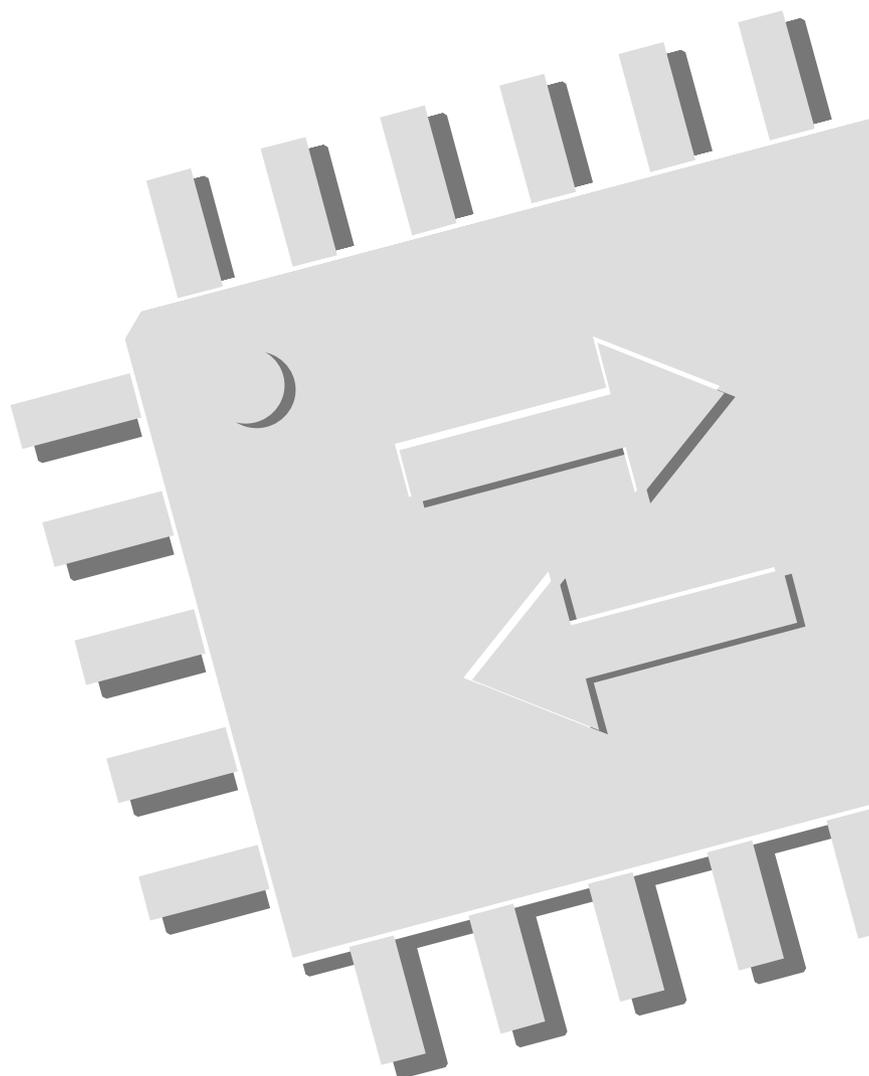


# LINK(IEEE 1394)

## MD8415B

### User's Manual

Preliminary



**MEMO**

---

**History of Revisions**

Issue No.	Date	Contents of Revision
0.8	14 May 1999	New descriptions for the MD8415.
0.9	02 July 1999	Description added and amended.
0.91	05 July 1999	Description added and amended.
0.92	05 July 1999	Wrong description amended.
0.93	23 July 2000	Description added and amended.

**Keys:**

Data MSB, LSB	: Left for MSB, right for LSB
Negative logic symbol description	: Mark # attached to the end of signal name.
Numerical descriptions	: Binary number           ****b or **** Decimal number       **** Hexadecimal number   ****h or 0x****
Table descriptions	: In the table of registers, etc., the hatched part without any name denotes an invalid area and there is no change in operation.
Terms	: Byte                       Data in 8-bit width. Word                       Data in 16-bit width. Quadlet                   Data in 32-bit width. Octlet                     Data in 64-bit width.

**Related materials:**

IEEE 1394-1995 Standard for a High Performance Serial Bus  
 IEEE p1394a  
 IEEE Std 1212-1991 Command and Status Register Architecture  
 MD8402 User's Manual  
 MD8404 User's Manual  
 MD8405 User's Manual

## Contents

1	Introduction .....	1
1-1	Features .....	1
1-2	Applications .....	1
1-3	Internal Block Diagram .....	2
1-4	Outline Functions .....	3
1-4-1	Host Interface.....	3
1-4-2	PHY Interface .....	3
1-4-3	Transmitter .....	3
1-4-4	Receiver .....	3
1-4-5	Built-in buffer .....	3
1-4-6	Asynchronous transfer function.....	4
1-4-7	Isochronous transfer function .....	4
1-4-8	Serial EEPROM I/F .....	4
2	Terminal description .....	5
2-1	Terminal arrays .....	5
2-2	Functional description of terminals.....	6
3	Control Register .....	9
3-1	Method of register access.....	9
3-2	Contents of register .....	10
3-2-1	Version Register .....	10
3-2-2	Control Register.....	11
3-2-3	Node Identification Register.....	15
3-2-4	Reset Register .....	17
3-2-5	Packet Control Register .....	18
3-2-6	Diagnostic Status and Control Register .....	21
3-2-7	Phy Control Register.....	25
3-2-8	ATRetry Register.....	26
3-2-9	Cycle Timer Register .....	27
3-2-10	Isochronous Transmit Configuration Register.....	28
3-2-11	Isochronous Transmit Configuration Register.....	29
3-2-12	Isochronous Bus Receive Configuration Register 1, 2, 3, 4.....	30
3-2-13	Isochronous Host Buffer Receive Configuration Register .....	32
3-2-14	Isochronous Bus Receive CT Configuration Register 1, 2.....	33
3-2-15	Isochronous Receive Error Register .....	34

3-2-16	Asynchronous Stream Receive Configuration Register.....	35
3-2-17	ATF Data Register .....	36
3-2-18	ARF Data Register .....	36
3-2-19	ITF Data Register.....	37
3-2-20	IRF Data Register.....	37
3-2-21	Buffer Status and Control Register .....	38
3-2-22	Asynchronous Bus Control Register 1 .....	40
3-2-23	Asynchronous Bus Control Register 2.....	41
3-2-24	Interrupt Register .....	42
3-2-25	Extended Interrupt Register .....	45
3-2-26	Interrupt Mask Register-1 .....	46
3-2-27	Interrupt Mask Register-2 .....	46
3-2-28	Split TimeOut Register .....	47
3-2-29	Split Timer Start Register.....	47
3-2-30	Bus Configuration Timer Register .....	48
3-2-31	Bus Configuration Timer Register 1 .....	49
3-2-32	Bus Configuration Timer Register 2.....	49
3-2-33	General Timer Register .....	50
3-2-34	EEPROM Control Register.....	51
3-2-35	EEPROM Data Register.....	52
3-2-36	GPIO Register .....	53
3-2-37	ATGo Register .....	54
3-2-38	ITGo Register.....	55
3-2-39	BInternal CSR Access Register .....	56
3-2-40	Internal CSR Access DataH Register.....	57
3-2-41	Internal CSR Access DataL Register .....	58
3-2-42	CSR Transaction Register 1 .....	59
3-2-43	CSR Transaction Register 2.....	59
3-2-44	CSR Transaction DataH Register.....	61
3-2-45	CSR Transaction DataL Register.....	62
3-2-46	CSR Config1 Register.....	63
3-2-47	CSR Config2 Register.....	65
3-2-48	CSR UserSpace Config Register.....	67
3-2-49	CSR UserSpace Config Register.....	68
3-2-50	CSR UserSpace Config Register.....	70
3-2-51	Active Node Map Register.....	72
3-2-52	Speed Map Register .....	73
3-3	List of registers .....	75

4	Host Access .....	80
4-1	Host Interface .....	80
4-2	Access flow after closure of the POWER switch.....	80
4-3	Register access timing.....	80
4-4	Host Bus Width.....	81
4-4-1	Host Bus Width.....	81
4-5	Access to the internal buffer .....	82
4-5-1	Software access.....	82
4-5-2	DMA access.....	83
5	PHY Chip Interface .....	86
5-1	Method of connection .....	86
5-2	Phy/Link I/F Timing .....	87
5-3	Access to the Phy register .....	89
5-4	Reading Request.....	90
5-5	Writing Request .....	90
6	Internal Buffer .....	91
7	Asynchronous Transmission/Reception.....	91
7-1	Internal CSR Access .....	91
7-1-1	When accessed via "1394" by the received Request packet.....	91
7-1-1-1	Read Request.....	93
7-1-1-2	Write/Lock request to Internal CSR (core).....	93
7-1-1-3	When set at Internal CSR.....	93
7-1-1-4	When set at Internal CSR (NoInt).....	95
7-1-1-5	When set at Host CSR.....	95
7-1-1-6	When set at Ack_Type_error .....	95
7-1-1-7	Write/Lock request to Internal CSR (UserSpace).....	95
7-1-1-8	Command Reset packet.....	95
7-2	Access to External CSR .....	96
7-2-1	Access to the buffer .....	96
7-2-2	Access via ATF/ARF register.....	97
7-2-3	Access via Async Bus.....	98
7-2-4	Writing in ATF buffer .....	99
7-2-5	Reading from the ARF buffer.....	100
7-3	Asynchronous Stream Transfer flow .....	101
7-3-1	Transmission of Asynchronous Stream.....	101
7-3-2	Reception of Asynchronous Stream .....	102
7-4	BusyCtrl processing .....	103

7-5	Retry Processing .....	104
7-6	Asynchronous Transmission/Reception Access flow .....	105
7-6-1	Transmission Access flow.....	110
7-6-2	Reception Access flow .....	110
7-7	BusNumber processing for Asynchronous Packet transmission .....	111
7-8	Processing for abnormality during Asynchronous Packet transmission .....	112
8	Internal Access to the Internal CSR .....	113
8-1	Write Access.....	113
8-2	Read Access.....	113
8-3	Lock Access.....	113
8-4	Access Flow.....	114
9	Definition of the Internal CSR.....	116
9-1	Internal CSR (UserSpace) .....	116
9-2	Internal CSR (core).....	116
9-2-1	STATE .....	116
9-2-2	STATE_CLEAR .....	118
9-2-3	STATE_SET .....	119
9-2-4	NODE_IDS .....	120
9-2-5	RESET_START.....	120
9-2-6	SPLIT_TIMEOUT .....	121
9-2-7	CYCLE_TIME.....	122
9-2-8	BUS_TIME .....	123
9-2-9	BUSY_TIMEOUT .....	124
9-2-10	BUS_MANAGER_ID.....	125
9-2-11	BANDWIDTH_AVAILABLE .....	126
9-2-12	CHANNEL_AVAILABLE.....	127
9-2-13	Configuration ROM .....	127
10	Isochronous Transmission/Reception.....	128
10-1	Master Mode.....	130
10-1-1	Isochronous Cycle Start Timing.....	130
10-1-2	Processing for Transmission request.....	131
10-1-3	Processing for Multi-Channel Transmmission.....	132
10-1-4	Processing for Reception.....	133
10-1-4-1	BE# Signal.....	134
10-1-4-2	SYNC Signal .....	134
10-1-5	CycleStartPacket Reception.....	135
10-1-6	Packet error announcement during reception.....	136

---

10-1-6-1	Error Announcement for Cycle Start Packet.....	136
10-1-6-2	Error Announcement for Isochronous Packet.....	136
10-1-7	Difference in timing of 100Mbps, 200Mbps, or 400Mbps.....	137
10-1-8	Sync Control .....	138
10-1-8-1	Sync control transmission by ITSYNC signal at auto-mode .....	138
10-1-9	BUSRST signal.....	139
10-1-10	ITF or IRF transfer.....	139
10-1-10-1	ITF transmission .....	139
10-1-10-2	IRF transmission .....	140
10-1-10-3	Normal/Auto mode .....	140
10-1-11	IsoBus added function .....	141
10-1-11-1	MyNodeID and SelfID Packet output.....	141
10-1-11-2	Snoop Mode .....	141
10-2	Slave Mode.....	142
10-2-1	Signals used in the IsochronousBus Slave mode.....	142
10-2-2	Signal for IsochronousBus Slave transmission.....	143
10-2-3	Signal for IsochronousBus Slave reception.....	144
10-2-4	Transmission/Reception flow .....	145
10-2-4-1	Transmission processing.....	145
10-2-4-2	Auto Mode .....	145
10-2-4-3	Reception processing .....	145
10-2-4-4	Auto Mode .....	145
11	EEPROM I/F.....	146
11-1	Type of EEPROM .....	146
11-2	Control signals .....	146
11-3	Storing format .....	147
11-4	Method of connection .....	147
11-5	Access flow .....	147
12	Internal Timer.....	148
13	SelfID Packet Analysis.....	149
14	Interrupt Processing.....	149
15	BusReset/CmdReset processing.....	150
15-1	BusReset processing.....	150
16	P1394a function.....	151
16-1	Isochronous Multi-Speed Concatenated Packet.....	151

16-2	Acceleration.....	151
17	Data Format .....	152
17-1	Asynchronous .....	152
17-1-1	Quadlet Transmit (WriteRequest / ReadRequest).....	152
17-1-2	Block Transmit (WriteRequest / LockRequest / ReadRequest).....	153
17-1-3	Quadlet Transmit (WriteResponse / ReadResponse).....	154
17-1-4	Block Transmit (LockResponse / ReadResponse).....	156
17-1-5	Quadlet Receive .....	157
17-1-6	Block Receive .....	159
17-1-7	Asynchronous Stream .....	161
17-1-7-1	Transmit.....	161
17-1-7-2	Receive .....	162
17-1-8	Asynchronous Bus Format.....	163
17-2	Isochronous Bus .....	164
17-2-1	For Normal Mode.....	164
17-2-1-1	Transmit.....	164
17-2-1-2	Receive .....	165
17-2-2	For Auto Mode.....	166
17-2-2-1	Transmit.....	166
17-2-2-2	Receive .....	166
17-3	Isochronous Buffer (ITF/IRF) .....	167
17-3-1	For Normal Mode.....	167
17-3-1-1	Transmit.....	167
17-3-1-2	Receive .....	167
17-3-2	For Auto Mode.....	168
17-3-2-1	Transmit.....	168
17-3-2-2	Receive .....	168
17-4	Snoop.....	169
17-5	MyNodeID.....	169
17-6	SelfID Packet.....	170
17-6-1	ARF Buffer.....	170
17-6-2	IsoBus.....	172
17-7	PHY Packet .....	173
17-7-1	Transmit .....	173
17-7-2	Reception.....	173
17-8	Code.....	174
18	Electrical Characteristics .....	177
18-1	Absolute Rating .....	177

---

18-2 Recommended Operating Condition.....	177
18-3 DC Characteristics .....	178
18-4 AC Characteristics .....	179
19 Package Outline.....	190
Notes.....	191

## Figure and Table Contents

Figure 1-3-1	MD8415 Block Diagram.....	2
Figure 2-2-1	MD8415 Terminals (3) .....	8
Figure 3-1-1	Register Address on 8-Bit Bus.....	9
Figure 3-1-2	Register Addresses for the 16-bit Bus.....	9
Figure 4-3-1	Host Access Timing .....	80
Figure 4-5-1	Register Operation for Software Access in 8-Bit Width (ATF) .....	82
Figure 4-5-2	Register Operation for Software Access in 8-Bit Width (ARF) .....	82
Figure 4-5-3	Register Operation for Software Access in 16-bit Width (ATF) .....	82
Figure 4-5-4	Register Operation for Software Access in 16-bit Width (ARF).....	83
Figure 4-5-5	Register Operation for DMA Access in 8-Bit Width (ATF).....	83
Figure 4-5-6	Register Operation for DMA Access in 8-Bit Width (ARF) .....	84
Figure 4-5-7	Register Operation for DMA Access in 16-Bit Width (ATF).....	84
Figure 4-5-8	Register Operation for DMA Access in 16-Bit Width (ARF) .....	85
Figure 5-1-1	Diagram of Connections between MD8415 and PHY Chip .....	86
Figure 5-1-2	Diagram of Connections between MD8415 and MD8404 .....	86
Figure 5-2-1	LPS output waveform for AC connection.....	87
Figure 5-2-2	PHY-LINK I/F Reset Sequence for PHYIFRST='0' and AC connection.....	88
Figure 5-2-3	PHY-LINK I/F Reset Sequence for PHYIFRST='1' and AC connection.....	88
Figure 5-3-1	Access to the PHY .....	89
Figure 7-2-1	Asynchronous Buffer Access.....	96
Figure 7-2-2	HostBus and AsyncBus.....	97
Figure 7-2-3	AsyncBus signal.....	98
Figure 7-2-4	AsyncBus signal (ATF).....	99
Figure 7-2-5	AsyncBus signal (ARF) .....	100
Figure 7-4-1	BusyCtrl Processing.....	103
Figure 7-5-1	Retry Transaction.....	104
Figure 7-6-1	InternalCSR(core) Access.....	105
Figure 7-6-2	ConfigROM Access .....	105
Figure 7-6-3	InternalCSR(UserSpace) Access.....	106
Figure 7-6-4	External CSR Access .....	106
Figure 8-4-1	Access Flow when Access from "1394" is for Read.....	114
Figure 8-4-2	Access Flow when Access from "1394" is for Write/Lock .....	115
Figure 10-0-1	For Master Mode, ITF.....	128
Figure 10-0-2	ForMaster Mode, IRF.....	128
Figure 10-0-3	For Slave Mode, IRF.....	129
Figure 10-0-4	For Slave Mode, ITF.....	129

Figure 10-1-1	Isochronous Transmission Timing .....	131
Figure 10-1-2	Multi-channel Isochronous Transmission Timing.....	132
Figure 10-1-3	Isochronous Reception Timing.....	133
Figure 10-1-4	CycleStartPacket Reception.....	135
Figure 10-1-5	Error Timing during Cycle Start Packet .....	136
Figure 10-1-6	IRERR# Appearing Pattern .....	137
Figure 10-1-7	100/200/400Mbps Timing .....	137
Figure 10-1-8	ITSYNC Timing .....	138
Figure 10-1-9	Relationship between ITSYNC Signal and sync Field.....	138
Figure 10-1-10	BUSRST signal.....	139
Figure 10-1-11	MyNodeID, SelfID, Packet.....	141
Figure 10-2-1	Buffer access for Isochronous slave .....	142
Figure 10-2-2	IsochronousBus signal (transmission) .....	143
Figure 10-2-3	IsochronousBus signal (reception) .....	144
Figure 11-4-1	Connection through EEPROM .....	147
Figure 11-3-1	EEPROM Format.....	147
Figure 18-4-1	Host Interface AC Characteristics (Read/Write).....	180
Figure 18-4-2	Host Interface AC Characteristics (Reset).....	180
Figure 18-4-3	Host Interface AC Characteristics (DMA) .....	181
Figure 18-4-4	Host Interface AC Characteristics (CYCLEOUT) .....	182
Figure 18-4-5	PHY Interface AC Characteristics (SCLK).....	182
Figure 18-4-6	PHY Interface AC Characteristics (CTL,D).....	183
Figure 18-4-7	PHY Interface AC Characteristics (LREQ).....	183
Figure 18-4-8	PHY Interface AC Characteristics (LPS) .....	184
Figure 18-4-9	Isochronous Bus AC Characteristics (For Master) 1 .....	185
Figure 18-4-10	Isochronous Bus AC Characteristics (For Master) 2.....	185
Figure 18-4-11	Isochronous Bus AC Characteristics 3 .....	186
Figure 18-4-12	Isochronous Bus AC Characteristics (For Slave) 1 .....	186
Figure 18-4-13	Isochronous Bus AC Characteristics (For Slave) 2 .....	187
Figure 18-4-14	Asynchronous Bus AC Characteristics 1.....	188
Figure 18-4-15	Asynchronous Bus AC Characteristics 2.....	188
Figure 18-4-16	EEPROM Interface AC Characteristics.....	188
Table 2-2-1	MD8415 Terminals (1).....	6
Table 2-2-2	MD8415 Terminals (2).....	7
Table 3-3-1	List of registers 1 .....	75
Table 3-3-2	List of registers 2 .....	76
Table 3-3-3	List of registers 3 .....	77
Table 3-3-4	List of registers 4 .....	78

Table 3-3-5	List of registers 5.....	79
Table 4-4-2	Little / Big Endian Mode.....	81
Table 4-4-1	Valid Host Data Bus for Access from the Host .....	81
Table 5-2-1	LPS output.....	87
Table 5-2-2	LPS output characteristics for AC connection.....	87
Table 7-3-1	Asynchronous Stream Transmit format .....	101
Table 7-3-2	Asynchronous Stream Receive format (Isochronous: normal) .....	102
Table 10-1-1	Conditions for Isochronous reception (Master) .....	133
Table 10-1-2	Contents of CTm.....	135
Table 10-1-3	Contents of CTl.....	135
Table 10-2-1	IsoReceive Conditions (Slave).....	145
Table 11-2-1	Slave Addresses .....	146
Table 11-2-2	Slave Addresses strings.....	146
Table 17-1-1	Quadlet Transmit format (Asynchronous) .....	152
Table 17-1-2	Block Transmit format (Asynchronous) .....	153
Table 17-1-3	Quadlet Transmit fresponse format (Asynchronous).....	154
Table 17-1-4	Block Transmit Response format (Asynchronous).....	156
Table 17-1-5	Quadlet Receive format (Asynchronous).....	157
Table 17-1-6	Block Receive format (Asynchronous).....	159
Table 17-1-7	Asynchronous Stream Transmit format .....	161
Table 17-1-8	Asynchronous Stream Receive format (Isochronous:normal).....	162
Table 17-1-9	Asynchronous Bus format.....	163
Table 17-2-1	Block Transmit format (Isochronous:normal) .....	164
Table 17-2-2	Block Receive format (Isochronous:normal).....	165
Table 17-2-3	Block Transmit format (Isochronous:auto).....	166
Table 17-2-4	Block Receive format (Isochronous:auto) .....	166
Table 17-3-1	Block Transmit format (Isochronous:normal) .....	167
Table 17-3-2	Block Receive format (Isochronous:normal).....	167
Table 17-3-3	Block Receive format (Isochronous:normal).....	168
Table 17-3-4	Block Receive format (Isochronous:auto) .....	168
Table 17-4-1	Snoop Receive format.....	169
Table 17-5-1	MyNodeID Output Format.....	169
Table 17-6-1	SelfID Packet Receive format (first quadlet).....	170
Table 17-6-2	SelfID Packet Receive format (SelfID Packet #0).....	170
Table 17-6-3	SelfID Packet Receive format (SelfID Packet #1, #2, & #3).....	170
Table 17-6-4	SelfID Packet Receive format (last quadlet).....	170
Table 17-6-5	SelfID Packet Receive format (pn) .....	170
Table 17-6-6	SelfID Packet Output format.....	172
Table 17-7-1	PHY packet format (first quadlet).....	173

---

Table 17-7-2	PHY packet format (first quadlet) .....	173
Table 17-7-3	PHY format.....	173
Table 17-7-4	PHYformat (last quadlet).....	173
Table 17-8-1	List of Retry code .....	174
Table 17-8-2	List of Transaction code (tCode) .....	174
Table 17-8-3	List of Bus Number / Node Number.....	174
Table 17-8-4	List of Data Length (Data Length) .....	174
Table 17-8-5	List of Extension Transaction Code (Extend tCode) .....	175
Table 17-8-6	List of Speed Codes (spd).....	175
Table 17-8-7	List of Acknowledge Codes (Ack) .....	176
Table 17-8-8	List of rcodes .....	176
Table 18-4-1	Host Interface AC Characteristics .....	179
Table 18-4-2	PHY Interface AC Characteristics.....	182
Table 18-4-3	Isochronous Bus AC Characteristics (For Master).....	184
Table 18-4-4	Isochronous Bus AC Characteristics (For Slave) .....	186
Table 18-4-5	Asynchronous Bus AC Characteristics.....	187
Table 18-4-6	EEPROM Interface AC Characteristics.....	189

**MEMO**

## 1 Introduction

The MD8415 is a link-layer controller of the high-speed serial bus, conforming to IEEE Standard 1394-1995. In addition to the various functions necessary for the link layer, the MD8415 is capable of supporting the register of the CSR space specified by IEEE 1394-1995. The asynchronous/isochronous packet offers maximum transfer performance by gaining access to the outside through an exclusive bus.

### 1-1 Features

- Packing during transmission and unpacking during reception, conforming to IEEE1394-1995.
- Supporting the CycleMaster.
- Supporting automatic activation of the Cycle Master.
- Parity generation by 32-bit CRC and error detection.
- Detection of a dropped cycle start message.
- Interface with the PHY chip (MD8402) supported by direct and AC coupling.
- Data bus exclusively provided for asynchronous transmission and reception.
- Controlling the number of transfer actions in each cycle during isochronous transfer.
- Automatic insertion of a header during isochronous packet transmission, and automatic separation of a header during reception.
- Full support of out-band retry sequence.
- Incorporating an exclusive data bus for isochronous transmission and reception, and an isochronous FIFO.
- Supporting a control signal toward the LPS Link Power Status) of the PHY (MD8402).
- Supporting the register of the CSR space specified by IEEE 1394-1995.

### 1-2 Applications

- |                               |               |                 |
|-------------------------------|---------------|-----------------|
| - Digital Camera              | - Digital VTR | - Digital Audio |
| - Electronic Music Instrument | - Scanner     | - Printer       |
| - Each Storage                | - DVD         | - Set Top Box   |

1-3 Internal Block Diagram

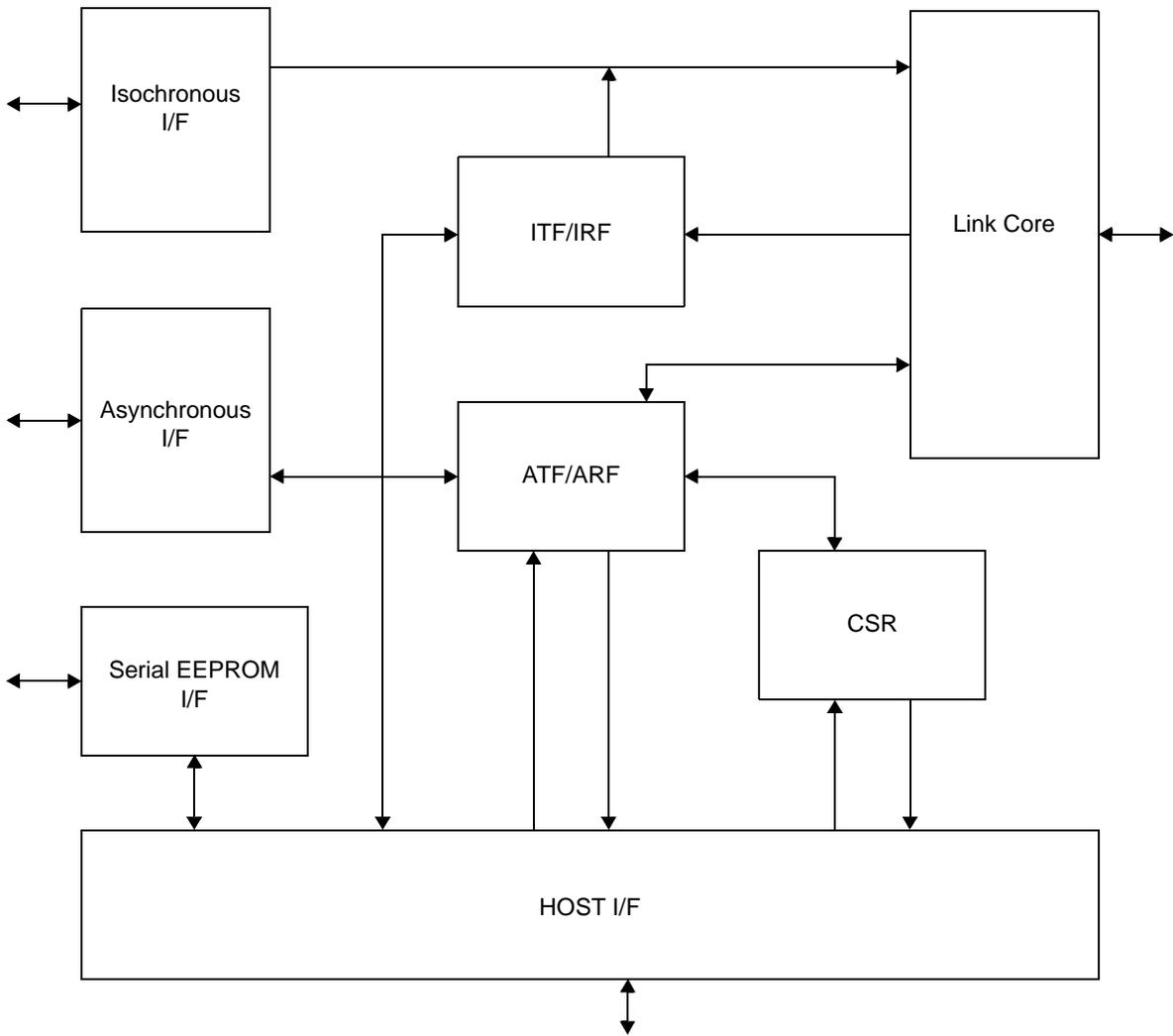


Figure 1-3-1 MD8415 Block Diagram

## 1-4 Outline Functions

### 1-4-1 Host Interface

The host interface is composed of an SRAM-like asynchronous bus with a bit width of 8/16/32 bits. Since it has a DMA control function inside, it is capable of high-speed data transfer by generating a DREQ signal according to the condition in the buffer.

The 8/16/32 bus width changeover can be controlled by the use of the USW#, UBE#, A1, and A0 signals. By this control, it is possible to change the register and buffer access. Any register can be directly accessed from the host. For DMA transfer, internal buffer selection is effected to enable gaining access to the selected buffer.

### 1-4-2 PHY Interface

The MD8415 is provided with an interface that can be directly connected with a PHY chip intended for the processing of the physical layer in accordance with the IEEE 1394 Standard. The objective PHY chip to be connected may be any of 100, 200, and 400Mbps.

The IEEE 1394 Draft defines two types of connection modes for the PHY chip and the LINK chip.

- DC connection

- AC connection

This IC supports the both way connection mode.

### 1-4-3 Transmitter

The transmitter reads out data from the asynchronous transmission buffer in the MD8415, or from the isochronous transmission data bus. The read data are arranged into various packet formats defined in IEEE1394. The formatted packets are sent to the PHY interface. When the CycleMaster bit is "1" and the node using the MD8415 is a root, a CycleStart packet for showing the head of the isochronous cycle is also sent out.

### 1-4-4 Receiver

The receiver receives a packet from the PHY interface and identifies whether the received packet is the one to be acquired by the MD8415 node. If it is found to be an asynchronous packet, this judgment is made with the node address of the MD8415. If it is an isochronous packet, such a judgment is conducted with the preset channel number. When the packet is found to be addressed to this node, writing is effected in the asynchronous reception buffer and data output is performed toward the isochronous data bus, respectively. No judgment is conducted in the case of a broadcast packet and the snoop mode. In this case, writing is effected in the respective buffers and data output is performed toward the data buses.

### 1-4-5 Built-in buffer

ATF and ARF are incorporated for asynchronous performance and ITF/IRF is used for isochronous performance.

Buffer	Meaning	Size
ATF	For ordinary asynchronous transmission	2048 + 56 byte
ARF	For packet reception in external CSR space	2048 + 56 byte
ATFi	For response packet transmission for a request toward internal CSR space	512 + 28 byte
ARFi	For packet reception in internal CSR space	28 byte
ITF/IRF	For isochronous transmission/reception	2048 + 8 byte

#### 1-4-6 Asynchronous transfer function

For asynchronous transfer, an ATF or ARF buffer is used. An access to the ATF/ARF buffer is gained by any one of the two methods; an access from the host via an internal register or an asynchronous transfer from the asynchronous bus. Each method can be switched over to the other at any time.

The core part of the CSR space has a function of processing to be accomplished inside the MD-8415. For this reason, this processing is always accomplished inside the MD-8415 even when, for example, a read request to the configuration ROM arises outside. As a result, a response packet is automatically transmitted without any support from the MPU.

#### 1-4-7 Isochronous transfer function

The MD8415 is provided with isochronous functions. It incorporates a cycle timer. When the node using the MD8415 is of the CycleMaster, it is possible to transmit a CycleStart packet in the unit of 125 $\mu$ sec. The trigger to be used may be produced the cycle from the 49.152MHz clock input from the PHY chip, or an 8kHz signal input from the CYCLEIN pin may be used. If the node is not for the CycleMaster, synchronism with the CycleMaster is secured by making compensation for the internal CycleTimer of the MD8415, each time a CycleStart packet is received from another CycleStart node, using the value in that packet.

Isochronous transfer is supported by any one of the two methods; transmission/reception is performed at a bus timing of IEEE 1394 without using any internal buffer, or isochronous transfer is effected through an internal ITF (or RTF).

The user can decide a preferable mode according to the nature of the data source to be handled in the isochronous transfer mode.

If no internal buffer is used, an access of isochronous packet to the outside is gained by synchronous transfer by the use of an exclusive isochronous data bus. At that time, data control in conjunction with the outside is effected by the MD8415 that functions as a master unit.

If any internal buffer is used, an access of isochronous packet to the outside is similarly gained by synchronous transfer by the use of an exclusive isochronous data bus. In this case, however, data control in conjunction with the outside is effected from the outside. The MD8415 performs data transfer according to the outside clock signal and the read/write request.

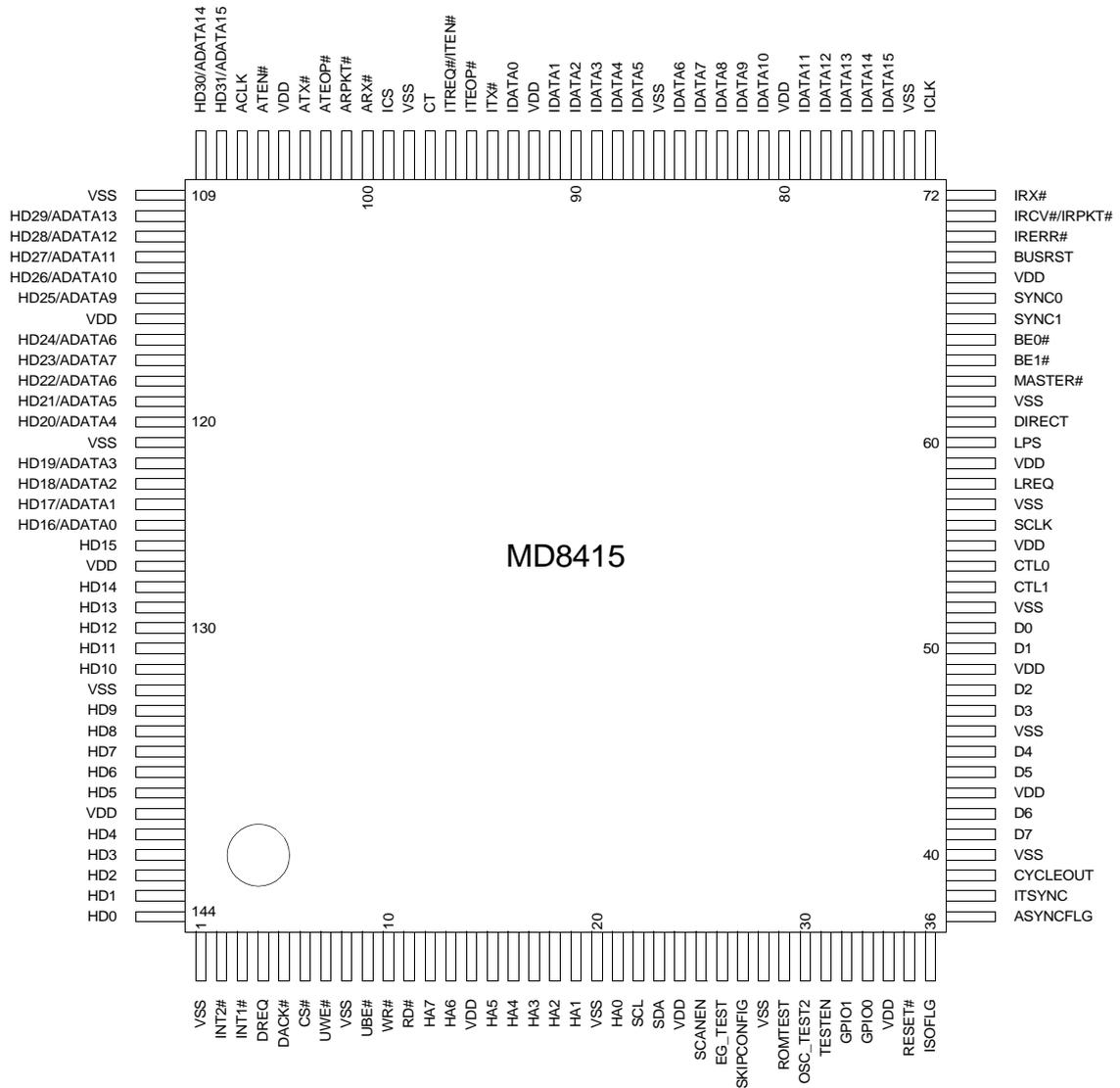
#### 1-4-8 Serial EEPROM I/F

The MD8415 incorporates an interface that can be directly connected to a serial EEPROM where a configuration ROM of the CSR space is loaded. The supporting device is specified below.

NM24Cxx by Fairchild, Inc.

## 2 Terminal description

### 2-1 Terminal arrays



## 2-2 Functional description of terminals

Pin name	I/O	Pin no.	No. of Pin	Contents															
PHY I/F																			
SCLK	I	56	1	Master clock: A clock signal of 49.152MHz supplied from the PHY chip. The MD8415 uses this clock as a master clock signal. Usually connected to this signal pin of the PHY chip.															
LREQ	O	58	1	link request: This signal is used by the MD8415 when gaining access to the register in the PHY chip and a request is made to use a serial bus. Usually, a connection is made to this signal pin of the PHY chip.															
CTL(0:1)	I/O	54, 53	2	PHY-LINK control: An interface control signal toward the PHY chip. Usually connected to this signal pin of the PHY chip.															
D(0:7)	I/O	51, 50, 48, 47, 45, 44, 42, 41	8	PHY I/F data bus: This is a data bus for data transmission and reception with the PHY chip. D[0:1] is used for packet transmission and reception at 100Mbps, and D[0:3] at 200bps.															
DIRECT	I	61	1	PHY I/F direct select signal: A changeover signal for a direct connection of the I/F to the PHY chip or an isolation connection. L: Isolation connection. H: Direct connection.															
LPS	O	60	1	Link power status: An LPS signal toward the PHY. According to the register setting, an output is generated in any of the following combinations. <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Pin input</th> <th>LPSOn bit</th> <th>LPS output</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>Clock output approx. 2MHz</td> </tr> </tbody> </table>	Pin input	LPSOn bit	LPS output	1	0	0	1	1	1	0	0	0	0	1	Clock output approx. 2MHz
Pin input	LPSOn bit	LPS output																	
1	0	0																	
1	1	1																	
0	0	0																	
0	1	Clock output approx. 2MHz																	
Host I/F																			
HA(7:0)	I	12,13,15, 16, 17, 18, 19, 21	8	Host address: Host address for register selection															
HD(15:0)	I/O	126,128,129, 130,131,132, 134,135,136, 137,138,140, 141,142,143, 144	16	Host data bus: Data bus for register and data access															
WR#	I	10	1	Write enable: Write signal for host data bus															
RD#	I	11	1	Read enable: Read signal for host data bus															
CS#	I	6	1	Chip select: Chip select signal of host data bus															
UBE#	I	9	1	Upper byte enable: Upper byte access signal for host data bus															
UWE#	I	7	1	Upper word enable: Upper word access signal for host data bus															
DREQ	O	4	1	Data request: A DMA transfer request signal. Effective only for the asynchronous packet or data transfer.															
DACK#	I	5	1	Data acknowledge: A DMA transfer acknowledge signal. Effective only for the asynchronous packet or data transfer.															
INT1#	O	3	1	Interrupt signal 1: An interrupt signal used for notification to the host. When a request arises in the interrupt register, this signal is asserted. The selection of this request can be setup with the register.															
INT2#	O	2	1	Interrupt signal 2: An interrupt signal used for notification to the host. When a request arises in the interrupt register, this signal is asserted. The selection of this request can be setup with the register.															

Table 2-2-1 MD8415 Terminals (1)

Pin name	I/O	Pin no.	No. of Pin	Contents
Isochronous I/F				
ICLK	I/O	73	1	Isochronous bus master clock: A master clock for each isochronous interface signal to be output by the MD8415 while Master#="L". An output of 24.576MHz is generated to count the cycle timer from the outside.
ITREQ#/ITEN#	I/O	96	1	Isochronous transmission request:
ITX#	I/O	94	1	Isochronous packet transmission data enable: : If there is a transmission request and this signal is placed, the MD8415 makes data request from the outside. In synchronization with this signal, the outside is required to perform data handling without fail.
IRX#	I/O	72	1	Isochronous packet reception data enable
ITEOP#	I	95	1	Isochronous transmission end of packet: A signal used to indicate the end of a packet during transmission from IsoBus.
IRCV#/IRPKT#	O	71	1	Isochronous packet reception enable:
IRERR#	O	70	1	Isochronous packet error flag: Asserted in the case of a CRC or data error in the received isochronous packet.
IDATA(15:0)	I/O	75,76,77,78, 79,81,82,83, 84,85,87,88, 89,90,91,93	16	Isochronous data bus: An exclusive bus for isochronous data access. The cycleTimer value in the CycleStart packet is also output from this bus.
CT	O	97	1	Cycle Timer enable: When a cycle-start packet is transmitted in the root mode or received in the non-root mode, this signal indicates that the cycle time data only in the packet are output to the IDATA bus.
BE#(1:0)	O	64,65	2	Byte enable: Indicates that the effective data of isochronous reception are output, to the upper 8 bits and the lower 8 bits of the IDATA bus, respectively. BE#(1) denotes IDATA(15:8) and BE#(0) denotes IDATA(7:0).
SYNC(1:0)	O	66,67	2	In the case of coincidence with the value that has been set with the Sync bit in the received isochronous packet, the following codes are output: 10b: Coincidence with StopSync 01b: Coincidence with StartSync 11b: Between Start and Stp 00b: Other than the above
BUSRST	O	69	1	When BusReset is occurred, a pulse of 25MHz is output.
ICS	O	99	1	A timing signal for transmission/reception of an isochronous Cycle Start packet is output.
MASTER#	I	63	1	Isochronous Bus Master/Slave Mode MASTER#="L": Master Mode MASTER#="H": Slave Mode

Table 2-2-2 MD8415 Terminals (2)

Pin name	I/O	Pin no.	No. of Pin	Contents
Asynchronous I/F				
ACLK	I	106	1	Asynchronous bus master clock
ATEN#	O	105	1	Asynchronous bus transmission enable signal
ATX#	I	103	1	Asynchronous bus transmission data enable signal
ARX#	I	100	1	Asynchronous bus reception data enable signal
ATEOP#	I	102	1	Asynchronous bus transmission data end signal
ARPKT#	O	101	1	Asynchronous bus reception packet enable signal
HD(31:16)/ ADATA(15:0)	I/O	107,108,110, 111,112,113, 114,116,117, 118,119,120, 122,123,124, 125	16	Host data bus upper, 6-bit [HD (31:16)]: A host data access bus for the 32-bit access mode. Asynchronous bus data bus (ADATA(15:0)): Data access bus for asynchronous bus
EEPROM I/F				
SCL	OD	22	1	EEPROM serial clock: A clock signal is output to gain access to the EEPROM.
SDA	I/OD	23	1	EEPROM serial data: Data are output or input for gaining access to the EEPROM.
others				
CYCLEOUT	O	39	1	CYCLE Isochronous cycle output: A cycle clock output generated by counting at the cycle timer in the MD8415.
ITSYNC	I	38	1	Sync control signal: Setting control signal for the sync field in the header, to be used during auto-mode transmission.
ASYNCF LG	O	37	1	Asynchronous flag: Asserted when an asynchronous packet is transmitted to the PHY or an asynchronous packet is received from the PHY. This is useful as a trigger signal for packet transmission/reception.
ISOFLG	O	36	1	Isochronous flag: Asserted when an isochronous packet is transmitted to the PHY or an isochronous packet is received from the PHY. This is useful as a trigger signal for packet transmission/reception.
RESET#	I	35	1	Reset: A system reset signal for the MD8415.
GPIO(1:0)	I/O	32,33	2	General-purpose I/O.
Test				
TESTEN		31	1	Test signal: In ordinary operation, this terminal must be connected to GND.
OSC_TEST2		30	1	Test signal: In ordinary operation, this terminal must be connected to GND.
ROMTEST		29	1	Test signal: In ordinary operation, this terminal must be connected to GND.
SKIPCONFIG		27	1	Test signal: In ordinary operation, this terminal must be connected to GND.
EG_TEST		26	1	Test signal: In ordinary operation, this terminal must be connected to GND.
SCANEN		25	1	Test signal: In ordinary operation, this terminal must be connected to GND.
Total Signal pin				

Figure 2-2-1 MD8415 Terminals (3)

### 3 Control Register

#### 3-1 Method of register access

In the 32-bit access mode, each register is provided with a description of the front address only. In the case of 8- or 16-bit access, therefore, the addresses specified below are used.

The read-only register is available only for reading operation. Writing operation should not be performed. If writing operation is forcibly performed, such operation is not guaranteed.

	7	6	5	4	3	2	1	0
Index +3	BYTE(adrs+3)							
	15	14	13	12	11	10	9	8
Index +2	BYTE(adrs+2)							
	23	22	21	20	19	18	17	16
Index +1	BYTE(adrs+1)							
	31	30	29	28	27	26	25	24
Index	BYTE(adrs)							

Figure 3-1-1 Register Address on 8-Bit Bus

	7	6	5	4	3	2	1	0
Index +2	WORD-L(adrs+2)							
	15	14	13	12	11	10	9	8
	WORD-H(adrs+2)							
	23	22	21	20	19	18	17	16
Index	WORD-L(adrs)							
	31	30	29	28	27	26	25	24
	WORD-H(adrs)							

Figure 3-1-2 Register Addresses for the 16-bit Bus

### 3-2 Contents of register

#### 3-2-1 Version Register

Address 00h

Initial value 0005 0000h

This register is used to indicate the version number of the MD8415. It is useful when the IEEE 1394-LINK chips are controlled in the software in the future.

7	6	5	4	3	2	1	0
Revision							
15	14	13	12	11	10	9	8
Revision							
23	22	21	20	19	18	17	16
Version							
31	30	29	28	27	26	25	24
Version							

**Bit 15~0**      **Revision:** Revision number of IC chip (R - Initial value: 0001h)

Indicates the revision number of the MD8415. The value begins with 0 and is increased each time revision is made.

**Bit 31~16**      **version:** Revision number of IC chip (R - Initial value: 0005h)

Indicates the version number of the MD8415. The MD8415 reads out "0005h" at all times.

### 3-2-2 Control Register

Address        04h  
Initial value   1803 0000h

This register is used to set up configuration, enable, etc. of each operation in the MD8415. Usually, this register is set up after making the setting of LPSON bit shortly after the closure of the POWER switch. At that time, the MD8415 configuration should have been decided.

7	6	5	4	3	2	1	0
PHYIFRST	ROMChkEnd					ReceiveEn	TransmitEn
15	14	13	12	11	10	9	8
		DMAWidth			IsoConEn	AckAccEn	cmstr
23	22	21	20	19	18	17	16
IsoBusModeH	IsoBusModeL	ITF/IRF	IsoDataMode	ITZERO		CycleMaster	CycleTimerEn
31	30	29	28	27	26	25	24
DackHigh	DreqLow	ITBusReset	ARFBusReset	CSEn	IACTIVE	AACTIVE	LPSON

#### Bit 0            **TransmitEn:** Transmitter Enable bit (RW - Initial value: 1b)

- 0    = Transmitter disabled.
- 1    = Transmitter enabled.

This setting is made to define whether the MD8415 transmitter should be enabled or not. When enabled, the transmission shown below is carried out.

- Asynchronous packet
- A CycleStart packet when the CycleMaster bit is enabled.
- An isochronous packet in the CycleStart mode.
- Phy packet

When disabled, no transmission is carried out.

#### Bit 1            **ReceiveEn:** Receiver Enable bit (RW - Initial value: 0b)

- 0    = Receiver disabled.
- 1    = Receiver enabled.

This setting is made to define whether the MD8415 receiver should be enabled or not. When enabled, the transmission shown below is carried out.

- An asynchronous packet addressed from other node to this node.
- An isochronous packet of the specified channel.
- Reception in the snoop mode.

When disabled, the following reception only is carried out.

- SelfID packet

#### Bit 6            **ROMChkEnd:** EEPROM Check End bit (RW - Initial value: 0b)

- 0    = EEPROM CRC being checked.
- 1    = EEPROM CRC check finished.

An announcement is given to indicate the end of EEPROM CRC check after hardware reset (RESET#).

- Bit 7**            **PHYIFRST:** Phy-Link I/F Reset bit (RW - Initial value: 0b)  
 0    = Initialized at the default timing.  
 1    = Initialized at a timing specified by P1394a.  
 Selection is made to determine whether Phy-Link I/F is initialized at a default timing or a timing specified by P1394a.
- Bit 8**            **cmstr:** cmstr bit (R - Initial value: 0b)  
 Reflection of the busdepend.cmstr for the STATE register of the internal CSR (core).
- Bit 9**            **AckAccEn:** Ack-acceleration Enable bit (RW - Initial value: 0b)  
 0    = Ack-Acceleration disabled.  
 1    = Ack-Acceleration enabled.  
 This is a setting bit to determine whether the Ack-Acceleration of the MD8415 should be enabled or not. When set to be enabled, the function of Ack-Acceleration defined by P1394a becomes effective.
- Bit 10**           **IsoConEn:** Isochronous Concatenation Enable bit (RW - Initial value: 0b)  
 0    = Isochronous Concatenation disabled.  
 1    = Isochronous Concatenation enabled.  
 This is a setting bit used to determine whether the isochronous concatenation function of the MD8415 should be enabled or not. When set to be enabled, the function of isochronous concatenation defined by P1394a becomes effective.
- Bit 12~13**       **DMAWidth:** DMA Transfer Data Width field (RW - Initial value: 00b)  
 00   = 8 bit (Byte) transfer  
 01   = 16 bit (Word) transfer  
 10   = 32 bit (Quadlet) transfer  
 11   = Reserve.
- Bit 16**           **CycleTimerEn:** Cycle Timer Enable bit (RW - Initial value: 1b)  
 0    = CycleTimer disabled.  
 1    = CycleTimer enabled.  
 This bit is used to determine whether the CycleTimer in the MD8415 should be enabled or not. Since the internal timer also uses a CycleTimer, the bit should be set at "1" even though the CycleTimer is not used.
- Bit 17**           **CycleMaster:** Cycle Master bit (RW - Initial value: 1b)  
 0    = A CycleStart packet is received from another root node in order to control the CycleTimer. This bit is set at "0" for a node that cannot be a root in ordinary cases.  
 1    = When this bit is "1", a CycleStart packet is generated each time the CycleTimer of the MD8415 is carried. Even though this bit is set, no CycleStart packet is generated if the root bit is "0".
- Bit 19**           **ITZERO:** Isochronous ZeroLength Packet bit (RW - Initial value: 0b)  
 0    = For isochronous transmission in AUTO mode, transmission is suspended for a cycle that has no request for transmission.  
 1    = For isochronous transmission in AUTO mode, a packet of Length=0 is transmitted for a cycle that has no request for transmission.  
 For isochronous transmission, this bit is used to determine whether a packet of Length=0 is automatically transmitted or whether packet transmission is not performed for a cycle where ITREQ# is not asserted, or in other words if there are no data to be transmitted for a cycle in AUTO mode.

