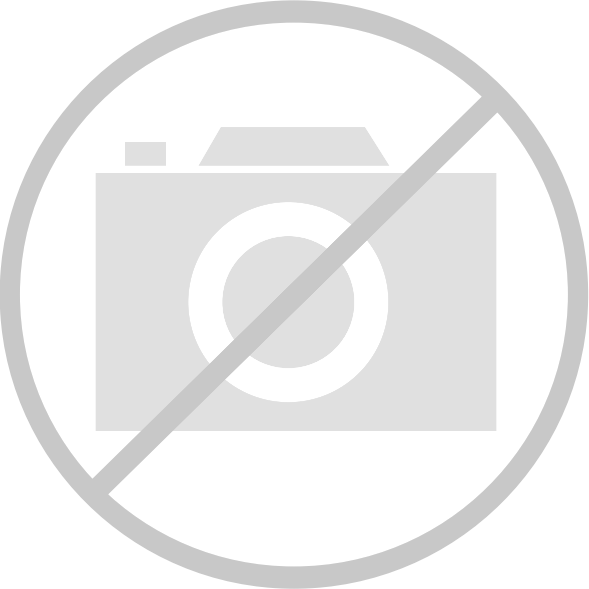
Infineon Technologies - MB86613SPFV-G-BND Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	-
Core Size	-
Speed	-
Connectivity	-
Peripherals	-
Number of I/O	-
Program Memory Size	-
Program Memory Type	-
EEPROM Size	-
RAM Size	-
Voltage - Supply (Vcc/Vdd)	-
Data Converters	-
Oscillator Type	-
Operating Temperature	-
Mounting Type	-
Package / Case	-
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/mb86613spfv-g-bnd

1. Introduction

Related Documents

This specification document was prepared based on the following documents:

1) IEEE1394- 1995 High Performance Serial Bus and P1394a draft2.0
2) 1394 Open Host Controller Interface (Open HCI) Specification Release1.1
3) PCI Local Bus Specification (Revision 2.2)
4) PCI Bus Power Management Specification (Version 1.1)

1.1. Overview

MB86613S is Fujitsu's IEEE1394- OHCI (Open Host Controller Interface) Controller LSI that is compliant with IEEE1394- 1995, P1394a and OHCI (revision 1.1, release) standard drafts. This LSI integrates both 1394 PHY and LINK layers including analog PLL, transceiver, and comparator circuits using Fujitsu's advanced full CMOS process for the cost- effective single- chip solution.

In addition to the 1394 block, the MB86613S contains various DMA engines called ContextProgram Controllers used for OHCI functions and PCI block. ContextProgram block consists of total 13 channels of independent DMA that are each dedicated to asynchronous and isochronous transmit and isochronous- asynchronous common receive operations. On- chip, 5V and 3.3V operable, PCI bus controller is compliant with PCI local bus standard (revision2.2) incorporating one 32- bit DMA controller and power management functions as specified in PCI bus power management specification (version 1.1).

For valuable host side design, this chip also incorporates serial Configuration ROM interface.

The device operates by +5V or +3.3V power supply for the PCI and DMA blocks and +3.3V for the whole 1394 block.

To provide with the cost- effective solution, the LSI is housed in a 100- pin plastic small QFP package.

1.2. Features

1) 1394 Serial Bus Controller Block:

    - Compliant with IEEE1394- 1995 and P1394a draft2.0
    - Integrates PHY and LINK layers into single- chip.
    - 1394 port number : 1 port
    - Transfer Data Rate : S100, S200, and S400
    - On- chip PLL : 400MHz for PHY and 50MHz for Link core.
    - Cycle- Master Function
    - On- chip Bus Management CSRs
    - 6- pin cable supported
    - On- chip transceiver and comparator
    - On- chip another comparator for detecting the cable power

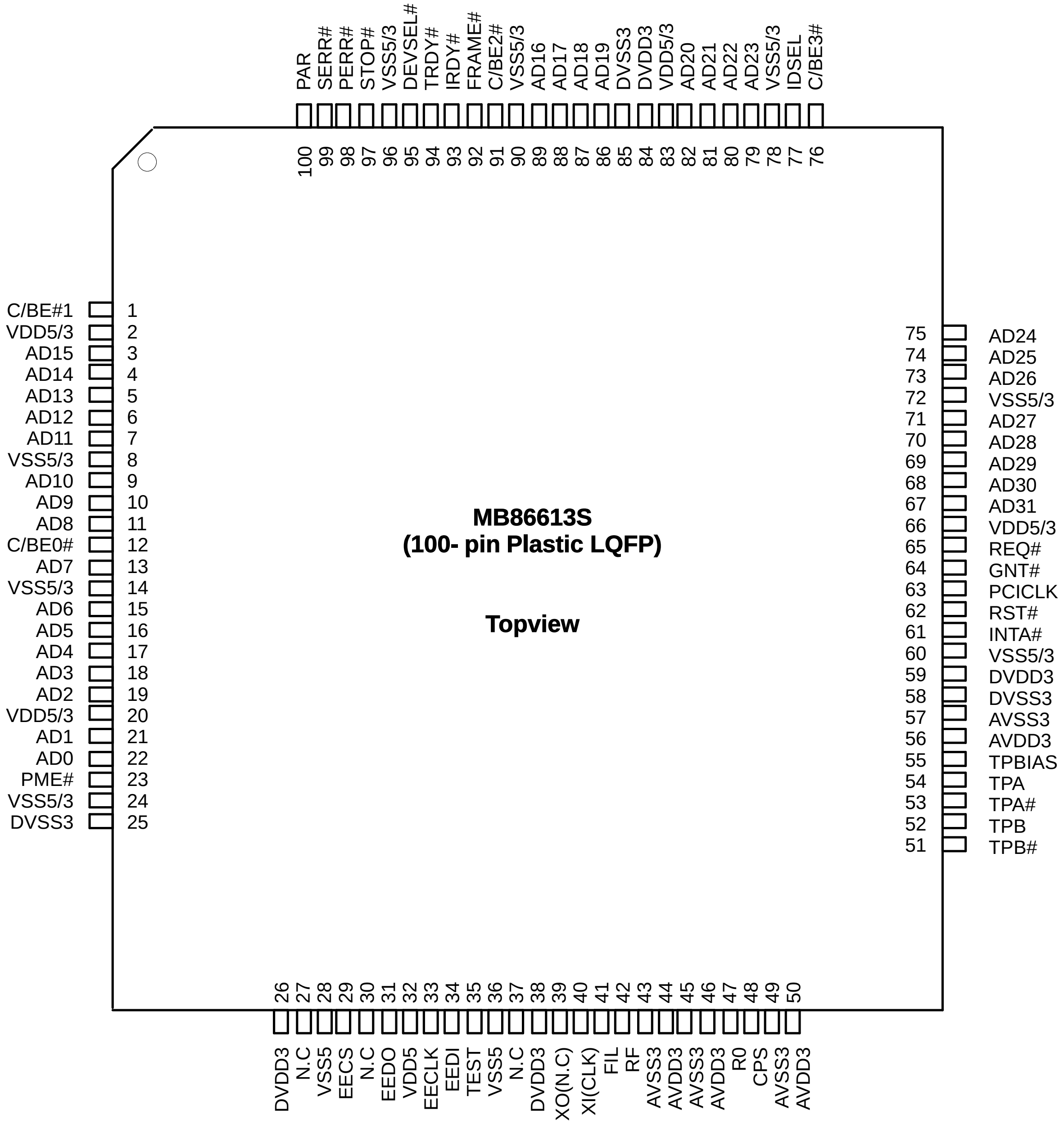
2) ContextProgram Controller Block :

