is read out from this FIFO area if the FIFO_Rd_0,1 or FIFO_ByteRd registers are read.

Bit0 JoinCPU_Wr

Performs CPU register access write transfers using the FIFO area AREAx{x=0-5} FIFO. In other words, data is written to this FIFO area if the FIFO_Wr_0,1 registers are written to.

Setting the JoinDMA bit enables the remaining quantity of data to be checked via the DMA_Remain_H,L registers if the DMA_Control.Dir bit is 1 and free space to be checked if the bit is 0.

If the JoinCPU_Rd and JoinCPU_Wr bits are set, FIFO_RdRemain_H,L and FIFO_WrRemain_H,L can be checked before reading or writing data to or from the FIFO_Rd_0,1, FIFO_ByteRd, and FIFO_Wr_0,1 registers.

Only one of the JoinDMA, JoinCPU_Rd, and JoinCPU_Wr bits should be set to “1” at any given time. Writing “1” to more than one bit simultaneously may destabilize operations.

3.1.100 Port0:0A1h / Port1:2A1h AREA0Join_1 (AREA 0 Join 1)
3.1.101 Port0:0A3h / Port1:2A3h AREA1Join_1 (AREA 1 Join 1)
3.1.102 Port0:0A5h / Port1:2A5h AREA2Join_1 (AREA 2 Join 1)
3.1.103 Port0:0A7h / Port1:2A7h AREA3Join_1 (AREA 3 Join 1)
3.1.104 Port0:0A9h / Port1:2A9h AREA4Join_1 (AREA 4 Join 1)
3.1.105 Port0:0ABh / Port1:2Abh AREA5Join_1 (AREA 5 Join 1)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0A1h	AREA0Join_1		7:	0:	1:	00h
/ Host	0A3h	AREA1Join_1		6:	0:	1:	
	0A5h	AREA2Join_1	R/W	5: JoinEPeCHe	0: Do nothing	1: Join to EPe / CHe	
	0A7h	AREA3Join_1	R/W	4: JoinEPdCHd	0: Do nothing	1: Join to EPd / CHd	
	0A9h	AREA4Join_1	R/W	3: JoinEPcCHc	0: Do nothing	1: Join to EPc / CHc	
	0ABh	AREA5Join_1	R/W	2: JoinEPbCHb	0: Do nothing	1: Join to EPb / CHb	
			R/W	1: JoinEPaCHa	0: Do nothing	1: Join to EPa / CHa	
			R/W	0: JoinEP0CH0	0: Do nothing	1: Join to EP0 / CH0	

These set endpoints and channels connect to the FIFO area AREAx{x=0-5}.

Bit7-6 Reserved

Bit5 JoinEPeCHe

This connects endpoint EPe or channel CHe to the FIFO area AREAx{x=0-5}. Connecting enables the performance of transactions related to data transfers using endpoint EPe or channel EPe.

Bit4 JoinEPdCHd

This connects endpoint EPd or channel CHd to the FIFO area AREAx{x=0-5}. Connecting enables performance of the transactions related to data transfers using endpoint EPd or channel EPd.

Bit3 JoinEPcCHc

This connects endpoint EPc or channel CHc to the FIFO area AREAx{x=0-5}. Connecting enables performance of the transactions related to data transfers using endpoint EPc or channel EPc.

Bit2 JoinEPbCHb

This connects endpoint EPb or channel CHb to the FIFO area AREAx{x=0-5}. Connecting enables performance of the transactions related to data transfers using endpoint EPb or channel EPb.

Bit1 JoinEPaCHa

This connects endpoint EPa or channel CHa to the FIFO area AREAx{x=0-5}. Connecting enables performance of the transactions related to data transfers using endpoint EPa or channel EPa.

Bit0 JoinEP0CH0

This connects endpoint EP0 or channel CH0 to the FIFO area AREAx{x=0-5}. Connecting enables performance of the transactions related to data transfers using endpoint EP0 or channel EP0.

Exercise caution when setting multiple JoinEPxCHx{x=0,a-e} bits simultaneously to the same FIFO area. Unforeseen operations may result, depending on the transaction order. We recommend against setting JoinEPxCHx{x=0,a-e}bits to the same FIFO area under normal conditions.

3.1.106 Port0:0AEh / Port1:2Aeh ClrAREAnJoin_0 (Clear AREA n Join 0)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	0AEh	ClrAREAnJoin_0	W	7: ClrJoinFIFO_Stat	0: Do nothing	1: Clear JoinFIFO_Stat	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
			W	2: ClrJoinDMA	0: Do nothing	1: Clear JoinDMA	
			W	1: ClrJoinCPU_Rd	0: Do nothing	1: Clear JoinCPU_Rd	
			W	0: ClrJoinCPU_Wr	0: Do nothing	1: Clear JoinCPU_Wr	

This write-only register clears the port connection corresponding to each FIFO area.

This register bit is automatically cleared to “0” after the connection is cleared.

Do not set this register bit to “1” while the FIFO area is connected to the port (the bit corresponding to the AREAn{n=0-5}Join_0 register is set to “1”) and each port is running. Doing so will lead to malfunctions.

3.1.107 Port0:0AFh / Port1:2Afh ClrAREAnJoin_1 (Clear AREA n Join 1)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device / Host	0AFh	ClrAREAnJoin_1		7:	0:	1:	00h
				6:	0:	1:	
			W	5: ClrJoinEPeCHe	0: Do nothing	1: Clear JoinEPeCHe	
			W	4: ClrJoinEPdCHd	0: Do nothing	1: Clear JoinEPdCHd	
			W	3: ClrJoinEPcCHc	0: Do nothing	1: Clear JoinEPcCHc	
			W	2: ClrJoinEPbCHb	0: Do nothing	1: Clear JoinEPbCHb	
			W	1: ClrJoinEPaCHa	0: Do nothing	1: Clear JoinEPaCHa	
			W	0: ClrJoinEP0CH0	0: Do nothing	1: Clear JoinEP0CH0	

This write-only register clears the endpoint and channel connections corresponding to each FIFO area.

This register bit is automatically cleared to “0” after the connection is cleared.

Do not set this register bit to “1” while the FIFO area is connected to the endpoint or channel (the bit corresponding to the AREAn{n=0-5}Join_1 register is set to “1”) and each endpoint and channel transaction is executed. Doing so will lead to malfunctions.

3.2 Device Register Details

Registers and register bits related to device functions are defined for Port 0 but not for Port 1. Thus, Port 1 does not include any registers classified as device registers described here.

Addresses are shown as offset addresses from the base address 000h for Port 0.

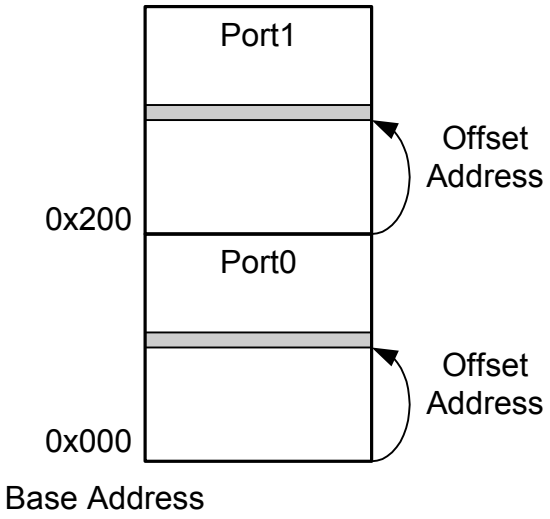


Fig. 3-1 Address notation

3.2.1 Port0:0B0h / Port1:N/A *D_SIE_IntStat (Device SIE Interrupt Status)*

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0B0h	***D_SIE_IntStat***		7:	0:	1:	00h
			R (W)	6: ***NonJ***	0: None	1: Detect Non J state	
			R (W)	5: RcvSOF	0: None	1: Received SOF	
			R (W)	4: DetectReset	0: None	1: Detect USB Reset	
			R (W)	3: DetectSuspend	0: None	1: Detect USB Suspend	
			R (W)	2: ChirpCmp	0: None	1: Chirp Complete	
			R (W)	1: RestoreCmp	0: None	1: Restore Complete	
			R (W)	0: SetAddressCmp	0: None	1: AutoSetAddress Complete	

This indicates the device SIE-related interrupts.

Writing "1" to each bit clears the interrupt factors.

Bit7 Reserved

Bit6 *NonJ*

Directly specifies interrupt factors.

This is set to "1" when a state other than J state is detected on the USB bus. This bit is enabled when the LSI is in SNOOZE state (PM_Control register InSnooze bit is "1") and when the USB_Control register InSUSPEND bit is set to "1" while the AutoNegotiation function is in use.

Bit5 RcvSOF

Directly specifies interrupt factors.

This is set to "1" when an SOF token is received.

Bit4 DetectReset

Directly specifies interrupt factors.

This is set to "1" when a USB RESET state is detected. The USB SUSPEND state cannot be detected (DetectSUSPEND cannot be set) while this bit is set.

Reset detection is enabled when the D_NegoControl register ActiveUSB bit is set to "1."

In "HS" operating mode, FS termination is automatically set to detect USB RESET/SUSPEND if there is no bus activity for a preset duration. Detection of SE0 is assumed to be a reset, and this bit is set to "1."

If this bit is set to "1" when the AutoNegotiation function is not in use, set the D_NegoControl register DisBusDetect bit to "1" to disable USB RESET/SUSPEND state detection to prevent incorrect detection of continuous resets. The DisBusDetect bit should be cleared to "0" to enable USB RESET/SUSPEND state detection after the reset processing has ended.

"HS Detection Handshake" can be started using the D_NegoControl register GoChirp bit on reset detection.

For detailed information on the AutoNegotiation function, refer to the section on the D_NegoControl register EnAutoNego bit.

Bit3 DetectSuspend

Directly specifies interrupt factors.

This is set to "1" if the USB SUSPEND state is detected. The USB RESET state cannot be detected (DetectRESET cannot be set) while this bit is set.

In "HS" operating mode, "FS" operating mode is automatically set to detect USB RESET/SUSPEND if there is no bus activity for a preset duration. Setting the PM_Control. GoSLEEP bit to "1" after detecting USB SUSPEND state allows the LSI to switch to SLEEP mode (internal PLL and oscillator oscillation stopped).

Bit2 ChirpCmp

Directly specifies interrupt factors.

This is set to "1" when the "HS Detection Handshake" started by the D_NegoControl register GoChirp bit ends.

The current operating mode (FS or HS) can be determined by reading the D_USB_Status register FSxHS bit after an interrupt has occurred.

Bit1 RestoreCmp

Directly specifies interrupt factors.

This is set to "1" when the Restore processing started by the D_NegoControl register RestoreUSB bit ends. Setting this bit to "1" returns the operating mode (FS or HS) to the state prior to Suspend.

Bit0 SetAddressCmp

Directly specifies interrupt factors.

The AutoSetAddress function (see "USB_Address Register") automatically performs the control transfer processing on receipt of a SetAddress() request. The status is set to "1" after the status stage has been performed and the control transfer for the SetAddress() request is complete. The address is set in the D_USB_Address register at the same time.

The synchronous bits (bits 5 to 0) cannot be read from or written to (clearing interrupt factors) unless the HostDeviceSel.HOST x DEVICE bit is "0" (i.e., in DEVICE mode) even if power management is in ACTIVE state. The firmware should perform the following processing to prevent assertion of the XINT interrupt signal by the interrupt status when switching from this state.

<When changing from DEVICE mode in ACTIVE state>

1) Process and clear the interrupt status (D_SIE_IntStat.Bit5 to 0)

2) Disable the interrupt status (D_SIE_IntEnb.Bit5 to 0)

<When changing to DEVICE mode in ACTIVE state>

1) Clear the interrupt status (D_SIE_IntStat.Bit5 to 0)

2) Enable the interrupt status (D_SIE_IntEnb.Bit5 to 0)

3.2.2 Port0:0B3h / Port1:N/A D_BulkIntStat (Device Bulk Interrupt Status)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0B3h	D_BulkIntStat	R (W)	7: CBW_Cmp	0: None	1: CBW Complete	00h
			R (W)	6: CBW_LengthErr	0: None	1: CBW Length Error	
			R (W)	5: CBW_Err	0: None	1: CBW Transaction Error	
			R (W)	4:	0:	1:	
			R (W)	3: CSW_Cmp	0: None	1: CSW Complete	
			R (W)	2: CSW_Err	0: None	1: CSW Error	
				1:	0:	1:	
				0:	0:	1:	

This indicates the bulk transfer function interrupt status. Writing "1" to each bit clears the interrupt factors.

Bit7 CBW_Comp

Directly specifies interrupt factors.

This is set to "1" when the 31 CBW bytes are received correctly.

Bit6 CBW_LengthErr

Directly specifies interrupt factors.

This is set to "1" when the received CBW packet length is not 31 bytes.

Bit5 CBW_Err

Directly specifies interrupt factors.

This is set to "1" when a transaction error or other CRC error is detected in the CBW received.

Bit4 Reserved

Bit3 CSW_Cmp

Directly specifies interrupt factors.

This is set to "1" when the 13 CSW bytes are sent correctly.

Bit2 CSW_Err

Directly specifies interrupt factors.

This is set to "1" when an error (ACK is not returned) occurs during CSW transmission.

Bit1-0 Reserved

3.2.3 Port0:0B4h / Port1:N/A D_EPrIntStat (Device EPr Interrupt Status)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0B4h	D_EPrIntStat	R	7: D_AlarmIN_IntStat	0: None	1: Alarm IN Interrupt	00h
			R	6: D_AlarmOUT_IntStat	0: None	1: Alarm OUT Interrupt	
				5:	0:	1:	
			R	4: D_EPeIntStat	0: None	1: EPe Interrupt	
			R	3: D_EPdIntStat	0: None	1: EPd Interrupt	
			R	2: D_EPcIntStat	0: None	1: EPc Interrupt	
			R	1: D_EPbIntStat	0: None	1: EPb Interrupt	
			R	0: D_EPaIntStat	0: None	1: EPa Interrupt	

This indicates the endpoint EPr{r=a-e} and AlarmIN/AlarmOUT interrupts.

Bit7 D_AlarmIN_IntStat

Indirectly specifies interrupt factors.

This is set to “1” when the D_AlarmIN_IntStat_H,L registers contain interrupt factors and the D_AlarmIN_IntEnb_H,L register bits corresponding to the interrupt factors are enabled.

Bit6 D_AlarmOUT_IntStat

Indirectly specifies interrupt factors.

This is set to “1” when the D_AlarmOUT_IntStat_H,L registers contain interrupt factors and the D_AlarmOUT_IntEnb_H,L register bits corresponding to the interrupt factors are enabled.

Bit5 Reserved

Bit4 D_EPeIntStat

Indirectly specifies interrupt factors.

This is set to “1” when the D_EPeIntStat register contains an interrupt factor and the D_EPeIntEnb register bit corresponding to the interrupt factor is enabled.

Bit3 D_EPdIntStat

Indirectly specifies interrupt factors.

This is set to “1” when the D_EPdIntStat register contains an interrupt factor and the D_EPdIntEnb register bit corresponding to the interrupt factor is enabled.

Bit2 D_EPcIntStat

Indirectly specifies interrupt factors.

This is set to “1” when the D_EPcIntStat register contains an interrupt factor and the D_EPcIntEnb register bit corresponding to the interrupt factor is enabled.

Bit1 **D_EPbIntStat**

Indirectly specifies interrupt factors.

This is set to “1” when the D_EPbIntStat register contains an interrupt factor and the D_EPbIntEnb register bit corresponding to the interrupt factor is enabled.

Bit0 **D_EPaIntStat**

Indirectly specifies interrupt factors.

This is set to “1” when the D_EPaIntStat register contains an interrupt factor and the D_EPaIntEnb register bit corresponding to the interrupt factor is enabled.

3.2.4 Port0:0B5h / Port1:N/A D_EP0IntStat (Device EP0 Interrupt Status)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0B5h	D_EP0IntStat	R (W)	7: DescriptorCmp	0: None	1: Descriptor Complete	00h
			R (W)	6: OUT_ShortACK	0: None	1: OUT Short-Packet ACK	
			R (W)	5: IN_TranACK	0: None	1: IN Transaction ACK	
			R (W)	4: OUT_TranACK	0: None	1: OUT Transaction ACK	
			R (W)	3: IN_TranNAK	0: None	1: IN Transaction NAK	
			R (W)	2: OUT_TranNAK	0: None	1: OUT Transaction NAK	
			R (W)	1: IN_TranErr	0: None	1: IN Transaction Error	
			R (W)	0: OUT_TranErr	0: None	1: OUT Transaction Error	

This indicates the endpoint EP0 interrupt status. Writing "1" to each bit clears the interrupt factors.

Bit7 DescriptorCmp

Directly specifies interrupt factors.

This is set to "1" when the Descriptor reply function returns the data quantity set in the DescriptorSize register.

It is also set to "1" together with the OUT_TranNAK bit when switching to the status stage (on receipt of OUT token) before sending the quantity set in the DescriptorSize register.

Bit6 OUT_ShortACK

Directly specifies interrupt factors.

This is set to "1" concurrently with OUT_TranACK when a short packet is received in an OUT transaction and an ACK is returned in response.

Bit5 IN_TranACK

Directly specifies interrupt factors.

This is set to "1" when an ACK is received in an IN transaction.

Bit4 OUT_TranACK

Directly specifies interrupt factors.

This is set to "1" when an ACK is returned in an OUT transaction.

Bit3 IN_TranNAK

Directly specifies interrupt factors.

This is set to "1" when a NAK is returned in an IN transaction.

Bit2 OUT_TranNAK

Directly specifies interrupt factors.

This is set to "1" when a NAK is returned in response to an OUT or PING transaction.

Bit1 **IN_TranErr**

Directly specifies interrupt factors.

This is set to “1” if a STALL is returned in an IN transaction, if a packet error occurs, or if the handshake times out.

Bit0 **OUT_TranErr**

Directly specifies interrupt factors.

This is set to “1” if a STALL is returned in an OUT transaction or if a packet error occurs.

3.2.5 Port0:0B6h / Port1:N/A D_EPaIntStat (Device EPa Interrupt Status)

3.2.6 Port0:0B7h / Port1:N/A D_EPbIntStat (Device EPb Interrupt Status)

3.2.7 Port0:0B8h / Port1:N/A D_EPcIntStat (Device EPc Interrupt Status)

3.2.8 Port0:0B9h / Port1:N/A D_EPdIntStat (Device EPd Interrupt Status)

3.2.9 Port0:0BAh / Port1:N/A D_EPeIntStat (Device EPe Interrupt Status)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0B6h	D_EPaIntStat		7:	0:	1:	00h
	0B7h	D_EPbIntStat	R (W)	6: OUT_ShortACK	0: None	1: OUT Short Packet ACK	
	0B8h	D_EPcIntStat	R (W)	5: IN_TranACK	0: None	1: IN Transaction ACK	
	0B9h	D_EPdIntStat	R (W)	4: OUT_TranACK	0: None	1: OUT Transaction ACK	
	0Bah	D_EPeIntStat	R (W)	3: IN_TranNAK	0: None	1: IN Transaction NAK	
			R (W)	2: OUT_TranNAK	0: None	1: OUT Transaction NAK	
			R (W)	1: IN_TranErr	0: None	1: IN Transaction Error	
			R (W)	0: OUT_TranErr	0: None	1: OUT Transaction Error	

These indicate the interrupt status of endpoint EPx{x=a-e}. Writing "1" to each bit clears the interrupt factors.

Bit7 Reserved

Bit6 OUT_ShortACK

Directly specifies interrupt factors.

This is set to "1" concurrently with OUT_TranACK when a short packet is received in an OUT transaction and an ACK is returned in response.

Bit5 IN_TranACK

Directly specifies interrupt factors.

This is set to "1" when an ACK is received in an IN transaction.

Bit4 OUT_TranACK

Directly specifies interrupt factors.

This is set to "1" when an ACK is returned in an OUT transaction.

Bit3 IN_TranNAK

Directly specifies interrupt factors.

This is set to "1" when a NAK is returned in an IN transaction.

Bit2 OUT_TranNAK

Directly specifies interrupt factors.

This is set to "1" when a NAK is returned in response to an OUT or PING transaction.

Bit1 **IN_TranErr**

Directly specifies interrupt factors.

This is set to “1” if a STALL is returned in an IN transaction, if a packet error occurs, or if the handshake times out.

Bit0 **OUT_TranErr**

Directly specifies interrupt factors.

This is set to “1” if a STALL is returned in an OUT transaction or if a packet error occurs.

3.2.10 Port0:0BCh-0BDh / Port1:N/A D_AlarmIN_IntStat_H,L (Device AlarmIN Interrupt Status High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0BCh-0BDh	D_AlarmIN	R (W)	15: AlarmEP15IN	0: Do nothing	1: EP15 received IN token	0000h
		_IntStat_H,L	R (W)	14: AlarmEP14IN	0: Do nothing	1: EP14 received IN token	
			R (W)	13: AlarmEP13IN	0: Do nothing	1: EP13 received IN token	
			R (W)	12: AlarmEP12IN	0: Do nothing	1: EP12 received IN token	
			R (W)	11: AlarmEP11IN	0: Do nothing	1: EP11 received IN token	
			R (W)	10: AlarmEP10IN	0: Do nothing	1: EP10 received IN token	
			R (W)	9: AlarmEP9IN	0: Do nothing	1: EP9 received IN token	
			R (W)	8: AlarmEP8IN	0: Do nothing	1: EP8 received IN token	
			R (W)	7: AlarmEP7IN	0: Do nothing	1: EP7 received IN token	
			R (W)	6: AlarmEP6IN	0: Do nothing	1: EP6 received IN token	
			R (W)	5: AlarmEP5IN	0: Do nothing	1: EP5 received IN token	
			R (W)	4: AlarmEP4IN	0: Do nothing	1: EP4 received IN token	
			R (W)	3: AlarmEP3IN	0: Do nothing	1: EP3 received IN token	
			R (W)	2: AlarmEP2IN	0: Do nothing	1: EP2 received IN token	
			R (W)	1: AlarmEP1IN	0: Do nothing	1: EP1 received IN token	
				0:	0:	1:	

This indicates the Alarm IN interrupt status. Writing “1” to each bit clears the interrupt factors.

All bits directly specify interrupt factors.

The response shown below is returned and the corresponding bit is set to “1” if a transaction is issued (IN token received) from the host to the endpoints enabled by D_EnEP_IN_H,L, except those appropriately set in the D_EPx{x=0,a-e} registers and for which joining is set by the AREAn{n=0-5}Join.JoinEPxCHx{x=0,a-e} bit. Note that the same operation is performed even when the D_EP0 registers are not set appropriately and the AREAn{n=0-5}Join.JoinEP0CH0 bit is not set to any FIFO area, since endpoint EP0 is always enabled.

