SONY

CXD3220R

IEEE1394 Link/Transaction Layer Controller LSI for SBP-2

Description

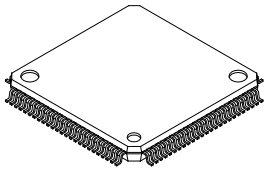
The CXD3220R is a Link/Transaction Layer LSI conforming to the IEEE1394 serial bus standard.

It is mainly used when connecting the IEEE1394 digital I/F to a storage device such as a hard disk, DVD-ROM, CD-ROM or tape streamer.

Data transfer conforms to the SBP-2 protocol.

This LSI utilizes Apple Computer's Fire Wire technology.

100 pin LQFP (Plastic)



Features

- Conforms to IEEE1394 serial bus standard
- Conforms to SBP-2 (serial bus protocol-2)
- Compatible with bidirectional data transfer of computer peripherals
- Compatible with 1394 transfer rate at 100/200Mbps
- Dedicated Asynchronous data transfer
- High-speed data transfer through the use of an ADP (automatic data pipe) circuit
- Cycle master function
- Direct connection to 1394 Phy chip
- Large capacity FIFO

Data transfer FIFO	532 quadlets
Asynchronous Transmit FIFO	24 quadlets
Asynchronous Receive FIFO	39 quadlets

Applications

Digital interface for computer peripheral

Structure

Silicon gate CMOS IC

Absolute Maximum Ratings (Ta = 25°C)

• Supply voltage	V_{DD}	V_{SS} – 0.5 to +4.6	V
• Input voltage	V_I	V_{SS} – 0.5 to V_{DD} + 0.5	V
• Output voltage	V_O	V_{SS} – 0.5 to V_{DD} + 0.5	V
• Operating temperature	Topr	–20 to +75	°C
• Storage temperature	Tstg	–55 to +150	°C

Recommended Operating Conditions

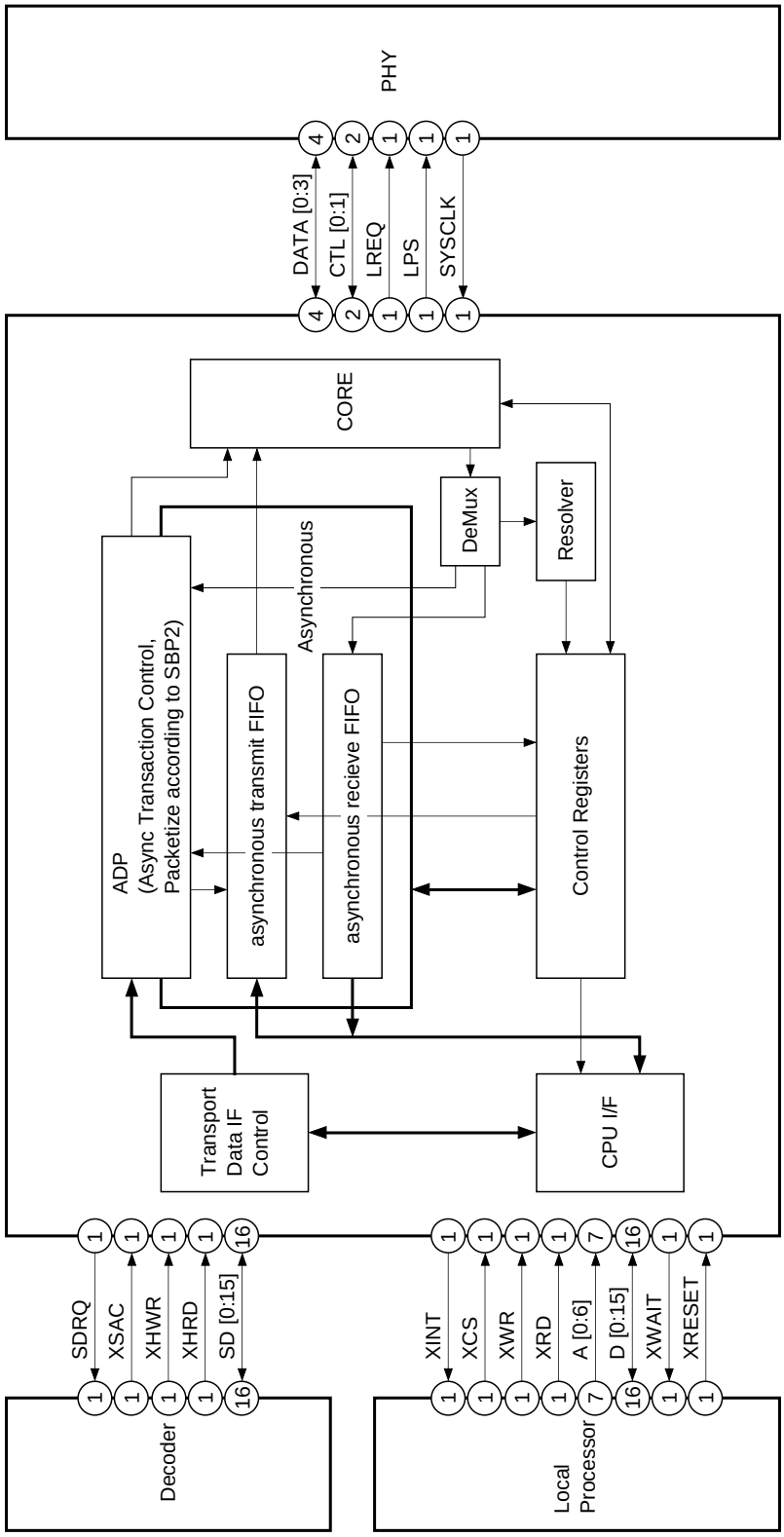
• Supply voltage	V_{DD}	3.0 to 3.6	V
• Operating temperature	Topr	–20 to +75	°C

E97320-PS

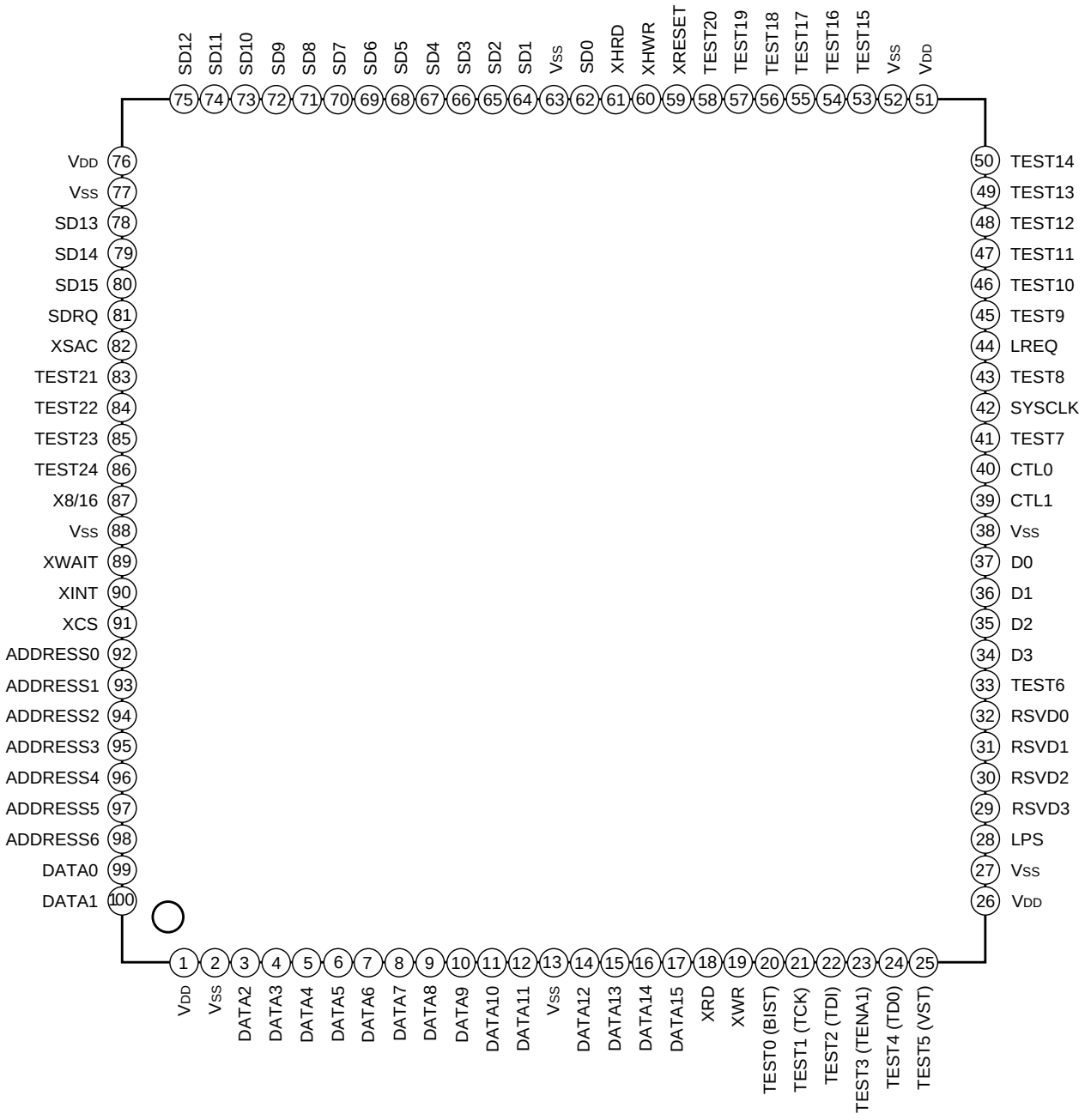
Contents

1. Block Diagram 3
2. Pin Configuration 4
3. Pin Description 5
4. Electrical Characteristics 8
   4-1. DC Characteristics 8
   4-2. AC Characteristics 8
   4-3. Input/Output Capacitance 8
   4-4. Timing Definition 9
5. System Configuration Example 10
   5-1. System Block Diagram 10
   5-2. System Connection Diagram 11
6. Asynchronous Communication 12
   6-1. CPU I/F 12
   6-2. CFR 15
   6-3. Asynchronous Packet Transmission 25
   6-4. Asynchronous Packet Reception 28
   6-5. CXD3220R Data Format 33
   6-6. Self-ID Packet Reception Error Processing 43
7. ADP (Asynchronous Data Pipe) 44
   7-1. Built-in FIFO 44
   7-2. Transport Data I/F 44
   7-3. ADP 47
   7-4. ADP Structure and Functions 49
   7-5. ADP Setting 51
8. Link-Phy Communication 58
   8-1. Link-Phy Interface Specifications 58
   8-2. Communication 58

1. Block Diagram



2. Pin Configuration



3. Pin Description

Pin No.	Symbol	I/O	Description
1	V_{DD}	—	Power Supply
2	V_{SS}	—	GND
3	DATA2	I/O	CPU I/F I/O data bit 2
4	DATA3	I/O	CPU I/F I/O data bit 3
5	DATA4	I/O	CPU I/F I/O data bit 4
6	DATA5	I/O	CPU I/F I/O data bit 5
7	DATA6	I/O	CPU I/F I/O data bit 6
8	DATA7	I/O	CPU I/F I/O data bit 7
9	DATA8	I/O	CPU I/F I/O data bit 8
10	DATA9	I/O	CPU I/F I/O data bit 9
11	DATA10	I/O	CPU I/F I/O data bit 10
12	DATA11	I/O	CPU I/F I/O data bit 11
13	V_{SS}	—	GND
14	DATA12	I/O	CPU I/F I/O data bit 12
15	DATA13	I/O	CPU I/F I/O data bit 13
16	DATA14	I/O	CPU I/F I/O data bit 14
17	DATA15	I/O	CPU I/F I/O data bit 15
18	XRD	I	CPU I/F read signal 0: read
19	XWR	I	CPU I/F write signal 0: write
20	TEST0	—	Test pin*1
21	TEST1	—	Test pin*1
22	TEST2	—	Test pin*1
23	TEST3	—	Test pin*1
24	TEST4	—	Test pin*1
25	TEST5	—	Test pin 2*2
26	V_{DD}	—	Power supply
27	V_{SS}	—	GND
28	LPS	O	Phy I/F Link power status signal (High level when XRESET input = low)
29	RSVD3	—	Reserved*3
30	RSVD2	—	Reserved*3
31	RSVD1	—	Reserved*3
32	RSVD0	—	Reserved*3
33	TEST6	—	Test pin*1

*1 The test pins should be used open.

*2 Connect the test pin 2 to GND.

*3 RSVD0 to 3 should be used open.

Pin No.	Symbol	I/O	Description
34	D3	I/O	Phy I/F data bus bit 3
35	D2	I/O	Phy I/F data bus bit 2
36	D1	I/O	Phy I/F data bus bit 1
37	D0	I/O	Phy I/F data bus bit 0
38	V_{SS}	—	GND
39	CTL1	I/O	Phy I/F control bus bit 1
40	CTL0	I/O	Phy I/F control bus bit 0
41	TEST7	—	Test pin*1
42	SYSCLK	I	Phy I/F system clock (49.195MHz)
43	TEST8	—	Test pin*1
44	LREQ	O	Phy I/F request signal
45	TEST9	—	Test pin*1
46	TEST10	—	Test pin*1
47	TEST11	—	Test pin*1
48	TEST12	—	Test pin*1
49	TEST13	—	Test pin*1
50	TEST14	—	Test pin*1
51	V_{DD}	—	Power supply
52	V_{SS}	—	GND
53	TEST15	—	Test pin*1
54	TEST16	—	Test pin*1
55	TEST17	—	Test pin*1
56	TEST18	—	Test pin*1
57	TEST19	—	Test pin*1
58	TEST20	—	Test pin*1
59	XRESET	I	Master reset signal 0: Active; 1: Non-active
60	XHWR	I	Transport data I/F data write enable signal 0: Non-active; 1: Active
61	XHRD	I	Transport data I/F data read enable signal 0: Non-active; 1: Active
62	SD0	I/O	Transport data I/F data bus bit 0
63	V_{SS}	—	GND
64	SD1	I/O	Transport data I/F data bus bit 1
65	SD2	I/O	Transport data I/F data bus bit 2
66	SD3	I/O	Transport data I/F data bus bit 3
67	SD4	I/O	Transport data I/F data bus bit 4

*1 The test pins should be used open.

Pin No.	Symbol	I/O	Description
68	SD5	I/O	Transport data I/F data bus bit 5
69	SD6	I/O	Transport data I/F data bus bit 6
70	SD7	I/O	Transport data I/F data bus bit 7
71	SD8	I/O	Transport data I/F data bus bit 8
72	SD9	I/O	Transport data I/F data bus bit 9
73	SD10	I/O	Transport data I/F data bus bit 10
74	SD11	I/O	Transport data I/F data bus bit 11
75	SD12	I/O	Transport data I/F data bus bit 12
76	V_{DD}	—	Power supply
77	V_{SS}	—	GND
78	SD13	I/O	Transport data I/F data bus bit 13
79	SD14	I/O	Transport data I/F data bus bit 14
80	SD15	I/O	Transport data I/F data bus bit 15
81	SDRQ	O	Transport data I/F data request signal
82	XSAC	I	Transport data I/F acknowledge signal
83	TEST21	—	Test pin*1
84	TEST22	—	Test pin*1
85	TEST23	—	Test pin*1
86	TEST24	—	Test pin*1
87	X8/16	I	CPU I/F I/O data bus select signal 0: 8 bits; 1:16 bits
88	V_{SS}	—	GND
89	XWAIT	O	CPU I/F wait signal active when XCS = 0, high impedance when XCS = 1
90	XINT	O	CPU I/F interrupt signal 0: Active; 1: Non-active
91	XCS	I	CPU I/F chip select signal 0: Active; 1: Non-active
92	ADDRESS0	I	CPU I/F address bus bit 0
93	ADDRESS1	I	CPU I/F address bus bit 1
94	ADDRESS2	I	CPU I/F address bus bit 2
95	ADDRESS3	I	CPU I/F address bus bit 3
96	ADDRESS4	I	CPU I/F address bus bit 4
97	ADDRESS5	I	CPU I/F address bus bit 5
98	ADDRESS6	I	CPU I/F address bus bit 6
99	DATA0	I/O	CPU I/F I/O data bit 0
100	DATA1	I/O	CPU I/F I/O data bit 1

*1 The test pins should be used open.

4. Electrical Characteristics

4-1. DC Characteristics

(Ta = 25°C, V_{SS} = 0V)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Input voltage	V_{IH}	All input pins		0.7V_{DD}			V
	V_{IL}	All input pins				0.2V_{DD}	V
Output voltage	V_{OH}	Output pins excluding D [3:0], CTL [1:0], LREQ	I_{OH} = –4mA	V_{DD} – 0.4			V
	V_{OL}		I_{OL} = 4mA			0.4	V
	V_{OH}	D [3:0], CTL [1:0], LREQ	I_{OH} = –8mA	V_{DD} – 0.4			V
	V_{OL}		I_{OL} = 8mA			0.4	V
Input leak current	$I_{I}1$	SD [15:0], D [3:0], CTL [1:0]		–40		40	µA
	$I_{I}2$	Normal input pins		–10		10	µA
	I_{IL}	XHRD, XHWR, XRD, XRESET, XSAC, XWR		–240	–100	–40	µA
Output leak current	I_{OZ}	XWAIT (for high impedance state) V_{IN} = Vss or V_{DD}		–40		40	µA
Power supply	$I_{CC}1$	For ADP operation	V_{DD} = 3.3V		90	120	mA
	$I_{CC}2$	For ADP not operation	V_{DD} = 3.3V		50	70	mA

4-2. AC Characteristics

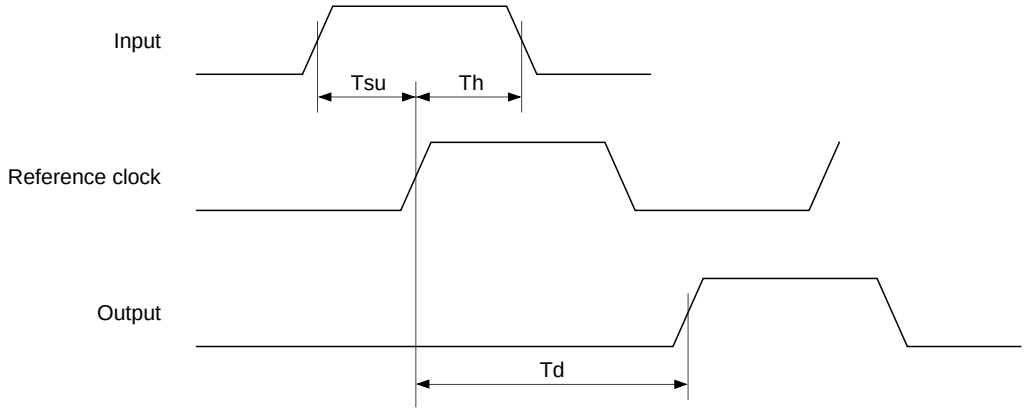
(V_{DD} = 3.0 to 3.6V)

Item	Applicable pins	Symbol	Reference clock	Conditions	Min.	Typ.	Max.	Unit
Input setup	SD [15:0], SDRQ, XSAC, XHRD, XHWR	Tsu1	Refer to 7-2. Transport data I/F write timing and Transport data I/F read timing					
Input hold		Th1						
Output delay		Td1						
Input setup	D [3:0], CTL [1:0]	Tsu2	SYSCLK	CL = 10pF	5			ns
Input hold		Th2			2			ns
Output delay	D [3:0], CTL [1:0], LREQ	Td2			2		15	ns
Input setup	ADDRESS [6:0], DATA [15:0], XCS, XWR, XRD	Tsu3	Refer to 6-1. ATF/CFR write timing and ATF/CFR read timing					
Input hold		Th3						
Output delay	DATA [15:0], XWAIT	Td3						