    - Compliant with Open HCI standard draft (revision 1.1)
    - Total 13 independent ContextProgram Controllers:
        a) Asynchronous Transmit DMA : 2 channels for response and request each
        b) Isochronous Transmit DMA : 4 channels
        c) Receive DMA : 7 channels for Asynchronous response and request each, 4 isochronous, and 1 self- ID receive

2. Pin Functions

2.2. Pin Assignment

Figure 2.1 shows the MB86613S pin assignment.



VDD5/3 & VSS5/3 : Power Supply pins and Ground pins for PCI I/O buffer (5V or 3.3V).
VDD5 & VSS5 : Power supply pins and Ground for memory interface (5V or 3.3V).
DVDD3 & DVSS3 : Power supply pins and Ground pins for 1394, OHCI, and PCI/DMA blocks. (3.3V)
AVDD3 & AVSS3 : Power supply pins and Ground pins for PLL, 1394 transceiver and comparator. (3.3V)

Fig. 2.1 Pin Assignment

2.3. Pin Function

2.3.1. PCI Bus Interface

Notes:

I/O denotes input/output pin.
O denotes output pin.
I denotes input pin.
OD denotes open- drain output pin.

Name of pin	I/O	Function
PCICLK	I	PCI bus clock input pin (Max. 33MHz)
RST#	I	System reset input pin.
AD31 : 0	I/O	32- bit PCI Address/Data multiplexed pins.
C/BE3# : 0#	I/O	PCI Bus Command / Byte Enable multiplexed pins.
PAR	I/O	Even Parity pin for AD31:0 and C/BE3#:0#. This pin state becomes valid after 1 PCICLK.
FRAME#	I/O	Frame signal pin that indicates the PCI bus is driven by the master.
IRDY#	I/O	Data Ready signal pin for bus master device.
TRDY#	I/O	Data Ready signal pin for target device.
STOP#	I/O	Stop signal pin for the data transfer from target to master.
IDSEL	I	Chip select pin to access the configuration register.
DEVSEL#	I/O	Device select pin. While the device is a target, this pin outputs the select signal that indicates the self device is selected. While the device is a master, this pin functions as an input pin to indicate that a device on the bus is selected.
REQ#	O	Request signal output pin to the bus arbiter to request for the PCI bus use.
GNT#	I	Grant signal input pin from the bus arbiter to receive the response to the REQ# signal.
PERR#	I/O	Data Parity Error input/output pin.
SERR#	OD	Address Parity Error output pin. (Open- drain type output pin.)
INTA#	OD	Interrupt output pin. (Open- drain type output pin.)
PME#	O	PCI power management enable

2.4.2. Filter Circuit

Figure 2.3 and 2.4 shows an example of connection diagram on PLL filter circuit. A circuit where the 390Ω and a 3300pF are connected is required between the FIL pin and GND. RF pin is connected with GND through a 5.1KΩ resister. The CLK pin requires a 24.576MHz of clock module operating at +3.3V. Pin39 must be open when using a external clock module. When using the crystal oscillator, connect it and capacitors as Figure 2.4.

Those resistor and capacitor are reference values and does not guarantee stable operation on your application system.

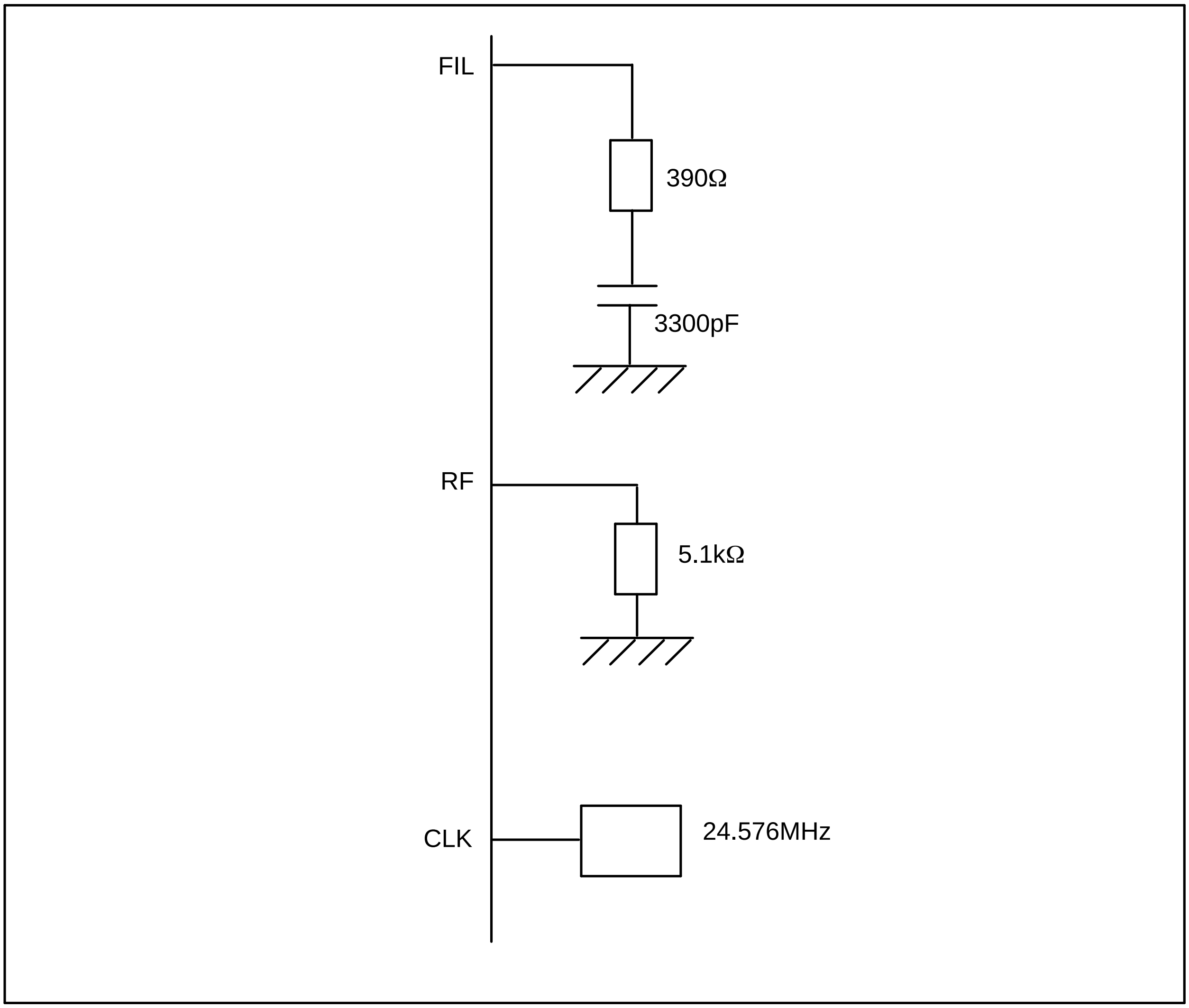
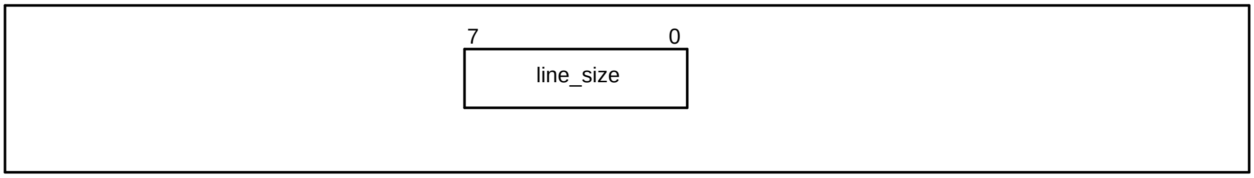