The response to the IN token here is based on the D_EnEP_IN_ISO_H,L settings. A zero-length packet is returned to the host for endpoints for which the corresponding bit is set to “1.” A NAK is returned to the host for endpoints cleared to “0.”

Enable transactions by joining the endpoint to the FIFO area by setting the D_EPx{x=0,a-e} registers appropriately and using the AREAn{n=0-5}Join.JoinEPxCHx{x=0,a-e} bit if the corresponding bit of this register is set.

3.2.11 Port0:0BEh-0BFh / Port1:N/A D_AlarmOUT_IntStat_H,L (Device AlarmOUT Interrupt Status High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0BEh-0BFh	D_AlarmOUT	R (W)	15: AlarmEP15OUT	0: Do nothing	1: EP15 received OUT token	0000h
		_IntStat_H,L	R (W)	14: AlarmEP14OUT	0: Do nothing	1: EP14 received OUT token	
			R (W)	13: AlarmEP13OUT	0: Do nothing	1: EP13 received OUT token	
			R (W)	12: AlarmEP12OUT	0: Do nothing	1: EP12 received OUT token	
			R (W)	11: AlarmEP11OUT	0: Do nothing	1: EP11 received OUT token	
			R (W)	10: AlarmEP10OUT	0: Do nothing	1: EP10 received OUT token	
			R (W)	9: AlarmEP9OUT	0: Do nothing	1: EP9 received OUT token	
			R (W)	8: AlarmEP8OUT	0: Do nothing	1: EP8 received OUT token	
			R (W)	7: AlarmEP7OUT	0: Do nothing	1: EP7 received OUT token	
			R (W)	6: AlarmEP6OUT	0: Do nothing	1: EP6 received OUT token	
			R (W)	5: AlarmEP5OUT	0: Do nothing	1: EP5 received OUT token	
			R (W)	4: AlarmEP4OUT	0: Do nothing	1: EP4 received OUT token	
			R (W)	3: AlarmEP3OUT	0: Do nothing	1: EP3 received OUT token	
			R (W)	2: AlarmEP2OUT	0: Do nothing	1: EP2 received OUT token	
			R (W)	1: AlarmEP1OUT	0: Do nothing	1: EP1 received OUT token	
				0:	0:	1:	

This indicates the Alarm OUT interrupt status. Writing “1” to each bit clears the interrupt factors.

All bits directly specify interrupt factors.

The response shown below is returned and the corresponding bit is set to “1” if a transaction is issued (OUT token received) from the host to the endpoints enabled by D_EnEP_OUT_H,L, except those appropriately set in the D_EPx{x=0,a-e} registers and for which joining is set by the AREAn{n=0-5}Join.JoinEPxCHx{x=0,a-e} bit. Note that the same operation is performed even when the D_EP0 registers are not set appropriately and the AREAn{n=0-5}Join.JoinEP0CH0 bit is not set to any FIFO area, since endpoint EP0 is always enabled.

The response to the OUT token here is based on the D_EnEP_OUT_ISO_H,L settings. The data sent by the host is not accepted, and a handshake is not returned to the host for endpoints for which the corresponding bit is set to “1.” A NAK is returned to the host for endpoints cleared to “0.” A NAK is returned if the device is set to HS and a PING token is issued by the host.

Enable transactions by joining the endpoint to the FIFO area by setting the D_EPx{x=0,a-e} registers appropriately and using the AREAn{n=0-5}Join.JoinEPxCHx{x=0,a-e} bit if the corresponding bit of this register is set.

3.2.12 Port0:0C0h / Port1:N/A *D_SIE_IntEnb (Device SIE Interrupt Enable)*

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0C0h	***D_SIE_IntEnb***		7:	0:	1:	00h
			R/W	6: ***EnNonJ***	0: Disable	1: Enable	
			R/W	5: EnRcvSOF	0: Disable	1: Enable	
			R/W	4: EnDetectRESET	0: Disable	1: Enable	
			R/W	3: EnDetectSUSPEND	0: Disable	1: Enable	
			R/W	2: EnChirpCmp	0: Disable	1: Enable	
			R/W	1: EnRestoreCmp	0: Disable	1: Enable	
			R/W	0: EnSetAddressCmp	0: Disable	1: Enable	

This permits or prohibits assertion of the MainIntStat register D_SIE_IntStat bit using the D_SIE_IntStat register interrupt factors.

The EnNonJ bit is enabled even in SLEEP state.

3.2.13 Port0:0C3h / Port1:N/A D_BulkIntEnb (Device Bulk Interrupt Enable)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0C3h	D_BulkIntEnb	R/W	7: EnCBW_Cmp	0: Disable	1: Enable	00h
			R/W	6: EnCBW_LengthErr	0: Disable	1: Enable	
			R/W	5: EnCBW_Err	0: Disable	1: Enable	
				4:	0:	1:	
			R/W	3: EnCSW_Cmp	0: Disable	1: Enable	
			R/W	2: EnCSW_Err	0: Disable	1: Enable	
				1:	0:	1:	
				0:	0:	1:	

This permits or prohibits assertion of the MainIntStat register D_BulkIntStat bit using the D_BulkIntStat register interrupt factors.

3.2.14 Port0:0C4h / Port1:N/A D_EPrIntEnb (Device EPr Interrupt Enable)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0C4h	D_EPrIntEnb	R/W	7: EnD_AlarmIN_IntStat	0:	1:	00h
			R/W	6: EnD_AlarmOUT_IntStat	0:	1:	
				5:	0:	1:	
			R/W	4: EnD_EPeIntStat	0: Disable	1: Enable	
			R/W	3: EnD_EPdIntStat	0: Disable	1: Enable	
			R/W	2: EnD_EPcIntStat	0: Disable	1: Enable	
			R/W	1: EnD_EPbIntStat	0: Disable	1: Enable	
			R/W	0: EnD_EPaIntStat	0: Disable	1: Enable	

This permits or prohibits assertion of the MainIntStat register D_EPrIntStat bit using the D_EPrIntStat register interrupt factors.

3.2.15 Port0:0C5h / Port1:N/A D_EP0IntEnb (Device EP0 Interrupt Enable)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0C5h	D_EP0IntEnb	R/W	7: EnDescriptorCmp	0: Disable	1: Enable	00h
			R/W	6: EnOUT_ShortACK	0: Disable	1: Enable	
			R/W	5: EnIN_TranACK	0: Disable	1: Enable	
			R/W	4: EnOUT_TranACK	0: Disable	1: Enable	
			R/W	3: EnIN_TranNAK	0: Disable	1: Enable	
			R/W	2: EnOUT_TranNAK	0: Disable	1: Enable	
			R/W	1: EnIN_TranErr	0: Disable	1: Enable	
			R/W	0: EnOUT_TranErr	0: Disable	1: Enable	

This permits or prohibits assertion of the MainIntStat register D_EP0IntStat bit using the D_EP0IntStat register interrupt factors.

3.2.16 Port0:0C6h / Port1:N/A D_EPaIntEnb (Device EPa Interrupt Enable)

3.2.17 Port0:0C7h / Port1:N/A D_EPbIntEnb (Device EPb Interrupt Enable)

3.2.18 Port0:0C8h / Port1:N/A D_EPcIntEnb (Device EPc Interrupt Enable)

3.2.19 Port0:0C9h / Port1:N/A D_EPdIntEnb (Device EPd Interrupt Enable)

3.2.20 Port0:0CAh / Port1:N/A D_EPeIntEnb (Device EPe Interrupt Enable)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0C6h	D_EPaIntEnb		7:	0:	1:	00h
	0C7h	D_EPbIntEnb	R/W	6: EnOUT_ShortACK	0: Disable	1: Enable	
	0C8h	D_EPcIntEnb	R/W	5: EnIN_TranACK	0: Disable	1: Enable	
	0C9h	D_EPdIntEnb	R/W	4: EnOUT_TranACK	0: Disable	1: Enable	
	0CAh	D_EPeIntEnb	R/W	3: EnIN_TranNAK	0: Disable	1: Enable	
			R/W	2: EnOUT_TranNAK	0: Disable	1: Enable	
			R/W	1: EnIN_TranErr	0: Disable	1: Enable	
			R/W	0: EnOUT_TranErr	0: Disable	1: Enable	

These permit or prohibit assertion of the D_EPrIntStat register EPx{x=a-e}IntStat bit using the D_EPx{x=a-e}IntStat register interrupt factors.

3.2.21 Port0:0CCh-0CDh / Port1:N/A D_AlarmIN_IntEnb_H,L (Device AlarmIN Interrupt Enable High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0CCh-0CDh	D_AlarmIN	R/W	15: EnAlarmEP15IN	0: Disable	1: Enable	0000h
		_IntEnb,L	R/W	14: EnAlarmEP14IN	0: Disable	1: Enable	
			R/W	13: EnAlarmEP13IN	0: Disable	1: Enable	
			R/W	12: EnAlarmEP12IN	0: Disable	1: Enable	
			R/W	11: EnAlarmEP11IN	0: Disable	1: Enable	
			R/W	10: EnAlarmEP10IN	0: Disable	1: Enable	
			R/W	9: EnAlarmEP9IN	0: Disable	1: Enable	
			R/W	8: EnAlarmEP8IN	0: Disable	1: Enable	
			R/W	7: EnAlarmEP7IN	0: Disable	1: Enable	
			R/W	6: EnAlarmEP6IN	0: Disable	1: Enable	
			R/W	5: EnAlarmEP5IN	0: Disable	1: Enable	
			R/W	4: EnAlarmEP4IN	0: Disable	1: Enable	
			R/W	3: EnAlarmEP3IN	0: Disable	1: Enable	
			R/W	2: EnAlarmEP2IN	0: Disable	1: Enable	
			R/W	1: EnAlarmEP1IN	0: Disable	1: Enable	
				0:	0:	1:	

This permits or prohibits assertion of the D_EPrIntStat register AlarmIN_IntStat bit using the D_AlarmIN_IntStat register interrupt factors.

3.2.22 Port0:0CEh-0CFh / Port1:N/A D_AlarmOUT_IntEnb_H,L (Device AlarmOUT Interrupt Enable High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0CEh-0CFh	D_AlarmOUT	R/W	15: EnAlarmEP15OUT	0: Disable	1: Enable	0000h
		_IntEnb_H,L	R/W	14: EnAlarmEP14OUT	0: Disable	1: Enable	
			R/W	13: EnAlarmEP13OUT	0: Disable	1: Enable	
			R/W	12: EnAlarmEP12OUT	0: Disable	1: Enable	
			R/W	11: EnAlarmEP11OUT	0: Disable	1: Enable	
			R/W	10: EnAlarmEP10OUT	0: Disable	1: Enable	
			R/W	9: EnAlarmEP9OUT	0: Disable	1: Enable	
			R/W	8: EnAlarmEP8OUT	0: Disable	1: Enable	
			R/W	7: EnAlarmEP7OUT	0: Disable	1: Enable	
			R/W	6: EnAlarmEP6OUT	0: Disable	1: Enable	
			R/W	5: EnAlarmEP5OUT	0: Disable	1: Enable	
			R/W	4: EnAlarmEP4OUT	0: Disable	1: Enable	
			R/W	3: EnAlarmEP3OUT	0: Disable	1: Enable	
			R/W	2: EnAlarmEP2OUT	0: Disable	1: Enable	
			R/W	1: EnAlarmEP1OUT	0: Disable	1: Enable	
				0:	0:	1:	

This permits or prohibits assertion of the D_EPrIntStat register AlarmOUT_IntStat bit using the D_AlarmOUT_IntStat register interrupt factors.

3.2.23 Port0:0D0h / Port1:N/A D_NegoControl (Device Negotiation Control)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0D0h	D_NegoControl	R/W	7: DisBusDetect	0: Enable BusDetect	1: Disable BusDetect	00h
			R/W	6: EnAutoNego	0: Disable AutoNegotiation	1: Enable AutoNegotiation	
			R/W	5: InSUSPEND	0: Do nothing	1: Monitor NonJ	
			R/W	4: DisableHS	0: HS mode	1: Disable HS mode	
			R/W	3: SendWakeup	0: Do nothing	1:Send Remotewakeup Signal	
			R/W	2: RestoreUSB	0: Do nothing	1: Restore operation mode	
			R/W	1: GoChirp	0: Do nothing	1: Do Chirp sequence	
			R/W	0: ActiveUSB	0: Disactivate USB	1: Activate USB	

This sets operations related to device negotiation.

Bit7 DisBusDetect

Setting this bit to "1" disables USB RESET/SUSPEND state automatic detection. USB bus activity is monitored to detect the USB RESET/SUSPEND state if this bit is cleared to "0."

If no bus activity is detected for at least 3 ms in "HS" mode, the mode automatically switches to "FS" mode. The USB is then determined to be in a RESET or SUSPEND state before setting the corresponding interrupt factor (DetectReset or DetectSuspend). In "FS" mode, the USB is determined to be in a SUSPEND state if no bus activity is detected for at least 3 ms or is determined to be in a RESET state if "SE0" is detected for at least 2.5 µs. The corresponding interrupt factor is then set.

If the DetectReset or DetectSuspend bits are set to "1," set the DisBusDetect bit to "1" and disable detection while the USB remains in RESET or SUSPEND states. Do not set this bit to "1" when using the AutoNegotiation function.

Bit6 EnAutoNego

This enables the AutoNegotiation function. The AutoNegotiation function automates the sequence from completion of speed negotiation to determining the speed mode when RESET is detected. For detailed information on the AutoNegotiation function, refer to the section that describes how functions are used.

Bit5 InSUSPEND

This is automatically set to "1" to enable the NonJ state detection function upon detection of the USB SUSPEND state when using the AutoNegotiation function. Clear this bit to "0" to reset from the USB SUSPEND state.

For detailed information on the AutoNegotiation function, refer to the "Functions: AutoNegotiation Function" description.

Bit4 DisableHS

Setting this bit to “1” when GoChirp is set to “1” forces a switch to FS mode without sending a DeviceChirp, then issues a ChirpCmp interrupt.

Bit3 SendWakeup

Setting this bit to “1” outputs a RemoteWakeup signal (K) to the USB port.

Clear this bit to “0” to halt transmission between 1 ms and 15 ms after the start of RemoteWakeup signal transmission.

Bit2 RestoreUSB

Setting this bit to “1” when resuming from USB SUSPEND state switches automatically to the operating mode (FS or HS) saved before entering USB SUSPEND and sets the corresponding interrupt factor (RestoreCmp).

This bit is automatically cleared to “0” once the operation ends.

Do not clear or set this bit when using the AutoNegotiation function; the function of this bit is controlled automatically.

Bit1 GoChirp

Setting this bit to “1” when the USB bus is in a RESET state performs an “HS Detection Handshake” with the host or hub and automatically sets the XcvrControl register TermSelect and XcvrSelect bits and the USB_Status register FSxHS bit. The interrupt factor (ChirpCmp) is set simultaneously when the operation ends.

This bit is automatically cleared to “0” when the operation ends. The “HS Detection Handshake” result can be checked by inspecting the USBStatus register FSxHS bit after the operation has ended.

Do not set or clear this bit when using the AutoNegotiation function; the function of this bit is controlled automatically.

Bit0 ActiveUSB

The LSI stops all USB device functions, since this bit is cleared to “0” after a hard reset. Setting this bit to “1” after setting the LSI allows operation as a USB device.

3.2.24 Port0:0D3h / Port1:N/A D_XcvrControl (Device Xcvr Control)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0D3h	D_XcvrControl	R/W	7: TermSelect	0: HS Termination	1: FS Termination	41h
			R/W	6: XcvrSelect	0: HS Transceiver	1: FS Transceiver	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
			R/W	1: OpMode [1]	OpMode [1:0]		
				0: OpMode [0]			

This sets device transceiver macros.

Bit7 TermSelect

This selects and enables either FS or HS termination. This bit is set automatically if a "HS detection handshake" is performed using the USB_Control register GoChirp bit or if the D_NegoControl register EnAutoNego bit is set to perform the AutoNegotiation function.

Bit6 XcvrSelect

This selects and enables either FS or HS transceiver. This bit is set automatically if a "HS detection handshake" is performed using the D_NegoControl register GoChirp bit or if the D_NegoControl register EnAutoNego bit is set to perform the AutoNegotiation function.

Bit5-2 Reserved

Bit1-0 OpMode

This sets the MTM operating mode.

This normally does not need to be set, except when the USB cable has been disconnected(*), the USB is in SUSPEND state, or in test mode.

OpMode		
00	"Normal Operation"	Normal usage state
01	"Non-Driving"	Set to this state when the USB cable has been disconnected.
10	"Disable Bitstuffing and NRZI encoding"	Set to this state when in USB test mode.
11	"Power-Down"	Set to this state when in USB SUSPEND state.

* We recommend setting this register to "41h" when the USB cable has been disconnected.

3.2.25 Port0:0D4h / Port1:N/A D_USB_Test (Device USB_Test)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0D4h	D_USB_Test	R/W	7: EnHS_Test	0: Do nothing	1: EnHS_Test	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
			R/W	3: Test_SE0_NAK	0: Do nothing	1: Test_SE0_NAK	
			R/W	2: Test_J	0: Do nothing	1: Test_J	
			R/W	1: Test_K	0: Do nothing	1: Test_K	
			R/W	0: Test_Packet	0: Do nothing	1: Test_Packett	

This sets device USB 2.0 test mode operations. Set the bit corresponding to the test mode specified by the SetFeature request, then set the EnHS_Test bit to "1" after the status stage ends to ensure operation using the test mode defined by the USB 2.0 standards.

Bit7 EnHS_Test

When this bit is set to "1", setting any one of the four lower bits in the D_USB_Test register to "1" switches to the test mode corresponding to such bit. In test mode, the D_NegoControl register DisBusDetect bit must be set to "1" to prevent USB SUSPEND and RESET detection. Additionally, clear the D_NegoControl register EnAutoNego bit to "0" to disable the AutoNegotiation function.

Be careful to avoid switching to test mode until the status stage has ended for the SetFeature request.

Bit6-4 Reserved

Bit3 Test_SE0_NAK

Setting this bit to "1" and setting the EnHS_Test bit to "1" allows use of Test_SE0_NAK test mode.

Bit2 TEST_J

Setting this bit to "1" and setting the EnHS_Test bit to "1" allows use of Test_J test mode. Note that the XcvrControl register TermSelect and XcvrSelect bits should be set according to speed, and OpMode should be set to "10" (Disable Bitstuffing and NRZI encoding) before setting the EnHS_Test bit to "1" in this test mode.

Bit1 TEST_K

Setting this bit to "1" and setting the EnHS_Test bit to "1" allows use of Test_K test mode. Note that the XcvrControl register TermSelect and XcvrSelect bits should be set according to speed, and OpMode should be set to "10" (Disable Bitstuffing and NRZI encoding) before setting the EnHS_Test bit to "1" in this test mode.

Bit0 Test_Packet

Setting this bit to "1" and setting the EnHS_Test bit to "1" allows use of Test_Packet test mode.

Use the following settings, since this test mode can be used for any endpoint except EP0.

1) Set the endpoint EPx{x=a-e} MaxPacketSize to at least 64, set the transfer direction to IN, and set EndpointNumber to "0xF" to enable. Assign a FIFO of at least 64 bytes for endpoint EPx{x=a-e}.
2) Make sure the settings for other endpoints do not duplicate the above settings for EPx{x=a-e}. Clear the AREAx{x=1-5}Join_1.JoinEPxCHx{x=a-e} bit.
3) Clear the EPx{x=a-e} FIFO, then write the following test packet data to this FIFO. Clear the D_EPx{x=a-e}IntStat register IN_TranErr bit to "0."
4) The IN_TranErr status is set to "1" after each test packet transmission is completed. The following 53 bytes of data are written to the FIFO in packet transmission test mode.

   00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h,

   00h, AAh, AAh, AAh, AAh, AAh, AAh, AAh,

   AAh, EEh, EEh, EEh, EEh, EEh, EEh, EEh,

   EEh, FEh, FFh, FFh, FFh, FFh, FFh, FFh,

   FFh, FFh, FFh, FFh, FFh, 7Fh, BFh, DFh,

   EFh, F7h, FBh, FDh, FCh, 7Eh, BFh, DFh,

   EFh, F7h, FBh, FDh, 7Eh

   PID and CRC are added by SIE when sending test packets. This means that the data written to the FIFO consists of data from the following DATA0 PID data to the CRC16 data of the test packet data described in the USB standard Rev 2.0.

3.2.26 Port0:0D6h / Port1:N/A D_EPnControl (Device Endpoint Control)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0D6h	D_EPnControl	W	7: AllForceNAK	0: Do nothing	1: Set All ForceNAK	XXh
			W	6: EPrForceSTALL	0: Do nothing	1: Set EP's ForceSTALL	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
				0:	0:	1:	

This write-only register sets endpoint operations.

Bit7 AllForceNAK

This sets the ForceNAK bit to "1" for all endpoints.

Bit6 EPrForceSTALL

This sets the ForceSTALL bit to "1" for the EPa, EPb, EPc, EPd, and EPe endpoints.

Bit5-0 Reserved

3.2.27 Port0:0D8h / Port1:N/A D_BulkOnlyControl (Device BulkOnly Control)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0D8h	D_BulkOnlyControl	R/W	7:AutoForceNAK_CBW	0: None	1: AutoForceNAK after CBW	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
			R/W	2: GoCBW_Mode	0: None	1: Begin CBW Mode	
			R/W	1: GoCSW_Mode	0: None	1: Begin CSW Mode	
				0:	0:	1:	

This controls the bulk-only support function.

Bit7 AutoForceNAK_CBW

Setting this bit to "1" sets the ForceNAK bit to "1" for the corresponding endpoint once the OUT transaction received by the CBW using CBW support is complete.

Bit6-3 Reserved

Bit2 GoCBW_Mode

Setting this bit to "1" runs CBW support for the corresponding endpoint. For detailed information on the endpoint running CBW support, refer to the BulkOnlyConfig register section.

Bit1 GoCSW_Mode

Setting this bit to "1" runs CSW support for the corresponding endpoint. For detailed information on the endpoint running CSW support, refer to the BulkOnlyConfig register section.

Bit0 Reserved

3.2.28 Port0:0D9h / Port1:N/A D_BulkOnlyConfig (Device BulkOnly Configuration)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0D9h	D_BulkOnlyConfig		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
			R/W	4: EPeBulkOnly	0: None	1: Enable BulkOnly on EPe	
			R/W	3: EPdBulkOnly	0: None	1: Enable BulkOnly on EPd	
			R/W	2: EPcBulkOnly	0: None	1: Enable BulkOnly on EPc	
			R/W	1: EPbBulkOnly	0: None	1: Enable BulkOnly on EPb	
			R/W	0: EPaBulkOnly	0: None	1: Enable BulkOnly on EPa	

This enables the bulk-only support function.