4-3. Input/Output Capacitance

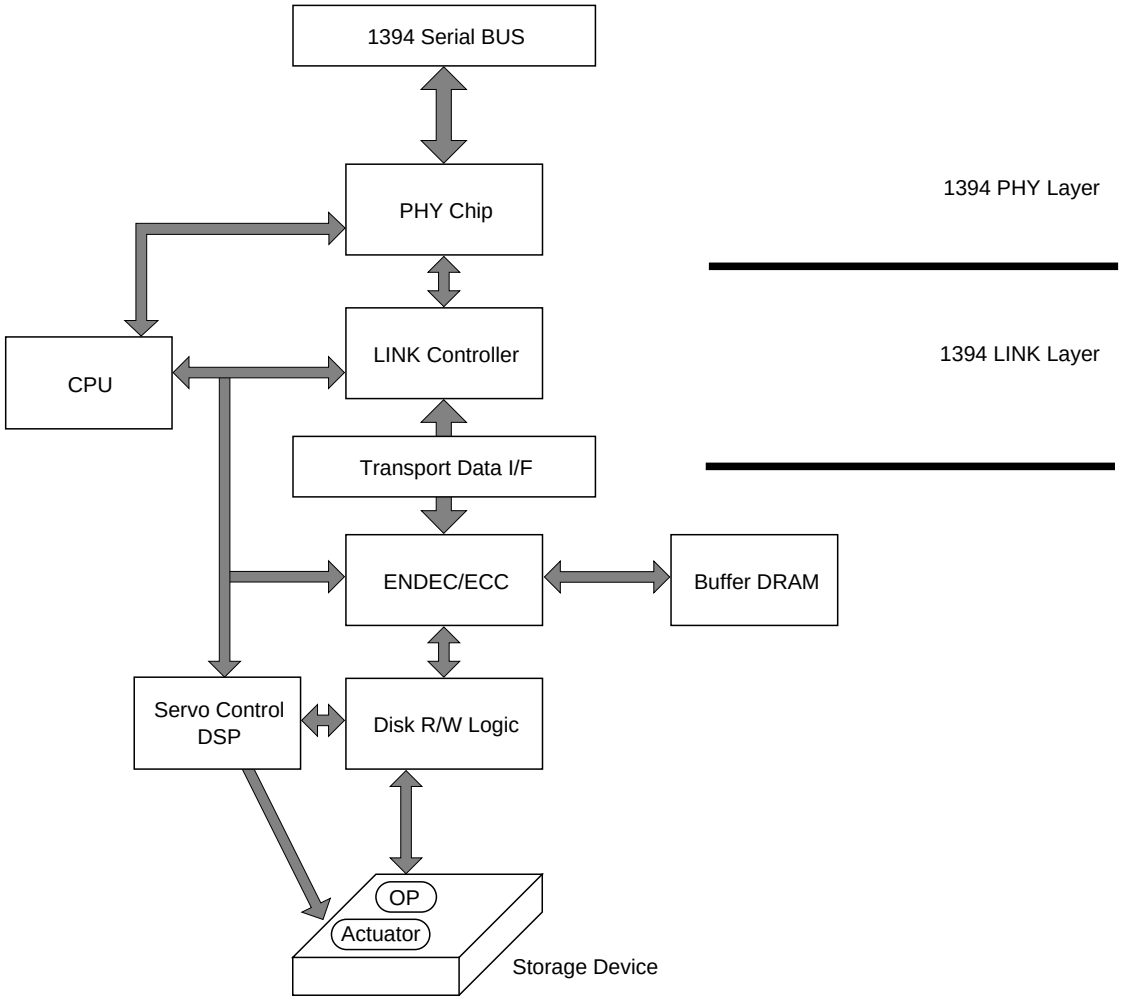
Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Input capacitance	C_{IN}	All input pins			9	pF
Output capacitance	C_{OUT}	All input pins			11	pF
I/O capacitance	$C_{I/O}$	D [3:0], CTL [1:0], SD [15:0]			11	pF

4-4. Timing Definition



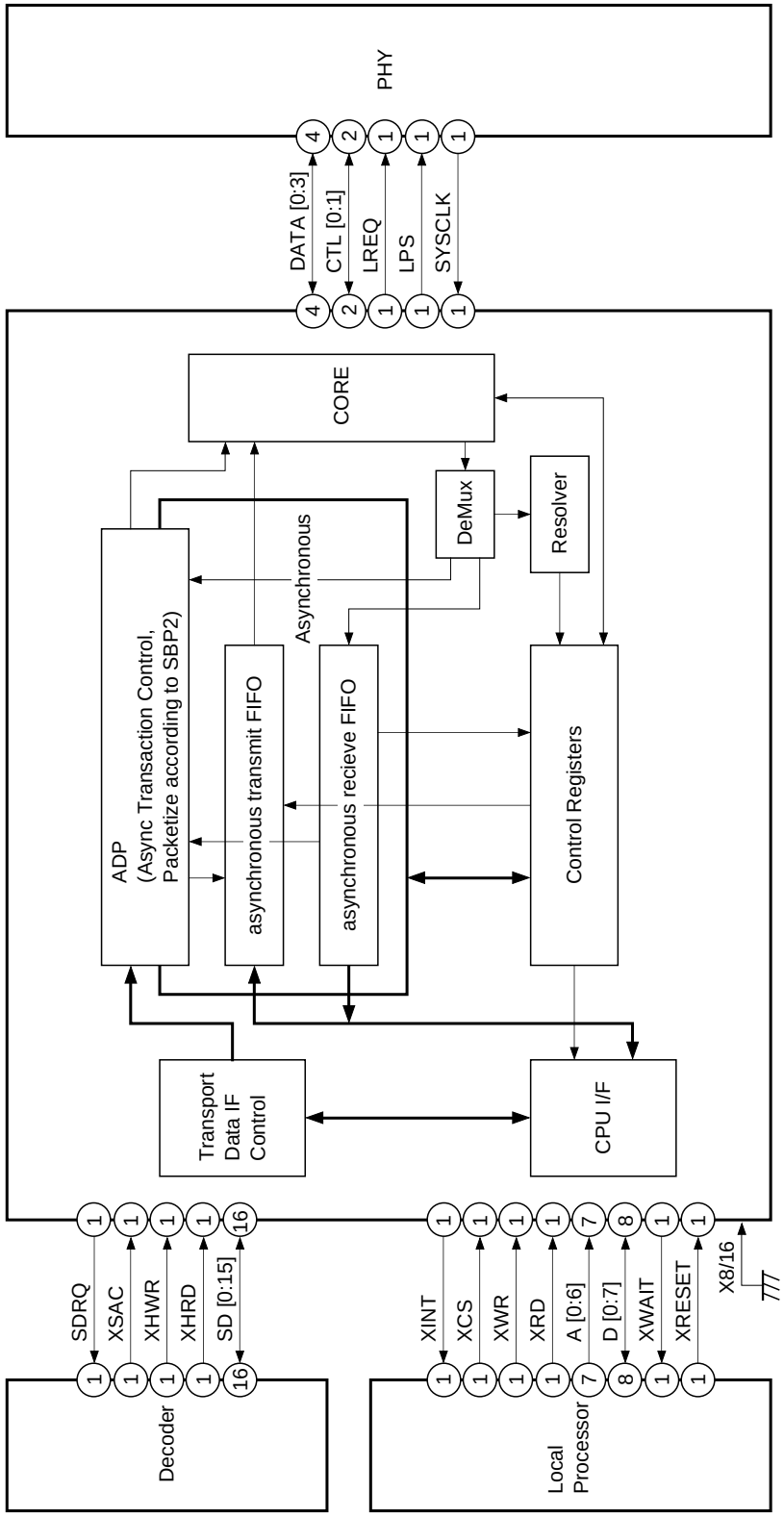
5. System Configuration Example

5-1. System Block Diagram



5-2. System Connection Diagram

- CPU Interface....8bit
- Transport Data Interface....16bit



6. Asynchronous Communication

6-1. CPU I/F

The CPU I/F controls data communication between the external CPU and the CXD3220R ATF/ARF/CFR*1, respectively.
Communications between the CPU and CXD3220R include:

1) CPU writes data to ATF → Asynchronous packet transmit
2) CPU reads data in ARF → Asynchronous packet receive
3) CPU writes data to CFR → mode, header data setting
4) CPU reads data in CFR → internal status, header data read
5) CXD3220R informs CPU of an interrupt event with an interrupt signal

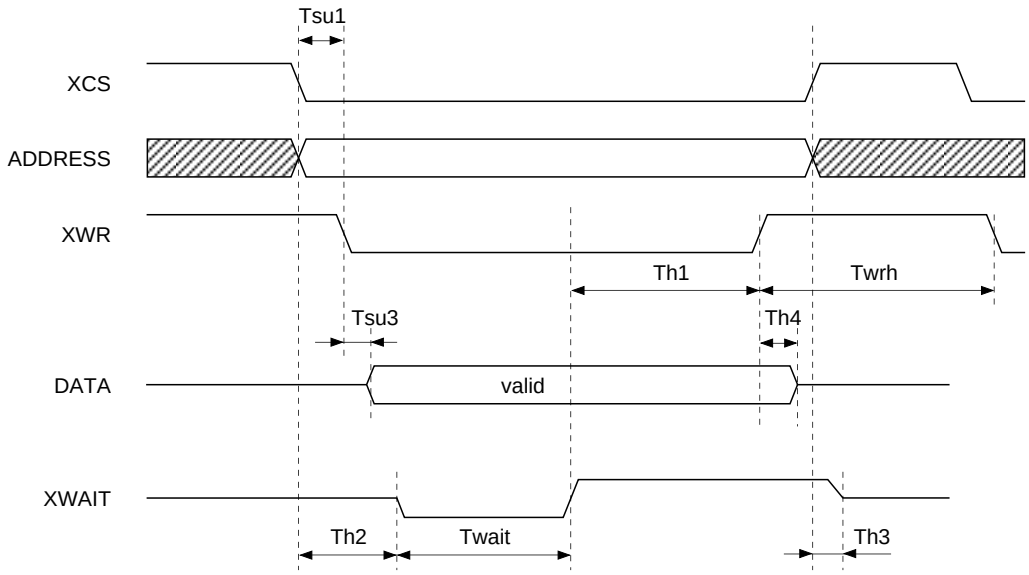
The CXD3220R supports 16-bit and 8-bit CPU I/F.
The ATF/ARF/CFR built in the CXD3220R have a 32-bit structure, so all bits can not be accessed with one access. The target address must be accessed two consecutive times for 16 bits and four consecutive times for 8 bits.

The roles played by the signals communicated between the CXD3220R and the external CPU are given bellow.

Data [15:0]	in/out	Data for writing to or reading from specified address
ADDRESS [6:0]	in	Address for writing or reading data Data destination (CFR or FIFO) and data breakpoint (Write or Confirm) are discriminated according to the address
XCS	in	Access enable from host bus (low active)
XWR	in	Data write enable signal (low: write)
XRD	in	Data read enable signal (low: read)
XWAIT	out	Indicates access (read or write) completed to specified address (low active)
XINT	out	Interrupt signal. Indicates some kind of interrupt when low Type of interrupt and mask specified by CFR
X8/16	in	CPU I/F data bus switching High: 16 bits; low: 8 bits

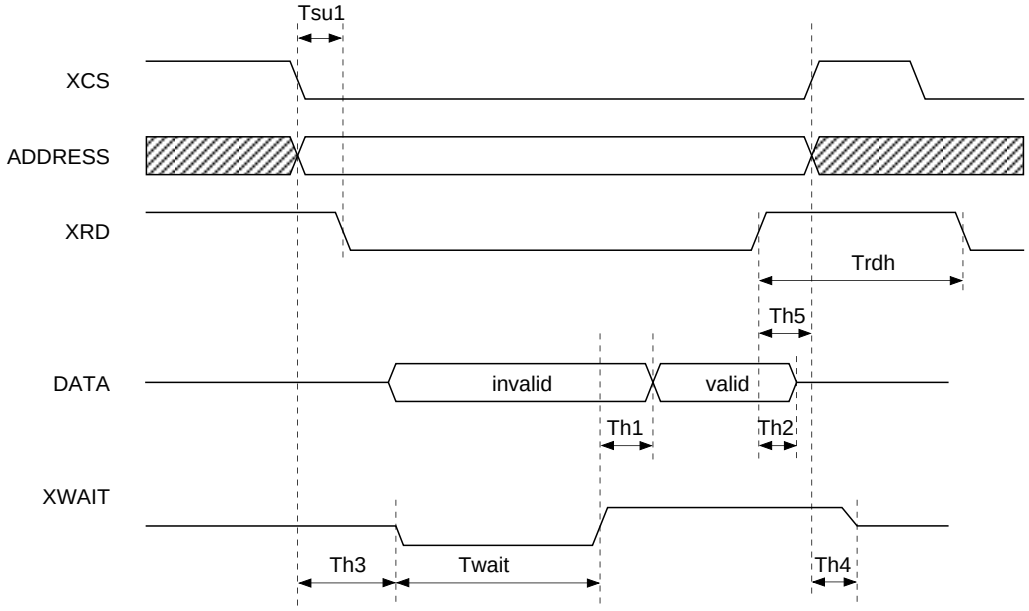
*1 ATF (Asynchronous Transmit FIFO), ARF (Asynchronous Receive FIFO), CFR (Configuration Register)
In the CXD3220R, the ATF has the capacity of 24 quadlets and the ARF has the capacity of 39 quadlets.

Writing Timing to ATF/CFR



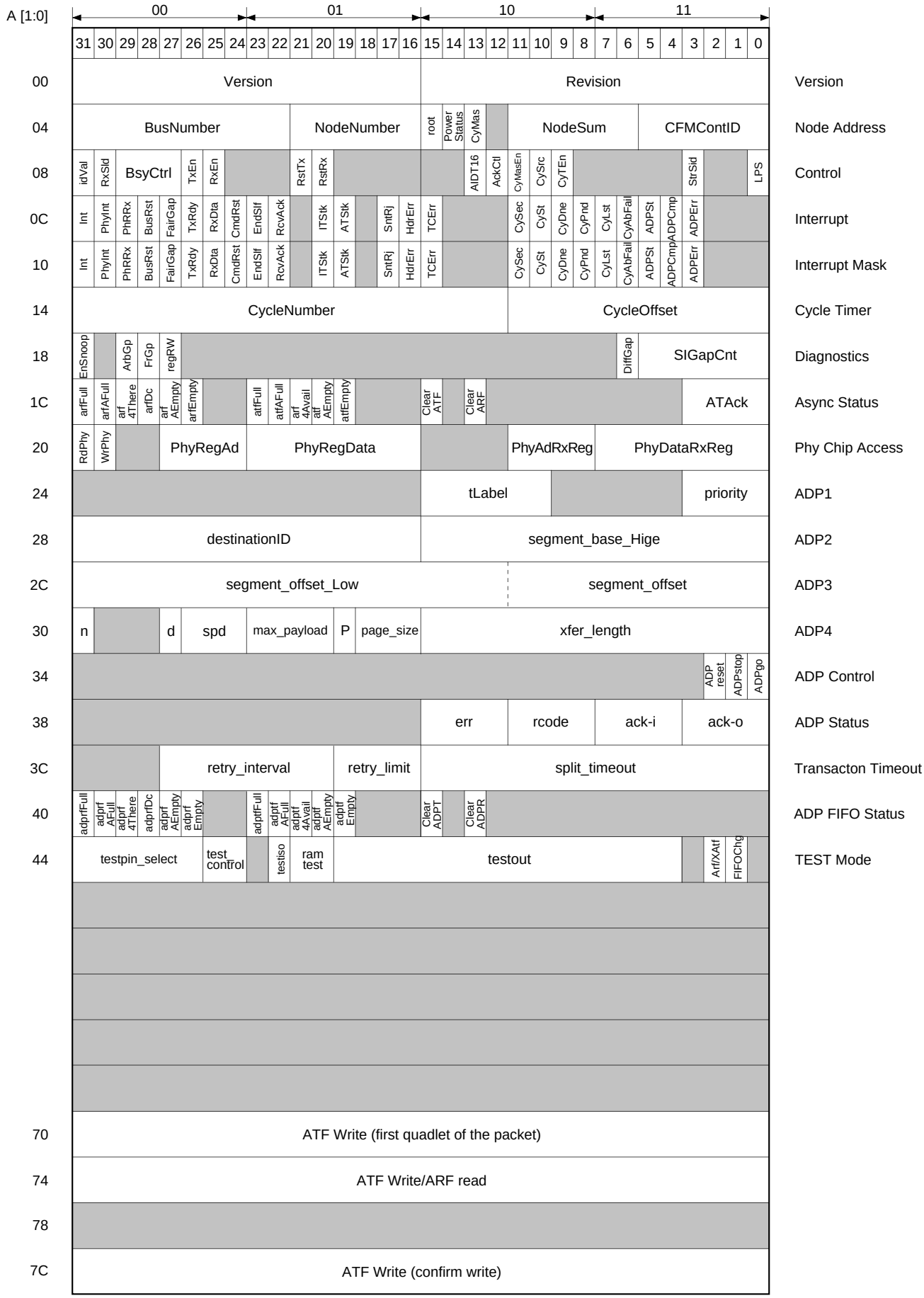
Tsu1 5nsec min, Tsu3 13nsec max, Th1 5nsec min, Twrh 60nsec min
Twait 100nsec max, Th2 8nsec max, Th3 14nsec max, Th4 5nsec min

Read Timing from ARF/CFR



Tsu1 5nsec min, Th1 3nsec max, Trdh 60nsec min
Twait 270nsec max, Th2 16nsec max, Th3 8nsec max, Th4 14nsec max, Th5 5nsec min

Configuration Register (CFR) Address Map



* The shaded areas ( ) are reserved and can not be used.

6-2. CFR (Configuration Register)

This is a memory space to store the status information, operation mode and packet header information in the chip. Read/write with the external CPU can be performed via the CPU I/F.
The address map and register contents are shown below.

Register Description

1) Version/Revision Register
These registers have the CXD3220R version/revision written in them.
The register address is 00h; they are read only, and the default value is 3220_0000h.

Bit	Name	Function
31 to 16	Version	CXD3220R version number
15 to 0	Revision	CXD3220R revision number

2) Node Address Register
These registers are used to monitor root/cycle master status and the total number of nodes connected, and so on.
The register address is 04h and the initial value is FFFF_0000h.
Only the bus number is for read/write, and the other registers are normally for read only, but the Diagnostic register can be read/write by setting regRW to "1".

Bit	Name	Function
31 to 22	Bus Number	Bus number of connected bus
21 to 16	Node Number	Node number of this link
15	root	Root/not root for this link 1: root; 0: not root
14	Power Status	Cable power status for this mode 1: CPS on; 0: CPS off
13	CyMas	Whether or not this link is cycle master 1: cycle master; 0: not
11 to 6	NodeSum	Total number of connected nodes. The value becomes "0" when an error occurs in the Self ID phase. This value is fixed when the Interrupt register EndSlf bit becomes "1" from "0".
5 to 0	CFMcontID	The Phy-ID value of the contender is loaded. However, when the CXD3220R node has an ability to become the contender and this LSI has the Phy-ID value larger than the loaded value, the CXD3220R itself is the contender. This value is fixed when the Interrupt register EndSlf bit becomes "1" from "0".

3) Control Register

These registers perform settings for the CXD3220R basic operations.

The register address is 08h; they are for read/write, and the initial value is C600_2A01h.

Bit	Name	Function
31	idVal	Receives packet from the address set in the Node Address register and packet at bus number "3FFh" node number "3Fh" when "1". Receives packet at bus number "3FFh" node number "3Fh" only when "0".
30	RxSld	Validates reception of Self ID packet when "1". Non-valid when "0". (Fixed at "1" in the CXD3220R)
29 to 27	BsyCtrl	Controls Busy status of input packet 000 = Returns Busy according to normal Busy/retry protocol when necessary. (Fixed at "000" in the CXD3220R)
26	TxEn	Transmitter does not transmit Arbitration and packet when "0".
25	RxEn	Receiver does not receive packet when "0".
21	RstTx	Sync resets transmitter when "1". This bit is cleared automatically. (Do not use for normal operation.)
20	RstRx	Sync resets receiver when "1". This bit is cleared automatically. (Do not use for normal operation.)
13	AIDT16	Selects SD bus width. 8 bits when "0" and 16 bits when "1".
12	AckCtl	Controls the Ack code that is sent back when a packet is received for which Tcode = 0, 1 (write request quadlet/block). 0: Ack code = 1 (complete), 1: Ack code = 2 (pending)
11	CyMasEn	The Cycle Master function operates if the CXD3220R becomes Root when "1".
10	CySrc	Incrementation of the cycle number and reset of Cycle Offset are performed with Cycle In when "1". Incrementation is performed with Cycle Offset when "0". (This is always set to "0" internally for this link.)
9	CyTEn	Validates Cycle Offset increment when "1". (This is always set to "1" internally for this link.)
3	StrSid	Takes received Self ID packet in at the ARF when "1". Does not take received Self ID packet in to the ARF when "0".
0	LPS	The LPS pin is high when "1". The LPS pin is low when "0".

4) Interrupt and Interrupt-Mask Registers

These registers combine the Interrupt register, which informs the CPU I/F of changes in the CXD3220R status, and the Interrupt-Mask register, which masks the Interrupt register.

The address of the Interrupt register is 0Ch, and when the regRW bit is "0", bits other than Int bit and ADPErr bit are cleared by writing "1". When the regRW bit is "1" all bits are for read/write.

The address of the Interrupt-Mask register is 10h and it is for read/write. When "1" is written to the corresponding bit, the interrupt becomes valid; when "0" is written, it becomes invalid.

The initial value for both registers is 0000_0000h. The Interrupt OR corresponding to the bit where "1" is written in the Interrupt-Mask register becomes the INT bit, resulting in the XINT output signal.

And the XINT output signal becomes valid when "1" is written to the Interrupt-Mask register INT bit; when "0" is written, invalid.

Bit	Name	Function
31	Int	All interrupt OR results and their interrupt mask bits.
30	PhyInt	Phy Interrupt was received from Phy chip.
29	PhyRegRx	Data was received from Phy to Phy register.
28	BusRst	Bus Reset was received from Phy.
27	FairGap	Fair Gap received from Phy.
26	TxRdy	Transmitter is able to transmit. "0" when a packet is transmitted; "1" when an Ack code is fixed.
25	RxDta	Receiver has received a correct packet. A packet is not loaded in the ARF when the Self-ID packet is received if the Control register Strsid is set to "0" and when the Response packet is received at the ADP circuit for ADP operation. However, RxDta Interrupt is set.
24	CmdRst	Receiver has received a packet addressed to CSR RESET_START register.
23	EndSlf	Indicates that Self ID phase has completed.
22	RcvAck	Ack code was received.
20	ITStk	Transmitter detected wrong data in Isochronous FIFO during Isochronous transmit. (Always set to "0" in this IC)
19	ATStk	Transmitter detected wrong data in Asynchronous FIFO during Asynchronous transmit.
17	SntRj	Receiver transmitted Busy Ack for a packet transmitted to this node because received FIFO is full.
16	HdrErr	Receiver detected Header CRC error in the packet transmitted to this note.
15	TCErr	Transmitter detected wrong tCode data in transmitted FIFO.
11	CySec	Cycle Timer register Cycle Number upper 7 bits were incremented. (This is generated almost every second when Cycle Timer is valid.)
10	CycSt	Transmitter/Receiver transmitted/received Cycle Start packet.
9	CycDne	After transmit or receive of Cycle Start packet, Fair Gap was detected on the bus. This means that the Isochronous cycle is complete.
8	CycPnd	Cycle Timer register Cycle Offset is "0". Stays as is until Isochronous cycle is complete.
7	CycLst	When not Cycle Master, Cycle Timer completed two cycles without receiving Cycle Start packet.
6	CyAbFail	Failure of Cycle Start packet transmission Arbitration.