- Bit 20**      **IsoDataMode:** Isochronous Data Mode bit (RW - Initial value: 0b)  
Possible in 1ch only, for both transmission and reception in AUTO mode.  
IsoDataMode  
0    = Isochronous transmission/reception mode for a normal mode where no isochronous header is automatically attached.  
1    = Isochronous transmission/reception mode for an AUTO mode where an isochronous header is automatically attached.
- Bit 21**      **ITF/IRF:** ITF/IRF Select bit (RW - Initial value: 0b)  
ITF/IRF  
0    = ITF/IRF set at ITF.  
1    = ITF/IRF set at IRF.
- Bit 22**      **IsoBusModeL:** Isochronous Bus Mode Low bit (RW - Initial value: 0b)  
IsoBusModeL  
0    = Isochronous transmission/reception mode. In reception mode, a maximum of 4 channels can be received.  
1    = Isochronous transmission/reception mode. In reception mode, an isochronous snoop is received.  
This is a reception mode changeover setting bit for an isochronous bus. If IsoBusModeH is in the slave mode, however, setting for IsoBusModeL is invalid and only one channel is received during reception. It is invalid in AUTO mode.
- Bit 23**      **IsoBusModeH:** Isochronous Bus Mode High bit (R - Initial value: 0b)  
The status of MASTER# is indicated for signal pins.  
IsoBusModeH  
0    =Master mode (MASTER#= "L")  
1    =Slave mode (MASTER#= "H")
- Bit 24**      **LPSOn:** Link Power Status On bit (RW - Initial value: 0b)  
This is a control bit for the LPS signal that is fed to the PHY chip. The contents of output are different according to the condition at the DIRECT terminal. Usually, a connection is made to the LPS terminal of the PHY chip. If this bit is "0", however, the PHY is informed that the LINK chip is not active, and the supply of the SCLK (49.152MHz) signal output from the PHY is suspended. In a state that this LPSOn bit is not set at "1", it is impossible to gain access to the PHY where the MD8415 is connected. In other words, all registers other than this control register cannot be accessed. For this reason, it is necessary to make setting of "1" after the closure of the POWER switch, without fail.
- | DIRECT terminal | LPSOn bit | LPS terminal output   |
|-----------------|-----------|-----------------------|
| High            | 0         | Low                   |
| High            | 1         | High                  |
| Low             | 0         | Low                   |
| Low             | 1         | Clock of approx. 2MHz |
- Bit 25**      **AACTIVE:** ASYNCFLG terminal bit (RW - Initial value: 0b)  
0    = The assert condition of the ASYNCFLG terminal is established for transmission or when an async packet addressed to this node is received.  
1    = The assert condition of the ASYNCFLG terminal is established for transmission or when all async packets are received.

This bit is used to determine the assert condition of the ASYNCFLG terminal. It is effective as a trigger signal for asynchronous packet transmission/reception.

- Bit 26**            **IACTIVE:** ISOFLG terminal bit (RW - Initial value: 0b)
- 0    = The assert condition of the ISOFLG terminal is established for transmission or when a channel set at the Isochronous Receive register is received.
  - 1    = The assert condition of the ISOFLG terminal is established for transmission or when all isochronous packets are received.
- This bit is used to determine the assert condition of the ISOFLG terminal. It is effective as a trigger signal for isochronous packet transmission/reception.

- Bit 27**            **CSEn:** Cycle Start Packet output enable bit (RW - Initial value: 1b)
- 0    = No Cycle Time data output in the CycleStart packet is sent to the IDATA bus.
  - 1    = The Cycle Time data output in the CycleStart packet is sent to the IDATA bus.
- When a CycleStart packet is received from the serial bus, the MD8415 sends out to the IDATA bus an output of the Cycle Timer data only in this packet. This bit is used to set up "output enable" for the said data.
- This setting is made to define whether the output should be addressed to the IsochronousBus side of the CycleStart packet. The status of a Link cannot be modified by this setting.

- Bit 28**            **ARFBusReset:** ARF BusReser mode bit (RW - Initial value: 1b)
- 0    = The contents of the ARF buffer are maintained after Bus Reset.
  - 1    = The contents of the ARF buffer are abandoned after Bus Reset.
- If the RxSelfID bit is set after bus reset, the MD8415 stores the SelfID packet in the ARF buffer. If this bit is set in this case, the data stored shortly before Bus Reset are abandoned in order to ensure the storage of the SelfID packet.

- Bit 29**            **ITFBusReset:** ITF BusReser mode bit (RW - Initial value: 0b)
- 0    = The contents of the ITF buffer are maintained after bus reset.
  - 1    = The contents of the ITF buffer are abandoned after bus reset.
- If an isochronous buffer is defined in the ITF (when the ITF/IRF bit is "0"), the data stored in the ITF buffer shortly before bus reset are abandoned after Bus Reset.

- Bit 30**            **DreqLow:** DREQ Low active bit (RW - Initial value: 0b)
- 0    = The DREQ terminal set at High Active.
  - 1    = The DREQ terminal set at Low Active.
- In the MD8415, a logic of active status is selected at the DREQ terminal.

- Bit 31**            **DackHigh:** DACK High active bit (RW - Initial value: 0b)
- 0    = The DACK terminal set at Low Active.
  - 1    = The DREQ terminal set at High Active.
- In the MD8415, a logic of active status is selected at the DACK terminal.

### 3-2-3 Node Identification Register

Address 08h

Initial value 3F3F FFC0h

7	6	5	4	3	2	1	0
BusNumber-L		NodeNumber					
15	14	13	12	11	10	9	8
BusNumber-H							
23	22	21	20	19	18	17	16
irmValid		irmNode					
31	30	29	28	27	26	25	24
IDValid	root	rootNode					

**Bit 5~0 NodeNumber:** Node Number field (R- Initial value: 00h)

This value is automatically set by the MD8415 after bus reset has been accomplished for the 6-bit node number defined in the IEEE1212 space. During transmission, this value is used in the source area of the IEEE 1394 packet format header. During reception, the node number of the receiving packet destination is checked. If the checked number coincides with this value, reception is forwarded. Otherwise, reception is rejected. Usually, at the end of the Self-Identification phase after the accomplishment of bus reset by the MD8415 itself, a request for NodeID readout is made toward the PHY and the obtained result is set in this field. This field is used for read only. If the MD8415 fails in NodeID readout from the PHY, "3Fh" is set up.

**Bit 15~6 BusNumber:** Bus Number field (R - Initial value: 3FFh)

This value can be used to read out a 10-bit bus number defined in the IEEE 1212 space. During transmission, this value is used in the source area of the header area belonging to the IEEE 1394 packet format. During reception, the bus number of the receiving packet destination is checked. If the checked number coincides with this value, reception is forwarded. Otherwise, reception is rejected. This field is used for read only. If writing is needed, it must be done via the CSR Host Access register.

**Bit 21~16 irmNode:** Isochronous Resource Mgr field (R- Initial value: 3Fh)

As a result of SelfID packet analysis, the node number of IRM is stored in this field. If the result of SelfID packet analysis is found to be incorrect, Value 3Fh is set up.

**Bit 23 irmValid:** Isochronous Resource Mgr Valid bit (R- Initial value: 0b)

This bit indicates that the values have been stored in the fields of rootNode and IrmNode as a result of SelfID packet analysis. This bit is cleared after VBusReset. It is set again when values are stored in the fields of rootNode and irmNode.

**Bit 24~29 rootNode:** Root Node ID field (R- Initial value: 3Fh)

As a result of SelfID packet analysis, the node number of root is stored in this field. If the result of SelfID packet analysis is found to be incorrect, Value 3Fh is set up.

**Bit 30 root:** root bit (R- Initial value: 00b)

- 0 = The connected Phy is not a root.
- 1 = The connected Phy is a root.

---

When Bus Reset occurs, this bit is usually set by examining whether the automatically connected PHY is a root or not. At that time, this setting can be confirmed by examining whether a PhyRegRcvd interrupt has occurred.

**Bit 31**      **IDValid:** ID Valid bit (RW- Initial value: 0b)

- 0    = A packet is received only if it has been addressed so that the BusNumber value is '3FFH' and the NodeNumber value is '3Fh'. All packets other than this condition are rejected.
- 1    = A packet is received only if it has been addressed in the IEEE 1212 address space that has been set by the above-mentioned register and under the conditions specified below. Broadcast packets are also received.
  - Both BusNumber and NodeNumber coincide with the register set values.
  - BusNumber coincides with the register set value and the NodeNumber value is '3Fh'.
  - The BusNumber value is '3FFH' and the NodeNumber value coincides with the register set value.
  - Both BusNumber and NodeNumber are '3Fh'.

### 3-2-4 Reset Register

Address        0Ch  
 Initial value   0000 0000h

7	6	5	4	3	2	1	0
		ResetDMA	ResetLink	ResetTx	ResetITF/IRF	ResetARF	ResetATF
15	14	13	12	11	10	9	8
23	22	21	20	19	18	17	16
31	30	29	28	27	26	25	24

**Bit 0            ResetATF:** Reset ATF bit (RW - Initial value: 0b)

0    = Normal condition.

1    = Buffer area initialize for asynchronous transmission.

Only the buffer area is restored to its initial state for asynchronous transmission. At that time, all data located in that area are lost. All status flags for this buffer are also restored to their initial state.

When '1' is set, this bit is automatically set at '0' upon completion of internal initialize operation.

**Bit 1            ResetARF:** Reset ARF bit (RW - Initial value: 0b)

0    = Normal condition.

1    = Buffer area initialize for asynchronous reception.

Only the buffer area is restored to its initial state for asynchronous reception. At that time, all data located in that area are lost. All status flags for this buffer are also restored to their initial state.

When '1' is set, this bit is automatically set at '0' upon completion of internal initialize operation.

**Bit 2            ResetITF/IRF:** Reset ITF/IRF bit (RW - Initial value: 0b)

0    = Normal condition.

1    = Buffer area initialize for isochronous transmission/reception.

Only the buffer is restored to its initial state for isochronous transmission/reception. A buffer being initialized by the ITF/IRF bit is selected. At that time, all data located in that area are lost. All status flags for this buffer are also restored to their initial state.

When '1' is set, this bit is automatically set at '0' upon completion of internal initialize operation.

**Bit 3            ResetTx:** Reset Transmitter bit (RW - Initial value: 0b)

0    = Normal condition.

1    = Transmitter reset.

The transmitter is reset to secure a transmission-enabled condition. If a packet is being transmitted, such a transmission operation is aborted. When '1' is set, this bit is automatically set at '0' upon completion of internal initialize operation.

**Bit 4            ResetLink:** Reset Link Core bit (RW - Initial value: 0b)

0    = Normal condition.

1 = Link Core reset.

The Link Core is reset and all operation is aborted. When '1' is set, this bit is automatically set at '0' upon completion of internal initialize operation.

In this case, however, the following registers are not initialized.

- CycleTimer Register
- BusTimer Register

The register bits to be initialized are as specified below.

- RetryTimeMax
- RetryCount
- AckStatus
- ATAck

There is a precautionary note for this bit setting. Prior to this setting, it is necessary to confirm that no transmission is presently carried out. If this bit is set in the middle of transmission, successive operation will become unstable.

**Bit 5 ResetDMA:** Reset DMA bit (RW - Initial value: 0b)

- 0 = Normal condition.
- 1 = DMA control reset.

DMA control is reset to secure a DMA transfer-enabled condition. The DMA is required to complete this transfer in the quadlet unit. The DMA transfer pointer in the quadlet unit is cleared with this bit and the pointer is set in the head position in the quadlet unit. When '1' is set, this bit is automatically set at '0' upon completion of internal initialize operation.

**3-2-5 Packet Control Register**

Address 10h  
Initial value 0000 1060h

7	6	5	4	3	2	1	0
SelfIDOut	SelfIDErr	RxSelfID	EnSnoop	PhyIDOut			
15	14	13	12	11	10	9	8
	RetryEn	ackAddressErr	WritePending	OnePacketRcv	BusyCtrl		
23	22	21	20	19	18	17	16
RetryInterval							
31	30	29	28	27	26	25	24

**Bit 3 PhyIDOut:** PhyID Isochronous Bus Output Enable bit (RW - Initial value: 0b)

- 0 = No PhyID output from Isochronous Bus.
- 1 = PhyID output generated from Isochronous Bus.

This bit is valid only if the MASTER# terminal is at "L" (Isochronous Bus in Master mode), in order to set up a condition whether a PhyID output should be generated from the Isochronous Bus.

**Bit 4 EnSnoop:** EnableSnoop bit (RW - Initial value: 0b)

- 0 = Usually, only the packet mapped to this node address is received.

1 = A snoop mode is assumed.

This bit is used to set up a snoop mode that receives any packet input entered from the PHY. When an asynchronous packet is received, no Ack code is returned. All packets received in this mode are output from the IDATA bus. No transmission is possible in this mode.

**Bit 5**            **RxSelfID:** Recive Self ID bit (RW - Initial value: 1b)

0 = No insertion of SelfID packet in the buffer.

1 = Insertion of SelfID packet in the buffer.

This bit is used to set up the condition whether a SelfID packet to be received in the SelfID phase after bus reset should be put in the buffer area for reception Async. Irrespective of this setting, SelfID packet analysis is carried out.

**Bit 6**            **SelfIDErr:** Recive Self ID Packet Error bit (R - Initial value: 1b)

0 = Normal accomplishment of transaction is indicated after SelfID packet analysis.

1 = Abnormal accomplishment of transaction is indicated after SelfID packet analysis.

Irrespective of RxSelfID bit setting, a SelfID packet to be received in the SelfID phase after bus reset is analyzed to examine the presence of reversal bits and the continuity of NodeID. If there is any problem found in the contents of the packet, this bit is set up. This value is held until the next BusReset occurs.

**Bit 7**            **SelfIDOut:** SelfID Packet Isochronous Bus Output Enable bit (RW - Initial value: 0b)

0 = No SelfID packet output generated from Isochronous Bus.

1 = SelfID packet output generated from Isochronous Bus.

This bit is valid only if the MASTER# terminal is at "L" (Isochronous Bus in Master mode), in order to set up a condition that a SelfID packet output should be generated from the Isochronous Bus.

**Bit 10~8**        **BusyCtrl:** Busy Control field (RW - Initial value: 000b)

000 = A busy acknowledge signal is returned according to the dual phase retry protocol only if there is no vacant area for one packet to be received in the internal Async reception buffer.

001 = An acknowledge signal is returned in the BusyA status only if there is no vacant area for one packet to be received in the internal Async reception buffer.

010 = An acknowledge signal is returned in the BusyB status only if there is no vacant area for one packet to be received in the internal Async reception buffer.

011 = An acknowledge signal is returned in the BusyX status only if there is no vacant area for one packet to be received in the internal Async reception buffer.

100 = A busy acknowledge signal is returned according to the dual phase retry protocol for all the received packets, irrespective of the presence of a vacant area for one packet to be received in the internal Async reception buffer.

101 = An acknowledge signal is returned in the BusyA status for all the received packets, irrespective of the presence of a vacant area for one packet to be received in the internal Async reception buffer.

110 = An acknowledge signal is returned in the BusyB status for all the received packets, irrespective of the presence of a vacant area for one packet to be received in the internal Async reception buffer.

111 = An acknowledge signal is returned in the BusyX status for all the received packets, irrespective of the presence of a vacant area for one packet to be received in the internal Async reception buffer.

When the MD8415 node stays in the inbound mode, an acknowledge signal of the busy status may be returned to the packet that has been sent from the outbound node to the MD8415. In this case, the contents of the status are set up in this register.

**Bit 11**            **A OnePacketRcv:** One-Packet-Receive-Only bit (RW - Initial value: 0b)

0 = Busy control for the internal Async reception buffer (ARF) is performed according to the BusyCtrl bit.

- 1 = Busy control for the internal Async reception buffer (ARF) is performed irrespective of the BusyCtrl bit. When one packet is stored in the ARF, Ack\_Busy is always returned to any packet entering successively.

**Bit 12 WritePending:** A Write Request Ack-Pending bit (RW - Initial value: 1b)

- 0 = An Ack code to the Write Request packet. In normal reception, an Ack-Complete signal is returned.  
 1 = An Ack code to the Write Request packet. In normal reception, an Ack-Pending signal is returned.

When a Write Request packet is normally received, the Ack code usually returns an ACK-Complete signal. In case of unusual reception such as lack of buffer capacity or the like, ACK-Busy is returned. If this bit is set at "1" and normal reception is accomplished, the Ack code returns ACK-Pending. In other words, Split Transaction of Write Request is executed. Upon completion of Write Request processing, the host is required to start transmission of the Write Response packet. The table below shows the types of Ack codes to be returned for the respective packets. The Ack codes indicated by circles (o) are those to be returned.

Packet	WritePending = '0b'			WritePending = '1b'		
	ack_complete	ack_pending	ack_busy	ack_complete	ack_pending	ack_busy
Write Request	o	-	o	-	o	o
Read Request	-	o	o	-	o	o
Write Response	o	-	o	o	-	o
Read Response	o	-	o	o	-	o
Lock Request	-	o	o	-	o	o
Lock Response	o	-	o	o	-	o

**Bit 13 ackAddressErr:** Ack\_Address\_error bit (RW - Initial value: 0b)

- 0 = An Ack code to the Write Request packet. When ack\_Type\_Error is set in the Internal CSR (core) block, ack\_Type\_Error signal is returned.  
 1 = An Ack code to the Write Request packet. When ack\_Type\_Error is set in the Internal CSR (core) block, ack\_address\_error signal is returned.

The 1394a newly specifies that ack\_address\_error be used in the case of register access to a non-defined address. This bit is used to cope with such a condition.

**Bit 14 RetryEn:** Retry Function Enable bit (RW - Initial value: 1b)

- 0 = No automatic retry operation in the MD8415.  
 1 = Automatic retry operation to be conducted according to the value set in the BUSY\_TIMEOUT register of the CSR.

This is a register intended to decide whether retry is performed by the MD8415 itself.

**Bit 16~23 RetryInterval:** Retry Interval Count field (RW - Initial value: 1h)

A retry interval is designated at the time of dual/single retry. The value 125μS corresponds to one Isochronous Cycle unit. When the set value is "0", retry is immediately performed.

### 3-2-6 Diagnostic Status and Control Register

Address 14h

Initial value 0000 0000h

It is possible to know various status information in this register.

7	6	5	4	3	2	1	0
			OnePacketProc Status				OnePacketProc End
15	14	13	12	11	10	9	8
Retry TimeMax		AckStatus		ATAck			
23	22	21	20	19	18	17	16
ATiRetry TimeMax		ATiAckStatus		ATiAck			
31	30	29	28	27	26	25	24
for_ackmiss	ack_cancel	LinkReady	EEPROM_ CRC		ATiBusy	ITBusy	ATBusy

**Bit 0** **OnePacketProcEnd:** One Packet End bit (RW - Initial value: 0b)

0 = Normal condition.

1 = End of One Packet processing is requested.

When the OnePacketRcv bit has been set for the Packet Control register and one packet is received, such a condition is assumed that no other packet (to an external CSR space) is received. When this bit is set up, the condition is changed to receive another packet (to an external CSR space). When set once, automatic clearing takes place thereafter.

**Bit 4** **OnePacketProcStatus:** One Packet Rcv Status bit (R - Initial value: 0b)

0 = Normal condition.

1 = A condition of OnePacket processing is assumed and reception is processed as Busy.

When the OnePacketRcv bit has been set for the Packet Control register and one packet is received, such a condition is assumed that no other packet (to an external CSR space) is received. This condition is maintained until the OnePacketProcEnd bit is set up. When the OnePacketProcEnd bit is set up, the OnePacketProcStatus bit is cleared.

**Bit 11~8** **ATAck:** AT Ack field (R - Initial value: 0000b)

0000 = No Ack

0001 = ack\_complete

0010 = ack\_pending

0011 = Reserved

0100 = ack\_busy\_X

0101 = ack\_busy\_A

0110 = ack\_busy\_B

0111

~1100= Reserved

1101 = ack\_data\_error

1110 = ack\_type\_error

1111 = ack\_address\_error

During transmission, an Acknowledge (Ack code) is returned from the destination node to the packet that has been transmitted from the ATF, and the contents of this Ack code are reflected in this register. The Acknowledge (Ack code) is not reflected for a packet that has been transmitted in the Internal CSR (Core, UserSpace).

The reflection timing is defined when a busy flag is negated, which is used to show the processing of a packet in the transmitting asynchronous buffer. This value is held until the said busy flag is negated for the next packet transmission.

**Bit 13~12      AckStatus:** Ack Status field (R - Initial value: 00b)

- 00 = Normal reception
- 01 = Parity error
- 10 = Packet lost (No Acknowledge packet arrived in the specified time.)
- 11 = Reserved

These bits are used to show the status of an Acknowledge packet returned from the destination node to the asynchronous packet that has been transmitted from the ATF. The status of the Acknowledge packet is not reflected for a packet that has been transmitted in the Internal CSR (Core, UserSpace).

**Bit 15          RetryTimeMax:** Retry Time Max bit (R - Initial value: 0b)

- 0 = Normal condition.
- 1 = Maximum number of retries.

This bit is used to indicate the maximum number of retry counts in the ATRetry register and the condition of retry set in time, for a transmission transaction attempted from the ATF at the beginning of a retry phase. At that time, an ATAck bit is used to identify whether the retry phase has been finished or whether a busy condition still remains. This value is held until the next packet transmission is conducted.

No reflection is performed for a retry to a packet that has been transmitted in the Internal CSR (Core, UserSpace).

**Bit 19~16      ATiAck:** AT Ack(ATi) field (R - Initial value: 0000b)

- 0000 = No Ack
- 0001 = ack\_complete
- 0010 = ack\_pending
- 0011 = Reserved
- 0100 = ack\_busy\_X
- 0101 = ack\_busy\_A
- 0110 = ack\_busy\_B
- 0111
- ~1100 = Reserved
- 1101 = ack\_data\_error
- 1110 = ack\_type\_error
- 1111 = ack\_address\_error

During transmission, an Acknowledge (Ack code) is returned from the destination node to the packet that has been transmitted from the ATFi, and the contents of this Ack code are reflected in this register.

The reflection timing is defined when a busy flag is negated, which is used to show the processing of a packet in the transmitting asynchronous buffer. This value is held until the said busy flag is negated for the next packet transmission.

**Bit 21~20      ATiAckStatus:** ATiAck Status field (R - Initial value: 00b)

- 00 = Normal reception
- 01 = Parity error
- 10 = Packet lost (No Acknowledge packet arrived in the specified time.)
- 11 = Reserved.

These bits are used to show the status of an Acknowledge packet returned from the destination node to the asynchronous packet that has been transmitted from the ATFi.

**Bit 23**            **ATiRetryTimeMax:** ATiRetryTimeMax bit (R - Initial value: 0b)

- 0    = Normal condition.
- 1    = Maximum number of retries.

This bit is used to indicate the maximum number of retry counts in the ATRetry register and the condition of retry set in time, for a transmission transaction attempted from the ATFi at the beginning of a retry phase. At that time, an ATiAck bit is used to identify whether the retry phase has been finished or whether a busy condition still remains. This value is held until the next packet transmission is conducted.

**Bit 24**            **ATBusy:** AT Busy bit (R - Initial value: 0b)

- 0    = ATGo issuing is enabled.
- 1    = ATGo issuing is disabled. It indicates present packet processing for ATGo that has been issued shortly before.

This bit is asserted upon the issuing of ATGo during asynchronous transmission attempted from the ATF. It is negated when the returned Acknowledge has been set in the ATAck register. While this bit is asserted, the host cannot issue the next ATGo. Even though it should be issued, it would be disregarded. When a packet transmission is put into retry operation, this bit is not negated till the completion of this retry. No reflection is made in regard to the status of transmission in the Internal CSR (Core, UserSpace).

**Bit 25**            **ITBusy:** IT Busy bit (R - Initial value: 0b)

- 0    = ITGo issuing is enabled.
- 1    = ITGo issuing is disabled. It indicates present packet processing for ITGo that has been issued shortly before.

This bit is asserted upon the issuing of ITGo during isochronous transmission. It is negated when transmission is completed. While this bit is asserted, the host cannot issue the next ITGo. Even though it should be issued, it would be disregarded.

**Bit 26**            **ATiBusy:** AT Busy(ATi) bit (R - Initial value: 0b)

- 0    = Internal transaction is idle.
- 1    = Internal transaction is busy for packet processing.

This bit is asserted upon the issuing of Go during asynchronous transmission attempted from the ATFi. It is negated when the returned Acknowledge has been set in the ATiAck register. When a packet transmission is put into retry operation, this bit is not negated till the completion of this retry.

**Bit 28**            **EEPROM\_CRC:** EEPROM CRC condition status bit (R - Initial value: 0b)

After hardware reset (RESET#) has been asserted, the MD8415 reads out data from the EEPROM and performs CRC16 checking. It is necessary to store the data image of the Configuration ROM from the EEPROM head. In this case, CRC calculation is carried out within the range of data length in the CRC\_Length field located in the heading 1 Quadlet. If it is found to be identical with the rom\_CRC value as a result of comparison, this bit is set up. The setting timing for this bit is defined when EEPROM\_Ready is set in the Interrupt register.

**Bit 29**            **LinkReady:** Link Ready Status bit (R - Initial value: 0b)

After hardware reset (RESET#) has been asserted, the MD8415 performs initial operation such as data read-out from the EEPROM. Upon completion of this initial operation, this bit is set up.

**Bit 30**            **ack\_cancel:** ack\_cancel bit (RW - Initial value: 1b)

- The cancel condition of 0 = Ack transmission must be Grant from Phy (current condition)
- The cancel condition of 1 = Ack transmission must be BusReset and SubActionGap only.

---

The condition to cancel the Ack transmission is designated.

**Bit 31**            **for\_ackmiss:** Ack Miss bit (RW - Initial value: 1b)

0    = AckSent=0x0 is stored when Ack transmission is canceled.

1    = The received packet is flushed when Ack transmission is canceled.

When Ack transmission is canceled, processing is selected for the receiving data for which the said Ack must be transmitted.

### 3-2-7 Phy Control Register

Address        18h  
 Initial value   0000 0000h

This register is used to gain access to a register in the PHY chip. When reading out a certain register, its register address is set at the RegAddr register and the RdReg bit is made to be active. When the RdReg bit is active, a read request is started for the register at that address. The contents of the register at that address from the PHY are entered in the RegData register. When making a Write request to the PHY register, the data in the RegData register are written in the PHY by activating the WrReg bit as a trigger signal at the PHY address set in the RegAddr register.

Note: In the case of 8-bit Write access, this access must be forwarded in the order of 1Bh, 1Ah, 19h, and 18h.

7	6	5	4	3	2	1	0
RegData							
15	14	13	12	11	10	9	8
	RegRcvd	RdReg	WrReg	RegAddr			
23	22	21	20	19	18	17	16
31	30	29	28	27	26	25	24

**Bit 7~0        RegData:** Register Data field (RW - Initial value: 00h)

The data to be transferred to the PHY are stored with a Write request. The data transferred from the PHY are also stored with a Read request. When reading out the contents of this register, the contents of the RegData register are the values read from the PHY upon the presentation of a Read request. In other words, the values written from the host cannot be read out directly from this register. When these values must be read out, it is necessary to send a Read request to the PHY.

**Bit 11~8      RegAddr:** Register Address field (RW - Initial value: 00h)

These bits are used to set up the address value of the register in the PHY being accessed. However, as a value to be read out of this field after the placement of the Write request (WrReg=1), a written address value is read out. As a value to be read out of this field after the placement of the Read request (RdReg=1), an actually read-out address value from the PHY is read out.

**Bit 12        WrReg:** Write Register bit (RW - Initial value: 0b)

0    = Normal condition.  
 1    = Issuing a Write request.  
 A Write request is issued toward the PHY register.

**Bit 13        RdReg:** Read Register bit (RW - Initial value: 0b)

0    = Normal condition  
 1    = Issuing a Read request.  
 A Read request is issued toward the PHY register.

**Bit 14        RegRcvd:** Register Data Received bit (R - Initial value: 0b)

0    = Normal condition.

1 = After the issuing of a Read request, this bit indicates that the data from the PHY have been stored in the RegData register.

After the issuing of a Read request toward the PHY register, "1" is set when the data from the PHY have been stored in the RegData register. When this register is once read out since then, it is cleared to '0'.

### 3-2-8 ATRetry Register

Address 1Ch  
Initial value 0000h

There is an automatic retry function to be effective when a Busy Acknowledge is returned from a destination node of the MD8415 during asynchronous packet transmission. Setting for the number of retries and retry time is made by a register in the CSR space (BUSY\_TIMEOUT) incorporated in the MD8415. Once a retry phase has been assumed, no busy flag of the ATGo register is negated until an Acknowledge other than Busy is returned from the destination node, or the preset number of retries and retry time are exceeded. Unless the Busy flag is negated, the next packet cannot be transmitted. It is, however, possible to use a RetryStop bit to finish the retry phase forcibly.

If the retry phase is not finished within the preset time period, a retry timeout status flag is set in the ATAck register in order to finish the retry phase presently executed by the MD8415. Since then, the packet data in the ATF buffer are flushed. If the RetryEn bit is not set in the Packet Control register, the MD8415 does not perform automatic retry operation. In this case, the packet data are flushed for a Busy Acknowledge. If an error Ack code is returned in the middle of retry phase, retry operation is interrupted at that moment and this operation is finished by flushing data in the buffer and setting a flag (AckErr).

This register sends out information about the condition during this retry.

7	6	5	4	3	2	1	0
				retryCount			
15	14	13	12	11	10	9	8
cycleCount							
23	22	21	20	19	18	17	16
cycleCount							
31	30	29	28	27	26	25	24
RetryStop		RetryX	RetryA/B	secondCount			cycleCount

**Bit 3~0** **RetryCount:** Retry Count field (R - Initial value: 0h)

During single-retry operation, the MD8415 indicates the present number of retries.

**Bit 24~12** **CycleCount:** Retry Cyclet Limit field (R - Initial value: 0000h)

During dual-retry operation, the MD8415 indicates the present retry lapse time and the Cycle unit.

**Bit 27~25** **SecondCount:** Retry Second Limit field (R - Initial value: 0000h)

During dual-retry operation, the MD8415 indicates the present retry lapse time and the Second unit.

**Bit 28** **RetryA/B:** Retyr X active bit (R - Initial value: 0b)

This bit is set when the MD8415 is in the middle of dual-phase retry.

**Bit 29** **RetryX:** Retyr X active bit (R - Initial value: 0b)

This bit is set when the MD8415 is in the middle of single-phase retry.

**Bit 31      RetryStop:** Retry Forced-End bit (RW - Initial value: 0b)

This bit is used to accomplish forced end of retry operation while the MD8415 is in the middle of automatic retry phase and the limit value is not attained yet. When this bit is set at '1', it is automatically cleared after the completion of the retry phase.

- 0: Normal condition.
- 1: Forced end.

**3-2-9 Cycle Timer Register**

Address      24h  
 Initial value    0000 0000h

The present CycleTimer value is indicated. As described below, this register is split into three areas. When the node using the MD8415 is for the Cycle Master, this register value is inserted at the time of CycleStart packet transmission. Otherwise, the CycleTimer value in the received CycleStart packet is loaded in this register to update the CycleTimer.

The host cannot load any CycleTimer value via this register. It is permitted to perform readout only. When loading a value, it is necessary to do it via the CSR Host Access register.

7	6	5	4	3	2	1	0
CycleOffset							
15	14	13	12	11	10	9	8
CycleCount				CycleOffset			
23	22	21	20	19	18	17	16
CycleCount							
31	30	29	28	27	26	25	24
CycleSeconds							CycleCount

**Bit 11~0      CycleOffset:** Cycle Offset field (R - Initial value: 00h)

This area is counted up with a clock of 24.576MHz. It works with Modulo3072.

**Bit 15~24      CycleCount:** Cycle Count field (R - Initial value: 00h)

This area is counted up when the CycleField register is carried. It counts isochronous cycles. It works with Modulo8000.

**Bit 24~16      CycleCount:** Cycle Count field (R - Initial value: 00h)

This area is counted up when the CycleField register is carried. It counts isochronous cycles. It works with Modulo8000.

**Bit 3~25      CycleSeconds:** Cycle Seconds field (R - Initial value: 00h)

This area is counted up when the CycleCount register is carried. It counts seconds. It works with Modulo128.

### 3-2-10 Isochronous Transmit Configuration Register

Address 28h

Initial value 0000 0000h

When IsoDataMode is in operation and its transmission mode is AUTO, this register is used to set up how the MD8415 handles this packet. The data value for one packet being transferred in each isochronous cycle is set in this register. In the isochronous AUTO mode, however, the number of channels permitted for transmission is limited to only one.

7	6	5	4	3	2	1	0
StopSync							SyncEn
15	14	13	12	11	10	9	8
Sync				StartSync			
23	22	21	20	19	18	17	16
						Speed	
31	30	29	28	27	26	25	24
Tag		Channel					

**Bit 0 SyncEn:** Sync Enable bit (RW - Initial value: 0b)

- 0 = Irrespective of the ITS SYNC signal, the contents of the SYNC register are reflected any time in the Sync area of the packet header.
- 1 = The Sync field value is determined in the packet header being transmitted upon the placement of the ITS SYNC signal. The contents of StartSync are entered in the transmission packet shortly after ITS SYNC Active. Since then, the contents of StopSync are entered in the transmission packet shortly after ITS SYNC Non-Active. When ITS SYNC is disabled, the following conditions are assumed according to the setting value of ITZERO:  
ITZERO=0: No transmission.  
ITZERO=1: Sync field is entered any time.

**Bit 7~4 StopSync:** Stop Sync field (RW - Initial value: 00h)

When the ITS SYNC terminal is turned to be non-active with SyncEn=1b, this value is written in the Sync field of the packet received shortly after.

**Bit 11~8 StartSync:** Start Sync field (RW - Initial value: 00h)

When the ITS SYNC terminal is turned to be active with SyncEn=1b, this value is written in the Sync field of the packet sent shortly after.

**Bit 15~12 Sync:** Sync field (RW - Initial value: 00h)

With SyncEn=1b, this value is written in a packet other than the specified StartSync packet/StopSync packet at the ITS SYNC terminal.

**Bit 17~16 Speed:** Speed field (RW - Initial value: 0b)

- 0 = Transmission at 100Mbps
- 1 = Transmission at 200Mbps
- 2 = Transmission at 400Mbps

In the transmission mode, the transfer speed toward the cable is set up.

**Bit 29~24**      **Channel:** Channel field (RW - Initial value: 00h)

The channel of a transmitting isochronous packet is designated. For transmission, the channel number set up here is inserted in the packet header. The setting range is from 0 to 63.

**Bit 31~30**      **Tag:** Tag field (RW - Initial value: 00h)

The isochronous packet tag is designated. The setting range is from 0 to 3.

### 3-2-11 Isochronous Transmit Configuration Register

Address      2Ch

Initial value    0000 0000h

When IsoMode is in operation and its transmission mode is AUTO, this register is used to set up how the MD8415 handles this packet. The data value for one packet being transferred in each isochronous cycle is set in this register. In the isochronous AUTO mode, however, the number of channels permitted for transmission is limited to only one.

7	6	5	4	3	2	1	0
ITLength							
15	14	13	12	11	10	9	8
ITLength							
23	22	21	20	19	18	17	16
31	30	29	28	27	26	25	24

**Bit 15~0**      **ITLength:** Isochronous Transmit Length field (RW - Initial value: 0h)

The length of a transmitting packet (in byte unit) is set in this register. For transmission, this value is used in the header block. It is, however, necessary to set it up in the Quadlet unit (a multiple of 4). Bits 0 and 1 are fixed at "0".

**3-2-12 Isochronous Bus Receive Configuration Register 1, 2, 3, 4**

Address 30h(1), 34h(2), 38h(3), 3Ch(4)

Initial value 0000 0000h

When IsoDataMode is normal in case other than snoop reception of an isochronous packet, this register group is used to set up which packet is to be received by the MD8415, how the packet should be received, and how it is output to the IsoBus side. The number of channels is 4, which can be received in case other than the snoop reception mode. When IsoDataMode is AUTO, only one channel is available for the packet that has been set with Isochronous Bus Receive Configuration 1. Reception is effective in the AND condition of Tag and Channel.

7	6	5	4	3	2	1	0
StopSync				CTEn		IsoRxEn	SyncEn
15	14	13	12	11	10	9	8
				StartSync			
23	22	21	20	19	18	17	16
31	30	29	28	27	26	25	24
Tag		Channel					

**Bit 0 SyncEn:** Sync Enable bit (RW - Initial value: 0b)

- 0 = Packet reception is effected irrespective of the contents of StartSync and StopSync.
- 1 = Packet reception is controlled with the preset Tag, Channel, and StartSync and StopSync.

**Bit 1 IsoRxEn:** Isochronous Receive Enable bit (RW - Initial value: 0b)

- 0 = Isochronous reception disabled.
- 1 = Isochronous reception enabled.

This setting is made to determine whether isochronous reception should be enabled or not.

**Bit 2~3 CTEn:** CycleTimer Enable field (RW - Initial value: 00b)

- 00 = Reception start function by Cycle Timer value is disabled.
- 01 = Reception start function by Cycle Timer value is enabled under the condition of Isochronous Bus Receive CT Configuration register 1.
- 10 = Reception start function by Cycle Timer value is enabled under the condition of Isochronous Bus Receive CT Configuration register 2.
- 11 = Reserved

This setting is made to determine whether the reception start function by Cycle Timer value should be enabled or not.

**Bit 7~4 StopSync:** Stop Sync field (RW - Initial value: 00h)

With SyncEn=1, reception control is effected according to this field and the Sync field value of a packet. Once packet reception is started with StartSync, this packet reception can be stopped when the Sync field of the preset Tag and Channel packet has coincided with the value in this register.

If receiving operation is stopped with the IsoRxEn bit prior to receiving this value, reception is not carried out any more until the next StartSync comes.

- 
- Bit 11~8**      **StartSync:** Start Sync field (RW - Initial value: 00h)  
With SyncEn=1, reception control is effected according to this field and the Sync field value of a packet. Packet reception is started when the Sync field of the preset Tag and Channel packet has coincided with the value in this register.
- Bit 31~30**      **Tag:** Tag field (RW - Initial value: 0h)  
A receiving isochronous tag is set up. The setting range is from 0 to 3.
- Bit 29~24**      **Channel:** Channel field (RW - Initial value: 00h)  
A receiving isochronous channel is set up. The setting range is from 0 to 63.

### 3-2-13 Isochronous Host Buffer Receive Configuration Register

Address 40h

Initial value 0000 0000h

This register is available only if Isochronous Bus is stored in the IRF buffer in the Master mode. This is a register intended to set up a condition to receive an isochronous packet in the IRF buffer. It is enabled only if ITF/IRF has been set in the IRF. Only one channel is available for the preset packet.

7	6	5	4	3	2	1	0
StopSync				CTEn		IsoRxEn	SyncEn
15	14	13	12	11	10	9	8
				StartSync			
23	22	21	20	19	18	17	16
31	30	29	28	27	26	25	24
Tag		Channel					

**Bit 0**      **SyncEn:** Sync Enable bit (RW - Initial value: 0b)

- 0 = Packet reception is effected irrespective of the contents of StartSync and StopSync.
- 1 = Packet reception is controlled with the preset Tag, Channel, and StartSync and StopSync.

**Bit 1**      **IsoRxEn:** Isochronous Receive Enable bit (RW - Initial value: 0b)

- 0 = Isochronous reception disabled.
- 1 = Isochronous reception enabled.

This setting is made to determine whether isochronous reception should be enabled or not.

**Bit 2~3**      **CTEn:** CycleTimer Enable field (RW - Initial value: 00b)

- 00 = Reception start function by Cycle Timer value is disabled.
- 01 = Reception start function by Cycle Timer value is enabled under the condition of Isochronous Bus Receive CT Configuration register 1.
- 10 = Reception start function by Cycle Timer value is enabled under the condition of Isochronous Bus Receive CT Configuration register 2.
- 11 = Reserved

This setting is made to determine whether the reception start function by Cycle Timer value should be enabled or not.

**Bit 7~4**      **StopSync:** Stop Sync field (RW - Initial value: 00h)

With SyncEn=1, reception control is effected according to this field and the Sync field value of a packet. Once packet reception is started with StartSync, this packet reception can be stopped when the Sync field of the preset Tag and Channel packet has coincided with the value in this register.

If receiving operation is stopped with the IsoRxEn bit prior to receiving this value, reception is not carried out any more until the next StartSync comes.

**Bit 11~8**      **StartSync:** Start Sync field (RW - Initial value: 00h)

With SyncEn=1, reception control is effected according to this field and the Sync field value of a packet. Packet reception is started when the Sync field of the preset Tag and Channel packet has coincided with the value in this register.

**Bit 31~30 Tag:** Tag field (RW - Initial value: 0h)  
A receiving isochronous tag is set up. The setting range is from 0 to 3.

**Bit 29~24 Channel:** Channel field (RW - Initial value: 00h)  
A receiving isochronous channel is set up. The setting range is from 0 to 63.

**3-2-14 Isochronous Bus Receive CT Configuration Register 1, 2**

Address 44h(1), 48h(2)

Initial value 0000 0000h

This register has a function of starting reception from the time point when its value has coincided with the Cycle Timer value preset at the time of isochronous packet reception. The Cycle Timer value to be preset at that time is set in this register. In regard to the function of starting reception from the time point upon coincidence with the Cycle Timer value, this function comes in two types only.

7	6	5	4	3	2	1	0
StopSync				CTEn		IsoRxEn	SyncEn
15	14	13	12	11	10	9	8
				StartSync			
23	22	21	20	19	18	17	16
31	30	29	28	27	26	25	24
Tag		Channel					

**Bit 12~24 CycleCount:** CycleCount Event field (RW - Initial value: 0b)  
CycleCount is set up for the Cycle Timer going to start reception.

**Bit 25~31 CycleSeconds:** CycleSeconds Event field (RW - Initial value: 0b)  
CycleSeconds is set up for the Cycle Timer going to start reception.

### 3-2-15 Isochronous Receive Error Register

Address 4Ch

Initial value 0000 0000h

This register indicates the contents of an error arising during the reception of an isochronous packet. The value is held until the next error occurs.

- 1) Contents of an error (CRC error, Length error, IRF full).
- 2) Cycle No. and Channel at the time of an error.

Since contents of the report are not accumulated, the report display register (Isochronous Receive Error register) provides only information of the latest error. When an error occurs and the Isochronous Receive Error register is read out after the occurrence of an error in another cycle, the register shows the last error only and the contents of the first error are destroyed.

7	6	5	4	3	2	1	0
Err_Tag		Err_Channel					
15	14	13	12	11	10	9	8
Err_CycleCount				Err_Code		Err_Speed	
23	22	21	20	19	18	17	16
Err_CycleCount							
31	30	29	28	27	26	25	24
Err_CycleSeconds							Err_CycleCount

**Bit 0~5** **Err\_Channel:** Error Channel field (R - Initial value: 0b)

The Channel, where an error has occurred, is indicated during reception.

**Bit 7~6** **Err\_Tag:** Error Tag field (R - Initial value: 0b)

The Tag, where an error has occurred, is indicated during reception.

**Bit 8~9** **Err\_Speed:** Error Speed field (R - Initial value: 0b)

Transfer speed is indicated for the packet, where an error has occurred during reception.

**Bit 10~11** **ErrCode:** Error Code field (R - Initial value: 0b)

- 0 = CYCLE START PACKET Lost
- 1 = Data CRC error
- 2 = Data length error
- 3 = IRF Full (Enabled only for IRF setting in the IsoBus Slave or IsoBus Master mode).

**Bit 12~24** **Err\_CycleCount:** Error CycleCount field (R - Initial value: 0b)

The CycleCount value of the Cycle Timer is shown at the time point when an error has occurred during reception.

**Bit 25~31** **Err\_CycleSeconds:** Error CycleSeconds field (R - Initial value: 0b)

The CycleSeconds value of the Cycle Timer is shown at the time point when an error has occurred during reception.

00

**3-2-16 Asynchronous Stream Receive Configuration Register**

Address 50h

Initial value 0000h

This is a register intended to set up a condition to receive an Asynchronous Stream packet in the ARF buffer. Two types of setting can be made.

7	6	5	4	3	2	1	0
ASTag2		ASChannel2					
15	14	13	12	11	10	9	8
ASEn2							
23	22	21	20	19	18	17	16
ASTag1		ASChannel1					
31	30	29	28	27	26	25	24
ASEn1							

**Bit 5~0** **ASChannel2:** Asynchronous Stream Receive Channel2 field (RW - Initial value: 00h)

These bits are used to set up a receiving Asynchronous Stream channel. The setting range is 0 to 63.

**Bit 7~6** **ASTag2:** Asynchronous Stream Receive Tag2 field (RW - Initial value: 00h)

These bits are used to set up a receiving Asynchronous Stream Tag. The setting range is 0 to 3.

**Bit 15** **ASEn2:** Asynchronous Stream Receive Enable2 bit (RW - Initial value: 0b)

0 = Asynchronous Stream Reception disabled.

1 = Asynchronous Stream Reception enabled.

This bit is used to set up whether the setting value of Asynchronous Stream Receive Channel (ASChannel2) for Setting 2 should be enabled or not.

**Bit 21~16** **ASChannel1:** Asynchronous Stream Receive Channel1 field (RW - Initial value: 00h)

These bits are used to set up a receiving Asynchronous Stream channel. The setting range is 0 to 63.

**Bit 23~22** **ASTag1:** Asynchronous Stream Receive Tag1 field (RW - Initial value: 00h)

These bits are used to set up a receiving Asynchronous Stream Tag. The setting range is 0 to 3.

**Bit 31** **ASEn1:** Asynchronous Stream Receive Enable1 bit (RW - Initial value: 0b)

0 = Asynchronous Stream Reception disabled.

1 = Asynchronous Stream Reception enabled.

This bit is used to set up whether the setting value of Asynchronous Stream Receive Channel (ASChannel 1) for Setting 1 should be enabled or not.

**3-2-17 ATF Data Register**

Address 54h

Initial value 0000 0000h

This is the transmission data writing register for asynchronous packets. Data must be written in the Quadlet unit at all times. The data once written at that time can be checked by reading out this register.

7	6	5	4	3	2	1	0
ATFData							
15	14	13	12	11	10	9	8
ATFData							
23	22	21	20	19	18	17	16
ATFData							
31	30	29	28	27	26	25	24
ATFData							

**3-2-18 ARF Data Register**

Address 58h

Initial value 0000 0000h

This is the reception data readout register for asynchronous packets. Data must be read out in the Quadlet unit at all times.

7	6	5	4	3	2	1	0
ARFData							
15	14	13	12	11	10	9	8
ARFData							
23	22	21	20	19	18	17	16
ARFData							
31	30	29	28	27	26	25	24
ARFData							

**3-2-19 ITF Data Register**

Address 5Ch

Initial value 0000 0000h

This is the transmission data writing register for isochronous packets. Data must be written in the Quadlet unit at all times. The data once written at that time can be checked by reading out this register.

7	6	5	4	3	2	1	0
ITFData							
15	14	13	12	11	10	9	8
ITFData							
23	22	21	20	19	18	17	16
ITFData							
31	30	29	28	27	26	25	24
ITFData							

**3-2-20 IRF Data Register**

Address 60h

Initial value 0000 0000h

This is the reception data readout register for isochronous packets. Data must be read out in the Quadlet unit at all times.

7	6	5	4	3	2	1	0
IRFData							
15	14	13	12	11	10	9	8
IRFData							
23	22	21	20	19	18	17	16
IRFData							
31	30	29	28	27	26	25	24
IRFData							

### 3-2-21 Buffer Status and Control Register

Address 64h

Initial value 0055 0000h

This register is used for the status control of the internal buffer for asynchronous transmission/reception and the flush control of the buffer for asynchronous reception.

7	6	5	4	3	2	1	0
IRFSize							
15	14	13	12	11	10	9	8
IRFSize							
23	22	21	20	19	18	17	16
	IRFEmpty	ITFFull	ITFEmpty		ARFEmpty	ATFFull	ATFEmpty
31	30	29	28	27	26	25	24
DackEn	DreqEn	SelectDreq		DMA_ Little			Little

**Bit 0~15**      **IRFSize:** IRFSize bit (R - Initial value: 1b)

When ITF/IRF is set at IRF, size of the data stored in the IRF is indicated in the Byte unit. However, since it is shown in the Quadlet unit, the lower two bits are fixed at "0". "0" for the ITF.

These bits must be latched at the falling point of the RD# signal.

**Bit 16**      **ATFEmpty:** ATF Empty bit (R - Initial value: 1b)

0 = Indicates that the buffer is not vacant.

1 = Indicates that the buffer is vacant.

This bit is used to indicate that the Asynchronous transmission buffer to be accessed from the ATF Data register is vacant.

**Bit 17**      **ATFFull:** ATF Full bit (R - Initial value: 0b)

0 = Indicates that the buffer is not full.

1 = Indicates that the buffer is full.

This bit is used to indicate that the Asynchronous transmission buffer to be accessed from the ATF Data register is full.

**Bit 18**      **ARFEmpty:** ARF Empty bit (R - Initial value: 1b)

0 = Indicates that the buffer is not vacant.

1 = Indicates that the buffer is vacant.

This bit is used to indicate that the Asynchronous reception buffer to be accessed from the ARF Data register is vacant.

**Bit 20**      **ITFEmpty:** ITF Empty bit (R - Initial value: 1b)

0 = Indicates that the buffer is not vacant.

1 = Indicates that the buffer is vacant.

This bit is used to indicate that the Isochronous transmission buffer to be accessed from the ITF Data register is vacant.

- Bit 21**      **ITFFull:** ITF Full bit (R - Initial value: 0b)  
0    = Indicates that the buffer is not full.  
1    = Indicates that the buffer is full.  
This bit is used to indicate that the Isochronous transmission buffer to be accessed from the ITF Data register is full.
- Bit 22**      **IRFEmpty:** IRF Empty bit (R - Initial value: 1b)  
0    = Indicates that the buffer is not vacant.  
1    = Indicates that the buffer is vacant.  
This bit is used to indicate that the Isochronous reception buffer to be accessed from the IRF Data register is vacant.
- Bit 24**      **Little:** Host Little-endian bit (RW - Initial value: 0b)  
0    = For the transfer from the MPU via the register, the data in the ATF, ARF, ITF, IRF buffer are handled as Big-endian.  
1    = For the transfer from the MPU via the register, the data in the ATF, ARF, ITF, IRF buffer are handled as little-endian.
- Bit 27**      **DMA\_Little:** DMA Little-endian bit (RW - Initial value: 0b)  
0    = For DMA transfer, the data in the ATF, ARF, ITF, IRF buffer are handled as Big-endian.  
1    = For DMA transfer, the data in the ATF, ARF, ITF, IRF buffer are handled as little-endian.
- Bit 28~29**    **SelectDreq:** Select Dreq bit (RW - Initial value: 0b)  
00b = To be reflected on the DREQ signal as the status equivalent to the ATFFull bit of the buffer to be accessed by the ATF Data register.  
01b = To be reflected on the DREQ signal as the status equivalent to the ARFEmpty bit of the buffer to be accessed by the ARF Data register.  
10b = To be reflected on the DREQ signal as the status equivalent to the ITFFull bit of the buffer to be accessed by the ITF Data register.  
11b = To be reflected on the DREQ signal as the status equivalent to the IRFEmpty bit of the buffer to be accessed by the IRF Data register.
- Bit 30**      **DreqEn:** Dreq Enable bit (RW - Initial value: 0b)  
0    = The DREQ signal is kept non-active at any time.  
1    = The DREQ signal is kept active as the status of contents chosen in the DREQ signal with SelectDreq.  
This bit is used to make the DREQ signal enabled.  
This register is used to select which condition of a buffer in the chip should be reflected on the DREQ signal.
- Bit 31**      **DackEn:** Dack Enable bit (RW - Initial value: 0b)  
0    = Even though the Dack# signal is active, the MD8415 disregards this signal.  
1    = The Dack# signal is enabled.