Bit7-5 Reserved

Bit4 EPeBulkOnly

Setting this bit to "1" enables the bulk-only support function for endpoint EPe. With the bulk-only support function enabled, CBW support is implemented by setting the BulkOnlyControl.GoCBW_Mode bit when endpoint EPe is the OUT endpoint. Similarly, CSW support is implemented by setting the BulkOnlyControl.GoCSW_Mode bit when endpoint EPe is the IN endpoint.

Do not enable the bulk-only support function with more than one OUT endpoint.

Similarly, do not enable the bulk-only support function with more than one IN endpoint.

Bit3 EPdBulkOnly

Setting this bit to "1" enables the bulk-only support function for endpoint EPd. With the bulk-only support function enabled, CBW support is implemented by setting the BulkOnlyControl.GoCBW_Mode bit when endpoint EPd is the OUT endpoint. Similarly, CSW support is implemented by setting the BulkOnlyControl.GoCSW_Mode bit when endpoint EPd is the IN endpoint.

Do not enable the bulk-only support function with more than one OUT endpoint.

Similarly, do not enable the bulk-only support function with more than one IN endpoint.

Bit2 EPcBulkOnly

Setting this bit to "1" enables the bulk-only support function for endpoint EPc. With the bulk-only support function enabled, CBW support is implemented by setting the BulkOnlyControl.GoCBW_Mode bit when endpoint EPc is the OUT endpoint. Similarly, CSW support is implemented by setting the BulkOnlyControl.GoCSW_Mode bit when endpoint EPc is the IN endpoint.

Do not enable the bulk-only support function with more than one OUT endpoint.

Similarly, do not enable the bulk-only support function with more than one IN endpoint.

Bit1 EPbBulkOnly

Setting this bit to "1" enables the bulk-only support function for endpoint EPb. With the bulk-only support function enabled, CBW support is implemented by setting the BulkOnlyControl.GoCBW_Mode bit when endpoint EPb is the OUT endpoint. Similarly, CSW support is implemented by setting the BulkOnlyControl.GoCSW_Mode bit when endpoint EPb is the IN endpoint.

Do not enable the bulk-only support function with more than one OUT endpoint.

Bit0 EPaBulkOnly

Setting this bit to "1" enables the bulk-only support function for endpoint EPa. With the bulk-only support function enabled, CBW support is implemented by setting the BulkOnlyControl.GoCBW_Mode bit when endpoint EPa is the OUT endpoint. Similarly, CSW support is implemented by setting the BulkOnlyControl.GoCSW_Mode bit when endpoint EPa is the IN endpoint.

Do not enable the bulk-only support function with more than one OUT endpoint.

3.2.29 Port0:0E0h / Port1:N/A D_EP0SETUP_0 (Device EP0 SETUP 0)
3.2.30 Port0:0E1h / Port1:N/A D_EP0SETUP_1 (Device EP0 SETUP 1)
3.2.31 Port0:0E2h / Port1:N/A D_EP0SETUP_2 (Device EP0 SETUP 2)
3.2.32 Port0:0E3h / Port1:N/A D_EP0SETUP_3 (Device EP0 SETUP 3)
3.2.33 Port0:0E4h / Port1:N/A D_EP0SETUP_4 (Device EP0 SETUP 4)
3.2.34 Port0:0E5h / Port1:N/A D_EP0SETUP_5 (Device EP0 SETUP 5)
3.2.35 Port0:0E6h / Port1:N/A D_EP0SETUP_6 (Device EP0 SETUP 6)
3.2.36 Port0:0E7h / Port1:N/A D_EP0SETUP_7 (Device EP0 SETUP 7)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Device	0E0h -0E7h	D_EP0SETUP_0 ~D_EP0SETUP_7	R	7: EP0SETUP_n [7]	Endpoint 0 SETUP Data 0 -Endpoint 0 SETUP Data 7	00h
				6: EP0SETUP_n [6]		
				5: EP0SETUP_n [5]		
				4: EP0SETUP_n [4]		
				3: EP0SETUP_n [3]		
				2: EP0SETUP_n [2]		
				1: EP0SETUP_n [1]		
				0: EP0SETUP_n [0]		

The 8 bytes of data received in the endpoint EP0 setup stage are stored in sequence from EP0SETUP_0.

EP0SETUP_0

Set with BmRequestType.

EP0SETUP_1

Set with BRequest.

EP0SETUP_2

Set with the lower 8 bits of Wvalue.

EP0SETUP_3

Set with the higher 8 bits of Wvalue.

EP0SETUP_4

Set with the lower 8 bits of WIndex.

EP0SETUP_5

Set with the higher 8 bits of WIndex.

EP0SETUP_6

Set with the lower 8 bits of WLength.

EP0SETUP_7

Set with the higher 8 bits of WLength.

3.2.37 Port0:0E8h / Port1:N/A D_USB_Address (Device USB Address)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0E8h	D_USB_Address		7:	0:	1:	00h
			R (W)	6: USB_Address [6]	USB Address		
				5: USB_Address [5]			
				4: USB_Address [4]			
				3: USB_Address [3]			
				2: USB_Address [2]			
				1: USB_Address [1]			
				0: USB_Address [0]			

The USB address is set by the AutoSetAddress function.

The corresponding control transfer is performed automatically by the AutoSetAddress function on receiving a SetAddress() request. The AutoSetAddress function issues SetAddressCmp status notification after the control transfer status stage for the SetAddress() request is complete and USB_Address has been set.

Bit7 Reserved

Bit6-0 USB_Address

This sets the USB address.

This is written to automatically by the AutoSetAddress function.

Writing is possible, but it will be rewritten automatically on receiving a SetAddress() request.

3.2.38 Port0:0EAh / Port1:N/A D_SETUP_Control(Device SETUP Control)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0EAh	D_SETUP_Control		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
			R/W	0: ProtectEP0	0: None	1: Protect EP0	

This sets items related to control transfers.

Bit7-1 Reserved

Bit0 ProtectEP0

This is set to "1" when the control transfer setup stage ends and the data received is stored in registers D_EP0SETUP_0 to D_EP0SETUP_7.

At the same time, the D_EP0ControlIN and D_EP0ControlOUT register ForceSTALL bits are automatically set to "0," the ForceNAK bits are set to "1," and the ToggleStat bits are set to "1."

Since the ProtectEP0 bit is set when the SETUP transaction is run, it is also set for a SetAddress() request.

Setting this bit to "1" prevents changes to EP0 ForceNAK and ForceSTALL bit settings.

3.2.39 Port0:0EEh-0EFh / Port1:N/A D_FrameNumber_H,L (Device FrameNumber High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0EEh-0EFh	D_FrameNumber_H,L	R	15: FnInvalid	0: Frame number is valid	1: Frame number is not valid	8000h
				14:	0:	1:	
				13:	0:	1:	
				12:	0:	1:	
				11:	0:	1:	
			R	10: FrameNumber [10]	Frame Number		
				9: FrameNumber [9]			
				8: FrameNumber [8]			
				7: FrameNumber [7]			
				6: FrameNumber [6]			
				5: FrameNumber [5]			
				4: FrameNumber [4]			
				3: FrameNumber [3]			
				2: FrameNumber [2]			
				1: FrameNumber [1]			
				0: FrameNumber [0]			

This indicates the USB frame number updated each time an SOF token is received.

Bit15 FnInvalid

This bit is set to "1" if an error occurs in the SOF packet received.

Bit14-11 Reserved

Bit10-0 FrameNumber[10:0]

This indicates FrameNumber for the SOF packet received.

3.2.40 Port0:0F0h / Port1:N/A D_EP0MaxSize (Device EP0 Max Packet Size)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0F0h	D_EP0MaxSize		7:	0:	1:	40h
			R/W	6: EP0MaxSize [6]	Endpoint [0] Max Packet Size		
				5: EP0MaxSize [5]			
				4: EP0MaxSize [4]			
				3: EP0MaxSize [3]			
				2:	0:	1:	
				1:	0:	1:	
				0:	0:	1:	

This sets the endpoint EP0.

Bit7 Reserved

Bit6-3 EP0MaxSize[6:3]

This sets MaxPacketSize for endpoint EP0.

This endpoint can be used with any of the sizes shown below selected.

FS: 8, 16, 32, or 64 bytes

HS: 64 bytes

Bit2-0 Reserved

3.2.41 Port0:0F1h / Port1:N/A D_EP0Control (Device EP0 Control)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0F1h	D_EP0Control	R/W	7: INxOUT	0: OUT	1: IN	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
			R/W	0: ReplyDescriptor	0: Do nothing	1: Reply Descriptor	

This sets the endpoint EP0.

Bit7 INxOUT

This sets the endpoint EP0 transfer direction.

Identify the request received in the setup stage, then set this bit to that value.

If there is a data stage, set this bit to the transfer direction for the data stage. Since the D_EP0ControlIN and D_EP0ControlOUT register ForceNAK bits are set when the setup stage is complete, this should be cleared when running the data or status stage.

Set this bit once again to suit the status stage direction once the data stage has ended. This bit should be set to "0" if the data stage transfer direction is IN, since the status stage will be in the OUT direction. Similarly, if the data stage direction is OUT or if there is no data stage, the status stage will be in the IN direction; the endpoint EP0 FIFO should be cleared and this bit set to "1."

A NAK response is returned to IN or OUT transactions with a direction differing from this bit setting. However, a STALL response is returned if the D_EP0ControlIN or D_EP0ControlOUT register ForceSTALL bit is set for the transaction direction.

Bit6-1 Reserved

Bit0 ReplyDescriptor

This runs the descriptor reply function.

Setting this bit to "1" returns descriptor data corresponding to the MaxPacketSize size from the FIFO in response to the endpoint EP0 IN transaction. The descriptor data refers to the data corresponding to the size set by the D_DescSize_H,L registers, starting with the addresses set in the D_DescAdrs_H,L registers. These settings should be set each time the ReplyDescriptor bit is set, since they are updated while the descriptor reply function is being run.

The D_DescAdrs_H,L registers are incremented by the amount of data transmitted for each transaction, and the D_DescSize_H,L registers are decremented by the amount of data transmitted.

When the amount of data set in D_DescSize_H,L has been sent or if a transaction other than an IN transaction is performed, the descriptor reply function ends, the ReplyDescriptor bit is cleared to "0," and the D_EP0IntStat register DescriptorCmp bit and D_EP0IntStat register IN_TranACK bit are set to "1."

For further details, refer to the section that describes how functions are used.

3.2.42 Port0:0F2h / Port1:N/A D_EP0ControlIN (Device EP0 Control IN)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0F2h	D_EP0ControlIN		7:	0:	1:	00h
			R/W	6: EnShortPkt	0: Do nothing	1: Enable short Packet	
				5:	0:	1:	
			R	4: ToggleStat	Toggle sequence bit		
			W	3: ToggleSet	0: Do nothing	1: Set Toggle sequence bit	
			W	2: ToggleClr	0: Do nothing	1: Clear Toggle sequence bit	
			R/W	1: ForceNAK	0: Do nothing	1: Force NAK	
			R/W	0: ForceSTALL	0: Do nothing	1: Force STALL	

This indicates the operation settings and status for endpoint EP0 IN transactions.

Bit7 Reserved

Bit6 EnShortPkt

Setting this bit to "1" allows FIFO data smaller than MaxPacketSize to be sent as a short packet to the endpoint EP0 IN transaction. This bit is automatically cleared to "0" on completion of the IN transaction sending the short packet. This bit is not cleared if a MaxPacketSize packet is sent.

Setting this bit to "1" when there is no data in the FIFO allows a zero-length packet to be sent to the IN token from the host. If data is written to the corresponding FIFO while this bit is set and packets are being sent, that data may also be sent, depending on the precise timing. Do not write data to the FIFO until the packet has been sent and this bit has been cleared.

Bit5 Reserved

Bit4 ToggleStat

This indicates the endpoint EP0 IN transaction toggle sequence bit state.

Bit3 ToggleSet

This sets the endpoint EP0 IN transaction toggle sequence bit to "1." If set at the same time as the ToggleClr bit, the ToggleClr bit function is given precedence.

Bit2 ToggleClr

This clears the endpoint EP0 IN transaction toggle sequence bit to "0." If set at the same time as the ToggleSet bit, this bit function is given precedence.

Bit1 ForceNAK

Setting this bit to “1” returns a NAK response to the endpoint EP0 IN transaction, regardless of FIFO data quantity.

The USB_DeviceIntStat register RcvEP0SETUP bit is set to “1” on completion of the setup stage. This bit is then set to “1” and cannot be cleared to “0” while the D_SETUP_Control.ProtectEP0 bit is “1.” Similarly, this bit is set to “1” on completion of the IN transaction sending a short packet.

If the transaction is already running when this bit is set to “1,” the bit is not set until the transaction ends, after which it is set to “1.” The bit is set to “1” immediately if the transaction is not underway.

Bit0 ForceSTALL

Setting this bit to “1” returns a STALL response to the endpoint EP0 IN transaction. This bit takes precedence over the ForceNAK bit setting.

If the USB_DeviceIntStat register RcvEP0SETUP bit is set to “1” on completion of the setup stage, this bit is cleared to “0” and cannot be set to “1” while the D_SETUP_Control.ProtectEP0 bit is “1.”

If a transaction is underway, this bit setting will be enabled from the subsequent transaction for a preset duration after the start of the transaction.

3.2.43 Port0:0F3h / Port1:N/A D_EP0ControlOUT (Device EP0 Control OUT)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0F3h	D_EP0ControlOUT	R/W	7: AutoForceNAK	0: Do nothing	1: Auto Force NAK	00h
				6:	0:	1:	
				5:	0:	1	
			R	4: ToggleStat	Toggle sequence bit		
			W	3: ToggleSet	0: Do nothing	1: Set Toggle sequence bit	
			W	2: ToggleClr	0: Do nothing	1: Clear Toggle sequence bit	
			R/W	1: ForceNAK	0: Do nothing	1: Force NAK	
			R/W	0: ForceSTALL	0: Do nothing	1: Force STALL	

This indicates the operation settings and status for endpoint EP0 OUT transactions.

Bit7 AutoForceNAK

This sets the register ForceNAK bit to "1" once the endpoint EP0 OUT transaction ends normally.

Bit6-5 Reserved

Bit4 ToggleStat

This indicates the endpoint EP0 OUT transaction toggle sequence bit state.

Bit3 ToggleSet

This sets the endpoint EP0 OUT transaction toggle sequence bit to "1." If set at the same time as the ToggleClr bit, the ToggleClr bit function is given precedence.

Bit2 ToggleClr

This clears the endpoint EP0 OUT transaction toggle sequence bit to "0." If set at the same time as the ToggleSet bit, this bit function is given precedence.

Bit1 ForceNAK

Setting this bit to "1" returns a NAK response to the endpoint EP0 OUT transaction, regardless of the FIFO free space.

The USB_DeviceIntStat register RcvEP0SETUP bit is set to "1" on completion of the setup stage. This bit is then set to "1" and cannot be cleared to "0" while the D_SETUP_Control.ProtectEP0 bit is "1."

If the transaction is already running when this bit is set to "1," the bit is not set until the transaction ends. It is set to "1" once the transaction ends. The bit is set to "1" immediately if the transaction is not underway.

Bit0 ForceSTALL

Setting this bit to “1” returns a STALL response to the endpoint EP0 OUT transaction. This bit takes precedence over the ForceNAK bit setting.

If the USB_DeviceIntStat register RcvEP0SETUP bit is set to “1” on completion of the setup stage, this bit is cleared to “0” and cannot be set to “1” while the D_SETUP_Control.ProtectEP0 bit is “1.”

If a transaction is underway, this bit setting will be enabled from the subsequent transaction for a preset duration after the start of the transaction.

3.2.44 Port0:0F8h-0F9h / Port1:N/A D_EPaMaxSize_H,L (Device EPa Max Packet Size High, Low)
3.2.45 Port0:100h-101h / Port1:N/A D_EPbMaxSize_H,L (Device EPb Max Packet Size High, Low)
3.2.46 Port0:108h-109h / Port1:N/A D_EPcMaxSize_H,L (Device EPc Max Packet Size High, Low)
3.2.47 Port0:110h-111h / Port1:N/A D_EPdMaxSize_H,L (Device EPd Max Packet Size High, Low)
3.2.48 Port0:118h-119h / Port1:N/A D_EPeMaxSize_H,L (Device EPe Max Packet Size High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0F8h-0F9h	D_EPaMaxSize_H,L		15:	0:	1:	
	100h-101h	D_EPbMaxSize_H,L		14:	0:	1:	
	108h-109h	D_EPcMaxSize_H,L		13:	0:	1:	
	110h-111h	D_EPdMaxSize_H,L		12:	0:	1:	
	118h-119h	D_EPeMaxSize_H,L		11:	0:	1:	
			R/W	10: MaxSize[10]	Endpoint [x] Max Packet Size		0000h
				9: MaxSize [9]			
				8: MaxSize [8]			
				7: MaxSize [7]			
				6: MaxSize [6]			
				5: MaxSize [5]			
				4: MaxSize [4]			
				3: MaxSize [3]			
				2: MaxSize [2]			
				1: MaxSize [1]			
				0: MaxSize [0]			

These set MaxPacketSize.

Bit15-11 Reserved

Bit10-0 MaxSize[10:0]

This sets MaxPacketSize for endpoint EPx{x=a-e}.

Use any one of the following settings when using this endpoint for bulk transfers.

FS: 8, 16, 32, or 64 bytes

HS: 512 bytes

The transfer quantity can be set as required within the ranges shown below when using the endpoint for interrupt transfers.

FS: Up to 64 bytes

HS: Up to 1,024 bytes

The transfer quantity can be set as required within the ranges shown below when using the endpoint for isochronous transfers.

FS: 1 to 1,023 bytes

HS: 1 to 1,024 bytes

3.2.49 Port0:0FAh / Port1:N/A D_EPaConfig (Device EPa Configuration)
3.2.50 Port0:102h / Port1:N/A D_EPbConfig (Devie EPb Configuration)
3.2.51 Port0:10Ah / Port1:N/A D_EPcConfig (Device EPc Configuration)
3.2.52 Port0:112h / Port1:N/A D_EPdConfig (Devie EPd Configuration)
3.2.53 Port0:11Ah / Port1:N/A D_EPeConfig (Device EPe Configuration)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0FAh 102h 10Ah 112h 11Ah	D_EPaConfig D_EPbConfig D_EPcCOnfig D_EPdConfig D_EPeConfig	R/W	7: INxOUT	0: OUT	1: IN	00h
			R/W	6: IntEP_Mode	0: Normal Toggle (IN) 0: Bulk OUT (OUT)	1: Always Toggle (IN) 1: Interrupt OUT (OUT)	
			R/W	5: ISO	0: Not Isochronous	1: Isochronous	
				4:	0:	1:	
			R/W	3: EndpointNumber [3]	Endpoint Number		
				2: EndpointNumber [2]			
				1: EndpointNumber [1]			
				0: EndpointNumber [0]			

These set the endpoint EPx{x=a-e}.

These should be set to ensure that the EndpointNumber and INxOUT combination does not duplicate other endpoints.

Bit7 INxOUT

This sets the endpoint transfer direction.

Bit6 IntEP_Mode

This sets the interrupt transfer.

Do not set this bit to “1” for bulk endpoints.

This bit setting will vary depending on the endpoint direction (IN or OUT) (the endpoint direction is set by Bit7 “INxOUT”).

The toggle sequence bit operating mode is set for the IN direction (INxOUT = 1). The toggle sequence operating mode depends on the application. Select the operating mode for the Interrupt IN endpoint.

0: Normal toggle – Performs the normal toggle sequence.

1: Always toggle – Toggles normally for each transaction.

For detailed information on this mode, refer to section 5.7.5 of the USB 2.0 standards.

Whether or not PING flow control is used for this endpoint is set for the OUT direction (INxOUT = 0). Set this bit to “1” for the Interrupt OUT endpoint.

0: Bulk OUT – Set for Bulk OUT endpoint.

1: Interrupt OUT – Set for Interrupt OUT endpoint.

Bit5 ISO

Set to “1” for Isochronous transfer. Set to “0” for endpoints using bulk transfer or interrupt transfer.

Bit4 Reserved

Bit3-0 EndpointNumber

This sets an endpoint number between 0x1 and 0xF.

3.2.54 Port0:0FCh / Port1:N/A D_EPaControl (Device EPa Control)
3.2.55 Port0:104h / Port1:N/A D_EPbControl (Device EPb Control)
3.2.56 Port0:10Ch / Port1:N/A D_EPcControl (Device EPc Control)
3.2.57 Port0:114h / Port1:N/A D_EPdControl (Device EPd Control)
3.2.58 Port0:11Ch / Port1:N/A D_EPeControl (Device EPc Control)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	0FCh	D_EPaControl	R/W	7: AutoForceNAK	0: Do nothing	1: Auto Force NAK	00h
	104h	D_EPbControl	R/W	6: EnShortPkt	0: Do nothing	1: Enable Short Packet	
	10Ch	D_EPcControl	R/W	5: DisAF_NAK_Short	0: Auto Force NAK Short	1 Disable Auto Force	
	114h	D_EPdControl	R	4: ToggleStat	Toggle sequence bit		
	11Ch	D_EPeControl	W	3: ToggleSet	0: Do nothing	1: Set Toggle sequence bit	
			W	2: ToggleClr	0: Do nothing	1: Clear Toggle sequence bit	
			R/W	1: ForceNAK	0: Do nothing	1: Force NAK	
			R/W	0: ForceSTALL	0: Do nothing	1: Force STALL	

These set the endpoint EPx{x=a-e} operation.

Bit7 AutoForceNAK

This sets the register ForceNAK bit to “1” once the endpoint EPx{x=a-e} transaction ends normally.

Bit6 EnShortPkt

Setting this bit to “1” allows FIFO data smaller than MaxPacketSize to be sent as a short packet to the endpoint EPx{x=a-e} IN transaction. This bit is automatically cleared to “0” on completion of the IN transaction sending the short packet. This bit is not cleared if a MaxPacketSize packet is sent.

Setting this bit to “1” when there is no data in the FIFO allows a zero-length packet to be sent to the IN token from the host. If data is written to the corresponding FIFO while this bit is set and packets are being sent, that data may also be sent, depending on the precise timing. Do not write data to the FIFO until the packet has been sent and this bit has been cleared.