Bit	Name	Function
5	ADPSt	The ADP has started.
4	ADPCmp	The ADP has completed.
3	ADPErr	An error has occurred during ADP processing. In order to clear ADPErr bit, write "1" to this bit after "1" is written to ADP Control register ADPreset bit.

5) Cycle Timer Registers

These registers are composed of the 24.576MHz clock cycle Cycle Offset and the 125µs in its host, and the Cycle Number that counts one second. The value of all nodes are regulated by the Cycle Master node.
The register address is 14h; it is for read/write, and the initial value is 0000_0000h.

Bit	Name	Function
31 to 12	CycleNumber	The upper 7 bits count seconds (1Hz) and the lower 13 bits count the Isochronous cycle (8kHz = 125µs). The values are controlled by Control register Cycle Master and Cycle Timer Enable.
11 to 0	CycleOffset	Counts the system clock (24.576MHz). The Cycle Number is incremented when this counter completes one cycle. The value is controlled by Control register Cycle Master and Cycle Timer Enable.

6) Diagnostic Register

This register controls or monitors the CXD3220R status.
The register address is 18h and the initial value is 0000_0000h.
Only the EnSp bit and regRW bit are for read/write; other bits are for read/write when the regRW bit is "1" and for read only when it is "0".

Bit	Name	Function
31	EnSnoop	Receives all packets on the bus regardless of receiver address and format when "1". Invalid when "0".
30	BsyF	Ack to be sent back next is "Ack_BusyB" when "1". Ack to be sent back next is "Ack_BusyA" when "0".
29	ArbGp	Bus is in idle state due to Arbitration Reset Gap.
28	FrGp	Bus is in idle state due to Fair Gap.
27	regRW	Almost all registers are for read/write when "1".
6	DiffGap	"1" when there is dispersion in Gap count values in received Self ID. This value is fixed when the Interrupt register EndSlf bit becomes "1" from "0".
5 to 0	SIGapCnt	The value is entered when all Gap count values in received Self ID are the same. "00h" when bus reset is generated.

7) Asynchronous Transmit and Received FIFO Status Registers

These registers can monitor and control the ATF/ARF statuses.

The register address is 1ch and the initial value is 0428_0000h.

Only the Clear ATF bit and Clear ARF bit are for read/write; other bits are for read/write when the regRW bit is "1" and read only when it is "0".

Bit	Name	Function
31	ARFFull	The ARF is full when "1" and receive is not possible.
30	ARFAFull	The ARF can receive only one more quadlet when "1".
29	ARF4Th	The ATF can write more than four quadlets of data when "1".
28	ARFDc	This is the control bit for reading a packet from ARF, and is "1" only for the first and last quadlets of the packet.
27	ARFAEmpty	Only one more quadlet of data is written in the ARF when "1".
26	ARFEmpty	The ARF is empty when "1" and there is no data to be read.
23	ATFFull	The ATF is full when "1" and write is not possible.
22	ATFAFull	Only one more quadlet can be written in the ATF when "1".
21	ATF4Avail	More than four quadlets of data can be written in the ATF when "1".
20	ATFAEmpty	The ATF has only one more quadlet of data not transmitted when "1".
19	ATFEmpty	The ATF is empty when "1" and there is no data for transmit.
15	ClearATF	Sync reset of ATF when "1" (Self Clear).
13	ClearARF	Sync reset of ARF when "1" (Self Clear).
3 to 0	ATAck	Value of received Ack code. This is fixed when the TxRdy bit becomes "1" from "0" and the fixed value is maintained till the next Act code is received.

8) Phy Chip Access Registers

These registers are used for read/write of the contents of the Phy chip Phy register connected to the CXD3220R.

The register address is 20h and the initial value is 0000_0000h.

Bit	Name	Function
31	RdPhy	The CXD3220R requests read to the address set in PhyRgAd via the Phy I/F when "1".
30	WrPhy	The CXD3220R requests write to the address set in PhyRgAd via the Phy I/F when "1".
27 to 24	PhyRegAd	Sets the read/write address of the connected Phy chip Phy register.
23 to 16	PhyRegData	Value of data for write to address specified by PhyRegAd.
11 to 8	PhyAdRxReg	Value of the read Phy register address during read.
7 to 0	PhyDataRxReg	Value of the read Phy register data during read.

9) ADP1 Registers

These registers are used to set the ADP.

The register address is 24h and the initial value is 0000_0000h.

Bit	Name	Function
15 to 10	tLabel	Indicates the Transaction Label and is used in a pair with the response packet to that request packet. (Do not use the tLabel set with ADP for packets transmitted from the ATF.)
3 to 0	priority	Indicates the priority level of the packet. In the case of a value other than "0", the transmitter uses priority Arbitration for this packet.

10) ADP2 Registers

These registers are used for setting of the ADP.

The register address is 28h and the initial value is 0000_0000h.

Bit	Name	Function
31 to 16	destinationID	The bus number of the destination of the packet is represented with 10 bits, while the node number is represented with 6 bits.
15 to 0	segment_base_High	For a continuous area (segment_base_High, segment_base_Low and, depending on the case, segment_offset), this indicates the address of the address space of the destination node.

11) ADP3 Registers

These registers are used for setting of the ADP.

The register address is 2Ch and the initial value is 0000_0000h.

All 32 bits are at segment_base_Low when in Mode0 or 1.

In Mode2, the lower bit (page_size + 8) is at segment_offset, while the upper bit is at segment_base_Low.

Mode0 and Mode1

Bit	Name	Function
31 to 0	segment_base_Low	For a continuous area (segment_base_High, segment_base_Low), this indicates the address of the address space of the destination node. This address must be in word units when the Control register AIDT16 = "1" in Mode1.

Mode2

Bit	Name	Function
31 to b	segment_base_Low	For 3 continuous areas (segment_base_High, segment_base_Low, segment_offset), this indicates the address of the address space of the destination node. This address must be in word units when the Control register AIDT16 = "1".
(b – 1) to 0	segment_offset	In the case of Mode2 that supports transfer by page_table, this indicates the lower bit of the first address of the element. It also sets the segment_offset value of ORB.

b = (page_size + 8)

12) ADP4 Registers

These registers are used for setting of the ADP.

The register address is 30h and the initial value is 0000_0000h.

Bit	Name	Function
31	notify (n)	This is the notify bit of the ORB format. This has not effect on this IC. Please use it as memory.
27	direction (d)	This is used to determine the direction of ADP transfer. 0: Reception of data from the initiator to this link. 1: Transmission of data from this link to the initiator. This is used to set the direction value of the ORB. (Since ADP transmission and reception is switched with this bit, only perform writing to this register after the series of Transactions with the ADP have been completed and the ADP is not operating.)
26 to 24	spd	This is the transfer rate of the 1394 serial bus. 0: S100 1: S200 2 to 7: Reserved (Do not set to these values.) This bit is used to set the spd value of the ORB.
23 to 20	max_payload	This indicates the maximum data_length with 2^ (max_payload + 2). It is used to set the max_payload value of the ORB. A value of 8 is set when a value larger than 8 is set with the CXD3220R.
19	p	This is the page_table_present bit. This is set to "1" when using the page_table, and set to "0" when not using. The device enters Mode2 when "1". This is used to set the p value of the ORB.
18 to 16	page_size	This bit represents the data_length of one page. page_length is represented with 2^ (page_size + 8). It is used to set the page_size value of the ORB.
15 to 0	xfer_length	This represents the data buffer length in Mode0 or 1, and is used to set the data_size value of the ORB. In Mode2, it represents the segment length, and is used to set the Segment_Length value of the Page Table. (Do not start the ADP when xfer_length = 0.)

13) ADP Control Registers

These registers are used for controlling the ADP.

The register address is 34h and the initial value is 0000_0000h.

Bit	Name	Function
2	ADPreset	Returns the ADP to the initial state. Clears the ADP status register.
1	ADPstop	This bit is set to "1" when stopping the ADP. The ADP is then stopped after it has normally completed the Transaction of the packet currently loaded in the ADPTF or an error has occurred, after which this bit is cleared.
0	ADPgo	This bit is set to "1" when starting the ADP. The ADP is then stopped after it has normally completed or an error has occurred, after which this bit is cleared.

14) ADP Status Registers

These registers are used for reading the ADP Status value.

The register address is 38h and the initial value is 0000_0000h.

This register is for read only.

Bit	Name	Function
15 to 12	err	This indicates the error code (see below) of the ADP. The Interrupt ADPErr bit rises when an error has occurred.
11 to 8	rcode	This indicates the response code of a response packet that has returned to the ADP from the initiator. "1111" is written in the case of a Write Transaction that has become a unified Transaction.
7 to 4	ack-i	Writes the Ack code for the request packet.
3 to 0	ack-o	Writes the Ack code transmitted by the ADP for the response packet.

These registers are cleared when a ADPgo bit has been set. After a ADPgo bit has been set and the ADP has started, the following occurs in the case any type of error occurs in the request packet or response packet.

1) Generation of a request packet is stopped.

2) The rcode and Ack are latched, and stored in the ADP Status register.

3) An interrupt is generated.

This register is cleared when ADPreset = 1.

<List of Error Codes>

err value	meaning
0 (all clear)	no error
1	error ack code received (for request packet)
2	error ack code sent (for response packet)
3	split transaction time-out
4	busy_timeout
5	bus reset occurred
6	bad rcode received
7	receive response packet from a node other than the specified node
8	bad tCode received (bad tCode is 2 [d = 0], 6/7 [d = 1])

(When two or more error codes occur at the same time, the code with the lower code value is displayed.)

15) Transaction Timeout Registers

In the case the Ack code of ack_busy has returned after the ADP has sent a request packet, there is a function for retransmitting the subject request packet.

These registers are used to set the limit value of the timeout required until a response packet is sent back after a request packet has been transmitted during Split Transaction, as well as the upper limit of the number of times retry is performed when an Ack code has returned as ack_busy. The register address is 3Ch, and the Initial value is 800 (dec).

Bit	Name	Function
27 to 20	retry_interval	This designates the retry interval. The packet is retransmitted after waiting for 125µs × (retry_interval). When set to "0", transmission is performed immediately without waiting for the interval.
19 to 16	retry_limit	The retry_limit bit controls retry when a single-phase retry protocol is in use. When this bit is set to "0", packet transfer that was busy is not retried. When set to a value other than "0", packet transfer is retried for the maximum number of retries (retry_limit) until any Ack code returns other than a busy acknowledgement. When a packet is unable to be transferred as a result of being busy after the maximum number of retries, the ADP stops the processing of that packet. A busy_timeout error is indicated in the err field of the ADP Status register.
15 to 0	split_timeout	When Split Transactions are being performed, the ADP stops processing when the amount of time for a response packet sent in response to a request packet to return exceeds (split_timeout × 125) µs. A split transaction_timeout error is displayed in the err field of the ADP Status register.

16) ADP FIFO Status Registers

These registers make it possible to monitor and control ADP status.

The register address is 40h and the initial value is 0428_0000h.

Reading and writing are only possible for the Clear ADPTF bit and Clear ADPRF bit. Reading and writing of other bits is possible only when the regRW bit is set to "1". These bits are for read only when it is set to "0".

Bit	Name	Function
31	ADPRFFull	Indicates that the ADPRF is full and reception is not possible when "1".
30	ADPRFAFull	Indicates that the ADPRF is able to receive only one more quadlet when "1".
29	ADPRF4Th	Indicates that four or more quadlets of data have been written into the ADPRF when "1".
28	ADPRFDc	This is a control bit for reading packets from the ADPRF. This bit is "1" only during the first and last quadlets of a packet.
27	ADPRFAEmpty	Indicates that only one quadlet of data has been written into the ADPRF when "1".
26	ADPRFEmpty	Indicates that the ADPRF is empty and there is no data that can be read when "1".
23	ADPTFFull	Indicates that the ADPTF is full and that writing is not possible when "1".
22	ADPTFAFull	Indicates that only one more quadlet can be written into the ADPTF when "1".
21	ADPTF4Avail	Indicates that only four more quadlets can be written into the ADPTF when "1".
20	ADPTFAEmpty	Indicates that there is only one quadlet of data that has not been transmitted in the ADPTF when "1".
19	ADPTFEmpty	Indicates that the ADPTF is empty and there is no data that can be transmitted when "1".
15	ClearADPTF	Sync resets the ADPTF when "1" (Self Clear).
13	ClearADPRF	Sync resets the ADPRF when "1" (Self Clear).

17) TEST Mode Registers

These registers are used to control the test mode of the CXD3220R.
They are normally set to 0000_0000h.
The register address is 44h and the initial value is 0000_0000h. Do not write in this register.

18) ATFWrite (first quadlet of the packet) Registers

The first quadlet of the transmitted Asynchronous packet is written in these registers.
The register address is 70h and the initial value is 0000_0000h.

Bit	Name	Function
31 to 0	ATFWrite (first quadlet of the packet)	Writes the first quadlet of the transmitted Asynchronous packet.

19) ATFWrite/ARFRead Registers

The second through the next to the last quadlets of the transmitted Asynchronous packet are written in these registers. Also, the Asynchronous packet read from the ARF during receive is written one quadlet at a time.
The register address is 74h and the initial value is 0000_0000h.

Bit	Name	Function
31 to 0	ATFWrite /ARFRead	Transmit: Writes the second through the next to the last quadlets of the transmitted Asynchronous packet. Receive: Reads one quadlet at a time for the Asynchronous packet read from ARF.

20) ATFWrite (confirm write) Registers

The last quadlet of the transmitted Asynchronous packet is written in these registers.
The register address is 7Ch and the initial value is 0000_0000h.

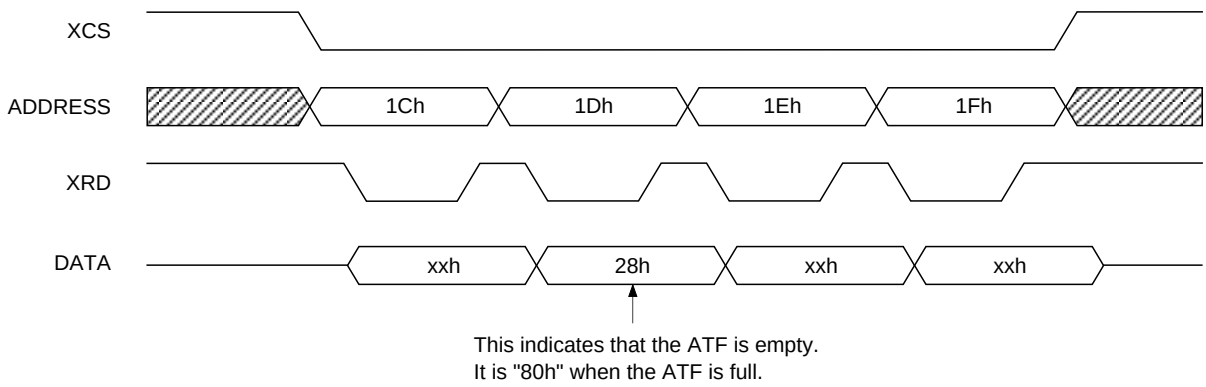
Bit	Name	Function
31 to 0	ATFWrite (confirm write)	Writes the last quadlet of the transmitted Asynchronous packet.

6-3. Asynchronous Packet Transmission

Packet data is written from the external CPU to the ATF inside the CXD3220R in order to transmit an Asynchronous packet. At this time the first quadlet of the packet only is written in the CFR ATFWrite (first quadlet of the packet) register (70h). The second through the next to the last quadlets are written in the CFR ATFWrite/ARFRead registers (74h). Then the last quadlet is written in the CFR ATFWrite (confirm write) register (7Ch) and the packet is stored in the ATF.
However, if the ATF is full, write will not actually be performed even when write is executed.
Once the bus is enabled, transmit takes place automatically.

The procedure for transmitting a Quadlet Write request packet is given here as an example.
(for 8-bit data interface)

(1) Confirming that the ATF is not full
The CFR Async Status register (1Ch to 1Fh) is read to confirm that the 23th bit (AtfFull bit) is low. If it is high it means that there are some unsent packets stored and it waits until they are transmitted.

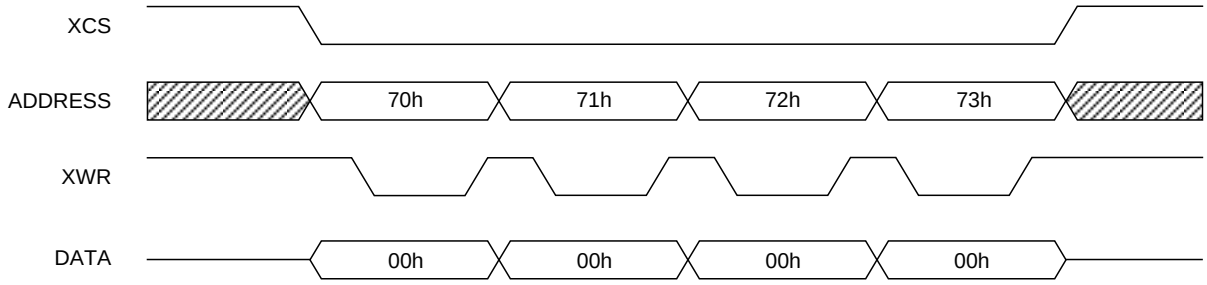


The number of quadlets that can be stored in the ATF can be found from the value of the Async Status register bits [23:19]. The following six states can be found, so a judgment must be made as to whether write is possible from the number of quadlets in the packet being sent from the external CPU.

AtfFull = High: Can't Write
AtfAFull = High: Only one quadlet
All bits low: 2 to 3 quadlets
Atf4Avail = High: 4 to 22 quadlets
Atf4Avail = High, AtfAEmpty = High: 4 to 23 quadlets
Atf4Avail = High, AtfEmpty = High: 4 to 24 quadlets

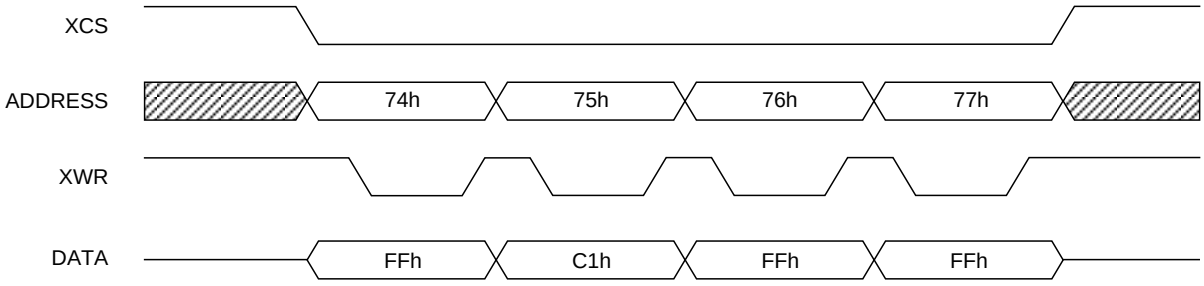
(2) First quadlet of the transmitted packet Write

Let the first quadlet of the Quadlet Write request packet be "00000000h".
This is written in the CFR ATFWrite (first quadlet of the packet) register.



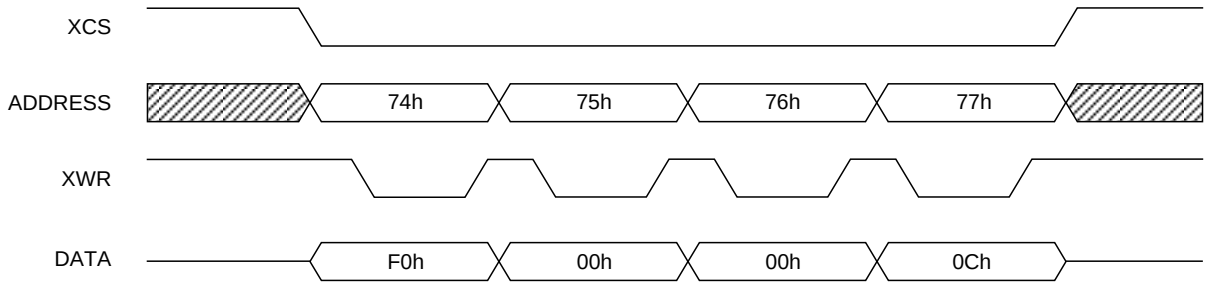
(3) Second quadlet of the transmitted packet Write

Let the second quadlet of the Quadlet Write request packet be "FFC1FFFFh".
This is written in the CFR ATFWrite/ARFRead register.