Fig. 2.3 PLL Filter Connection Example A

Bit	Field Name	rwcu	reset	description
31:24	base_class	r	0Ch	These bits indicate ”0Ch” for the serial bus controller.
23:16	sub_class	r	00h	These bits indicate ”00h” for the IEEE1394 compliant.
15:8	prog_if	r	10h	These bits indicate ”10h” for the Open HCI.

3.1.7. Cache Line Size

This register specifies the cache line size in 32- bit long- word that is guaranteed for the memory write and invalidate command.

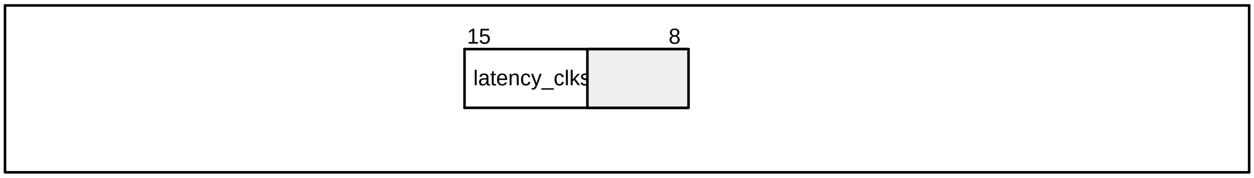


Bit	Field Name	rwcu	reset	description
7:0	line_size	rw	00h	These bits specify the cache line size with the following setting:

Value	Size
00h	Unused for memory write & invalidate command
01h - FFh	1- to 255- long word.

3.1.8. Latency Timer

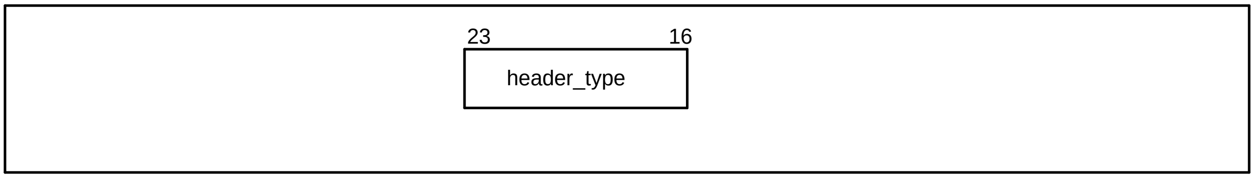
This register specifies the PCI latency timer value. The latency timer counts the time from the FRAME# asserted until the PCI bus occupied.

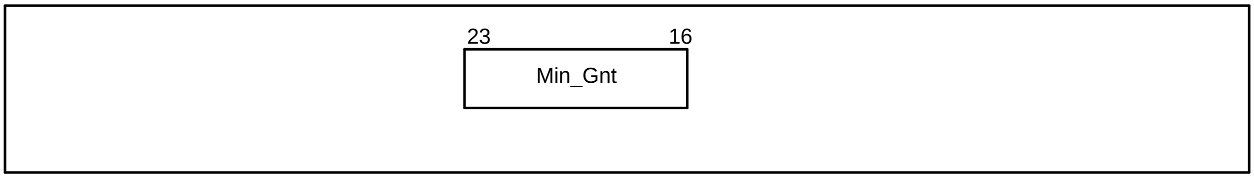


Bit	Field Name	rwcu	reset	description
15:11	latency_clks	rw	00h	These bits specify the latency timer. The unit is 8 PCI clocks.

3.1.9. Header Type

This register indicates the register configuration at addresses 10h to 3Fh in configuration space and the supported function(s). MB86613S supports the single- function so this register indicates ”00h”.

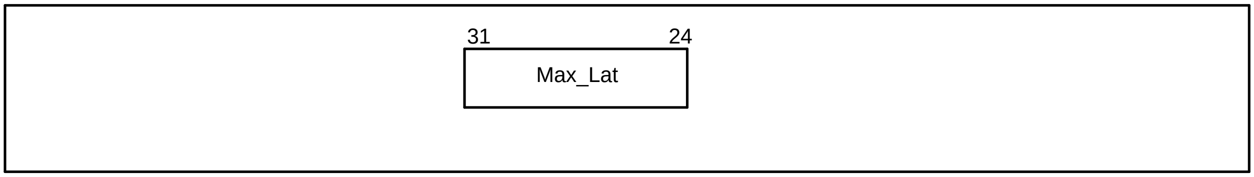




Bit	Field Name	rwcu	reset	description
23:16	Min_Gnt	r	20h	These bits indicate ”20h” (8μs).