Bit5 DisAF_NAK_Short

This sets whether the Auto Force NAK Short (AF_NAK_Short*) function is enabled or disabled.

* This automatically sets the ForceNAK bit to “1” if a short packet is received at the end of a normal OUT transaction.

The default setting is AF_NAK_Short function enabled.

Setting this bit to “1” disables the AF_NAK_Short function.

If the AutoForceNAK bit is set to “1,” the AutoForceNAK bit takes precedence.

Bit4 ToggleStat

This indicates the endpoint EPx{x=a-e} toggle sequence bit state.

Bit3 ToggleSet

This sets the endpoint EPx{x=a-e} toggle sequence bit to "1." If set at the same time as the ToggleClr bit, the ToggleClr bit function is given precedence.

Bit2 ToggleClr

This clears the endpoint EPx{x=a-e} toggle sequence bit to "0." If set at the same time as the ToggleSet bit, this bit function is given precedence.

Bit1 ForceNAK

Setting this bit to "1" returns a NAK response to the endpoint EPx{x=a-e} transaction, regardless of FIFO data quantity or free space.

If the transaction is already running when this bit is set to "1," the bit is not set until the transaction ends. It is set to "1" once the transaction ends. The bit is set to "1" immediately if the transaction is not underway.

Bit0 ForceSTALL

Setting this bit to "1" returns a STALL response to the endpoint EPx{x=a-e} transaction. This bit takes precedence over the ForceNAK bit setting.

If a transaction is underway, this bit setting will be enabled from the subsequent transaction for a preset duration after the start of the transaction.

3.2.59 Port0:120h-121h / Port1:N/A D_DescAdrs_H,L (Device Descriptor Address High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	120h-121h	D_DescAdrs_H,L		15:	0:	1:	0000h
				14:	0:	1:	
				13:	0:	1:	
			R/W	12: DescAdrs[12]	Descriptor Address		
				11: DescAdrs [11]			
				10: DescAdrs [10]			
				9: DescAdrs [9]			
				8: DescAdrs [8]			
				7: DescAdrs [7]			
				6: DescAdrs [6]			
				5: DescAdrs [5]			
				4: DescAdrs [4]			
				3: DescAdrs [3]			
				2: DescAdrs [2]			
				1: DescAdrs [1]			
				0: DescAdrs [0]			

This specifies the descriptor address.

Bit15-13 Reserved

Bit12-0 DescAdrs[12:0]

This specifies the initial FIFO address for the descriptor reply function when starting the descriptor reply operation.

The descriptor address does not assign FIFO areas to the descriptor reply function. The descriptor address can specify all areas of the FIFO from 0x0000 to 0x11FF (4.5 kbytes), regardless of FIFO area settings.

DescAdrs is updated by the amount of data transmitted each time an IN transaction is completed for endpoint EP0 when returning the descriptor reply. For detailed information on the descriptor reply function, refer to D_EP0Control register ReplyDescriptor section.

Since the FIFO area for the descriptor reply function is not explicitly assigned, avoid duplication with the FIFO for other endpoints using D_DescAdrs_H,L and D_DescSize_H,L register specifications. Ideally, this should be from the address (0x0030) after the CSW area has been reserved to the first address of the area reserved in AREA0 to 5.

3.2.60 Port0:122h-123h / Port1:N/A D_DescSize_H,L (Device Descriptor Size High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	122h-123h	D_DescSize_H,L		15:	0:	1:	0000h
				14:	0:	1:	
				13:	0:	1:	
				12:	0:	1:	
				11:	0:	1:	
				10:	0:	1:	
			R/W	9: DescSize [9]	Descriptor Size		
				8: DescSize [8]			
				7: DescSize [7]			
				6: DescSize [6]			
				5: DescSize [5]			
				4: DescSize [4]			
				3: DescSize [3]			
				2: DescSize [2]			
				1: DescSize [1]			
				0: DescSize [0]			

This specifies the descriptor size.

Bit15-10 Reserved

Bit9-0 DescSize[9:0]

Descriptor size specifies the total data quantity returned by the descriptor reply function. For detailed information on the descriptor reply function, refer to the D_EP0Control register ReplyDescriptor bit section.

The value specified for descriptor size can range from 0x000 to 0x3FF, regardless of FIFO size or area settings. DescSize is updated by the amount of data transmitted each time an IN transaction is completed for endpoint EP0 when returning the descriptor reply.

Since the FIFO area for the descriptor reply function is not explicitly assigned, avoid duplication with the FIFO for other endpoints using D_DescAdrs_H,L and D_DescSize_H,L register specifications. Ideally, this should be from the address (0x0030) after the CSW area has been reserved to the first address of the area reserved in AREA0 to 5.

3.2.61 Port0:126h / Port1:N/A D_EP_DMA_Ctrl (Device EP DMA Control)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	126h	D_EP_DMA_Ctrl	R	7: FIFO_Running	0: FIFO is not running	1: FIFO is running	00h
			R/W	6: AutoEnShort	0: Do nothing	1: Auto Enable Short Packet	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
				0:	0:	1:	

This indicates and sets the FIFO state during DMA transfers.

Bit7 FIFO_Running

This indicates that the endpoint FIFO connected to the DMA is operating. It is set to "1" when DMA is started and is cleared to "0" when the FIFO becomes empty after DMA ends.

Bit6 AutoEnShort

This sets the corresponding endpoint EnShortPkt bit to "1" if the amount of data remaining in the FIFO is less than the maximum packet size after DMA ends.

It is enabled when the endpoint connected to DMA is in the IN direction.

Bit5-0 Reserved

3.2.62 Port0:128h-129h / Port1:N/A D_EnEP_IN_H,L (Device Enable Endpoint-IN High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	128h-129h	D_EnEP_IN_H,L	R/W	15: EnEP15IN	0: Disable	1: Enable	0000h
			R/W	14: EnEP14IN	0: Disable	1: Enable	
			R/W	13: EnEP13IN	0: Disable	1: Enable	
			R/W	12: EnEP12IN	0: Disable	1: Enable	
			R/W	11: EnEP11IN	0: Disable	1: Enable	
			R/W	10: EnEP10IN	0: Disable	1: Enable	
			R/W	9: EnEP9IN	0: Disable	1: Enable	
			R/W	8: EnEP8IN	0: Disable	1: Enable	
			R/W	7: EnEP7IN	0: Disable	1: Enable	
			R/W	6: EnEP6IN	0: Disable	1: Enable	
			R/W	5: EnEP5IN	0: Disable	1: Enable	
			R/W	4: EnEP4IN	0: Disable	1: Enable	
			R/W	3: EnEP3IN	0: Disable	1: Enable	
			R/W	2: EnEP2IN	0: Disable	1: Enable	
			R/W	1: EnEP1IN	0: Disable	1: Enable	
				0:	0:	1:	

This sets the endpoint to be enabled as IN when in device mode.

The response shown below is returned if a transaction is issued (IN token received) from the host to the endpoints enabled by D_EnEP_IN_H,L, except those appropriately set in the D_EPx{x=0,a-e} registers and for which joining is set by the AREAn{n=0-5}Join.JoinEPxCHx{x=0,a-e} bit.

The response returned to the IN token depends on the D_EnEP_IN_ISO_H,L setting. A zero-length packet is returned to the host for an endpoint for which the corresponding bit is set to “1.” A NAK is returned to the host for an endpoint cleared to “0.”

3.2.63 Port0:12Ah-12Bh / Port1:N/A D_EnEP_OUT_H,L (Device Enable Endpoint-OUT High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	12Ah-12Bh	D_EnEP_OUT_H,L	R/W	15: EnEP15OUT	0: Disable	1: Enable	0000h
			R/W	14: EnEP14OUT	0: Disable	1: Enable	
			R/W	13: EnEP13OUT	0: Disable	1: Enable	
			R/W	12: EnEP12OUT	0: Disable	1: Enable	
			R/W	11: EnEP11OUT	0: Disable	1: Enable	
			R/W	10: EnEP10OUT	0: Disable	1: Enable	
			R/W	9: EnEP9OUT	0: Disable	1: Enable	
			R/W	8: EnEP8OUT	0: Disable	1: Enable	
			R/W	7: EnEP7OUT	0: Disable	1: Enable	
			R/W	6: EnEP6OUT	0: Disable	1: Enable	
			R/W	5: EnEP5OUT	0: Disable	1: Enable	
			R/W	4: EnEP4OUT	0: Disable	1: Enable	
			R/W	3: EnEP3OUT	0: Disable	1: Enable	
			R/W	2: EnEP2OUT	0: Disable	1: Enable	
			R/W	1: EnEP1OUT	0: Disable	1: Enable	
				0:	0:	1:	

This sets the endpoint to be enabled as OUT when in device mode.

The response shown below is returned and the corresponding bit is set to “1” if a transaction is issued (OUT token received) from the host to the endpoints enabled by D_EnEP_OUT_H,L, except those appropriately set in the D_EPx{x=0,a-e} registers and for which joining is set by the AREAn{n=0-5}Join.JoinEPxCHx{x=0,a-e} bit.

The response returned to the OUT token depends on the D_EnEP_OUT_ISO_H,L setting. The data transmitted by the host is not accepted, and no handshake is returned for an endpoint for which the corresponding bit is set to “1”; a NAK is returned to the host for an endpoint cleared to “0.” A NAK is returned if the device is set to HS and a PING token is issued by the host.

If the corresponding bit in this register is set, enable transactions by appropriately setting the D_EPx{x=0,a-e} related registers and joining the endpoint to the FIFO area using the AREAn{n=0-5}Join.JoinEPxCHx{x=0,a-e} bit.

3.2.64 Port0:12Ch-12Dh / Port1:N/A D_EnEP_IN_ISO_H,L (Device Enable Endpoint-IN Isochronous High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	12Ch-12Dh	D_EnEP_IN	R/W	15: EnEP15IN_ISO	0: Disable	1: Enable	0000h
		_ISO_H,L	R/W	14: EnEP14IN_ISO	0: Disable	1: Enable	
			R/W	13: EnEP13IN_ISO	0: Disable	1: Enable	
			R/W	12: EnEP12IN_ISO	0: Disable	1: Enable	
			R/W	11: EnEP11IN_ISO	0: Disable	1: Enable	
			R/W	10: EnEP10IN_ISO	0: Disable	1: Enable	
			R/W	9: EnEP9IN_ISO	0: Disable	1: Enable	
			R/W	8: EnEP8IN_ISO	0: Disable	1: Enable	
			R/W	7: EnEP7IN_ISO	0: Disable	1: Enable	
			R/W	6: EnEP6IN_ISO	0: Disable	1: Enable	
			R/W	5: EnEP5IN_ISO	0: Disable	1: Enable	
			R/W	4: EnEP4IN_ISO	0: Disable	1: Enable	
			R/W	3: EnEP3IN_ISO	0: Disable	1: Enable	
			R/W	2: EnEP2IN_ISO	0: Disable	1: Enable	
			R/W	1: EnEP1IN_ISO	0: Disable	1: Enable	
				0:	0:	1:	

This sets as ISO mode the endpoint to be enabled as IN when in device mode.

The response shown below is returned if a transaction is issued (IN token received) from the host to the endpoints enabled by D_EnEP_IN_H,L, except those appropriately set in the D_EPx{x=0,a-e} registers and for which joining is set by the AREAn{n=0-5}Join.JoinEPxCHx{x=0,a-e} bit.

The response returned to the IN token depends on the D_EnEP_IN_ISO_H,L setting. A zero-length packet is returned to the host for an endpoint for which the corresponding bit is set to “1.” A NAK is returned to the host for an endpoint cleared to “0.”

3.2.65 Port0:12Eh-12Fh / Port1:N/A D_EnEP_OUT_ISO_H,L (Device Enable Endpoint-OUT Isochronous High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Device	12Eh-12Fh	D_EnEP_OUT	R/W	15: EnEP15OUT_ISO	0: Disable	1: Enable	0000h
		_ISO_H,L	R/W	14: EnEP14OUT_ISO	0: Disable	1: Enable	
			R/W	13: EnEP13OUT_ISO	0: Disable	1: Enable	
			R/W	12: EnEP12OUT_ISO	0: Disable	1: Enable	
			R/W	11: EnEP11OUT_ISO	0: Disable	1: Enable	
			R/W	10: EnEP10OUT_ISO	0: Disable	1: Enable	
			R/W	9: EnEP9OUT_ISO	0: Disable	1: Enable	
			R/W	8: EnEP8OUT_ISO	0: Disable	1: Enable	
			R/W	7: EnEP7OUT_ISO	0: Disable	1: Enable	
			R/W	6: EnEP6OUT_ISO	0: Disable	1: Enable	
			R/W	5: EnEP5OUT_ISO	0: Disable	1: Enable	
			R/W	4: EnEP4OUT_ISO	0: Disable	1: Enable	
			R/W	3: EnEP3OUT_ISO	0: Disable	1: Enable	
			R/W	2: EnEP2OUT_ISO	0: Disable	1: Enable	
			R/W	1: EnEP1OUT_ISO	0: Disable	1: Enable	
				0:	0:	1:	

This sets the endpoint to be enabled as OUT when in device mode.

The response shown below is returned and the corresponding bit is set to “1” if a transaction is issued (OUT token received) from the host to the endpoints enabled by D_EnEP_OUT_H,L, except those appropriately set in the D_EPx{x=0,a-e} registers and for which joining is set by the AREAn{n=0-5}Join.JoinEPxCHx{x=0,a-e} bit.

The response returned to the OUT token depends on the D_EnEP_OUT_ISO_H,L setting. The data transmitted by the host is not accepted, and no handshake is returned for an endpoint for which the corresponding bit is set to “1”; a NAK is returned to the host for an endpoint cleared to “0.” A NAK is returned if the device is set to HS and a PING token is issued by the host.

If the corresponding bit in this register is set, enable transactions by appropriately setting the D_EPx{x=0,a-e} related registers and joining the endpoint to the FIFO area using the AREAn{n=0-5}Join.JoinEPxCHx{x=0,a-e} bit.

3.3 Host Register Details

Registers and register bits related to host functions are defined for both Port 0 and Port 1.

Addresses are shown as offset addresses from the base address 000h for Port 0 and from the base address 200h for Port 1.

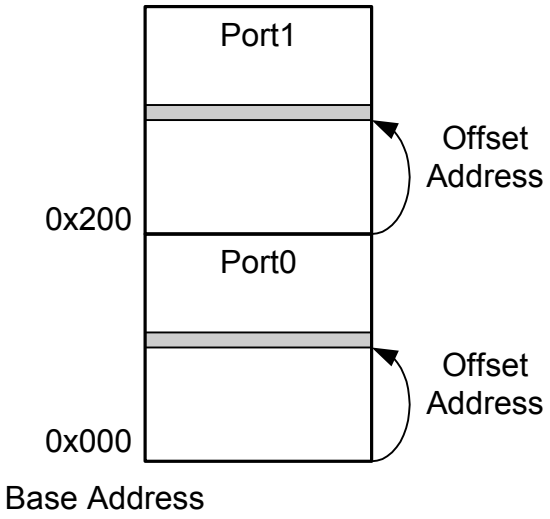


Fig. 3-2 Address notation

3.3.1 Port0:140h / Port1:340h H_SIE_IntStat_0 (Host SIE Interrupt Status 0)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	140h	H_SIE_IntStat_0		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
			R (W)	4: DetectCon	0: None	1: Detect Connect	
			R (W)	3: DetectDiscon	0: None	1: Detect Disconnect	
			R (W)	2: DetectRmtWkup	0: None	1: Detect Remote WakeUp	
			R (W)	1: DetectDevChirpOK	0: None	1: Detect Device Chirp OK	
			R (W)	0: DetectDevChirpNG	0: None	1: Detect Device Chirp NG	

This indicates host SIE-related interrupts.

Writing "1" to each bit clears the interrupt factors.

Bit7-5 Reserved

Bit4 DetectCon

Directly specifies interrupt factors.

Set to "1" if the USB cable is detected as being connected.

Bit3 DetectDiscon

Directly specifies interrupt factors.

Set to "1" if the USB cable is detected as disconnected.

Bit2 DetectRmtWkup

Directly specifies interrupt factors.

Set to "1" if a Remote WakeUp signal is detected from the device in SUSPEND state.

Bit1 DetectDevChirpOK

Directly specifies interrupt factors.

Set to "1" if a normal chirp signal is received from the device.

The hardware determines the presence or absence of a chirp signal from the device using this bit when the host is in RESET state. H_NegoControl_1.PortSpeed is set to "00b(HS)" if this bit is set to "1", and subsequent processing corresponds to HS devices. This bit must therefore always be cleared to "0" when the device is disconnected.

Bit0 DetectDevChirpNG

Directly specifies interrupt factors.

Set to "1" if an error chirp signal is received from the device.

The synchronous bits (Bit4 to 0) cannot be read from or written to (to clear interrupt factors) unless the HostDeviceSel.HOSTxDEVICE bit is “1” (i.e., in HOST mode) even if power management is ACTIVE. The firmware should perform the following processing to prevent assertion of the interrupt signal XINT by the interrupt status when switching from this state.

<Switching from HOST mode in the ACTIVE state>

1) Process and clear the interrupt status (H_SIE_IntStat_0.Bit4 to 0)

2) Disable the interrupt status (H_SIE_IntEnb_0.Bit4 to 0)

<Switching to HOST mode in the ACTIVE state>

1) Clear the interrupt status (H_SIE_IntStat_0.Bit4 to 0)

2) Enable the interrupt status (H_SIE_IntEnb_0.Bit4 to 0)

3.3.2 Port0:141h / Port1:341h H_SIE_IntStat_1 (SIE Host Interrupt Status 1)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	141h	H_SIE_IntStat_1		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1	
				4:	0:	1:	
			R (W)	3: DisabledCmp	0: None	1: Disabled Complete	
			R (W)	2: ResumeCmp	0: None	1: Resume Complete	
			R (W)	1: SuspendCmp	0: None	1: Suspend Cmplete	
			R (W)	0: ResetCmp	0: None	1: Reset Cmplete	

This indicates host SIE-related interrupts. Writing "1" to each bit clears the interrupt factors.

Bit7-4 Reserved

Bit3 DisabledCmp

Directly specifies interrupt factors.

Set to "1" **on completion of the switch to the DISABLED state** when H_NegoControl_0.AutoMode[3:0] is set to GoDISABLED and the state management function has run.

Bit2 ResumeCmp

Directly specifies interrupt factors.

Set to "1" **on successful completion of the switch to RESUME state** when H_NegoControl_0.AutoMode[3:0] is set to GoRESUME and the state management function has run.

Bit1 SuspendCmp

Directly specifies interrupt factors.

Set to "1" on completion of the switch to SUSPEND state when H_NegoControl_0.AutoMode[3:0] is set to GoSUSPEND and the state management function has run.

Bit0 ResetCmp

Directly specifies interrupt factors.

Set to "1" on successful completion of the switch to the USB RESET state when H_NegoControl_0.AutoMode[3:0] is set to GoRESET and the state management function has run.

The synchronous bits (Bit3 to 0) cannot be read from or written to (to clear interrupt factors) unless the HostDeviceSel.HOSTxDEVICE bit is "1" (i.e., in HOST mode), even if power management is ACTIVE.

The firmware should perform the following processing to prevent assertion of the interrupt signal XINT by the interrupt status when switching from this state.

<Switching from HOST mode in ACTIVE state>

1) Process and clear the interrupt status (H_SIE_IntStat_1.Bit3 to 0)

2) Disable the interrupt status (H_SIE_IntEnb_1.Bit3 to 0)

<Switching to HOST mode in ACTIVE state>

1) Clear the interrupt status (H_SIE_IntStat_1.Bit3 to 0)

2) Enable the interrupt status (H_SIE_IntEnb_1.Bit3 to 0)

3.3.3 Port0:143h / Port1:343h H_FrameIntStat (Host Frame Interrupt Status)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	143h	H_FrameIntStat		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
			R (W)	2: PortErr	0: None	1: Port Error	
			R (W)	1: FrameNumOver	0: None	1: Frame Number Over	
			R (W)	0: SOF	0: None	1: SOF	

This indicates host frame-related interrupts.

Writing “1” to each bit clears the interrupt factors.

Bit7-3 Reserved

Bit2 PortErr

Directly specifies interrupt factors.

Set to “1” if a port error is detected during host operations.

Bit2 FrameNumOver

Directly specifies interrupt factors.

Set to “1” when the frame number counter overflows (FrameNumber_H registerMSb (bit 2) changes from “1” to “0”). Any shortfall in the number of count digits in the FrameNumber_H,L registers can be offset by counting this interrupt.

Bit0 SOF

Directly specifies interrupt factors.

Set to “1” in the cases shown below, based on the transfer speed.

HS: When the host controller issues a micro frame 0 SOF token

FS: When the host controller issues an SOF token

LS: When the host controller issues keepalive

3.3.4 Port0:144h / Port1:344h H_CHrIntStat (Host CHr Interrupt Status)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	144h	H_CHrIntStat		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
			R	4: H_CHeIntStat	0: None	1: CHe Interrupt	
			R	3: H_CHdIntStat	0: None	1: CHd Interrupt	
			R	2: H_CHcIntStat	0: None	1: CHc Interrupt	
			R	1: H_CHbIntStat	0: None	1: CHb Interrupt	
			R	0: H_CHaIntStat	0: None	1: CHa Interrupt	

This indicates channel CHr interrupts.

Bit7-5 Reserved

Bit4 H_CHeIntStat

Indirectly specifies interrupt factors.

Set to "1" when the H_CHeIntStat register contains an interrupt factor and the H_CHeIntEnb register bit corresponding to that interrupt factor is enabled.

Bit3 H_CHdIntStat

Indirectly specifies interrupt factors.

Set to "1" when the H_CHdIntStat register contains an interrupt factor and the H_CHdIntEnb register bit corresponding to that interrupt factor is enabled.

Bit2 H_CHcIntStat

Indirectly specifies interrupt factors.

Set to "1" when the H_CHcIntStat register contains an interrupt factor and the H_CHcIntEnb register bit corresponding to that interrupt factor is enabled.

Bit1 H_CHbIntStat

Indirectly specifies interrupt factors.

Set to "1" when the H_CHbIntStat register contains an interrupt factor and the H_CHbIntEnb register bit corresponding to that interrupt factor is enabled.

Bit0 H_CHaIntStat

Indirectly specifies interrupt factors.

Set to "1" when the H_CHaIntStat register contains an interrupt factor and the H_CHaIntEnb register bit corresponding to that interrupt factor is enabled.