### 3-2-22 Asynchronous Bus Control Register 1

Address 68h

Initial value 0000 0000h

Note: To be accessed in the order of 6Ah and 68h for a 16-bit Write access, and in the order of 6Bh, 6Ah, 69h, 68h for an 8-bit Write access.

7	6	5	4	3	2	1	0
ABTxLen							
15	14	13	12	11	10	9	8
				ABTxLen			
23	22	21	20	19	18	17	16
ABTCount							
31	30	29	28	27	26	25	24
ABTxEn			ABTxStatus		ABTxCount		

**Bit 10~0** **ABTxLen:** Asynchronous Bus Transmitt Length field (RW - Initial value: 0b)

The number of write actions from the Asynchronous But to the ATF is set here. Writing is conducted in the Byte unit. The lower two bits are fixed at "0".

**Bit 26~16** **ABTxCount:** Asynchronous Bus Transmit Count field (R - Initial value: 0b)

Initialized at "0" when ABTxEn is asserted. Since then, ATX# is asserted and counting up is advanced each time writing is accomplished in the ATF. One writing advances the counting up of 2 bytes. It is done in the Byte unit. The lower one bit is fixed at "0".

**Bit 29~28** **ABTxStatus:** Asynchronous Bus Transmit Status field (R - Initial value: 0b)

When ABTxEn is cleared, the condition is as shown below.

00 = Cleared because ABTxEn=0 is written in this register.

01 = Cleared because ABTxLen is attained.

10 = Cleared because of ResetATF.

11 = Cleared because ATEOP is asserted.

**Bit 31** **ABTxEn:** Asynchronous Bus Transmitt Enable bit (RW - Initial value: 0b)

This bit is set up when ATF writing is modified to the Asynchronous Bus side. When this bit is set up, the ATEN# signal of the Asynchronous Bus is asserted. This bit is cleared if the ATEOP# signal has been asserted or when the number of ATF writing actions has attained ABTxLen after the assertion of the ATEN# signal.

### 3-2-23 Asynchronous Bus Control Register 2

Address 6Ch

Initial value 0000 0000h

Note: To be accessed in the order of 6Eh and 6Ch for a 16-bit Write access, and in the order of 6Fh, 6Eh, 6Dh, 6Ch for an 8-bit Write access.

7	6	5	4	3	2	1	0	
ABRxLen								
15	14	13	12	11	10	9	8	
				ABRxLen				
23	22	21	20	19	18	17	16	
ABRxCount								
31	30	29	28	27	26	25	24	
ABRxEn			ABRxStatus					ABRxCount

**Bit 0~10 ABRxLen:** Asynchronous Bus Receive Length field (RW - Initial value: 0b)

The number of outputs from the ARF to the Asynchronous Bus is set here. Writing is conducted in the Byte unit. The lower two bits are fixed at "0".

**Bit 26~16 ABRxCount:** Asynchronous Bus Receive Count field (RW - Initial value: 0b)

Initialized at "0" when ABRxEn is asserted. Since then, ARX# is asserted and counting up is advanced each time readout is accomplished in the ARF. One readout advances the counting up of 2 bytes. It is done in the Byte unit. However, the lower one bit is fixed at "0".

**Bit 29~28 ABRxStatus:** Asynchronous Bus Receive Status field (R - Initial value: 0b)

When ABRxEn is cleared, the condition is as shown below.

- 00 = Cleared because ABRxEn=0 is written in this register.
- 01 = Cleared because ABRxLen is attained.
- 10 = Cleared because of ResetARF.
- 11 = Cleared because of ARFEmpty.

**Bit 31 ABRxEn:** Asynchronous Bus Receive Enable bit (RW - Initial value: 0b)

This bit is set up when an output is transferred from the ARF buffer to the Asynchronous Bus side. When ABRxEn is set up and more than one packet is stored in the ARF, the ARPKT# signal is asserted. This bit is cleared when the number of ARF readout actions has attained ABRxLen after the assertion of the ARPKT# signal.

### 3-2-24 Interrupt Register

Address 70h

Initial value 0000 0000h

When this register is read out by the host, it is possible to know the various interrupt factors of the MD8415. When all bits are "1" in this register, such a condition indicates that an interrupt factor has arisen. The interrupt factor can be cleared by writing "1" in this register.

7	6	5	4	3	2	1	0
CycleSeconds	CycleStart	CycleDone	CycleLost	EEPROM_Ready			extended
15	14	13	12	11	10	9	8
PhyInt	BusReset	BusReady	PhyRegRcvd	AckErr	TCodeErr	HdrErr	DatErr
23	22	21	20	19	18	17	16
ATxEnd	ATxCSREnd	ARxCSREnd	ARxEnd	ITxEnd	ITFNoTx	IRxErr	IRxEnd
31	30	29	28	27	26	25	24
ABTxEnd	ABRxEnd	SelfDPktRcvd	ARFRej	BTInt	GTInt	STO	RetryLimit

**Bit 0** **extended:** extended Interrupt bit (RW - Initial value: 0b)

This bit indicates that an interrupt factor has been set up in the Extended Interrupt register. This bit is not cleared even though "1" is written in. This setting is maintained until the Extended Interrupt register is completely cleared.

**Bit 3** **EEPROM\_Ready:** EEPROM Ready bit (RW - Initial value: 0b)

After hardware reset by asserting the RESET# signal, the MD8415 performs CRC check for the EEPROM. This bit indicates that the said check has been finished. In this case, setting is made at "1".

**Bit 4** **CycleLost:** Cycle Lost bit (RW - Initial value: 0b)

When the node of the MD8415 is not of the CycleMaster, the next CycleStart packet may not be received in 250μsec of the Cycle Timer after the reception of a CycleStart packet to update the internal Cycle Timer. In such a case, setting is made at "1".

**Bit 5** **CycleDone:** Cycle Done bit (RW - Initial value: 0b)

When an isochronous cycle is finished, setting is made at "1".

**Bit 6** **CycleStart:** Cycle Start bit (RW- Initial value: 0b)

When a new isochronous cycle is started, setting is made at "1".

**Bit 7** **CycleSeconds:** Cycle Seconds bit (RW - Initial value: 0b)

When one second is counted by the Cycle Timer possessed by the MD8415, setting is made at "1".

**Bit 8** **DatErr:** Data(Payload) Error bit (RW - Initial value: 0b)

During the reception of an Asynchronous packet or an Asynchronous Stream packet, this bit is set when the Payload block includes a CRC error or if there is no coincidence of Length.

**Bit 9** **HdrErr:** Header Error bit (RW - Initial value: 0b)

During packet reception, setting is made at "1" if the received packet has a header with an error.

**Bit 10**            **TCodeErr:** TCode Error bit (RW - Initial value: 0b)

During packet transmission, setting is made at "1" if a code not supported by the MD8415 has been set in the Tcode area of the packet header.

**Bit 11**            **AckErr:** Ack Error bit (RW - Initial value: 0b)

For an Asynchronous packet sent from the ATF, an Acknowledge packet is returned from the destination node. If this Acknowledge packet cannot be received normally, setting is made at "1".

**Bit 12**            **PhyRegRcvd:** Phy Register Received bit (RW - Initial value: 0b)

When the data from the PHY are stored in RegData after the issuing of a read request to the PHY register, setting is made at "1".

**Bit 13**            **BusReady:** Bus Ready bit (RW - Initial value: 0b)

This bit indicates that the SelfID phase is finished shortly after Bus Reset and the Bus has assumed an idle condition.

**Bit 14**            **BusReset:** Bus Reset bit (RW - Initial value: 0b)

When the PHY stays in the Bus Reset mode, setting is made at "1".

**Bit 15**            **PhyInt:** Phy Interrupt bit (RW - Initial value: 0b)

When an interrupt factor comes from the PHY that is connected to the MD8415, setting is made at "1".

**Bit 16**            **IRxEnd:** Isochronous Receive End bit (RW - Initial value: 0b)

When isochronous reception is performed by the MD8415 and data are stored in the IRF buffer, setting is made at "1". For IsochronousBus reception, this bit is not set up. This bit is effective only if data have been stored in the IRF.

**Bit 17**            **IRxErr:** Isochronous Receive Error bit (RW - Initial value: 0b)

This bit is set when an error occurs during isochronous reception by the MD8415. In this case, this setting is also made at the time of IRF reception or IsochronousBus reception.

**Bit 18**            **ITNoTx:** Isochronous No Transmit bit (RW - Initial value: 0b)

When the MD8415 performs isochronous transmission in the AUTO mode, setting is made at "1" under the following conditions:

- Transmission is not carried out after the reception or transmission of a CycleStart packet with ITZERO=0.
- A packet of Length=0 is transmitted after the reception or transmission of a CycleStart packet with ITZERO=1.

**Bit 19**            **ITxEnd:** Isochronous Transmitt End bit (RW - Initial value: 0b)

When isochronous transmission is performed by the MD8415 from the ITF buffer, setting is made at "1" upon the completion of transmitting operation. This setting is not made for the transmission attempted from the IsochronousBus.

**Bit 20**            **ARxEnd:** Asynchronous Receive End bit (RW - Initial value: 0b)

When asynchronous reception is performed by the MD8415 and data have been stored in the ARF buffer, setting is made at "1".

- Bit 21**      **ARxCSREnd:** Asynchronous Receive End (Internal CSR Area) bit (RW - Initial value: 0b)  
When asynchronous reception is performed by the MD8415 and the space belongs to the Internal CSR area, setting is made at "1" if data have been stored in the ARFi buffer. However, this setting is not made if the attribute of the associated CSR register does not cause any interrupt [Internal CSR (NoInt)].
- Bit 22**      **ATxCSREnd:** Asynchronous Transmitt End (Internal CSR Area) bit (RW - Initial value: 0b)  
When the MD8415 makes asynchronous transmission (response) during the processing of the Internal CSR space, setting is made at "1" upon the completion of transmitting operation and the reception of an Ack code from destination. If retry operation is performed, setting is also made at "1" at the end of the said retry phase. Setting is also made at "1" even when AckErr is set on the way. However, this setting is not made if the attribute of the associated CSR register does not cause any interrupt [Internal CSR (NoInt)].
- Bit 23**      **ATxEnd:** Asynchronous Transmitt End bit (RW - Initial value: 0b)  
When the MD8415 performs transmitting operation from the ATF, setting is made at "1" upon the completion of this transmitting operation and the reception of an Ack code from the destination. If retry operation is performed, setting is also made at "1" at the end of the said retry phase. Setting is also made at "1" even when AckErr is set on the way.
- Bit 24**      **RetryLimit:** Retry Limit Detect bit (RW - Initial value: 0b)  
When a retry phase is being accomplished by the MD8415 by the use of a transmission transaction from the ATF, setting is made at "1" if this retry phase cannot be accomplished even though the preset limit value has been exceeded.
- Bit 25**      **STO:** Split Time Out Detect bit (RW - Initial value: 0b)  
When STStart is issued and the count value is delivered to the Split Time Out register, setting is made at "1".
- Bit 26**      **GTInt:** General Timer Interrupt bit (RW - Initial value: 0b)  
This setting is made when the Interrupt condition defined by General Timer is assumed.
- Bit 27**      **BTInt:** Bus Configuration Timer Interrupt bit (RW - Initial value: 0b)  
This setting is made when the Interrupt condition defined by Bus ConfigurationTimer is assumed.
- Bit 28**      **ARFRej:** ARF Reject bit (RW - Initial value: 0b)  
When asynchronous reception is performed by the MD8415 and data cannot be stored due to lack of capacity in the ARF buffer, setting is made at "1".
- Bit 29**      **SelfIDPktRcvd:** SelfID Packet Received bit (RW - Initial value: 0b)  
When the MD8415 receives a SelfID packet, setting is made at "1". This setting is made when a SelfID packet returned at the time of SelfID packet reception is received after ordinary Bus Reset. This bit is valid only if the RxSelfID bit of the Packet Control register is set.
- Bit 30**      **ABRxEnd:** Asynchronous Bus Receive End bit (RW - Initial value: 0b)  
When AsynchronousBus transfer (readout from the ARF) is finished by the MD8415, setting is made at "1".
- Bit 31**      **ABTxEnd:** Asynchronous Bus Transmitt End bit (RW - Initial value: 0b)  
When AsynchronousBus transfer (writing in the ARF) is finished by the MD8415, setting is made at "1".

Status	When FIFO is full in the middle of reception and normal reception is impossible to carry out.	When a CRC error is contained in the data, or a Length error.	When normal reception is possible.
ARxEnd Interrupt	0	0	1
ARFRej Interrupt	1	0	0
DatErr Interrupt	0	1	0
ARF status	Same as the condition before reception. Nothing is written.	Same as the condition before reception. Nothing is written.	All stored.

### 3-2-25 Extended Interrupt Register

Address 74h

Initial value 0000 0000h

This register is used to indicate an interrupt factor that is reflected on the extended bit of the Interrupt register. When all bits are "1" in this register, such a condition indicates that an interrupt factor has arisen. The interrupt factor can be cleared by writing "1" in this register.

7	6	5	4	3	2	1	0
					CmdReset	LinkOff	cmstr
15	14	13	12	11	10	9	8
23	22	21	20	19	18	17	16
31	30	29	28	27	26	25	24

**Bit 0** **cmstr:** cmstr bit (RW - Initial value: 0b)

This bit is set up when dependent.cmstr is set or cleared in the STATE register of the Internal CSR (core).

**Bit 1** **LinkOff:** Link Off bit (RW - Initial value: 0b)

This bit is set up when dependent.linkoff is set or cleared in the STATE register of the Internal CSR (core).

**Bit 2** **CmdReset:** Command Reset bit (RW - Initial value: 0b)

When a packet addressed to the reset area in the CSR space, setting is made at "1".

In the case of CSRAtrib4==HostCSR, however, the received packet is stored in the ARF. At that time, the interrupt factor does not arise and the internal CSR register is not initialized.

In the case of CSRAtrib4!=HostCSR, on the other hand, the received packet is not stored in the ARF, and the interrupt factor arises. The internal CSR register is also initialized.

**3-2-26 Interrupt Mask Register-1**

Address 78h

Initial value 0000 0000h

If each interrupt factor in the Interrupt register is not wanted to be reflected in the INT-1# signal, this register is used to mask it. The layout of this register is the same as that for the Interrupt register. Each bit can be masked by the setting of "1". When this register is used, combination of interrupt factors is enabled for both INT-1# and INT-2# signals. It must be noted that the interrupt factors of the Extended Interrupt register cannot be masked. Everything is masked by the use of an extended bit.

**3-2-27 Interrupt Mask Register-2**

Address 7Ch

Initial value 0000 0000h

If each interrupt factor in the Interrupt register is not wanted to be reflected in the INT-2# signal, this register is used to mask it. The layout of this register is the same as that for the Interrupt register. Each bit can be masked by the setting of "1". When this register is used, combination of interrupt factors is enabled for both INT-1# and INT-2# signals. It must be noted that the interrupt factors of the Extended Interrupt register cannot be masked. Everything is masked by the use of an extended bit.

### 3-2-28 Split TimeOut Register

Address 80h

Initial value 0320 0000h

When a transaction becomes a split transaction, this register is used as a setting register for the control of the arising timeout. When the STStart bit is set at "1", counting is started. Counting stops when this bit is set at "0". If the counted value coincides with the preset value in this register in the middle of counting, Split Time Out is noticed to the host by the use of STO Interrupt. This counter performs counting up at the intervals of 125 $\mu$ sec. Therefore, the maximum setting value is not longer than 8 seconds.

7	6	5	4	3	2	1	0
SplitTime							
15	14	13	12	11	10	9	8
SplitTime							
23	22	21	20	19	18	17	16
SplitTimeLimit							
31	30	29	28	27	26	25	24
SplitTimeLimit							

### 3-2-29 Split Timer Start Register

Address 84h

Initial value 0000 0000h

7	6	5	4	3	2	1	0
							STStart
15	14	13	12	11	10	9	8
23	22	21	20	19	18	17	16
31	30	29	28	27	26	25	24

**Bit 0**      **STStart:** Split Timer Start bit (RW - Initial value: 0b)

STStart=0: The cleared condition of the Split Timer is maintained.

STStart=1: The counter of the Split Timer is started. The initial condition is assumed at "0".

When an STO Interrupt arises, the STStart bit is automatically cleared.

### 3-2-30 Bus Configuration Timer Register

Address 88h

Initial value 0000 0000h

After the occurrence of Bus Reset, an Interrupt factor arises when the value of each Timer 1 to 4 is attained. This is a BCTint bit of the Interrupt register. This bit is set when the value at any of Timer 1 to 4 coincides with the preset value.

7	6	5	4	3	2	1	0
BCCTime							
15	14	13	12	11	10	9	8
BCCTime							
23	22	21	20	19	18	17	16
31	30	29	28	27	26	25	24
IntTimer4	IntTimer3	IntTimer2	IntTimer1				

**Bit 31 IntTimer4:** Interrupt Timer4 bit (RW - Initial value: 0b)

This bit is set when the value at BCTimer4 coincides with the preset value. This bit is cleared when Bus Reset occurs. When this bit is set, it can also be cleared if "1" is written.

**Bit 30 IntTimer3:** Interrupt Timer3 bit (RW - Initial value: 0b)

This bit is set when the value at BCTimer3 coincides with the preset value. This bit is cleared when Bus Reset occurs. When this bit is set, it can also be cleared if "1" is written.

**Bit 29 IntTimer2:** Interrupt Timer2 bit (RW - Initial value: 0b)

This bit is set when the value at BCTimer2 coincides with the preset value. This bit is cleared when Bus Reset occurs. When this bit is set, it can also be cleared if "1" is written.

**Bit 28 IntTimer1:** Interrupt Timer1 bit (RW - Initial value: 0b)

This bit is set when the value at BCTimer1 coincides with the preset value. This bit is cleared when Bus Reset occurs. When this bit is set, it can also be cleared if "1" is written.

**Bit 15~0 BCCTime:** Bus Configuration Current Time bit (R - Initial value: 0b)

The lapse time after the occurrence of Bus Reset is indicated. This bit is cleared when Bus Reset occurs. The unit is 125μsec.

**3-2-31 Bus Configuration Timer Register 1**

Address 8Ch  
 Initial value 0000 0000h

An Interrupt factor arises when the value at each Timer 1 to 4 is attained after the occurrence of Bus Reset. If "0" is set in the value, no Interrupt occurs.

7	6	5	4	3	2	1	0
BCTimer2							
15	14	13	12	11	10	9	8
BCTimer2							
23	22	21	20	19	18	17	16
BCTimer1							
31	30	29	28	27	26	25	24
BCTimer1							

**Bit 32~16** BCTimer1: Bus Configuration Timer1 field (RW - Initial value: 0b)

**Bit 15~0** BCTimer2: Bus Configuration Timer2 field (RW - Initial value: 0b)

**3-2-32 Bus Configuration Timer Register 2**

Address 90h  
 Initial value 0000 0000h

7	6	5	4	3	2	1	0
BCTimer4							
15	14	13	12	11	10	9	8
BCTimer4							
23	22	21	20	19	18	17	16
BCTimer3							
31	30	29	28	27	26	25	24
BCTimer3							

**Bit 32~16** BCTimer3: Bus Configuration Timer3 field (RW - Initial value: 0b)

**Bit 15~0** BCTimer4: Bus Configuration Timer4 field (RW - Initial value: 0b)

### 3-2-33 General Timer Register

Address 94h

Initial value 0000 0000h

Two kinds of Interrupt, one shot and periodic, can be chosen with the general-purpose timer. When the preset value is attained, an Interrupt factor arises. This is a GTInt bit of the Interrupt register.

Note: To be accessed in the order of 96h and 94h for a 16-bit Write access, and in the order of 97h, 96h, 95h, 94h for an 8-bit Write access.

7	6	5	4	3	2	1	0
GTimer							
15	14	13	12	11	10	9	8
GTimer							
23	22	21	20	19	18	17	16
GCTime							
31	30	29	28	27	26	25	24
GTEn	GTOneShot	GCTime					

**Bit 31**      **GTEn:** General Timer Enable bit (RW - Initial value: 0b)

This is a start/stop control bit for the periodic timer. It is also used as a start trigger bit in OneShot mode. When the GTEn bit is cleared, the Timer value is also cleared.

**Bit 30**      **GTOneShot:** General Timer OneShot Mode bit (RW - Initial value: 0b)

When this bit is set, the General Timer operates in the OneShot mode. In this case, the Timer begins to operate when GTEn is set. At the preset time, a GTInt Interrupt takes place. At that time the GTEn bit is cleared by the MD8415.

If it is not set up, a periodic Interrupt occurs. If GTEn is not set, a GTInt Interrupt occurs at the preset time and the Timer is cleared to start counting up again. To stop Interrupt operation, it is necessary to clear the GTEn bit.

**Bit 29~16**      **GCTime:** General Current Time field (R - Initial value: 0b)

A present value at the General Timer is indicated. The unit is 125 $\mu$ S.

**Bit 13~0**      **GTimer:** General Time field (RW - Initial value: 0b)

The time value needed to make Interrupt is set up. "0" results in no operation. The unit is 125 $\mu$ S. The setting time is 125 $\mu$ S to 2.048S.

### 3-2-34 EEPROM Control Register

Address 98h

Initial value 0000h

This register is used for data write/read operation in the EEPROM.

7	6	5	4	3	2	1	0
EEPROM_ ProcEnd	EEPROM_ ProcErr					EEPROM_ ReadEn	EEPROM_ WriteEn
15	14	13	12	11	10	9	8
23	22	21	20	19	18	17	16
EEPROM_adrs						0	0
31	30	29	28	27	26	25	24
EEPROM_adrs							

- Bit 0**      **EEPROM\_WriteEn:** EEPROM Write Enable bit (RW - Initial value: 0b)  
This bit is set up when a write request is issued. It is automatically cleared when writing is finished.  
Prior to setting this bit, it is necessary to store the writing data in the EEPROM Data register.  
When writing data in the EEPROM, it is necessary to take an action of not accepting a read request from 1394. In other words, writing is possible in the EEPROM when the LPSON bit of the Control register is "0" or IDValid is "0" in the Node Identification register.
- Bit 1**      **EEPROM\_ReadEn:** EEPROM Read Enable bit (RW - Initial value: 0b)  
This bit is set up when a read request is issued. It is automatically cleared when reading is finished.  
Prior to setting this bit, it is necessary to store the writing data address in the EEPROM\_adrs.
- Bit 6**      **EEPROM\_ProcErr:** EEPROM Process Error bit (RW - Initial value: 0b)  
This bit is set up if writing or reading cannot be finished normally due to an access error such as gaining access to an absent address. This bit is automatically cleared once reading is started.
- Bit 7**      **EEPROM\_ProcEnd:** EEPROM Process End bit (RW - Initial value: 0b)  
This bit is set up when a read or write request is assumed and the MD8415 has completed the processing toward the EEPROM. This bit is automatically cleared once reading is started.
- Bit 16~31**      **EEPROM\_adrs:** EEPROM Write address field (RW - Initial value: 0b)  
However, actually effective bits are 18 to 31. Therefore, the only access in the Quadlet unit is possible.

**3-2-35 EEPROM Data Register**

Address 9Ch

Initial value 0000 0000h

This register is used for data write operation in the EEPROM.

7	6	5	4	3	2	1	0
EEPROM_data							
15	14	13	12	11	10	9	8
EEPROM_data							
23	22	21	20	19	18	17	16
EEPROM_data							
31	30	29	28	27	26	25	24
EEPROM_data							

**Bit 31~0**      **EEPROM\_data:** EEPROM Write data field (RW - Initial value: 0b)

**3-2-36 GPIO Register**

Address        A0h  
 Initial value    0000 0000h

It is possible to make setting for each pin to determine whether the status of each GPIO pin should be arranged for input or output.

7	6	5	4	3	2	1	0
						GPIOCTL(1:0)	
15	14	13	12	11	10	9	8
23	22	21	20	19	18	17	16
						GPIO(1:0)	
31	30	29	28	27	26	25	24

**Bit 1~0        GPIOCTL(1:0):** GPIO control field (RW - Initial value: 0b)  
 GPIOCTL(3:0)=0: Each GPIO pin is set for input.  
 GPIOCTL(3:0)=1: Each GPIO pin is set for output.

**Bit 17~16     GPIO(1:0):** GPIO data field (RW - Initial value: 0b)  
 When each GPIO pin is set for input with GPIOCTL, reading is enabled and data for each GPIO pin can be read out. When it is set for output, writing only is enabled. Output is generated from each GPIO pin at "High" level when GPIOCTL is "1" and at "Low" level when GPIOCTL is "0".

### 3-2-37 ATGo Register

Address        A4h  
Initial value    0000 0000h

For asynchronous packet transmission, "1" is set in the ATGo register located in this register, in order to give the MD8415 information about packet transmission. The host sends out the transmitting packet to the internal buffer of the MD8415 and then makes setting for this ATGo register. Based on "1" in the ATGo register, the MD8415 starts packet transmission for the data presently existing in the buffer regarded as one packet. Since a Go command is issued, ATBusy is turned to be "1" for asynchronous packet transmission and the host can identify that the MD8415 has started transmitting operation. The timing for the next ATGo issuing is when ATBusy is turned "0" for asynchronous packet transmission. Even though ATGo should be issued while ATBusy is "1", the MD8415 will disregard such an action. Even when "0" is written in this register, there will be no interruption of transmission. If the tCode block of the heading 1-Quadlet data contains wrong data, the ATGo bit and the ATBusy bit are left to be enabled at all times, and therefore transmitting operation is suspended. In such a case, resetting is possible by writing "0" in this ATGo and setting up ResetATF and ResetTx.

7	6	5	4	3	2	1	0
							ATGo
15	14	13	12	11	10	9	8
23	22	21	20	19	18	17	16
31	30	29	28	27	26	25	24

**Bit 0**            **ATGo:** AT Go bit (RW - Initial value: 0b)

The start of asynchronous packet transmission is announced to the MD8415 by writing "1" in this register.

**3-2-38 ITGo Register**

Address        A8h  
 Initial value    0000 0000h

For isochronous packet transmission via the ITF, information about packet transmission is sent to the MD8415 by setting "1" in the ITGo register that is located in this register. The host sends out the transmitting packet to the internal buffer of the MD8415 and then makes setting for this ITGo register. Based on "1" in the ITGo register, the MD8415 starts packet transmission for the data presently existing in the buffer regarded as one packet. Since a Go command is issued, ITBusy is turned to be "1" for isochronous packet transmission and the host can identify that the MD8415 has started transmitting operation. The timing for the next ITGo issuing is when ITBusy is turned "0" for isochronous packet transmission. Even though ITGo should be issued while ITBusy is "1", the MD8415 will disregard such an action. Even when "0" is written in this register, this will be disregarded.

7	6	5	4	3	2	1	0
							ITGo
15	14	13	12	11	10	9	8
23	22	21	20	19	18	17	16
31	30	29	28	27	26	25	24

**Bit 0**                    **ITGo:** AT Go bit (RW - Initial value: 0b)

The start of isochronous packet transmission is announced to the MD8415 by writing "1" in this register.

### 3-2-39 BInternal CSR Access Register

Address      ACh  
Initial value    0000 0000h

This is a register used to gain access to the register in the CSR space incorporated in the MD8415. To gain access to the incorporated register in the CSR space, it is always necessary to access this register after making the setting of the CSRDataH field and the CSRDataL field.

Note: To be accessed in the order of AEh and ACh for a 16-bit Write access, and in the order of AFh, AEh, ADh, ACh for an 8-bit Write access.

7	6	5	4	3	2	1	0
CSRAdrOff							
15	14	13	12	11	10	9	8
CSRAdrOff							
23	22	21	20	19	18	17	16
					CSRCore	USBank	
31	30	29	28	27	26	25	24
CSRAccErr	CSRRcv	Busy	Block	Lock	Write	Read	

**Bit 15~0**      **CSRAdrOff:** CSR Address Offset field (RW - Initial value: 0b)

The address offset is set up for each CSR register. The lower two bits are fixed at "0".

**Bit 17~16**      **USBank:** CSR User Space Bank Index field (RW - Initial value: 0b)

The UserSpace bank is set up.

**Bit 18**      **CSRCore:** CSR Core space access bit (RW - Initial value: 0b)

This bit is used to indicate the condition that a request has been applied to the Core register of CSR.

**Bit 25**      **Write:** Write Access bit (RW - Initial value: 0b)

This bit is set up when sending a Write request to the CSR register. It is automatically cleared after the completion of writing.

**Bit 26**      **Read:** Read Access bit (RW - Initial value: 0b)

This bit is set up when sending a Read request to the CSR register. It is automatically cleared after the completion of reading.

**Bit 27**      **Lock:** Lock Access bit (RW - Initial value: 0b)

This bit is set up when sending a Lock request to the CSR register. Only compare\_swap is supported. This bit is automatically cleared after the completion of operation.

**Bit 28**      **Block:** Block Access bit (RW - Initial value: 0b)

This bit is set up when sending a Block request to the CSR register. In this case, only Length=8 consisting of CSRDataH and CSRDataL is supported.

**Bit 29**      **Busy:** Access Busy bit (R - Initial value: 0b)  
 This bit is set when an access to the CSR is preceded by another access from 1394, and the preceding access is being processed. This bit is held until the next access request (Read/Write/Lock) is sent to the internal CSR. It is cleared when the request is accomplished.

**Bit 31**      **CSRAccErr:** CSR Access Error bit (R - Initial value: 0b)  
 This bit is used to indicate the condition that a request has been applied to the CSR register.  
 CSRAccErr=0: Normal condition.  
 CSRAccErr=1: Set when Lock fails. Or, it is set with an access to an address not supported. It is cleared when readout begins.

**Bit 30**      **CSRRcvd:** CSR data Received bit (R - Initial value: 0b)  
 If a Read/Lock request is made to the CSR register, this bit is set when the CSR data are stored. It is cleared when readout begins.

**3-2-40 Internal CSR Access DataH Register**

Address      B0h  
 Initial value    0000 0000h

This register is used to store the data for gaining access to the register in the CSR space that is incorporated in the MD8415, or it is used to store the data that have been read out. These are the data for which the Block bit is not set up. If the Block bit is set, the upper 32 bits are the data.

7	6	5	4	3	2	1	0
CSRDataH							
15	14	13	12	11	10	9	8
CSRDataH							
23	22	21	20	19	18	17	16
CSRDataH							
31	30	29	28	27	26	25	24
CSRDataH							

**Bit 31~0**      **CSRDataH:** CSR data High field (RW - Initial value: 0h)  
 When a Write/Lock request is sent to the CSR register, the CSR data are stored. In the case of Lock, the "arg" value is stored.

**3-2-41 Internal CSR Access DataL Register**

Address B4h

Initial value 0000 0000h

This is a register used to gain Block or Lock access to the register in the CSR space that is incorporated in the MD8415.

7	6	5	4	3	2	1	0
CSRDataL							
15	14	13	12	11	10	9	8
CSRDataL							
23	22	21	20	19	18	17	16
CSRDataL							
31	30	29	28	27	26	25	24
CSRDataL							

**Bit 31~0**      **CSRDataL:** CSR Data Low value field (RW - Initial value: 0h)

Data of the CSR data value are stored when a Block request is sent to the CSR register, and when the lower 32-bit data or Lock request is made.

### 3-2-42 CSR Transaction Register 1

Address        B8h  
 Initial value    0000 0000h

7	6	5	4	3	2	1	0
CSRAdrsOff							
15	14	13	12	11	10	9	8
CSRAdrsOff							
23	22	21	20	19	18	17	16
sourceBusID		sourceNodeID					
31	30	29	28	27	26	25	24
sourceBusID							

- Bit 0~15        CSRAdrsOff:** CSR Address Offset field (R - Initial value: 0b)  
 The received Offset data of the internal CSR space are stored. The lower two bits are fixed at "0".
  
- Bit 21~16        sourceNodeID:** Source Node ID field (R - Initial value: 0b)  
 This is NodeID accessed in the internal CSR space.
  
- Bit 31~22        sourceBusID:** source Bus ID field (R - Initial value: 0b)  
 This is BusID accessed in the internal CSR space.

### 3-2-43 CSR Transaction Register 2

Address        BCh  
 Initial value    0000 0000h

Note: To be accessed in the order of BEh and BCh for a 16-bit Write access, and in the order of BFh, BEh, BDh, BCh for an 8-bit Write access.

7	6	5	4	3	2	1	0
AckCode				RespCode			
15	14	13	12	11	10	9	8
						spd	
23	22	21	20	19	18	17	16
			BusID	CSRCore		USBank	
31	30	29	28	27	26	25	24
RespGo	CompOK	Abort	Block	LockReq	WriteReq		

- Bit 3~0        RespCode:** Response Code field (RW - Initial value: 0h)

When a Response packet is sent by setting up the RespGo bit, this field is sent as a Response code. In this case, writing is accomplished in the specified internal register of the MD8415 when resp\_complete(0x0) is stored and the RespGo bit is set up. In cases other than resp\_complete(0x0), no writing is accomplished in the specified internal register of the MD8415. This field value is reflected on the Response code of the Response packet.

**Bit 7~4            AckCode:** Ack Code field (R - Initial value: 0b)

The Ack code returned at the time of reception is stored.

**Bit 9~8            spd:** spd field (R - Initial value: 0b)

The speed code returned at the time of reception is stored.

**Bit 17~16        USBank:** CSR User Space Bank Index field (R - Initial value: 0b)

This bit is used to indicate that the heading address of CSRAdrsOff is the Bank of UserSpace.

**Bit 18            CSRCore:** CSR Core Space Access bit (R - Initial value: 0b)

This bit is used to indicate that the heading address of CSRAdrsOff is the Core(0xFFFF F000 0000) of CSR.

**Bit 20            BusID:** BusID bit (RW - Initial value: 0b)

To use SourceBusID=0x3F when this bit is set, "1" is set in bit19 of the heading first Quadlet being stored in AFTi. Rewriting of this bit is possible only with host access.

**Bit 26            WriteReq:** Write Request bit (R - Initial value: 0b)

This bit is used to indicate that a Write request is sent to the CSR register. When the RespGo bit is set up, this bit is automatically cleared. When BusReset/CmdReset arises and a transaction is abandoned, this bit is also automatically cleared.

**Bit 27            LockReq:** Lock Request bit (R - Initial value: 0b)

This bit is used to indicate that a Lock request is sent to the CSR register. When the RespGo bit is set up, this bit is automatically cleared. When BusReset/CmdReset arises and a transaction is abandoned, this bit is also automatically cleared.

**Bit 28            Block:** Block Request bit (R - Initial value: 0b)

This bit is used to indicate that a Block request is sent to the CSR register. When the RespGo bit is set up, this bit is automatically cleared. In this case, and also in the case of a Lock request, ResponseDataH and ResponseDataL become valid. If no setting has been made, ResponseDataH only is valid.

When BusReset/CmdReset arises and a transaction is aborted, this bit is also automatically cleared.

**Bit 29            Abort:** Transaction Abort bit (R - Initial value: 0b)

This bit is set up when BusReset/CmdReset arises in the sequence of a transaction and this transaction is suspended and aborted. This bit is held until this transaction arises again. It is cleared to '0' if a request to the CSR is made.

**Bit 30            CompOK:** Compare OK bit (R - Initial value: 0b)

When a Lock request is sent to the CSR register, this bit is set if a coincidence is confirmed as a result of comparison.

**Bit 31            RespGo:** Response Packet Go bit (RW - Initial value: 0b)

When a request is sent to the CSR register under the condition of WritePending=1 to send out a Response packet, this Response Packet transmission is enabled by setting up a required value in the RespData field and

the ResponseData field, and setting this bit. Upon completion of transmission, the value set in ResponseData is written in the register of this address. Since then, this bit is automatically cleared.

In the state of WritePending=0, A complete Ack signal must have been returned. Therefore, no Response packet is sent out. In the case of RespCode=resp\_complete(0x0) having been set, however, the value set in the Response data is written in the register of this address. Since then, this bit is automatically cleared.

The next packet is not accepted regardless of the WritePending bit until RespGo is issued to complete transmission, and Ack\_Busy is returned.

When a Lock request is sent, not the value of ResponseData field but that of LockData field is stored in the internal register of the MD8415. The data to be stored in the Response packet are those available before modification. If RespCode is not resp\_complete at that time, a Lock Response packet of Length=0 is returned.

This bit is disregarded even though it has been set while the WriteReq and LockReq bits are not set.

### 3-2-44 CSR Transaction DataH Register

Address        C0h  
 Initial value    0000 0000h

7	6	5	4	3	2	1	0
ResponseDataH							
15	14	13	12	11	10	9	8
ResponseDataH							
23	22	21	20	19	18	17	16
ResponseDataH							
31	30	29	28	27	26	25	24
ResponseDataH							

**Bit 0~31        ResponseDataH:** Response Data High field (RW - Initial value: 0b)

When a packet is received, its Payload block is stored. When sending a Response packet by setting the RespGo bit, this field is sent as ResponseData. If the Block bit is not set, the data of this register only are valid. If it is set, the upper 32-bit data are valid.

When a Lock request is made, the arg\_value block is stored at first.

**3-2-45 CSR Transaction DataL Register**

Address C4h

Initial value 0000 0000h

7	6	5	4	3	2	1	0
ResponseDataL							
15	14	13	12	11	10	9	8
ResponseDataL							
23	22	21	20	19	18	17	16
ResponseDataL							
31	30	29	28	27	26	25	24
ResponseDataL							

**Bit 0~31**      **ResponseDataL:** Response Data Low field (RW - Initial value: 0b)

When the Block bit is set, the lower 32-bit data are stored. At the time of Lock request, the data\_value block is stored.

**3-2-46 CSR Config1 Register**

Address C8h

Initial value 00F0 C3CFh

7	6	5	4	3	2	1	0
CSRAttrib13		CSRAttrib14		CSRAttrib15		CSRAttrib16	
15	14	13	12	11	10	9	8
CSRAttrib9		CSRAttrib10		CSRAttrib11		CSRAttrib12	
23	22	21	20	19	18	17	16
CSRAttrib5		CSRAttrib6		CSRAttrib7		CSRAttrib8	
31	30	29	28	27	26	25	24
CSRAttrib1		CSRAttrib2		CSRAttrib3		CSRAttrib4	

**Bit 30~31 CSRAttrib1:** CSR Attribute CSR 0x0000 field (RW - Initial value: 0b)

- 00 : InternalCSR
- 01 : InternalCSR(NoInt)
- 10 : HostCSR
- 11 : Ack\_Type\_Error

**Bit 28~29 CSRAttrib2:** CSR Attribute CSR 0x0004 field (RW - Initial value: 0b)

- 00 : InternalCSR
- 01 : InternalCSR(NoInt)
- 10 : HostCSR
- 11 : Ack\_Type\_Error

**Bit 26~27 CSRAttrib3:** CSR Attribute CSR 0x0008 field (RW - Initial value: 0b)

- 00 : InternalCSR
- 01 : InternalCSR(NoInt)
- 10 : HostCSR
- 11 : Ack\_Type\_Error

**Bit 24~25 CSRAttrib4:** CSR Attribute CSR 0x000C field (RW - Initial value: 0b)

- 00 : InternalCSR
- 01 : InternalCSR(NoInt)
- 10 : HostCSR
- 11 : Ack\_Type\_Error

**Bit 22~23 CSRAttrib5:** CSR Attribute CSR 0x0010 field (RW - Initial value: 11b)

- 00 : reserved
- 01 : reserved
- 10 : HostCSR
- 11 : Ack\_Type\_Error

- 
- Bit 20~21**      **CSRAttrib6:** CSR Attribute CSR 0x0014 field (RW - Initial value: 11b)  
00 : reserved  
01 : reserved  
10 : HostCSR  
11 : Ack\_Type\_Error
- Bit 18~19**      **CSRAttrib7:** CSR Attribute CSR 0x0018 field (RW - Initial value: 0b)  
00 : InternalCSR  
01 : InternalCSR(NoInt)  
10 : HostCSR  
11 : Ack\_Type\_Error
- Bit 16~17**      **CSRAttrib8:** CSR Attribute CSR 0x001C field (RW - Initial value: 0b)  
00 : InternalCSR  
01 : InternalCSR(NoInt)  
10 : HostCSR  
11 : Ack\_Type\_Error
- Bit 14~15**      **CSRAttrib9:** CSR Attribute CSR 0x0020-0x01FC field (RW - Initial value: 11b)  
00 : reserved  
01 : reserved  
10 : HostCSR  
11 : Ack\_Type\_Error
- Bit 12~13**      **CSRAttrib10:** CSR Attribute CSR 0x0200 field (RW - Initial value: 0b)  
00 : InternalCSR  
01 : InternalCSR(NoInt)  
10 : HostCSR  
11 : Ack\_Type\_Error
- Bit 10~11**      **CSRAttrib11:** CSR Attribute CSR 0x0204 field (RW - Initial value: 0b)  
00 : InternalCSR  
01 : InternalCSR(NoInt)  
10 : HostCSR  
11 : Ack\_Type\_Error
- Bit 8~9**          **CSRAttrib12:** CSR Attribute CSR 0x0208 field (RW - Initial value: 11b)  
00 : reserved  
01 : reserved  
10 : HostCSR  
11 : Ack\_Type\_Error
- Bit 6~7**          **CSRAttrib13:** CSR Attribute CSR 0x020C field (RW - Initial value: 11b)  
00 : reserved  
01 : reserved  
10 : HostCSR  
11 : Ack\_Type\_Error

**Bit 4~5**      **CSRAttrib14:** CSR Attribute CSR 0x0210 field (RW - Initial value: 0b)

- 00 : InternalCSR
- 01 : InternalCSR(NoInt)
- 10 : HostCSR
- 11 : Ack\_Type\_Error

**Bit 2~3**      **CSRAttrib15:** CSR Attribute CSR 0x0214 field (RW - Initial value: 11b)

- 00 : reserved
- 01 : reserved
- 10 : HostCSR
- 11 : Ack\_Type\_Error

**Bit 0~1**      **CSRAttrib16:** CSR Attribute CSR 0x0218 field (RW - Initial value: 11b)

- 00 : reserved
- 01 : reserved
- 10 : HostCSR
- 11 : Ack\_Type\_Error

**3-2-47 CSR Config2 Register**

Address      CCh  
 initial value    00FC 0000h

7	6	5	4	3	2	1	0
15	14	13	12	11	10	9	8
23	22	21	20	19	18	17	16
CSRAttrib21		CSRAttrib22		CSRAttrib23		CSRAttrib24	
31	30	29	28	27	26	25	24
CSRAttrib17		CSRAttrib18		CSRAttrib19		CSRAttrib20	

**Bit 17~16**      **CSRAttrib24:** CSR Attribute CSR 0x0400\_0x07FC field (RW - Initial value: 01b)

- 00 : reserved
- 01 : InternalCSR(NoInt)
- 10 : HostCSR
- 11 : Ack\_Type\_Error

**Bit 30~31**      **CSRAttrib17:** CSR Attribute CSR 0x021C field (RW - Initial value: 0b)

- 00 : InternalCSR
- 01 : InternalCSR(NoInt)
- 10 : HostCSR
- 11 : Ack\_Type\_Error

- 
- Bit 28~29**      **CSRAttrib18:** CSR Attribute CSR 0x0220 field (RW - Initial value: 0b)  
00 : InternalCSR  
01 : InternalCSR(NoInt)  
10 : HostCSR  
11 : Ack\_Type\_Error
- Bit 26~27**      **CSRAttrib19:** CSR Attribute CSR 0x0224 field (RW - Initial value: 0b)  
00 : InternalCSR  
01 : InternalCSR(NoInt)  
10 : HostCSR  
11 : Ack\_Type\_Error
- Bit 24~25**      **CSRAttrib20:** CSR Attribute CSR 0x0228 field (RW - Initial value: 0b)  
00 : InternalCSR  
01 : InternalCSR(NoInt)  
10 : HostCSR  
11 : Ack\_Type\_Error
- Bit 22~23**      **CSRAttrib21:** CSR Attribute CSR 0x022C field (RW - Initial value: 11b)  
00 : reserved  
01 : reserved  
10 : HostCSR  
11 : Ack\_Type\_Error
- Bit 20~21**      **CSRAttrib22:** CSR Attribute CSR 0x0230 field (RW - Initial value: 11b)  
00 : reserved  
01 : reserved  
10 : HostCSR  
11 : Ack\_Type\_Error
- Bit 18~19**      **CSRAttrib23:** CSR Attribute CSR 0x0234-0x03FC field (RW - Initial value: 11b)  
00 : reserved  
01 : reserved  
10 : HostCSR  
11 : Ack\_Type\_Error
- Bit 17~16**      **CSRAttrib24:** CSR Attribute CSR 0x0400-0x07FC field (RW - Initial value: 01b)  
00 : reserved  
01 : InternalCSR(NoInt)  
10 : HostCSR  
11 : Ack\_Type\_Error

### 3-2-48 CSR UserSpace Config Register

Address        D0h  
Initial value   0000 0000h

7	6	5	4	3	2	1	0
US2_Size							
15	14	13	12	11	10	9	8
				US2_Lock	US2_Read	US2_Write	US2_Size
23	22	21	20	19	18	17	16
US1_Size							
31	30	29	28	27	26	25	24
				US1_Lock	US1_Read	US1_Write	US1_Size

Setting is made for Bank0 (US1) and Bank1 (US2) of UserSpace.

**Bit 0~8        US2 Size:** User Space Address field (RW - Initial value: 0b)

Size of Bank1 is set up in the unit of bytes. However, the lower two bits are fixed at "0".

**Bit 9         US2\_Write:** User Space WRITE Attribute field (RW - Initial value: 0b)

The Write Authorize attribute is specified for the UserSpace register of Bank1. When this bit is set, writing with a Write Request packet is possible in the UserSpace register of Bank1. However, when accessed via the Internal CSR Access register, writing is possible even though this bit has not been set up. If there is a request while writing is not authorized, processing is forwarded with Ack\_type\_error.

**Bit 10        US22\_Read:** User Space READ Attribute field (RW - Initial value: 0b)

The Read Authorize attribute is specified for the UserSpace register of Bank1. When this bit is set, writing with a Read Request packet is possible in the UserSpace register of Bank1. However, when accessed via the Internal CSR Access register, writing is possible even though this bit has not been set up. If there is a request while writing is not authorized, processing is forwarded with Ack\_type\_error.

**Bit 11        US2\_Lock:** User Space LOCK Attribute field (RW - Initial value: 0b)

The Lock Authorize attribute is specified for the UserSpace register of Bank1. When this bit is set, writing with a Lock Request packet is possible in the UserSpace register of Bank1. If there is a request while writing is not authorized, processing is forwarded with Ack\_type\_error.

**Bit 16~24    US1 Size:** User Space Address field (RW - Initial value: 0b)

Size of Bank0 is set up in the unit of bytes. However, the lower two bits are fixed at "0".

**Bit 25        US1\_Write:** User Space WRITE Attribute field (RW - Initial value: 0b)

The Write Authorize attribute is specified for the UserSpace register of Bank0. When this bit is set, writing with a Write Request packet is possible in the UserSpace register of Bank0. However, when accessed via the Internal CSR Access register, writing is possible even though this bit has not been set up. If there is a request while writing is not authorized, processing is forwarded with Ack\_type\_error.

**Bit 26        US1\_Read:** User Space READ Attribute field (RW - Initial value: 0b)

The Read Authorize attribute is specified for the UserSpace register of Bank0. When this bit is set, writing with a Read Request packet is possible in the UserSpace register of Bank0. However, when accessed via the Internal CSR Access register, writing is possible even though this bit has not been set up. If there is a request while writing is not authorized, processing is forwarded with Ack\_type\_error.

**Bit 27 US1\_Lock:** User Space LOCK Attribute field (RW - Initial value: 0b)

The Lock Authorize attribute is specified for the UserSpace register of Bank0. When this bit is set, writing with a Lock Request packet is possible in the UserSpace register of Bank1. If there is a request while writing is not authorized, processing is forwarded with Ack\_type\_error.

### 3-2-49 CSR UserSpace Config Register

Address D4h  
Initial value 0000 0000h

7	6	5	4	3	2	1	0
US4_Size							
15	14	13	12	11	10	9	8
				US4_Lock	US4_Read	US4_Write	US4_Size
23	22	21	20	19	18	17	16
US3_Size							
31	30	29	28	27	26	25	24
				US3_Lock	US3_Read	US3_Write	US3_Size

**Bit 0~8 US4 Size:** User Space Address field (RW - Initial value: 0b)

Size of Bank3 is set up in the unit of bytes. However, the lower two bits are fixed at "0".

**Bit 9 US4\_Write:** User Space WRITE Attribute field (RW - Initial value: 0b)

The Write Authorize attribute is specified for the UserSpace register of Bank3. When this bit is set, writing with a Write Request packet is possible in the UserSpace register of Bank1. However, when accessed via the Internal CSR Access register, writing is possible even though this bit has not been set up. If there is a request while writing is not authorized, processing is forwarded with Ack\_type\_error.

**Bit 10 US4\_Read:** User Space READ Attribute field (RW - Initial value: 0b)

The Read Authorize attribute is specified for the UserSpace register of Bank3. When this bit is set, writing with a Read Request packet is possible in the UserSpace register of Bank1. However, when accessed via the Internal CSR Access register, writing is possible even though this bit has not been set up. If there is a request while writing is not authorized, processing is forwarded with Ack\_type\_error.

**Bit 11 US4\_Lock:** User Space LOCK Attribute field (RW - Initial value: 0b)

The Lock Authorize attribute is specified for the UserSpace register of Bank3. When this bit is set, writing with a Lock Request packet is possible in the UserSpace register of Bank1. If there is a request while writing is not authorized, processing is forwarded with Ack\_type\_error.

**Bit 16~24 US3\_Size:** User Space Address field (RW - Initial value: 0b)

Size of Bank2 is set up in the unit of bytes. However, the lower two bits are fixed at "0".

**Bit 25**      **US3\_Write:** User Space WRITE Attribute field (RW - Initial value: 0b)

The Write Authorize attribute is specified for the UserSpace register of Bank2. When this bit is set, writing with a Write Request packet is possible in the UserSpace register of Bank2. However, when accessed via the Internal CSR Access register, writing is possible even though this bit has not been set up. If there is a request while writing is not authorized, processing is forwarded with Ack\_type\_error.

**Bit 26**      **US3\_Read:** User Space READ Attribute field (RW - Initial value: 0b)

The Read Authorize attribute is specified for the UserSpace register of Bank2. When this bit is set, writing with a Read Request packet is possible in the UserSpace register of Bank0. However, when accessed via the Internal CSR Access register, writing is possible even though this bit has not been set up. If there is a request while writing is not authorized, processing is forwarded with Ack\_type\_error.