3.1.19. MAXLAT

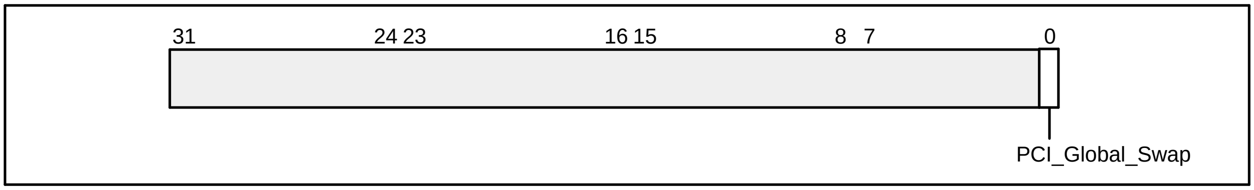
This read- only register indicates PCI bus access time required for the device. The time is 1/4 micro seconds unit.



Bit	Field Name	rwcu	reset	description
31:24	Max_Lat	r	50h	These bits indicate ”50h” (20μs).

3.1.20. PCI_HCI Control

This register specifies the byte- swap control defined in the Open HCI specification.



Bit	Field Name	rwcu	reset	description
0	PCI_Global_Swap	rw	0b	Writing 1 at this bit performs the byte- swap for the data accessed to/from PCI interface.

3.2. Open HCI Register

The addresses for the following listed Open HCI register set must be specified with the MEM Base Address register in the PCI configuration register.

r denotes the register can be read.
w denotes the register can be written.
s denotes the bit can be written (1b)
c denotes the bit can be cleared (0b)
u denotes the read value undefined depending on the MB86613S device status.

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	
IRMultiChanMaskHiSet	070h
IRMultiChanMaskHiClear	074h
IRMultiChanMaskLoSet	078h
IRMultiChanMaskLoClear	07Ch
IntEventSet	080h
IntEventClear	084h
IntMaskSet	088h
IntMaskClear	08Ch
IsoXmitIntEventSet	090h
IsoXmitIntEventClear	094h
IsoXmitIntMaskSet	098h
IsoXmitIntMaskClear	09Ch
IsoRecvIntEventSet	0A0h
IsoRecvIntEventClear	0A4h
IsoRecvIntMaskSet	0A8h
IsoRecvIntMaskClear	0ACh
InitialBandwidthAvailable	0B0h
InitialChannelsAvailableHi	0B4h
InitialChannelsAvailableLo	0B8h
▨	0BCh to 0D8h
FairnessControl	0DCh
LinkControlSet	0E0h
LinkControlClear	0E4h
NodeID	0E8h
PhyControl	0ECh
IsochronousCycleTimer	0F0h

(Continued)

▨ : denotes the reserved (unused) area.

4. Context Program Controller (CPC)

Basically Context Program Controller’s (CPC) state machine consists of Program Load, Program Analysis, and Interrupt Handle.

CPC starts the operation after the ContextControl.run bit is set by the software.   CPC, however, stops the operation when ContextControl.active and .dead bits are cleared as a result of ContextControl.run bit cleared by the software.

In Program Loading Unit (PLU), MB86613S loads the context program from the host memory in which the CommandPtr.descriptorAddress indicates and stores the program into the on- chip work RAM.

In Program Analysis Unit (PAU), MB86613S analyzes the loaded context program and moves the required transmit- packet from/to the work RAM/Host memory to/from FIFO.

In Interrupt Handling Unit (IHU), MB86613S handles/controls the event- code and interrupt as a result from the program analysis. In this case, if Z field in the last descriptor contains ”1” or more, the device stores the address set in the branchAddress field of the last descriptor in the CommandPtr.descriptorAddress field in order to store the next context program and returns to the program loading unit.

Figure 4.1 shows the state machine diagram, and Figure 4.2 shows the flag transition of ContextControl Register with the explanation as follows:

1) If the wake bit is set by software, clear the wake bit and set the active bit, then go to the program loading unit. When starting the CPC with the run bit, set the active bit and go to the program loading unit.

2) When the program analysis is completed, go to the interrupt handling unit. If an error occurred in the IHU, set the dead flag and clear the active bit, then wait until the dead flag is cleared by software.

3) If the next context program to be processed is not prepared yet, clear the active bit and wait until the wake bit is set by software; i.e. until the next context program to be processed is prepared.

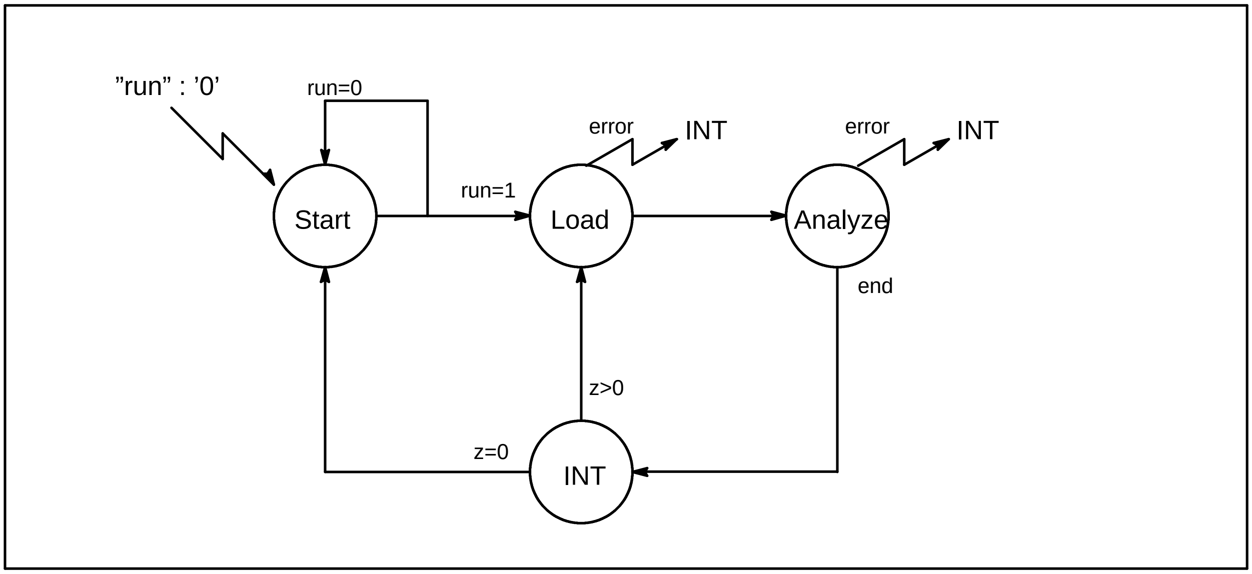


Fig. 4.1 Block Diagram of CPC State Machine

2) response packet Transmit:

Time specified in the timeStamp section in the OUTPUT_MORE_Immediate or OUTPUT_LAST_Immediate command is compared with the cycle- timer value to validate the evt_timeout.