3.3.5 Port0:145h / Port1:345h H_CH0IntStat (Host CH0 Interrupt Status)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	145h	H_CH0IntStat	R (W)	7: TotalSizeCmp	0: None	1: TotalSize Complete	00h
			R (W)	6: TranACK	0: None	1: Tran ACK	
			R (W)	5: TranErr	0: None	1: Tran Error	
			R (W)	4: ChangeCondition	0: None	1: Change Condition	
				3:	0:	1:	
				2:	0:	1:	
			R (W)	1: CTL_SupportCmp	0: None	1: CTL_Support Cmplete	
			R (W)	0: CTL_SupportStop	0: None	1: CTL_Support Stop	

This indicates the channel CH0 interrupt status.

Writing “1” to each bit clears the interrupt factors.

Bit7 TotalSizeCmp

Directly specifies interrupt factors.

This bit is set to “1” when the IRP packet transfer completes normally.

When the control transfer support function is running, this bit is set to “1” once the setup stage, data stage, and status stage ends normally.

Bit6 TranACK

Directly specifies interrupt factors.

This bit is set to “1” when the number of transactions set in the H_CH0Config_0.ACK_Cnt bit ends normally.

When the control transfer support function is running, this bit is set to “1” only for the data stage.

Bit5 TranErr

Directly specifies interrupt factors.

This bit is set to “1” when any one of the transactions ends in a retry error – i.e., time-out errors, CRC errors, bit stuffing errors, PID errors (including unforeseen PID), or toggle mismatch errors.

Bit4 ChangeCondition

Directly specifies interrupt factors.

This bit is set to “1” upon a condition code stall, data overrun, data underrun or upon three successive retry errors for a transaction.

This bit is also set to “1” when the H_CH0Config_0.TranGo bit is cleared by the firmware. In this case, ConditionCode indicates the final transaction results.

Code	Meaning	Description
000	NoError	Transaction completed normally.
001	Stall	Endpoint returned a stall PID.

010	DataOverrun	• The data packet received exceeds the maximum packet size.
		  * Processed as a retry error if detected at same time as a CRC or bit stuffing error.
		• The data quantity received exceeds the IRP (TotalSize).
		  * Processed as a retry error if detected at same time as a CRC or bit stuffing error.
		  * Processed as a toggle mismatch rather than a data overrun if the data packet is smaller than the maximum packet size and the data toggle included in the data packet fails to match the expected value.

011	DataUnderrun	• The data packet received is smaller than the maximum packet size; the data quantity is less than the IRP (TotalSize). * Processed as a retry error if detected at same time as a CRC or bit stuffing error.
100	RetryError	• Device fails to respond to tokens within the specified timeframe (IN) or fails to issue a handshake within the specified timeframe (OUT). • The data packet from the endpoint contains a CRC error. • The data packet from the endpoint contains a bit stuffing error. • The PID inspection bit from the endpoint failed for data PID (IN) or handshake (OUT). • The PID received is disabled or the PID value is undefined. • The data toggle contained in the data packet from the endpoint fails to match the expected value (toggle mismatch).
Other	Reserved	

Bit3-2 Reserved

Bit1 CTL_SupportCmp

Directly specifies interrupt factors.

This bit is set to “1” when all stages of the control transfer end normally with the control transfer support function.

This bit is also set to “1” if the status stage ends normally and the stop processing is completed for the control transfer support function stop processing because the H_CTL_SupportControl register CTL_SupportGo bit is cleared.

Bit0 CTL_SupportStop

Directly specifies interrupt factors.

This bit is set to “1” when the control transfer is terminated in mid-course with an error by the control transfer support function.

This bit is also set to “1” if stop processing ends for a stage other than the status stage or if the status stage ends with a transaction error for the control transfer support function stop processing because the H_CTL_SupportControl register CTL_SupportGo bit is cleared.

3.3.6 Port0:146h / Port1:346h H_CHaIntStat (Host CHa Interrupt Status)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	146h	H_CHaIntStat	R (W)	7: TotalSizeCmp	0: None	1: TotalSize Complete	00h
			R (W)	6: TranACK	0: None	1: Tran ACK	
			R (W)	5: TranErr	0: None	1: Tran Error	
			R (W)	4: ChangeCondition	0: None	1: Change Condition	
				3:	0:	1:	
				2:	0:	1:	
			R (W)	1: BO_SupportCmp	0: None	1: BO Support Cmplete	
			R (W)	0: BO_SupportStop	0: None	1: BO Support Stop	

This indicates the channel CHa interrupt status.

Writing "1" to each bit clears the interrupt factors.

Bit7 TotalSizeCmp

Directly specifies interrupt factors.

This bit is set to "1" when the IRP packet transfer completes normally.

When the bulk-only support function is running, this bit is set to "1" once CBW transport, data transport, and CSW transport end normally.

Bit6 TranACK

Directly specifies interrupt factors.

This bit is set to "1" when the number of transactions set in the H_CHaConfig_0.ACK_Cnt bit ends normally.

When the bulk-only support function is running, this bit is set to "1" only for data transport.

Bit5 TranErr

Directly specifies interrupt factors.

This bit is set to "1" when any one of the transactions ends in a retry error – i.e., time-out errors, CRC errors, bit stuffing errors, PID errors (including unforeseen PID), or toggle mismatch errors.

Bit4 ChangeCondition

Directly specifies interrupt factors.

This bit is set to "1" when a retry error occurs three times in succession for a transaction, including condition code stalls, data overruns, and data underruns.

This bit is also set to "1" when the H_CHaConfig_0.TranGo bit is cleared by the firmware. In this case, ConditionCode indicates the final transaction results.

Code	Meaning	Description
000	NoError	Transaction completed normally.
001	Stall	Endpoint returned stall PID.

010	DataOverrun	• The data packet received exceeds the maximum packet size.
		  * Processed as a retry error if detected at same time as a CRC or bit stuffing error.
		• The data quantity received exceeds the IRP (TotalSize).
		  * Processed as a retry error if detected at same time as a CRC or bit stuffing error.
		  * Processed as a toggle mismatch rather than a data overrun if the data packet is smaller than the maximum packet size and the data toggle included in the data packet fails to match the expected value.

011	DataUnderrun	• The data packet received is smaller than the maximum packet size; the data quantity is less than the IRP (TotalSize). * Processed as a retry error if detected at same time as a CRC or bit stuffing error.
100	RetryError	• Device fails to respond to tokens within the specified timeframe (IN) or fails to issue a handshake within the specified timeframe (OUT). • The data packet from the endpoint contains a CRC error. • The data packet from the endpoint contains a bit stuffing error. • The PID inspection bit from the endpoint failed for data PID (IN) or handshake (OUT). • The PID received is disabled or the PID value is undefined. • The data toggle contained in the data packet from the endpoint fails to match the expected value (toggle mismatch).
Other	Reserved	

Bit3-2 Reserved

Bit1 BO_SupportCmp

Directly specifies interrupt factors.

This bit is set to “1” when the status transport ends normally with the bulk-only support function.

This bit is also set to “1” if the CSW transport ends normally and the stop processing is completed for the bulk-only support function stop processing due to the H_CHaBO_SupportCtl register BO_SupportGo bit being cleared.

Bit0 BO_SupportStop

Directly specifies interrupt factors.

This bit is set to “1” when any transfer ends with an error by the bulk-only support function.

This bit is also set to “1” if stop processing ends for a transport other than the CSW transport during bulk-only support function stop processing because the H_CHaBO_SupportCtl register BO_SupportGo bit is cleared or if an error is detected in the CSW transport.

3.3.7 Port0:147h / Port1:347h H_CHbIntStat (Host CHb Interrupt Status)

3.3.8 Port0:148h / Port1:348h H_CHcIntStat (Host CHc Interrupt Status)

3.3.9 Port0:149h / Port1:349h H_CHdIntStat (Host CHd Interrupt Status)

3.3.10 Port0:14Ah / Port1:34Ah H_CHeIntStat (Host CHe Interrupt Status)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	147h	H_CHbIntStat	R (W)	7: TotalSizeCmp	0: None	1: TotalSize Complete	00h
	148h	H_CHcIntStat	R (W)	6: TranACK	0: None	1: Tran ACK	
	149h	H_CHdIntStat	R (W)	5: TranErr	0: None	1: Tran Error	
	14Ah	H_CHeIntStat	R (W)	4: ChangeCondition	0: None	1: Change Condition	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
				0:	0:	1:	

These indicate the interrupt status of channel CHx{x=b-e}.

Writing "1" to all the bits clears the interrupt factors.

Bit7 TotalSizeCmp

Directly specifies interrupt factors.

This bit is set to "1" when the transfer completes normally via IRP packet transfer.

Bit6 TranACK

Directly specifies interrupt factors.

This bit is set to "1" when the number of transactions set in the H_CHx{x=b-e}Config_0.ACK_Cnt bit completes normally.

Bit5 TranErr

Directly specifies interrupt factors.

This bit is set to "1" when any one of the transactions ends in a retry error – i.e., time-out errors, CRC errors, bit stuffing errors, PID errors (including unforeseen PID), or toggle mismatch errors.

Bit4 ChangeCondition

Directly specifies interrupt factors.

This bit is set to "1" when a retry error occurs three times in succession for a transaction, including condition code stalls, data overruns, and data underruns.

This bit is also set to "1" when the H_Chx{x=b-e}Config_0.TranGo bit is cleared by the firmware. In this case, ConditionCode indicates the final transaction results.

Code	Meaning	Description
000	NoError	Transaction completed normally.
001	Stall	Endpoint returned stall PID.

010	DataOverrun	• The data packet received exceeds the maximum packet size.
		  * Processed as a retry error if detected at same time as a CRC or bit stuffing error.
		• The data quantity received exceeds the IRP (TotalSize).
		  * Processed as a retry error if detected at same time as a CRC or bit stuffing error.
		  * Processed as a toggle mismatch rather than a data overrun if the data packet is smaller than the maximum packet size and the data toggle included in the data packet fails to match the expected value.

011	DataUnderrun	• The data packet received is smaller than the maximum packet size; the data quantity is less than the IRP (TotalSize). * Processed as a retry error if detected at same time as a CRC or bit stuffing error.
100	RetryError	• Device fails to respond to tokens within the specified timeframe (IN) or fails to issue a handshake within the specified timeframe (OUT). • The data packet from the endpoint contains a CRC error. • The data packet from the endpoint contains a bit stuffing error. • The PID inspection bit from the endpoint failed for data PID (IN) or handshake (OUT). • The PID received is disabled or the PID value is undefined. • ERR handshake received in interrupt transfer split transaction. • NYET handshake received three times in succession in interrupt transfer split transaction. • The data toggle contained in the data packet from the endpoint fails to match the expected value (toggle mismatch).
110	BufferOverrun	• No transaction was performed because the FIFO free space was smaller than the maximum packet size in an isochronous transfer.
111	BufferUnderrun	• No transaction was performed due to insufficient FIFO valid data in an isochronous transfer.
Other	Reserved	

Bit3-0 Reserved

3.3.11 Port0:150h / Port1:350h H_SIE_IntEnb_0 (Host SIE Interrupt Enable)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	150h	H_SIE_IntEnb_0		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
			R/W	4: EnDetectCon	0: Disable	1: Enable	
			R/W	3: EnDetectDiscon	0: Disable	1: Enable	
			R/W	2: EnDetectRmtWkup	0: Disable	1: Enable	
			R/W	1: EnDetectDevChirpOK	0: Disable	1: Enable	
			R/W	0: EnDetectDevChirpNG	0: Disable	1: Enable	

This permits or prohibits assertion of the MainIntStat register H_SIE_IntStat_0 bit using the H_SIE_IntStat_0 register interrupt factors.

3.3.12 Port0:151h / Port1:351h H_SIE_IntEnb_1(SIE Host Interrupt Enable 1)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	151h	H_SIE_IntEnb_1		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1	
				4:	0:	1: F	
			R/W	3: EnDisabledCmp	0: Disable	1: Enable	
			R/W	2: EnResumeCmp	0: Disable	1: Enable	
			R/W	1: EnSuspendCmp	0: Disable	1: Enable	
			R/W	0: EnResetCmp	0: Disable	1: Enable	

This permits or prohibits assertion of the MainIntStat register H_SIE_IntStat_1 bit using the H_SIE_IntStat_1 register interrupt factors.

3.3.13 Port0:153h / Por1:353h H_FrameIntEnb(Host Frame Interrupt Enable)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	153h	H_FrameIntEnb		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
				3:	0:	1:	
			R/W	2: EnPortErr	0: Disable	1: Enable	
			R/W	1: EnFrameNumOver	0: Disable	1: Enable	
			R/W	0: EnSOF	0: Disable	1: Enable	

This permits or prohibits assertion of the MainIntStat register H_FrameIntStat bit using the H_FrameIntStat register interrupt factors.

3.3.14 Port0:154h / Port1:354h H_CHrIntEnb(Host CHr Interrupt Enable)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	154h	H_CHrIntEnb		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
			R/W	4: EnH_CHeIntStat	0: Disable	1: Enable	
			R/W	3: EnH_CHdIntStat	0: Disable	1: Enable	
			R/W	2: EnH_CHcIntStat	0: Disable	1: Enable	
			R/W	1: EnH_CHbIntStat	0: Disable	1: Enable	
			R/W	0: EnH_CHaIntStat	0: Disable	1: Enable	

This permits or prohibits assertion of the MainIntStat register H_CHrIntStat bit using the H_CHrIntStat register interrupt factors.

3.3.15 Port0:155h / Port1:355h H_CH0IntEnb(Host CH0 Interrupt Enable)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	155h	H_CH0IntEnb	R/W	7: EnTotalSizeCmp	0: Disable	1: Enable	00h
			R/W	6: EnTranACK	0: Disable	1: Enable	
			R/W	5: EnTranErr	0: Disable	1: Enable	
			R/W	4: EnChangeCondition	0: Disable	1: Enable	
				3:	0:	1:	
				2:	0:	1:	
			R/W	1: EnCTL_SupportCmp	0: Disable	1: Enable	
			R/W	0: EnCTL_SupportStop	0: Disable	1: Enable	

This permits or prohibits assertion of the MainIntStat register H_CH0IntStat bit using the H_CH0IntStat register interrupt factors.

3.3.16 Port0:156h / Port1:356h H_CHaIntEnb (Host CHa Interrupt Enable)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	156h	H_CHaIntEnb	R/W	7: EnTotalSizeCmp	0: Disable	1: Enable	00h
			R/W	6: EnTranACK	0: Disable	1: Enable	
			R/W	5: EnTranErr	0: Disable	1: Enable	
			R/W	4: EnChangeCondition	0: Disable	1: Enable	
				3:	0:	1:	
				2:	0: Disable	1:	
			R/W	1: EnBO_Support_Cmp	0: Disable	1: Enable	
			R/W	0: EnBO_Support_Stop	0: Disable	1: Enable	

This permits or prohibits assertion of the H_CHrIntStat register CHaIntStat bit using the H_CHaIntStat register interrupt factors.

3.3.17 Port0:157h / Port1:357h H_CHbIntEnb (Host CHb Interrupt Enable)

3.3.18 Port0:158h / Port1:358h H_CHcIntEnb (Host CHc Interrupt Enable)

3.3.19 Port0:159h / Port1:359h H_CHdIntEnb (Host CHd Interrupt Enable)

3.3.20 Port0:15Ah / Port1:35Ah H_CHeIntEnb (Host CHe Interrupt Enable)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	157h	H_CHbIntEnb	R/W	7: EnTotalSizeCmp	0: Disable	1: Enable	00h
	158h	H_CHcIntEnb	R/W	6: EnTranACK	0: Disable	1: Enable	
	159h	H_CHdIntEnb	R/W	5: EnTranErr	0: Disable	1: Enable	
	15Ah	H_CHeIntEnb	R/W	4: EnChangeCondition	0: Disable	1: Enable	
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
				0:	0:	1:	

These permit or prohibit assertion of the H_CHrIntStat register CHx{x=b-e}IntStat bit using the H_CHx{x=b-e}IntStat register interrupt factors.

3.3.21 Port0:160h / Port1:360h H_NegoControl_0 (Host NegoControl 0)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	160h	H_NegoControl_0	R/W	7: AutoModeCancel	0: None	1: Cancel	1Xh
			R	6: HostState[2]	HostState[2:0]		
			R	5: HostState[1]			
			R	4: HostState[0]			
			W	3: AutoMode[3]	AutoMode[3:0]		
			W	2: AutoMode[2]			
			W	1: AutoMode[1]			
			W	0: AutoMode[0]			

This sets host negotiation operations.

Bit7 AutoModeCancel

Setting this bit to "1" stops the host state management support function and maintains that state (the H_NegoControl_0.AutoMode and H_XcvrControl settings are retained, the signal line status is retained, internal timer is stopped, and the connection, disconnection, device chirp, remote wakeup detection functions are turned off).

The host state management support function must be stopped using this bit before performing the operations listed below.

- When switching host state to IDLE state
- When switching host state to DISABLED state without waiting for the reset completion status (H_SIE_IntStat_1.ResetCmp) to be issued when a chirp error has been detected from the device
- When overwritting AutoMode
- When running test mode using H_USB_Test.EnHS_Test

Setting this bit to "1" performs host state management support function stop processing and sets this bit to "0" once stop processing is completed (requires approximately 6 cycles for a 60 MHz clock). In this case, check that the bit has been changed to "0" before setting H_NegoControl_0.AutoMode GoIDLE or GoDISABLED or before setting H_USB_Test.EnHS_Test.

Bit6-4 HostState[2:0]

This indicates the current host state when running the host state management support function. The state is indicated as one of the following:

000: Reserved

001: IDLE

010: WAIT_CONNECT

011: DISABLED

100: USB_RESET

101: USB_OPERATIONAL

110: USB_SUSPEND

111: USB_RESUME

Bit3-0 AutoMode[3:0]

This sets the new host state when running the host state management support function.

This register is write-only and sets any of the following:

0001: GoIDLE (switches to IDLE state)

0010: GoWAIT_CONNECT (switches to WAIT_CONNECT state)

0011: GoDISABLED (switches to DISABLED state)

0100: GoRESET (switches to RESET state)

0101: GoOPERATIONAL (switches to OPERATIONAL state)

0110: GoSUSPEND (switches to SUSPEND state)

0111: GoRESUME (switches to RESUME state)

1001: GoWAIT_CONNECTtoDIS (switches continuously from WAIT_CONNECT to DISABLED state)

1010: GoWAIT_CONNECTtoOP (switches continuously from WAIT_CONNECT to OPERATIONAL state)

1100: GoRESETtoOP (switches continuously from RESET to OPERATIONAL state)

1110: GoSUSPENDtoOP (switches continuously from SUSPEND to OPERATIONAL state)

1111: GoRESUMEtoOP (switches continuously from RESUME to OPERATIONAL state)

All others: Reserved

Use the following procedure to switch from any state to the IDLE state (using GoIDLE).

- Write 0x80 to the H_NegoControl_0 register.
- Confirm that the H_NegoControl_0.AutoModeCancel bit has changed to 0.
- Write 0x01 to the H_NegoControl_0 register.

3.3.22 Port0:162h / Port1:362h H_NegoControl_1 (Host NegoControl 1)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	162h	H_NegoControl_1		7:	0:	1:	10h
				6:	0:	1:	
			R/W	5: PortSpeed[1]	PortSpeed[1:0]		
			R/W	4: PortSpeed[0]			
				3:	0:	1:	
				2:	0:	1:	
			R/W	1: DisChirpFinish	0: Normal	1: DisableChirpFinish	
			R/W	0: RmtWkupDetEnb	0: Disable	1: Enable	

This sets host negotiation operations.

Note: The reset value for this register can be read when the power management state is ACTIVE. In all other states, the reset value is read as 00h.

Bit7-6 Reserved

Bit5-4 PortSpeed[1:0]

This indicates and sets the transfer speed.

00: High Speed

01: Full Speed

10: Reseved

11: Low Speed

Bit3-2 Reserved

Bit1 DisChirpFinish

This sets the operating mode when the device chirp does not end within the specified timeframe.

0: Ends the USB Reset by sending a USB Reset within the specified timeframe after raising the device chirp error status.

1: Waits for the device chirp to end after raising the device chirp error status; ends the USB Reset after running the host chirp on completion of the device chirp.

Bit0 RmtWkupDetEnb

Enables or disables the remote wakeup detection function.

3.3.23 Port0:164h / Por1:364h H_USB_Test (Host USB_Test)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	164h	H_USB_Test	R/W	7: EnHS_Test	0: Do nothing	1: EnHS_Test	00h
				6:	0:	1:	
				5:	0:	1:	
			R/W	4: Test_Force_Enable	0: Do nothing	1: Test_Force_Enable	
			R/W	3: Test_SE0_NAK	0: Do nothing	1: Test_SE0_NAK	
			R/W	2: Test_J	0: Do nothing	1: Test_J	
			R/W	1: Test_K	0: Do nothing	1: Test_K	
			R/W	0: Test_Packet	0: Do nothing	1: Test_Packett	

This sets the host USB 2.0 test mode operations.

Test mode can be run in WAIT_CONNECT, DISABLED, and SUSPEND states.

The respective state processing must be stopped before switching to test mode from these states. Use the following procedure when switching to test mode:

- Set TranGo bit (H_CHx{x=0,a-e}Config_0.TranGo), H_CTL_SupportControl.CTL_SupportGo, and H_CHaBO_SupportCtl.BOSupportGo to "0" for all channels.
- Write 0x80 to the H_NegoControl_0 register.
- Confirm that the H_NegoControl_0.AutoModeCancel bit has changed to "0."
- Set any one of the lower five bits of the register to "1" concurrently with the EnHS_Test bit.

Write 0x00 to this register when switching from one test mode to another test mode or when ending test mode. Test mode ends, and the host state switches to IDLE.

Bit7 EnHS_Test

Setting any one of the lower five bits of the H_USB_Test register to "1" concurrently with this bit switches to the test mode corresponding to the bit.

Bit6-5 Reserved

Bit4 Test_Force_Enable

Setting this bit to "1" concurrently with the EnHS_Test bit allows switching to TestForceEnable test mode. This test mode enables the host port to send SOF in HS mode and detect disconnection.

Bit3 Test_SE0_NAK

Setting this bit to "1" concurrently with the EnHS_Test bit allows switching to Test_SE0_NAK test mode. This test mode enables the host port to receive data in HS mode.

Bit2 TEST_J

Setting this bit to "1" concurrently with the EnHS_Test bit allows switching to Test_J test mode. This test mode enables the host port to send "J" in HS mode.

Bit1 TEST_K

Setting this bit to "1" concurrently with the EnHS_Test bit allows switching to Test_K test mode. This test mode enables the host port to send "K" in HS mode.