**Bit 27**      **US3\_Lock:** User Space LOCK Attribute field (RW - Initial value: 0b)

The Lock Authorize attribute is specified for the UserSpace register of Bank2. When this bit is set, writing with a Lock Request packet is possible in the UserSpace register of Bank1. If there is a request while writing is not authorized, processing is forwarded with Ack\_type\_error.

**3-2-50 CSR UserSpace Config Register**

Address D8h, DCh, E0h, E4h

Initial value 0000 0000h

7	6	5	4	3	2	1	0
US1_Adrs							
15	14	13	12	11	10	9	8
US1_Adrs							
23	22	21	20	19	18	17	16
US1_Adrs							
31	30	29	28	27	26	25	24
				US1_Adrs			

**Bit 0~27 US1 Adrs: User Space1 Address field (RW - Initial value: 0b)**

The Bank0 register is specified to define in what position of the CSR space this register should be allocated. The lower two bits are fixed at "0". Since the designated value is of 28 bits, the remaining 20 bits are fixed at 0x000F FFFF. Therefore, the UserSpace can be allocated in the following spaces:

0xFFFF F000 0000 to 0xFFFF FFFF FFFF

7	6	5	4	3	2	1	0
US2_Adrs							
15	14	13	12	11	10	9	8
US2_Adrs							
23	22	21	20	19	18	17	16
US2_Adrs							
31	30	29	28	27	26	25	24
				US2_Adrs			

**Bit 0~27 US2 Adrs: User Space2 Address field (RW - Initial value: 0b)**

The Bank1 register is specified to define in what position of the CSR space this register should be allocated. The lower two bits are fixed at "0". Since the designated value is of 28 bits, the remaining 20 bits are fixed at 0x000F FFFF. Therefore, the UserSpace can be allocated in the following spaces:

0xFFFF F000 0000 to 0xFFFF FFFF FFFF

7	6	5	4	3	2	1	0
US3_Adrs							
15	14	13	12	11	10	9	8
US3_Adrs							
23	22	21	20	19	18	17	16
US3_Adrs							
31	30	29	28	27	26	25	24
				US3_Adrs			

**Bit 0~27 US3 Adrs: User Space 3Address field (RW - Initial value: 0b)**

The Bank2 register is specified to define in what position of the CSR space this register should be allocated. The lower two bits are fixed at "0". Since the designated value is of 28 bits, the remaining 20 bits are fixed at 0x000F FFFF. Therefore, the UserSpace can be allocated in the following spaces:

0xFFFF F000 0000 to 0xFFFF FFFF FFFF

7	6	5	4	3	2	1	0
US4_Adrs							
15	14	13	12	11	10	9	8
US4_Adrs							
23	22	21	20	19	18	17	16
US4_Adrs							
31	30	29	28	27	26	25	24
				US4_Adrs			

**Bit 0~27 US4 Adrs: User Space4 Address field (RW - Initial value: 0b)**

The Bank3 register is specified to define in what position of the CSR space this register should be allocated. The lower two bits are fixed at "0". Since the designated value is of 28 bits, the remaining 20 bits are fixed at 0x000F FFFF. Therefore, the UserSpace can be allocated in the following spaces:

0xFFFF F000 0000 to 0xFFFF FFFF FFFF

### 3-2-51 Active Node Map Register

Address E8h, ECh

Initial value 0000 0000h

After the reception of SelfID, this SelfID is analyzed as described above and it is then stored. ActiveNodeMap[0] is B8h, and activeNodeMap[1] is Bch. "1" is always set for the own node.

7	6	5	4	3	2	1	0
Node24	Node25	Node26	Node27	Node28	Node29	Node30	Node31
15	14	13	12	11	10	9	8
Node16	Node17	Node18	Node19	Node20	Node21	Node22	Node23
23	22	21	20	19	18	17	16
Node8	Node9	Node10	Node11	Node12	Node13	Node14	Node15
31	30	29	28	27	26	25	24
Node0	Node1	Node2	Node3	Node4	Node5	Node6	Node7

**Bit 0~31**      **activeNode31 to0:** Active Node 31~0 bit (R - Initial value: 0b)

0 : Not Active Node

1 : Active Node

7	6	5	4	3	2	1	0
Node56	Node57	Node58	Node59	Node60	Node61	Node62	
15	14	13	12	11	10	9	8
Node48	Node49	Node50	Node51	Node52	Node53	Node54	Node55
23	22	21	20	19	18	17	16
Node40	Node41	Node42	Node43	Node44	Node45	Node46	Node47
31	30	29	28	27	26	25	24
Node32	Node33	Node34	Node35	Node36	Node37	Node38	Node39

**Bit 1~31**      **activeNode62 to32:** Active Node 62~32 bit (R - Initial value: 0b)

0 : Not Active Node

1 : Active Node

### 3-2-52 Speed Map Register

Address F0h, F4h, F8h, FCh

Initial value 0000 0000h

After the reception of SelfID, this SelfID is analyzed as described above and it is then stored. SpeedNodeMap[0] is C0h, and.....speedMap[0] is CCh. When a SelfID packet of own node is received, a Speed code of this SelfID is stored. If the SelfID packet is not received, "11b" is set up.

7	6	5	4	3	2	1	0
Speed12		Speed13		Speed14		Speed15	
15	14	13	12	11	10	9	8
Speed8		Speed9		Speed10		Speed11	
23	22	21	20	19	18	17	16
Speed4		Speed5		Speed6		Speed7	
31	30	29	28	27	26	25	24
Speed0		Speed1		Speed2		Speed3	

7	6	5	4	3	2	1	0
Speed28		Speed29		Speed30		Speed31	
15	14	13	12	11	10	9	8
Speed24		Speed25		Speed26		Speed27	
23	22	21	20	19	18	17	16
Speed20		Speed21		Speed22		Speed23	
31	30	29	28	27	26	25	24
Speed16		Speed17		Speed18		Speed19	

7	6	5	4	3	2	1	0
Speed44		Speed45		Speed46		Speed47	
15	14	13	12	11	10	9	8
Speed40		Speed41		Speed42		Speed43	
23	22	21	20	19	18	17	16
Speed36		Speed37		Speed38		Speed39	
31	30	29	28	27	26	25	24
Speed32		Speed33		Speed34		Speed35	

7	6	5	4	3	2	1	0
Speed60		Speed61		Speed62			
15	14	13	12	11	10	9	8
Speed56		Speed57		Speed58		Speed59	
23	22	21	20	19	18	17	16
Speed52		Speed53		Speed54		Speed55	
31	30	29	28	27	26	25	24
Speed48		Speed49		Speed50		Speed51	

**Bit 0~1**      **Speed0~62:** Speed Node 0~62 field (R - Initial value: 0b)

00 : S100

01 : S200

10 : S400

11 : reserved

3-3 List of registers

Register	Adrs	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Version	00h	Version								Version								Revision								Revision													
Control	04h	DackHigh	DreqLow	ITFBusReset	ARFBusReset	CSEn	IACTIVE	AACTIVE	LPSON	IsoBusModelH	IsoBusModelL	ITF/IRF	IsoDataMode	ITZERO	CycleMaster	CycleTimerEn					DMAWidth			IsoConEn	AckAccEn	cmstr	PHYIFRST	ROMChkEnd					ReceiveEn	TransmitEn					
Node Identification	08h	IDValid	root	rootNode								irmValid	irmNode								BusNumber-H								BusNumber-L	NodeNumber									
Reset	0Ch																																	ResetDMA	ResetLink	ResetTx	ResetITF/IRF	ResetARF	ResetATF
Packet Control	10h																				RetryEn	ackAddressErr	WritePending	OnePacketRcv	BusyCtrl				SelfIDOut	SelfIDErr	RxSelfID	EnSnoop	PhyIDOut						
Diagnostic Status and Control	14h			LinkReady	EEPROM_C			ITBusy	ATBusy																												OnePacketProcStatus	OnePacketProcEnd	
Phy Control	18h																				RegRcvd	RdReg	WrReg	RegAddr				RegData											
ATRetry	1Ch	RetryStop		RetryX	RetryAB	secondCount			cycleCount	cycleCount								cycleCount				retryCount																	
Reserved	20h																																						
Cycle Timer	24h	CycleSeconds								CycleCount	CycleCount								CycleCount	CycleOffset	CycleOffset																		
Isochronous Transmit Configuration	28h	Tag	Channel														Speed				Sync	StartSync	StopSync															SyncEn	
Isochronous Transmit Configuration	2Ch																				ITLength								ITLength										

Table 3-3-1 List of registers 1

Register	Adrs	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Isochronous Bus Receive Configuration 1	30h	Tag	Channel																			StartSync	StopSync				CTEn	IsoRxEn	SyncEn				
Isochronous Bus Receive Configuration 2	34h	Tag	Channel																			StartSync	StopSync				CTEn	IsoRxEn	SyncEn				
Isochronous Bus Receive Configuration 3	38h	Tag	Channel																			StartSync	StopSync				CTEn	IsoRxEn	SyncEn				
Isochronous Bus Receive Configuration 4	3Ch	Tag	Channel																			StartSync	StopSync				CTEn	IsoRxEn	SyncEn				
Isochronous Buffer Receive Configuration	40h	Tag	Channel																			StartSync	StopSync				CTEn	IsoRxEn	SyncEn				
Isochronous Bus Receive CT Configuration 1	44h	CycleSeconds							CycleCount	CycleCount																							
Isochronous Bus Receive CT Configuration 2	48h	CycleSeconds							CycleCount	CycleCount																							
Isochronous Receive Error	4Ch	Err_CycleSeconds							Err_CycleCount	Err_CycleCount							Err_Code	Err_Speed	Err_Tag	Err_Channel													
Asynchronous Stream Receive Configuration	50h	ASEn1								ASTag1	ASChannel1							ASEn2								ASTag2	ASChannel2						
ATF Data	54h	ATFData							ATFData							ATFData							ATFData										
ARF Data	58h	ARFData							ARFData							ARFData							ARFData										
ITFData	5Ch	ITFData							ITFData							ITFData							ITFData										
IRFData	60h	IRFData							IRFData							IRFData							IRFData										

Table 3-3-2 List of registers 2

Register	Adrs	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Buffer Status and Control	64h	DackEn	DreqEn	SelectDreq	DMA_Little			Little		IRFEmpty	ITFFull	ITFEmpty			ARFEmpty	ATFFull	ATFEmpty	IRFSize										IRFSize									
Asynchronous Bus Control 1	68h	ABTxEn																							ABTxLen	ABTxLen											
Asynchronous Bus Control 2	6Ch	ABRxEn																							ABRxLen	ABRxLen											
Interrupt	70h	ABTxEnd	ABRxEnd	SelfDpkRcvd	ARFRej	BTInt	GTInt	STO	ReTryLimit	ATxEnd	ATxCSREnd	ARxCSREnd	ARxEnd	ITxEnd	ITFNoTx	IRxErr	IRxEnd	PhyInt	BusReset	BusReady	PhyRegRcvd	AckErr	TCodeErr	HdrErr	DatErr	CycleSeconds	CycleStart	CycleDone	CycleLost	EEPROM_Ready			extended				
Extended Interrupt	74h																																CmdReset	LinkOff	cmstr		
Interrupt Mask 1	78h	ABTxEnd	ABRxEnd	SelfDpkRcvd	ARFRej	BTInt	GTInt	STO	ReTryLimit	ATxEnd	ATxCSREnd	ARxCSREnd	ARxEnd	ITxEnd	ITFNoTx	IRxErr	IRxEnd	PhyInt	BusReset	BusReady	PhyRegRcvd	AckErr	TCodeErr	HdrErr	DatErr	CycleSeconds	CycleStart	CycleDone	CycleLost	EEPROM_Ready			extended				
Interrupt Mask 2	7Ch	ABTxEnd	ABRxEnd	SelfDpkRcvd	ARFRej	BTInt	GTInt	STO	ReTryLimit	ATxEnd	ATxCSREnd	ARxCSREnd	ARxEnd	ITxEnd	ITFNoTx	IRxErr	IRxEnd	PhyInt	BusReset	BusReady	PhyRegRcvd	AckErr	TCodeErr	HdrErr	DatErr	CycleSeconds	CycleStart	CycleDone	CycleLost	EEPROM_Ready			extended				
Split TimeOut	80h	SplitTimeLimit					SplitTimeLimit					SplitTime					SplitTime																				
Split Timer Start	84h																																	STStart			
Bus Configuration Timer	88h	IntTimer4	IntTimer3	IntTimer2	IntTimer1																					BCCTime					BCCTime						
Bus Configuration Timer 1	8Ch	BCTimer1					BCTimer1					BCTimer2					BCTimer2																				
Bus Configuration Timer 2	90h	BCTimer3					BCTimer3					BCTimer4					BCTimer4																				
General Timer	94h	GTEn	GTOneShot	GCTime					GCTime					GTimer					GTimer																		

Table 3-3-3 List of registers 3

Register	Adrs	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
EEPROM Control	98h	EEPROM_adrs								EEPROM_adrs								0	0																	EEPROM_ProcErr	EEPROM_ProcErr					EEPROM_ReadErr	EEPROM_WriteErr						
EEPROM Data	9Ch	EEPROM_data								EEPROM_data								EEPROM_data								EEPROM_data																							
GPIO	A0h																	GPIO(1:0)																		GPIOCTL(1:0)													
ATGo	A4h																																	ATGo															
ITGo	A8h																																	ITGo															
Internal CSR Access	ACh	CSRAccErr	CSRRCv	Block	Lock	Write	Read									CSRCore	USBank	CSRAdrOff								CSRAdrOff																							
Internal CSR Access DataH	B0h	CSRDataH								CSRDataH								CSRDataH								CSRDataH																							
Internal CSR Access DataL	B4h	CSRDataL								CSRDataL								CSRDataL								CSRDataL																							
CSR Transaction 1	B8h	sourceBusID								sourceBusID	sourceNodeID								CSRAdrsOff								CSRAdrsOff																						
CSR Transaction 2	BCh	RespGo	CompOK	Block	LockReq	WriteReq									BusID	CSRCore	USBank	spd								AckCode				RespCode																			
CSR Transaction DataH	C0h	ResponseDataH								ResponseDataH								ResponseDataH								ResponseDataH																							
CSR Transaction DataL	C4h	ResponseDataL								ResponseDataL								ResponseDataL								ResponseDataL																							
CSR Config1	C8h	CSRAttrib1	CSRAttrib2	CSRAttrib3	CSRAttrib4	CSRAttrib5	CSRAttrib6	CSRAttrib7	CSRAttrib8	CSRAttrib9	CSRAttrib10	CSRAttrib11	CSRAttrib12	CSRAttrib13	CSRAttrib14	CSRAttrib15	CSRAttrib16																																

Table 3-3-4 List of registers 4

Register	Adrs	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
CSR Config2	CCh	CSRAttrib17		CSRAttrib18		CSRAttrib19		CSRAttrib20		CSRAttrib21		CSRAttrib22		CSRAttrib23																				
CSR UserSpace Config	D0h					US1_Read	US1_Write	US1_Size		US1_Size												US2_Lock	US2_Read	US2_Write	US2_Size	US2_Size								
CSR UserSpace Config	D4h					US3_Lock	US3_Read	US3_Write	US3_Size	US3_Size													US4_Lock	US4_Read	US4_Write	US4_Size	US4_Size							
CSR UserSpace Config	D8h					US1_Adrs				US1_Adrs				US1_Adrs				US1_Adrs																
CSR UserSpace Config	DCh					US2_Adrs				US2_Adrs				US2_Adrs				US2_Adrs																
CSR UserSpace Config	E0h					US3_Adrs				US3_Adrs				US3_Adrs				US3_Adrs																
CSR UserSpace Config	E4h					US4_Adrs				US4_Adrs				US4_Adrs				US4_Adrs																
Active Node Map	E8h	Node0	Node1	Node2	Node3	Node4	Node5	Node6	Node7	Node8	Node9	Node10	Node11	Node12	Node13	Node14	Node15	Node16	Node17	Node18	Node19	Node20	Node21	Node22	Node23	Node24	Node25	Node26	Node27	Node28	Node29	Node30	Node31	
Active Node Map	ECh	Node32	Node33	Node34	Node35	Node36	Node37	Node38	Node39	Node40	Node41	Node42	Node43	Node44	Node45	Node46	Node47	Node48	Node49	Node50	Node51	Node52	Node53	Node54	Node55	Node56	Node57	Node58	Node59	Node60	Node61	Node62		
Speed Map	F0h	Speed0		Speed1	Speed2	Speed3	Speed4	Speed5	Speed6	Speed7	Speed8	Speed9	Speed10	Speed11	Speed12	Speed13	Speed14	Speed15																
Speed Map	F4h	Speed16	Speed17	Speed18	Speed19	Speed20	Speed21	Speed22	Speed23	Speed24	Speed25	Speed26	Speed27	Speed28	Speed29	Speed30	Speed31																	
Speed Map	F8h	Speed32	Speed33	Speed34	Speed35	Speed36	Speed37	Speed38	Speed39	Speed40	Speed41	Speed42	Speed43	Speed44	Speed45	Speed46	Speed47																	
Speed Map	FCh	Speed48	Speed49	Speed50	Speed51	Speed52	Speed53	Speed54	Speed55	Speed56	Speed57	Speed58	Speed59	Speed60	Speed61	Speed62																		

Table 3-3-5 List of registers 5

## 4 Host Access

### 4-1 Host Interface

Control of the MD8415 and operation for transmit/receive data transfer are always conducted from the host interface. The signal timing of the host interface is controlled by the respective signals of CS# RD#, WR#, HA(6:0), and HD(31:0) for SRAM-like asynchronous transfer.

The internal registers and the "1394" packet format are basically arranged on the basis of 32-bit width. For the MD8415, however, the bus width can be controlled to enable direct connection to the MPU that has 8-bit, 16-bit, or 32-bit data bus.

### 4-2 Access flow after closure of the POWER switch

After the power circuit has been turned on, hardware reset (RESET#) is completed and then the MD8415 makes CRC check for the connected EEPROM.

At this time point, however, no bits other than those located in the registers below can be accessed.

Control Register

ROMChkEnd bit, PHYIFRST bit, LPSON bit

- 1) Polling for the ROMChkEnd bit of the Control register
- 2) LPSON bit setting by examining the rise of the ROMChkEnd and LPSON bits
- 3) Polling for the LinkReady bit of the Diagnostic and Control register
- 4) Confirmation of the rise of the LinkReady bit
- 5) Access to the internal register of the MD8415 enabled since then

The end of CRC check for this EEPROM can be confirmed by examining whether the ROMChkEnd bit of the Control register has been set or not. If the 49.152MHz clock input cannot be applied to the SCLK terminal for reasons of no power source available for the connected PHY or the like, no internal register of the MD8415 can be accessed. Even though reading out is attempted, an unstable value may be output.

### 4-3 Register access timing

As shown in Fig. 4-3-1, the SRAM-like asynchronous bus is used for register access.

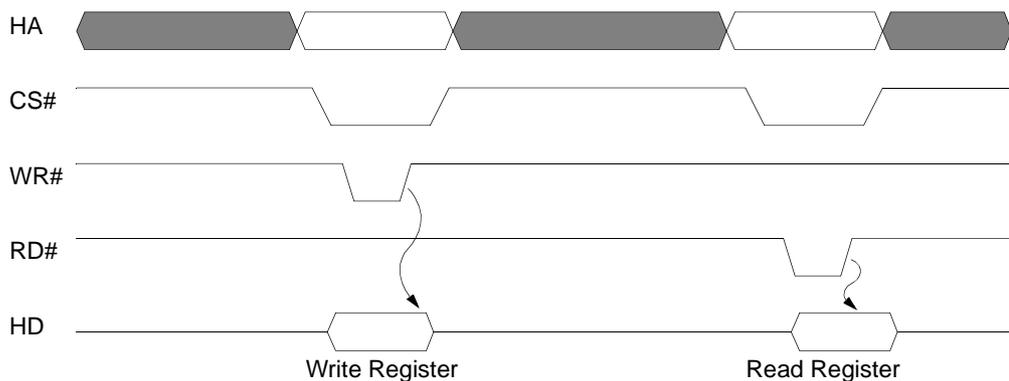


Figure 4-3-1 Host Access Timing

### 4-4 Host Bus Width

#### 4-4-1 Host Bus Width

As shown in Table 4-4-1, the valid bits for access from the host are determined by UWE#, UBE#, HA1, and HA0. Therefore, when controlling the MD8415 with the 8-bit MPU, direct control can be effected by connecting the address and the data bus respectively to HA(7:0) and HD(7:0).

UWE#	UBE#	HA1	HA0	access	Valid Host Bus	Buffer Bit Position
0	0	0	0	quadlet	31 - 0	31 - 0
1	0	0	0	word	15 - 0	31 - 16
1	0	1	0	word	15 - 0	15 - 0
1	1	0	0	byte	7 - 0	31 - 24
1	1	0	1	byte	7 - 0	23 - 16
1	1	1	0	byte	7 - 0	15 - 8
1	1	1	1	byte	7 - 0	7 - 0

Table 4-4-1 Valid Host Data Bus for Access from the Host

For Write/Read at the 8-bit MPU in the Byte unit, this operation must be carried out in the unit of 4 bytes. For the 16-bit MPU, it must be done the unit of 2 words.

The register performs normal operation if Write/Read is effected by designating a normal address . When Write/Read is performed in the Transmit/Receive buffer, data are stored in the buffer sequentially from the upper, as shown in Table 4-4-2. This is the same as for DMA transfer, and Write/Read is effected from the upper, starting with the first byte. When the Little bit is set at "1" in the 16- or 32-bit width, byte allocations are changed to perform Write/Read in the buffer, as shown in Table 4-4-2.

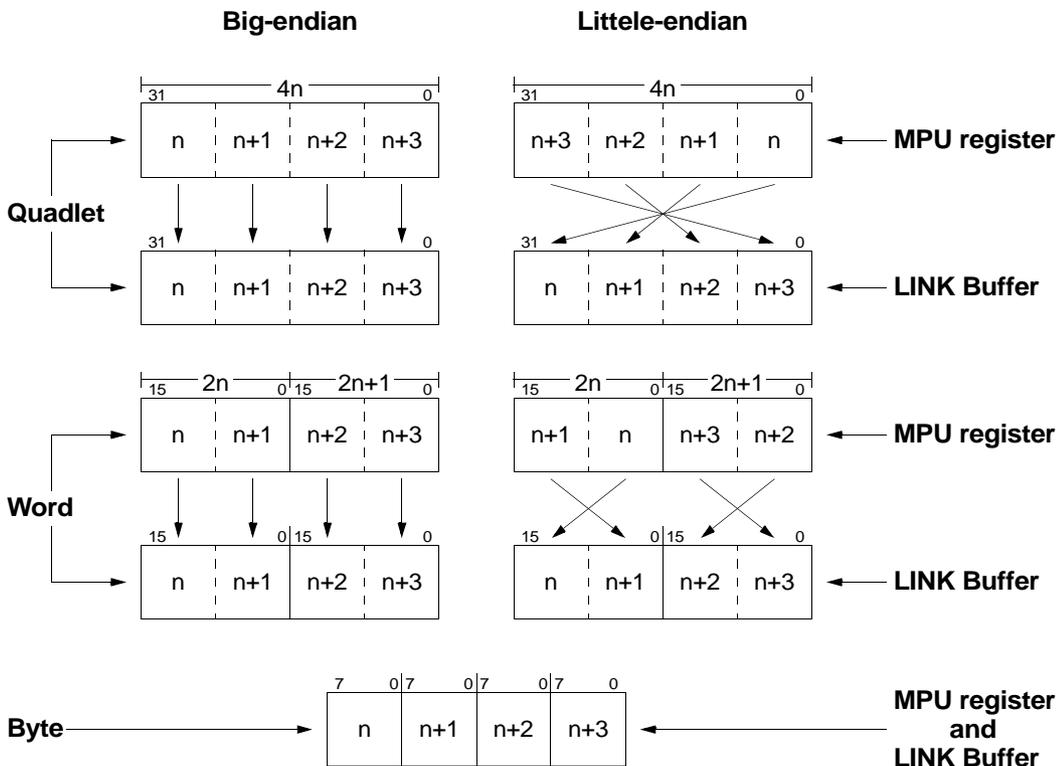


Table 4-4-2 Little / Big Endian Mode

## 4-5 Access to the internal buffer

In the MD8415, access to the buffer may be performed by two methods; software access and DMA access. Each method is explained below.

### 4-5-1 Software access

According to IEEE1394, packets are formed in the Quadlet unit. Therefore, accessing method can differ according to the bus width. In the case of 32-bit width, Write/Read of 1 Quadlet data can be performed from the register that corresponds to the buffer directly controlled. For the 8/16-bit width, however, Write/Read of 1 Quadlet data can be conducted by splitting operation into two actions.

At first, procedures for writing in the ATF buffer are explained for the ATF buffer.

Data for 1 Byte are written in the 54h register. Then writing is continued in the order of 55h, 56h, and 57h. When writing to 57h has been finished, the data for one Quadlet become available. For this reason, writing must be done in the order of 54h, 55h, 56h, and 57h.

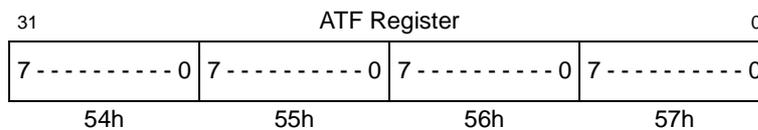


Figure 4-5-1 Register Operation for Software Access in 8-Bit Width (ATF)

For data reading from the ARF buffer, readout must be done in the order of 58h, 59h, 5Ah, and 5Bh. Same as for the Isochronous ITF/IRF or IRF buffer

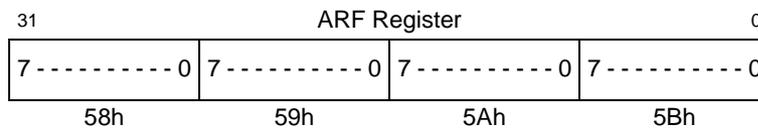


Figure 4-5-2 Register Operation for Software Access in 8-Bit Width (ARF)

For writing in the ATF buffer in 16-bit width, writing must be repeated 4 times for the 8-bit width. Since only two actions are available in this case, access in the similar procedures is needed for accessing.

First of all, data for one word are written in the 54h register. Then data are written in the order to 56h. When writing is finished by 56h, the data for one quadlet become valid. For this reason, writing order must be 54h and then 56h.

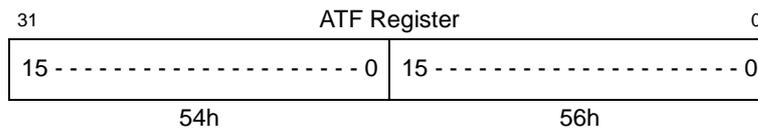


Figure 4-5-3 Register Operation for Software Access in 16-bit Width (ATF)

For data reading from the ARF buffer, similar procedure should be followed in the order of 58h and then 5Ah. The same thing can be said for the Isochronous ITF/IRF and IRF buffers.

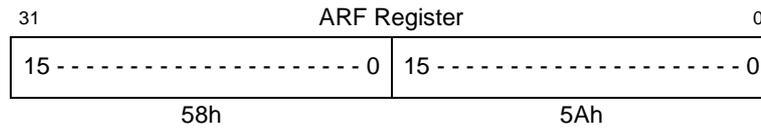


Figure 4-5-4 Register Operation for Software Access in 16-bit Width (ARF)

### 4-5-2 DMA access

For DMA access, it is necessary to designate an objective buffer according to the SelectDreq bit.

In the first place, procedures will be explained for ATF buffer writing in the 8-bit width.

At first, SelectDreq=00b is set up to designate the ATF to be an object of DMA transfer, and DMAWidth=00b is set up to specify data transfer in 8-bit width. When DMAC is started and DreqEn is set at "1," a DREQ request is issued for DMAC and DMA transfer is thus started. In this case, the first one byte is stored in the 31 to 24 bits of the ATF buffer. The second byte to the fourth byte are stored in the 23 to 16 bits, 15 to 8 bits, and 7 to 0 bits, respectively. Finally, these data become valid as the one-Quadlet data. For this reason, the number of DMA transfer actions must always be a multiple of 4.

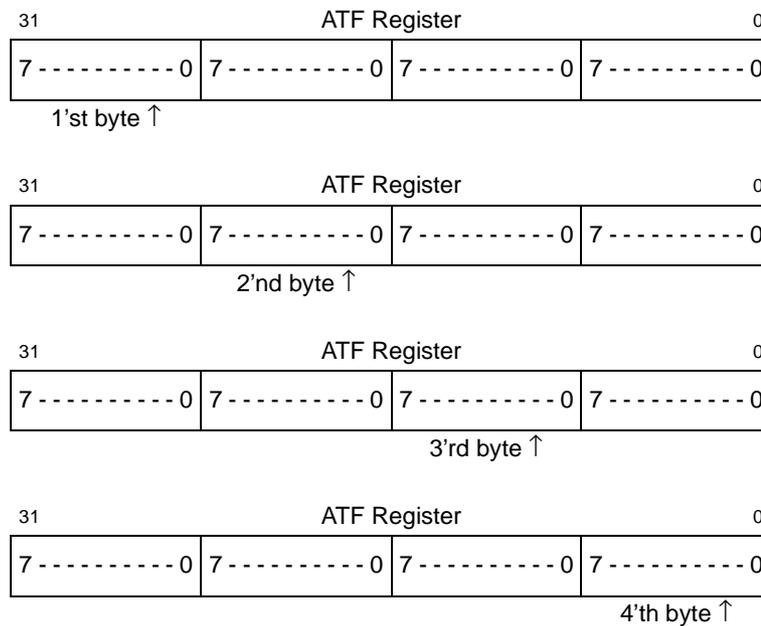


Figure 4-5-5 Register Operation for DMA Access in 8-Bit Width (ATF)

Similarly, the same operation is conducted for reading from the ARF buffer by DMA transfer. Data are read out, starting with the upper 32 bits.



Figure 4-5-6 Register Operation for DMA Access in 8-Bit Width (ARF)

Operation is the same as for 16-bit width. SelectDreq=00b is set to make ATF an object of DMA transfer, and DMAW-idth=01b is set to designate 16-bit width transfer. When DMAC is started and DreqEn is turned '1' thereafter, a DREQ request is issued toward the DMAC, and DMA transfer is started. In this case, the first 1 Word is stored in 31~16 bit of the ATF buffer and the second Word is stored in 15~0 bit. Finally, data become valid as 1-Quadlet data. Accordingly, the number of DMA transfers must be always a multiple of 2.

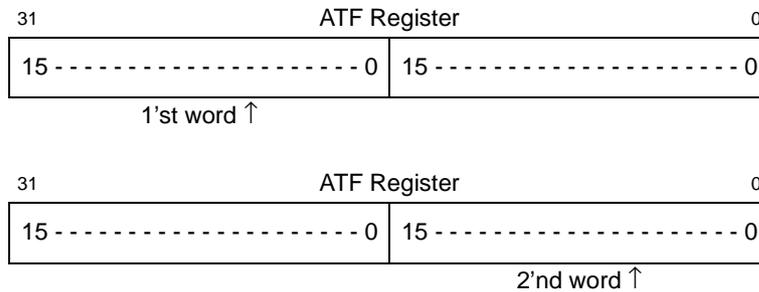


Figure 4-5-7 Register Operation for DMA Access in 16-Bit Width (ATF)

Similarly, the same operation is conducted for reading from the ARF buffer by DMA transfer. Data are read out, starting with the upper 32 bits.



Figure 4-5-8 Register Operation for DMA Access in 16-Bit Width (ARF)

## 5 PHY Chip Interface

### 5-1 Method of connection

As a means of connections with the PHY chip, the MD8415 supports both DC and AC connections. For DC connections, connections are made by the method as shown in Fig. 5-1-1.

To realize a connection with the MD8404 that is a PHY chip of 400Mbps, such a connection is made to the terminal of D (7:0) that is intended for the AC connection. (Refer to Fig. 5-1-2.)

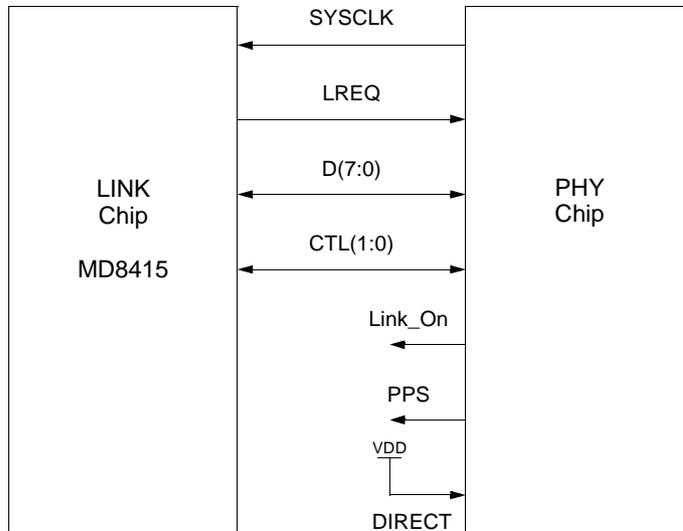


Figure 5-1-1 Diagram of Connections between MD8415 and PHY Chip

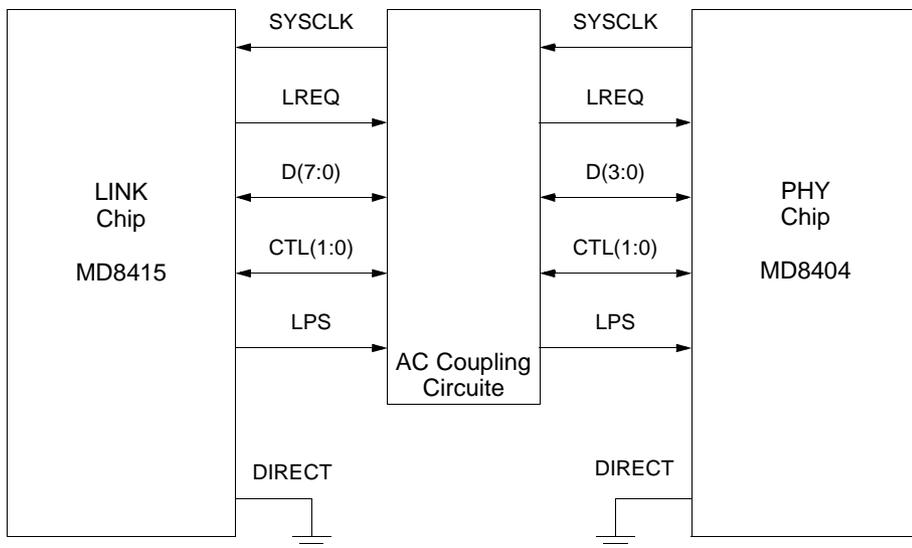


Figure 5-1-2 Diagram of Connections between MD8415 and MD8404

### 5-2 Phy/Link I/F Timing

Disable/enable control of the PHY-LINK interface is conducted through the terminals of the MD8415 and LPS.

The LPS signal is used for the output control as described below, by the use of the LPSON bit in the related register of this device and through the external and DIRECT terminals.

DIRECT	LPSON	LPS Output
1	0	0
1	1	1
0	0	0
0	1	Clock of Approx. 0.6 to 3.6MHz (Duty 33%)

Table 5-2-1 LPS output

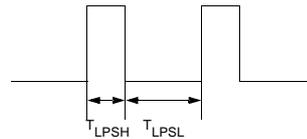


Figure 5-2-1 LPS output waveform for AC connection

Symbol	Explanation	MIN	MAX	Unit
T <sub>LPSH</sub>	LPS"H" period for AC connection	0.09	0.50	μs
T <sub>LPSL</sub>	LPS"L" period for AC connection	0.19	1.00	μs

Table 5-2-2 LPS output characteristics for AC connection

This device has two systems of PHY-LINK interface reset timing. Changeover for the reset timing is conducted with the PHYIFRST bit in the register.

If the PHY chip to be connected to this device is one other than the devices conforming to 1394a, PHYIFRST is set at "0".

In this case, shortly after the LPSON bit in the register is set at "L," the LPS terminal begins to generate the "L" output to start up the PHY-LINK interface reset sequence. Within 1.2μs after the rise of LPS, this device generates from CTL(1:0) and D(7:0) an output of High-Z for an AC connection and "L" for a DC connection. When the LPSON bit is set at "1" again, a clock output begins to be generated from the LPS terminal for an AC connection and an "H" output for a DC connection.

At that time, after the rise of the LPS signal, "L" of CTL(1:0) is output at a timing of the heading SCLK and the PHY-LINK interface reset sequence is completed.

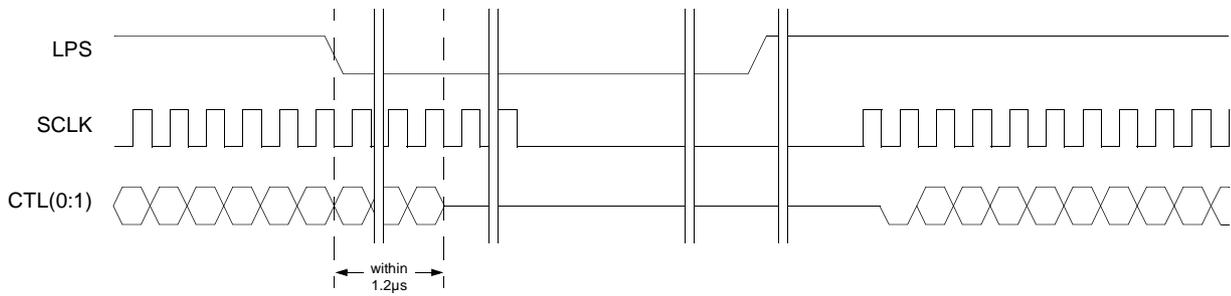


Figure 5-2-2 PHY-LINK I/F Reset Sequence for PHYIFRST='0' and AC connection

If the PHY chip to be connected to this device is the one conforming to 1394a, PHYIFRST is set at "1".

In this case, shortly after the LPSOn bit in the register is set at "1," the LPS terminal begins to generate the "L" output to start up the PHY-LINK interface reset sequence. Within 1.2µs after the fall of LPS, this device generates from CTL(1:0) and D(7:0) an output of High-Z for an AC connection and "L" for a DC connection. When the LPSOn bit is set at "1" again, a clock output begins to be generated from the LPS terminal for an AC connection and an "H" output for a DC connection.

At that time, after the rise of the LPS signal, an output of the one-SCLK "L" component of CTL(1:0) and D(7:0) is generated within six cycles after the first SCLK. Similarly as for Lreq, an output of the one-SCLK "L" component is generated for a DC connection. The "L" output is generated for a DC connection. Since then, DataPrefix -> Idle is received from the PHY chip and the PHY-LINK interface reset sequence is completed.

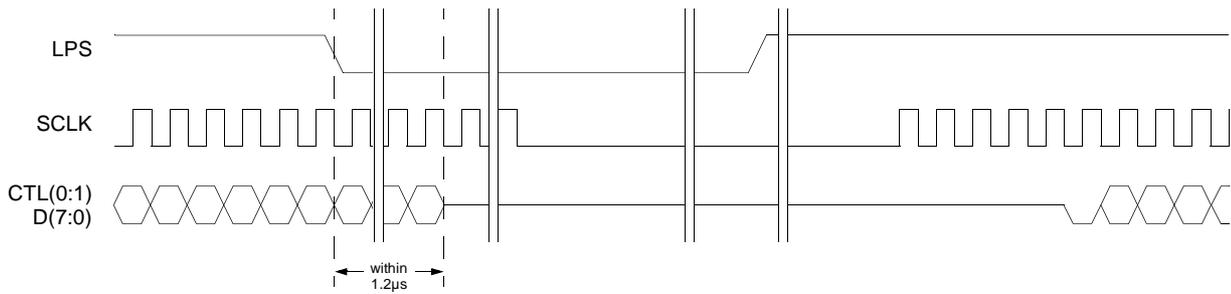


Figure 5-2-3 PHY-LINK I/F Reset Sequence for PHYIFRST='1' and AC connection

### 5-3 Access to the Phy register

After the completion of the SelfID phase, various information about the decided node number, root data, GapCount, etc., is stored in the Internal PHY register. Procedures for gaining access to the connected PHY are explained below.

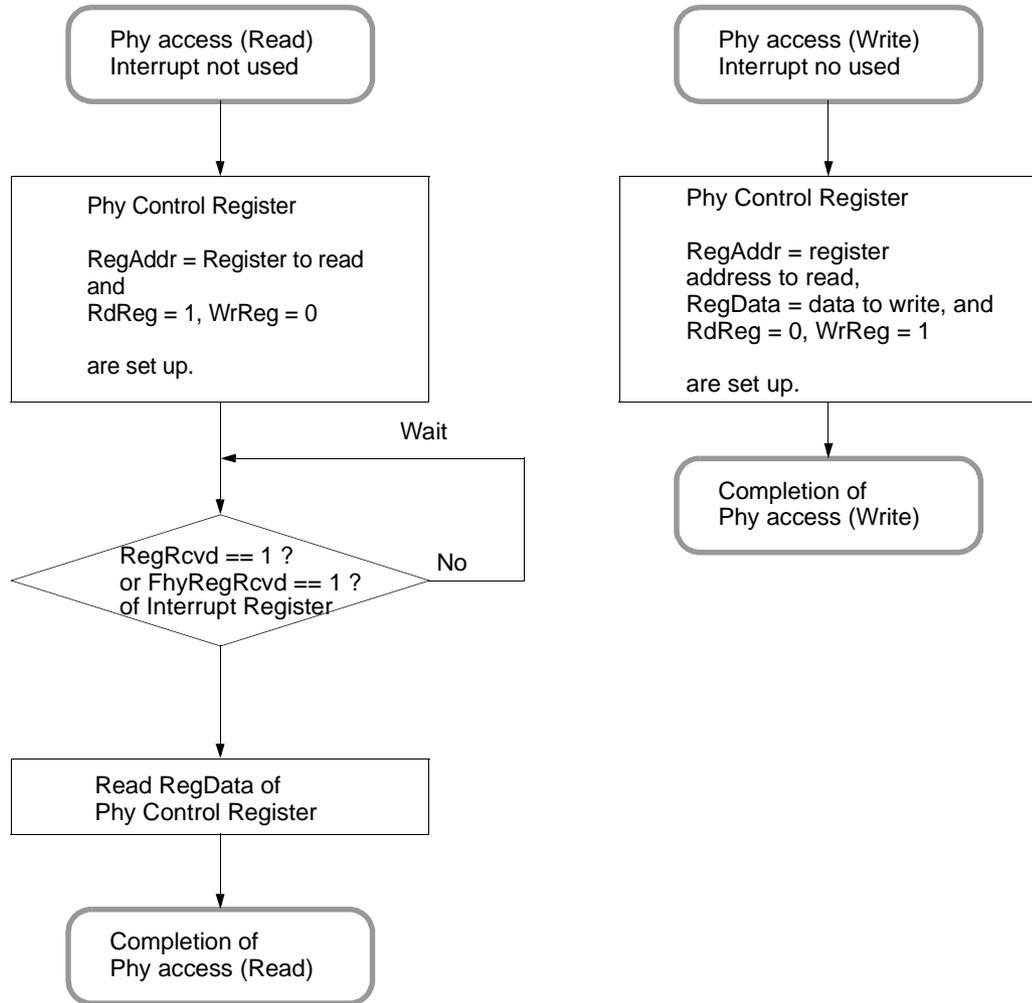


Figure 5-3-1 Access to the PHY

#### 5-4 Reading Request

A reading request may be sent to the PHY when obtaining information about own node number and root after BusReset.

As shown in Fig. 5-3-1, an internal register address of the reading PHY is set at RegAddr and RdReg is set at "1". By doing this, it is possible to issue a Read request toward the PHY. At that time, the RegData block becomes invalid.

When writing is performed in the PHY Control register by this setting, a Read request command is issued toward the PHY and the PHY data are output to the MD8415. Then, the RegRcvd bit is set at "1" by the MD8415.

When the host receives this announcement, the contents of the PHY Control register are read out and the read value is stored in the RegData block.

#### 5-5 Writing Request

A Read request to the PHY may be issued when setting GapCount or the like.

As shown in Fig. 5-3-1, an internal register address of the writing PHY is set at RegData and WrReg is set at "1". By doing this, it is possible to issue a Write request toward the PHY.

When writing is performed in the PHY Control register by this setting, a Write request command is issued toward the PHY and the data are written in the internal register of the PHY.

## 6 Internal Buffer

The MD8415 incorporates the internal buffers as specified below.

Buffer	Definition	Size
ATF	For ordinary asynchronous transmission	2048 + 56 byte
ARF	For packet reception in the external CSR space	2048 + 56 byte
ATFi	For response packet transmission to the internal CSR space, responding to a Request	512 + 28 byte
ARFi	For packet reception in the internal CSR space	28 byte
ITF/IRF	For isochronous transmission/reception	2048 + 8 byte

## 7 Asynchronous Transmission/Reception

### 7-1 Internal CSR Access

Access to the Internal CSR core and the UserSpace register comes in the two types as described below.

- 1) When accessed via "1394" by the received Request packet
- 2) Host accessed via Internal CSR Access register

When a party is accessed while the other party is being processed, the former is kept waiting until the present transaction is finished. Since then, the transaction for the former ensues immediately.

#### 7-1-1 When accessed via "1394" by the received Request packet

When an own packet is received, the destination\_offset block of the packet header is analyzed and the result comes in the three types to be described below. Transactions for Internal (core) and Internal (UserSpace) are accomplished in similar manner. Only difference is that operation for writing in the Internal CSR register designated by destinationOffset is performed in a different manner and the supporting tCode is different. If the space defined by Internal CSR (UserSpace) overlaps Internal CSR (core), the latter is in higher preference. Access to the External CSR is an ordinary reception at the ARF.

The Internal CSR register in Internal (core) performs operation defined for each bit of the respective registers. However, the register of the Internal CSR in Internal (UserSpace) is simply a register of 32-bit.

	destination_offset	Storing Interior FIFO	tCode (Type of packet stored)
Internal CSR (core)	0xFFFF F000 0000 ~ 0xFFFF F000 07FF	ARFi or ARF	Quadlet Write Request Quadlet Read Request Block Read Request Lock Request
Internal CSR (UserSpace)	Space defined by the (UserSpace) CSR UserSpace Config Register	ARFi	Quadlet Write Request Quadlet Read Request Lock Request Block(8) Write Request Block(8) Read Request
External CSR	Except above	ARF	Quadlet Write Request Quadlet Read Request Block Write Request Block Read Request Lock Request

\* The type of a packet being stored can differ according to the related register. Block(8) is for a Block request and Length is 8 or less.

	Address	Name	QR	BR	QW	BW	L
<b>Core CSR</b>							
	0x000	START_CLEAR	Enabled	Disabled	Enabled	Disabled	Disabled
	0x004	START_SET	Enabled	Disabled	Enabled	Disabled	Disabled
	0x008	NODE_IDS	Enabled	Disabled	Enabled	Disabled	Disabled
	0x00c	RESET_START	Disabled	Disabled	Enabled	Disabled	Disabled
	0x010	INDIREST_ADDRESS	Disabled	Disabled	Disabled	Disabled	Disabled
	0x014	INDIREST_DATA	Disabled	Disabled	Disabled	Disabled	Disabled
	0x018	SPRIT_TIME_OUT_HI	Enabled	Disabled	Enabled	Disabled	Disabled
	0x01c	SPRIT_TIME_OUT_L	Enabled	Disabled	Enabled	Disabled	Disabled
	0x020~0x1FC	ARGUMENT_HI	Disabled	Disabled	Disabled	Disabled	Disabled
<b>Serial Bus-dependent Register</b>							
	0x200	CYCLE_TIME	Enabled	Disabled	Enabled	Disabled	Disabled
	0x204	BUS_TIME	Enabled	Disabled	Enabled	Disabled	Disabled
	0x208	POWER_FAIL_IMMINENT	Disabled	Disabled	Disabled	Disabled	Disabled
	0x20c	POWER_SOURCE	Disabled	Disabled	Disabled	Disabled	Disabled
	0x210	BUSY_TIMEOUT	Enabled	Disabled	Enabled	Disabled	Disabled
	0x214	reserved	Disabled	Disabled	Disabled	Disabled	Disabled
	0x218	reserved	Disabled	Disabled	Disabled	Disabled	Disabled
	0x21c	BUS_MANAGER_ID	Enabled	Disabled	Disabled	Disabled	Enabled
	0x220	BANDWIDTH_AVAILABLE	Enabled	Disabled	Disabled	Disabled	Enabled
	0x224	CHANNEL_AVAILABLE	Enabled	Disabled	Disabled	Disabled	Enabled
	0x228	CHANNEL_AVAILABLE	Enabled	Disabled	Disabled	Disabled	Enabled
	0x22c	MAINT_CONTROL	Disabled	Disabled	Disabled	Disabled	Disabled
	0x230	MAINT_UTILITY	Disabled	Disabled	Disabled	Disabled	Disabled
	0x234~0x3FC	reserved	Disabled	Disabled	Disabled	Disabled	Disabled
<b>Configuration ROM</b>							
	0x400~0x7FC		Enabled	Enabled	Disabled	Disabled	Disabled

**QR** Quadlet Read Request  
**BR** Block Read Request  
**QW** Quadlet Write Request  
**BW** Block Write Request  
**L** Look Request (compare\_swap)

When access is performed while Attribute of each register area is other than the Host CSR, and if this action is attempted in a domain not permitted in the table above, type\_error (or address\_error) is returned.

If "1" is present in the lower two bits, address\_error then becomes effective.

### 7-1-1-1 Read Request

For a Read request, a packet to this space is stored in ARFi. In the case of a CSR error or a Length error, Ack\_data\_error is returned. If there is no error, Ack\_pending is returned.

From this time point, Ack\_Busy is returned to all Request packets address ed to Internal CSR (Core) and Internal CSR (UserSpace). No reception is performed. However, reception is performed only for a packet (Request/Response) destined to a space defined by HostCSR or to an area of the External CSR space.

Then, the ResponsePacketHeader data are produced from the received packet information and the resultant data are writ-ten in ATFi. At the Payload block, the data of EEPROM or the Internal CSR register specified by destinationOffset are read out for the amount specified by Length and then written in ATi.

At the next stage, a Transmit request (ATGo) is issued. Upon completion of transmission, the RespGo bit is cleared and ATxCSREnd of the Interrupt register is set up. At this time point, Internal CSR processing has been finished and the subse-quent reception of a Request packet is enabled at Internal CSR (Core) and Internal CSR (UserSpace).

In the case of a Read request to the ConfigROM space, a response is given to the request irrespective of the result of EEPROM" s check (EEPROM\_CRC bit of the Diagnostic and Status register) shortly after reset. If the data are illegible as a result of readout at the EEPROM (due to EEPROM not connected, accessing to an address not supported, and so on), the MD8415 does not generate any Response packet. In other words, Ack\_Pending is returned to Ack for the Read request, but no Response packet is transmitted.

### 7-1-1-2 Write/Lock request to Internal CSR (core)

For a Write/Lock request, four types of transactions are conducted by setting at the CSR Config register.

- |                       |                |
|-----------------------|----------------|
| 1) Internal CSR       | Stored in ARFi |
| 2) InternalCSR(NoInt) | Stored in ARFi |
| 3) HostCSR            | Stored in ARF  |
| 4) Ack_type_error     |                |

### 7-1-1-3 When set at Internal CSR

When a packet oriented to this space is stored in ARFi and a CRC error or a Length error occurs, Ack\_data\_error is returned. If there is no error, Ack is returned by WritePending located in the Packet Control register. If a request is addressed to Internal CSR (UserSpace), the Block" s Write/Read is enabled, but type\_error is made valid if Length has exceeded 8.