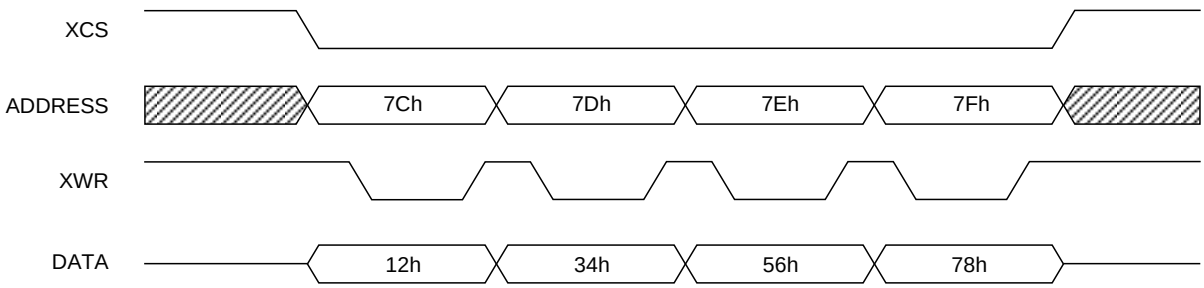
(4) Third quadlet of the transmitted packet Write

Let the third quadlet of the Quadlet Write request packet be "F000000Ch".
This is written in the CFR ATFWrite/ARFRead register.



(5) Last quadlet of the transmitted packet Write

Let the last quadlet of the Quadlet Write request packet be "12345678h".
This is written in the CFR ATFWrite (confirm write) register.



The Quadlet Write request packet is stored in the ATF as shown above. When the bus is enabled, the CXD3220R transmits automatically. If transmit does not take place, the CFR interrupt register (0Ch to 0Fh) must be read to confirm if the ATStk bit or TCErr bit is high. If these bits are high, the packet stored in the ATF may not be correct.

ATStk = High: If the first quadlet of the packet was not written in the CFR ATFWrite (first quadlet of the packet) register but was written in the ATFWrite/ARFRead register or the ATFWrite (confirm write) register.

TCErr = High: A value that is not a Transaction code able to be transmitted by Asynchronous packet is written in the tCode field of the first quadlet of the packet.
The Transaction codes that can be transmitted as Asynchronous packets are any of (0, 1, 2, 4, 5, 6, 7, 9, B, Eh).

For either of ATStk or TCErr above, the next packet for write will not be transmitted even if it is correct.
At this time "1" must be written in the CFR Async Status register ClearATF bit in order to clear the ATF. Transmit is then enabled when a correct packet is written.

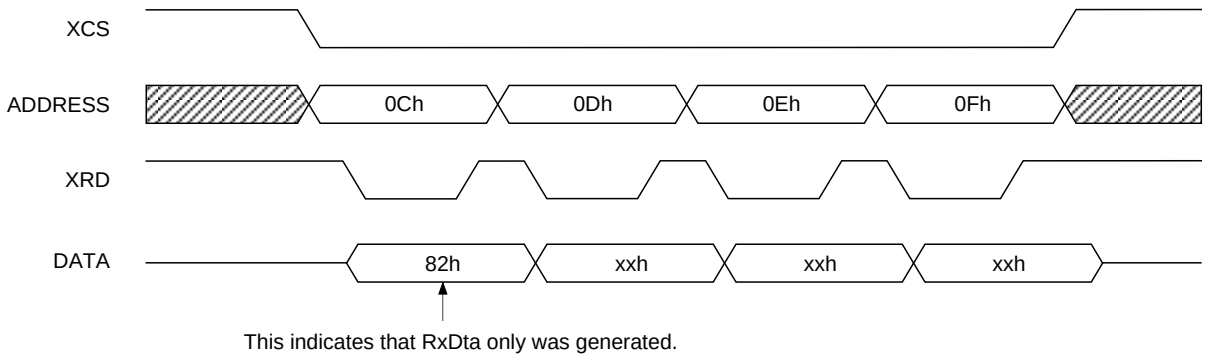
6-4. Asynchronous Packet Reception

Basically, if there is room to write the packet in FIFO and the destination_ID matches, then Asynchronous packets are received. Receive is completed when the packet data is read from the ARF inside the CXD3220R by the external CPU.

The CXD3220R raises an RxDta flag when a packet is received. (Normally, if the RxDta bit of the CFR Interrupt Mask register (10h to 13h) is set to "1", XINT goes low when a packet is received and this can be detected.)
Next, the CFR Async Status register (1C to 1Fh) ArfEmpty bit should be low. This indicates that a correct packet was received.
After this, one quadlet at a time can be read by reading the CFR ATFWrite/ARFRead registers (74h to 77h).
Packet receive is completed by repeating this until the ArfEmpty bit goes high.
However, if the ARF status is empty, read will not be done even if it is executed. In this case, the data read by the CPU will be the previously read value.

The procedure for receiving a Quadlet Write request packet is given here as an example.
(for 8-bit data interface)

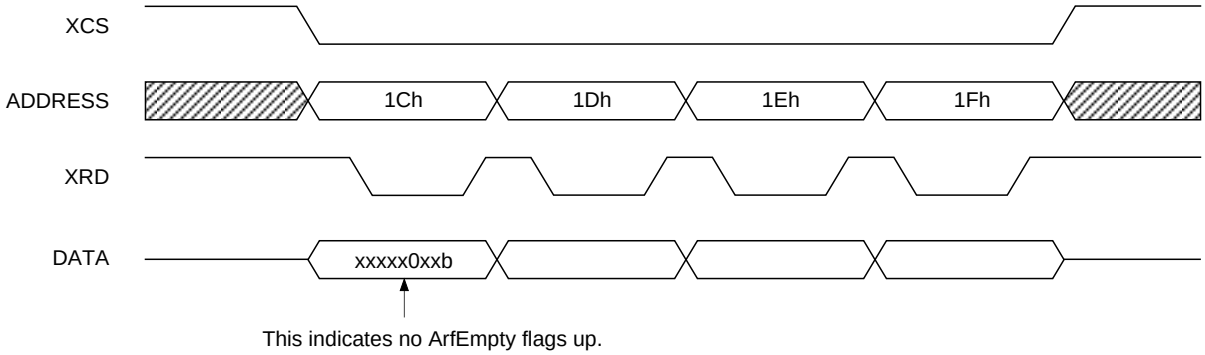
(1) Confirming that the packet was received
The CFR Interrupt register (0Ch) is read to confirm that the 25th bit (RxDta bit) is high.



When only desiring to know information about the register of the lower 2 bits A [1:0] = 00 of the address, only the address of A [1:0] = 00 may be read. In the case of reading register information for A [1:0] = 01, 10, 11 read the addresses in order starting from the address of A [1:0] = 00.

(2) Confirming that the received packet was stored correctly in FIFO

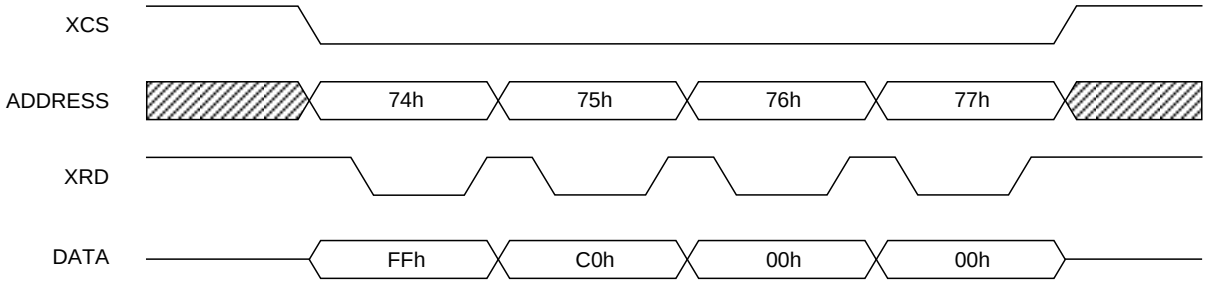
The CFR Async Status register (1C to 1Fh) is read to confirm that the 26th bit (ArfEmpty bit) is low.
If this bit is high it means that reception may be in progress (all quadlets have not arrived). In this state, do not read the ARF read register (74 to 77h). Wait some time and again read the Async Status register to confirm the ArfEmpty bit.



In the above example, ArfEmpty is low.
Data read is possible because the ArfEmpty bit is low.

(3) First quadlet of the received packet Read

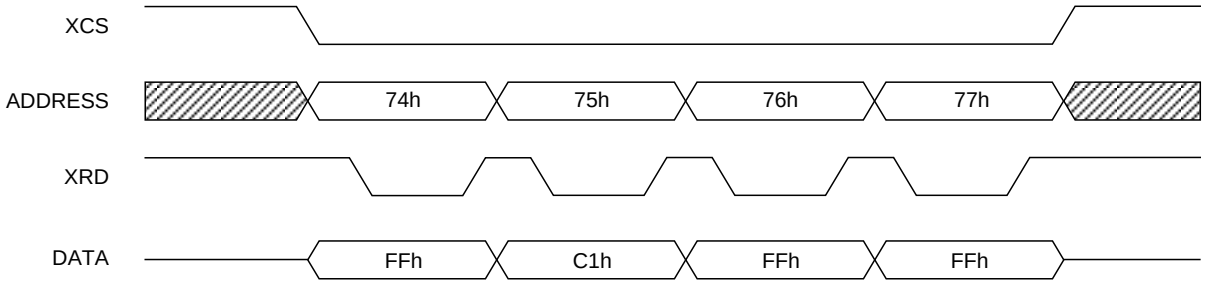
The CFR ATFWrite/ARFRead register is read.



The data read is "FFC00000h".
At this time, the ArfDc bit is high (from (2) above), so this quadlet is the first quadlet.

(4) Second quadlet of the received packet Read

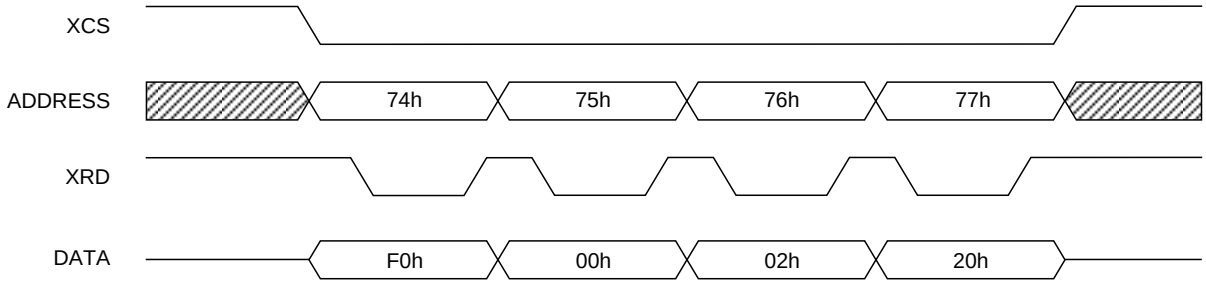
The CFR ATFWrite/ARFRead register is read.



The data read is "FFC1FFFFh".

(5) Third quadlet of the received packet Read

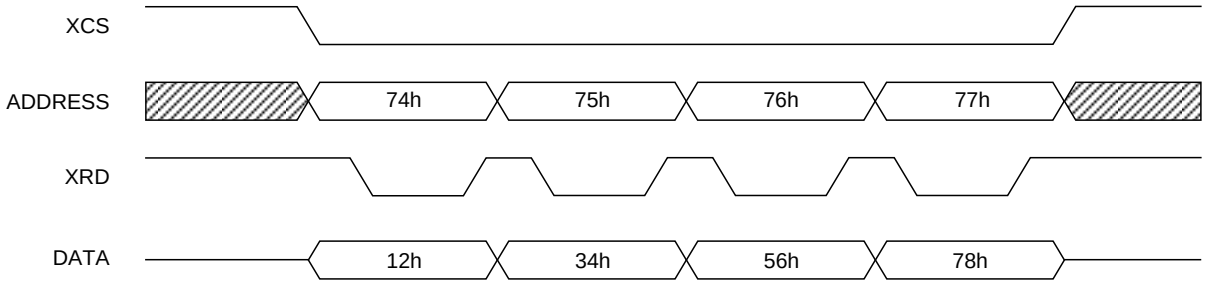
The CFR ATFWrite/ARFRead register is read.



The data read is "F0000220h".

(6) Fourth quadlet of the received packet Read

The CFR ATFWrite/ARFRead register is read.



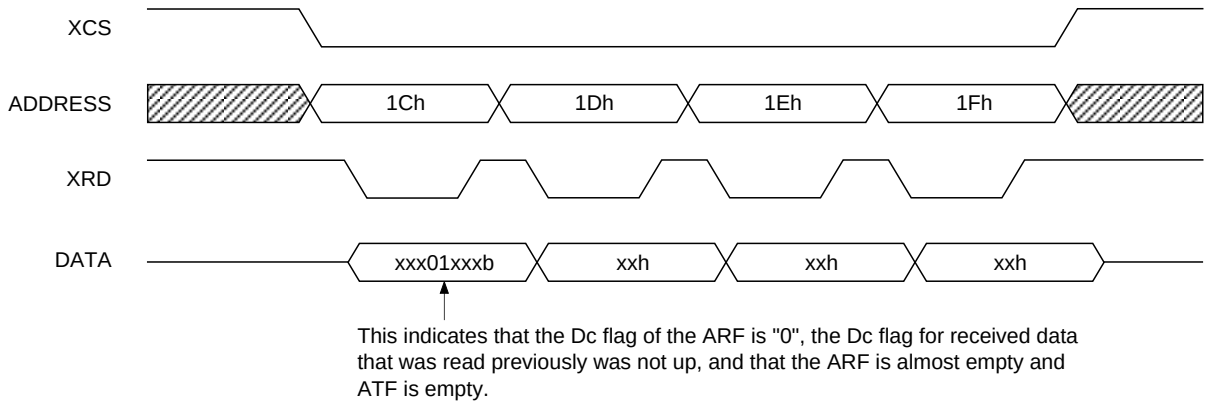
The data read is "12345678h".

(7) Checking for remaining packets still in FIFO

Four quadlets were read in preceding items (1) to (6). They were read continuously because Arf4There was high.

If Arf4There was low, Async Status must be read after one quadlet is read, to find out if ArfEmpty is high.

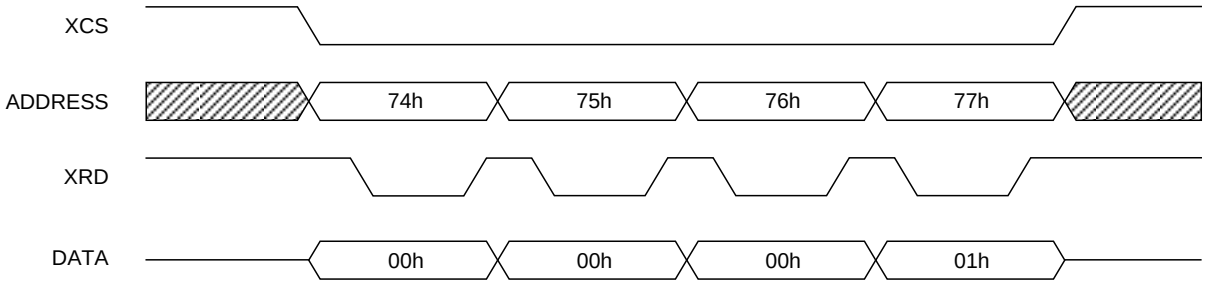
Even if Arf4There is high, as in this case, after the fourth quadlet read must be done while checking ArfEmpty and ArfDc in the same way.



In the above example the Arf4There bit is low, so a maximum of three more quadlets can be predicted, but the ArfAEmpty bit is high, so there is only one more quadlet in FIFO.

(8) Fifth quadlet of the received packet Read

The CFR ATFWrite/ARFRead register is read.



The data read is "00000001h".

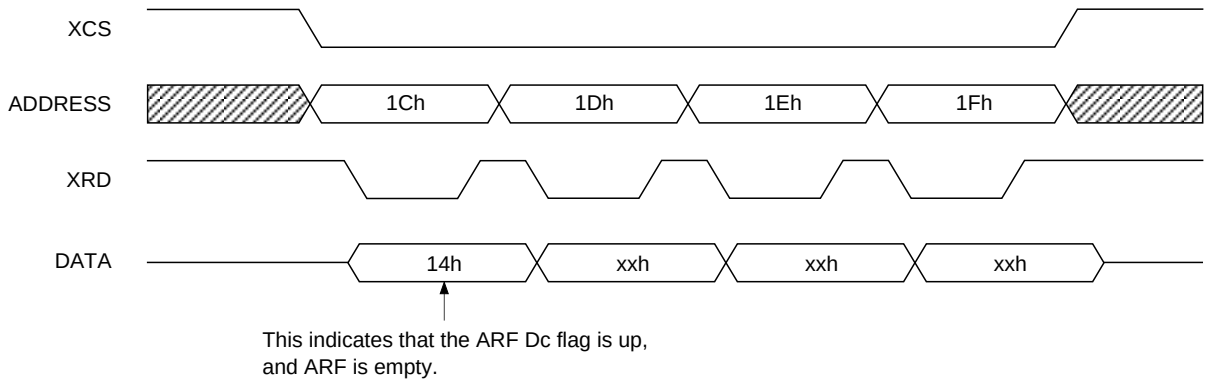
The lower 4 bits of this quadlet are the ackSent field, and this indicates "01h" transmitted as this packet's Ack_code. This is always written even if the packet is one which does not have Ack_code transmitted, such as a broadcast packet.

If this value is "04h", the ARF may have become full during receive and quadlets may be missing.

If it is "0Dh", an error was detected in the data field CRC check of the received packet, or data_length and the actual data length do not match.

(9) Checking for remaining packets still in FIFO

The last quadlet was read in (8) above, so there should be no more packets in FIFO. This is checked as follows.



This confirms that the ARF is empty.

This completes Asynchronous packet reception.

6-5. CXD3220R Data Format

6-5-1. Asynchronous Transmit

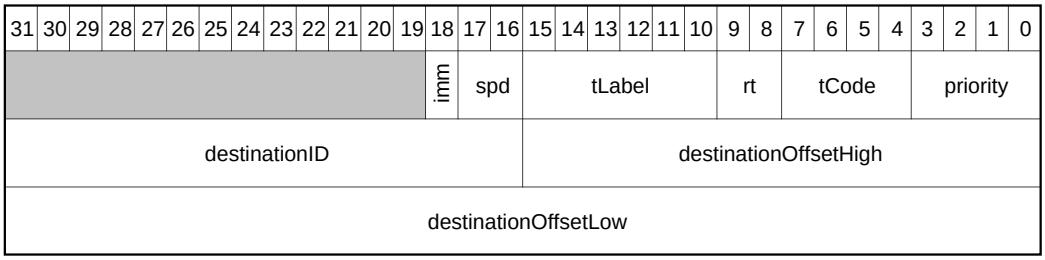
The following are the four basic formats for Asynchronous data during transmit.

a) No-data Packets (Used for Quadlet Read requests and all Write responses.)
b) Quadlet Packets (Used for Quadlet Write requests, Quadlet Read responses and Block Read requests.)
c) Block Packets (Used for Lock requests, Lock responses, Block Write requests and Block Read responses.)
d) Unformatted data

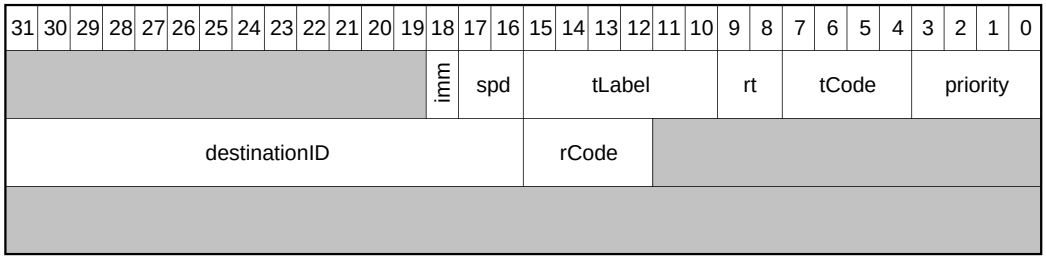
6-5-1-1. No-data Transmit

The data format for no-data transmit is shown below.
The first quadlet contains packet control information. The second and third quadlets contain 16-bit Destination ID and 48-bit Destination Offset for request, or Response code for response.

Quadlet Read Request Transmit Format



Write Response Transmit Format



No-data Transmit Fields

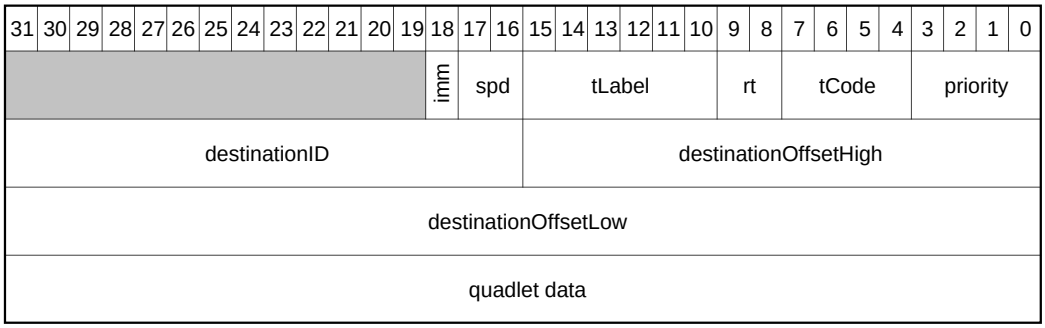
Field Name	Description
imm	Immediately tries to transmit continuously after Acknowledge is sent, if "1" is set. (Used for Phy Read and Lock Response.)
spd	Transmit speed 00: 100Mbps; 01: 200Mbps; 10: 400Mbps; 11: Reserved
tLabel	Transaction Label. Used as a pair with response packet relative to request packet.
rt	This packet's Retry code. 00: Retry_1; 01: Retry_X, 10: Retry_A, 11: Retry_B
tCode	This packet's Transaction code.
priority	This packet's Priority level. For values other than "0", the transmitter uses Priority Arbitration relative to this packet.
destinationID	Indicates this packet's Destination bus number in 10 bits and the Node number in 6 bits.
destinationOffsetHigh, destinationOffsetLow	These two continuous areas indicate Destination Node address space address. This address must be in quadlet units.
rCode	Response code for write response packet.