3) ping packet Transmit:

When the ’ p’ flag in the OUTPUT_LAST or OUTPUT_LAST_Immediate command is set, time after the packet is transmitted until the acknowledge or self- ID packet is received is stored in the timeStamp section by the cycle- timer value.

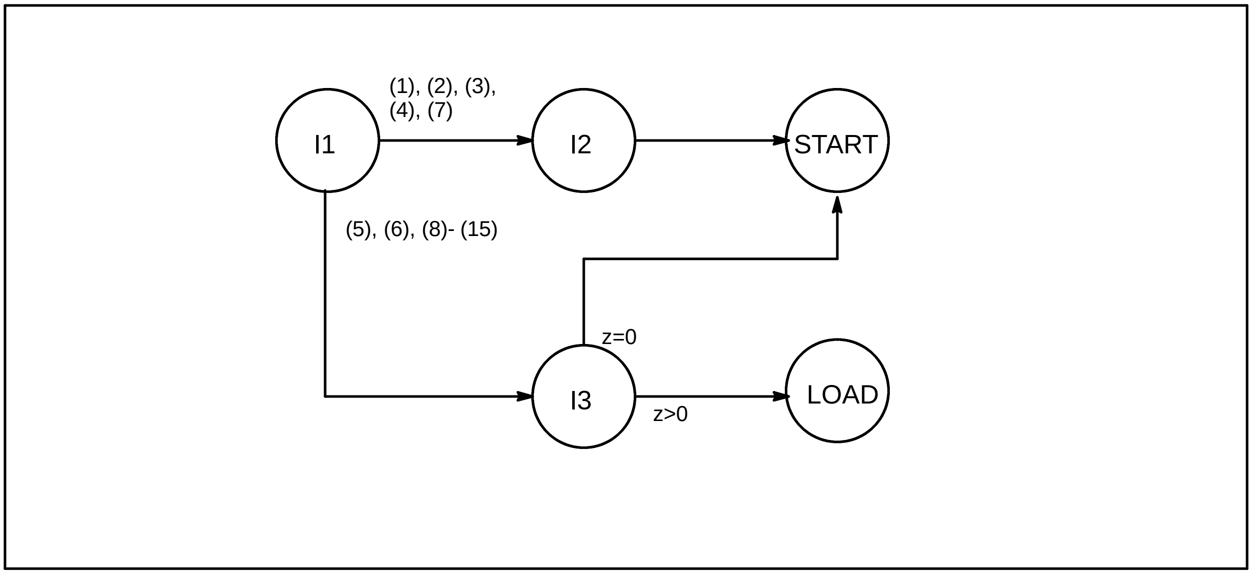


Fig. 4.4 State Machine of Interrupt Handle for Asynchronous Packet Transmit

4.1.3. Packet Format

Figures 4- 5 to 4- 15 show various packets’ format when stored in the FIFO from the host memory or work RAM. Link- Tx block converts format of these packets shown from Open HCI to 1394 and transmits the packets onto the 1394 bus.