From this time point, Ack\_Busy is returned to all Request packets addressed to Internal CSR (Core) and Internal CSR (UserSpace). No reception is performed. However, reception is performed only for a packet (Request/Response) destined to a space defined by HostCSR or to an area of the External CSR space.

The MD8415 reads out a packet from ARFi and holds it as described below.

Received Packet	CSR Transaction Register	ResponsePacket generation
destinationID	None	None
tLabel	None	tLabel
rt	None	rt
tCode	If tCode is WriteRequest, WriteReq=1. Block=1 in case of Block. If tCode is LockRequest, LockReq=1.	WriteResponse/ ReadResponse/ LockResponse
priority	None	priority
sourceID	sourceBusID, sourceNodeID	destinationID
destinationOffset	If destinationOffset is Internal (Core), CSRCore=1.  If destinationOffset is Internal (UserSpace), the packet is set at a bank value of UserSpace, USBank, which comes in 4 types.  If destinationOffset is Internal (Core), an offset value from 0xFFFF F000 0000 is entered in CSRAdrsOff. If it is Internal (UserSpace), then an offset value from the heading address of each UserSpace is entered.	None
dataLength	None	dataLength
extended_tCode	None	extended_tCode
spd	spd	spd
ackSent	AckCode	None
-	BusID	ID(Bit19)

Received Packet	Register	ResponsePacket generation
data (or arg_value)	CSR Transaction DataH Register	None
data (or data_value)	CSR Transaction DataL Register	None

In case of extended\_tCode=Compare\_swap for tCode=LockRequest, a register value corresponding to the destination-Offset address is compared with arg\_value located in the packet. If there is coincidence, the CompOK bit of the CSR Transaction register is set up.

The ARxCSREnd bit is the Interrupt register is then set up. Waiting for the setting of RespGo, the following procedures are forwarded upon completion of this setting:

1) When AckCode of the CSR Transaction register is ack\_complete

When RespCode of the CSR Transaction register is resp\_complete, the contents of the CSR Transaction Data register are written in the Internal CSR register designated by destinationOffset. If RespCode of the CSR Transaction register is other than resp\_complete, nothing is written in the Internal CSR register. Since then, the RespGo bit is cleared. At this time point, Internal CSR processing has been finished and the subsequent reception of a Request packet is enabled at Internal CSR (Core) and Internal CSR (UserSpace).

2) When AckCode of the CSR Transaction register is ack\_pending and for a Write request

When RespCode of the CSR Transaction register is resp\_complete, the contents of the CSR Transaction Data register are written in the Internal CSR register designated by destinationOffset. If RespCode of the CSR Transaction register is other than resp\_complete, nothing is written in the Internal CSR register. ResponsePacketHeader is then generated from the received packet information and the resultant data are written in ATFi. In this case, however, the contents of RespCode in the rCodeCSR Transaction register are used.

At the next stage, a Transmit request (ATGo) is issued. Upon completion of transmission, the RespGo bit is cleared and ATxCSREnd of the Interrupt register is set up. At this time point, Internal CSR processing has been finished and the subsequent reception of a Request packet is enabled at Internal CSR (Core) and Internal CSR (UserSpace).

#### **7-1-1-4 When set at Internal CSR (NoInt)**

When a packet oriented to this space is stored in ARFi and a CRC error or a Length error occurs, Ack\_data\_error is returned. If there is no error, Ack is returned by WritePending located in the Packet Control register.

From this time point, Ack\_Busy is returned to all Request packets addressed to Internal CSR (Core) and Internal CSR (UserSpace). No reception is performed. However, reception is performed only for a packet (Request/Response) destined to a space defined by HostCSR or to an area of the External CSR space.

The MD8415 reads out a packet from ARFi and writes the contents of the Payload block in the Internal CSR register designated by destinationOffset. At this time point, Internal CSR processing has been finished and the subsequent reception of a Request packet is enabled at Internal CSR (Core) and Internal CSR (UserSpace).

#### **7-1-1-5 When set at Host CSR**

Ordinary processing is carried out in the same manner as for the External CSR space. A packet is stored in the ARF without performing control by tCode.

#### **7-1-1-6 When set at Ack\_Type\_error**

For a Read/Lock request, a response is returned by ack\_pending and a Response packet of resp\_type\_error is returned.

For the Write request, the following conditions are assumed:

- 1) When AckCode of the CSR Transaction register is ack\_complete

Data of ack\_type\_error are returned irrespective of the setting of the ackAddressErr bit in the Packet Control register.

- 2) When AckCode of the CSR Transaction register is ack\_pending and for a Write request

Data of resp\_type\_error are returned irrespective of the setting of the ackAddressErr bit in the Packet Control register.

The occurrence of ack\_address\_error/resp\_address\_error is caused when the ackAddressEr bit is "1" and the lower two bits of the address have been set up.

#### **7-1-1-7 Write/Lock request to Internal CSR (UserSpace)**

For a Write/Lock request, operation is the same as for Internal CSR of the CSR Config register for Internal CSR (core).

However, handling of the Ack code is different. If accessing is disabled to the related register, ack\_type\_error or resp\_type\_error is used.

#### **7-1-1-8 Command Reset packet**

When a Command Reset packet (0x00c) is received and the condition of CSRAtLib4=HostCSR has been defined, the data are stored in the ARF buffer and the CmdReset bit in the Interrupt register is set at "1". However, if there is no vacancy in the ARF buffer, processing is focused on CmdReset interruption only.

## 7-2 Access to External CSR

The External CSR suggests an access to an address other than the register space of Core and UserSpace. In this case, transmission is effected via the ATF buffer and the received packet is stored in the ARF buffer.

### 7-2-1 Access to the buffer

Access to the Asynchronous buffer (ATF/ARF) comes in two types; an access attempted from AsynchronousBus and HostBus.

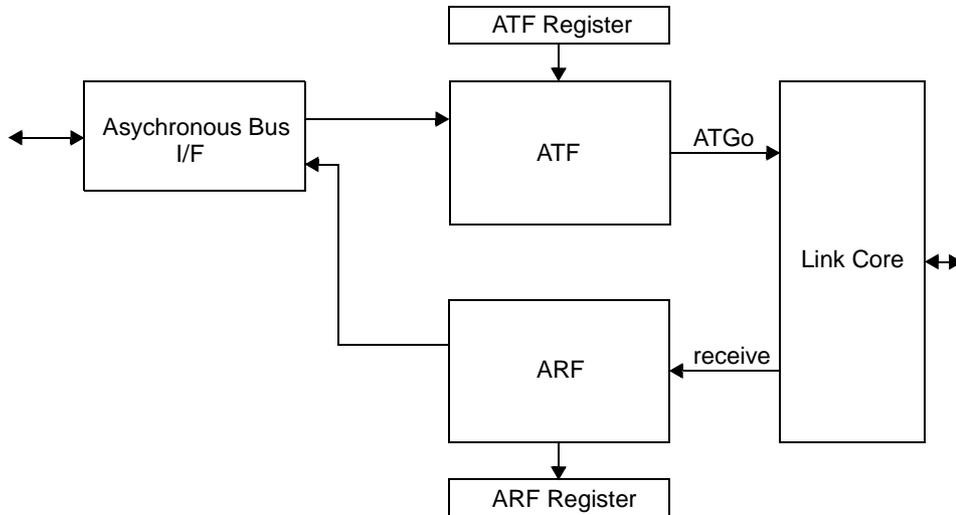


Figure 7-2-1 Asynchronous Buffer Access

As shown in the figure, this access cannot be made from AsynchronousBus and HostBus at the same time. Access route changeover is possible at the register at any time.

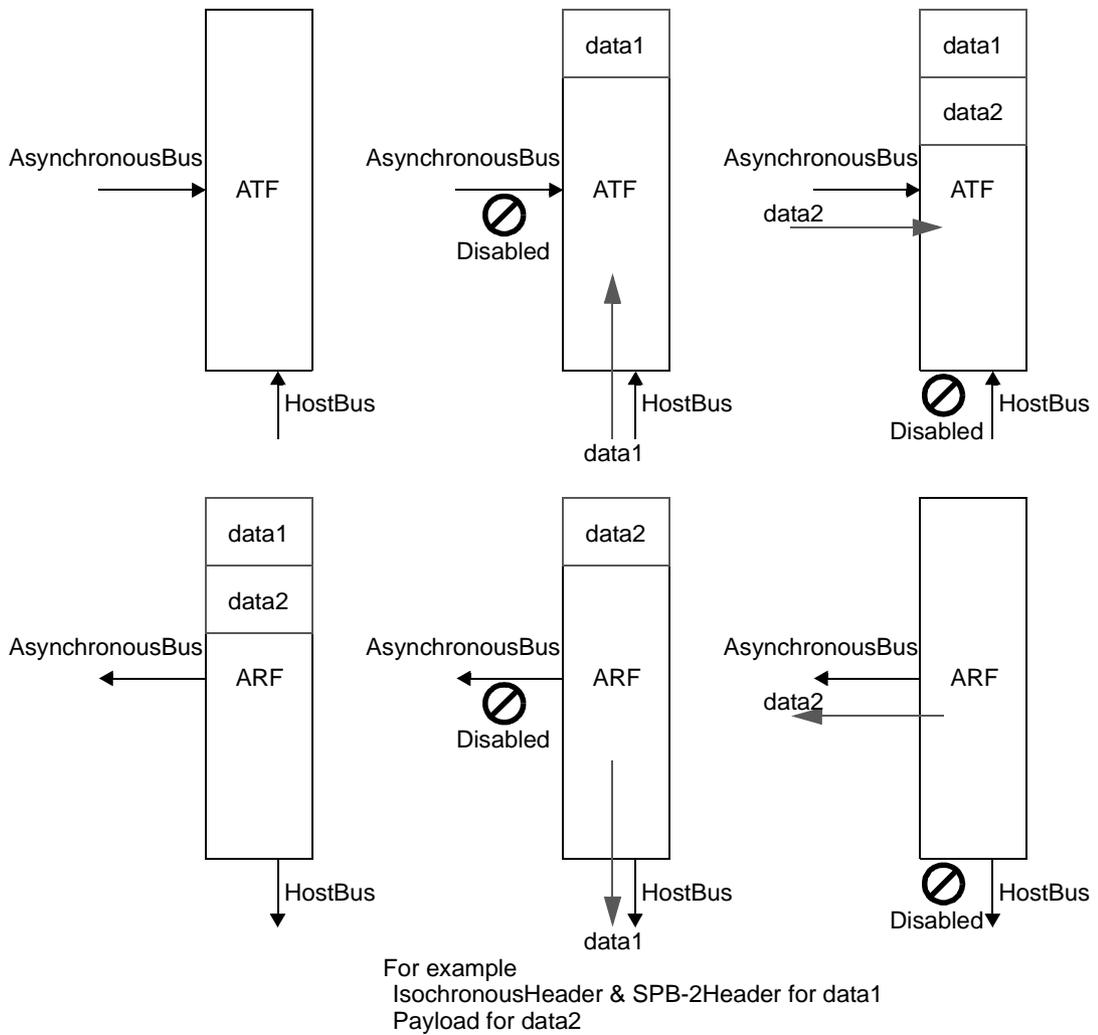


Figure 7-2-2 HostBus and AsyncBus

### 7-2-2 Access via ATF/ARF register

In an ordinary access to the host, the ATF buffer can be accessed by writing data in the ATF register. The data of the ARF buffer can be read from the ARF register. In this case, any access must be accomplished in the Quadlet unit. If the Little bit has been set, the ATF or ARF can be accessed after the rearrangement of bytes.

If ABTxEn of the Asynchronous Bus Control register has been set, access to the ATF register is disregarded. Similarly, if ABRxEn of the Asynchronous Bus Control register has been set, access to the ARF register is also disregarded.

### 7-2-3 Access via Async Bus

The interface is SynchronousFIFO-like and the following signals are used for control:

- ACLK (I) : AsynchronousBus clock (33MHz max.)
- ATEN# (O) :Transmission Enable signal (For assertion, Write is enabled in ATF by asserting the ATX# signal)
- ATX# (I) : Write Enable signal (For assertion, Write from ADATA to ATF is enabled with the rise of ACLK.)
- ARX# (I) : Read Enable signal (For assertion, Read from ADATA to ARF is enabled with the rise of ACLK.)
- ATEOP# (I) : End of Data instruction signal for transmission
- ARPKT# (O) : FIFO storage signal for reception
- ADATA (I/O) : Data bus

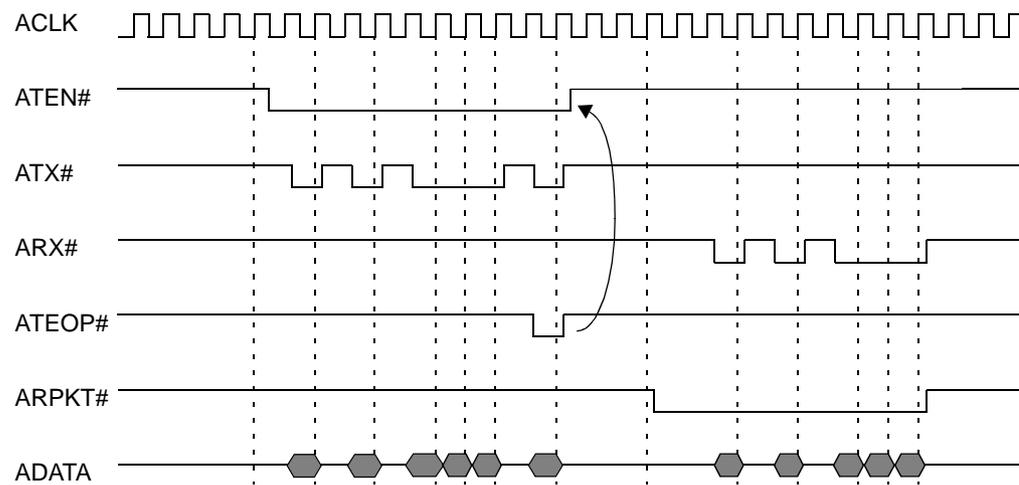


Figure 7-2-3 AsyncBus signal

### 7-2-4 Writing in ATF buffer

When the ABTxEn bit is set in the AsynchronousBus Control register, the ATEN# signal is asserted on the AsynchronousBus side. All of these signals are output in synchronization with ACLK.

In the external circuit connected with the Asynchronous bus, the data input from ADATA at the rise of ACLK are written in the ATF by asserting the ATX# signal that is synchronized with ACLK while ATEN# is asserted.

The writing cycle depends on the external circuit. While ATX# is asserted under the condition that ATEN# is asserted, ADATA is picked up always at the rise of ACLK and the acquired data are written in the ATF. Practically, the condition for data writing in the ATF is assumed with the ATF WrEn# signal shown in the figure. (This signal is an internal signal of the MD8415.) Generally, it is a signal produced from the ATEN# signal that is delayed for one clock period. When ATEOP# is asserted, data writing is not effected at the rise of the next ACLK.

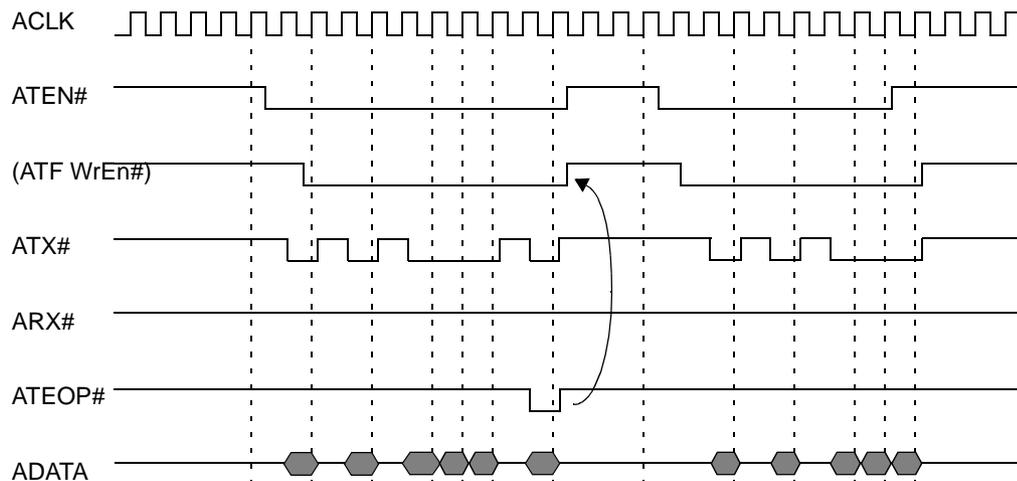


Figure 7-2-4 AsyncBus signal (ATF)

End of data transfer is defined by the two methods; it is determined by the external circuit and determined on the MD8415 side. When the end of data transfer is defined by the external circuit, ATEOP# is asserted from the external circuit. When ATEOP# is asserted, the MD8415 negates the ATEN# signal and sets up ABTxEnd of the Interrupt register. After ABTxEn in the AsynchronousBus Control register has been set up, the number of Assert actions is counted for the ATX# signal input from the external circuit. When the counted number amounts to the number preset in the ABTxLen field of the AsynchronousBus Control register, ATEN# is negated and ABTxEn is cleared. Since then, ABTxEnd of the Interrupt register is set up. If the preset value of the ABTxLen field is 0 in the AsynchronousBus Control register, there is no function of negating ATEN# forcibly as a result of counting. ATEN# is also negated when the ATF buffer is full while a data input is obtained from AsynchronousBus. If there is vacancy in the ATF buffer, ATEN# is asserted.

The transfer data are always handled in the Quadlet unit.

In conclusion, the condition for negating ATEN# and clearing ABTxEn is as follows:

The condition that the ATEN# pin is negated and ABTxEn is cleared:

- 1) ATEOP# assert -> ABTxEnd Interrupt occurs
- 2) End of transmission for ABTxLen only -> ABTxEnd Interrupt occurs
- 3) ABTxEn register clear -> ABTxEnd Interrupt occurs
- 4) ATF Full (In this case, ABTxEn is not cleared. Restarted when ATF Full is cleared.)
- 5) ResetATF set (ATF is cleared at the same time.)

### 7-2-5 Reading from the ARF buffer

When the Receive Enable bit (ABRxE<sub>n</sub>) is set in the AsynchronousBus Control register and data of more than one packet are stored in the ARF at that time, all of these signals are output in synchronization with ACLK.

In the external circuit connected with the Asynchronous bus, data are read out from ARF to ADATA at the rise of ACLK by asserting the ARX# signal that is synchronized with ACLK while ARPKT# is asserted.

The reading cycle depends on the external circuit. While ARX# is asserted under the condition that ARPKT# is asserted, ADATA is picked up always at the rise of ACLK in the MD8415 and the acquired data are written in the ATF. The ADATA bus is driven only when ARX# is kept asserted. The signal (ARF RdEn#) produced from the ARPKT# signal that is delayed for one clock period is used to determine the period of readout from the ARF.

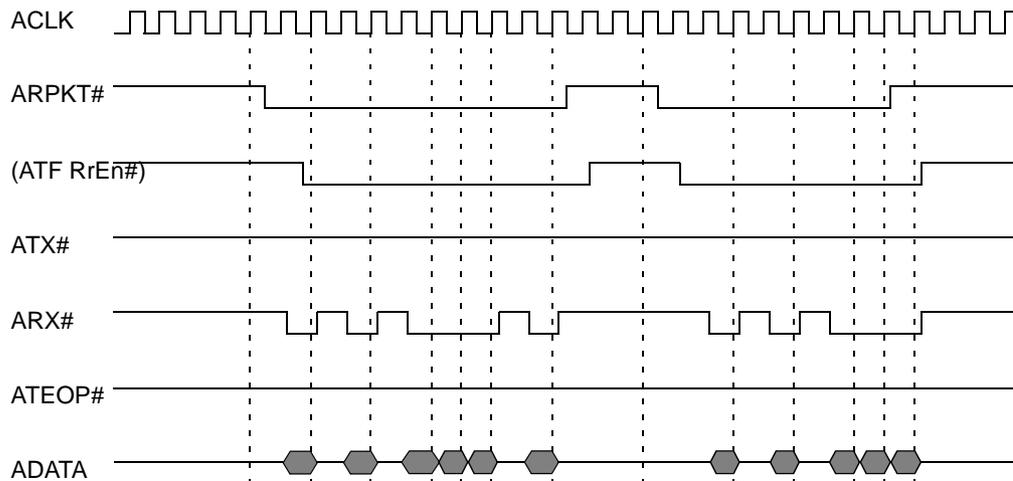


Figure 7-2-5 AsyncBus signal (ARF)

The ARPKT# signal is once negated for each packet.

After ABRxE<sub>n</sub> in the AsynchronousBus Control register has been set up, the amount of data is counted. When the counted amount attains the number of transfer actions preset in the ABRxLen field of the AsynchronousBus Control register, ARPKT# is negated and the ABRxE<sub>n</sub> bit is also cleared. Since then, ABRxEnd of the Interrupt register is set up.

ARPKT# is negated when the ARF buffer becomes empty while a data output is sent from AsynchronousBus. When data of one packet are stored in the ARF buffer, ARPKT# is asserted.

The transfer data are always handled in the Quadlet unit.

In conclusion, the condition for negating ARPKT# and clearing the ABRxE<sub>n</sub> bit is as described below.

The condition that the ARPKT# pin is negated and ABRxE<sub>n</sub> is cleared:

- 1) Reception complete only for ABRxLen -> ABRxEnd Interrupt occurs
- 3) ABRxE<sub>n</sub> register clear -> ABRxEnd Interrupt occurs
- 4) ARF Empty (Differ from ATF, clear also ABRxE<sub>n</sub>) -> ABRxEnd Interrupt occurs
- 5) Busreset and ARFBusReset=1 (Clear ARF simultaneously) -> ABRxEnd Interrupt occurs
- 6) ResetARF set(Clear ARF simultaneously)

If BusReset occurs and ARFBusReset of the Control register is set up, the ARPKT# signal is negated and the ARF buffer is cleared.

Operation is continued if ARFBusReset of the Control register is not set up.

### 7-3 Asynchronous Stream Transfer flow

This unit is provided with a function of asynchronous stream transfer.

#### 7-3-1 Transmission of Asynchronous Stream

During transmission, a packet by the name of AsynchronousStream Packet is transmitted under the condition that the identification data of AsynchronousStream Packet for one Quadlet are put before the Isochronous Packet format in the ATF buffer, as specified in the table below, and that ATGo is issued. In this case, a condition of tCode=0xA need be assumed. Practically, the data to be transmitted are those of the second Quadlet and thereafter. Similarly as for ordinary asynchronous transmission, arbitration is used in AsynchronousStream transmission and data are transmitted to destination other than IsochronousCycle.

The header CRC and the payload CRC are automatically generated in the MD8415 and added during transmission.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved													spd	reserved						tCode	reserved										
length													tag	channel				reserved		app											
Asynchronous data 1																															
Asynchronous data 2																															
other Asynchronous data																															
													padding (if necessary)																		

Table 7-3-1 Asynchronous Stream Transmit format

### 7-3-2 Reception of Asynchronous Stream

During reception, the MD8415 stores a packet in the ARF buffer when this packet has been identified as coinciding with the channel of Asynchronous Stream. The data format is specified below. In this case, a condition of tCode=0xA is assumed. The last AckSent indicates the condition of reception in the MD8415 and the same value as for asynchronous reception is stored. If the ARF is found to be full, ARFRej Interrupt occurs and no data are stored in the ARF. (This is not an interruption related to Iso.)

The receiving channel setting register is enabled when ASEn of each register is set under the two conditions of the Asynchronous Stream Receive Configuration register. The Asynchronous Stream Receive condition is not dependent on the setting of IsoDataMode, IsoBusModeL, and IsoBusModeR in the Control register.

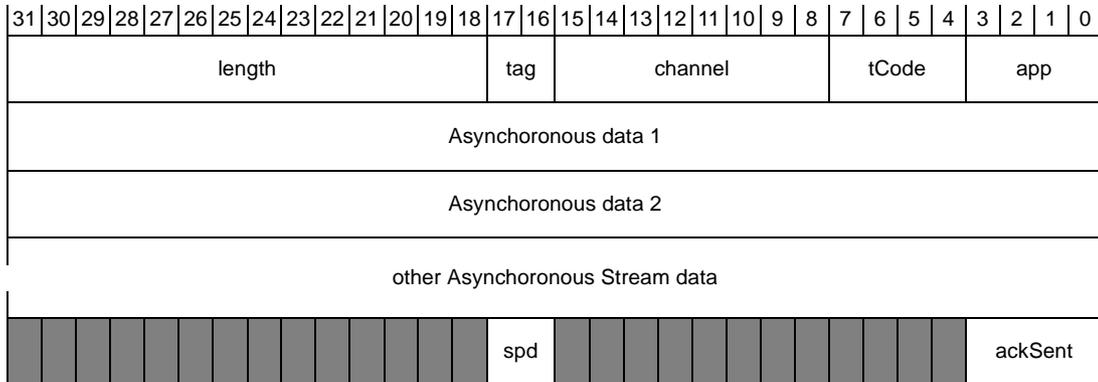


Table 7-3-2 Asynchronous Stream Receive format (Isochronous: normal)

### 7-4 BusyCtrl processing

According to the state of vacancy in the ARF buffer, it is not always possible to receive all the packets transmitted. Even in such a case, however, a Retry function is available for Asynchronous Transfer in order to enable the data to be transferred correctly. To define this retry function, it is possible to distinguish whether retry has been performed with the Ack code or data have been received normally. The MD8415 is provided with a function of being dependent on the status of the ARF buffer or receiving only one packet in the ARF buffer.

When the MD8415 uses the function of receiving only one packet in the ARF buffer, it returns Ack\_Busy to all packets coming successively. When packet processing is completed on the side of the MPU that controls the MD8415, the OnePacketProcEnd bit in the Diagnostic Status & Control register is set up for the MD8415, in order to enable reception at the ARF buffer. The MD8415 performs reception only if it is enabled.

This function is applicable only to the External CSR space. In regard to the Internal CSR core and the UserSpace space, reception is possible only for one packet in the ARFi buffer.

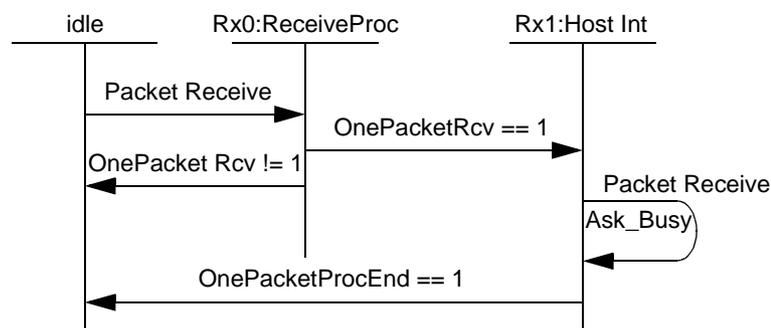


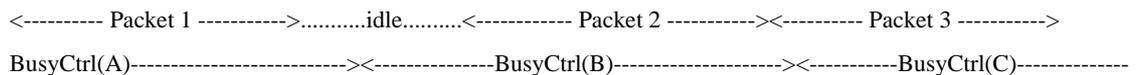
Figure 7-4-1 BusyCtrl Processing

Rx0:ReciveProc The received packet data are stored in FIFO.

Rx1:Host Int When this bit is set, Ack\_Busy is returned to all the received packets upon completion of reception.

Since then, reception is enabled when the OnePacketProcEnd bit is set by the MPU.

It is also possible to control the Ack code when reception is completed by the BusyCtrl field located in the Packet Control register. If there is no vacancy in the ARF buffer, the MD8415 generally returns Ack\_Busy as an Ack code without receiving the sent packet and gives a retry instruction to the sending party. Retry may be performed in the single phase or the dual phase. Either one is chosen according to the setting value for the BusyCtrl field. According to the condition of the setting value for the BusyCtrl field, Ack\_Busy can be returned irrespective of the state of vacancy in the ARF buffer. In this case, the BusyCtrl field decisive timing is established when the heading data have been received. In cases shown below, the setting value of BusyCtrl (A) is reflected on Packet 1, BusyCtrl (B) on Packet 2, and BusyCtrl (C) on Packet 3. When a packet is received after Packet 3, BusyCtrl (C) is then applied.



However, even though setting has been made to generate an output of the Ack code according to the amount of vacancy in the ARF buffer of the BusyCtrl field, the OnePacketRcv function becomes to be in higher preference if the OnePacketRcv bit has been set. This OnePacketRcv function is disabled, however, if the setting has been made to return Ack\_Busy irrespective of the condition of vacancy in the BusyCtrl field.

## 7-5 Retry Processing

Retry is carried out if it has been impossible for the receiving party to receive the transmitted data (not stored in FIFO at the receiving party) for a certain reason. The same data are transmitted again if the Ack code returns Busy after the completion of transmission.

The MD8415 has a function to accomplish retry processing by the use of hardware on the transmitter side (Outbound). If the returned Ack code is for Busy after the transmitting data have been stored in the ATF buffer and then transmitted. In this case, the contents of the ATF buffer are not cleared and retry transmission is carried out.

The maximum number of retry actions can be preset in the ATretry register. If the Ack code is found still Busy after the repetition of retry till the attainment of the maximum number, an announcement of Interrupt is sent to the host side, indicating this Busy condition.

Retry operation to be performed in accordance with IEEE1394 is based either on the dual phase retry protocol controlled by RetryA/B or on the single phase retry protocol controlled by RetryX. Both protocols are available for the MD8415. Which retry protocol should be used is determined by the setting made at the receiving party. The MD8415 is designed to choose any one of both by the use of the BusyStatus bit in the Packet Control register.

In addition, it is possible to change the packet transmission intervals for retry operation. In this case, retry transmission is conducted at the intervals defined by the RetryInterval field of the Packet Control register. For example, if the RetryInterval field is for "01h", the retry intervals become approx. 125 $\mu$ Sec.

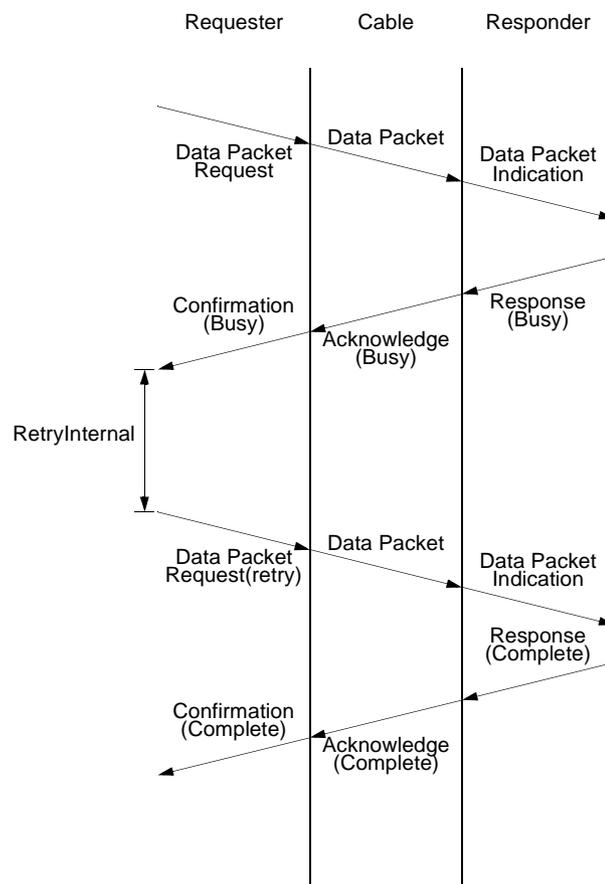


Figure 7-5-1 Retry Transaction

### 7-6 Asynchronous Transmission/Reception Access flow

In the MD8415, the following route is to be followed:

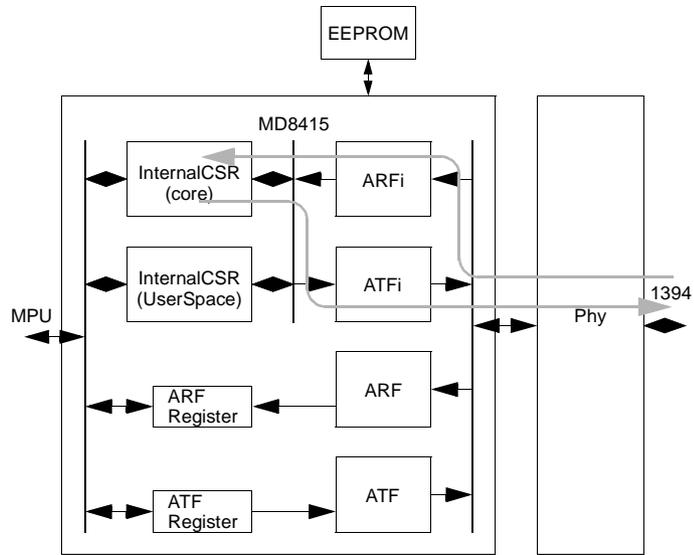


Figure 7-6-1 InternalCSR(core) Access

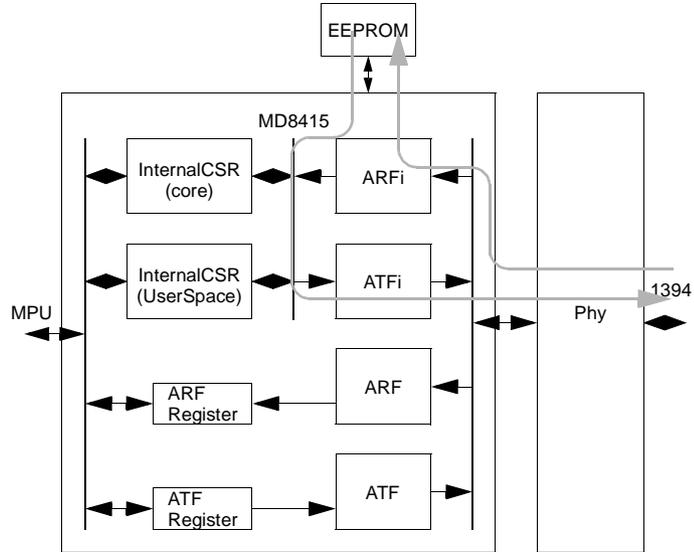


Figure 7-6-2 ConfigROM Access

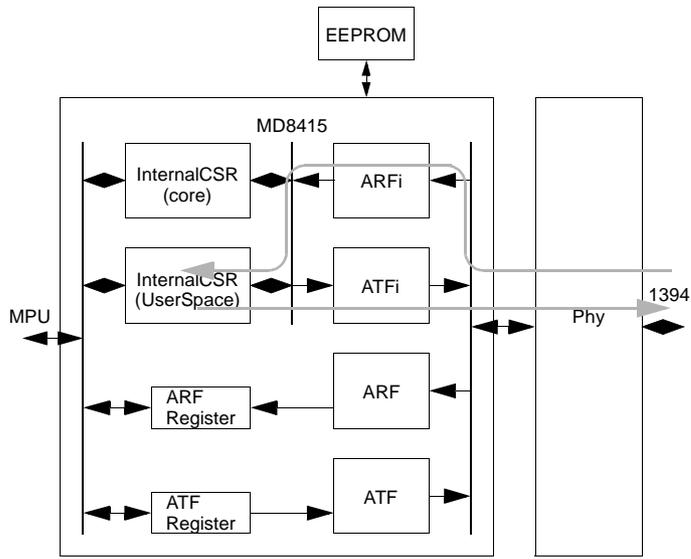


Figure 7-6-3 InternalCSR(UserSpace) Access

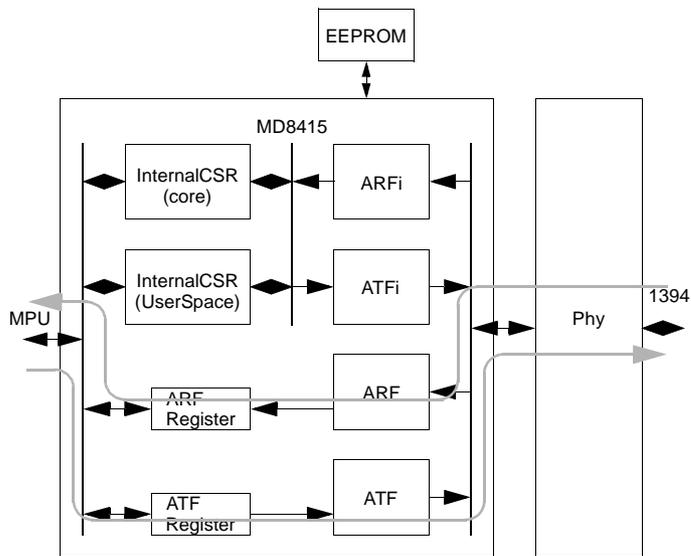
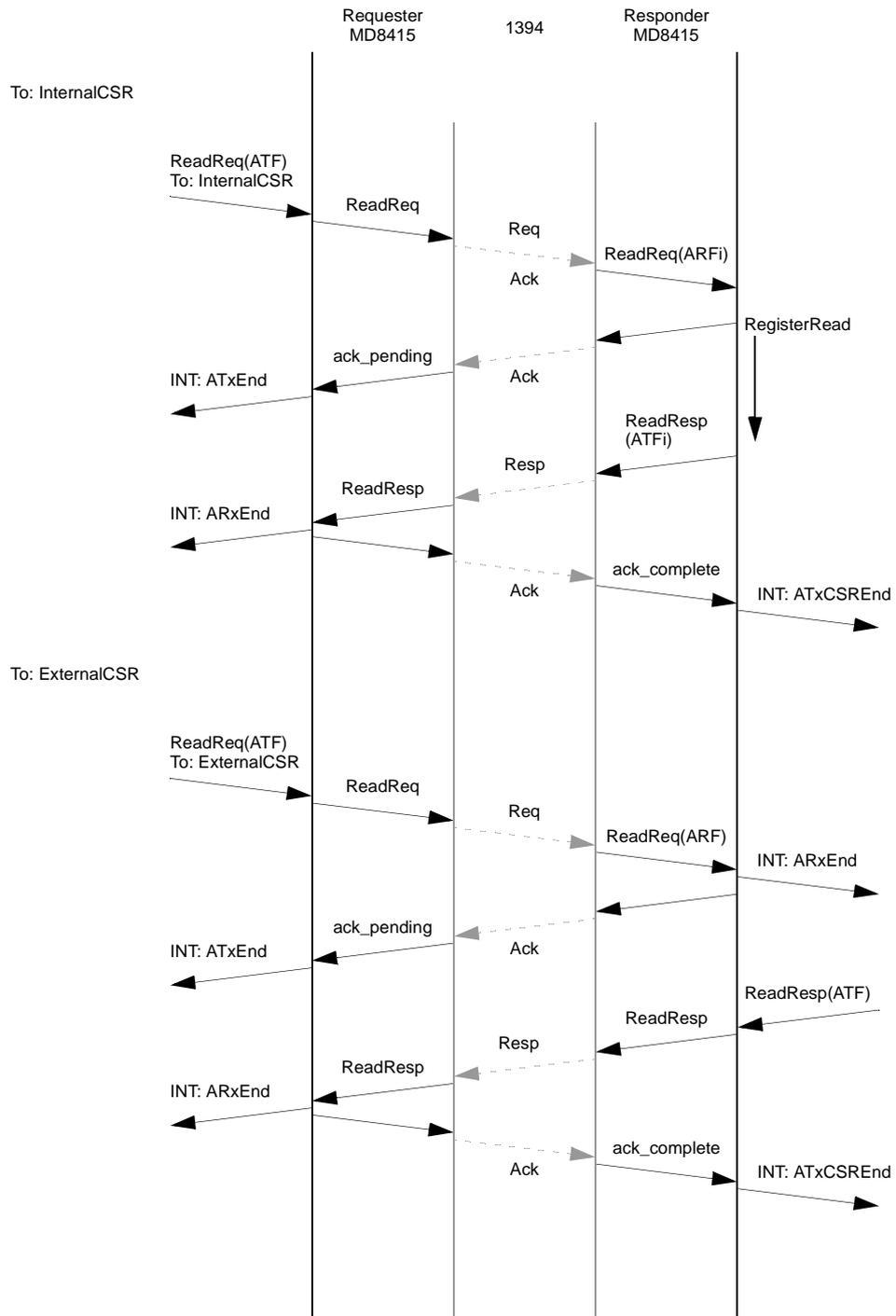


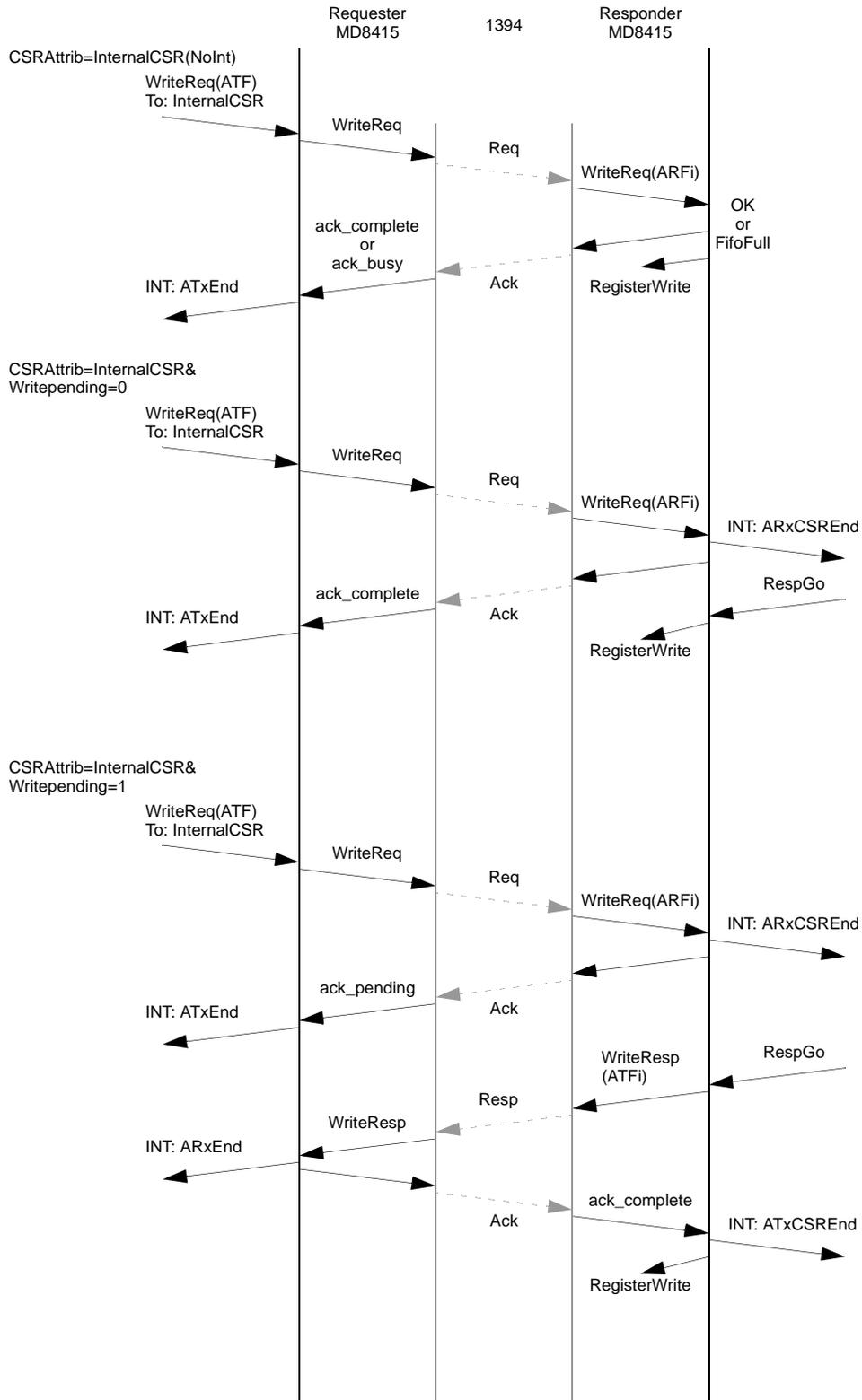
Figure 7-6-4 External CSR Access

The Asynchronous Transfer/Receive Access flow is as described below. Detailed operation is explained in the next chapter.

1) Read



2) Write/Lock





### 7-6-1 Transmission Access flow

To accomplish Asynchronous transmission, the following requirements have to be met:

- 1) TransmitEn=1b of Control Register(4h)
- 2) ReceiveEn=1b of Control Register(4h)
- 3) ROMChkEnd=1b of ControlRegister(4h)
- 4) LPSOn=1b of Control Register(4h) (However, it is assumed that the LPS signal of the MD8415 is connected to the LPS signal of the PHY.)
- 5) CycleTimerEn=1b of Control Register(4h)
- 6) NodeNumber!=3FFh of Node Identification Register(8h)
- 7) IDValid=1b of NodeIdentification Register(8h)
- 8) EnSnoop=0b= of PacketControlR egister(10h)
- 9) LinkReady=1b of DiagnosticStatusAndControlR egister(14h)

The transmission procedures are as described below.

- 1) According to the relevant format, the transmitting packet is written in the ATF buffer.
- 2) If the ATGo bit is "1b" in the ATGo register (A4h), this means that previous transmission is still in the middle of processing. Therefore, this transaction is suspended until "0b" is secured.
- 3) The ATGo bit is set at "1b" in the ATGo register (A4h). This condition enables the start of transmission.
- 4) When transmission is finished, ATxEnd is set in the Interrupt register (70h) and a condition of ATxEnd Interrupt is assumed.
- 5) To confirm the status of transmission, the successive register is read out.
  - AckStatus of DiagnosticStatusAndControl (14h)
  - ATAck of DiagnosticStatusAndControl (14h)
- 6) If AckStatus is other than 00b, this is a result of abnormal transmission. Processing for abnormality is required.
- 7) If AckStatus is 00b, processing can be forwarded according to the Ack code located in ATAck.

### 7-6-2 Reception Access flow

The condition for Asynchronous reception is the same as for transmission. Reception is performed under the condition of the NODE\_IDS register in the Core CSR space. The received data are stored in the ARF buffer. However, no packet is stored in the ARF buffer if a CRC error or a Length error occurs during reception.

The reception procedures are as follows:

- 1) When reception of all data is completed for a packet that meets the requirements of Asynchronous reception, ARxEnd is set in the Interrupt register (70h) and a condition of ARxEnd Interrupt is assumed.
- 2) Data of the AsynchronousHeader block are read from the ARF buffer. By this operation, packet type (tCode) and packet length (Length) are identified and data of the Payload block are read out.
- 3) The last one Quadlet is read out. These data involve the speed (spd) of the received packet and the receiving condition (ackSent). In the MD8415, abnormal packets of CRC error, Length error, etc., are not stored in the ARF buffer. Therefore, the ackSent value becomes ack\_complete or ack\_pending at all times.

### 7-7 BusNumber processing for Asynchronous Packet transmission

Generally, the SourceID for transmission is not included in the transmission packet format. It is set at the value preset in the NODE\_IDS register of the Core CSR located in the MD8415.

In addition to this function, the MD8415 has another function to set up "0x3FF" defined as LocalBus, without using BusNumber that is defined in this NODE\_IDS register.

To select these two functions, the following setting is to be made:

- 1) When using BusNumber of the NODE\_IDS register as the BusNumber for the SourceID: The BusID bit of the first Quadlet is set at "0b" for the Asynchronous transmission packet format.
- 2) When using "0x3FF" as the BusNumber for the SourceID: The BusID bit of the first Quadlet is set at "1b" for the Asynchronous transmission packet format.

An example of formats is shown below on the cable of a packet transmitted with the setting of the BusID bit.

[Condition]

NODE\_IDS register= 0x0042(busNumber=1, NodeNumber=2)

Transmission packet : WriteRequest

- 1) BusID= 0b

Format for storage in the ATF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	spd	tLabel				rt	tCode			priority							
destinationID															destinationOffset_H																
destinationOffset_L																															
quadlet data																															

Format on the cable

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
destinationID															tLabel				rt	tCode			priority								
BusNumber=1										Node=2					destinationOffset_H																
destinationOffset_L																															
quadlet data																															

1)BusID=1b

Format for storage in the ATF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	spd	tLabel				rt	tCode				priority							
destinationID														destinationOffset_H																	
destinationOffset_L																															
quadlet data																															

Format on the cable

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
destinationID														tLabel				rt	tCode				priority								
BusNumber=0x3FF										Node=2				destinationOffset_H																	
destinationOffset_L																															
quadlet data																															

## 7-8 Processing for abnormality during Asynchronous Packet transmission

When "1" is written in ATGo, the MD8415 identifies it as the header of the Asynchronous packet according to the first one-Quadlet data written in the ATF buffer. It determines the packet format, seeing the field that falls on tCode of these one-Quadlet data. If the field value corresponding to this tCode is abnormal (Asynchronous packet, and other than "0xe0), the TCodeErr bit is set in the Interrupt register and transmitting operation is suspended. At that time, the ATBusy bit and the ATGo bit are left set at "1". For restoration, it is necessary to clear the ATF buffer and reset the transmitting operation. The resetting flow is shown below.

- 1) Write "0" in the ATGo bit.
- 2) Write "1" in ResetATF and ResetTx of the Reset register.

## 8 Internal Access to the Internal CSR

When gaining access to the Internal CSR in the MD8415 from the host, such operation is carried out via the Internal CSR Access register. The same operation is performed for both Internal CSR (Core) and Internal (UserSpace).

### 8-1 Write Access

When the Write bit is set up in the Internal CSR Access register, the contents of the CSRData field in the Internal CSR Access Data register are written in the Internal CSR register of the MD8415. When the CSRCore bit of the Internal CSR Access register has been set up, the Internal CSR register being written in has an address that is specified by the CSRAdrOff field of the CSR Access register in the area of Internal CSR (Core). If the CSRCore bit has not been set up, the register to be used is the offset register designated by the CSRAdrOff field from the head of the Internal CSR (UserSpace) area specified by the USBank field.

Access from the host is also enabled even though Write is disabled in the CSR UserSpace Config register where register Attribute of the Internal CSR (UserSpace) is defined.

When writing is finished, the Write bit is cleared in the Internal CSR Access register.

Write Access to the ConfigROM area (EEPROM area) of the Internal CSR (Core) is not supported.

When accessing to the area not supported, CSRAccErr of the CSR Access Data register is set and the Write bit of the Internal CSR Access register is cleared.

### 8-2 Read Access

When the Read bit is set up in the Internal CSR Access register, the contents of the Internal CSR register of the MD8415 are written in the CSRData field of the Internal CSR Access Data register. When the CSRCore bit of the Internal CSR Access register has been set up, the Internal CSR register being read out has an address that is specified by the CSRAdrOff field of the CSR Access register in the area of Internal CSR (Core). If the CSRCore bit has not been set up, the register to be used is the offset register designated by the CSRAdrOff field from the head of the Internal CSR (UserSpace) area specified by the USBank field.

When writing is finished in the CSRData field of the CSR Access Data register, the Read bit is cleared in the Internal CSR Access register and the CSRRcv bit of the CSR Access Data register is set up.

Read Access to the ConfigROM area (EEPROM area) of the Internal CSR (Core) is supported only in the Quadlet unit. Access to the Internal CSR (UserSpace) is supported only for Length=8 of Quadlet and Block. When accessing to the area not supported, CSRAccErr of the CSR Access Data register is set and the Read bit of the Internal CSR Access register is cleared.