6-5-1-2. Quadlet Transmit

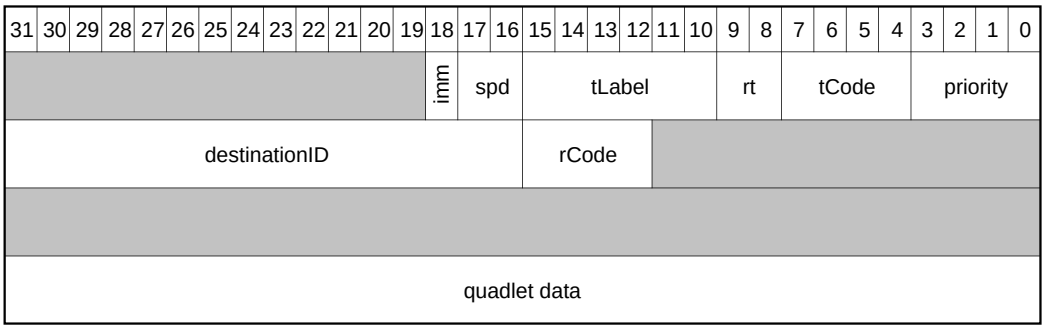
The data format for quadlet transmit is shown below.

The first quadlet contains packet control information. The second and third quadlets contain 16-bit Destination ID and 48-bit Destination Offset for request, or Response code for response. The fourth quadlet is quadlet data for Read response and Quadlet Write request, and Data Length and Reserved for Block Read request.

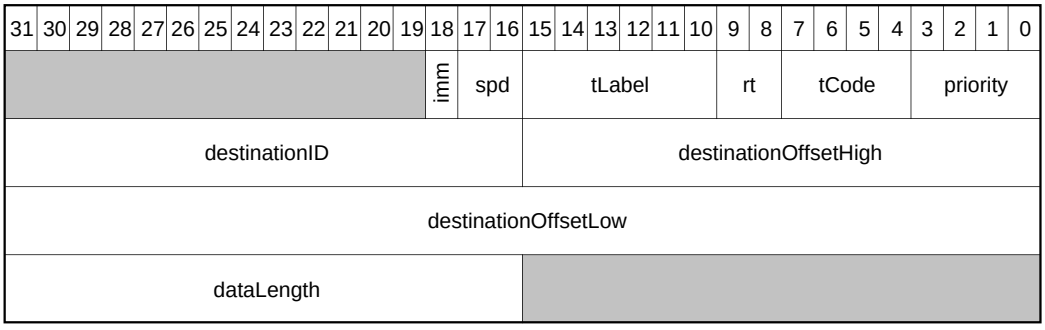
Quadlet Write Request Transmit Format



Quadlet Read Response Transmit Format



Block Read Request Transmit Format



Quadlet Transmit Fields

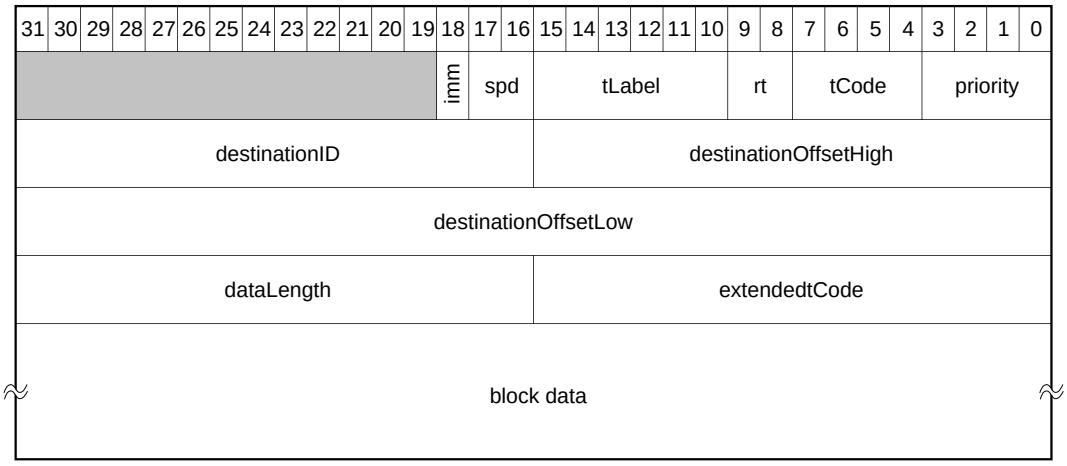
Field Name	Description
imm	Immediately tries to transmit continuously after Acknowledge is sent, if "1" is set. (Used for Phy Read and Lock Response.)
spd	Transmit speed. 00: 100Mbps; 01: 200Mbps; 10: 400Mbps; 11: Reserved
tLabel	Transaction Label. Used as a pair with response packet relative to request packet.
rt	This packet's Retry code. 00: Retry_1; 01: Retry_X, 10: Retry_A, 11: Retry_B
tCode	This packet's Transaction code.
priority	This packet's Priority level. For values other than "0", the transmitter uses Priority Arbitration relative to this packet.
destinationID	Indicates this packet's Destination bus number in 10 bits and the Node number in 6 bits.
destinationOffsetHigh, destinationOffsetLow	These two continuous areas indicate Destination Node address space address. This address must be in quadlet units.
quadlet data	Writes transmitted data for Quadlet Write requests and Quadlet Read response.
rCode	Response code for Quadlet response packet.
dataLength	Writes how many bytes requested for Block Read request.

6-5-1-3. Block Transmit

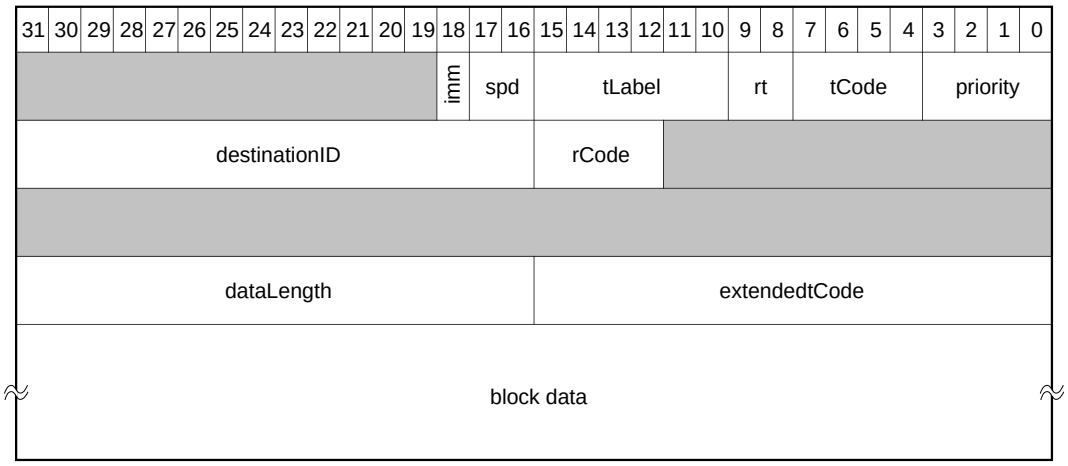
The data format for block transmit is shown below.

The first quadlet contains packet control information. The second and third quadlets contain 16-bit Destination ID and 48-bit Destination Offset for request, or Response code for response. The fourth quadlet contains Data Length and Extended Transaction code (all "0" except for Lock Transaction).

Block Transmit Format



Block Read or Lock Response Transmit Format



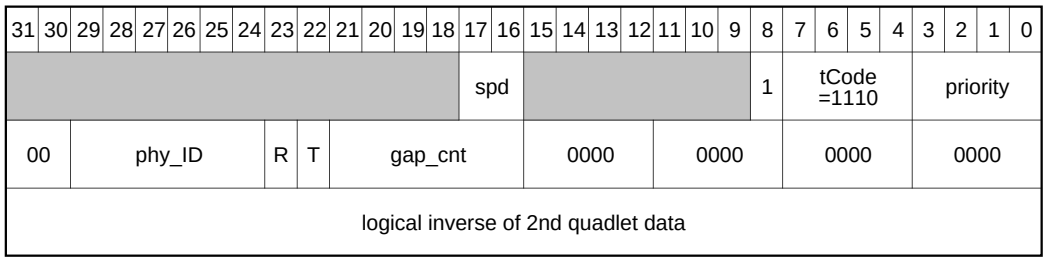
Block Transmit Fields

Field Name	Description
imm	Immediately tries to transmit continuously after Acknowledge is sent, if "1" is set. (Used for Phy Read and Lock Response.)
spd	Transmit speed. 00: 100Mbps; 01: 200Mbps; 10: 400Mbps; 11: Reserved
tLabel	Transaction Label. Used as a pair with response packet relative to request packet.
rt	This packet's Retry code. 00: Retry_1; 01: Retry_X, 10: Retry_A, 11: Retry_B
tCode	This packet's Transaction code.
priority	This packet's Priority level. For values other than "0", the transmitter uses Priority Arbitration relative to this packet.
destinationID	Indicates this packet's Destination bus number in 10 bits and the Node number in 6 bits.
destinationOffsetHigh, destinationOffsetLow	These two continuous areas indicate Destination Node address space address.
quadlet data	Writes transmitted data for Quadlet Write request and Quadlet Read response.
rCode	Response code for Quadlet response packet.
dataLength	Writes how many bytes requested for Block Read request.
extendedtCode	Specifies actual Lock Action performed by this packet data when tCode is a Lock Transaction.
block data	Writes transmitted data. This data is not written in FIFO when dataLength = 0. The first byte of the block must indicate the upper byte of the first data, regardless of data Destination or Source listing.

6-5-1-4. Unformatted Transmit (Phy Configuration Packet)

The data format for unformatted transmit during Phy Configuration packet transmit is shown below.
The first quadlet contains packet control information. The remaining quadlets contain data, and get on the bus and are transmitted regardless of format. There is no CRC attached to the packet data.
Further, there is no CRC attached to the first quadlet.
Logical-inverse is not added at Link Core, so it must be added when transmitting.

Unformatted Transmit Format 1 (Phy Configuration Packet)



Unformatted Transmit (Phy Configuration Packet) Fields

Field Name	Description
00	Indicates that the transmit packet is a Phy configuration packet.
phy_ID	Sets the force_root bit of the node with this Phy_ID sets to "1". (Only valid when R is set to "1".)
R	Sets the force_root bit of the node with this Phy_ID to "1" when "1", and clears other nodes' force_root bit. The Phy_ID area is ignored when "0".
T	Sets the value of the Phy_CONFIGURATION.gap_Count of the Phy register to the value of gap_cnt when "1".
gap_cnt	Indicates values of all node new Phy_CONFIGURATION.gap_Count. These values are received immediately and stored in the register. It becomes valid after the next bus reset.

6-5-1-5. Unformatted Transmit (Link-on Packet)

The data format for unformatted transmit during Link-on packet transmit is shown below.

The first quadlet contains packet control information. The remaining quadlets contain data, and are transmitted regardless of format. There is no CRC attached to the packet data. Further, there is no CRC attached to the first quadlet.

Logical-inverse is not added at Link Core, so it must be added when transmitting.

Unformatted Transmit Format 1 (Link-on Packet)

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
														spd									1	tCode =1110				priority			
01		phy_ID						0000				0000				0000				0000				0000				0000			
logical inverse of 2nd quadlet data																															

Unformatted Transmit (Link-on Packet) Fields

Field Name	Description
01	Indicates that the transmit packet is a Link-on packet.
phy_ID	Indicates the Phy chip ID of this packet's Destination.

6-5-2. Asynchronous Receive

The following are the three basic formats for Asynchronous data during receive.

a) No-data Packets (Used for Quadlet Read requests and all Write responses.)
b) Quadlet Packets (Used for Quadlet Write requests, Quadlet Read responses and Block Read requests.)
c) Block Packets (Used for Lock requests, Lock responses, Block Write requests and Block Read responses.)

The names of received data areas and their contents are given below.

Asynchronous Receive Fields

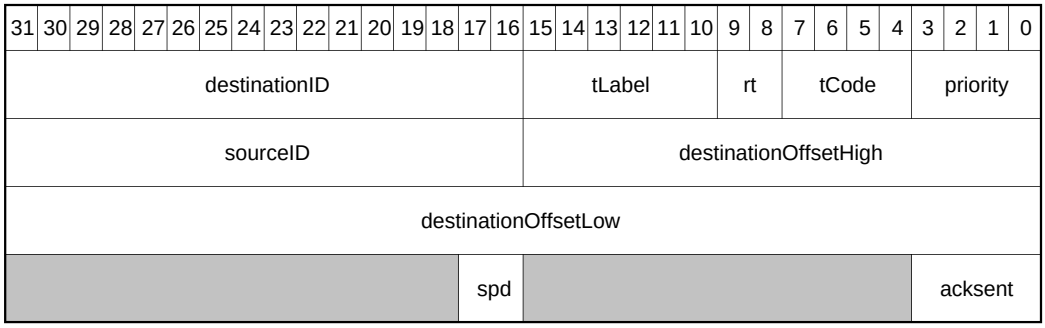
Field Name	Description
destinationID	This node's BusNumber (all "0" if "local bus") and NodeNumber (all "1" if broadcast).
tLabel	Transaction Label. Used as a pair with response packet relative to request packet.
rt	This packet's Retry code. 00: Retry_1; 01: Retry_X, 10: Retry_A, 11: Retry_B
tCode	This packet's Transaction code.
priority	This packet's Priority level.
sourceID	The Node ID of the node that sent this packet.
destinationOffsetHigh, destinationOffsetLow	These two continuous areas indicate Destination Node address space address.
rCode	Response code for response packet.
quadlet data	Received data is written for Quadlet Write requests and quadlet read response.
dataLength	The number of bytes in received block type's packet data.
extendedtCode	Specifies actual Lock Action performed by this packet data when tCode is a Lock Transaction.
block data	Received data is written. This data is not written in FIFO when dataLength = 0. The first byte of the block must indicate the upper byte of the first data, regardless of data Destination or Source listing.
spd	Speed of received packet. 00: 100Mbps; 01: 200Mbps; 10: 400Mbps; 11: Reserved
acksent	Acknowledge code sent by Link Core relative to this packet is written.

6-5-2-1. No-data Receive

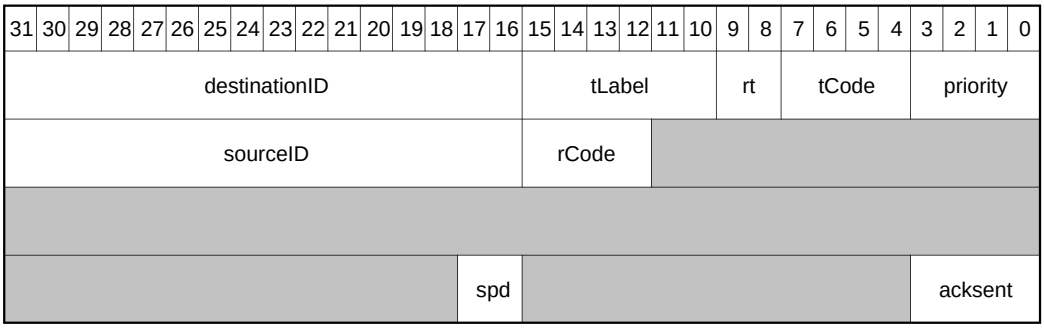
The data format for no-data receive is shown below.

The first quadlet contains the Destination ID and other packet headers. The second and third quadlets contain 16-bit Source ID and 48-bit Destination Offset for request, or Response code for response. The last quadlet contains packet receive status.

Quadlet Read Request Receive Format



Write Response Receive Format

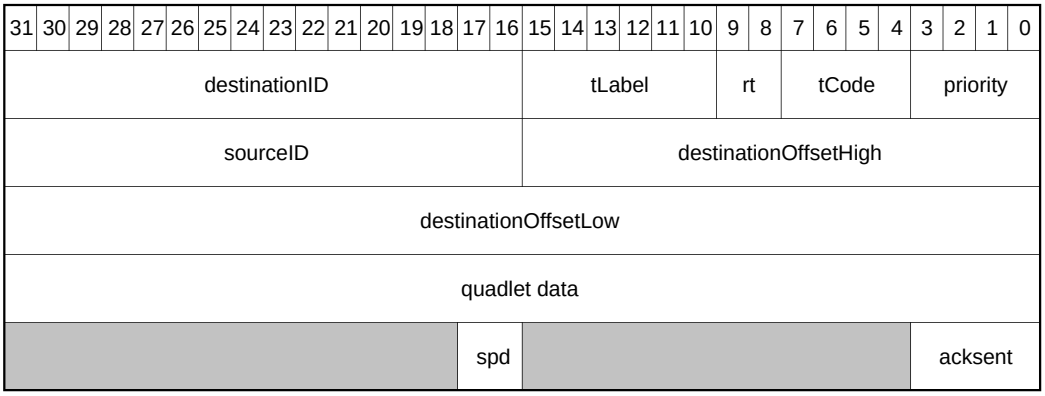


6-5-2-2. Quadlet Receive

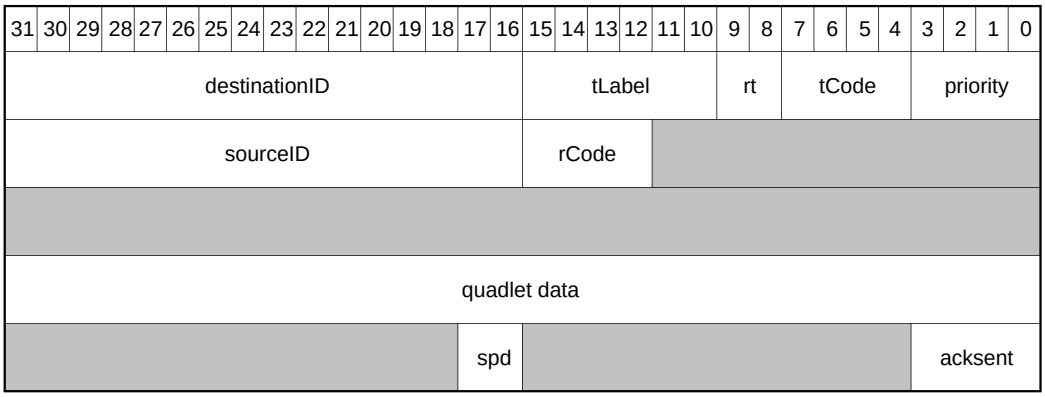
The format for Quadlet Receive is shown below.

The first quadlet contains the Destination ID and other packet headers. The second and third quadlets contain 16-bit Source ID and 48-bit Destination Offset for request, or Response code for response. The fourth quadlet contains data for Read request and Quadlet Write request, and Data Length and Reserved for Block Read request. The last quadlet contains packet receive status.