Bit0 Test_Packet

Setting this bit to "1" concurrently with the EnHS_Test bit allows switching to Test_Packet test mode. This test mode can be used only with CH0. Set the FIFO area joined to CH0 to 64 bytes before switching to this test mode, clear the FIFO area, and write the test packet data shown below to the FIFO area. Also set H_CH0Config_1.TID = 00b.

The following 53 bytes are written to the FIFO in packet transmission test mode:

00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h,

00h, AAh, AAh, AAh, AAh, AAh, AAh, AAh,

AAh, EEh, EEh, EEh, EEh, EEh, EEh, EEh,

EEh, FEh, FFh, FFh, FFh, FFh, FFh, FFh,

FFh, FFh, FFh, FFh, FFh, 7Fh, BFh, DFh,

EFh, F7h, FBh, FDh, FCh, 7Eh, BFh, DFh,

EFh, F7h, FBh, FDh, 7Eh

PID and CRC are added by SIE when sending test packets. This means that the data written to the FIFO consists of data from the following DATA0 PID data to the CRC16 data of the test packet data described in the USB standard Rev 2.0.

3.3.24 Port0:170h / Port1:370h H_CH0SETUP_0 (Host CH0 SETUP 0)
3.3.25 Port0:171h / Port1:371h H_CH0SETUP_1 (Host CH0 SETUP 1)
3.3.26 Port0:172h / Port1:372h H_CH0SETUP_2 (Host CH0 SETUP 2)
3.3.27 Port0:173h / Port1:373h H_CH0SETUP_3 (Host CH0 SETUP 3)
3.3.28 Port0:174h / Port1:374h H_CH0SETUP_4 (Host CH0 SETUP 4)
3.3.29 Port0:175h / Port1:375h H_CH0SETUP_5 (Host CH0 SETUP 5)
3.3.30 Port0:176h / Port1:376h H_CH0SETUP_6 (Host CH0 SETUP 6)
3.3.31 Port0:177h / Port1:377h H_CH0SETUP_7 (Host CH0 SETUP 7)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Host	170h -177h	H_CH0SETUP_0 -H_CH0SETUP_7	R/W	7: CH0SETUP_n [7]	Channel 0 SETUP Data 0 - Channel 0 SETUP Data 7	00h
				6: CH0SETUP_n [6]		
				5: CH0SETUP_n [5]		
				4: CH0SETUP_n [4]		
				3: CH0SETUP_n [3]		
				2: CH0SETUP_n [2]		
				1: CH0SETUP_n [1]		
				0: CH0SETUP_n [0]		

These registers set the 8 bytes of data in sequence for sending in the channel CH0 setup stage.

CH0SETUP_0

Sets BmRequestType.

CH0SETUP_1

Sets Brequest.

CH0SETUP_2

Sets the lower 8 bits of Wvalue.

CH0SETUP_3

Sets the higher 8 bits of Wvalue.

CH0SETUP_4

Sets the lower 8 bits of WIndex.

CH0SETUP_5

Sets the higher 8 bits of WIndex.

CH0SETUP_6

Sets the lower 8 bits of WLength.

CH0SETUP_7

Sets the higher 8 bits of WLength.

3.3.32 Port0:17Eh-17Fh / Port1:37Eh-37Fh H_FrameNumber_H,L (Host FrameNumber High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	17Eh-17Fh	H_FrameNumber_H,L		15:	0:	1:	07FFh
				14:	0:	1:	
				13:	0:	1:	
				12:	0:	1:	
				11:	0:	1:	
			R/W	10: FrameNumber [10]	Frame Number		
				9: FrameNumber [9]			
				8: FrameNumber [8]			
				7: FrameNumber [7]			
				6: FrameNumber [6]			
				5: FrameNumber [5]			
				4: FrameNumber [4]			
				3: FrameNumber [3]			
				2: FrameNumber [2]			
				1: FrameNumber [1]			
				0: FrameNumber [0]			

This indicates the USB frame number updated each time an SOF token is sent.

Note: The reset value for this register is the value that can be read by power management in the ACTIVE state. The reset value is read as 0000h in all other states.

Bit15-11 Reserved

Bit10-0 FrameNumber [10:0]

Indicates FrameNumber for the SOF packet to be sent.

3.3.33 Port0:180h / Port1:380h H_CH0Config_0(Host Channel 0 Configuration0)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	180h	H_CH0Config_0	R/W	7: ACK_Cnt[3]	ACK Count		00h
				6: ACK_Cnt[2]			
				5: ACK_Cnt[1]			
				4: ACK_Cnt[0]			
			R/W	3: SpeedMode[1]	Speed Mode		
				2: SpeedMode[0]			
			R/W	1: Toggle	0: Toggle0	1: Toggle1	
			R/W	0: TranGo	0: Stand by	1: Transaction Start	

This sets the basic channel CH0 settings for host operations.

Bit7-4 ACK_Cnt[3:0]

This sets the ACK count number for transfers performed with channel CH0.

The H_CH0IntStat register TranACK bit is set once the ACK number set has been counted.

0000: 16 ACK counts

0001 to 1111: 1 to 15 ACK counts

Only data stage transactions are counted when the control transfer support function is running. Setup stage and status stage transactions are not counted.

Bit3-2 SpeedMode[1:0]

This sets the operating mode for the device performing the transfer over channel CH0.

00: HS mode – Use this setting for an HS device.

01: FS mode – Use this setting for an FS device.

10: Reserved – Use of this value is prohibited.

11: LS mode – Use this setting for an LS device.

Bit1 Toggle

This sets the initial value of the toggle sequence bit at the start of a transaction. It also indicates the toggle sequence bit state after the transaction has been started or completed.

0: Toggle 0

1: Toggle 1

Bit0 TranGo

Setting this bit to “1” starts the channel CH0 transaction. After being started, the transaction process can be stopped by clearing the bit to “0.” This bit also indicates whether a transaction is running on channel CH0.

0: Stops the transaction (transaction stopped)

1: Starts the transaction (transaction running)

The H_CH0IntStat register TotalSizeCmp bit is set to “1” as soon as the number of bytes set in registers H_CH0TotalSize_H to L have been transferred. This bit automatically reverts to “0.” It is reset to “0” if the H_CH0IntStat register ChangeCondition bit has been set. The cause in this case is set to the H_CH0ConditionCode register to enable inspection.

The H_CH0IntStat register ChangeCondition bit is set as soon as a transaction in progress ends when stopped by clearing the bit. The data in the FIFO, (remaining) total size, and channel-related settings remain unchanged even when the transaction is stopped. This allows resumption of the transaction from the point at which it was stopped by resetting this bit to “1.” (To perform a new transaction, clear the FIFO and reset the channel information.)

This bit does not need to be set when using the control transfer support function.

3.3.34 Port0:181h / Port1:381h H_CH0Config_1(Host Channel 0 Configuration1)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	181h	H_CH0Config_1	R/W	7: TID[1]	Transaction ID		00h
				6: TD[0]			
				5:	0:		
				4:	0:		
				3:	0:		
				2:	0:		
				1:	0:		
				0:	0:		

This sets the basic channel CH0 settings for host operations.

Bit7-6 TID[1:0]

This sets the transaction type (SETUP/OUT/IN) to be issued with channel CH0. This bit setting is disabled when the CTL_SupportControl register CTL_SupportGo bit is set to "1" and the transaction is started.

00: SETUP – Issues a SETUP token.

01: OUT – Issues an OUT token.

10: IN – Issues an IN token.

11: Reserved – Use of this value is prohibited.

This bit does not need to be set when using the control transfer support function.

Bit5-0 Reserved

3.3.35 Port0:183h / Port1:383h H_CH0MaxPktSize (Host Channel 0 Max Packet Size)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	183h	H_CH0MaxPktSize		7:	0:	1:	00h
			R/W	6: MaxPktSize[6]	Max Packet Size		
				5: MaxPktSize[5]			
				4: MaxPktSize[4]			
				3: MaxPktSize[3]			
				2: MaxPktSize[2]			
				1: MaxPktSize[1]			
				0: MaxPktSize[0]			

This sets the channel CH0 MaxPacketSize for host operations.

Bit7 Reserved

Bit6-0 MaxPktSize[6:0]

This sets the channel CH0 MaxPacketSize.

Set to one of the following:

LS: 8bytes

FS: 8, 16, 32, 64 bytes

HS: 64 bytes

All other settings are prohibited.

3.3.36 Port0:186h-187h / Port1:386h-387h H_CH0TotalSize_H,L (Host Channel 0 Total Size High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Host	186h-187h	H_CH0TotalSize_H,L	R/W	15: TotalSize[15]	Total Size	0000h
				14: TotalSize[14]		
				13: TotalSize[13]		
				12: TotalSize[12]		
				11: TotalSize[11]		
				10: TotalSize[10]		
				9: TotalSize[9]		
				8: TotalSize[8]		
				7: TotalSize[7]		
				6: TotalSize[6]		
				5: TotalSize[5]		
				4: TotalSize[4]		
				3: TotalSize[3]		
				2: TotalSize[2]		
				1: TotalSize[1]		
				0: TotalSize[0]		

This sets the TotalSize for the data to be transferred over channel CH0 for host operations.

Bit15-0 TotalSize[15:0]

This sets the total byte count of the data to be transferred over channel CH0 (max. 65,535 bytes = approx. 64 Kbytes).

The remaining transfer quantity can be read by reading this register after the transaction has been started by the H_CH0Config_0 register TranGo bit.

A zero-length packet is issued when an OUT transaction is performed with TotalSize = 0.

This register does not need to be set when performing a SETUP transaction or when using the control transfer support function.

3.3.37 Port0:188h / Port1:388h H_CH0HubAdrs (Host Channel 0 Hub Address)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	188h	H_CH0HubAdrs	R/W	7: HubAdrs[3]	Hub Address		00h
				6: HubAdrs[2]			
				5: HubAdrs[1]			
				4: HubAdrs[0]			
				3:	0:	1:	
			R/W	2: Port[2]	Port Number		
				1: Port[1]			
				0: Port[0]			

This sets the hub connecting to channel CH0 for host operations.

Bit7 HubAdrs[3:0]

This sets the USB address of the hub to which the function performing the transfer over channel CH0 connects.

It can be set to any value from 0 to 15.

Bit3 Reserved

Bit2-0 Port[2:0]

This sets the port number of the hub to which the function performing the transfer over channel CH0 connects.

It can be set to any value from 0 to 7.

3.3.38 Port0:189h / Port1:389h H_CH0FuncAdrs (Host Channel 0 Function Address)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Host	189h	H_CH0FuncAdrs	R/W	7: FuncAdrs[3]	Function Address	00h
				6: FuncAdrs[2]		
				5: FuncAdrs[1]		
				4: FuncAdrs[0]		
			R/W	3: EP_Number[3]	Endpoint Number	
				2: EP_Number[2]		
				1: EP_Number[1]		
				0: EP_Number[0]		

This sets the address of the function performing the transfer over channel CH0 for host operations.

Bit7-4 FuncAdrs[3:0]

This sets the USB address of the function, including the endpoint managed by channel CH0.

It can be set to any value from 0 to 15.

Bit3-0 EP_Number[3:0]

This sets the endpoint number for the transfer over channel CH0.

It can be set to any value from 0 to 15.

3.3.39 Port0:18Bh / Port1:38Bh H_CTL_SupportControl (Host ControlTransfer Support Control)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	18Bh	H_CTL_Support-Control		7:	0:	1:	00h
				6:	0:	1:	
			R/W	5: CTL_SupportState[1]	ControlTransfer Support State		
				4: CTL_SupportState[0]			
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
			R/W	0: CTL_SupportGo	0: Stand by	1: Control Transfer Go	

This sets the support function for control transfers over channel CH0 for host operations.

Bit7-6 Reserved

Bit5-4 CTL_SupportState[1:0]

This sets the CTL_SupportGo to "1" and indicates the stage currently running during a transfer using the control support function.

00: Idle – Indicates that the transfer has not been started or has been completed without problems.

01: Setup Stage – Indicates that the setup stage is running.

10: Data Stage – Indicates that the data stage is running.

11: Status Stage – Indicates that the status stage is running.

Bit3-1 Reserved

Bit0 CTL_SupportGo

Setting this bit to "1" switches to the setup stage (and data stage) to the status stage automatically with the control transfer support function using channel CH0.

In the setup stage, a SETUP token is automatically issued, and the requests set in H_CH0SETUP_0 to 7 are sent.

If another data stage follows, the transaction runs automatically with the specified direction and size.

Finally, in the status stage, an appropriate PID token is automatically issued and a zero-length packet sent and received depending on the presence of a data stage and the direction.

The H_CH0IntStat register CTL_SupportCmp bit is set once the transactions above and the stage sequence end normally. If a packet error is detected in mid-sequence, the H_CH0IntStat register CTL_SupportStop bit is set, and the transaction stops. In this case, the cause is set to the ConditionCode register to enable inspection.

This bit is automatically cleared once the control transfer ends (either ends normally or in error).

The control transfer can be stopped by clearing this bit while running the control transfer support function. The CTL_SupportCmp bit is set if the control transfer ends normally in the status stage. In all other cases, the CTL_SupportStop bit is set. Refer to CTL_SupportState for the stage in which the control transfer stopped.

3.3.40 Port0:18Eh / Port1:38Eh H_CH0ConditionCode (Host Channel 0 Condition Code)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	18Eh	H_CH0ConditionCode		7:	0:	1:	00h
			R	6: ConditionCode[2]	Condition Code		
				5: ConditionCode[1]			
				4: ConditionCode[0]			
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
				0:	0:	1:	

This indicates channel CH0 transfer completion results for host operations.

Bit7 Reserved

Bit6-4 ConditionCode[2:0]

Indicates results when the transfer is completed over channel CH0.

Code	Meaning	Description
000	NOERROR	Transaction completed without errors.
001	STALL	Endpoint returned Stall PID.

010	DATAOVERRUN	• The data packet received exceeds the maximum packet size.
		  * Treated as a retry error if CRC error and bit stuffing error are detected simultaneously.
		• The data size received exceeds the IRP (TotalSize).
		  * Treated as a retry error if CRC error and bit stuffing error are detected simultaneously.
		  * Treated as a toggle mismatch rather than a data overrun if the data packet is less than the maximum packet size and the data toggle included in the data packet fails to match the expected value.

011	DATAUNDERRUN	• The data packet received is less than the maximum packet size; the data size is less than the IRP (TotalSize). * Treated as a retry error if CRC error and bit stuffing error are detected simultaneously.
100	RETRYERROR	• The device does not respond to a token (IN) or does not issue a handshake (OUT) within the specified timeframe. • The data packet from the endpoint includes a CRC error. • The data packet from the endpoint includes a bit stuffing error. • The PID inspection bit from the endpoint failed in data PID (IN) or handshake (OUT). • The PID received is invalid or no PID value is defined. • The data toggle included in the data packet from the endpoint fails to match the expected value (toggle mismatch).
Other	Reserved	

Bit3-0 Reserved

3.3.41 Port0:190h / Port1:390h H_CHaConfig_0(Host Channel a Configuration0)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	190h	H_CHaConfig_0	R/W	7: ACK_Cnt[3]	ACK Count		00h
				6: ACK_Cnt[2]			
				5: ACK_Cnt[1]			
				4: ACK_Cnt[0]			
			R/W	3: SpeedMode[1]	Speed Mode		
				2: SpeedMode[0]			
			R/W	1: Toggle	0: Toggle0	1: Toggle1	
			R/W	0: TranGo	0: Stand by	1: Transaction Start	

This sets the basic channel CHa settings for host operations.

Bit7-4 ACK_Cnt[3:0]

This sets the ACK count number for transfers performed with channel CHa.

The H_CHaIntStat register TranACK bit is set once the ACK number set has been counted.

0000: 16 ACK counts

0001 to 1111: 1 to 15 ACK counts

When the bulk-only support function is running, only data transport transactions are counted. CBW and CSW transport transactions are not counted.

Bit3-2 SpeedMode[1:0]

This sets the operating mode for the device performing the transfer over channel CHa.

00: HS mode – Use this setting for an HS device.

01: FS mode – Use this setting for an FS device.

10-11: Reserved – Use of this value is prohibited.

Bit1 Toggle

This sets the initial value of the toggle sequence bit at the start of a transaction. It also indicates the toggle sequence bit state after the transaction has been started or completed.

0: Toggle 0

1: Toggle 1

This bit does not need to be set when using the bulk-only support function.

Bit0 TranGo

Setting this bit to “1” starts the channel CHa transaction. After being started, the transaction process can be stopped by clearing the bit to “0.” This bit also indicates whether a transaction is running on channel CHa.

0: Stops the transaction (transaction stopped)

1: Starts the transaction (transaction running)

The H_CHaIntStat register TotalSizeCmp bit is set to “1” as soon as the number of bytes set in registers H_CHaTotalSize_HH to LL have been transferred. This bit automatically reverts to “0.” It is reset to “0” if the H_CHaIntStat register ChangeCondition bit has been set. In this case, the cause is set to the H_CHaConditionCode register to enable inspection.

The H_CHaIntStat register ChangeCondition bit is set as soon as a transaction in progress ends when stopped by clearing the bit. The data in the FIFO, (remaining) total size, and channel-related settings remain unchanged even when the transaction is stopped. This allows resumption of the transaction from the point at which it was stopped by resetting this bit to “1.” (To perform a new transaction, clear the FIFO and reset the channel information.)

This bit does not need to be set when using the bulk-only support function.

3.3.42 Port0:191h / Port1:391h H_CHaConfig_1(Host Channel a Configuration1)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	191h	H_CHaConfig_1	R/W	7: TID[1]	Transaction ID		00h
				6: TID[0]			
				5:	0:	1:	
				4:	0:	1:	
			R/W	3: AutoZerolen	0: Do nothing	1: Add Zerolen	
				2:	0:	1:	
				1:	0:	1:	
			R/W	0: TotalSizeFree	0: Do nothing	1: Total Size Free	

This sets the basic channel CHa settings for host operations.

Bit7-6 TID[1:0]

This sets the transaction type (OUT/IN) to be issued with channel CHa. This bit setting is disabled when the H_CHaBO_SupportCtl register BO_SupportGo bit is set to "1" and the transaction is started.

00: Reserved – Use of this value is prohibited.

01: OUT – Issues an OUT token.

10: IN – Issues an IN token.

11: Reserved – Use of this value is prohibited.

This bit does not need to be set when using the bulk-only support function.

Bit5-4 Reserved

Bit3 AutoZerolen

Setting this bit to "1" automatically adds a zero-length packet after the transfer size set in the H_CHaTotalSizeHH to LL registers ends at exactly the Max Packet Size. This bit is enabled only for OUT transfers.

Bit2-1 Reserved

Bit0 TotalSizeFree

Setting this bit to "1" cancels any restrictions on transfer size, regardless of H_CHaTotalSizeHH to LL register settings.

3.3.43 Port0:192h-193h / Port1:392h-393h H_CHaMaxPktSize_H,L (Host Channel a Max Packet Size High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	192h-193h	H_CHaMaxPktSize_H,L		15:	0:	1:	0000h
				14:	0:	1:	
				13:	0:	1:	
				12:	0:	1:	
				11:	0:	1:	
				10:	0:	1:	
			R/W	9: MaxPktSize[9]	Max Packet Size		
				8: MaxPktSize[8]			
				7: MaxPktSize[7]			
				6: MaxPktSize[6]			
				5: MaxPktSize[5]			
				4: MaxPktSize[4]			
				3: MaxPktSize[3]			
				2: MaxPktSize[2]			
				1: MaxPktSize[1]			
				0: MaxPktSize[0]			

These set the channel CHa MaxPacketSize for host operations.

Bit15-10 Reserved

Do not write “1” to reserved bits.

Bit9-0 MaxPktSize[9:0]

These set the channel CHa MaxPacketSize.

Set to one of the following:

FS: 8, 16, 32, 64 bytes (32 or 64 bytes when using bulk-only support function)

HS: 512 bytes

All other settings are prohibited.

3.3.44 Port0:194h-195h / Port1:394h-395h H_CHaTotalSize_HH,HL (Host Channel a Total Size High-High, High-Low)

3.3.45 Port0:196h-197h / Port1:396h-397h H_CHaTotalSize_LH,LL (Host Channel a Total Size Low-High, Low-Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Host	194h-195h	H_CHaTotalSize_HH,HL	R/W	15: TotalSize[31]	Total Size High	0000h
				14: TotalSize[30]		
				13: TotalSize[29]		
				12: TotalSize[28]		
				11: TotalSize[27]		
				10: TotalSize[26]		
				9: TotalSize[25]		
				8: TotalSize[24]		
				7: TotalSize[23]		
				6: TotalSize[22]		
				5: TotalSize[21]		
				4: TotalSize[20]		
				3: TotalSize[19]		
				2: TotalSize[18]		
				1: TotalSize[17]		
				0: TotalSize[16]		

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Host	196h-197h	H_CHaTotalSize_LH,LL	R/W	15: TotalSize[15]	Total Size Low	0000h
				14: TotalSize[14]		
				13: TotalSize[13]		
				12: TotalSize[12]		
				11: TotalSize[11]		
				10: TotalSize[10]		
				9: TotalSize[9]		
				8: TotalSize[8]		
				7: TotalSize[7]		
				6: TotalSize[6]		
				5: TotalSize[5]		
				4: TotalSize[4]		
				3: TotalSize[3]		
				2: TotalSize[2]		
				1: TotalSize[1]		
				0: TotalSize[0]		

These set the Total Size of the data transferred over channel CHa for host operations.

These set the total number of bytes of the data to be transferred over channel CHa (max. 4,294,967,295 bytes = approx. 4 Gbytes).

The remaining transfer quantity can be read by reading these registers after the transaction has been started by the H_CHaConfig_0 register TranGo bit.

A zero-length packet is issued when an OUT transaction is performed with TotalSize = 0.

This register does not need to be set when using the bulk-only support function.

3.3.46 Port0:198h / Port1:398h H_CHaHubAdrs (Host Channel a Hub Address)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	198h	H_CHaHubAdrs	R/W	7: HubAdrs[3]	Hub Address		00h
				6: HubAdrs[2]			
				5: HubAdrs[1]			
				4: HubAdrs[0]			
				3:	0:	1:	
			R/W	2: Port[2]	Port Number		
				1: Port[1]			
				0: Port[0]			

This sets the hub connecting to channel CHa for host operations.

Bit7 HubAdrs[3:0]

This sets the USB address of the hub to which the function performing the transfer over channel CHa connects.

It can be set to any value from 0 to 15.

Bit3 Reserved

Bit2-0 Port[2:0]

This sets the port number of the hub to which the function performing the transfer over channel CHa connects.

It can be set to any value from 0 to 7.