### 8-3 Lock Access

Lock Access mentioned here is supported only for Compare\_swap. When the Lock bit is set up in the Internal CSR Access register, the contents of the Internal CSR register of the MD8415 are compared with those of the CSRData field of the Internal CSR Access Data register. When coincidence is perceived, the contents of the CSRLockData field of the Internal CSR Access Lock Data register are written in. When the CSRCore bit of the Internal CSR Access register has been set up, the Internal CSR register being written in has an address that is specified by the CSRAdrOff field of the CSR Access register in the area of Internal CSR (Core). If the CSRCore bit has not been set up, the register to be used is the offset register designated by the CSRAdrOff field from the head of the Internal CSR (UserSpace) area specified by the USBank field.

If no coincidence is perceived, or in the case of Access to the area that is not supported, writing is not performed but CSRAccErr of the CSR Access Data register is set up.

Since then, the contents of the Internal CSR Access register before rewriting are written in the CSRData field of the CSR Access Data register. When reading is finished, the Lock bit is cleared in the Internal CSR Access register and the CSRRcv bit of the CSR Access Data register is set up.

### 8-4 Access Flow

An ordinary access is started with the first request when there is an access (Read/Write/Lock) from "1394" simultaneously when setting has been made in the Internal CSR Access register. This corresponds to the case when Read Access is attempted from "1394" side. In the case of Write/Lock, different operation is performed according to whether ARxCSRInt Interrupt holds or not. If there is no ARxCSRInt Interrupt, the same processing as for Read is carried out. If there is ARxCSRInt Interrupt, however, Access from the host is interrupted for reasons of Busy. This interruption can be confirmed because the Busy bit of the Internal CSR Access register is set up. Therefore, the host is required to make another access via the Internal CSR Access register again.

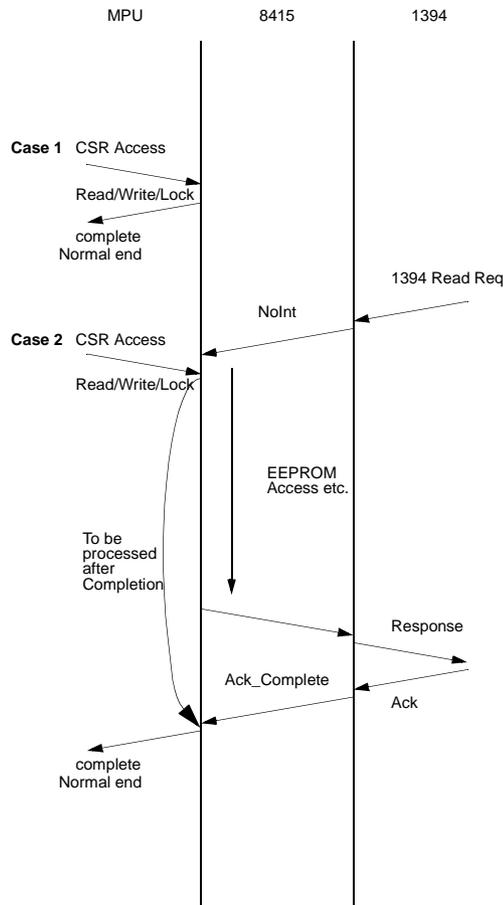


Figure 8-4-1 Access Flow when Access from "1394" is for Read

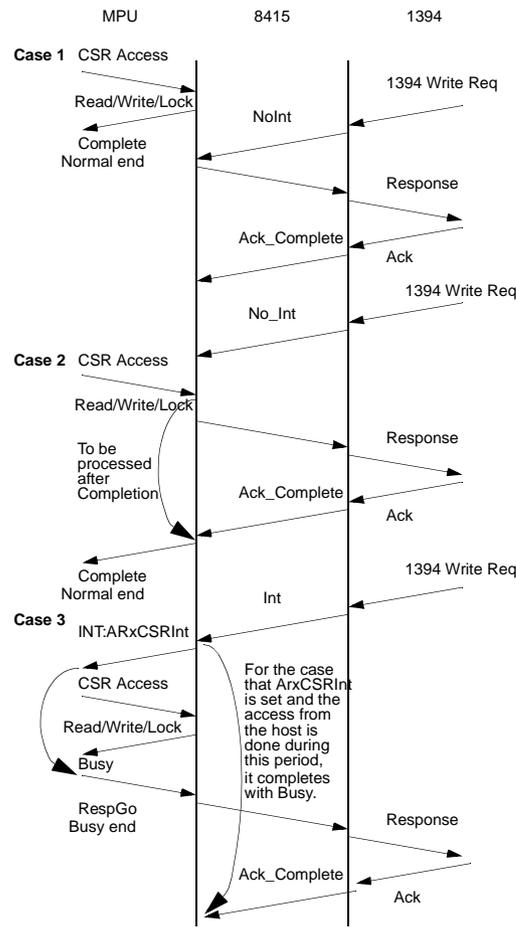


Figure 8-4-2 Access Flow when Access from "1394" is for Write/Lock

## 9 Definition of the Internal CSR

### 9-1 Internal CSR (UserSpace)

The space of the Internal CSR (UserSpace) can be divided into a maximum of four sections (banks). The size is 128 bytes in all. The address specification in the CSR space of each bank is defined by the CSR UserSpace Config register. "x" denotes the bank value.

The US<sub>x</sub>\_Size field is used to specify size of each bank (in the Quadlet unit). When "0" is specified, no register of Internal CSR (UserSpace) is used.

The US<sub>x</sub>\_Write bit is used to define that Write is enabled in the register in this bank.

The US<sub>x</sub>\_Read bit is used to define that Read is enabled in the register in this bank.

The US<sub>x</sub>\_Lock bit is used to define that Lock is enabled (compare\_swap only) in the register in this bank.

The US<sub>x</sub>\_Adrs field is used to define the head address of this bank. The lower 28 bits are specified in the CSR space the remaining upper 20 bits are fixed at 0x000F FFFF. This suggests that UserSpace can be arranged in the area from 0xFFFF F000 0000 to 0xFFFF FFFF FFFF of destination\_offset in the CSR space.

Access to this area is supported only for Length=8 of Quadlet and Block. In case of a Block request and when Length exceeds 8, the condition is regarded as type\_error. If the space defined by Internal CSR (UserSpace) overlaps Internal CSR (Core), the latter is in higher preference.

### 9-2 Internal CSR (core)

Operation is described here when an access is made via "1394". There is an additional description about an access attempted via the host register.

write1394 : Write via 1394

writeHost : Write via CSR Host Access register

lock1394 : Lock via 1394

LockHost : Lock via CSR Host Access register

#### 9-2-1 STATE

This is the reality of the register accessed via the STATE\_CLEAR or STATE\_SET register in the CSR space. ReadRequest to the STATE\_CLEAR and STATE\_SET registers is the contents of this register. Regarding WriteRequest, data writing in the STATE\_CLEAR register denotes that the register bit of this reality is cleared. Data writing in the STATE\_SET register means that the register bit of this reality is set up.

7	6	5	4	3	2	1	0
lost	dreq	r	elog	atn	off	state	
15	14	13	12	11	10	9	8
gone	r				abdicate	linkoff	cmstr
23	22	21	20	19	18	17	16
unit_depend							
31	30	29	28	27	26	25	24
unit_depend							

	initial	bus reset	cmd reset	Writing conditions
<b>unit_depend (16)</b>	zeros			Host Access only
<b>bus_depend (8)</b>				
<b>gone (1)</b>	zero	one	one	The following conditions are required: Set at "1" by STATE_SET, or at "0" by STATE_CLEAR. Host Access enabled.
<b>r (4)</b>	zero			Host Access only
<b>abdicate (1)</b>	zero	zero	zero	The following conditions are required: Set at "1" by STATE_SET, or at "0" by STATE_CLEAR. Host Access enabled.
<b>linkoff (1)</b>	zero	unchange	unchange	The following conditions are required: Set at "1" by STATE_SET, or at "0" by STATE_CLEAR. LinkOff bit of the Interrupt register when set at "1" by STATE_SET. Host Access enabled.
<b>cmstr (1)</b>	zero	unchange	unchange	The following conditions are required: When CycleMaster of the Control register is "1" in the own root, this bit is set up. It is set at "0" if there is no own root available. When set at "1" by STATE_SET or cleared to "0," cmstr of the Interrupt register is set up. Set at "0" by STATE_CLEAR. Host Access enabled.
<b>lost (1)</b>	zero			Host Access only
<b>dreq (1)</b>	zero			Host Access only
<b>r (1)</b>	zero			Host Access only
<b>elog (1)</b>	zero			Host Access only
<b>atn (1)</b>	zero			Host Access only
<b>off (1)</b>	zero			Host Access only
<b>state (2)</b>	zero			Host Access disabled.

State is "1" until analysis is completed from BusReset to SelfID. Others are set at "0".

For all bits, Write is enabled from the Internal CSR Access register.

### 9-2-2 STATE\_CLEAR

CSR address 0xFFFF F000 0000h

Attribute Quadlet Read/Write only. For Lock, Ack\_type\_error is returned.

7	6	5	4	3	2	1	0
lost	dreq	r	elog	atn	off	state	
15	14	13	12	11	10	9	8
gone	r				abdicate	linkoff	cmstr
23	22	21	20	19	18	17	16
unit_depend							
31	30	29	28	27	26	25	24
unit_depend							

	initial	bus reset	cmd reset	read	write1394	writeHost
<b>unit_depend (16)</b>	zeros	zeros	zeros	last write	ignored	*1
<b>bus_depend (8)</b>						
<b>gone (1)</b>	zero	one	one	last write	*1	*1
<b>r (4)</b>	zero	zero	zero	zero	ignored	*1
<b>abdicate (1)</b>	zero	zero	zero	last write	*1	*1
<b>linkoff (1)</b>	zero	unchange	unchange	last write	*1	*1
<b>cmstr (1)</b>	zero	unchange	unchange*2	last write	*1	*1
<b>lost (1)</b>	zero	zero	zero	last write	ignored	*1
<b>dreq (1)</b>	zero	zero	zero	last write	ignored	*1
<b>r (1)</b>	zero	zero	zero	zero	ignored	*1
<b>elog (1)</b>	zero	zero	zero	last write	ignored	*1
<b>atn (1)</b>	zero	zero	zero	last write	ignored	*1
<b>off (1)</b>	zero	zero	zero	last write	ignored	*1
<b>state (2)</b>	zero	zero	zero	last write	ignored	ignored

\*1: When the field is "1" in the WriteRequest Receive packet, this field of the STATE register is cleared when "1" is written in. Nothing occurs when "0" is written in.

For Read, the contents of the STATE register are read out. This register involves nothing of reality, but the reality stays in the STATE register specified in 9-2-1. Writing operation mentioned here means that a bit of the real STATE register is cleared.

### 9-2-3 STATE\_SET

CSR address 0xFFFF F000 0004h

Attribute Quadlet Write/Read only. For Lock, Ack\_type\_error is returned.

7	6	5	4	3	2	1	0
lost	dreq	r	elog	atn	off	state	
15	14	13	12	11	10	9	8
gone	r				abdicate	linkoff	cmstr
23	22	21	20	19	18	17	16
unit_depend							
31	30	29	28	27	26	25	24
unit_depend							

	initial	bus reset	cmd reset	read	write1394	writeHost
<b>unit_depend (16)</b>	zeros	zeros	zeros	last write	gnored	*1
<b>bus_depend (8)</b>						
<b>gone (1)</b>	zero	zero	zero	last write	*1	*1
<b>r (4)</b>	zeros	zeros	zeros	zeros	ignored	*1
<b>abdicate (1)</b>	zero	zero	zero	last write	*1	*1
<b>linkoff (1)</b>	zero	zero	zero	last write	*1	*1
<b>cmstr (1)</b>	zero	zero	zero	last write	*1	*1
<b>lost (1)</b>	zero	zero	zero	last write	ignored	*1
<b>dreq (1)</b>	zero	zero	zero	last write	ignored	*1
<b>r (1)</b>	zero	zero	zero	zero	ignored	*1
<b>elog (1)</b>	zero	zero	zero	last write	ignored	*1
<b>atn (1)</b>	zero	zero	zero	last write	ignored	*1
<b>off (1)</b>	zero	zero	zero	last write	ignored	*1
<b>state (2)</b>	zeros	zeros	zeros	zeros	ignored	ignored

\*1: When the field is "1" in the WriteRequest Receive packet, this field of the STATE register is set up. Nothing occurs when "0" is written in. This register involves nothing of reality, but the reality stays in the STATE register specified in 9-2-1. Writing operation mentioned here means that a bit of the real STATE register is setup.

### 9-2-4 NODE\_IDS

CSR address 08h

Attribute Quadlet Read/Write only. For Lock, Ack\_type\_error is returned.

7	6	5	4	3	2	1	0
bus_dependent							
15	14	13	12	11	10	9	8
bus_dependent							
23	22	21	20	19	18	17	16
bus_id		offset_id					
31	30	29	28	27	26	25	24
bus_id							

	initial	bus reset	cmd reset	read	write1394	writeHost
<b>bus_id (10)</b>	ones	unchanged	unchanged	last update	stored	stored
Value of Node Identification Register is reflected.						
<b>offset_id (6)</b>	physical_ID	unchanged	unchanged	last update	ignored	ignored
Value of Node Identification Register is reflected.						
<b>bus_dependent (16)</b>	zeros	zeros	zeros	last update	ignored	stored

For the bits of bus\_dependent and bus\_id, writing is enabled from the Internal CSR Access register.

### 9-2-5 RESET\_START

CSR address 0xFFFF F000 000Ch

Attribute Quadlet Write only. For Read/Lock, Ack\_type\_error is returned.

If there is a Write request to this register, CmdReset Interrupt takes place. In addition, the value of internal CSR is modified. There is no reality of the register.

**9-2-6 SPLIT\_TIMEOUT**

CSR address 0xFFFF F000 0018h,1Ch

Attribute Quadlet Read/Write only. For Lock, Ack\_type\_error is returned.

7	6	5	4	3	2	1	0
integer of a second					integer of a second		
15	14	13	12	11	10	9	8
integer of a second							
23	22	21	20	19	18	17	16
integer of a second							
31	30	29	28	27	26	25	24
integer of a second							

7	6	5	4	3	2	1	0
fractions of second							
15	14	13	12	11	10	9	8
fractions of second							
23	22	21	20	19	18	17	16
fractions of second				fractions of second			
31	30	29	28	27	26	25	24
fractions of second							

	initial	bus reset	cmd reset	read	write1394	writeHost
<b>SPLIT_TIMEOUT_HI. integers of a second (29)</b>	zeros	zeros	zeros	zeros	ignored	ignored
<b>SPLIT_TIMEOUT_HI. integers of a second (3)</b>	zeros	unchanged	unchanged	last write	stored	stored
<b>SPLIT_TIMEOUT_LO. fractions of a second (13)</b>	800	unchanged	unchanged	last write	stored	stored
<b>SPLIT_TIMEOUT_LO. fractions of a second (19)</b>	zeros	zeros	zeros	zeros	ignored	ignored

For all bits, Write is enabled from the Internal CSR Access register.

**9-2-7 CYCLE\_TIME**

CSR address 0xFFFF F000 0200h

Attribute Quadlet Read/Write only. For Lock, Ack\_type\_error is returned.

7	6	5	4	3	2	1	0
cycle_offset							
15	14	13	12	11	10	9	8
cycle_count				cycle_offset			
23	22	21	20	19	18	17	16
cycle_count							
31	30	29	28	27	26	25	24
second_count							cycle_count

	initial	bus reset	cmd reset	read	write1394	writeHost
<b>second_count (7)</b>	zeros	unchanged	unchanged	last update	stored*1	stored*1
<b>cycle_count (13)</b>	zeros	unchanged	unchanged	last update	stored*1	stored*1
<b>cycle_offset (12)</b>	zeros	unchanged	unchanged	last update	stored*1	stored*1

The same value of the Cycle Time register is reflected. If written here, on the contrary, it is reflected on the Cycle Time register.

\*1: The Link CtleTimer value is reflected.

For all bits, Write is enabled from the Internal CSR Access register.

**9-2-8 BUS\_TIME**

CSR address 0xFFFF F000 0204h

Attribute Quadlet Read/Write only. For Lock, Ack\_type\_error is returned.

7	6	5	4	3	2	1	0
second_count_hi	second_count_lo						
15	14	13	12	11	10	9	8
second_count_hi							
23	22	21	20	19	18	17	16
second_count_hi							
31	30	29	28	27	26	25	24
second_count_hi							

	initial	bus reset	cmd reset	read	write1394	writeHost
<b>second_count_hi (25)</b>	<b>zeros</b>	unchanged	unchanged	last write	stored	stored
<b>second_count_lo (7)</b>	<b>zeros</b>	unchanged	unchanged	last update	ignored	ignored

For all bits, Write is enabled from the Internal CSR Access register.

**9-2-9 BUSY\_TIMEOUT**

CSR address 0xFFFF F000 0210h

Attribute Quadlet Read/Write only. For Lock, Ack\_type\_error is returned.

7	6	5	4	3	2	1	0
r				retry_limit			
15	14	13	12	11	10	9	8
cycle_limit				r			
23	22	21	20	19	18	17	16
cycle_limit							
31	30	29	28	27	26	25	24
r				second_limit			cycle_limit

	initial	bus reset	cmd reset	read	write1394	writeHost
<b>reserved (4)</b>	zeros	zeros	zeros	zeros	ignored	ignored
<b>second_limit (3)</b>	zeros	unchaged	unchanged	last write	stored	stored
<b>cycle_limit (13)</b>	200	unchaged	unchanged	last write	stored	stored
<b>reserved (8)</b>	zeros	zeros	zeros	zeros	ignored	ignored
<b>retry_limit (4)</b>	zeros	unchaged	unchanged	last write	stored	stored

The retry setting value is stored. Based on this value, the retry function is activated.

For all bits, Write is enabled from the Internal CSR Access register. The Reserved bit has no reality.

**9-2-10 BUS\_MANAGER\_ID**

CSR address 0xFFFF F000 021Ch

Attribute Quadlet Read/Write only. For Write, Ack\_type\_error is returned.

7	6	5	4	3	2	1	0
r		bus_mngr_id					
15	14	13	12	11	10	9	8
r							
23	22	21	20	19	18	17	16
r							
31	30	29	28	27	26	25	24
r							

	initial	bus reset	cmd reset	read	lock1394	lockHost
<b>reserved (26)</b>	zeros	zeros	zeros	zeros	ignored	ignored
<b>bus_mngr_id (6)</b>	3Fh	3Fh	unchanged	last Lock	write*1	write*1

\*1: Lock only is enabled for writing in this register. Writing is conducted only if Lock is successful.

Bus\_mngr\_id is enabled only if there is a Lock request from the Internal CSR Access register. The Reserved bit has no reality.

**9-2-11 BANDWIDTH\_AVAILABLE**

CSR address 0xFFFF F000 0220h

Attribute Quadlet Read/Lock only. For Write, Ack\_type\_error is returned.

7	6	5	4	3	2	1	0
bw_remaining							
15	14	13	12	11	10	9	8
r				bw_remaining			
23	22	21	20	19	18	17	16
r							
31	30	29	28	27	26	25	24
r							

	initial	bus reset	cmd reset	read	lock1394	lockHost
<b>reserved (19)</b>	zeros	zeros	zeros	zeros	ignored	ignored
<b>bw_remaining (13)</b>	4915	4915	unchanged	last Lock	write*1	write*1

\*1: Lock only is enabled for writing in this register. Writing is conducted only if Lock is successful.

The bw\_remaining bit is enabled only if there is a Lock request from the Internal CSR Access register. The Reserved bit has no reality.

**9-2-12 CHANNEL\_AVAILABLE**

CSR address 0xFFFF F000 0224h, 228h

Attribute Quadlet Read/Lock only. For Write, Ack\_type\_error is returned.

7	6	5	4	3	2	1	0
CHANNEL_AVAILABLE_hi							
15	14	13	12	11	10	9	8
CHANNEL_AVAILABLE_hi							
23	22	21	20	19	18	17	16
CHANNEL_AVAILABLE_hi							
31	30	29	28	27	26	25	24
CHANNEL_AVAILABLE_hi							

7	6	5	4	3	2	1	0
CHANNEL_AVAILABLE_lo							
15	14	13	12	11	10	9	8
CHANNEL_AVAILABLE_lo							
23	22	21	20	19	18	17	16
CHANNEL_AVAILABLE_lo							
31	30	29	28	27	26	25	24
CHANNEL_AVAILABLE_lo							

	initial	bus reset	cmd reset	read	lock1394	lockHost
<b>CHANNEL_AVAILABLE_hi (32)</b>	ones	ones	unchanged	last Lock	write*1	write*1
<b>CHANNEL_AVAILABLE_lo (32)</b>	ones	ones	unchanged	last Lock	write*1	write*1

\*1: Lock only is enabled for writing in this register. Writing is conducted only if Lock is successful.

The CHANNEL\_AVAILABLE\_Hi,Lo bit is enabled only for a Lock request from the Internal CSR Access register.

**9-2-13 Configuration ROM**

CSR address 0xFFFF F000 0400h~07FFh

Attribute Quadlet Block/Read only. For Write/Lock, Ack\_type\_error is returned.

To a Read request to this area, a response is sent by reading out the size data designated from the EEPROM. Therefore, this space is mapped in the EEPROM.

### 10 Isochronous Transmission/Reception

The Isochronous bus comes in two types of Master and Slave. This setting is made with the aid of the MASTER# terminal. In the MD8415, different operation is performed according to the modes described below.

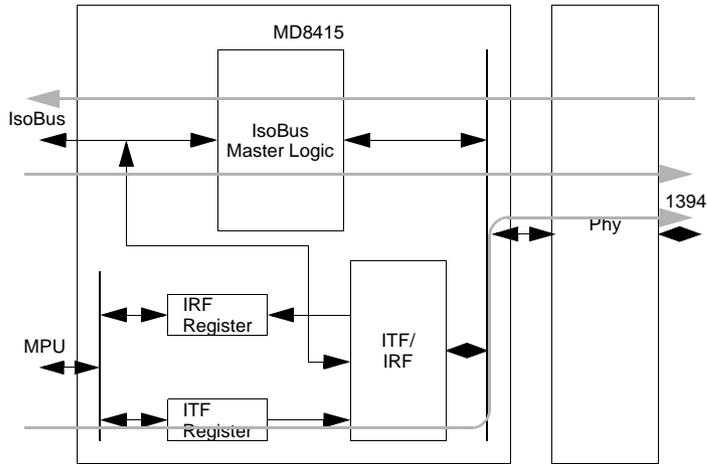


Figure 10-0-1 For Master Mode, ITF

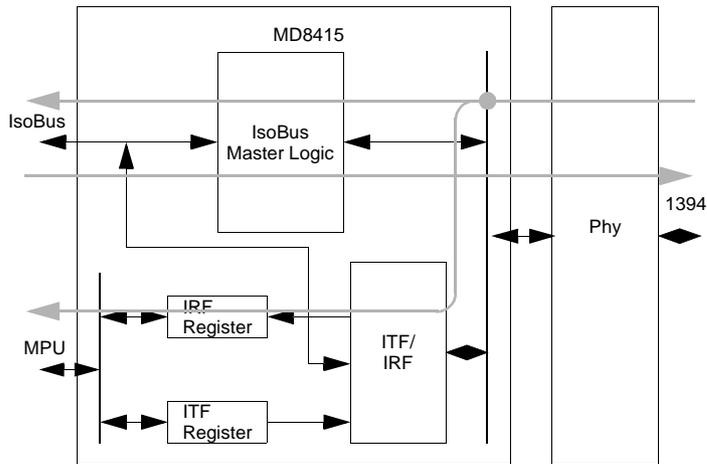


Figure 10-0-2 For Master Mode, IRF

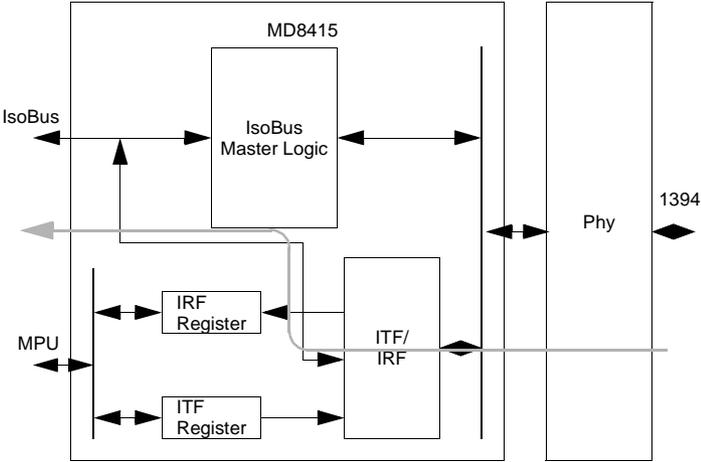


Figure 10-0-3 For Slave Mode, IRF

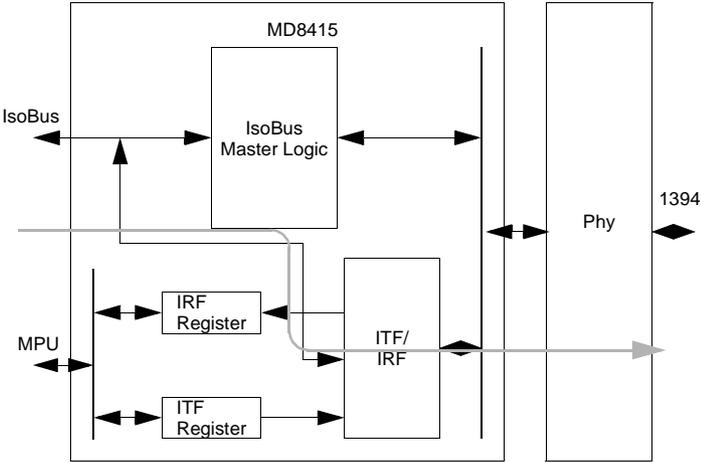


Figure 10-0-4 For Slave Mode, ITF

## 10-1 Master Mode

The MD8415 handles the Isochronous data with the external system via an exclusive bus of IDATA(15:0). The MD8415 functions as a non-periodic master unit, and makes a request for data input from the outside during transmission and for data output to the outside during reception. It performs synchronous transfer using a variety of control signals.

Since the MD8415 has no buffer in it for Isochronous packets, the external system is always required to be reactive to each data input request raised from the MD8415 during transmission. In addition, the external system is always required to acquire data at an output timing from the MD8415 during reception. During transmission, control in the packet unit is effected with the ITREQ# signal from the external system.

For both transmission and reception, the transfer rate of the intra-packet data, involving the Isochronous data with the external system, is a frequency of SCLK/8 if the said packet is of 100Mbps and SCLK/4 if it is of 200Mbps. SCLK is a system clock signal of the MD8415, input from the PHY.

The difference from the MD8415 is that the signal of CH(1:0) has been deleted.

### 10-1-1 Isochronous Cycle Start Timing

The isochronous cycle is started at the intervals of 125 $\mu$ sec. Because of interrupt by asynchronous packets, however, actual cycle start has some jitter. As a signal of indicating the start of actual isochronous cycle, the MD8415 generates an output of ICS signal. The ICS signal is asserted for 4 clocks of ICLK.

A transmission request is made based on this ICS signal, and the successive isochronous cycle. Data output is also generated during reception if there is a receive packet preceded by ICS.

### 10-1-2 Processing for Transmission request

When the isochronous source of the external system makes a request to send a packet to the MD8415, this request is raised toward the MD8415 using the ITREQ# signal in the first place. When ITREQ# is asserted, the MD8415 identifies the occurrence of a request from the outside to send a packet, and recognizes the end of this request in the cycle by examining the negate data.

This identification is effected under the condition that the ITREQ# signal is asserted in the enable period of the ICS signal. As shown in Fig. 8-2, information of ITREQ# assert is picked up with any one of ICLK 1, 2, 3, and 4 while ICS is in the enable period. According to an example of Fig. 10-1-1, ITREQ# assert is identified with ICLK 2.

Using the isochronous cycle started shortly after the presentation of the ICS signal, the MD8415 executes the packet transmit request by actuating the PHY device. At the time point when arbitration is prevalent on the bus, the MD8415 sends out a data request toward the external system using the ITX# signal. For actual data request at that time, the MD8415 sends out a request of data pre-read operation toward the external system for the amount of 4 words (2 Quadlets) after the lapse of 1 ICLK clock from the rising point of ICLK that has identified the ITREQ# being asserted. Therefore, regardless of whether the arbitration is acquired, the MD8415 always makes a request for data pre-read operation. Consequently, the external system is required to provide for at least 2 Quadlets (4 words) of data until ITREQ# is asserted and this is identified in the ICS period. In the same manner as for 200Mbps, the pre-reading rate is requested at the intervals of 2ICLK.

Transmission packet data on the IDATA bus are picked up at the rising point of ICLK during the ITX# enabled period. According to the example in the figure, Iso-U(1) of the initial data is picked up at the rising point of ICLK 4.

To finish a packet, IEOP# from the external system is asserted in conjunction with the ITX# timing when the last data request is made for the packet. This IEOP# assert is identified with the rising point of ICLK in the ITX# enable period of the last data, similarly as for packet data pickup timing. In case that the transmission during 1Cycle is completed in this packet, ITREQ# must be negated before IEOP# timing. As shown in the section 10-2-3 that ITREQ# is entered with successive assert, the successive multi-channel transmission is possible.

In the case of packet transmission only with an isochronous header and "0" for data payload, IEOP# is asserted at the second word out of one Quadlet (2 words) of the header, and presence of a vacant packet of data is shown to the MD8415. Thus, the MD8415 does not place a request for data pre-read operation of these two words, and packet transmission for the header only is effected.

Handling for transmission is the same as for isochronous normal mode and auto mode. Only difference is that contents of data for data pre-read operation in normal mode are given by the first Quadlet in the isochronous header and the payload area. They are given by the two Quadlets of the payload area in auto mode.

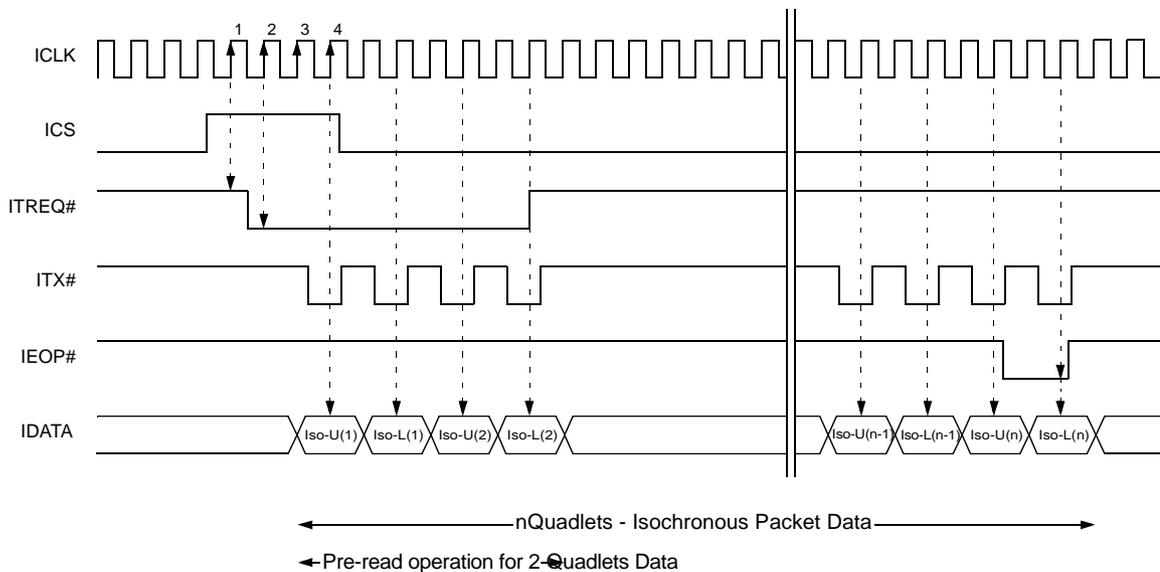


Figure 10-1-1 Isochronous Transmission Timing

### 10-1-3 Processing for Multi-Channel Transmission

Fig. 10-1-2 shows an example multi-channel (multiple packets) transmission using a certain isochronous cycle. As shown, ITREQ# is asserted in the enable period of the first (Isochronous Packet Data (1)) IEOP#, and the MD8415 identifies the presence of the next transmit packet according to this fact. When this has been identified, the MD8415 performs pre-read operation for the next packet data with the timing shown (period of pre-read for Isochronous Packet Data (2)). In the same manner as described in 10-1-2, a packet data request is successively forwarded to the external system at the time point when the arbitration is prevalent after pre-read operation. In the example of the figure, transmission of two packets is effected and then ITREQ# is negated with IEOP# of Isochronous Packet Data (2) to complete that cycle, because packet transmission is finished in the period of Isochronous Cycle (N).

For multiple packet transmission as described, ITREQ# is kept asserted and this action is finished by negating ITREQ# with IEOP# of the last packet.

When ITREQ# is once negated in the enable period of IEOP# in an isochronous cycle, the MD8415 identifies it as the end of transmission of this cycle at the negate time point and finishes this cycle, even though another assert is retired. When this assert condition is maintained until presentation of the next ICS, this is recognized as a request for transmission of the next isochronous cycle.

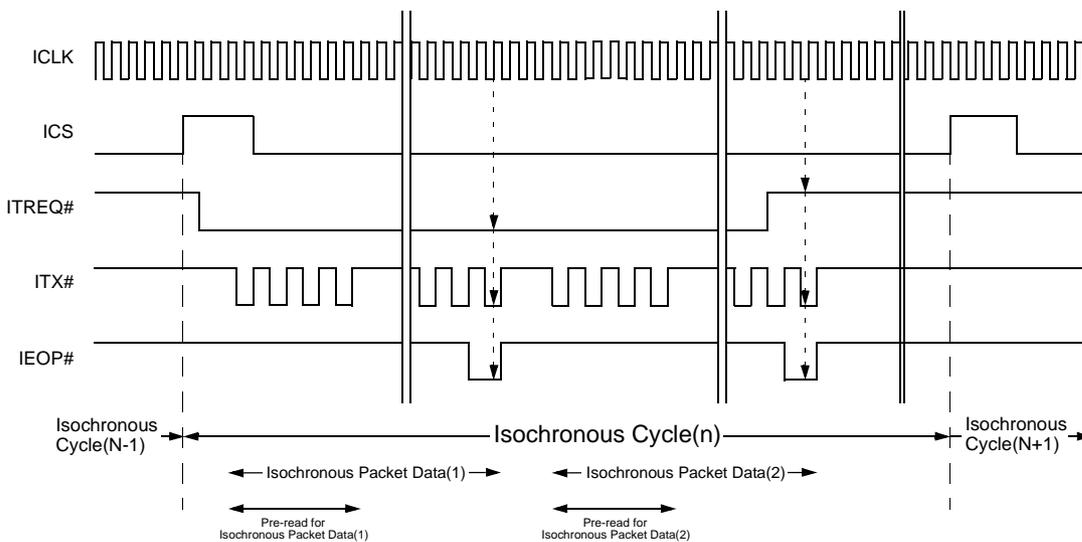


Figure 10-1-2 Multi-channel Isochronous Transmission Timing

### 10-1-4 Processing for Reception

With IsoMode in the Control Register and the contents of Isochronous Receive Configuration Register, reception of an isochronous packet is successively started after the generation of a Cycle Start packet. The start of a receive packet is indicated at the external system by asserting IRCV#. Until reception of this packet is finished, IRCV# is kept asserted. In other words, IRCV# asserted shows that a packet is being output. In the assert period for IRCV#, the both way IDATA bus is in the output condition.

After IRCV# assert, isochronous receive packet data are output from the IDATA bus in synchronization with IRX# asserted.

When IRCV# is asserted, the MD8415 continues to output packet data. Therefore, the external system must be arranged so that these data can be picked up without fail.

The reception condition is shown as following table:

IsoBusMode	IsoDataMode	ITF/IRF	Receiving Condition register	Output destination	IsoModeL	Data Type
Master	0 (Normal)	0 (ITF)	Iso Bus Receive Config Register 1, 2, 3, 4	IsoBus	0	Header available Non-Status
			All Iso		1	
		1 (IRF)	Iso Bus Receive Config Register 1, 2, 3, 4	IsoBus	0	Header available Non-Status
			All Iso	IsoBus	1	Header available Non-Status
	1 (Auto)	1 (IRF)	Iso Buffer Receive Config Register	IRF	-	Header available Status available
			Iso Bus Receive Config Register 1	IsoBus	-	Non-Header Non-Status
1 (Auto)	0 (ITF)	Iso Bus Receive Config Register 1	IsoBus	-	Non-Header Non-Status	
		Iso Buffer Receive Config Register	IRF	-	Header available Status available	

Table 10-1-1 Conditions for Isochronous reception (Master)

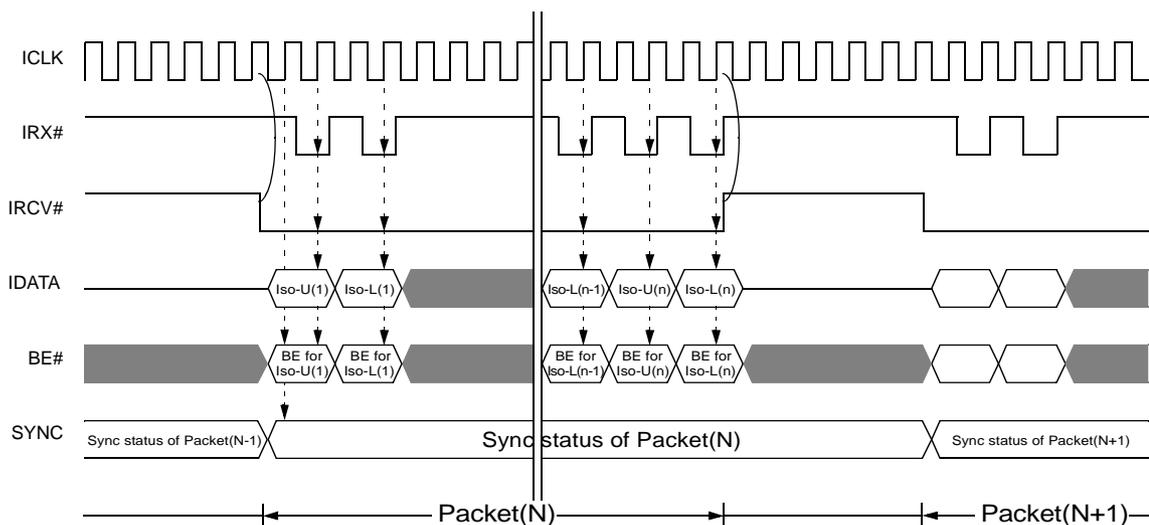


Figure 10-1-3 Isochronous Reception Timing

### 10-1-4-1 BE# Signal

It is possible to know the byte position of zero padding in the payload area of the received packet, using the BE# signal in the unit of upper 8-bit and lower 8-bit. BE# is output, synchronized with a data output on the IDATA bus. BE#(1) shows data of IDATA(15:8) and BE#(0) shows data of IDATA(7:0). BE#="0" indicates valid data and BE#="1" shows a padding byte. The MD8415 identifies zero padding according to the length value in the header (Length) and the Quadlet number of the packet payload (Quadlet), when the Length value is in the relationship of  $(\text{Quadlet}-1) < \text{Length}/4 < \text{Quadlet}$ .

### 10-1-4-2 SYNC Signal

According to the respective data in the SYNC area in the Isochronous header of the received packet and those set with the Isochronous Receive Configuration Register, it is possible to know the status with the SYNC signal, meeting the requirements shown below.

#### SYNC Values

- "01" : When the StartSync value set in the Register coincides with the SYNC value in the packet header.
- "10" : When the StopSync value set in the Register coincides with the SYNC value in the packet header.
- "11" : When the StartSync value set in the Register coincides with the SYNC value in the packet header, and further coincides with the StopSync value.
- "00" : Conditions other than the above.

At the time point when IRCV# is asserted, the above conditional values are output according to that packet. This condition is retained until the next IRCV# is asserted (next packet reception).

### 10-1-5 CycleStartPacket Reception

When the node is of root and the MD8415 is sending a CycleStart packet, the MD8415 provides output to the IDATA bus only for the Cycle\_Time\_Data field in the packet. The same operation is performed when the MD8415 receives a Cycle Start packet in the non-root mode. As shown in Fig. 10-1-4, the external system identifies the condition with the CT signal enabled that Cycle\_Time\_Data is output on the IDATA bus.

Each Cycle\_Time\_Data block of 32 bits is output twice to the IDATA bus by dividing it into two blocks, each in 16-bit bus width. Accuracy of each Cycle\_Time\_Data spacing of the first output data is relatively the same as the timing accuracy when the cycle timer in the MD8415 is updated with Cycle\_Time\_Data. Accordingly, the external system can construct a local cycle timer with the same accuracy as for the cycle timer in the MD8415, based on the output timing of the CT signal.

(As shown in Fig. 10-1-4, Delta t (=t2-t1) being a spacing from t2 to t1 offers the same accuracy as for the one in the MD8415.)

According to this function of the MD8415, the external system can use a cycle timer which provides a bus time for the 1394 bus. For example, time-stamp processing becomes possible.

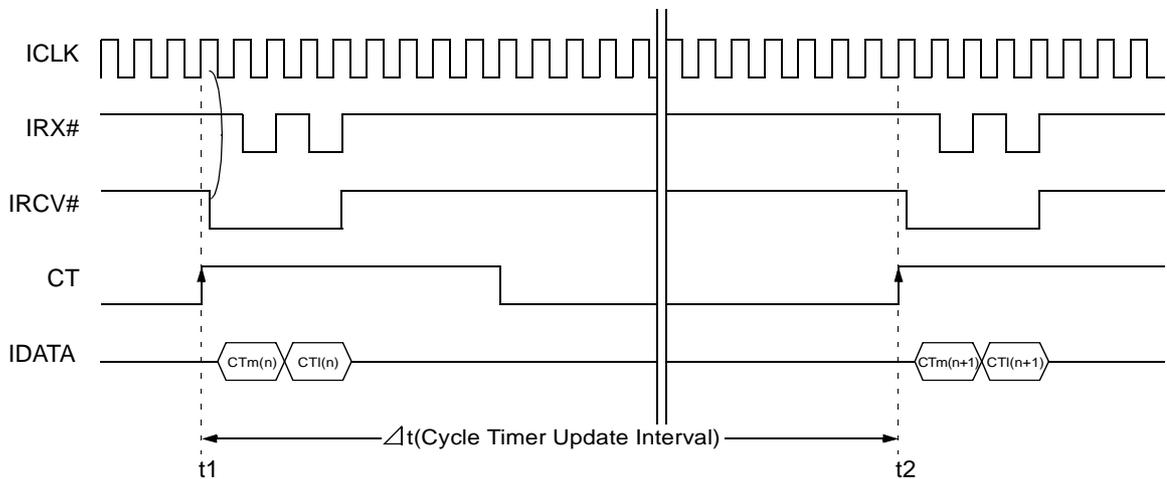


Figure 10-1-4 CycleStartPacket Reception

Cycle\_Time\_Data in a Cycle Start packet appearing on the IDATA bus in Fig. 10-1-4 is a format for CTm in Table 10-1-3 and for CTl in Table 10-1-3.

IDATA(15:0)															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
second_count							cycle_count								

Table 10-1-2 Contents of CTm

IDATA(15:0)															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cycle_count				cycle_offset											

Table 10-1-3 Contents of CTl

### 10-1-6 Packet error announcement during reception

When an isochronous packet is received, and if this reception is unusual due to CRC error or any defect in the packet, the MD8415 sends out information of such a packet error to the external system, by asserting the IRERR# signal. With this IRERR# signal, the external system can abandon a packet that is suffering from an error.

The condition is stored in the Isochronous Receive Error register.

#### 10-1-6-1 Error Announcement for Cycle Start Packet

Regarding announcement of an error in the Cycle Start packet, an ordinary CRC error is indicated by the IRERR# signal after the lapse of 8 clocks of ICLK after the CT signal has been asserted as shown in Fig. 10-1-5.

Even though there is no occurrence of a CRC error, the CT signal is asserted till the last timing for the occurrence of an error in the Cycle Start packet. (For 9 ICLK clocks) An error such as Cycle Start packet defect, etc., accompanies the appearance of IRERR# before this timing, but IRERR# occurring in the enable period of CT signal is always regarded as an error in the Cycle Start packet.

Therefore, when updating a local Cycle Timer counter using this Cycle Timer value from the external system, it is possible to avoid updating with a cycle timer value in the faulty Cycle Start packet, by loading the cycle timer value at the falling timing of the CT signal and updating the Cycle Timer counter under the condition that no IRERR# has occurred in the assert period of the CT signal.

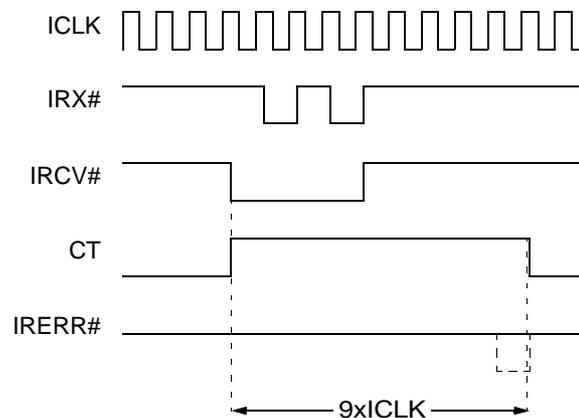


Figure 10-1-5 Error Timing during Cycle Start Packet

#### 10-1-6-2 Error Announcement for Isochronous Packet

According to errors, such as CRC error of the header/payload, discrepancy between length value in the header and data number in actual packet payload, lack of data in the header, mid-lack of payload, etc., the appearance pattern of IRERR# can differ in many ways. Whether a packet being received during IRCV# assertion is encountering an error or not is known in the period after the assertion of IRCV# or the negation of IRCV# of the packet and before the assertion of IRCV# in the next packet received. The relationship among IRX#, IRCV#, and IRERR# is shown by the patterns in Fig. 10-1-6. In Pattern (1), IRCV# and IRERR# are simultaneously asserted and reception of this packet is finished when IRCV# is negated without the assertion of IRX#. In this case, the error pattern indicates that reception is finished only with the first Quadlet of the packet. Reception of isochronous packet header itself is also ended on the way, without attaining the specified number of quadlets.

In Patterns (2) and (3), there is a header CRC check error, lack of a packet, etc. during reception. In Pattern (4), this is mainly a CRC check error in the payload area and IRERR# is asserted.

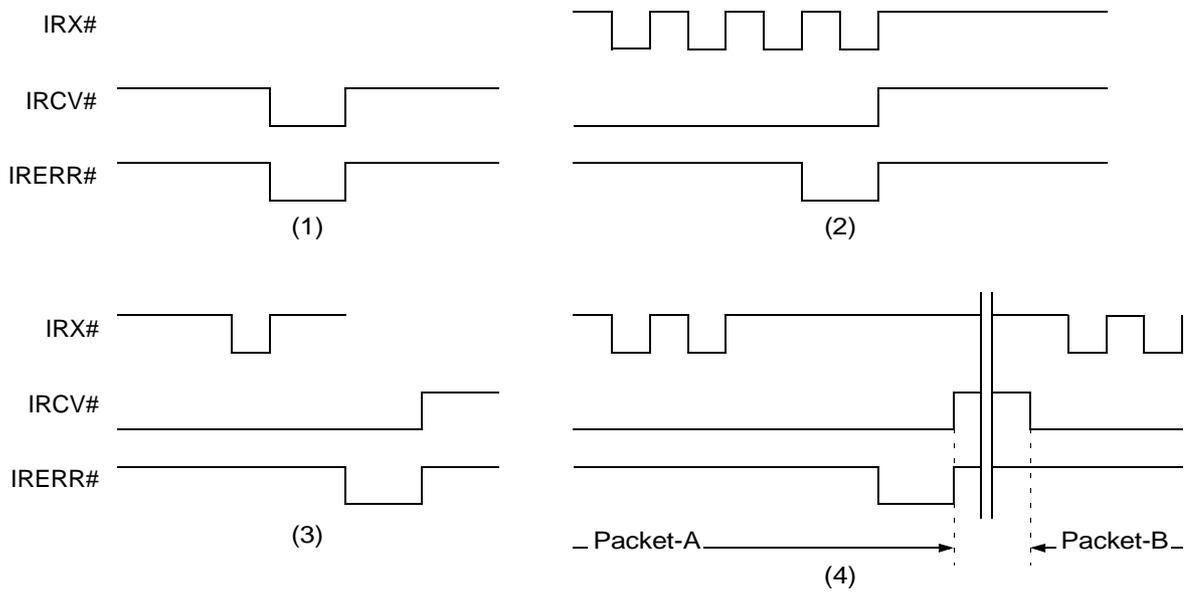


Figure 10-1-6 IRERR# Appearing Pattern

### 10-1-7 Difference in timing of 100Mbps, 200Mbps, or 400Mbps

For packet reception at 100Mbps, 200Mbps and 400Mbps, timing is different as shown in Fig. 10-1-7 as a matter of course. Normally, followings are output from IDATA bus: Packet received with 400Mbps is output at 1CLK each (25MHz), Packet received with 200Mbps is output at every 2-clock for 1CLK (12.5MHz), Packet received with 100Mbps is output, every 4-clock of 1CLK. In case of 100Mbps, as shown in Fig. 10-1-7, the quadlet of 2-word (16-bit) is output, every 2-clock of 1CLK.

Due to the overhead caused by calculation of header CRC, some timing may have more spacing.

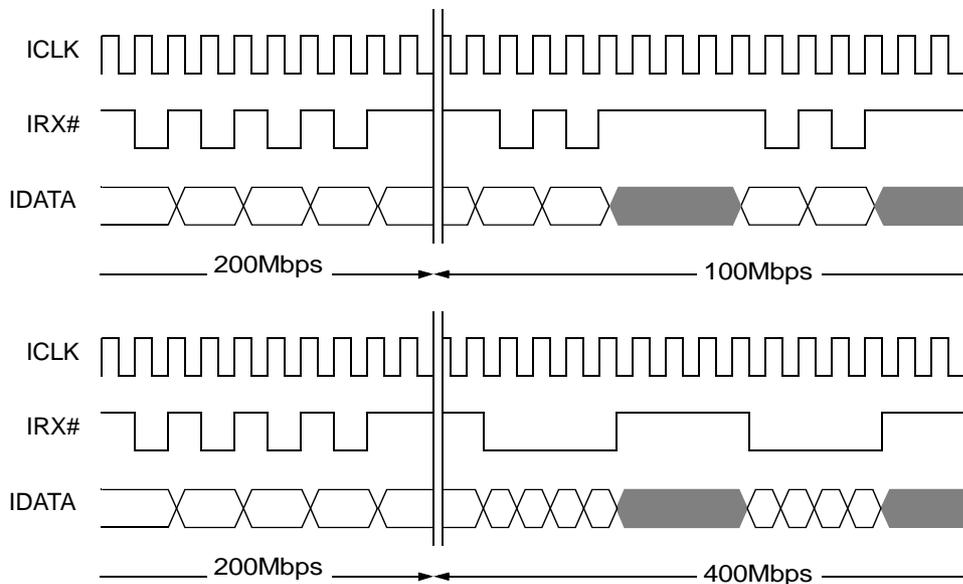


Figure 10-1-7 100/200/400Mbps Timing

### 10-1-8 Sync Control

MD8415 is able to control the reception via the sync field in Isochronous header. During transmission, any special processing is not made in normal mode because the Isochronous header is also input from the Isochronous Bus by the user. However, in the auto-mode, the sync field can be controlled by the external signal because the MD8415 itself generates and transmit the Isochronous header.