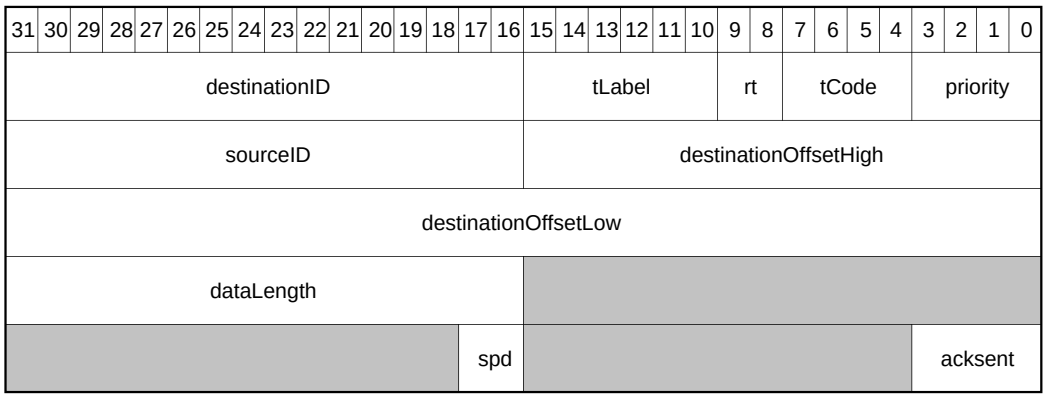
Quadlet Write Request Receive Format



Quadlet Read Response Receive Format



Block Read Request Receive Format

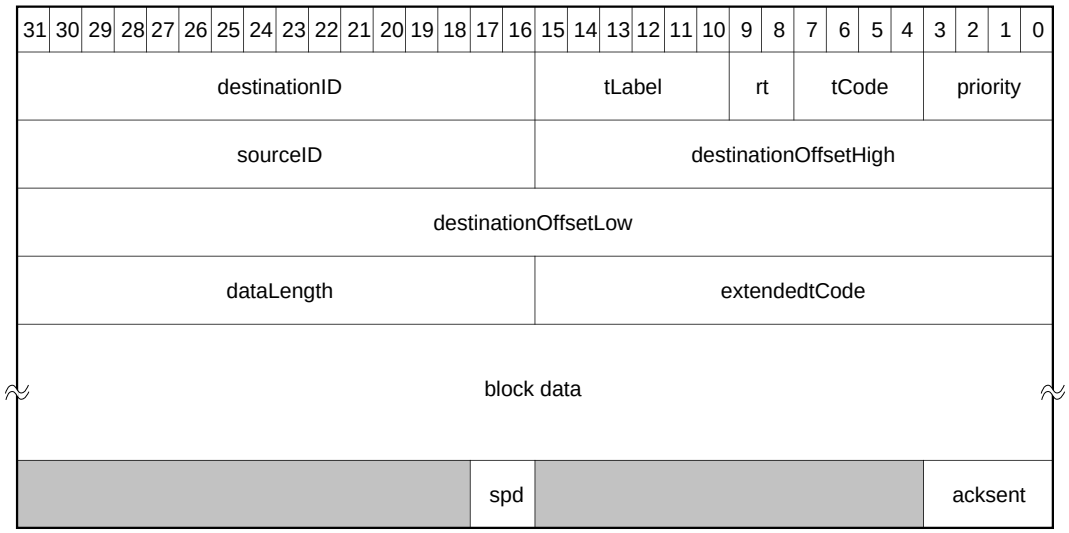


6-5-2-3. Block Receive

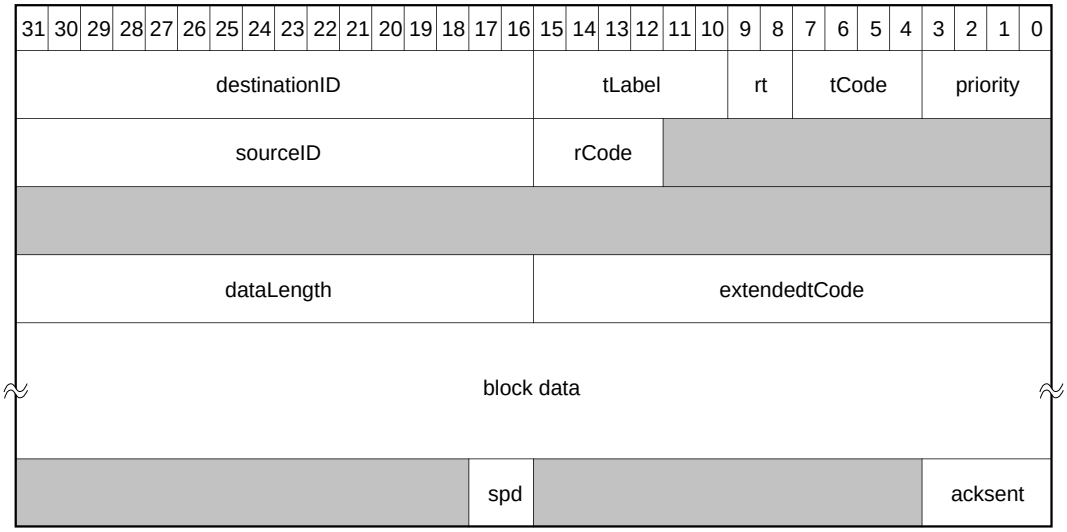
The format for Block Receive is shown below.

The first quadlet contains the Destination ID and other packet headers. The second and third quadlets contain 16-bit Source ID and 48-bit Destination Offset for request or the Response code for response. The fourth quadlet contains Data Length and Extended Transaction code (all "0" except for Lock Transaction). This is followed by Block data. The last quadlet contains packet receive status.

Block Write or Lock Request Receive Format



Block Read or Lock Response Receive Format



6-6. Self-ID Packet Receiving Error Processing

In the Self ID phase after bus reset on the CXD3220R, if the Self ID packet could not be received correctly, Self ID packet receive is stopped immediately and the Node_sum value becomes "0".

The external CPU thus can judge that the Self ID phase could not be completed correctly.

7. ADP (Asynchronous Data Pipe)

The CXD3220R is equipped with a function referred to as ADP for automatically transmitting and receiving the data of computer peripherals in the form of Asynchronous packets based on the SBP-2 protocol.
A dedicated I/O data bus and several control signal pins are used to perform exchange of data with the decoder/encoder of various systems (see below).
This function also supports the transport data I/F compatible with both 8-bit and 16-bit data.

Name	Width	I/O	Description
SD	16	I/O	Data bus
XHWR	1	Input	Data write strobe signal
XHRD	1	Input	Data read strobe signal
SDRQ	1	Output	Data request signal
XSAC	1	Input	Acknowledge signal for SDRQ

7-1. Built-in FIFO

The CXD3220R is equipped with a built-in dedicated FIFO for SBP2 data transfer.
There are two types of FIFO: a large-capacity FIFO_A and a small-capacity FIFO_B. The capacity of FIFO_A is 532 quadlets, and is able to contain two 1 KB Asynchronous packets. The capacity of the FIFO_B is 12 quadlets.
Use the CXD3220R with a maximum Asynchronous packet size of 1024 bytes.

7-2. Transport Data I/F

7-2-1. Data Bus

This data interface is 8 bits/16 bits, and switching is done by accessing the CFR AIDT16 register. (The default value is 16 bits.)

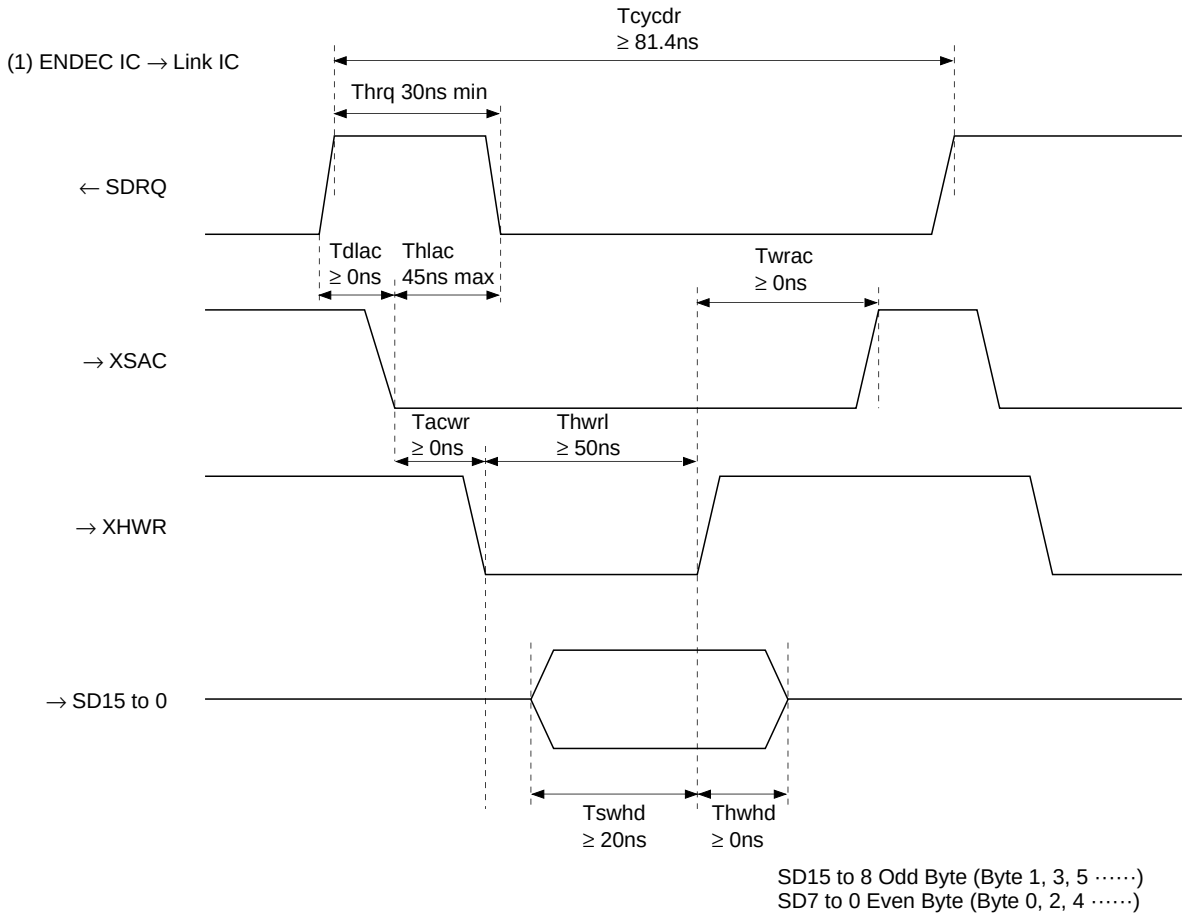
7-2-2. Transmit Interface

The CXD3220R supports only Asynchronous communication for a single login of one initiator per target. It does not support communication of multiple transport data with Asynchronous packets. The ADP cannot be used for simultaneous transmission and reception.

The timing chart for the interface is indicated below.

The restriction on the transport data SDRQ output frequency is 12.288MHz (max.).

Transmit Interface



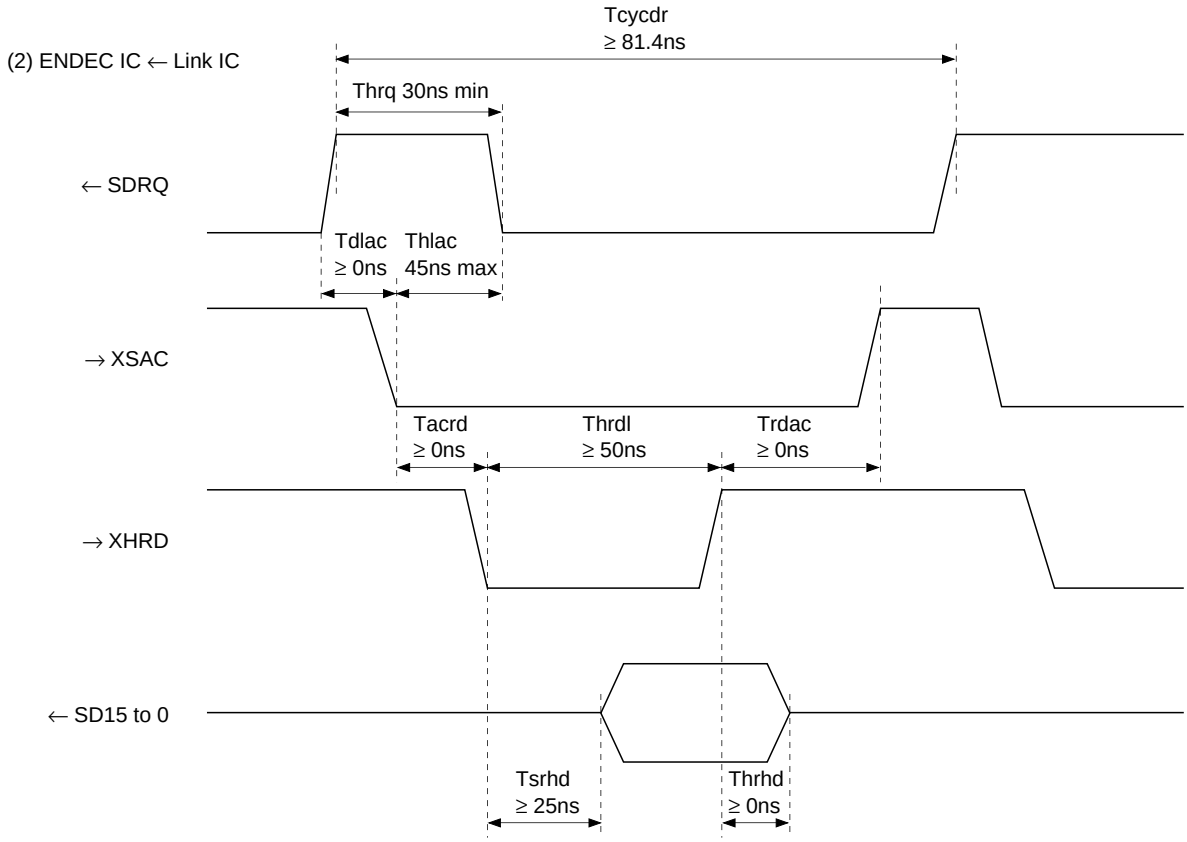
7-2-3. Received Interface

The CXD3220R only supports Asynchronous communication for a single login of one initiator per target. It does not support communication of multiple transport data with Asynchronous packets. The ADP cannot be used for simultaneous transmission and reception.

The timing chart is indicated below.

The restriction on the transport data SDRQ output frequency is 12.288MHz (max.).

Received Interface



7-3. Asynchronous Data Pipe (ADP)

Performance is an important factor for data transfer of computer peripherals. Although the Transaction Layer has conventionally been controlled mainly with software, the CXD3220 realizes control with hardware. The result is faster data transfer processing.
The CXD3220R contains a built-in circuit referred to as ADP that controls the 1394 Asynchronous Transaction Layer in accordance with the IEEE1394 protocol. This ADP enables packet transfer to be performed automatically via the 1394 serial bus in accordance with SBP-2 protocol.
Consequently, sequences based on the SBP-2 protocol such as ORB (operation request block) fetch, data transfer, status transmission to the initiator and so forth can be simplified, enabling the use of the optimum design when connecting the data of a disk drive, tape streamer or other computer peripherals to IEEE1394.

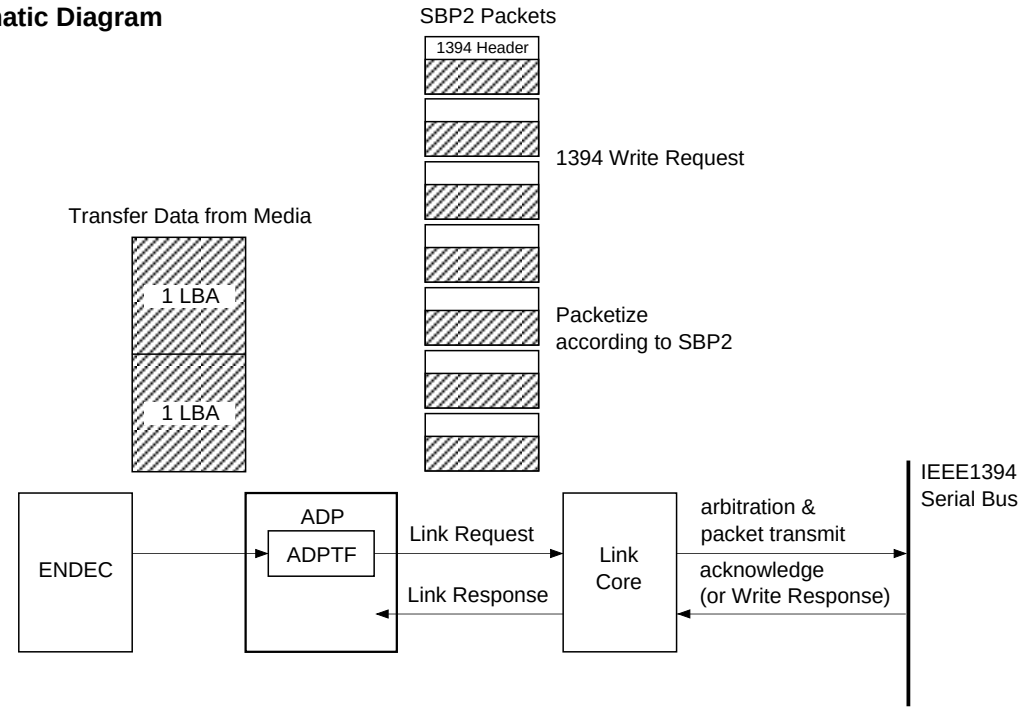
7-3-1. ADP Sequence (Data Transfer)

The following provides an explanation of those ADP functions relating to transmission based on the diagram below.
As is defined in the SBP-2 protocol, data transfer with respect to transmission is initiated by a target. Therefore, data transfer is performed in the form in which the target (this link IC) is written into initiator memory using the 1394 Write Block Command for the initiator (e.g., host computer). The following illustrates a brief overview of that sequence.

1) First, the CPU initializes the ADP register according to the contents of the ORB obtained.
2) Once the ADP has been initiated, the CXD3220 asserts the SDRQ and begins to request data from the decoder. Data output from the decoder in synchronization with the SDRQ is input to the FIFO of the ADP (ADPTF).
3) Data is then packetized in accordance with the SBP-2 data transfer format, and packet data to which the 1394 Header has been added is automatically stored in the ADPTF. (A Block Write request is used for the packets.)
4) When data equal to or greater than the size of one 1394 packet is read into the ADPTF, that data is sent to the 1394 Link Core. The Link Core then applies Arbitration to the 1394 bus (via Phy IC).
5) Once a bus has been acquired, the Block Write request containing the transport data is transmitted to the initiator.
6) After transmission, an Ack code for the Write request packet and, depending on the case, a Write response packet, are sent to the target from the initiator.
   If this Ack code and Response code are normal, the ADP transmits the next data. In the case of an error, the status is returned as an error and an Interrupt is generated to the CPU.
7) The tCode and Transaction Label of packets that are received by the Link Core are checked, and only the response packet returned to the ADP is input to the ADP. (This is not stored in the ADPRF.) Other packets are input to the ARF.
8) The ADP writes the rCode of the received response packet into the register.

This is the role of the ADP with regard to transmission.

ADP Schematic Diagram



7-3-2. ADP Sequence (Data Receive)

The following provides an explanation of those ADP functions relating to reception based on the diagram below.

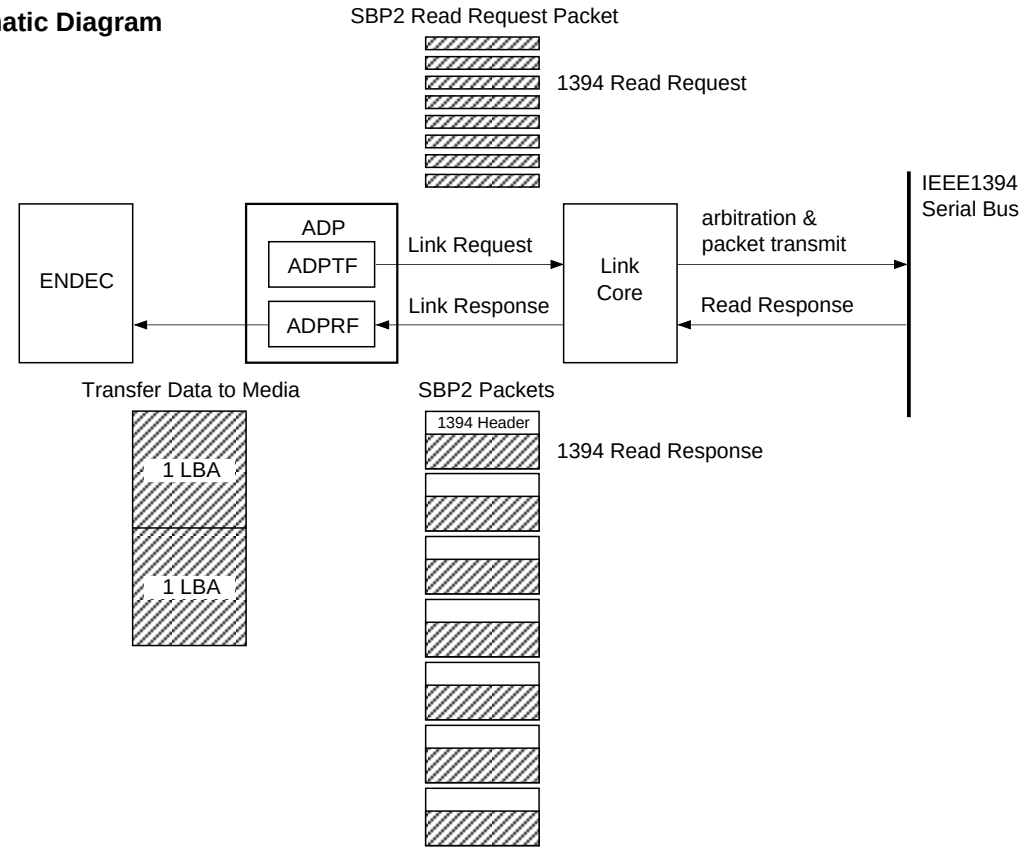
As is defined in the SBP-2 protocol, data transfer with respect to reception is also initiated by a target.

Therefore, data transfer is performed in the form in which the target (this link IC) reads the initiator memory using the 1394 Read Block command for the initiator (e.g., host computer). The following illustrates a brief overview of that sequence.

1) First, the CPU initializes the ADP register according to the contents of the ORB obtained.
2) The designated address and 1394 Block Read request command corresponding to the Data_length are stored in the ADPTF FIFO in accordance with the SBP-2 data transfer format. The stored Read request packet is then sent to the 1394 Link Core, and the Link Core applies Arbitration to the 1394 bus (via Phy IC).
3) Once a bus has been acquired, a Block Read request packet is transmitted to the initiator.
4) After transmission, an Ack code for the Read request packet and a Read response packet containing data corresponding to the Data_length are sent to the target from the initiator.
   If this Ack code and Response code are normal, the ADP transmits the next data. In the case of an error, the status is returned as an error and an interrupt is generated to the CPU.
5) The tCode and Transaction Label of packets that are received by the Link Core are checked, and only the response packet returned to the ADP is input to the ADP. (Stored in the ADPRF.) Other packets are input to the ARF.
6) The ADP removes the 1394 Header from the received response packet, and automatically stores the packet data in the ADPRF.
7) Data stored in the ADPRF is output to a peripheral LSI from the transport I/F in synchronization with the SDRQ.