3.3.47 Port0:199h / Port1:399h H_CHaFuncAdrs (Host Channel a Function Address)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Host	199h	H_CHaFuncAdrs	R/W	7: FuncAdrs[3]	Function Address	00h
				6: FuncAdrs[2]		
				5: FuncAdrs[1]		
				4: FuncAdrs[0]		
			R/W	3: EP_Number[3]	Endpoint Number	
				2: EP_Number[2]		
				1: EP_Number[1]		
				0: EP_Number[0]		

This sets the address of the function performing the transfer over channel CHa for host operations.

Bit7-4 FuncAdrs[3:0]

This sets the USB address of the function, including the endpoint managed by channel CHa.

It can be set to any value from 0 to 15.

Bit3-0 EP_Number[3:0]

This sets the endpoint number for the transfer over channel CHa.

It can be set to any value from 0 to 15.

This bit does not need to be set when using the bulk-only support function.

3.3.48 Port0:19Ah / Port1:39Ah H_CHaBO_SupporotCtl (Host CHa Bulk Only Transfer Support Control)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	19Ah	H_CHaBO_Support-Ctl		7:	0:	1:	00h
				6:	0:	1:	
			R/W	5: BO_TransportState[1]	Bulk Only Transfer Transport State		
				4: BO_TransportState[0]			
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
			R/W	0: BO_SupportGo	0: Stand by	1: BO Transfer Go	

This sets the channel CHa bulk-only support function for host operations.

Bit7-6 Reserved

Bit5-4 BO_TransportState[1:0]

This sets the BO_SupportGo bit to "1" and indicates which transport is running during a transfer using the bulk-only support function.

00: Idle – Indicates that the transfer has not been started or has been completed without problems.

01: CBW Transport – Indicates that CBW transport is running.

10: Data Transport – Indicates that data transport is running.

11: CSW Transport – Indicates that CSW transport is running.

Bit3-1 Reserved

Bit0 BO_SupportGo

Setting this bit to "1" automatically performs the CBW transport (and data transport) to CSW transport with the bulk-only support function using channel CHa.

In CBW transport, an OUT token is automatically issued, and the data set in the FIFO CBW area is sent.

If a data stage follows, the data transport runs automatically with the specified direction and size.

Finally, in CSW transport, an IN token is automatically issued, and data is received in the FIFO CSW area.

The H_BO_SupportIntStat register BO_SupportCmp bit is set once the above transports end normally. If a packet error is detected in mid-transport or if the CSW value is incorrect, the H_CHaIntStat register BO_SupportStop bit is set and the transaction stops. In this case, the cause is set to the H_CHaConditionCode register to enable inspection. If the ConditionCode value is "000" when the H_CHaIntStat register BO_SupportStop bit is set to "1," the CSW value is incorrect.

This bit is automatically cleared once the transport sequence ends (either ends normally or in error).

The transport can be stopped by clearing this bit while running the bulk-only support function. The BO_SupportCmp bit is set if the CSW transport ends normally here. The BO_SupportStop bit is set in all other cases. Refer to BO_TransportState for the stopped transport.

3.3.49 Port0:19Bh / Port1:39Bh H_CHaBO_CSW_RcvSize (Host CHa Bulk Only Transfer Support CSW Receive Data Size)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	19Bh	H_CHaBO		7:	0:	1:	00h
		_CSW_RcvDataSize		6:	0:	1:	
				5:	0:	1:	
				4:	0:	1:	
			R	3: CSW_RcvDataSize[3]	CSW Resceive Data Size		
				2: CSW_RcvDataSize[2]			
				1: CSW_RcvDataSize[1]			
				0: CSW_RcvDataSize[0]			

This indicates the data size received when running CSW transport using the channel CHa bulk-only support function for host operations.

Bit7-4 Reserved

Bit3-0 CSW_RcvDataSize[3:0]

This indicates the CSW received data size.

The amount of data received can be checked using this register when fewer than 13 bytes of data have been received in CSW transport.

This register value has no meaning if a handshake was received in CSW transport or other than for CSW transport.

3.3.50 Port0:19Ch / Port1:39Ch H_CHaBO_OUT_EP_Ctl (Host CHa Bulk Only Transfer Support OUT Endpoint Control)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	19Ch	H_CHaBO		7:	0:	1:	00h
		_OUT_EP_Ctl		6:	0:	1:	
				5:	0:	1:	
			R/W	4: OUT_Toggle	0: Toggle0	1: Toggle1	
			R/W	3: OUT_EP_Number[3]	OUT EP Number		
				2: OUT_EP_Number[2]			
				1: OUT_EP_Number[1]			
				0: OUT_EP_Number[0]			

This sets the channel CHa bulk-only support function for host operations.

Bit7-5 Reserved

Bit4 OUT_Toggle

The H_CBW_Control register BO_SupportGo bit is set to "1" and the initial value for the toggle sequence bit for OUT-direction transfer (CBW transport or Data OUT transport) is set using the bulk-only support function.

0: Toggle 0

1: Toggle 1

The toggle sequence bit is automatically retained at this bit if the OUT-direction transport ends normally.

Bit3-0 OUT_EP_Number[3:0]

The H_CBW_Control register BO_SupportGo bit is set to "1" and the endpoint number of the transfer destination device for OUT-direction transfer (CBW transport or Data OUT transport) is set using the bulk-only support function.

It can be set to any value from 0 to 15.

3.3.51 Port0:19Dh / Port1:39Dh H_CHaBO_IN_EP_Ctl (Host CHa Bulk Only Transfer Support IN Endpoint Control)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	19Dh	H_CHaBO_IN_EP_Ctl		7:	0:	1:	00h
				6:	0:	1:	
				5:	0:	1:	
			R/W	4: IN_Toggle	0: Toggle0	1: Toggle1	
			R/W	3: IN_EP_Number[3]	IN EP Number		
				2: IN_EP_Number[2]			
				1: IN_EP_Number[1]			
				0: IN_EP_Number[0]			

This sets the channel CHa bulk-only support function for host operations.

Bit7-5 Reserved

Bit4 IN_Toggle

The H_CBW_Control register BO_SupportGo bit is set to "1" and the initial value for the toggle sequence bit for IN-direction transfer (CSW transport or Data IN transport) is set using the bulk-only support function.

0: Toggle 0

1: Toggle 1

The toggle sequence bit is automatically retained at this bit if the IN-direction transport ends normally.

Bit3-0 IN_EP_Number[3:0]

The H_CBW_Control register BO_SupportGo bit is set to "1" and the endpoint number of the transfer destination device for IN-direction transfer (CSW transport or Data IN transport) is set using the bulk-only support function.

It can be set to any value from 0 to 15.

3.3.52 Port0:19Eh / Port1:39Eh H_CHaConditionCode (Host Channel a Condition Code)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	19Eh	H_CHaConditionCode		7:	0:	1:	00h
			R	6: ConditionCode[2]	Condition Code		
				5: ConditionCode[1]			
				4: ConditionCode[0]			
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
				0:	0:	1:	

This indicates channel CHa transfer completion results for host operations.

Bit7-4 ConditionCode[2:0]

Indicates results when the transfer is completed over channel CHa.

Code	Meaning	Description
000	NOERROR	Transaction completed without errors.
001	STALL	Endpoint returned Stall PID.

010	DATAOVERRUN	• The data packet received exceeds the maximum packet size.
		  * Treated as a retry error if CRC error and bit stuffing error are detected simultaneously.
		• The data size received exceeds the IRP (TotalSize).
		  * Treated as a retry error if CRC error and bit stuffing error are detected simultaneously.
		  * Treated as a toggle mismatch rather than a data overrun if the data packet is less than the maximum packet size and the data toggle included in the data packet fails to match the expected value.

011	DATAUNDERRUN	• The data packet received is less than the maximum packet size; the data size is less than the IRP (TotalSize). * Treated as a retry error if CRC error and bit stuffing error are detected simultaneously.
100	RETRYERROR	• The device does not respond to a token (IN) or does not issue a handshake (OUT) within the specified timeframe. • The data packet from the endpoint includes a CRC error. • The data packet from the endpoint includes a bit stuffing error. • The PID inspection bit from the endpoint failed in data PID (IN) or handshake (OUT). • The PID received is invalid or no PID value is defined. • The data toggle included in the data packet from the endpoint fails to match the expected value (toggle mismatch).
Other	Reserved	

Bit3-0 Reserved

3.3.53 Port0:1A0h / Port1:3A0h H_CHbConfig_0(Host Channel b Configuration0)
3.3.54 Port0:1B0h / Port1:3B0h H_CHcConfig_0(Host Channel c Configuration0)
3.3.55 Port0:1C0h / Port1:3C0h H_CHdConfig_0(Host Channel d Configuration0)
3.3.56 Port0:1D0h / Port1:3D0h H_CHeConfig_0(Host Channel e Configuration0)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	1A0h	CHbConfig_0	R/W	7: ACK_Cnt[3]	ACK Count		00h
	1B0h	CHcConfig_0		6: ACK_Cnt[2]			
	1C0h	CHdConfig_0		5: ACK_Cnt[1]			
	1D0h	CHeConfig_0		4: ACK_Cnt[0]			
			R/W	3: SpeedMode[1]	Speed Mode		
				2: SpeedMode[0]			
			R/W	1: Toggle	0: Toggle0	1: Toggle1	
			R/W	0: TranGo	0: Stand by	1: Transaction Start	

These set the basic channel CHx{x=b-e} settings for host operations.

Bit7-4 ACK_Cnt [3:0]

This sets the ACK count number for transfers performed with channel CHx{x=b-e}.

The H_CHx{x=b-e}IntStat register TranACK bit is set once the ACK number set has been counted.

0000: 16 ACK counts

0001 to 1111: 1 to 15 ACK counts

Bit3-2 SpeedMode [1:0]

This sets the operating mode for the device performing the transfer over channel CHx{x=b-e}.

00: HS mode – Use this setting for an HS device.

01: FS mode – Use this setting for an FS device.

10: Reserved – Use of this value is prohibited.

11: LS mode – Use this setting for an LS device.

Bit1 Toggle

This sets the initial value of the toggle sequence bit at the start of a transaction. It also indicates the toggle sequence bit state after the transaction has been started or completed.

0: Toggle 0

1: Toggle 1

Bit0 TranGo

Setting this bit to “1” starts the channel CHx{x=b-e} transaction. After being started, the transaction process can be stopped by clearing the bit to “0.” This bit also indicates whether a transaction is running on channel CHx{x=b-e}.

0: Stops the transaction (transaction stopped)

1: Starts the transaction (transaction running)

The H_CHx{x=b-e}IntStat register TotalSizeCmp bit is set to “1” as soon as the number of bytes set in registers H_CHx{x=b-e}TotalSize_HH to LL have been transferred. This bit automatically reverts to “0.” For bulk transfers and interrupt transfers, it is reset to “0” if the H_CHx{x=b-e}IntStat register ChangeCondition bit has been set. In this case, the cause is set to the H_CHx{x=b-e}ConditionCode register to enable inspection.

The H_CHx{x=b-e}IntStat register ChangeCondition bit is set as soon as a transaction in progress ends when stopped by clearing the bit. The data in the FIFO, (remaining) total size, and channel-related settings remain unchanged even when the transaction is stopped. This allows resumption of the transaction from the point at which it was stopped by resetting this bit to “1.” (To perform a new transaction, clear the FIFO and reset the channel information.)

3.3.57 Port0:1A1h / Port1:3A1h H_CHbConfig_1(Host Channel b Configuration1)
3.3.58 Port0:1B1h / Port1:3B1h H_CHcConfig_1(Host Channel c Configuration1)
3.3.59 Port0:1C1h / Port1:3C1h H_CHdConfig_1(Host Channel d Configuration1)
3.3.60 Port0:1D1h / Port1:3D1h H_CHeConfig_1(Host Channel e Configuration1)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	1A1h	H_CHbConfig_1	R/W	7: TID[1]	Transaction ID		00h
	1B1h	H_CHcConfig_1		6: TID[0]			
	1C1h	H_CHdConfig_1	R/W	5: TranType[1]	Transfer Type		
	1D1h	H_CHeConfig_1		4: TranType[0]			
			R/W	3: AutoZerolen	0: Do nothing	1: Add Zerolen	
			R/W	2: Audio441	0: Disable	1: Enable	
			R/W	1: TotalSizeFree[1]	TotalSizeFree[1:0]		
				0: TotalSizeFree[0]			

These set the basic channel CHx{x=b-e} settings for host operations.

Bit7-6 TID[1:0]

This sets the transaction type (OUT/IN) to be issued with channel CHx{x=b-e}.

00: Reserved – Use of this value is prohibited.

01: OUT – Issues an OUT token.

10: IN – Issues an IN token.

11: Reserved – Use of this value is prohibited.

Bit5-4 TranType[1:0]

Sets the transfer type performed with channel CHx{x=b-e}.

00: Reserved – Use of this value is prohibited.

01: Isochronous – Uses isochronous transfers.

10: Bulk – Uses bulk transfers.

11: Interrupt – Uses interrupt transfers.

Bit3 AutoZerolen

Setting this bit to "1" automatically adds a zero-length packet after the transfer size set in the H_CHx{x=b-e}TotalSizeHH to LL registers ends at exactly the MaxPacketSize. This bit is enabled only for OUT transfers.

Bit2 Audio441

Setting this bit to "1" enables the audio class assist function. The audio class assist function is used when transmitting 16-bit 2-channel PCM data with a sampling frequency of 44.1 kHz via isochronous transfer at a 1-ms cycle. The size of the data packets transmitted is automatically adjusted when the audio class assist function is enabled. Nine consecutive transactions with a 176-byte data packet size are followed by one transaction with a 180-byte data packet size.

Bit1-0 TotalSizeFree[1:0]

These set the transfer control using the H_CHx{x=b-e}TotalSize_HH to LL registers.

00: Clears the TranGo bit and ends the transfer once the size set in the H_CHx{x=b-e}TotalSize_HH to LL registers has been transferred. A TotalSizeCmp interrupt then occurs here. For OUT transfers, the final data packet size will be the smaller of MaxPktSize and TotalSize. For IN transfers, the expected final data packet size will be the smaller of MaxPktSize and TotalSize.

01: Performs the transfer regardless of the H_CHx{x=b-e}TotalSize_HH to LL register values. For OUT transfers, the data packet size will be MaxPktSize. For IN transfers, the expected data packet size will be MaxPktSize. The TranGo bit is not cleared and no TotalSizeCmp interrupt occurs for this setting, even when the size set in the H_CHx{x=b-e}TotalSize_HH to LL registers has been transferred.

10: A TotalSizeCmp interrupt occurs when the size set in the H_CHx{x=b-e}TotalSize_HH to LL registers has been transferred. The TranGo bit is not cleared here, but channel transaction issuing is stopped while the TotalSize value is "0." For OUT transfers, the data packet size will be the smaller of MaxPktSize and TotalSize. For IN transfers, the expected data packet size will be the smaller of MaxPktSize and TotalSize.

11: Reserved

3.3.61 Port0:1A2h-1A3h / Port1:3A2h-3A3h H_CHbMaxPktSize_H,L (Host Channel b Max Packet Size High, Low)

3.3.62 Port0:1B2h-1B3h / Port1:3B2h-3B3h H_CHcMaxPktSize_H,L (Host Channel c Max Packet Size High, Low)

3.3.63 Port0:1C2h-1C3h / Port1:3C2h-3C3h H_CHdMaxPktSize_H,L (Host Channel d Max Packet Size High, Low)

3.3.64 Port0:1D2h-1D3h / Port1:3D2h-3D3h H_CHeMaxPktSize_H,L (Host Channel e Max Packet Size High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	1A2h-1A3h	H_CHbMaxPktSize_H,L		15:	0:	1:	0000h
	1B2h-1B3h	H_CHcMaxPktSize_H,L		14:	0:	1:	
	1C2h-1C3h	H_CHdMaxPktSize_H,L		13:	0:	1:	
	1D2h-1D3h	H_CHeMaxPktSize_H,L		12:	0:	1:	
				11:	0:	1:	
				10:	0:	1:	
			R/W	9: MaxPktSize[9]	Max Packet Size		
				8: MaxPktSize[8]			
				7: MaxPktSize[7]			
				6: MaxPktSize[6]			
				5: MaxPktSize[5]			
				4: MaxPktSize[4]			
				3: MaxPktSize[3]			
				2: MaxPktSize[2]			
				1: MaxPktSize[1]			
				0: MaxPktSize[0]			

These set the channel CHx{x=b-e} MaxPacketSize for host operations.

Bit15-10 Reserved

Do not write "1" to reserved bits.

Bit9-0 MaxPktSize[9:0]

These set the channel CHx{x=b-e} MaxPacketSize.

Set to one of the following when using this channel for bulk transfers.

FS: 8, 16, 32, 64 bytes

HS: 512 bytes

The transfer size can be set as follows when using this channel for interrupt transfers.

LS: Up to 8 bytes

FS: Up to 64 bytes

HS: Up to 1,024 bytes

The transfer size can be set as follows when using this channel for isochronous transfer.

FS: Up to 1,023 bytes

HS: Up to 1,024 bytes

All other settings are prohibited.

3.3.65 Port0:1A4h-1A5h / Port1:3A4h-3A5h H_CHbTotalSize_HH,HL (Host Channel b Total Size High-High, High-Low)

3.3.66 Port0:1A6h-1A7h / Port1:3A6h-3A7h H_CHbTotalSize_LH,LL (Host Channel b Total Size Low-High, Low-Low)

3.3.67 Port0:1B4h-1B5h / Port1:3B4h-3B5h H_CHcTotalSize_HH,HL (Host Channel c Total Size High-High, High-Low)

3.3.68 Port0:1B6h-1B7h / Port1:3B6h-3B7h H_CHcTotalSize_LH,LL (Host Channel c Total Size Low-High, Low-Low)

3.3.69 Port0:1C4h-1C5h / Port1:3C4h-3C5h H_CHdTotalSize_HH,HL (Host Channel d Total Size High-High, High-Low)

3.3.70 Port0:1C6h-1C7h / Port1:3C6h-3C7h H_CHdTotalSize_LH,LL (Host Channel d Total Size Low-High, Low-Low)

3.3.71 Port0:1D4h-1D5h / Port1:3D4h-3D5h H_CHeTotalSize_HH,HL (Host Channel e Total Size High-High, High-Low)

3.3.72 Port0:1D6h-1D7h / Port1:3D6h-3D7h H_CHeTotalSize_LH,LL (Host Channel e Total Size Low-High, Low-Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Host	1A4h-1A5h	H_CHbTotalSize_HH,HL	R/W	15: TotalSize[31]	Total Size High	0000h
	1B4h-1B5h	H_CHcTotalSize_HH,HL		14: TotalSize[30]		
	1C4h-1C5h	H_CHdTotalSize_HH,HL		13: TotalSize[29]		
	1D4h-1D5h	H_CHeTotalSize_HH,HL		12: TotalSize[28]		
				11: TotalSize[27]		
				10: TotalSize[26]		
				9: TotalSize[25]		
				8: TotalSize[24]		
				7: TotalSize[23]		
				6: TotalSize[22]		
				5: TotalSize[21]		
				4: TotalSize[20]		
				3: TotalSize[19]		
				2: TotalSize[18]		
				1: TotalSize[17]		
				0: TotalSize[16]		

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Host	1A6h-1A7h	H_CHbTotalSize_LH,LL	R/W	15: TotalSize[15]	Total Size Low	0000h
	1B6h-1B7h	H_CHcTotalSize_LH,LL		14: TotalSize[14]		
	1C6h-1C7h	H_CHdTotalSize_LH,LL		13: TotalSize[13]		
	1D6h-1D7h	H_CHeTotalSize_LH,LL		12: TotalSize[12]		
				11: TotalSize[11]		
				10: TotalSize[10]		
				9: TotalSize[9]		
				8: TotalSize[8]		
				7: TotalSize[7]		
				6: TotalSize[6]		
				5: TotalSize[5]		
				4: TotalSize[4]		
				3: TotalSize[3]		
				2: TotalSize[2]		
				1: TotalSize[1]		
				0: TotalSize[0]		

These set the total number of bytes of the data to be transferred over channel CHx{x=b-e} for host operations (max. 4,294,967,295 bytes = approx. 4 Gbytes).

The remaining transfer quantity can be read by reading these registers after the transaction has been started by the H_CHx{x=b-e}Config_0 register TranGo bit.

A zero-length packet is issued when an OUT transaction is performed with TotalSize = 0.

3.3.73 Port0:1A8h / Port1:3A8h H_CHbHubAdrs (Host Channel b Hub Address)
3.3.74 Port0:1B8h / Port1:3B8h H_CHcHubAdrs (Host Channel c Hub Address)
3.3.75 Port0:1C8h / Port1:3C8h H_CHdHubAdrs (Host Channel d Hub Address)
3.3.76 Port0:1D8h / Port1:3D8h H_CHeHubAdrs (Host Channel e Hub Address)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	1A8h	H_CHbHubAdrs	R/W	7: HubAdrs[3]	Hub Address		00h
	1B8h	H_CHcHubAdrs		6: HubAdrs[2]			
	1C8h	H_CHdHubAdrs		5: HubAdrs[1]			
	1D8h	H_CHeHubAdrs		4: HubAdrs[0]			
				3:	0:	1:	
			R/W	2: Port[2]	Port Number		
				1: Port[1]			
				0: Port[0]			

These set the hub connecting to channel CHx{x=b-e} for host operations.

Bit7 HubAdrs[3:0]

This sets the USB address of the hub to which the function performing the transfer over channel CHx{x=b-e} connects.

It can be set to any value from 0 to 15.

Bit3 Reserved

Bit2-0 Port[2:0]

This sets the port number of the hub to which the function performing the transfer over channel CHx{x=b-e} connects.

It can be set to any value from 0 to 7.

3.3.77 Port0:1A9h / Port1:3A9h H_CHbFuncAdrs (Host Channel b Function Address)
3.3.78 Port0:1B9h / Port1:3B9h H_CHcFuncAdrs (Host Channel c Function Address)
3.3.79 Port0:1C9h / Port1:3C9h H_CHdFuncAdrs (Host Channel d Function Address)
3.3.80 Port0:1D9h / Port1:3D9h H_CHeFuncAdrs (Host Channel d Function Address)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description	Reset
Host	1A9h	H_CHbFuncAdrs	R/W	7: FuncAdrs[3]	Function Address	00h
	1B9h	H_CHcFuncAdrs		6: FuncAdrs[2]		
	1C9h	H_CHdFuncAdrs		5: FuncAdrs[1]		
	1D9h	H_CHeFuncAdrs		4: FuncAdrs[0]		
			R/W	3: EP_Number[3]	Endpoint Number	
				2: EP_Number[2]		
				1: EP_Number[1]		
				0: EP_Number[0]		

These set the address of the function performing the transfer over channel CHx{x=b-e} for host operations.