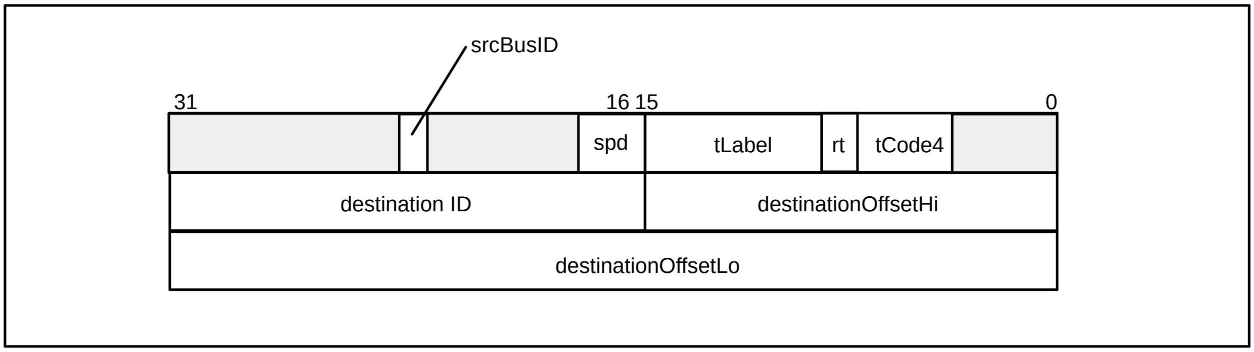


Fig. 4.5 Quadlet Read Request Transmit Packet

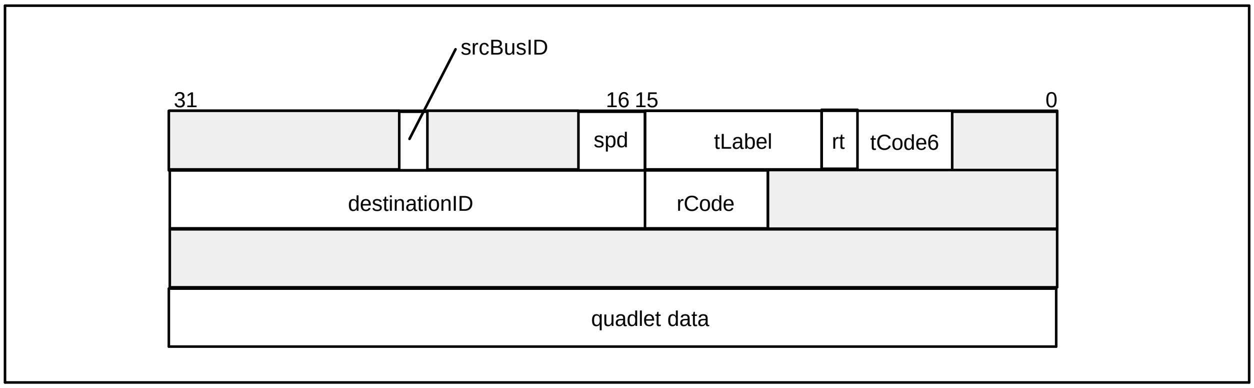


Fig. 4.12 Quadlet Read Response Transmit Packet

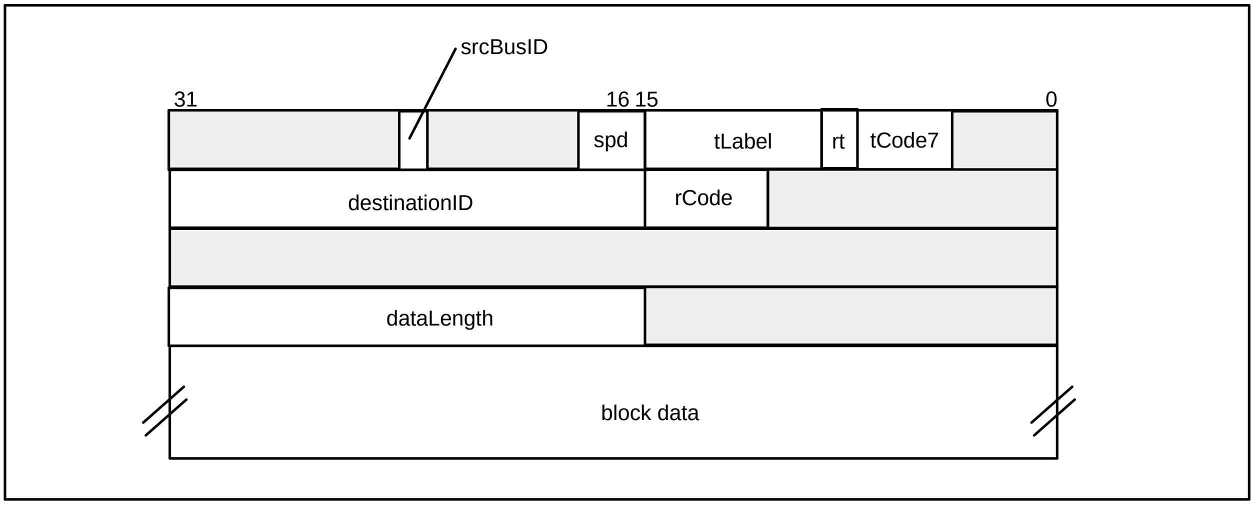


Fig. 4.13 Block Read Response Transmit Packet

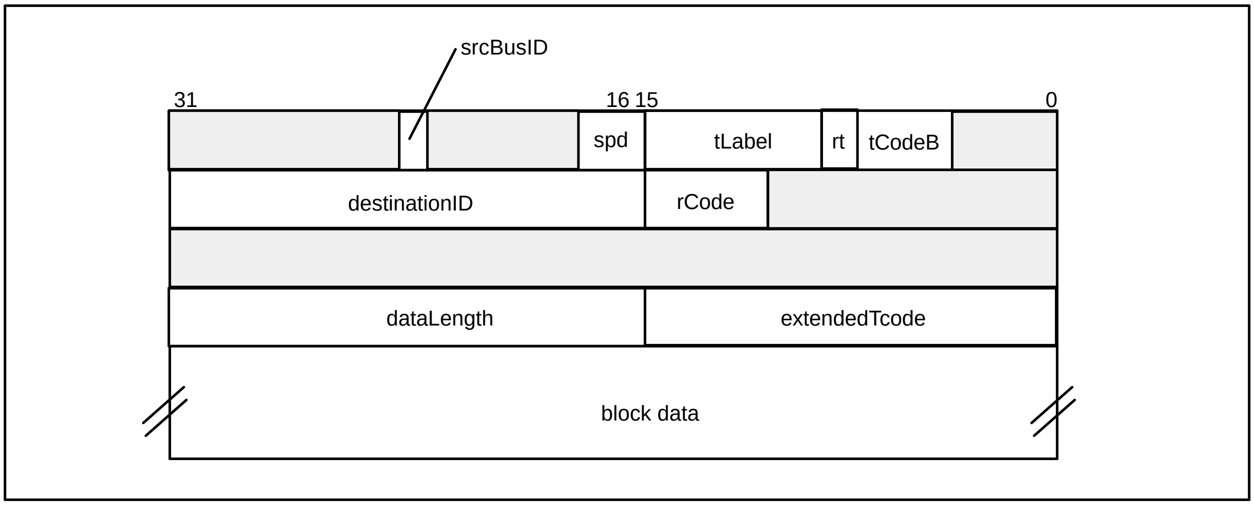


Fig. 4.14 Lock Response Transmit Packet

(7) ack_type_error : 1Eh

when the block- write- request packet that exceeds the maximum payload count is received. or, when the tcode in the received packet is undefined.

(8) ack_pending : 12h

when a packet is received normally and the response packet needs to be transmitted.

(9) ack_complete : 11h

when a packet is completely received.

The following describes and shows the state machine of interrupt handling:

I1 . . . . Set the interrupt event code in the ContextControl.eventcode field and then further set the interrupt code and remained byte count of the host memory in the xferStatus and resCount fields in the last descriptor. After completing these processes, go to the procedure I3 if the context program has been processed correctly. If it could not be processed correctly then go to the procedure I2.

I2 . . . . Set the IntEvent.unrecoverableError bit if the ' i ' flag in the last descriptor indicates '11b'. Then store the start address of host memory where the error descriptor is contained in the CommandPtr.descriptorAddress field, and return to Start.

I3 . . . . If the ' i ' flag in the last descriptor indicates '11b', set the IntEvent.ARRQ or ARRS bit. After that, if "0" is indicated in the Z field in the last descriptor, return to the START. If "1" is indicated, then store the address set in the branchAddress field in the CommandPtr.descriptorAddress of the last descriptor. After the process completed, go to the program loading process.

The MB86613S device has a function that automatically processes for error when it occurs. This is called "back- out" process. Back- out process removes packets from the host memory when an error is found on the packet such as data length error and data CRC error. This function allows the device to store only the correct packets in the memory, that results in having the proper acknowledge such as ack_complete and ack_pending The following describes some cases where this back- out process is taken:

1) FIFO was full while receiving packets.
2) data_length_error or data_CRC_error occurred on receive packets.
3) Bus reset occurred while receiving packets.

Also, in case where the above error occurs at the packets that are stored over two host memory(s), like the packet- 2 shown in Figure 4.18, the descriptor control is returned to the descriptor- 1's.