This is the role of the ADP with regard to reception.

ADP Schematic Diagram



7-4. ADP Structure and Functions

7-4-1. ADP Functions

Retry Function

The ADP is equipped with a function that retransmits a request packet when the Ack code of ack_busy_* has returned after transmitting that request packet. The CXD3220R supports only single-phase retry. When resending a packet, the ADP transmits after changing the rt code from 00 to 01 (retry_X). The time interval during retransmission is defined with the retry_interval set with the Transaction Timeout register. When retransmission has been retried for the number of times set for the retry_limit with the Transaction Timeout register, and ack_busy_* is still returned, this is considered to be an error and a busy_timeout is set in the err code of the ADP Status register followed by generation of ADPErr Interrupt.

Split Timeout Detection Function

The ADP is equipped with a Split Timeout detection function that detects the Timeout until a response packet returns in the case of a Split Transaction. After a request packet has been transmitted during a Split Transaction, when the response packet has not returned even after the Split Timeout time defined in the Transaction Timeout register has elapsed, a busy_timeout is set in the err code of the ADP Status register followed by the generation of Interrupt.

7-4-2. ADP Structure

Switching Between Transmission and Reception (FIFO Switching)

Switching between transmission and reception (FIFO switching) is controlled with the Direction (d) bit of the ADP4 register. Two types of FIFO are available to the ADP consisting of a 2KB FIFO and 48 byte FIFO. During transmission, the 2KB of FIFO becomes the ADPTF (ADP Transmit FIFO), and the 48 bytes of FIFO is not used. During reception, the 48 byte FIFO becomes the ADPTF, and the 2 KB FIFO becomes the ADPRF (ADP Receive FIFO). Switching between ADP transmission and reception, including this FIFO switching, is controlled with the Direction (d) bit of the ADP5 register.

The d bit is read into the ADP when it is started, and the direction of transmission and reception cannot be changed until the ADP is finished.

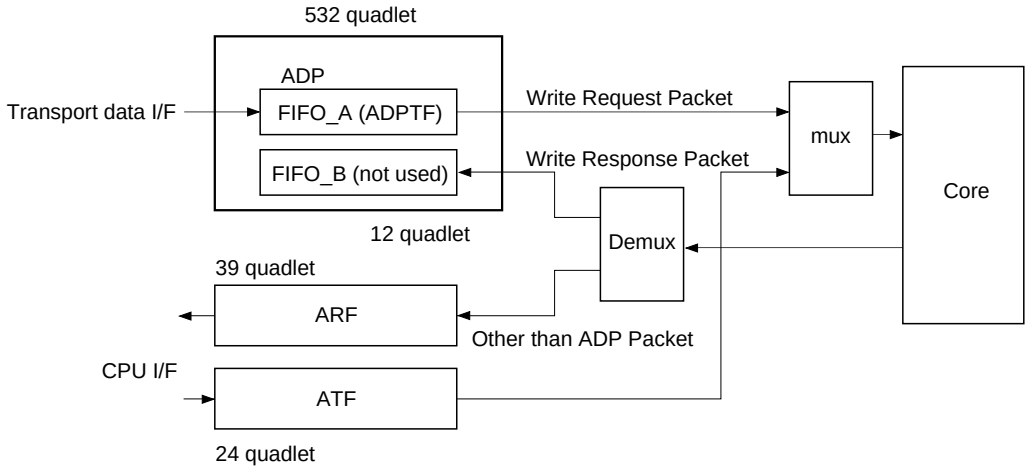
Parallel Two-Pair Transmission and Reception FIFO

Other FIFO such as the ARF and ATF are also available to the CXD3220R in parallel with ADP FIFO. This enables it to perform transmission and reception of normal 1394 packets other than data in parallel with data exchange performed by the ADP FIFO.

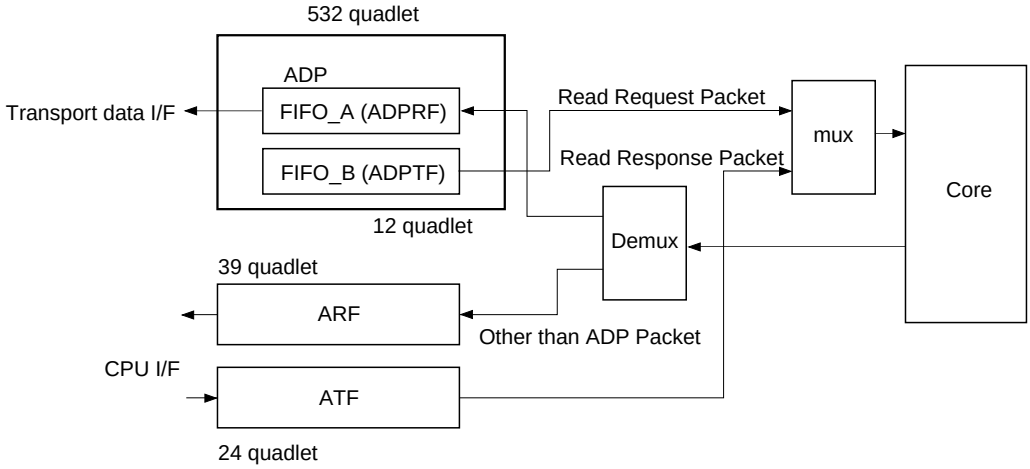
For example, this enables the CXD3220R to accommodate the following:

- Response when a read request has arrived at a CSR or Configuration ROM from another node during data transfer.
- Response to a Task Management ORB from the initiator during data transfer.

FIFO Structure and Operation during ADP Transmission



FIFO Structure and Operation during ADP Reception



7-5. ADP Setting

According to the SBP-2 protocol, a normal command block ORB (shown below) is fetched in the form of a Block Read response packet from the initiator as a result of the target sending a Block Read response packet to the initiator.
The Block Read response packet containing the normal command block ORB is incorporated into the ARF in the format shown below. (Refer to the draft of the Serial Bus Protocol2 (SBP-2) for a detailed explanation regarding the SBP-2 protocol.)

The following settings are made in the case of automatic data transfer by the ADP after fetching the ORB. To begin with, there are the 3 modes indicated below for automatic data transfer by the ADP.

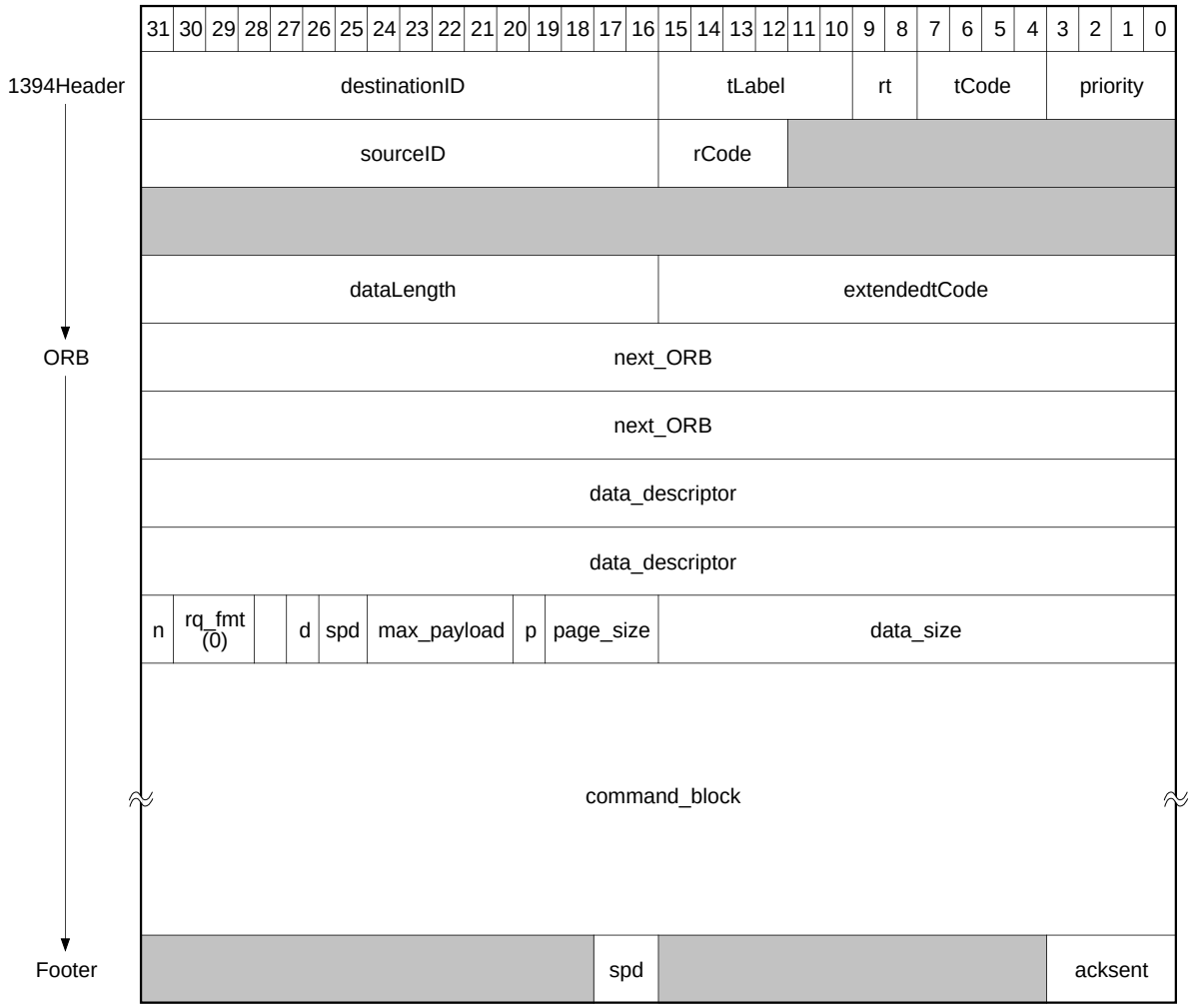
Mode 0) Page_table_present (p) = 0, Page_size = 0
When page_table_present (p) = 0, page_size = 0 are set, the ADP enters a transfer mode in which there are no restrictions on address boundary. In this case, since the address of the initiator is directly indicated in the data_descriptor obtained with the normal command block ORB, set the node_ID & offset_hi & offset_lo of the data_descriptor to the Destination ID & destinationOffsetHigh & destinationOffsetLow of the ADP register. Also set d, spd, max_payload, p, page_size and data_size. ADP transfer starts when the ADPgo bit of the ADP control register is set to "1".
The transfer speed of request packets generated with the ADP is selected between either S100 for spd = 0 or S200 for spd = 1.

There are no restrictions on packet size in this mode. Request packets are generated corresponding to the data_size represented with max_payload, and data is transmitted and received sequentially. Finally, data corresponding to the number of bytes remaining is transmitted and received with a request packet.

The following page indicates a summary of packetizing with respect to this mode.

Basic Configuration

Block Read or Clock Response Receive Format



Mode 0 (Page_table_present = 0, Page_size = 0, Not use page boundary)

Map of ADP Control Registers

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
													imm			tLabel						(rt)		(tCode)				priority			
destinationID																destinationOffsetHigh															
destinationOffsetLow																															
				d		spd			max_ payload			P	page_ size			xfer_length (data buffer length)															
Control																															
Status																															

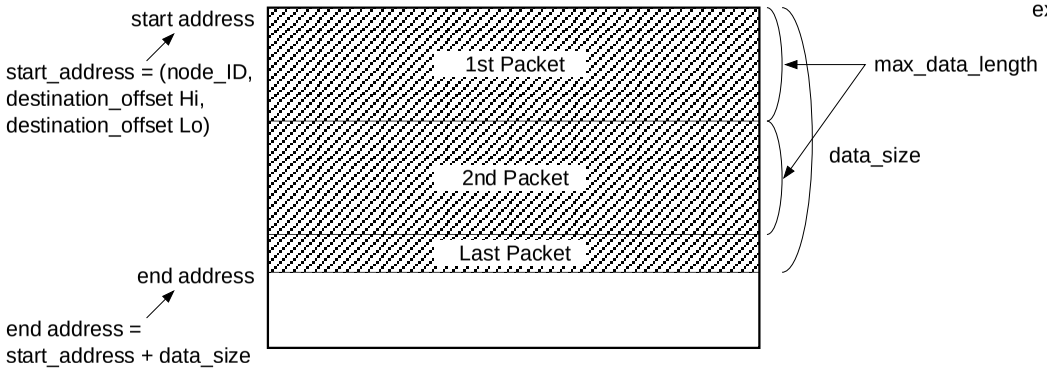
ADP Registers

p (RW, 1bit)	page_table_present = 0, page_size = 0 → mode 0
page_size (RW, 3bits)	= 0
tl (RW, 6bits)	Transaction Label
destination_ID (RW, 16bits)	
destination_offset Hi (RW, 16bits)	
destination_offset Lo (RW, 32bits)	
d (RW, 1bit)	Direction
	0 (Target ← Initiator, 1 (Target → Initiator)
sp (RW, 3bits)	Speed. 0 (S100),1 (S200)
max_payload (RW, 4bits)	data_length (16bits) ≤ 2^ (max_payload + 2) [Byte]
xfer_length (RW, 16bits)	data buffer length [Byte]
Control (RW, 32bits)	
Status (RO, 32bits)	

1394 Header Control by ADP

tCode (4bits)	Transaction Code. tCode = 0 (Quadlet Write),
	1 (Block Write).
	4 (Quedlet Read), 5 (Block Read)
rt (2bits)	retry code. 00 (first packet), 01 (retry)
extended_tCode (16bits)	This value must be zero

Data Transfer



Mode1) Page_table_present (p) = 0, Page_size = non-zero, Use page boundary

When page_table_present (p) = 0 and page_size = non-zero are set, the ADP enters a transfer mode in which there are restrictions on address boundary. In this case, since the address of the initiator is directly indicated in the data_descriptor obtained with the normal command block ORB, set the node_ID & offset_hi & offset_lo of the data_descriptor to the Destination ID & destinationOffsetHigh & destinationOffsetLow of the ADP register. Also set d, spd, max_payload, p, page_size and data_size. ADP transfer starts when the ADPgo bit of the Command register is set to "1".
The transfer speed of request packets generated with the ADP is selected between either S100 for spd = 0 or S200 for spd = 1.

In this mode, there is a restriction in the form of an address boundary in which data must not be transferred across this address. The address boundary is the address where the 1394 serial bus address lower (page_size + 8) bits change from all "1" to all "0".
Thus, transmission and reception of data that is larger than the corresponding data_size represented with max_payload or data that crosses the address boundary are performed by dividing the packet.

The following page indicates a summary of packetizing with respect to this mode.

Mode 1 (Page_table_present = 0, Page_size = nonzero, Use page boundary)

Map of ADP Contorl Registers

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
													imm			tLabel						(rt)		(tCode)			priority			
destinationID																destinationOffsetHigh															
destinationOffsetLow																															
				d		spd		max_ payload			P		page_ size			xfer_length (data buffer length)															
Control																															
Status																															

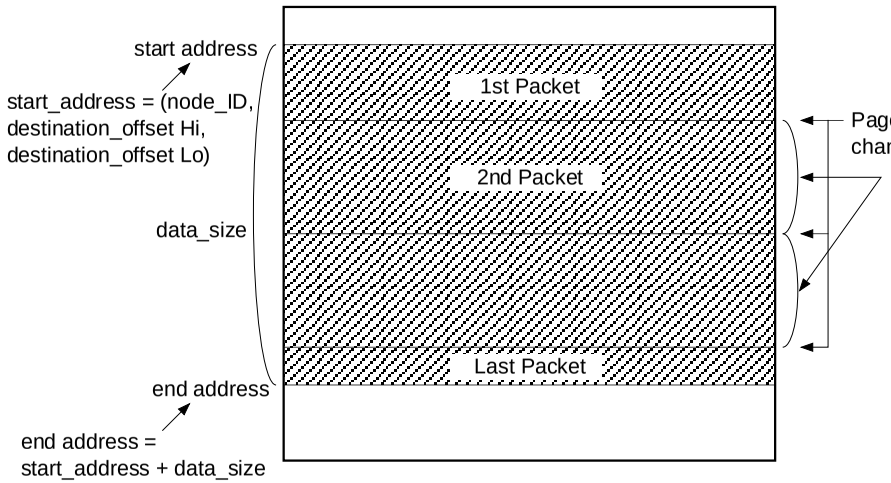
ADP Registers

p (RW, 1bit)	p = 0, page_size = nonzero → mode 1
page_size (RW, 3bits)	= nonzero, page_length = 2^ (page_size + 8) [Byte]
tl (RW, 6bits)	Transaction Label
destination_ID (RW, 16bits)	
destination_offset Hi (RW, 16bits)	
destination_offset Lo (RW, 32bits)	
d (RO, 1bit)	Direction 0 (Target ← Initiator), 1 (Target → Initiator)
sp (RW, 3bits)	Speed. 0 (S100),1 (S200)
max_payload (RW, 4bits)	data_length (16bits) ≤ 2^ (max_payload + 2) [Byte]
xfer_length (RW, 16bits)	data buffer length [Byte]
Control (RW, 32bits)	
Status (RO, 32bits)	

1394 Header Control by ADP

tCode (4bits)	Transaction Code. tCode = 0 (Quadlet Write), 1 (Block Write). 4 (Quedlet Read), 5 (Block Read)
rt (2bits)	retry code. 00 (first packet), 01 (retry)
extended_tCode (16bits)	This value must be zero

Data Transfer



Restriction

1) Block Read or Block Write Transaction does not cross address boundary.
2) The maximum data payload of one packet is 2^ (max_payload + 2).

Mode 2) Page_table_present (p) = 1, Page_size = non-zero, Use page boundary

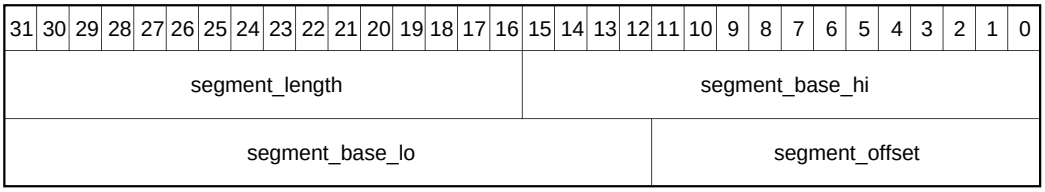
[Page Table Access]

This mode is used in the case of using the page_table, which is an indirect address table of the SBP-2 protocol. When page_table_present (p) = 1 and page_size = non-zero are set, the ADP enters a transfer mode using the page_table with the address boundary restriction in effect.

According to the SBP-2 protocol, in the case page_table_present (p) obtained with a normal command block ORB is equal to "1", since the address of the page_table is indicated in the data_descriptor, it is necessary that a Block Read request described in section 6.5.1.2 be issued to the address indicated in the data_descriptor, and that the contents of each page_table (see diagram below) be read. The Block Read response packet is returned from the initiator and stored in the ARF in the format of a Block Read response (see section 6.5.2.3).

In this mode, there is a page in which segment_offset changes from all "0" to all "1" in the form of a page boundary. When the lower (page_size + 8) bits of the 1394 serial bus address change to all "1", since this indicates the end of one page, there is a restriction in which packets cannot be generated that perform data transmission and reception to an address that goes beyond that page boundary. The number of pages (number of elements) required for data transfer are indicated in the data_size of the normal command block ORB. When one page ends, packet transmission and reception is performed again starting at the serial bus address comprised of segment_base_hi, segment_base_lo and segment_offset of the next element.

Reference: Page table (Example: page_size = 4)



[Data Transfer Using Page Table]

The CXD3220R automatically performs Page Table data transfer for one element. Set segment_length, segment_base_hi, segment_base_lo and segment_offset written in the page_table of each element obtained by block reading to xfer_length, segment_base_hi, segment_base_lo and segment_offset of the ADP register (same register as destination OffsetHigh & destinationOffsetLow). Set the destination_ID obtained from the data_descriptor in the normal command block ORB for the Destination ID. Set d, spd, max_payload, p and page_size. ADP transfer starts when the ADPgo bit of the Command register is set to "1".

The ADP sequentially transmits and receives data while generating request packets corresponding to the data_size represented with max_payload for data of the Page Table corresponding to one element. Data corresponding to any remaining bytes is transmitted and received with a request packet. The transfer speed of request packets generated with the ADP is selected between either S100 for spd = 0 or S200 for spd = 1.

The following page indicates a summary of packetizing with respect to this mode.