Bit7-4 FuncAdrs[3:0]

This sets the USB address of the function, including the endpoint managed by channel CHx{x=b-e}.

It can be set to any value from 0 to 15.

Bit3-0 EP_Number[3:0]

This sets the endpoint number for the transfer over channel CHx{x=b-e}.

It can be set to any value from 0 to 15.

3.3.81 Port0:1AAh-1ABh / Port1:3AAh-3ABh H_CHbInterval_H,L (Host Channel b Interval High, Low)

3.3.82 Port0:1BAh-1BBh / Port1:3BAh-3BBh H_CHcInterval_H,L (Host Channel c Interval High, Low)

3.3.83 Port0:1CAh-1CBh / Port1:3CAh-3CBh H_CHdInterval_H,L (Host Channel d Interval High, Low)

3.3.84 Port0:1DAh-1DBh / Port1:3DAh-3DBh H_CHeInterval_H,L (Host Channel e Interval High, Low)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	1AAh-1ABh	H_CHbInterval_H,L		15:	0:	1:	0000h
	1BAh-1BBh	H_CHcInterval_H,L		14:	0:	1:	
	1CAh-1CBh	H_CHdInterval_H,L		13:	0:	1:	
	1DAh-1DBh	H_CHeInterval_H,L		12:	0:	1:	
				11:	0:	1:	
			R/W	10: Interval[10]	Transfer Interval		
				9: Interval[9]			
				8: Interval[8]			
				7: Interval[7]			
				6: Interval[6]			
				5: Interval[5]			
				4: Interval[4]			
				3: Interval[3]			
				2: Interval[2]			
				1: Interval[1]			
				0: Interval[0]			

These set the interval for interrupt transfers and isochronous transfers over channel CHx{x=b-e} for host operations.

Bit15-11 Reserved

Bit10-0 Interval[10:0]

These registers specify the interrupt transfer and isochronous transfer token issuing interval (period). The lower 3 bits specify the interval in 125-µs microframes, while the higher 7 bits specify the interval in 1-ms frames. This register setting is enabled only when the H_CHx{x=b-e}Config1 register TranType bit is "11" (interrupt transfer) or "01" (isochronous transfer). The "0d0" setting is disabled for these registers.

The interval set in this register is also used when resending transactions.

Interval[2:0] µFrame – Specifies the interval in 125 µs units. Set to 1, 2, or 4 microframes. Settings to any other values are prohibited. Interval[10:3] must be set entirely to "0" when setting this bit.

Interval[10:3] Frame – Specifies the interval in ms units. It can be set to any value from 1 to 255 frames. Interval[2:0] must be set entirely to "0" when setting this bit.

3.3.85 14Ch H_CHbTranPause(Host Channel b Transaction Pause)

3.3.86 15Ch H_CHcTranPause(Host Channel c Transaction Pause)

3.3.87 16Ch H_CHdTranPause(Host Channel d Transaction Pause)

3.3.88 17Ch H_CHeTranPause(Host Channel e Transaction Pause)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	14Ch	H_CHbTranPause		7:	0:	1:	00h
	15Ch	H_CHcTranPause		6:	0:	1:	
	16Ch	H_CHdTranPause		5:	0:	1:	
	17Ch	H_CHeTranPause		4:	0:	1:	
				3:	0:	1:	
				2:	0:	1:	
			R/W	1: EnTranPause	0: Disable	1: Enable	
			R/W	0: TranPause	0: Do nothing	1: Transaction Pause	

Bit7-2 Reserved

Bit1 EnTranPause

Setting this bit to “1” sets the H_CHx{x=b-e}TranPause register TranPause bit to “1” at the same time that the H_CHx{x=b-e}IntStat register TranACK bit is set.

Bit0 TranPause

When this bit is set to “1,” no transactions are performed using this channel, even if the H_CHx{x=b-e}Config_0 TranGo bit is set to “1.” If this channel is set for interrupt or isochronous transfers, the transfer cycle is maintained, even if no transactions can be performed because the bit is set to “1.”

3.3.89 Port0:1AEh / Port1:3AEh H_CHbConditionCode (Host Channel b Condition Code)
3.3.90 Port0:1BEh / Port1:3BEh H_CHcConditionCode (Host Channel c Condition Code)
3.3.91 Port0:1CEh / Port1:3CEh H_CHdConditionCode (Host Channel d Condition Code)
3.3.92 Port0:1DEh / Port1:3DEh H_CHeConditionCode (Host Channel e Condition Code)

Base Address: Port0=000h, Port1=200h

Mode	Ofst Adrs	Register Name	R/W	Bit Symbol	Description		Reset
Host	1AEh	H_CHbConditionCode		7:	0:	1:	00h
	1BEh	H_CHcConditionCode	R	6: ConditionCode[2]	Condition Code		
	1CEh	H_CHdConditionCode		5: ConditionCode[1]			
	1DEh	H_CHeConditionCode		4: ConditionCode[0]			
				3:	0:	1:	
				2:	0:	1:	
				1:	0:	1:	
				0:	0:	1:	

These indicate the channel CHx{x=b-e} transfer completion results for host operations.

Bit7-4 ConditionCode[2:0]

Indicates results when the transfer is completed over channel CHx{x=b-e}.

Code	Meaning	Description
000	NOERROR	Transaction completed without errors.
001	STALL	Endpoint returned Stall PID.
010	DATAOVERRUN	• The data packet received exceeds the maximum packet size. * Treated as a retry error if CRC error and bit stuffing error are detected simultaneously. • The data size received exceeds the IRP (TotalSize). * Treated as a retry error if CRC error and bit stuffing error are detected simultaneously. * Treated as a toggle mismatch rather than a data overrun if the data packet is less than the maximum packet size and the data toggle included in the data packet fails to match the expected value.
011	DATAUNDERRUN	• The data packet received is less than the maximum packet size; the data size is less than the IRP (TotalSize). * Treated as a retry error if CRC error and bit stuffing error are detected simultaneously.

Code	Meaning	Description
100	RETRYERROR	• The device does not respond to a token (IN) or does not issue a handshake (OUT) within the specified timeframe. • The data packet from the endpoint includes a CRC error. • The data packet from the endpoint includes a bit stuffing error. • The PID inspection bit from the endpoint failed in data PID (IN) or handshake (OUT). • The PID received is invalid or no PID value is defined. • ERR handshake received in interrupt transfer split transaction. • NYET handshake received 3 times consecutively in interrupt transfer split transaction. • The data toggle included in the data packet from the endpoint fails to match the expected value (toggle mismatch).
110	BUFFEROVERRUN	• No transaction was performed because the FIFO free space was smaller than the maximum packet size in an isochronous transfer.
111	BUFFERUNDERRUN	• No transaction was performed due to insufficient FIFO valid data in an isochronous transfer.
Other	Reserved	

Bit3-0 Reserved

Appendix A: Connection to Little-endian CPU

The LSI internal bus has a big-endian configuration, with the even addresses as the higher bytes and odd addresses as the lower bytes. In contrast, this section describes the methods for connecting to a little-endian CPU.

<Circuit board>

Connect the little-endian CPU and LSI terminals as per terminal names for data bus and write control signals. In other words, connect the LSI CD15 to CD8 to the CPU data bus bits 15 to 8 (higher bytes) and connect the LSI CD7 to CD0 to the CPU data bus bits 7 to 0 (lower bits). Connect the write control signal terminals to high and low as usual.

Note that the write signal specifications themselves will differ with the actual CPU used.

<Firmware>

Make the following settings before initializing to operate the LSI with a little-endian CPU.

(1) Set the CPUIF_MODE.CPU_Endian bit to "1"

This register is word-assigned to address 0x07E and can be written to only when uninitialized. Set using the first access after hard resetting. Setting the CPU_Endian bit 10 or bit2 or both to "1" sets to little endian.

(2) Dummy read or dummy write the CPUIF_MODE register

This dummy access ends the initialization period and defines the register as the address indicated in the register map below.

Using these settings enables all internal registers to be accessed by Char or Short. Access using the CPU DMAC is also possible. (Refer to table below.)

Example: Accessing FIFO_Rd_0/1 register when receiving data from USB in sequence 01_02_03_04_05_06

Access by Short	CPU access method			
	Big-endian		Little-endian	
	CD[15:8]	CD[7:0]	CD[15:8]	CD[7:0]
1st	01	02	02	01
2nd	03	04	04	03
3rd	05	06	06	05

For registers larger than Short, use divided access with Short and cast in the CPU memory for use.

The registers for which odd and even addresses are swapped are shown below for CPU_Endian="1" (little endian). Reading and writing is possible to/from CD[15:0] for these registers with the 16-bit registers arranged unchanged.

Big Endian		Little Endian	
Device/host common registers		**Device/host common registers**	
0x014	***WakeupTim_H***	0x014	***WakeupTim_L***
0x015	***WakeupTim_L***	0x015	***WakeupTim_H***
0x024	FIFO_RdRemain_H	0x024	FIFO_RdRemain_L
0x025	FIFO_RdRemain_L	0x025	FIFO_RdRemain_H
0x026	FIFO_WrRemain_H	0x026	FIFO_WrRemain_L
0x027	FIFO_WrRemain_L	0x027	FIFO_WrRemain_H
0x030	RAM_RdAdrs_H	0x030	RAM_RdAdrs_L
0x031	RAM_RdAdrs_L	0x031	RAM_RdAdrs_H
0x038	RAM_WrAdrs_H	0x038	RAM_WrAdrs_L
0x039	RAM_WrAdrs_L	0x039	RAM_WrAdrs_H
0x064	DMA_Remain_H	0x064	DMA_Remain_L
0x065	DMA_Remain_L	0x065	DMA_Remain_H
0x068	DMA_Count_HH	0x068	DMA_Count_HL
0x069	DMA_Count_HL	0x069	DMA_Count_HH
0x06A	DMA_Count_LH	0x06A	DMA_Count_LL
0x06B	DMA_Count_LL	0x06B	DMA_Count_LH
0x080	AREA0StartAdrs_H	0x080	AREA0StartAdrs_L
0x081	AREA0StartAdrs_L	0x081	AREA0StartAdrs_H
0x082	AREA0EndAdrs_H	0x082	AREA0EndAdrs_L
0x083	AREA0EndAdrs_L	0x083	AREA0EndAdrs_H
0x084	AREA1StartAdrs_H	0x084	AREA1StartAdrs_L
0x085	AREA1StartAdrs_L	0x085	AREA1StartAdrs_H
0x086	AREA1EndAdrs_H	0x086	AREA1EndAdrs_L
0x087	AREA1EndAdrs_L	0x087	AREA1EndAdrs_H
0x088	AREA2StartAdrs_H	0x088	AREA2StartAdrs_L
0x089	AREA2StartAdrs_L	0x089	AREA2StartAdrs_H

0x08A	AREA2EndAdrs_H
0x08B	AREA2EndAdrs_L

0x08A	AREA2EndAdrs_L
0x08B	AREA2EndAdrs_H

0x08C	AREA3StartAdrs_H
0x08D	AREA3StartAdrs_L

0x08C	AREA3StartAdrs_L
0x08D	AREA3StartAdrs_H

0x08E	AREA3EndAdrs_H
0x08F	AREA3EndAdrs_L

0x08E	AREA3EndAdrs_L
0x08F	AREA3EndAdrs_H

0x090	AREA4StartAdrs_H
0x091	AREA4StartAdrs_L

0x090	AREA4StartAdrs_L
0x091	AREA4StartAdrs_H

0x092	AREA4EndAdrs_H
0x093	AREA4EndAdrs_L

0x092	AREA4EndAdrs_L
0x093	AREA4EndAdrs_H

0x094	AREA5StartAdrs_H
0x095	AREA5StartAdrs_L

0x094	AREA5StartAdrs_L
0x095	AREA5StartAdrs_H

0x096	AREA5EndAdrs_H
0x097	AREA5EndAdrs_L

0x096	AREA5EndAdrs_L
0x097	AREA5EndAdrs_H

Device register (HOSTxDEVICE=0)

0x0BC	D_AlarmIN_IntStat_H
0x0BD	D_AlarmIN_IntStat_L

Device register (HOSTxDEVICE=0)

0x0BC	D_AlarmIN_IntStat_L
0x0BD	D_AlarmIN_IntStat_H

0x0BE	D_AlarmOUT_IntStat_H
0x0BF	D_AlarmOUT_IntStat_L

0x0BE	D_AlarmOUT_IntStat_L
0x0BF	D_AlarmOUT_IntStat_H

0x0CC	D_AlarmIN_IntEnb_H
0x0CD	D_AlarmIN_IntEnb_L

0x0CC	D_AlarmIN_IntEnb_L
0x0CD	D_AlarmIN_IntEnb_H

0x0CE	D_AlarmOUT_IntEnb_H
0x0CF	D_AlarmOUT_IntEnb_L

0x0CE	D_AlarmOUT_IntEnb_L
0x0CF	D_AlarmOUT_IntEnb_H

0x0EE	D_FrameNumber_H
0x0EF	D_FrameNumber_L

0x0EE	D_FrameNumber_L
0x0EF	D_FrameNumber_H

0x0F8	D_EPaMaxSize_H
0x0F9	D_EPaMaxSize_L

0x0F8	D_EPaMaxSize_L
0x0F9	D_EPaMaxSize_H

0x100	D_EPbMaxSize_H
0x101	D_EPbMaxSize_L

0x100	D_EPbMaxSize_L
0x101	D_EPbMaxSize_H

0x108	D_EPcMaxSize_H
0x109	D_EPcMaxSize_L

0x108	D_EPcMaxSize_L
0x109	D_EPcMaxSize_H

0x110	D_EPdMaxSize_H
0x111	D_EPdMaxSize_L

0x110	D_EPdMaxSize_L
0x111	D_EPdMaxSize_H

0x118	D_EPeMaxSize_H
0x119	D_EPeMaxSize_L

0x118	D_EPeMaxSize_L
0x119	D_EPeMaxSize_H

0x120	D_DescAdrs_H
0x121	D_DescAdrs_L

0x120	D_DescAdrs_L
0x121	D_DescAdrs_H

0x122	D_DescSize_H
0x123	D_DescSize_L

0x122	D_DescSize_L
0x123	D_DescSize_H

0x128	D_EnEP_IN_H
0x129	D_EnEP_IN_L

0x128	D_EnEP_IN_L
0x129	D_EnEP_IN_H

0x12A	D_EnEP_OUT_H
0x12B	D_EnEP_OUT_L

0x12A	D_EnEP_OUT_L
0x12B	D_EnEP_OUT_H

0x12C	D_EnEP_IN_ISO_H
0x12D	D_EnEP_IN_ISO_L

0x12C	D_EnEP_IN_ISO_L
0x12D	D_EnEP_IN_ISO_H

0x12E	D_EnEP_OUT_ISO_H
0x12F	D_EnEP_OUT_ISO_L

0x12E	D_EnEP_OUT_ISO_L
0x12F	D_EnEP_OUT_ISO_H

Host register (HOSTxDEVICE=1)

0x17E	H_FrameNumber_H
0x17F	H_FrameNumber_L

Host register (HOSTxDEVICE=1)

0x17E	H_FrameNumber_L
0x17F	H_FrameNumber_H

0x182	
0x183	H_CH0MaxPktSize

0x182	H_CH0MaxPktSize
0x183	

0x186	H_CH0TotalSize_H
0x187	H_CH0TotalSize_L

0x186	H_CH0TotalSize_L
0x187	H_CH0TotalSize_H

0x192	H_CHaMaxPktSize_H
0x193	H_CHaMaxPktSize_L

0x192	H_CHaMaxPktSize_L
0x193	H_CHaMaxPktSize_H

0x194	H_CHaTotalSize_HH
0x195	H_CHaTotalSize_HL

0x194	H_CHaTotalSize_HL
0x195	H_CHaTotalSize_HH

0x196	H_CHaTotalSize_LH
0x197	H_CHaTotalSize_LL

0x196	H_CHaTotalSize_LL
0x197	H_CHaTotalSize_LH

0x1A2	H_CHbMaxPktSize_H
0x1A3	H_CHbMaxPktSize_L

0x1A2	H_CHbMaxPktSize_L
0x1A3	H_CHbMaxPktSize_H

0x1A4	H_CHbTotalSize_HH
0x1A5	H_CHbTotalSize_HL

0x1A4	H_CHbTotalSize_HL
0x1A5	H_CHbTotalSize_HH

0x1A6	H_CHbTotalSize_LH
0x1A7	H_CHbTotalSize_LL

0x1A6	H_CHbTotalSize_LL
0x1A7	H_CHbTotalSize_LH

0x1AA	H_CHbInterval_H
0x1AB	H_CHbInterval_L

0x1AA	H_CHbInterval_L
0x1AB	H_CHbInterval_H

0x1B2	H_CHcMaxPktSize_H
0x1B3	H_CHcMaxPktSize_L

0x1B2	H_CHcMaxPktSize_L
0x1B3	H_CHcMaxPktSize_H

0x1B4	H_CHcTotalSize_HH
0x1B5	H_CHcTotalSize_HL

0x1B4	H_CHcTotalSize_HL
0x1B5	H_CHcTotalSize_HH

0x1B6	H_CHcTotalSize_LH
0x1B7	H_CHcTotalSize_LL

0x1B6	H_CHcTotalSize_LL
0x1B7	H_CHcTotalSize_LH

0x1BA	H_CHcInterval_H
0x1BB	H_CHcInterval_L

0x1BA	H_CHcInterval_L
0x1BB	H_CHcInterval_H

0x1C2	H_CHdMaxPktSize_H
0x1C3	H_CHdMaxPktSize_L

0x1C2	H_CHdMaxPktSize_L
0x1C3	H_CHdMaxPktSize_H

0x1C4	H_CHdTotalSize_HH
0x1C5	H_CHdTotalSize_HL

0x1C4	H_CHdTotalSize_HL
0x1C5	H_CHdTotalSize_HH

0x1C6	H_CHdTotalSize_LH

0x1C6	H_CHdTotalSize_LL

0x1C7	H_CHdTotalSize_LL

0x1C7	H_CHdTotalSize_LH

0x1CA	H_CHdInterval_H
0x1CB	H_CHdInterval_L

0x1CA	H_CHdInterval_L
0x1CB	H_CHdInterval_H

0x1D2	H_CHeMaxPktSize_H
0x1D3	H_CHeMaxPktSize_L

0x1D2	H_CHeMaxPktSize_L
0x1D3	H_CHeMaxPktSize_H

0x1D4	H_CHeTotalSize_HH
0x1D5	H_CHeTotalSize_HL

0x1D4	H_CHeTotalSize_HL
0x1D5	H_CHeTotalSize_HH

0x1D6	H_CHeTotalSize_LH
0x1D7	H_CHeTotalSize_LL

0x1D6	H_CHeTotalSize_LL
0x1D7	H_CHeTotalSize_LH

0x1DA	H_CHeInterval_H
0x1DB	H_CHeInterval_L

0x1DA	H_CHeInterval_L
0x1DB	H_CHeInterval_H

Revision History (Rev. 1.00)

Date	Revision details			
	Rev.	Section (old version)	Type	Details
10/12/2007	0.79	All pages	New	Newly established
7/15/2008	1.00	96	Revision	“Disconnection detection is performed for the uSOF(HS_SOF) EOP time period, and the device is determined to be disconnected if identified as disconnected three times in succession” →” Disconnection detection is performed for the uSOF(HS_SOF) EOP time period, and the device is determined to be disconnected if identified as disconnected consecutive three times in succession”
		223	Revision	“XCS” → “XCS and CA9”
		333, 347, 363	Revision	“H_BO_SupportControl” → “H_CHaBO_SupportCtl”
		335, 388	Revision	Addition of “BufferOverrun” and “BufferUnderrun” in a table
		343	Revision	Addition of “When overwritting AutoMode” in AutoModeCancel
		369	Revision	“Supporoto” → “Support”
				Blank below this line

EPSON

International Sales Operations

AMERICA

EPSON ELECTRONICS AMERICA, INC.
HEADQUARTERS
2580 Orchard Parkway
San Jose , CA 95131,USA
Phone: +1-800-228-3964 FAX: +1-408-922-0238

SALES OFFICES
Northeast
301 Edgewater Place, Suite 210
Wakefield, MA 01880, U.S.A.
Phone: +1-800-922-7667 FAX: +1-781-246-5443

EUROPE

EPSON EUROPE ELECTRONICS GmbH
HEADQUARTERS
Riesstrasse 15 Muenchen Bayern, 80992 GERMANY
Phone: +49-89-14005-0 FAX: +49-89-14005-110

ASIA

EPSON (CHINA) CO., LTD.
7F, Jinbao Bldg.,No.89 Jinbao St.,Dongcheng District,
Beijing 100005, China
Phone: +86-10-6410-6655 FAX: +86-10-6410-7320

SHANGHAI BRANCH
7F, Block B, Hi-Tech Bldg., 900, Yishan Road,
Shanghai 200233, CHINA
Phone: +86-21-5423-5522 FAX: +86-21-5423-5512

EPSON HONG KONG LTD.
20/F., Harbour Centre, 25 Harbour Road
Wanchai, Hong Kong
Phone: +852-2585-4600 FAX: +852-2827-4346
Telex: 65542 EPSCO HX

EPSON (CHINA) CO., LTD.
SHENZHEN BRANCH
12/F, Dawning Mansion, Keji South 12th Road,
Hi- Tech Park, Shenzhen
Phone: +86-755-2699-3828 FAX: +86-755-2699-3838

EPSON TAIWAN TECHNOLOGY & TRADING LTD.
14F, No. 7, Song Ren Road,
Taipei 110
Phone: +886-2-8786-6688 FAX: +886-2-8786-6660

EPSON SINGAPORE PTE., LTD.
1 HarbourFront Place,
#03-02 HarbourFront Tower One, Singapore 098633
Phone: +65-6586-5500 FAX: +65-6271-3182

SEIKO EPSON CORPORATION
KOREA OFFICE
50F, KLI 63 Bldg., 60 Yoido-dong
Youngdeungpo-Ku, Seoul, 150-763, KOREA
Phone: +82-2-784-6027 FAX: +82-2-767-3677

GUMI OFFICE
2F, Grand B/D, 457-4 Songjeong-dong,
Gumi-City, KOREA
Phone: +82-54-454-6027 FAX: +82-54-454-6093

SEIKO EPSON CORPORATION
SEMICONDUCTOR OPERATIONS DIVISION

IC Sales Dept.
IC International Sales Group
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-42-587-5814 FAX: +81-42-587-5117

Document Code: 411283401
First Issue October 2007
Revised July 2008, in Japan

©