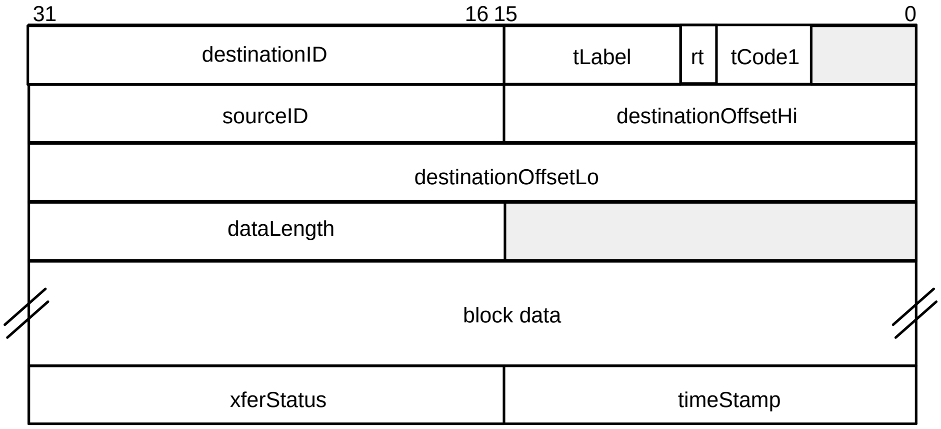


Fig. 4.22 Block Write Request Receive Packet

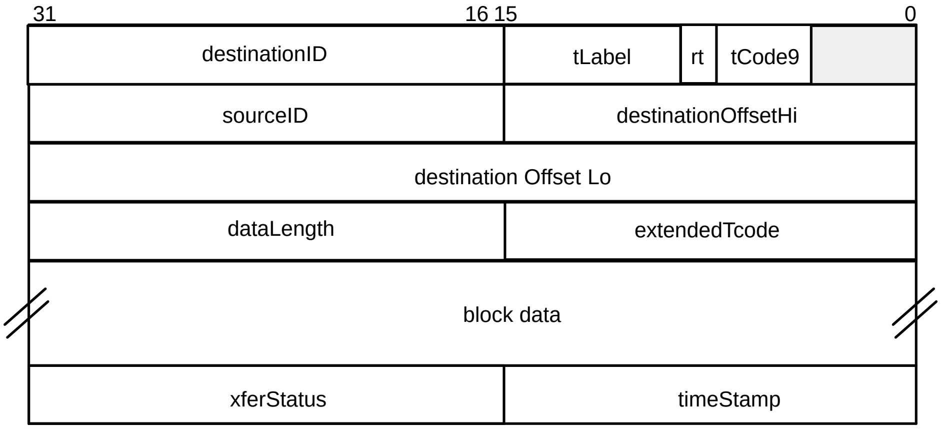


Fig. 4.23 Lock Request Receive Packet

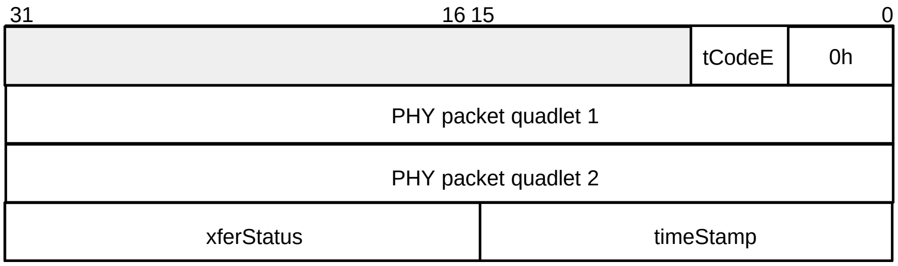


Fig. 4.24 PHY Receive Packet

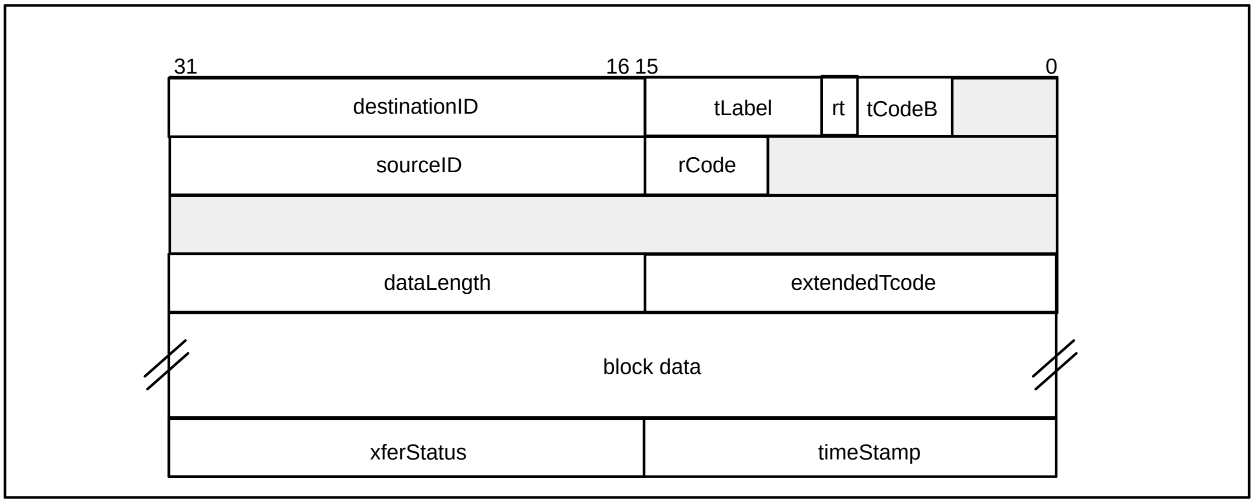


Fig. 4.28 Lock Response Receive Packet

Bit	Field Name	description
16- bit	destinationID	Indicates the bus number in the upper 10- bit field and set the node number of the destination node in the lower 6- bit field.
6- bit	tLabel	Indicates the tLabel (transaction label).
2- bit	rt	Indicates the rt (retry) code as follows: 00b : retry1 01b : retryX 10b : retryA 11b : retryB
4- bit	tCode	Indicates the tCode (transaction code).
16- bit	sourceID	Indicates the bus number in the upper 10- bit field and set the node number of the source node in the lower 6- bit field.
48- bit	destinationOffset	Indicates the address to transmit the packet.
4- bit	rCode	Indicates the rcode (response code).
16- bit	dataLength	Indicates the byte count of block data section for the receive packet.
16- bit	extendedTcode	Indicates the code that clarifies the lock request/response packet format.
16- bit	xferStatus	Indicates the result of packet receive with ARcontextControl register format.
16- bit	timeStamp	Indicates the time when the packet was received with the IsoCycleTimer.cycleCount value and cycle timer value in the lower 3- bit of cycleSecond field.