Mode 2 (Page_table_present = 1, Page_size = non-zero, Use page boundary)

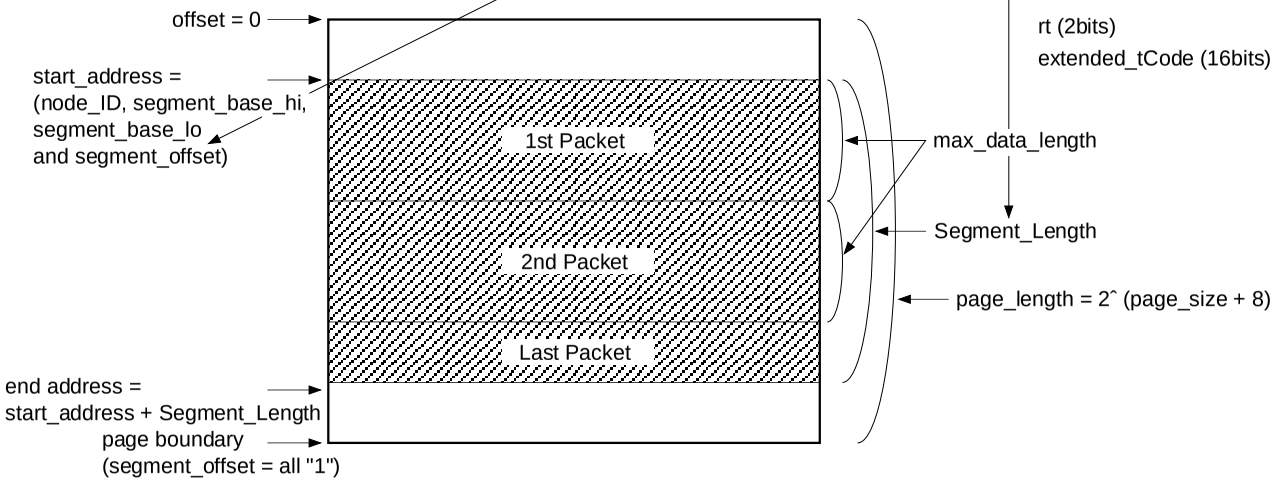
Map of ADP Control Registers

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
													imm			tLabel						(rt)		(tCode)			priority			
destinationID																segment_base_High															
segment_base_Low																segment_offset															
				d		spd		max_ payload				P	page_ size			xfer_length (Segment_Length)															
Control																															
Status																															

ADP Registers

p (RW, 1bit)	page_table_present = 1 → mode 2
tl (RW, 6bits)	Transaction Label
destination_ID (RW, 16bits)	
segment_base Hi (RW, 16bits)	
segment_base Lo (RW, 32–16bits)	
segment_offset (RW, 32–16bits)	
d (RW, 1bit)	Direction
	0 (Target ← Initiator), 1 (Target → Initiator)
sp (RW, 3bits)	Speed. 0 (S100),1 (S200)
max_payload (RW, 4bits)	data_length (16bits) ≤ 2^ (max_payload + 2) [Byte]
page_size (RW, 3bits)	segment_length [Byte]
xfer_length (RW, 16bits)	
Control (RW, 32bits)	
Status (RO, 32bits)	
1394 Header Control by ADP	Transaction Code. tCode = 0 (Quadlet Write),
tCode (4bits)	1 (Block Write).
	4 (Quedlet Read), 5 (Block Read)
	retry code. 00 (first packet), 01 (retry)
rt (2bits)	This value must be zero
extended_tCode (16bits)	

Data Transfer



Restriction

1) Block Read or Block Write Transaction does not cross address boundary.
2) The maximum data payload of one packet is 2^ (max_payload + 2).

Transaction Control

In the case of data transmission (d = 1) when the ADP has been started, a Quadlet Write request packet (see section 6-5-1-2) and Block Write request packet (see section 6-5-1-3), containing the address and Data Length generated by the ADP, is automatically generated and sent to the initiator.
The initiator either returns Ack_complete for the Ack code or returns Ack_pending for the Ack code, after which it sends back a Write response packet.
In the case of Ack_busy_*, retry is performed according to the value of BUSY_TIMEOUT field of the register. When a response packet is not returned, a timeout is detected according to the value of the SPLIT_TIMEOUT field of the register.

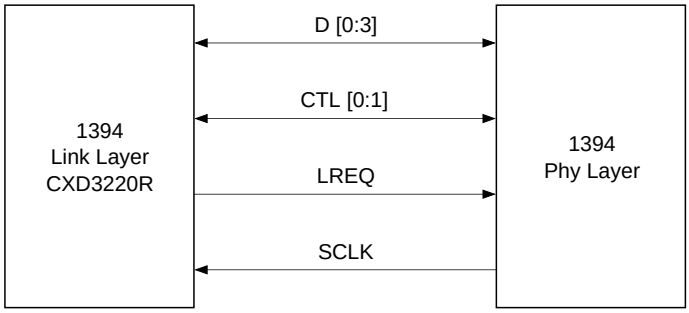
In the case of data reception (d = 0), a Quadlet Read request packet (see section 6-5-1-1), containing the address and Data Length generated by the ADP, and Block Read request packet (see section 6-5-1-2) are automatically generated and sent to the initiator.
The initiator returns Ack_pending for the Ack code and then sends back a Read response packet that contains the data of the designated address.
In the case of Ack_busy_*, retry is performed according to the value of BUSY_TIMEOUT field of the register. When a response packet is not returned, a timeout is detected according to the value of the SPLIT_TIMEOUT field of the register.

The exchange for a single Transaction is performed in the manner described above. Data transfer of three modes is performed in accordance with SBP-2 when exchange of this Transaction is continuous.

8. Link-Phy Communication

8-1. Link-Phy Interface Specifications

The CXD3220R and Phy Layer chip communicate using the four signals shown in the block diagram below: D [0:3], CTL [0:1], LREQ and SCLK.



The roles of the signals are as follows.

D [0:3]	in/out	Bidirectional data line. D [0:1] and D [0:3] are used for 100Mbps and 200Mbps, respectively.
CTL [0:1]	in/out	Bidirectional control line.
LREQ	out	Request signal line from the CXD3220R to Phy chip. Used for bus access and Phy register Read/Write requests.
SYSCLK	in	System clock (49.12MHz) supplied from Phy to the CXD3220R.

The types of communication and their contents are described below.

8-2. Communication

There are four types of communication between Phy Link: request, status, transmit and receive. Except for request, all commands are initialized by the Phy chip.

8-2-1. Bus controlling

CTL [0:1] controls communication between Phy and the CXD3220R. The communication contents differ depending on if Phy or the CXD3220R is controlling.

a) Phy controlling

CTL [0:1]	Name	Description of Activity
00	Idle	Bus is idle (Default mode).
01	Status	Phy is sending status information to the CXD3220R.
10	Receive	Phy is sending a packet to the CXD3220R.
11	Transmit	Packet transmit authorized for the CXD3220R.

b) CXD3220R controlling

CTL [0:1]	Name	Description of Activity
00	Idle	The CXD3220R completed transmit.
01	Hold	The CXD3220R is holding the bus until transmit preparations are complete. Or, the CXD3220R is trying to transmit another packet without Arbitration.
10	Transmit	The CXD3220R is transmitting a packet to Phy.
11	Reserved	Not used.

8-2-2. Request

The CXD3220R always uses serial communication of LREQ to send a request to Phy when a request to the bus or access to the Phy register is required.

There are three types of request: Bus request, Read register request and Write register request. The timing chart and contents are illustrated below.

a) Bus Request (Length of Stream: 7bit)

Bit	Name	Description
0	Start Bit	Transmit start bit. Always "1".
1 to 3	Request Type	Indicates type of request. (Refer to Request Type table.)
4, 5	Request Speed	Indicates request communication speed. (Refer to Request Speed table.)
6	Stop Bit	Last transmit bit. Always "0".

b) Read-Register Request (Length of Stream: 9bit)

Bit	Name	Description
0	Start Bit	Transmit start bit. Always "1".
1 to 3	Request Type	Indicates type of request. (Refer to Request Type table.)
4 to 7	Address	Address for Phy register read.
8	Stop Bit	Last transmit bit. Always "0".

c) Write-Register Request (Length of Stream: 17bit)

Bit	Name	Description
0	Start Bit	Transmit start bit. Always "1".
1 to 3	Request Type	Indicates type of request. (Refer to Request Type table.)
4 to 7	Address	Address for Phy register write.
8 to 15	Data	Write data for Phy register specified by Address.
16	Stop Bit	Last transmit bit. Always "0".

Request Type

LREQ [1:3]	Name	Description of Activity
000	ImmReq	Immediate bus acquisition request. To output Ack for an Asynchronous packet reception, immediate bus acquisition is requested without Arbitration when Idle is detected. Used to transmit Acknowledge.
001	IsoReq	Isochronous request. Requests execution of Arbitration. Used for Isochronous transmit.
010	PriReq	Priority request. Requests execution of Arbitration after Subaction Gap, ignoring Fair protocol. Used for Cycle Master request.
011	FairReq	Fair request. Requests execution of Arbitration after Subaction Gap according to fair protocol. Used for Fair transmit.
100	RdReq	Read request. Requests return of register contents according to Status Transfer.
101	WrReq	Write request. Requests write to specified address.
110, 111	Reserved	Reserved.

Request Speed

LREQ [4:5]	Data Rate
00	100Mb/s
01	200Mb/s
10	400Mb/s
11	Reserved

8-2-2-1. Bus Request

In order to access Fair or Priority, waits at least one clock after the CXD3220R becomes idle to send the request. When the CTL pin is in received state (CTL [0:1] = 10), the CXD3220R interprets the request as being refused. It is reissued one clock after the next idle state.

In the Cycle Master node, the cycle start message is sent using Priority request. In order to request sending of Isochronous data, the CXD3220R can issue request after receiving cycle start. Phy clears the Isochronous request only after the bus is acquired successfully.

The CXD3220R must issue ImmReq while it is receiving a packet addressed to itself in order to send Acknowledge. When packet reception is completed, Phy immediately acquires the bus and gives authorization to the CXD3220R. However, if the header CRC is erroneous, the CXD3220R immediately releases the bus. The CXD3220R can not use this authorization to send other packets. In order to ensure this operation, the CXD3220R must wait 160ns after completion of packet reception when Phy uses a bus for transmitting Acknowledge. Then it releases the bus and continues with another request.

Consider a case in which two different nodes confirm that the packet sent is addressed to them (one is correct, one is wrong), and both nodes issue an Acknowledge request before CRC check. The Phy of both nodes try to capture the bus immediately after packet receive is completed. In this state, a momentary collision occurs on the local bus at some point between the two Phy that sent back Acknowledge. This can be detected by all of the Phy connected to the bus. This collision is not interpreted as bus reset, but as high impedance state. After CRC check is completed, the wrong node will withdraw its request and the high impedance state is discontinued. The expected Acknowledge is lost as a side effect of this, but is processed by the host protocol.

8-2-2-2. Read/Write Request

When the CXD3220R requests reading of a specific register's contents, Phy transmits the register contents to the CXD3220R by Status Transfer. Even if packets are received while Phy is sending status information to the CXD3220R, Phy continues processing until the register contents are transferred.

For a Write request, Phy loads the data fields into the appropriate register as soon as transmission is completed. The CXD3220R can read/write at any time.

8-2-3. Status

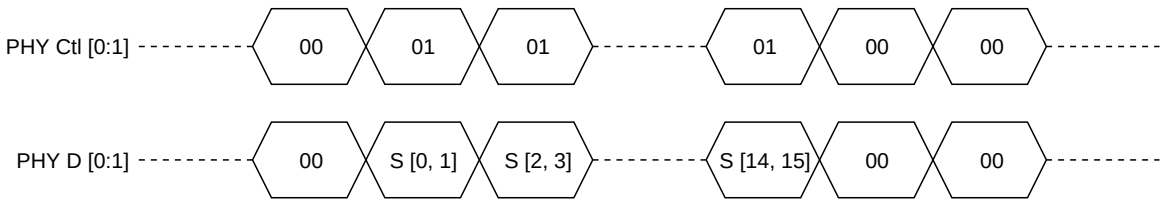
Status transmission is started by Phy when it has some data to transmit to the CXD3220R. Phy begins transmission by simultaneously setting CTL [0:1] to "01b" and the first 2 bits of Status information to D [0: 1]. Phy maintains CTL = Status during status transmission.

Phy may finish Status transmission early by setting the CTL value to some other value. This happens if a packet arrives before Status transmit is completed.

There must be at least one idle cycle in a continuous Status transmission.

Phy normally sends the first four bits of Status to the CXD3220R. These bits are the Status Flags required for the CXD3220R state machine. When transmission of a request containing a Read Request is completed, or when Phy has information to send to the CXD3220R or the Transaction Layer, Phy sends the first Status packet to the CXD3220R.

The only state in which Phy sends register contents automatically to the CXD3220R is that after completion of Self-Identification, and Physical_ID register contents containing a new node address are transmitted. The transmit timing and bit definitions are illustrated below.



Status Bit (Length of Stream: 16bit)

Bit	Name	Description
0	Arbitration Reset Gap	Indicates detection of bus idle state for Arbitration Reset Gap Time. This bit is used by the CXD3220R busy/retry state machine.
1	Subaction Gap	Indicates detection of bus idle state for Subaction Gap Time. This bit is used by the CXD3220R to detect the end of the Isochronous cycle.
2	Bus Reset	Indicates Phy in bus reset state.
3	State Time-out	Indicates that Phy state machine is stopped in a certain state for a long time. Normally used for cable topology loop detection.
4 to 7	Address	Holds the address of the register being read when Phy is trying to send register contents to the CXD3220R; for example, when responding to Read via the LReq pin.
8 to 15	Data	Holds the register being sent to the CXD3220R.

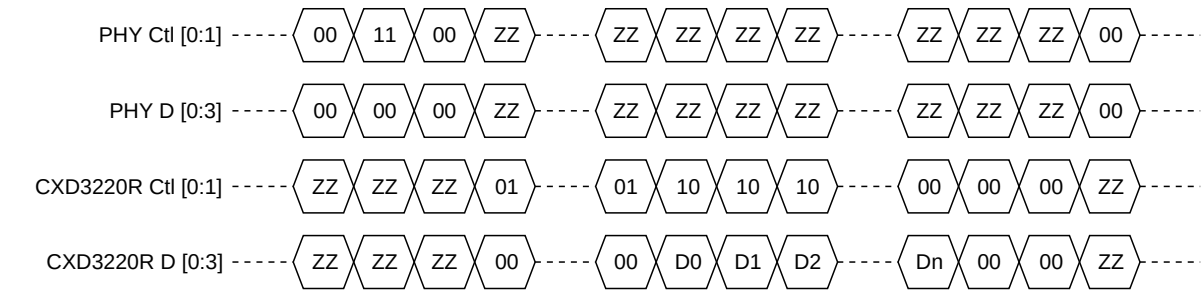
8-2-4. Transmit

When the CXD3220R requests bus access via the LReq pin, Phy performs Arbitration for bus access.
If Phy wins the Arbitration, Transmit is asserted to the Ctl pin for one SYSCLK cycle, and then Idle is asserted to the Ctl pin for one cycle to give the bus to the CXD3220R. After detecting transmitted state from Phy, the CXD3220R asserts either Hold or Transmit to the CTL pins to take over interface control. The CXD3220R asserts Hold until the data is ready, in order to keep bus initiative. During this time, Phy asserts Data-on state to the bus. When the packet is ready to transmit, the CXD3220R transmits the first bit of the packet, and at the same time asserts Transmit to the CTL pins. After sending the last bit of the packet, the CXD3220R asserts either Idle or Hold to the CTL pins for one cycle. Then it asserts Idle for one cycle before these pins become high impedance.

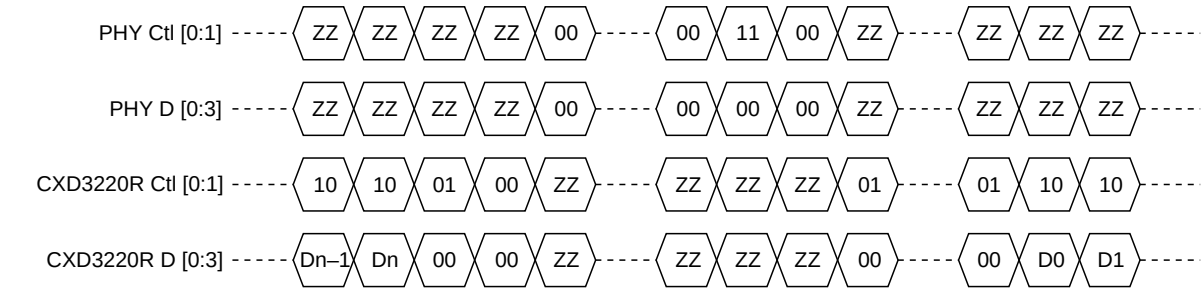
Here, when it is necessary for the CXD3220R to send another packet without releasing the bus, Hold is indicated to Phy. In response to this Hold, Phy asserts Transmit in the same way as before after waiting for the minimum required time.
This function is used after Acknowledge has been sent when the CXD3220R has attempted to send a unified response or when sending continuous Isochronous packets for one cycle. When sending a multiple number of packets during a single bus initiative, all packets must be transmitted at the same speed. Consequently, packet transmission speed is set prior to the first packet.
As described above, when the CXD3220R completes sending the last packet on the newest bus initiative, it releases the bus by asserting Idle to the CTL pins for 2 SYSCLK. When Phy detects Idle from the CXD3220R, it starts to assert Idle to CTL for one clock.

The timing chart for transmit is shown below.

Single Packet



Continued Packet

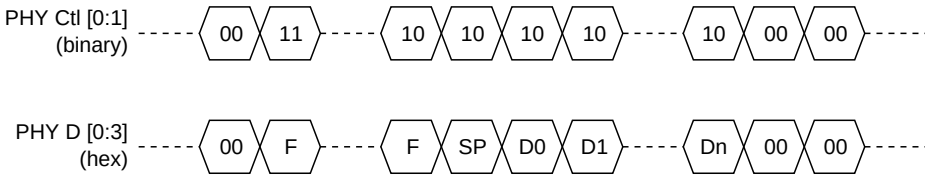


ZZ: High-impedance state, D0 to Dn: packet data

8-2-5. Receive

When data from the bus is received at Phy, it is sent from Phy to the CXD3220R in the following order.
Phy asserts Receive to the CTL pins and "all 1" to the D pin. Phy indicates the packet header by placing a Speed code on the D pin. Next it indicates the contents of the packet, and until transmission of the last symbol in the packet is completed, it holds the CTL pins at Receive. Phy indicates the end of the packet by asserting Idle to the CTL pins. The Speed code is specified by Phy-Link protocol, and does not include CRC calculation or other data protect.
Phy can identify if there is data on the bus or not without looking at the packet. This also applies if a packet is being sent at a faster speed than Phy can receive. In this case, the packet is completed by asserting Idle when the Data-on state is completed.
If Phy supports a faster transmission speed than the CXD3220R, the CXD3220R detects the Speed code and ignores the packet until it becomes Idle again.

The timing chart for reception is illustrated below.



Note) SP means Speed code.

Speed codes for receive

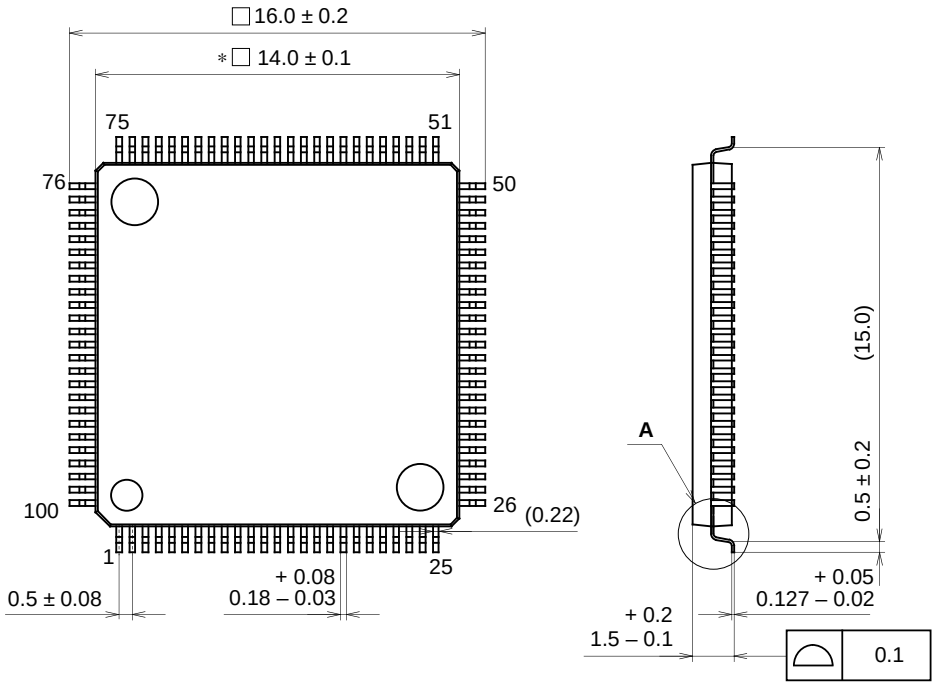
D [0:7]	Data Rate
00xxxxxx	100Mbit/s
0100xxxx	200Mbit/s
10000000	400Mbit/s

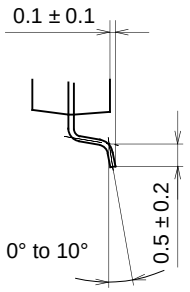
Notes)

1. "xx" means that "0" was transmitted, but it is ignored for receive.
2. This LSI supports 100Mbits/s and 200Mbits/s communications.

Package Outline Unit: mm

100PIN LQFP (PLASTIC)





DETAIL **A**

NOTE: Dimension "∗" does not include mold protrusion.

SONY CODE	LQFP-100P-L01
EIAJ CODE	∗QFP100-P-1414-A
JEDEC CODE	————

PACKAGE STRUCTURE

PACKAGE MATERIAL	EPOXY/PHENOL RESIN
LEAD TREATMENT	SOLDER PLATING
LEAD MATERIAL	42 ALLOY
PACKAGE WEIGHT	————