#### 10-1-8-1 Sync control transmission by ITSYNC signal at auto-mode

As shown in Fig. 10-1-8, ITSYNC signal is sampled at "2" of ICS signal. As shown in Fig. 10-1-9, the sync field value of Isochronous cycle is determined by this information. When the SyncEn in Isochronous Transmit Configuration register is not set, the sync field can not be controlled by ITSYNC signal.

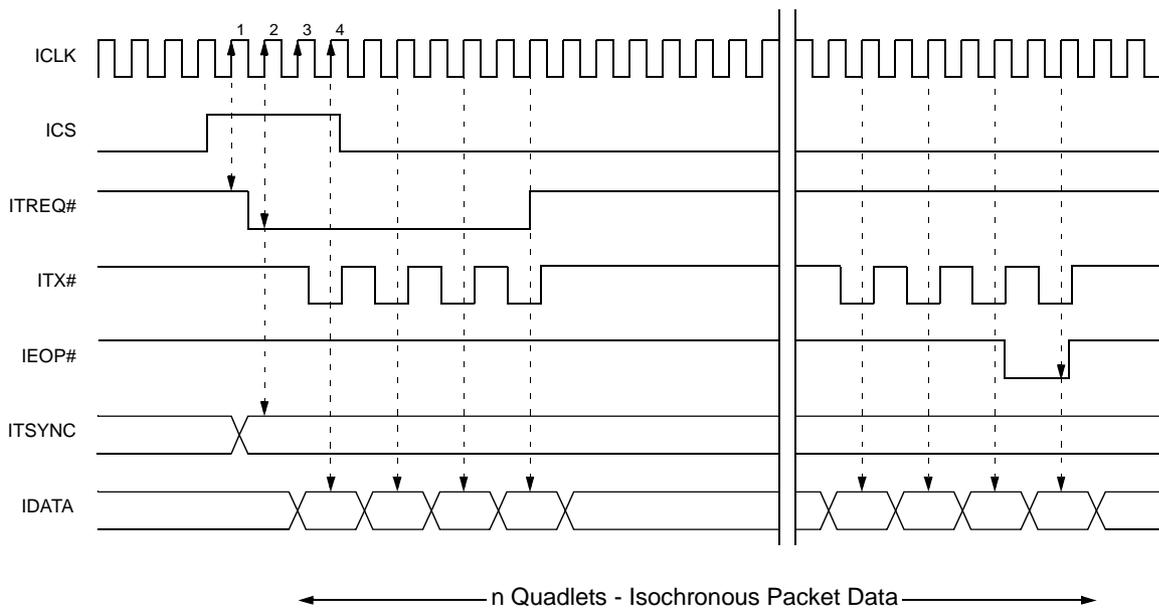


Figure 10-1-8 ITSYNC Timing

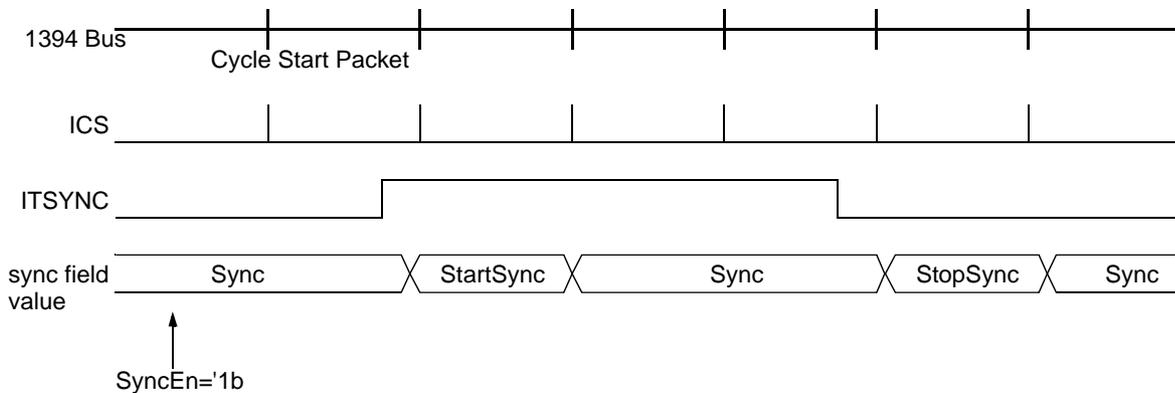


Figure 10-1-9 Relationship between ITSYNC Signal and sync Field

### 10-1-9 BUSRST signal

When announcement of BusReset is sent from the PHY, a pulse output is generated for a length (duration) from the start of BusReset to the end of SelfID.

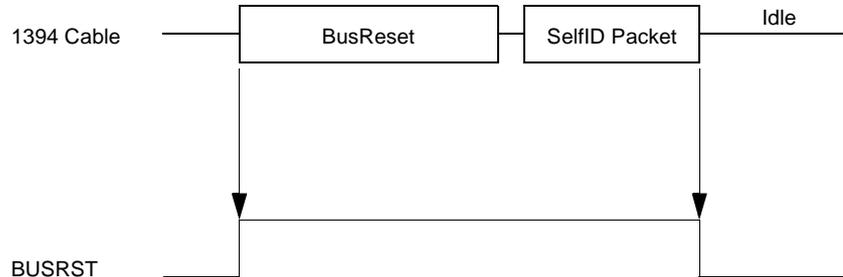


Figure 10-1-10 BUSRST signal

### 10-1-10 ITF or IRF transfer

By ITF/IRF bit setting in the Control register, setting is made to define whether the Isochronous buffer is for ITF or IRF. In the ITF mode, Access to the ITF/IRF buffer is treated to be valid only for the ITF register. Access to the IRF register is disregarded. In the IRF mode similarly, Access to the ITF/IRF buffer is valid only for the IRF register and Access to the ITF register is disregarded. The Isochronous Packet format intended to be stored in the buffer is only the one that includes a header.

#### 10-1-10-1 ITF transmission

The data read from the ITF register are stored in the ITF buffer. The storing format includes a header block.

When the data for a transferring packet have been stored in the ITF buffer, ITGo is set up. With the CycleStartPacket coming shortly after, the contents of ITF are transmitted. If there is already a Send request from the Isochronous Bus side at that time, the contents of the ITF buffer are sent out first of all, and the data from the Isochronous Bus side are then sent out.

### 10-1-10-2 IRF transmission

Data are received with the setting value of the Isochronous Buffer Receive Configuration register and stored in the IRF buffer. The storing format includes the header and status blocks. It is for one channel only. Even when the data are the same as those of the receiving channel at the Isochronous Bus, they are stored in the IRF buffer. Upon completion of storage in the IRF buffer, IRxEnd of the Interrupt register is set up. In case there is an error arising at the time of storage in the buffer, no storage is performed in the IRF buffer. In such a state, the data are stored in the Isochronous Receive Error register.

### 10-1-10-3 Normal/Auto mode

The Normal/Auto mode of data in the Master mode is determined by setting up the IsoDataMode bit of the Control register, and the condition becomes as described below. This designation is made only to the Isochronous bus. In regard to the data for the ITF/IRF buffer, the Isochronous header is also included in the same manner as for the Normal mode.

#### 1) Normal mode

Transmission via the Isochronous bus depends on the signal that is controlled from an external logic. Reception is also determined by setting up the IsoBusModeL bit of the Control register. In other words, a condition is assumed to accept 4-channel reception or all-channel reception (IsoSnoop).

The packet format includes the Isochronous Header.

#### 2) Auto mode

Only one-channel transmission is enabled via the Isochronous bus. The Isochronous header block is automatically generated inside the MD8415. Therefore, the packet that is input from the Isochronous bus is the Payload block only. The Tag, Channel, and sync blocks use parameters defined at the Isochronous Transmit Configuration register, and the transmission data length is determined by the ITLength field. In this case, however, ITEOP# is invalid and Isochronous transmission is effected when an input of data length defined in the ITLength field is entered.

Reception is effected under the condition defined by Isochronous Bus Receive Configuration Register 1. Setting for the IsoBusModeL bit of the Control register is disregarded and only-one-channel reception is conducted.

The packet format does not include the Isochronous header but includes the Payload block only.

### 10-1-11 IsoBus added function

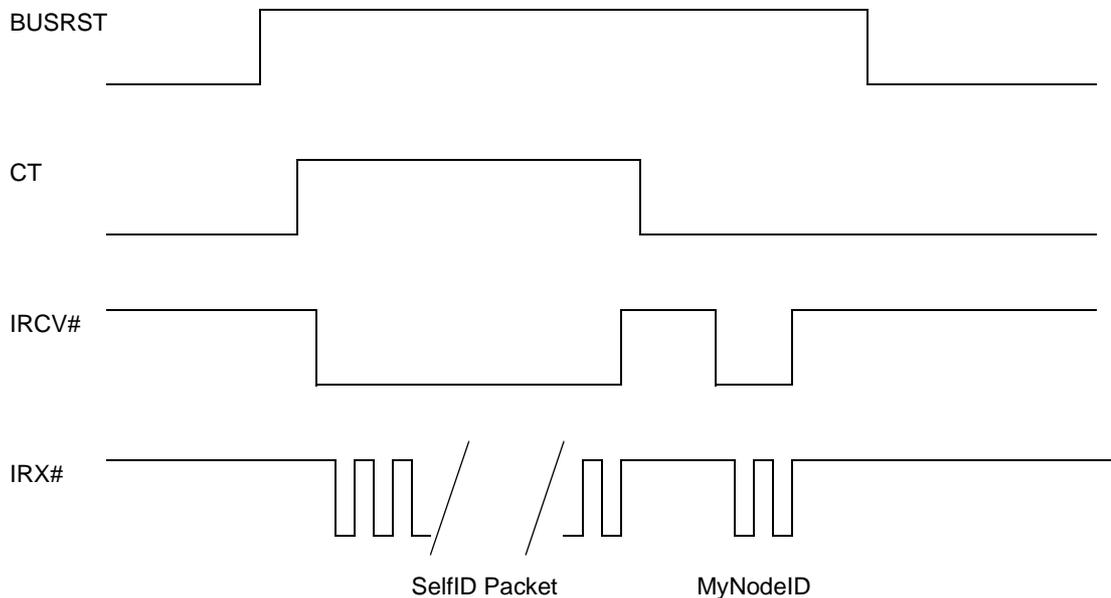
The IsoBus has a function of outputting information other than the data to be generated in ordinary Isochronous Transfer operation. There are two types of functions available; sending out the NodeID information of the PHY connected to the MD8415 to the IsoBus side (MyNodeID) and sending out all contents of the SelfID packet to the IsoBus side after BusReset.

Instead of losing the capability of both Asynchronous Transfer and Isochronous Transfer, all information about the "1394" Bus is snooped and it can be output to the IsoBus side (snoop function).

#### 10-1-11-1 MyNodeID and SelfID Packet output

If PhyIDOut of the Packet Control register has been set, the Reg0 information (8 bits) from Phy to Phy is output to IDATA[7:0] while the BUSRST# signal is asserted. IDATA[15:8] is unformatted. In such a case, both IRCV# and IRX# are asserted in the same manner as for ordinary IDATA output. Only difference is that the BUSRST# signal is asserted.

Similarly, if SelfIDOut of the Packet Control register has been set, the SelfID packet information from Phy is immediately output to IDATA[15:0] while the BUSRST# signal is asserted. In this case, both IRCV# and IRX# are asserted in the same manner as for ordinary 100Mbps reception. Only difference is that the BUSRST# signal is asserted. There is no relation to the RxSelfID bit that is located in the Packet Control register. Only SelfID packet shortly after BusReset is output. If SelfID cannot be received normally (the last being not 1 during storage in ARF), IRERR# is asserted for the amount of one clock signal.



$$\text{SelfID Packet} = \text{CT} \ \& \ \text{!IRCV\#} \ \& \ \text{!IRX\#} \ \& \ \text{BUSRST}$$

$$\text{PhyID(MyNodeID)} = \text{!CT} \ \& \ \text{!IRCV\#} \ \& \ \text{!IRX\#} \ \& \ \text{BUSRST}$$

Figure 10-1-11 MyNodeID, SelfID, Packet

#### 10-1-11-2 Snoop Mode

When the EnSnoop bit is set at "1" in the Packet Control register (0x10), it becomes possible to assume the Snoop mode. At that time, all data from PhyI/F are output to the IsoBus side. This mode is valid only if IsoBus is in the Master mode. Transmission is enabled in the Snoop mode. The ATF, ARF, and ITF/IRF buffers are also invalid. (IsoBusModeH=0)

## 10-2 Slave Mode

When the MASTER# terminal is at the "H" level, the Slave mode is assumed. The Isochronous buffer is set at ITF or IRF by setting up the ITF/IRF bit of the Control register. In this case, however, Isochronous Transfer cannot be performed from the host.

The ITF/IRF buffer is connected only to the Isochronous bus. In the Slave mode, transmission or reception can be performed under the control from an external circuit using the ITF/IRF buffer in the MD8415. Mode setting can be made from a register. In this case, transmission and reception cannot be performed at the same time; either one is to be selected.

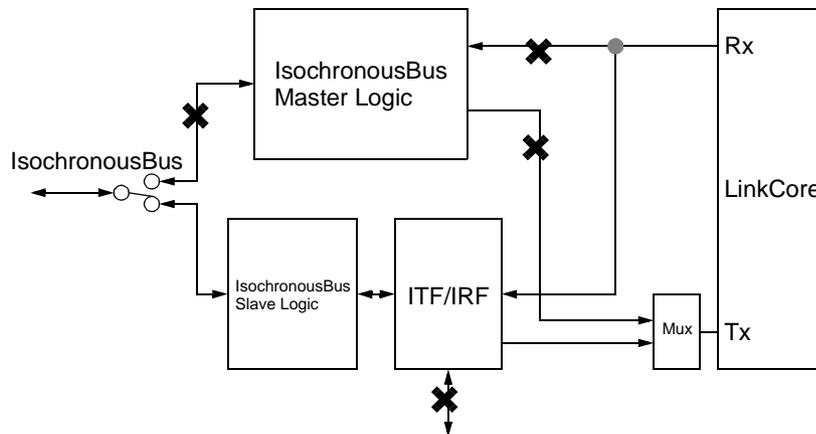


Figure 10-2-1 Buffer access for Isochronous slave

### 10-2-1 Signals used in the IsochronousBus Slave mode

When the Isochronous Bus is set in the Slave mode, the control signals are as specified below.

The interface is SynchronousFIFO-like and the following signals are used for control:

- ICLK (I) : IsochronousBus clock. 33MHz max.
- ITEN# (O) : ITF Write Enable signal. (For Assert, the ITX# signal is asserted to enable Write in ITF.)
- ITX# (I) : Write Enable signal. (For Assert, Write from IDATA to ITF is effected at the rise of ICLK.)
- IRX# (I) : Read Enable signal. (For Assert, Read from ITF to IDATA is effected at the rise of ICLK.)
- ITEOP# (I) : Last data signal for transmission
- IRPKT# (O) : FIFO storage signal for reception
- IDATA (I/O) : Data bus

The IRX# input is disregarded when the ITF/IRF buffer is set at ITF (for transmission), and the IRX# input is disregarded when the buffer is set at IRF (for reception).

### 10-2-2 Signal for IsochronousBus Slave transmission

When the Isochronous bus is set at Slave Mode and the ITF/IRF buffer is set at ITF, the data input from the Isochronous bus is written in the ITF buffer.

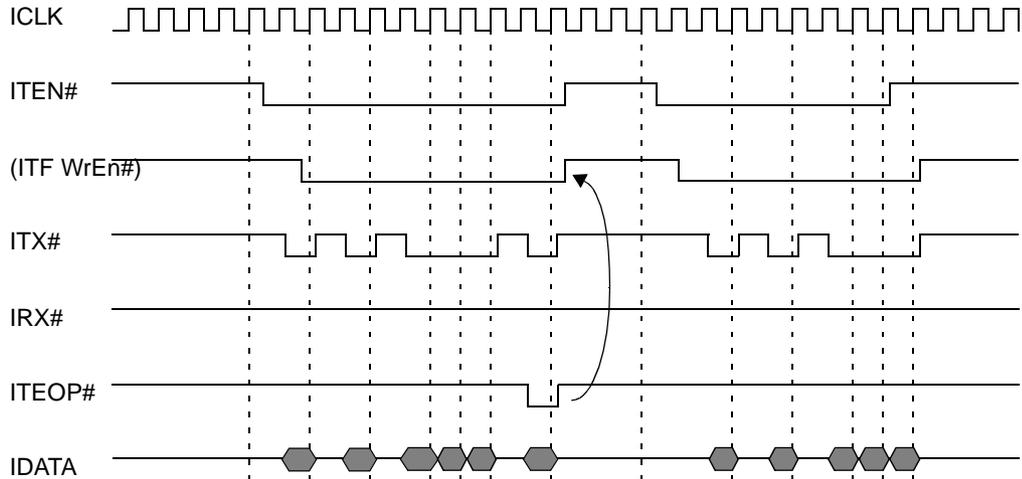


Figure 10-2-2 IsochronousBus signal (transmission)

If there is a storage area for the ITF buffer, ITEN# is asserted. Namely, the same operation is performed except for the case of ITFFull in the Buffer Status and Control register and the conditions specified below.

ITEN# is not asserted in the time period from the start of BusReset to the end the SelfID phase, and after storage has been finished in the ITF buffer and ITEOP# is asserted. The condition of asserting ITEOP#, negating it, and asserting it again is established when transmission is started in the transmission-enabled Isochronous Cycle. At that time, the IRPKT# signal is kept fixed in the Negate state and the IRX# signal is disregarded. Transmission is effected with the next Isochronous Cycle after one packet has been finished for the data where ITEOP# is asserted.

### 10-2-3 Signal for IsochronousBus Slave reception

If there is a received packet, it is read out from the Isochronous bus when this the Isochronous bus has been set at the Slave mode and the ITF/IRF buffer has been set at IRF.

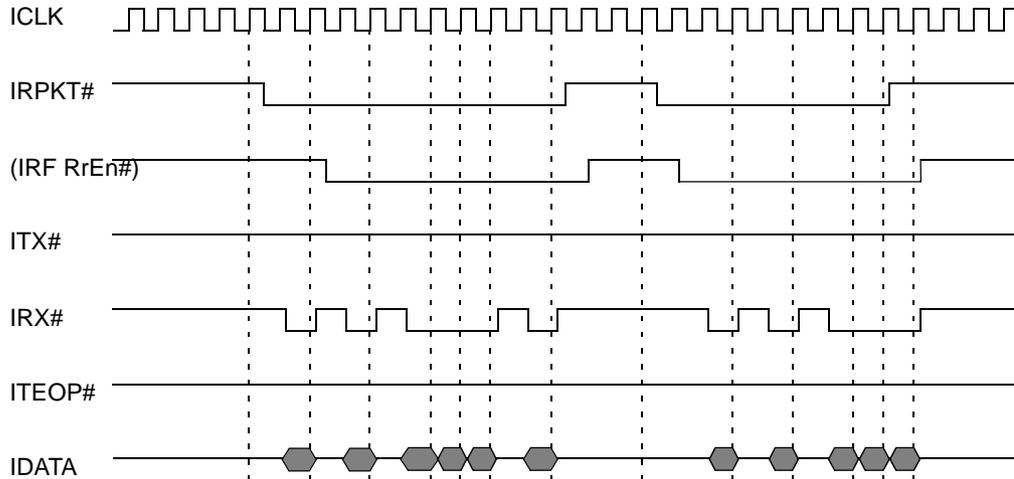


Figure 10-2-3 IsochronousBus signal (reception)

The same operation is performed by ITPKT# as for the IRFPacket bit of the Buffer Status and Control register. Only in this mode, however, this operation is enabled. In other modes, it is negated and fixed.

Upon completion of reception of one packet in the IRF, IRFPKT# is asserted. By IRX#, data are read from the IRF and output is generated from IDATA. The IDATA output is obtainable only in the period when IRX# is asserted. When all data have been read from the IRF, the IRFPKT# signal is negated. In this mode, the ITX# and ITEOP# signals are disregarded.

## 10-2-4 Transmission/Reception flow

### 10-2-4-1 Transmission processing

#### Normal Mode

In the Normal mode, a format inclusive of the Isochronous header is stored in the ITF buffer. When ITEOP# is asserted in the last data part of a packet being stored inclusive of a header, that packet is sent out.

### 10-2-4-2 Auto Mode

For transmission in the Auto mode, the Isochronous Header block is automatically generated inside the MD8415. Therefore, the packet that is input from the Isochronous bus to the ITF buffer is the Payload block only. The Tag, Channel, and sync blocks use parameters defined at the Isochronous Transmit Configuration register, and the transmission data length is determined by the ITLength field. When an amount of data defined by the ITLength field has been stored in the ITF, the MD8415 begins to perform Isochronous transmission. It is unnecessary to assert the ITEOP# signal.

### 10-2-4-3 Reception processing

#### Normal Mode

A maximum of 4 packets can be received and the packet format is arranged to include the Isochronous header.

### 10-2-4-4 Auto Mode

The number of packets that can be received is only one. The format does not include the Isochronous header, but it includes the Payload block only.

IsoBusMode	IsoDataMode	ITF/IRF	Receive condition register	Output destination	IsoModeL	Data Type
Slave	0 (Normal)	0 (ITF)	-	-	-	-
		1 (IRF)	Iso Bus Receive Config Register 1, 2, 3, 4	IRF -> IsoBus	-	Header available Status available
	1 (Auto)	0 (ITF)	-	-	-	-
		1 (IRF)	Iso Bus Receive Config Register 1	IRF -> IsoBus	-	Non-Header Non-Status

Table 10-2-1 IsoReceive Conditions (Slave)

## 11 EEPROM I/F

### 11-1 Type of EEPROM

The EEPROM is specified as follows:

Maker : NationalSemiconductor

Model No. : NM24Cxx

### 11-2 Control signals

The following two lines only are used for the control.

SDA : DATA I/O

SCLK : Clock Input

Drive addressing for the EEPROM interior is as described below. In order to accomplish addressing by designating A0 to A2, the MD8415 is designed to work as a 4k memory even when two NM24C02 units are connected in parallel.

Device Type Identifier							
1	0	1	0	A2	A1	A0	R/W

Table 11-2-1 Slave Addresses

Device	A0	A1	A2	Page Blks	Page Block Addresses
NM24C02/03	A	A	A	1(2k)	(none)
NM24C04/05	P	A	A	2(4k)	0 1
NM24C08/09	P	P	A	4(8k)	00 01 10 11
NM24C16/17	P	P	P	8(16k)	000 001 010 011 ..111

A: Refers to a hardware configured Address pin.  
P: Refers to an internal PAGE BLOCK memory segment.

Table 11-2-2 Slave Addresses strings

### 11-3 Storing format

As described below, data are stored in the Configuration ROM format, starting with the head of the EEPROM. The ROM format shown below is an example. The storing length of ROM is determined by CRC\_Length that is located in the first one Quadlet. The data of one Quadlet are accessed by Big-Endian. Practically, bit31:24 corresponds to Address n, bit23:16 to Address n+1, bit15:8 to Address n+2, and bit7:0 to Address n+3.

Configuration ROM (template)																																
Adrs	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	info_length (4)								CRC_length (0x15)								rom_CRC															
0x04	0x31 ("1")								0x33 ("3")								0x39 ("9")								0x34 ("4")							
0x08																																
0x0C	node_vender_ID																chip_ID_hi															
0x10																	chip_ID_lo															
0x14	root_derectry_length																root_derectry_CRC															
0x18	0x03								module_vender_ID																							
0x1C	0x0C								node_capabilities (at aminimum 0x0083c0)																							
0x20	0x8D								node_unique_ID_offser (2)																							
0x24	0xD1								unit_derectry (4)																							
0x28	2																node_unique_ID_CRC															
0x2C	node_vender_ID																chip_ID_hi															
0x30																	chip_ID_lo															
0x34	unit_derectry_length (3)																unit_derectry_CRC															
0x38	0x12								unit_spec_ID																							
0x3C	0x13								unit_sw_version																							

Figure 11-3-1 EEPROM Format

### 11-4 Method of connection

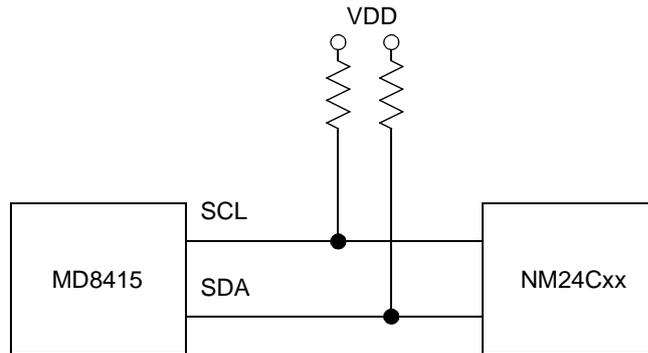


Figure 11-4-1 Connection through EEPROM

### 11-5 Access flow

When the power supply is turned on and IC-interior reset is completed, the MD8415 performs CRC check for the EEPROM data. Therefore, readout is started toward the EEPROM.

The CRC check by the MD8415 is performed simply to examine rom\_CRC in the heading one Quadlet.

## 12 Internal Timer

The MD8415 is provided with the following timers:

### 1) Bus Configuration timer

This is a timer intended to accomplish timer processing after the occurrence of BusReset in 1394.

When BusReset occurs and the SelfID phase is finished, this timer is started and the lapse time is used to define the BTInt bit for the Interrupt register. This timer is used mainly to obtain timings of 125 $\mu$ S, 625 $\mu$ S, and 1S.

### 2) General timer

This is a General timer and its starting time point can be controlled by a register. This timer can be used to set up a mode to cause a repeated interruption at the predetermined time intervals. It is also enabled to set up a mode to cause an interruption only once after starting.

The setting value is used to define the GTInt bit for the Interrupt register.

### 3) Split TimeOut timer

This timer is used to cause an interruption when the preset timer value has been attained. This timer can be used mainly to detect TimeOut in the Split transaction. The setting value is used to define the STO bit for the Interrupt register.

### 4) Retry timer

This is a timer to be used with the retry protocol of 1394. This timer starts up the MD8415 itself. It comes in two types; one for the single phase and the other for the dual phase. Either type is selected by the preset Retry protocol.

### 13 SelfID Packet Analysis

SelfID packet analysis is carried out to check the following items:

- 1) Matching between Quadlet data and reversal data
- 2) Continuity of node numbers
- 3) C(CMC) bit check
- 4) L(LinkActive) bit check
- 5) Sp(Speed) bit check

According to the contents of these checks, the following values are obtained:

- 1) Generation of rootNode number
- 2) Generation of irm Node number
- 3) Generation of Active Node Map
- 4) Generation of Speed Map

### 14 Interrupt Processing

The MD8415 has two types of registers; Interrupt register and Extended Interrupt register. For the Interrupt register, the Interrupt factor is directly reflected on the Interrupt signal of INT1# or INT2#.

For Interrupt factors located in the Extended Interrupt register, however, all logical sums are reflected on the extended bit of the Interrupt register and an Interruption occurs.

Therefore, when an Interrupt factor is set in the Extended Interrupt register and everything is then cleared, the extended bit of the Interrupt register is also cleared.

The allocation of Interrupt factors in conjunction with the INT1# and INT2# signals is determined by Interrupt Mast Register-1 and Interrupt Mast Register-2. If the setting has been made with the Interrupt Mast Register, the resultant Interrupt factor is masked and is not reflected on the INT1# or INT2# signal.

## 15 BusReset/CmdReset processing

### 15-1 BusReset processing

When BusReset occurs, the following registers are initialized.

ATBusy and ITBusy are cleared to '0b' in the DiagnosticStatusAndControlRegister

IDValid, root, and irmValid are cleared to '0b' in the NodeIdentificationRegister

irmNode of the Node Identification register is initialized.

The ActiveNodeMapRegister is cleared.

The SpeedMapRegister is cleared.

The BusConfigTimerRegister is initialized and started.

The ATRetryRegister is initialized.

BusReset of InternalCSR register is initialized.

In addition, the following FIFOs are cleared:

ATFi

ARFi

ARF (only for ControlRegisterÇÃARFBusReset=1b)

ITF (only for ITF/IRF=0b with ITFBusReset=1b in the Control register)

When a SelfID packet is received, the following operation is performed:

SelfID packet analysis (generation of irmNode, ActiveNodeMap, and SpeedMap)

Storage of SelfID packet in ARF (only for RxSelfID=1b in the Packet Control register)

Definite ÇÃroot bit of NodeIdentificationRegister

When the requirements specified below are met, a CycleStart packet begins to be transmitted.

TransmitEn='1b' of ControlRegister

CycleTimerEn='1b' of ControlRegister

CycleMaster='1b' of ControlRegister

root='1b' of NodeIdentificationRegister

## 16 P1394a function

The MD8415 has the function of P1394a. To use this function, it is necessary for the connected PHY to conform to P1394a.

### 16-1 Isochronous Multi-Speed Concatenated Packet

When two or more channels are transmitted during Isochronous transmission, such transmission is enabled by the use of a Multi-Speed Concatenated packet.

In order to use this function, it is necessary to set up the IsoConEn bit in the Control register. It is also necessary to set up the Enab\_multi bit located in a relevant register of the connected PHY.

In other words, both of the IsoConEn bit in the MD8415 and the Enab\_multi bit of the PHY must have been set at "1".

In case the Multi-Speed Concatenated Packet function is not used, both of the IsoConEn bit in the MD8415 and the Enab\_multi bit of the PHY must have then been cleared to "0". No operation is assured in any condition other than the above-mentioned setting.

### 16-2 Accelaration

It is possible to enable the functions of Ack Acceleration and Fly-by. To use these functions, it is necessary to set up the AckAccEn bit of the Control register. In addition, it is also necessary to set up the Enab\_accel bit located in a relevant register of the connected PHY.

In other words, both of the AckAcc bit in the MD8415 and the Enab\_accel bit of the PHY must have been set at "1".

In case the Multi-Speed Concatenated Packet function is not used, both of the AckAcc bit in the MD8415 and the Enab\_accel bit of the PHY must have then been cleared to "0". No operation is assured in any condition other than the above-mentioned setting.

## 17 Data Format

### 17-1 Asynchronous

#### 17-1-1 Quadlet Transmit (WriteRequest / ReadRequest)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													ID	spd		tLabel				rt	tCode			priority							
destinationID													destinationOffset_H																		
destinationOffset_L																															
quadlet data (for write request)																															

Table 17-1-1 Quadlet Transmit format (Asynchronous)

**ID** : BusID bit

This bit designates the Bus ID to be set for the source ID in the header region of IEEE1394 packet format; using Bus ID in MD8415 register or setting Local Bus (0x3FF).

- 0 = BusID in Node Identification Register of MD8415 is used.
- 1 = Set at 0x3FF.

**spd** : speed field

This field is used to designate the transfer speed. Refer to Table 17-8-6 regarding setting values.

**tLabel** : Transaction label field

This field is used to define a unique tag for each transferred transaction. The tLabel sent for requesting is used as a transaction label for correct response. The values used are valid in the range of 0h~Fh.

**rt** : retry field

This field is used to define whether this packet is in the middle of making a retry. The retry protocol is followed by the destination node. Refer to Table 17-8-1 regarding setting values.

**tCode** : Transaction code field

This field is used to set up a transaction code. The transaction code is used to define the packet type. Refer to Table 17-8-2 regarding setting values.

**priority** : priority field

This field is valid in the back plane environment. Therefore, the MD8415 is required to set up 0000b, without fail.

**destinationID** : destination ID field

This field is used to set up destination bus and node ID. The upper 10 bits out of the length of 16 bits are used to set up the destination bus ID and the lower 6 bits are used to set up the destination physical ID. Some figures in the setting value may have special meanings. Refer to Table 17-8-3 regarding details.

**destinationOffset**: destination Offset Address field

This field is used to define the lower 48-bit address of the destination node for the requested packet. The setting value need be defined in the quadlet unit in the case of a read request for quadlet data and a write request for quadlet data.

**quadletData:** quadlet data field

This field is used to set up actual transfer data (1 Quadlet).

**17-1-2 Block Transmit (WriteRequest / LockRequest / ReadRequest)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													ID	spd		tLabel				rt		tCode			priority						
destinationID													destinationOffset_H																		
destinationOffset_L																															
dataLength																extended_tCode															
block data quadlet 1																															
other block data quadlets																															
																padding (if necessary)															

Table 17-1-2 Block Transmit format (Asynchronous)

- ID** : BusID bit  
 This bit designates the Bus ID to be set for the source ID in the header region of IEEE1394 packet format; using Bus ID in MD8415 register or setting Local Bus (0x3FF).  
 0 = BusID in Node Identification Register of MD8415 is used.  
 1 = Set at 0x3FF.
- spd** : speed field  
 This field is used to designate the transfer speed. Refer to Table 17-8-6 regarding setting values.
- tLabel** : Transaction label field  
 This field is used to define a unique tag for each transferred transaction. The tLabel sent for requesting is used as a transaction label for correct response. The values used are valid in the range of 0h~Fh.
- rt** : retry field  
 This field is used to define whether this packet is in the middle of making a retry. The retry protocol is followed by the destination node. Refer to Table 17-8-1 regarding setting values.
- tCode** : Transaction code field  
 This field is used to set up a transaction code. The transaction code is used to define the packet type. Refer to Table 17-8-2 regarding setting values.
- priority** : priority field

This field is valid in the back plane environment. Therefore, the MD8415 is required to set up 0000b, without fail.

**destinationID** : destination ID field

This field is used to set up destination bus and node ID. The upper 10 bits out of the length of 16 bits are used to set up the destination bus ID and the lower 6 bits are used to set up the destination physical ID. Some figures in the setting value may have special meanings. Refer to Table 17-8-3 regarding details.

**destinationOffset**: destination Offset Address field

This field is used to define the lower 48-bit address of the destination node for the requested packet. The setting value need be defined in the quadlet unit in the case of a read request for quadlet data and a write request for quadlet data.

**dataLength**: data length field

This field is used to set up data length for the blockData field. The maximum value of the setting value depends on that of the speed field. Refer to Table 17-8-4 regarding details.

**extended\_tCode**: extended transaction code field

This field is used to define the extension tCode. This extended\_tCode becomes valid only if tCode is EAlock requestEC or EAlock responseEC. For other tCodes, 0000h must be set in this register. Refer to Table 17-8-5 regarding details.

**blockData**: block data field

This field is used to set up actual transfer data. If the dataLength field is not defined with a multiple of 4, this field need be filled with 00h to complete it in the Quadlet unit.

### 17-1-3 Quadlet Transmit (WriteResponse / ReadResponse)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												ID	spd		tLabel				rt		tCode		priority								
destinationID												rcode		reserved																	
reserved																															
quadlet data (for read response)																															

Table 17-1-3 Quadlet Transmit fresponse format (Asynchronous)

**ID** : BusID bit

This bit is set for the source ID in the header region of IEEE1394 packet format. It is designated for Bus ID to use Bus ID in the register of MD8415 or to set at Local Bus (0x3FF).

0 = BusID in Node Identification Register of MD8415 is used.

1 = Set at 0x3FF.

**spd** : speed field

This field is used to designate the transfer speed. Refer to Table 17-8-6 regarding setting values.

**tLabel** : Transaction label field

This field is used to define a unique tag for each transferred transaction. The tLabel sent for requesting is used as a transaction label for correct response. The values used are valid in the range of 0h~Fh.

**rt** : retry field

This field is used to define whether this packet is in the middle of making a retry. The retry protocol is followed by the destination node. Refer to Table 17-8-1 regarding setting values.

**tCode** : Transaction code field

This field is used to set up a transaction code. The transaction code is used to define the packet type. Refer to Table 17-8-2 regarding setting values.

**priority** : priority field

This field is valid in the back plane environment. Therefore, the MD8415 is required to set up 0000b, without fail.

**destinationID** : destination ID field

This field is used to set up destination bus and node ID. The upper 10 bits out of the length of 16 bits are used to set up the destination bus ID and the lower 6 bits are used to set up the destination physical ID. Some figures in the setting value may have special meanings. Refer to Table 17-8-3 regarding details.

**rcode** : response code field

Response code (rcode) is stored in this field. Refer to Table 17-8-8 regarding setting values.

**quadletData** : quadlet data field

This field is used to set up actual transfer data (1 Quadlet).

## 17-1-4 Block Transmit (LockResponse / ReadResponse)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													ID	spd		tLabel				rt	tCode			priority							
destinationID													rcode		reserved																
reserved																															
dataLength																extended_tCode															
block data quadlet 1																															
other block data quadlets																															
																padding (if necessary)															

Table 17-1-4 Block Transmit Response format (Asynchronous)

**ID** : BusID bit

This bit is set for the source ID in the header region of IEEE1394 packet format. It is designated for Bus ID to use Bus ID in the register of MD8415 or to set at Local Bus (0x3FF).

0 = BusID in Node Identification Register of MD8415 is used.

1 = Set at 0x3FF.

**spd** : speed field

This field is used to designate the transfer speed. Refer to Table 17-8-6 regarding setting values.

**tLabel** : Transaction label field

This field is used to define a unique tag for each transferred transaction. The tLabel sent for requesting is used as a transaction label for correct response. The values used are valid in the range of 0h~Fh.

**rt** : retry field

This field is used to define whether this packet is in the middle of making a retry. The retry protocol is followed by the destination node. Refer to Table 17-8-1 regarding setting values.

**tCode** : Transaction code field

This field is used to set up a transaction code. The transaction code is used to define the packet type. Refer to Table 17-8-2 regarding setting values.

**priority** : priority field

This field is valid in the back plane environment. Therefore, the MD8415 is required to set up 0000b, without fail.

**destinationID** : destination ID field

This field is used to set up destination bus and node ID. The upper 10 bits out of the length of 16 bits are used to set up the destination bus ID and the lower 6 bits are used to set up the destination physical ID. Some figures in the setting value may have special meanings. Refer to Table 17-8-3 regarding details.

**rcode** : response code field

Response code (rcode) is stored in this field. Refer to Table 17-8-8 regarding setting values.

**dataLength**: data length field

This field is used to set up data length for the blockData field. The maximum value of the setting value depends on that of the speed field. Refer to Table 17-8-4 regarding details.

**extended\_tCode**: extended transaction code field

This field is used to define the extension tCode. This extended\_tCode becomes valid only if tCode is EALock requestEC or EALock responseEC. For other tCodes, 0000h must be set in this register. Refer to Table 17-8-5 regarding details.

**blockData** : block data field

This field is used to set up actual transfer data. If the dataLength field is not defined with a multiple of 4, this field need be filled with 00h to complete it in the Quadlet unit.

### 17-1-5 Quadlet Receive

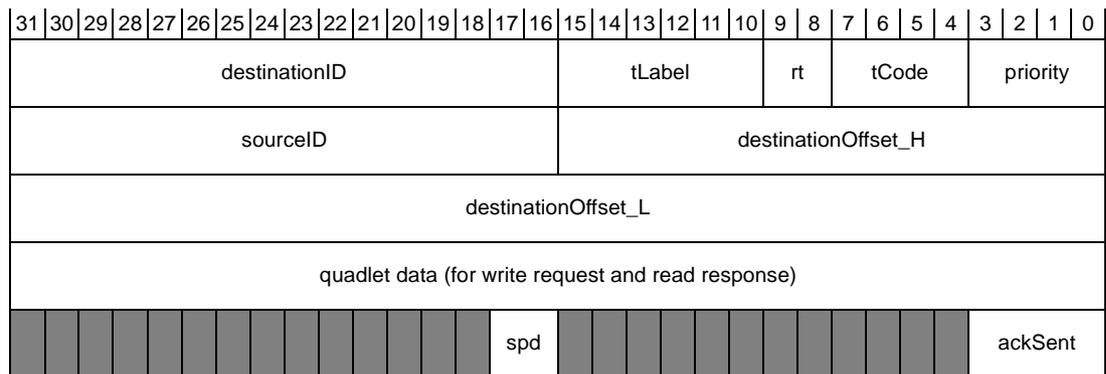


Table 17-1-5 Quadlet Receive format (Asynchronous)

**destinationID** : destination ID field

The destination bus of this packet and the node ID are saved in this field. The upper 10 bits out of the length of 16 bits are used to set up the destination bus ID and the lower 6 bits are used to set up the destination physical ID. Some figures in the setting value may have special meanings. Refer to Table 17-8-3 regarding details.

**destinationID** : destination ID field

The destination bus of this packet and the node ID are saved in this field. The upper 10 bits out of the length of 16 bits are used to set up the destination bus ID and the lower 6 bits are used to set up the destination physical ID. Some figures in the setting value may have special meanings. Refer to Table 17-8-3 regarding details.

**tLabel** : Transaction label field

This field is used to define a unique tag for each transferred transaction.

**rt** : retry field

This field is used to define whether this packet having been sent is in the middle of making a retry. Refer to Table 17-8-1 regarding setting values.

**tCode** : Transaction code field

A transaction code is saved in this field. The transaction code is used to define the packet type. Refer to Table 17-8-2 regarding setting values.

**priority** : priority field

This field is valid in the back plane environment. Therefore, 0000b is always saved.

**sourceID** : source ID field

This field is used to save the source bus and node ID. The upper 10 bits out of the length of 16 bits are used to set up the destination bus ID and the lower 6 bits are used to set up the destination physical ID. Some figures in the setting value may have special meanings. Refer to Table 17-8-3 regarding details.

**destinationOffset**: destination Offset Address field

This field is used to save the lower 48-bit address of the destination node for the requested packet.

**quadletData** : quadlet data field

This field is used to save transferred data.

**spd** : speed field

This field is used to designate the received speed. Refer to Table 17-8-6 regarding setting values.

**ackSent** : ackSent field

This field saves the Ack code returned as an acknowledge signal after this packet has been received. Refer to Table 17-8-7 regarding details.

**17-1-6 Block Receive**

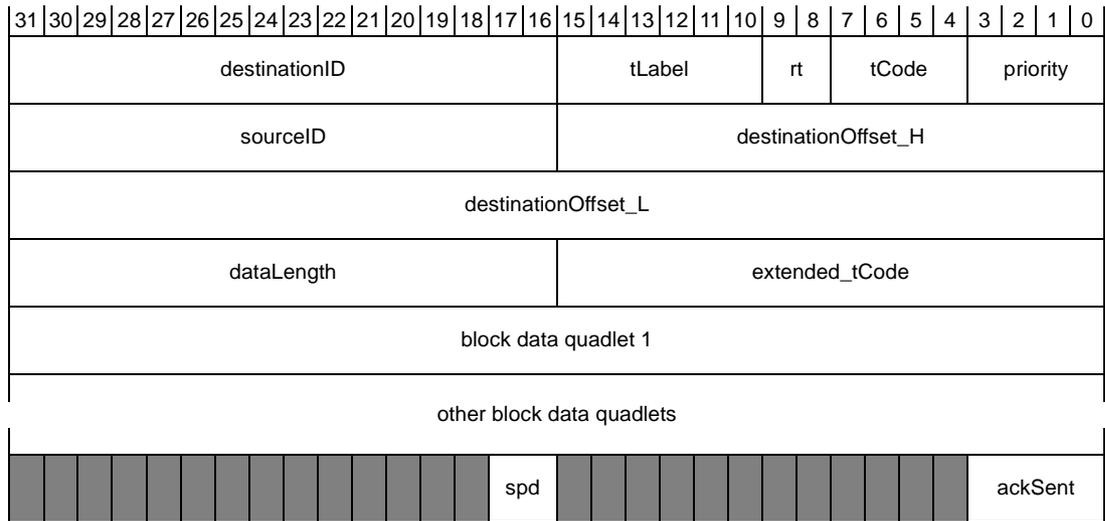


Table 17-1-6 Block Receive format (Asynchronous)

**destinationID** : destination ID field

The destination bus of this packet and the node ID are saved in this field. The upper 10 bits out of the length of 16 bits are used to set up the destination bus ID and the lower 6 bits are used to set up the destination physical ID. Some figures in the setting value may have special meanings. Refer to Table 17-8-3 regarding details.

**tLabel** : Transaction label field

This field is used to define a unique tag for each transferred transaction.

**rt** : retry field

This field is used to define whether this packet having been sent is in the middle of making a retry. Refer to Table 17-8-1 regarding setting values.

**tCode** : Transaction code field

A transaction code is saved in this field. The transaction code is used to define the packet type. Refer to Table 17-8-2 regarding setting values.

**priority** : priority field

This field is valid in the back plane environment. Therefore, 0000b is always saved.

**sourceID** : source ID field

This field is used to save the source bus and node ID. The upper 10 bits out of the length of 16 bits are used to set up the destination bus ID and the lower 6 bits are used to set up the destination physical ID. Some figures in the setting value may have special meanings. Refer to Table 17-8-3 regarding details.

**destinationOffset**: destination Offset Address field

This field is used to save the lower 48-bit address of the destination node for the requested packet.

**dataLength** : data length field

---

This field is used to set up data length for the blockData field.

**extended\_tCode**: extended transaction code field

This extended\_tCode becomes valid only if tCode is 'lock request' or 'lock response' For other tCodes, 0000h is saved in this register.

**blockData** : block data field

The transferred data are saved in this field.

**spd** : speed field

This field is used to designate the received speed. Refer to Table 17-8-6 regarding setting values.

**ackSent** : ackSent field

This field saves the Ack code returned as an acknowledge signal after this packet has been received. Refer to Table 17-8-7 regarding details.

**17-1-7 Asynchronous Stream**

**17-1-7-1 Transmit**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved														spd		reserved						tCode		reserved							
length														tag		channel				reserved		app									
Asynchronous data 1																															
Asynchronous data 2																															
other Asynchronous data																															
														padding (if necessary)																	

Table 17-1-7 Asynchronous Stream Transmit format

**tCode** : tCode field  
 tCode is stored in this field. The value must be 0xA.

**dataLength** : data length field  
 This field is used to set up data length for the blockData field.

**tag** : tag field  
 This field is used to set up Tag for Asynchronous transmit.

**channel** : channel field  
 This field is used to set up Channel number for Asynchronous transmit.

**spd** : speed field  
 This field is used to designate the received speed. Refer to Table 17-8-6 regarding setting values.

**app** : app field  
 This field is used to set up Sync data for Isochronous transmit.

**Asynchronous Stream Data:** Asynchronous Stream data field

This field is used to set up actual transfer data. If the dataLength field is not defined with a multiple of 4, this field need be filled with 00h to complete it in the Quadlet unit.

17-1-7-2 Receive

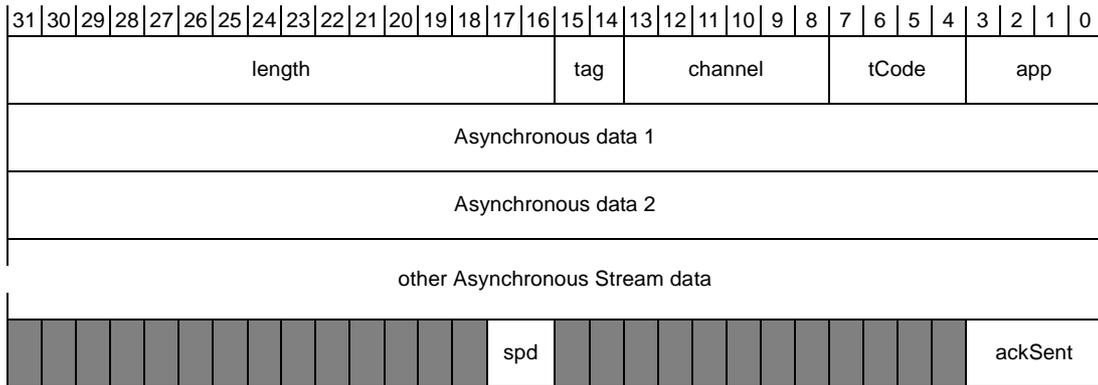


Table 17-1-8 Asynchronous Stream Receive format (Isochronous:normal)

**dataLength** : data length field

This field is used to designate the received data length for the blockData field.

**tag** : tag field

This field is used to set up Tag for Asynchronous transmit.

**channel** : channel field

This field is used to set up Channel number for Asynchronous transmit.

**app** : app field

This field is used to set up app data for Isochronous transmit.

**tCode** : Transaction code field

A transaction code is saved in this field. The transaction code values is 'Ah'.

**Asynchronous Stream data:** Asynchronous Stream data field

This field is used to set up actual transfer data.

**spd** : speed field

This field is used to designate the received speed. Refer to Table 17-8-6 regarding setting values.

**ackSent** : ackSent field

This field saves the Ack code returned as an acknowledge signal after this packet has been received. Refer to Table 17-8-7 regarding details.

**17-1-8 Asynchronous Bus Format**

Following format is performed when the packet data is written in ATF or is read out from ARF from Asynchronous Bus. As Async Bus is 16-bit width for 1 Quadlet data, upper 16 bit and lower 16 bit in due order.

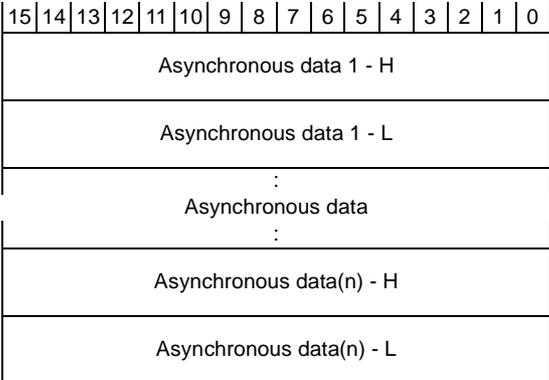


Table 17-1-9 Asynchronous Bus format

## 17-2 Isochronous Bus

### 17-2-1 For Normal Mode

#### 17-2-1-1 Transmit

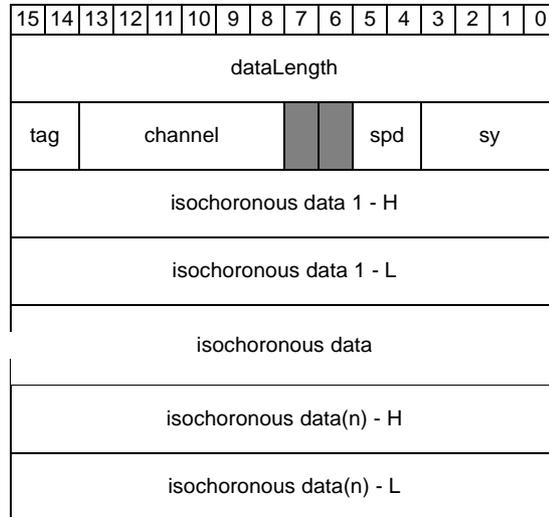


Table 17-2-1 Block Transmit format (Isochronous:normal)

**dataLength** : data length field

This field is used to set up data length for the blockData field.

**tag** : tag field

This field is used to set up Tag for Isochronous transmit.

**channel** : channel field

This field is used to set up Channel number for Isochronous transmit.

**spd** : speed field

This field is used to designate the transfer speed. Refer to Table 17-8-6 regarding setting values.

**sy** : sync field

This field is used to set up Sync data for Isochronous transmit.

**Isochronous Data** : Isochronous data field

This field is used to set up actual transfer data. If the dataLength field is not defined with a multiple of 4, this field need be filled with 00h to complete it in the Quadlet unit.

17-2-1-2 Receive

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
dataLength															
tag				channel				tCode				sy			
isochronous data 1 - H															
isochronous data 1 - L															
:															
isochronous data															
:															
isochronous data(n) - H															
isochronous data(n) - L															

Table 17-2-2 Block Receive format (Isochronous:normal)

**dataLength** : data length field  
 This field is used to designate the received data length for the blockData field.

**tag** : tag field  
 This field is used to set up Tag for Isochronous transmit.

**channel** : channel field  
 This field is used to set up Channel number for Isochronous transmit.

**sy** : sync field  
 This field is used to set up Sync data for Isochronous transmit.

**tCode** : Transaction code field  
 A transaction code is saved in this field. The transaction code values is 'Ah'.

**isochronous Data** : isochronous data field  
 This field is used to set up actual transfer data.

## 17-2-2 For Auto Mode

### 17-2-2-1 Transmit

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
isochronous data 1 - H															
isochronous data 1 - L															
:															
isochronous data															
:															
isochronous data(n) - H															
isochronous data(n) - L															

Table 17-2-3 Block Transmit format (Isochronous:auto)

#### **isochronous Data:** Isochronous data field

This field is used to set up actual transfer data. If the dataLength field is not defined with a multiple of 4, this field need be filled with 00h to complete it in the Quadlet unit.

### 17-2-2-2 Receive

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
isochronous data 1 - H															
isochronous data 1 - L															
:															
isochronous data															
:															
isochronous data(n) - H															
isochronous data(n) - L															

Table 17-2-4 Block Receive format (Isochronous:auto)

#### **isochronous Data:** Isochronous data field

This field is used to set up actual transfer data.

**17-3 Isochronous Buffer (ITF/IRF)**

**17-3-1 For Normal Mode**

**17-3-1-1 Transmit**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
length															tag	channel					spd		sync								
Isochronous data 1																															
Isochronous data 2																															
:																															
other Isochronous data																															
:																															
															padding (if necessary)																

Table 17-3-1 Block Transmit format (Isochronous:normal)

**17-3-1-2 Receive**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
length															tag	channel					tCode			sync							
Isochronous data 1																															
Isochronous data 2																															
:																															
other Isochronous data																															
:																															
															spd										ackSent						

Table 17-3-2 Block Receive format (Isochronous:normal)

**spd** : speed field

This field is used to designate the received speed. Refer to Table 17-8-6 regarding setting values.

**ackSent** : ackSent field

This field saves the Ack code returned as an acknowledge signal after this packet has been received. Refer to Table 17-8-7 regarding details.

**17-3-2 For Auto Mode**

**17-3-2-1 Transmit**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Isochronous data 1																															
Isochronous data 2																															
:																															
Isochronous data-n																															

Table 17-3-3 Block Receive format (Isochronous:normal)

**isochronous Data:** Isochronous data field

This field is used to set up actual transfer data. If the dataLength field is not defined with a multiple of 4, this field need be filled with 00h to complete it in the Quadlet unit.

**17-3-2-2 Receive**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Isochronous data 1																															
Isochronous data 2																															
:																															
Isochronous data-n																															

Table 17-3-4 Block Receive format (Isochronous:auto)

**isochronous Data:** Isochronous data field

This field is used to set up actual transfer data.

### 17-4 Snoop

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
snooped data 1 - H															
snooped data 1 - L															
:															
snooped data															
:															
snooped data(n) - H															
snooped data(n) - L															

Table 17-4-1 Snoop Receive format

**snoopedData** : snooped data field

This field is used to save the snooped data. IsochronousPacket, AsynchronousPacket, and Phy Packet data which has been received by snoop reception are read out from IsochrnousBus. In case of Ack code of Asynchronous Packet, the speed code of Packet is also read out.

### 17-5 MyNodeID

The format to output MyNodeID to IsoBus is as follows:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
r															
Bus Number										Node Number					

Table 17-5-1 MyNodeID Output Format

**r** :root bit  
Nod ID of PHY chip which has transferred this packet.

**Bus Number** :Bus Number field  
Bus Number of Node Identification Register which is set in MD8415 at present.

**Node Number** :Node Number field  
Physical ID of Phy to be connected

**17-6 SelfID Packet**

**17-6-1 ARF Buffer**

After the identification quadlet data shown in Table 4-4-1 have been saved, an actual SelfID packet is saved.

This operation is completed with the last quadlet ID data as shown in Table 4-4-4.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 17-6-1 SelfID Packet Receive format (first quadlet)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	phy_ID						0	L	gap_cnt				sp	del	c	pwr	p0	p1	p2	i	m									
logical inverse of first quadlet																															

Table 17-6-2 SelfID Packet Receive format (SelfID Packet #0)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	phy_ID						1	n	rsv	pa	pb	pc	pd	pe	pf	pg	ph	r	m											
logical inverse of first quadlet																															

Table 17-6-3 SelfID Packet Receive format (SelfID Packet #1, #2, & #3)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 17-6-4 SelfID Packet Receive format (last quadlet)

	n	pa	pb	pc	pd	pe	pf	pg	ph
pkt #1	0	p3	p4	p5	p6	p7	p8	p9	p10
pkt #2	1	p11	p12	p13	p14	p15	p16	p17	p18
pkt #3	2	p19	p20	p21	p22	p23	p24	p25	p26

Table 17-6-5 SelfID Packet Receive format (pn)

**phy\_ID** : physical\_ID field  
 A node ID of the PHY chip used to send this packet.

**L** : link\_active field  
 0 = LINK is not active.  
 1 = Active link and transaction layer are present in this node.

**gap\_cnt** : gap\_count field  
 A present value of the PHY\_CONFIGURATION.gap\_count field for this node is saved.

---

<b>sp</b>	: PHY_SPEED field.Preliminary MD8412
	00 = 98.304Mbps
	01 = 98.304 and 196.608Mbps
	10 = 98.304 and 196.608 and 393.216Mbps
	11 = Reserved
	Available speeds are saved.
<b>del</b>	: PHY_DELAY field
	00 = 144ns or less (~14/BASE_RATE)
	01~11 = Reserved.
	The delay time of the repeater in the worst case is saved.
<b>C</b>	: CONTENDER field
	When this field is set and the link_active field is also set, this node indicates that it can be a bus or isochronous resource manager.
<b>pwr</b>	: POWER_CLASS field
	000 = The node does not require power supply.
	001 = The node does has its own power supply that can feed a minimum of 15W.
	010 = The node does has its own power supply that can feed a minimum of 30W.
	011 = The node does has its own power supply that can feed a minimum of 45W.
	100 = The node consumes a maximum of 1W of power from the cable. In addition, it consumes a maximum of 2W to enable the LINK and upper layers.
	101 = The node consumes a maximum of 1W of power from the cable. In addition, it consumes a maximum of 2W to enable the LINK and upper layers.
	110 = The node consumes a maximum of 1W of power from the cable. In addition, it consumes a maximum of 5W to enable the LINK and upper layers.
	111 = The node consumes a maximum of 1W of power from the cable. In addition, it consumes a maximum of 9W to enable the LINK and upper layers.
<b>p0 ... p26</b>	: NORT,child[NPORT],connected[NPORT]field
	11 = Connected to the parent node.
	10 = Connected to the parent node.
	01 = Not connected to another PHY.
	00 = This PHY is not offered.
	The port status is shown.
<b>i</b>	: initiated_reset field
	If it is set, this node has issued present bus reset.
<b>m</b>	: more_packets field
	If it is set, this node indicates that another SelfID packet of this node is closely following.
<b>n</b>	: Extended field
	An extension SelfID packet sequence number (value from 0~2).
<b>r, rsv</b>	: reserved field
	Reserved.

17-6-2 IsoBus

Followings are the format when SelfID packet is output into IsoBus;

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1110b				0	0	0	0
1	0	phy_ID					0	L	gap_cnt						
sp	del	C	pwr		p0		p1	p2	i	m					
logical inverse of first quadlet															
logical inverse of first quadlet															
1	0	phy_ID					1	n	rsv	pa					
pb	pc	pd	pe	pf	pg	ph	r	m							
logical inverse of first quadlet															
logical inverse of first quadlet															
:															
:															
1	0	phy_ID					o	L	gap_cnt						
sp	del	C	pwr		p0		p1	p2	i	m					
logical inverse of first quadlet															
logical inverse of first quadlet															
1	0	phy_ID					1	n	rsv	pa					
pb	pc	pd	pe	pf	pg	ph	r	m							
logical inverse of first quadlet															
logical inverse of first quadlet															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	ackSent			

Table 17-6-6 SelfID Packet Output format



## 17-8 Code

The codes used for packet formatting specified by 1394 are shown. Refer to the 1394 for details of each code.

Code	Name
00b	retry_1
01b	retry_X
10b	retry_A
11b	retry_B

Table 17-8-1 List of Retry code

Code	Name	Code	Name
0h	write request for data quadlet	8h	cycle start
1h	write request for data block	9h	lock request
2h	write response	Ah	Isochronous data block/Asynchronous S
3h	reserved	Bh	lock response
4h	read request for data quadlet	Ch	reserved
5h	read request for data block	Dh	reserved
6h	read response for data quadlet	Eh	reserved
7h	read response for data block	Fh	reserved

Table 17-8-2 List of Transaction code (tCode)

destination bus_ID	destination node_ID	Contents
0 ~ 3FEh	0 ~ 3Eh	Transferred to the node defined by bus_ID and node_ID.
3FFh	0 ~ 3Eh	Transferred to the node defined by node_ID in local bus.
0 ~ 3FEh	3Fh	Broadcast transfer to the bus defined by bus_ID.
3FFh	3Fh	Broadcast transfer in local bus.

Table 17-8-3 List of Bus Number / Node Number

Data rate	Maximum payload size (byte)
100Mbps	512
200Mbps	1024
400Mbps	2048

Table 17-8-4 List of Data Length (Data Length)

Code	Name
0000h	reserved
0001h	mask_swap
0002h	compare_swap
0003h	fetch_add
0004h	little_add
0005h	bounded_add
0006h	wrap_add
0007h	vender_dependent
0008h ~ FFFFh	reserved

Table 17-8-5 List of Extension Transaction Code (Extend tCode)

Code	Speed
00b	100Mbps
01b	200Mbps
10b	400Mbps
11b	reserved

Table 17-8-6 List of Speed Codes (spd)

Code	Name
0h	reserved
1h	ack_complete
2h	ack_pending
3h	reserved
4h	ack_busy_X
5h	ack_busy_A
6h	ack_busy_B
7h	reserved
8h	reserved
9h	reserved
Ah	reserved
Bh	reserved
Ch	reserved
Dh	ack_data_error
Eh	ack_type_error
Fh	ack_address_error

Table 17-8-7 List of Acknowledge Codes (Ack)

Code	Name
0h	resp_complete
1h	reserved
2h	reserved
3h	reserved
4h	resp_conflict_error
5h	resp_data_error
6h	resp_type_error
7h	resp_address_error
8h ~ Fh	reserved

Table 17-8-8 List of rcodes

## 18 Electrical Characteristics

### 18-1 Absolute Rating

(VSS = 0V)

Symbol	Parameter	Rating	Units
VDD	Supply Voltage	-0.3 ~ +7.0	V
VIN	Input Voltage	-0.8 ~ 5.7	V
IIN	Input Current	±10	mA
TSTG	Storage Temp.	-40 ~ +125	°C

### 18-2 Recommended Operating Condition

(VSS = 0V)

Symbol	Parameter	Rating	Units
VDD	Supply Voltage	3.00 ~ 3.60	V
VIN	Input Voltage	0 ~ 5.0	V
TA	Ambient temp.	0 ~ +70	°C

### 18-3 DC Characteristics

DC characteristics under the recommended conditions except special mention.

(VSS=0V)

Symbol	Item	Pin	Test Condition	MIN	TYP	MAX	Unit
VIH	High Level Input Voltage	SCLK, CTL, D		VDD-0.7		VDD+0.7	V
		RESET#		2.4		5.7	
		Except above		2.0		5.7	
VIL	Low Level Input Voltage	SCLK, CTL, D				VSS+0.7	V
		else				0.8	
IIH	High Level Input Current		VIN= VDD	-10		10	μA
IIL	Low Level Input Current		VIN= VSS	-10		10	μA
VOH	High Level Output Voltage	LPS, LREQ, CTL, D	IOH= -12mA	2.4			V
		Signal for (Note)	IOH= -10mA	2.4			
		Except above	IOH= -6mA	2.4			
VOL	Low Level Output Voltage	LPS, LREQ, CTL, D	IOL= 12mA			0.5	V
		Signal for (Note)	IOL= 10mA			0.5	
		Except above	IOL= 6mA			0.5	
IOZ	Output Disable Current		VOUT= VDD or VSS	-10		10	μA
IDD	Dynamic Power Supply Current	VDD	VDD= 3.3V		90	130	mA

(Note) : BE#(1:0), SYNC(1:0), BUSRST, IRERR#, IRCV#, IRX#, ICLK, IDATA(15:0), ITX#, CT, ICS, CYCLEOUT

All signal terminals except SCLK, CTL, and D are possible to connect directly with 5V system logic.

## 18-4 AC Characteristics

Symbol	Item	MIN	TYP	MAX	Unit
TCSS	CS#, DACK# setup time	5			nS
TADS	HA, UWE#, UBE# setup time	10			nS
TADH	HA, UWE#, UBE# holding time	0			nS
TCSWH	CS#, DACK# holding time (while WRITE)	0			nS
TCSRH	CS#, DACK# holding time (while READ)	0			nS
TDREQWD	DREQ output lagging time(while WRITE)	18		54	nS
TDREQRD	DREQ output lagging time(while READ)	18		54	nS
TRWL	READ WRITE pulse width low	80			nS
TRWH	READ WRITE pulse width high	40			nS
TRWC	READ/WRITE cycle time	160			nS
TDTD	Read data output lagging time	40		80	nS
TDTH	Read data output holding time	5		15	nS
TWRDS	Write data setup time	-40			nS
TWRDH	Write data holding time	0			nS
TRSW	Reset pulse width	160			nS
TCYCD	CYCLEOUT signal output lagging time	5		18	nS
TCYCH	CYCLEOUT signal high level time		62.5		μS
TCYCL	CYCLEOUT signal low level time		62.5		μS

(Load Capacitor 50pF)

Table 18-4-1 Host Interface AC Characteristics

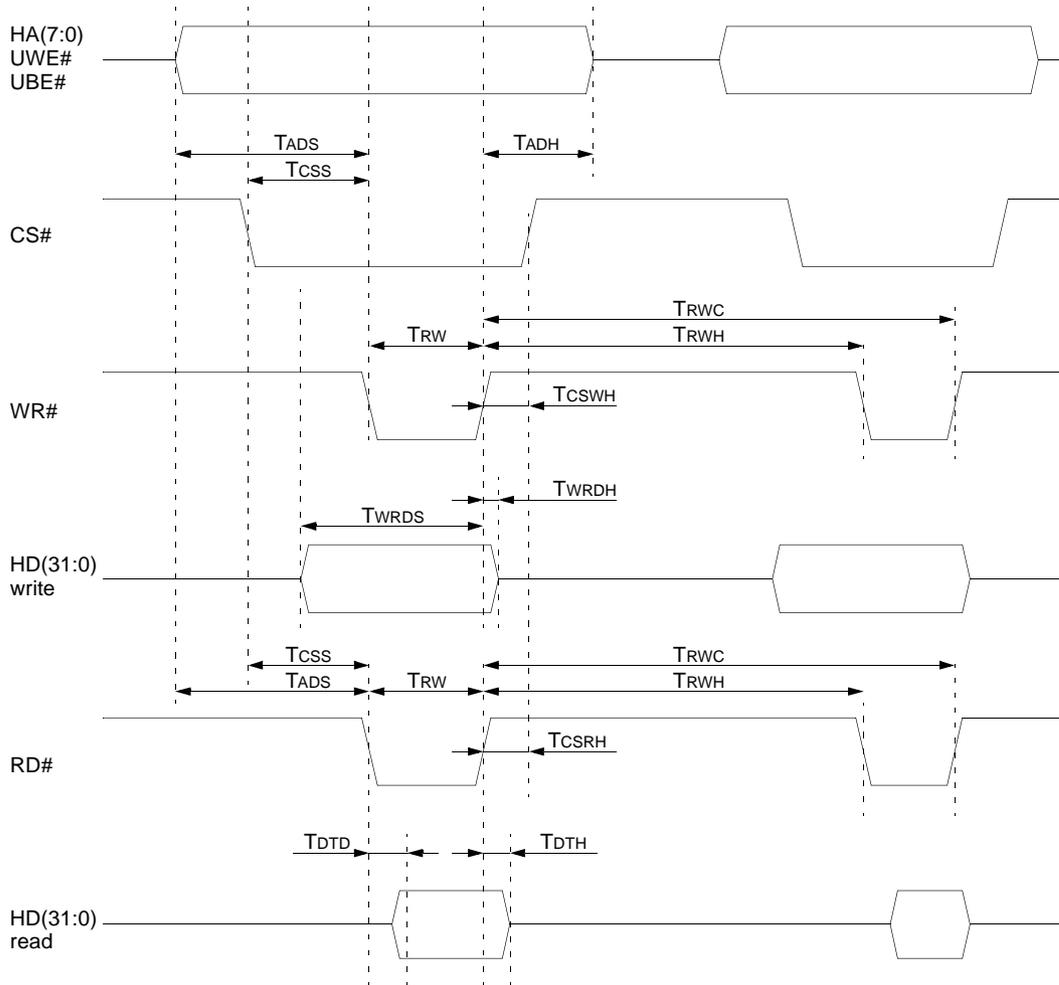


Figure 18-4-1 Host Interface AC Characteristics (Read/Write)

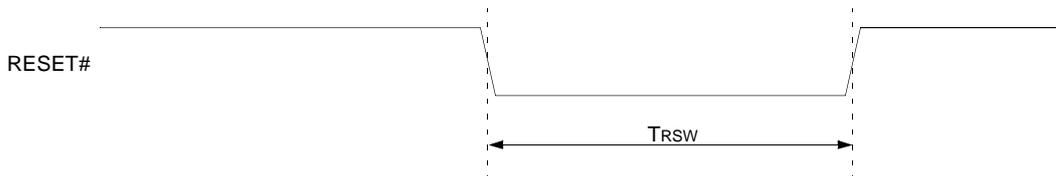


Figure 18-4-2 Host Interface AC Characteristics (Reset)

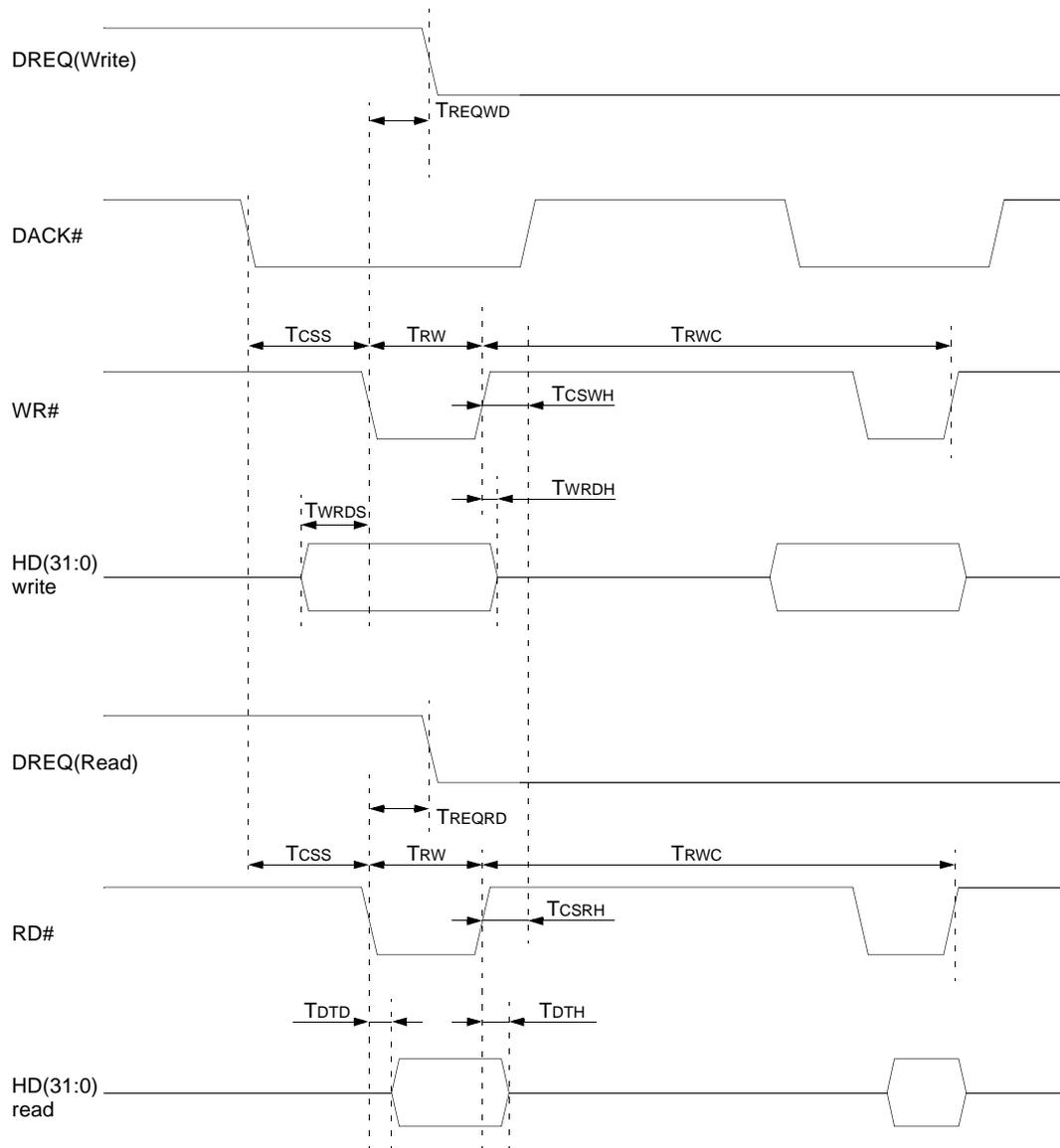


Figure 18-4-3 Host Interface AC Characteristics (DMA)

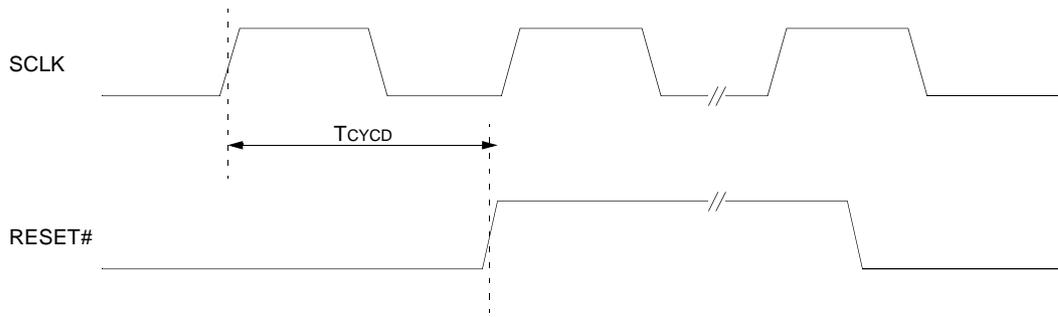


Figure 18-4-4 Host Interface AC Characteristics (CYCLEOUT)

Symbol	Item	MIN	TYP	MAX	Unit
TCTD	Control output lagging time	3		10	nS
TPDD	PHY data output lagging time	3		10	nS
TCTS	Control setup time	6			nS
TCTH	Control holding time	0			nS
TPDS	PHY data setup time	6			nS
TPDH	PHY data holding time	0			nS
TLRD	LREQ data output lagging time	3		10	nS
TSCKC	SCLK cycle time	20			nS
TSCKH	SCLK high level time	8		12	nS
TSCKL	SCLK low level time	8		12	nS
TLPSC	LPS cycle time	272		1500	nS
TLPSH	LPS high level time	90		500	nS
TLPSL	LPS low level time	180		1000	nS

(Condition: Load Capacitor 20pF)

Table 18-4-2 PHY Interface AC Characteristics

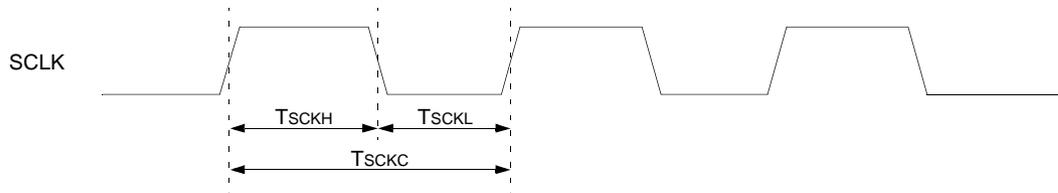


Figure 18-4-5 PHY Interface AC Characteristics (SCLK)

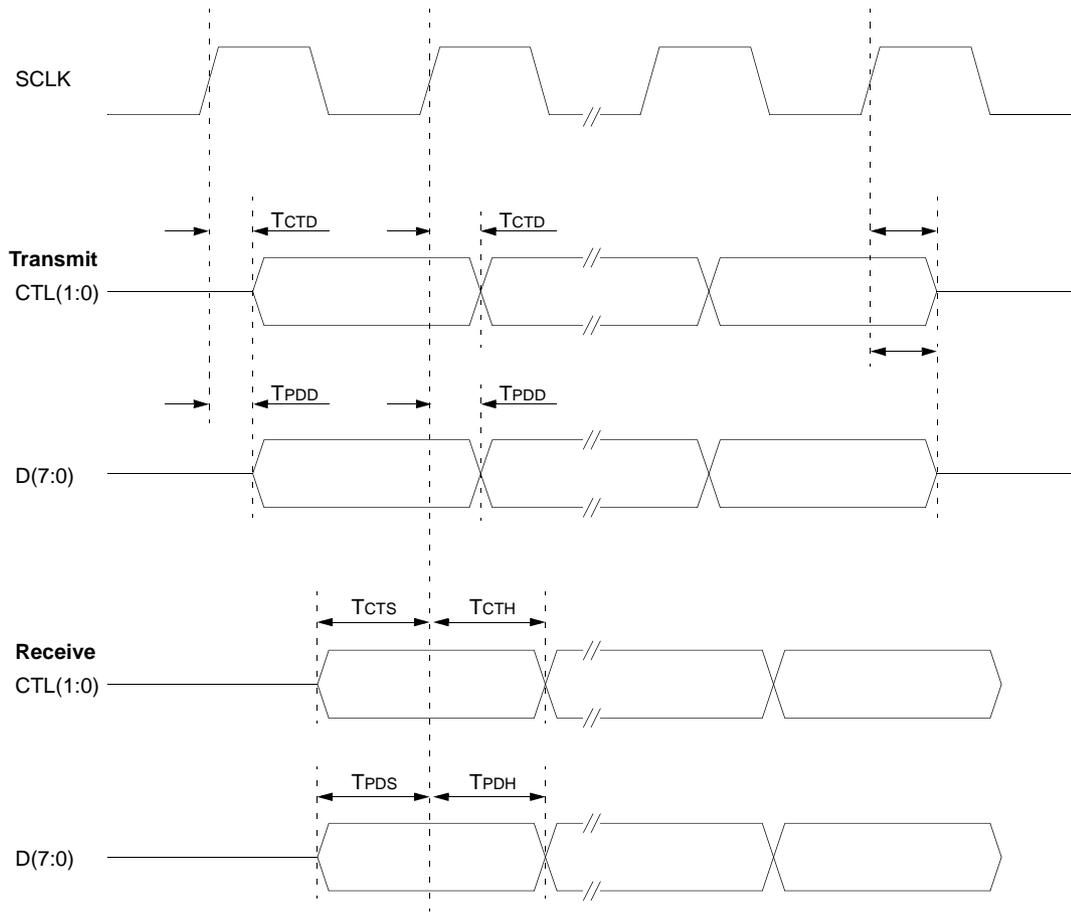


Figure 18-4-6 PHY Interface AC Characteristics (CTL,D)

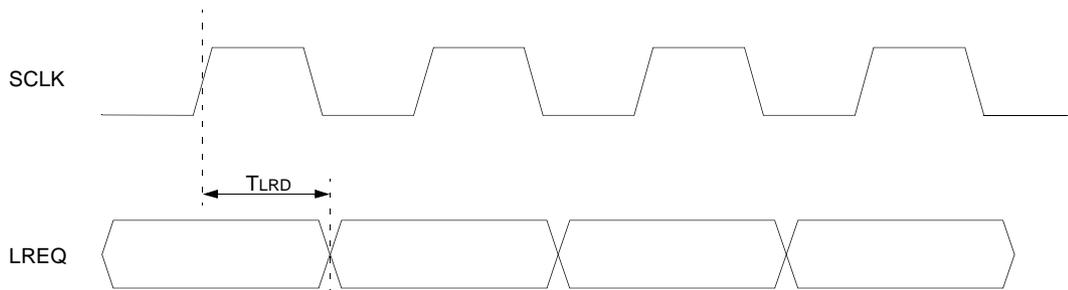


Figure 18-4-7 PHY Interface AC Characteristics (LREQ)

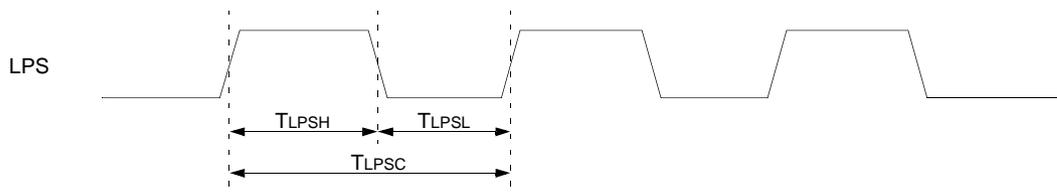


Figure 18-4-8 PHY Interface AC Characteristics (LPS)

Symbol	Item	MIN	TYP	MAX	Unit
TICKD	ICLK output lagging time	5		15	nS
TICSD	ICS output lagging time	1		10	nS
TITRS	ITREQ# setup time	20			nS
TITRH	ITREQ# holding time	0			nS
TITXD	ITX# output lagging time	1		10	nS
TIEPS	IEOP# setup time	20			nS
TIEPH	IEOP# holding time	0			nS
TIDTS	IDATA[15:0] write setup time	13			nS
TIDTH	IDATA[15:0]write holding time	0			nS
TIRXD	IRX# output lagging time	1		10	nS
TIRVD	IRCV# output lagging time	1		10	nS
TIDFD	IDATA[15:0] output lagging time(IRCV#)	3		12	nS
TIDTD	IDATA[15:0] output lagging time(ICLK#)	2		14	nS
TIDRD	IDATA[15:0] output holding tim	3		12	nS
TBCSD	BE#,CH,SYNC output lagging time	1		10	nS
TIERD	IRERR# output lagging time	1		10	nS
TCTD	CT output lagging time	1		10	nS
TIBRSTD	BUSRST output lagging time	1		10	nS

(Condition: Load Capacitor50pF)

Table 18-4-3 Isochronous Bus AC Characteristics (For Master)

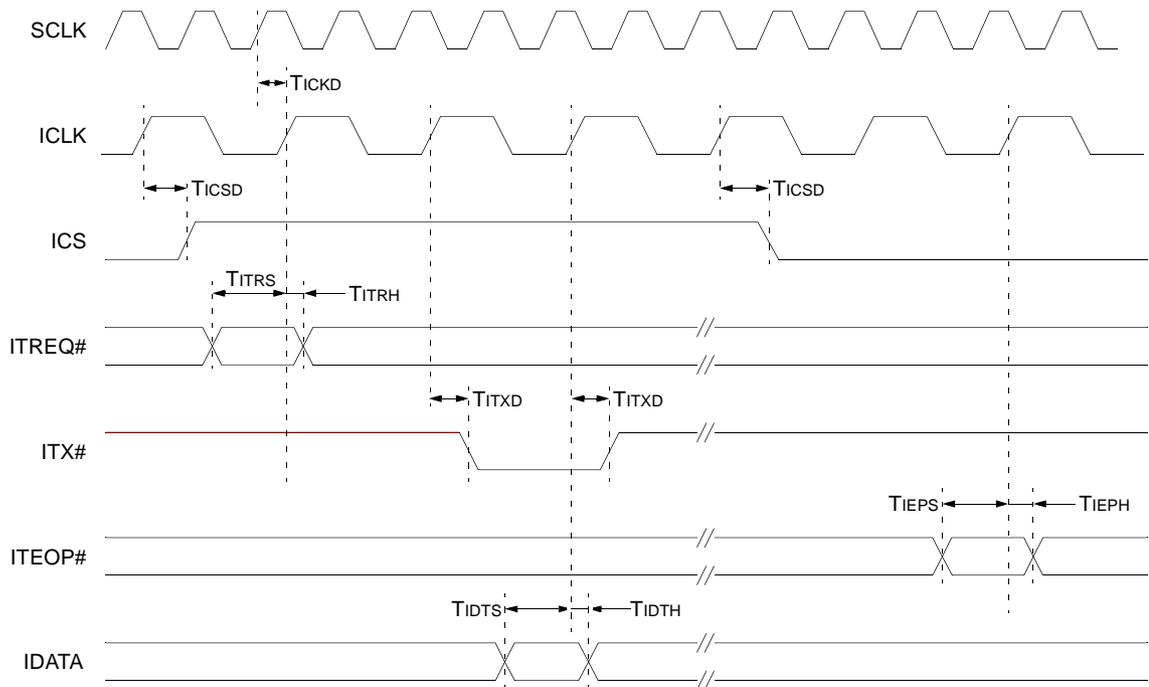


Figure 18-4-9 Isochronous Bus AC Characteristics (For Master) 1

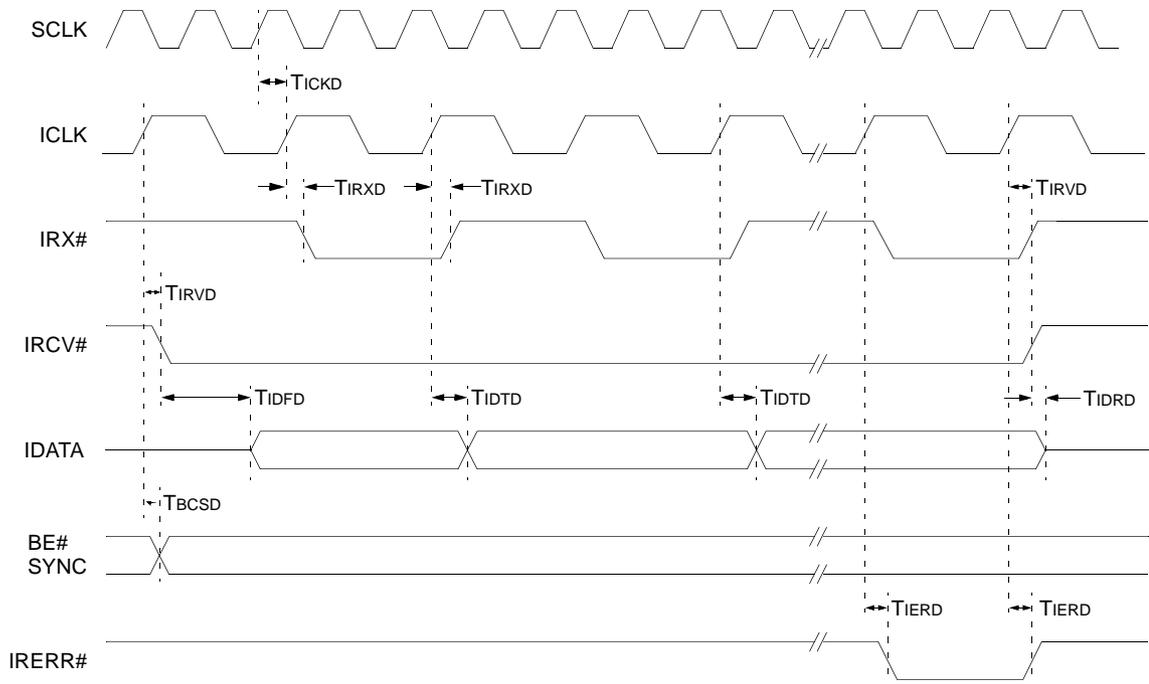


Figure 18-4-10 Isochronous Bus AC Characteristics (For Master) 2

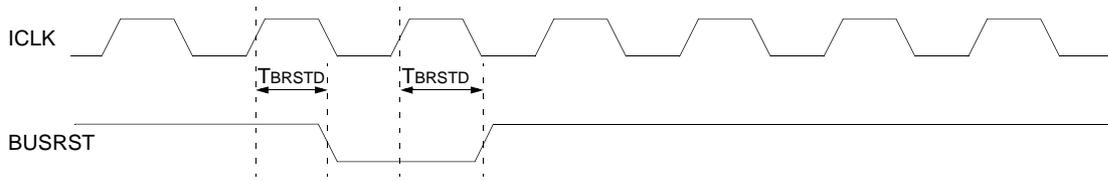


Figure 18-4-11 Isochronous Bus AC Characteristics 3

Symbol	Item	MIN	TYP	MAX	Unit
TITXS	ITX# setup time	10			nS
TITXH	ITX# holding time	0			nS
TITEND	ITEN# output lagging time	4		16	nS
TIEPS	IEOP# setup time	10			nS
TIEPH	IEOP# holding time	0			nS
TIDTS	IDATA[15:0] write setup time	10			nS
TIDTH	IDATA[15:0] write holding time	0			nS
TIRPKTD	IRPKT# output lagging time	4		16	nS
TIRXS	IRX# setup time	10			nS
TIRXH	IRX# holding time	0			nS
TIDFD	IDATA[15:0] output lagging time (IRPKT#)	4		22	nS
TIDTD	IDATA[15:0] output lagging time (ICLK#)	4		22	nS
TIDRD	IDATA[15:0] output holding time	4		12	nS

(Condition: Load Capacitor50pF)

Table 18-4-4 Isochronous Bus AC Characteristics (For Slave)

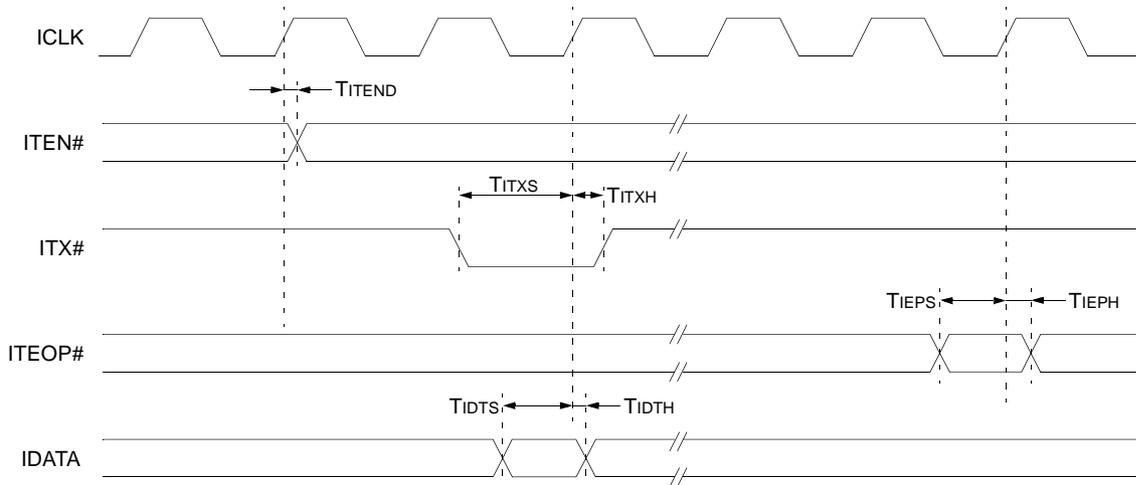


Figure 18-4-12 Isochronous Bus AC Characteristics (For Slave) 1

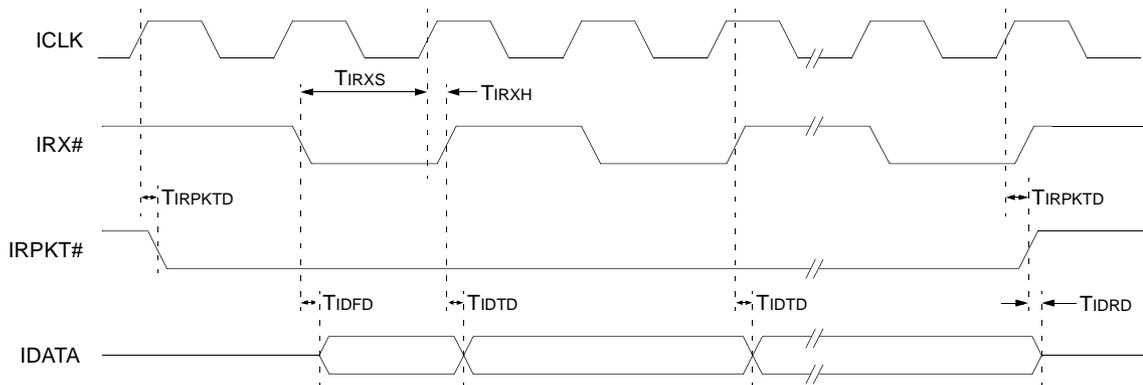


Figure 18-4-13 Isochronous Bus AC Characteristics (For Slave) 2

Symbol	Item	MIN	TYP	MAX	Unit
TATXS	ATX# setup time	10			nS
TATXH	ATX# holding time	0			nS
TATEND	ATEN# output lagging time	4		16	nS
TAEPS	AEOP# setup time	10			nS
TAEPH	AEOP# holding time	0			nS
TADTS	ADATA[15:0] write setup time	10			nS
TADTH	ADATA[15:0] write holding time	0			nS
TARPKTD	ARPKT# output lagging time	4		16	nS
TARXS	ARX# setup time	10			nS
TARXH	ARX# holding tim	0			nS
TADFD	ADATA[15:0] output lagging time (IRPKT#)	4		22	nS
TADTD	ADATA[15:0] output lagging time (ICLK#)	4		22	nS
TADRDR	ADATA[15:0] output holding time	4		12	nS

(Condition: Load Capacitor 50pF)

Table 18-4-5 Asynchronous Bus AC Characteristics

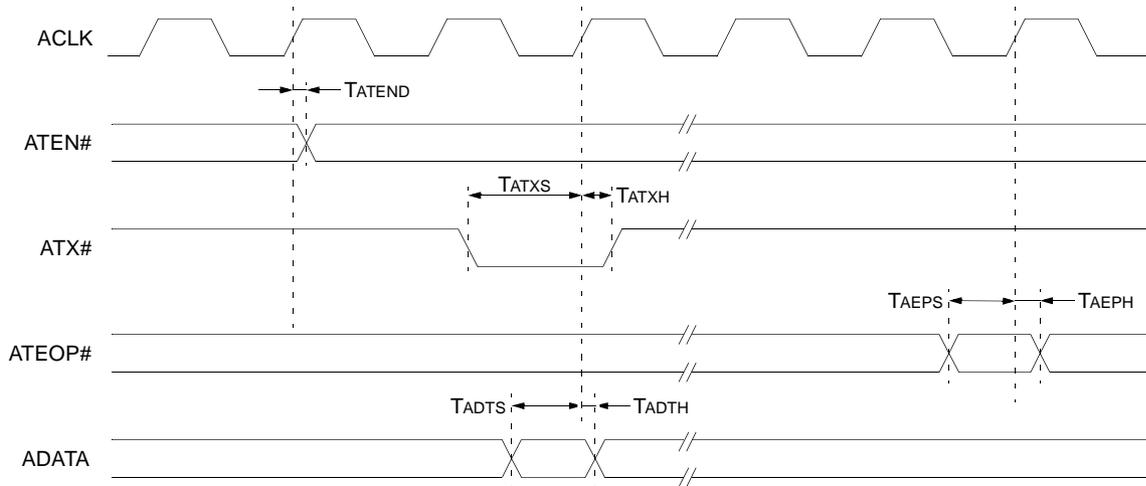


Figure 18-4-14 Asynchronous Bus AC Characteristics 1

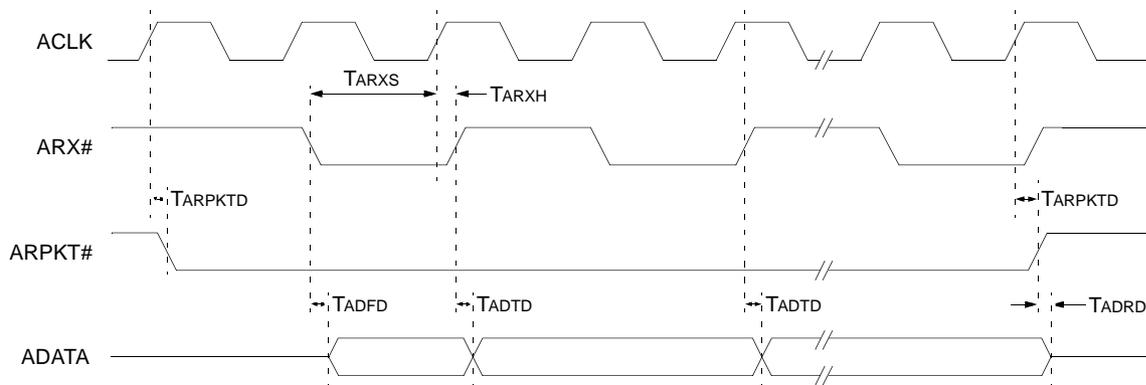


Figure 18-4-15 Asynchronous Bus AC Characteristics 2

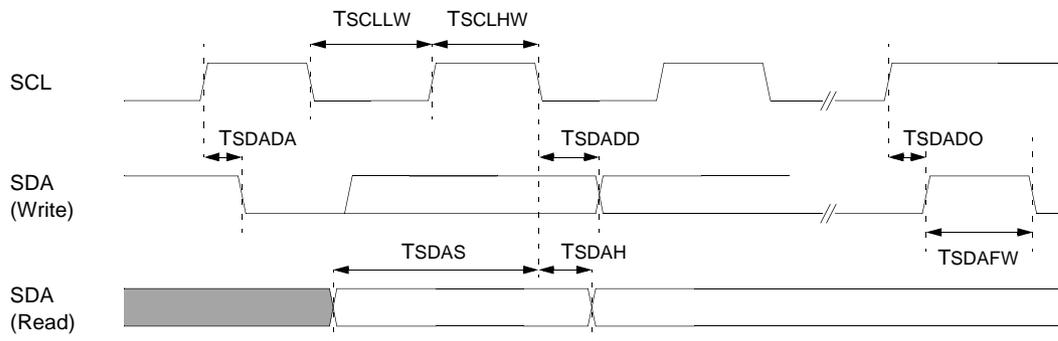
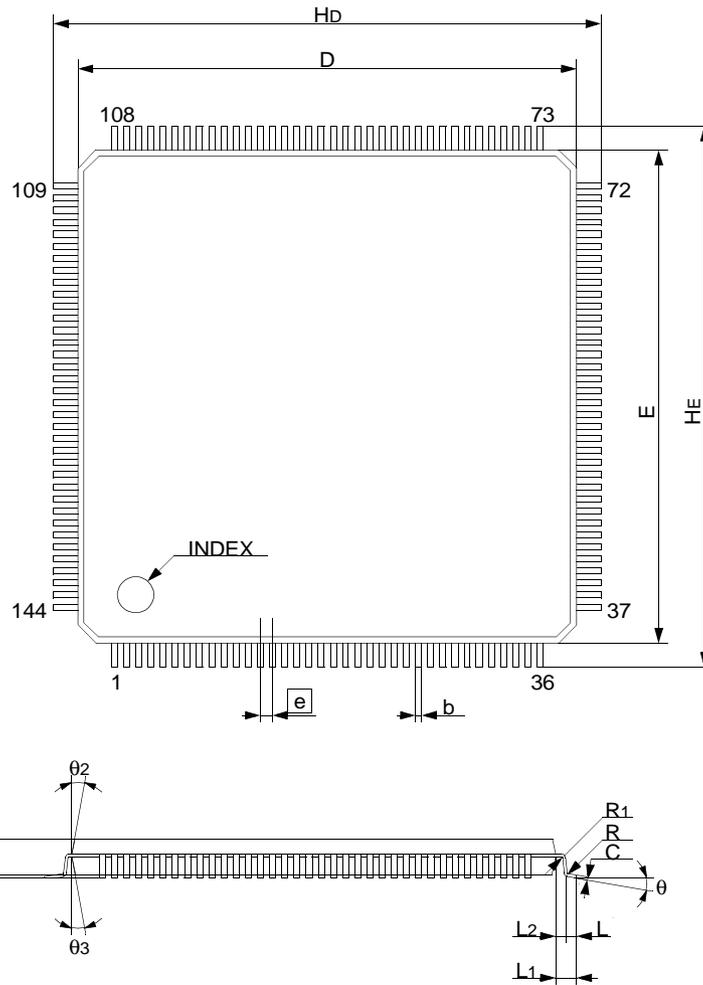


Figure 18-4-16 EEPROM Interface AC Characteristics

Symbol	Item	MIN	TYP	MAX	Unit
TSCLLW	Clock low level time	5000			nS
TSCLHW	Clock high level time	5000			nS
TSDADA	Data output lagging time	5000			nS
TSDADD	Data output lagging time	80		150	nS
TSDADO	Data output lagging time	5000			nS
TSDAFW	Data Hiz output time	10000			nS
TSDAS	Data setup time	0			nS
TSDAH	Data holding time	100			nS

Table 18-4-6 EEPROM Interface AC Characteristics

19 Package Outline



Lead type STD (QFP20-144pin STD)						
Symbol	Dimension in Millimeters			Dimension in Inches*		
	Min.	Nom.	Max.	Min.	Nom.	Max.
E	19.9	20	20.1	(0.784)	(0.787)	(0.791)
D	19.9	20	20.1	(0.784)	(0.787)	(0.791)
A			1.7			(0.066)
A1		0.1			(0.004)	
A2	1.3	1.4	1.5	(0.052)	(0.055)	(0.059)
		0.5			(0.020)	
b	0.15	0.2	0.3	(0.006)	(0.008)	(0.011)
C	0.1	0.125	0.175	(0.004)	(0.005)	(0.006)
q	0°		10°	(0°)		(10°)
L	0.3	0.5	0.7	(0.012)	(0.020)	(0.027)
L1		1			(0.039)	
L2		0.5			(0.020)	
HE	21.6	22	22.4	(0.851)	(0.866)	(0.881)
HD	21.6	22	22.4	(0.851)	(0.866)	(0.881)
theta 2		12°			(12°)	
theta 3		12°			(12°)	
R		0.2			(0.008)	
R1		0.2			(0.008)	

\* for reference

## Notes

- Contents of this manual may be modified without notice.
- Partial or overall reproduction and reprinting of this manual shall be prohibited without permission of the company.
- Owing this manual shall not fall out permission for the use of industrial and copy rights reserved by the company and Fuji Photo Film Co., Ltd. The company shall not be responsible for industrial and copy rights of the third party.
- The company shall not be responsible for any damage caused from this manual and from this chip.
- Prior to use, exchange of invoice specifications is required.
- Information available from:
  - Head Office
    - 1-6, Matsusakadaira, Taiwa-cho
    - Kurokawa-gun, MIYAGI, 981-3493 JAPAN
    - TEL: 81-22-347-1128 FAX: 81-22-347-1136
  - Tokyo Office
    - Fukumasu Bldg. 6F
    - 1-20, Ageba-cho, Shinjuku-ku
    - Tokyo, 162-0824 JAPAN
    - TEL: 81-3-3269-2141 FAX: 81-3-5229-7381