4.3. Isochronous Transmit

4.3.1. Program Analysis

Context program is analyzed by the following procedures:

P1 . . . . If the first descriptor is STORE_VALUE command, store the 32- bit data adding ”0000h” to the upper case of the value specified in the storeDoublet section into the address specified in dataAddress. If the first descriptor is OUTPUT_MORE_Immediate or _LAST_Immediate command, move the packet header stored in the lower 16- byte of descriptor from the work RAM to the IT- FIFO as specified in the reqCount.

After the processes completed, if that descriptor is STORE_VALUE command, go to the next descriptor process. if it is OUTPUT_LAST_Immediate command, go to the interrupt handling process, and if it is OUTPUT_MORE_Immediate command, go to the procedure P2.

P2 . . . . Move the block data from the host memory where the address is specified in the next descriptor’s dataAddress section to IT- FIFO.
After the processes completed, if that descriptor is OUTPUT_LAST command, go to the interrupt handling process. If it is OUTPUT_MORE command, repeat the P2.

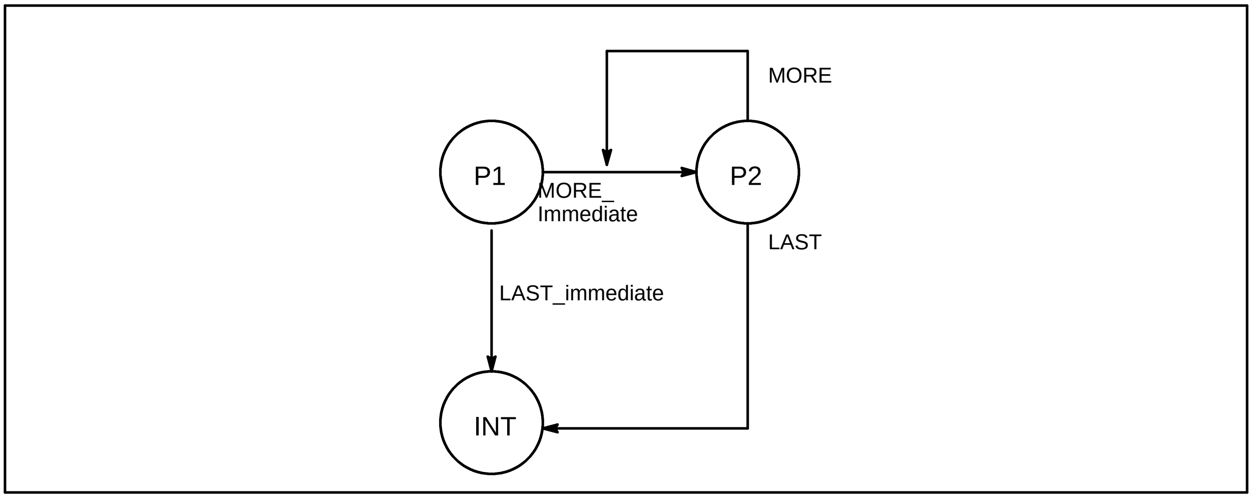


Fig. 4.29 State Machine of Program Analysis for Isochronous Packet Transmit

4.3.2. Interrupt Handle

There are a number of interrupts possibly occur in the isochronous transmit. The following lists the error name and the condition:

(1) evt_descriptor_read : 06h

when a PCI bus error occurs while the context program moves from the host memory into the work RAM.

(2) evt_unknown : 0Eh

when a context program has some problem and it cannot be processed. Or when the device stops the context program process because a long bus reset or cycle lost period is encountered.

(3) evt_data_read : 07h

when a PCI bus error occurs while the packet data moves from the host memory into the IT- FIFO.

packet, packet- 3.

IR- CRC stores the result of process in the xferStatus and resCount fields of descriptor in which the context program has completed the process. This means, the result is stored in the INPUT_LAST command in case of packet- 1 and in the INPUT_MORE command in case of packet- 2.

When a packet size is larger than the size of host memory prepared by one context program, this packet will not completely be stored in the memory and evt_long_packet is reported.

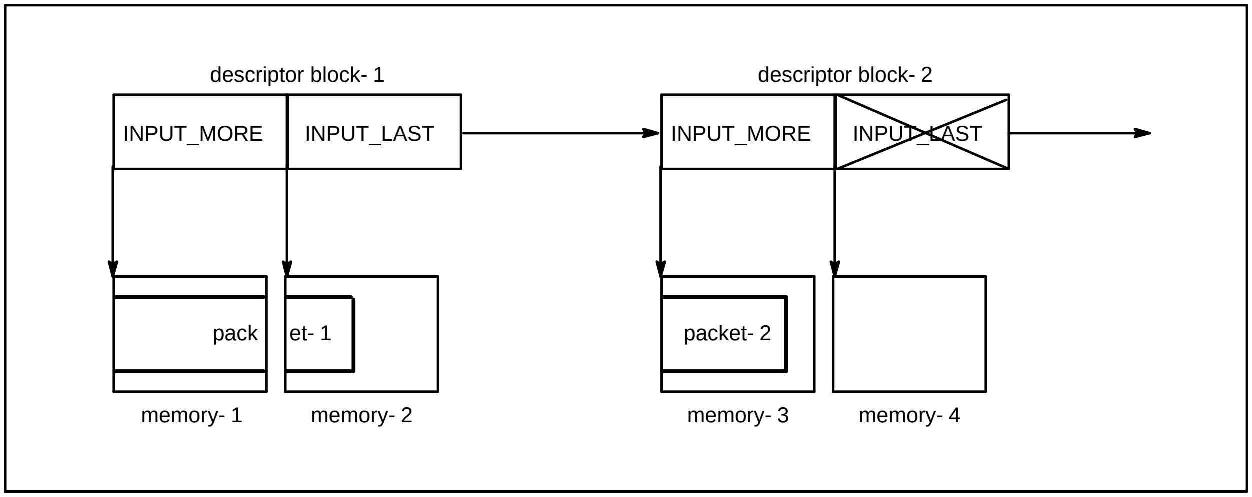


Fig. 4.36 Packet- Per- Buffer Mode (Example)

4.4.4. Packet Format

Figures 4.37 to 4.40 show the isochronous receive packet format, that is to be received and stored in the host memory. Link- Rx block converts format of these packets shown from 1394 to Open- HCI and stores it into IR- FIFO. When storing the received packet into the host memory after removing the packet header and trailer data (i.e., when IRContextControl.isoHeader bit is cleared), the alignment process is taken as described in section 7.1.

Also, like Figure 4.39, if both packer header and trailer data are stored in the memory in packet- per- buffer mode (i.e., IRContextControl.isochHeader bit is set), the alignment process is done only when the packet data and packet header are stored in different memory locations.

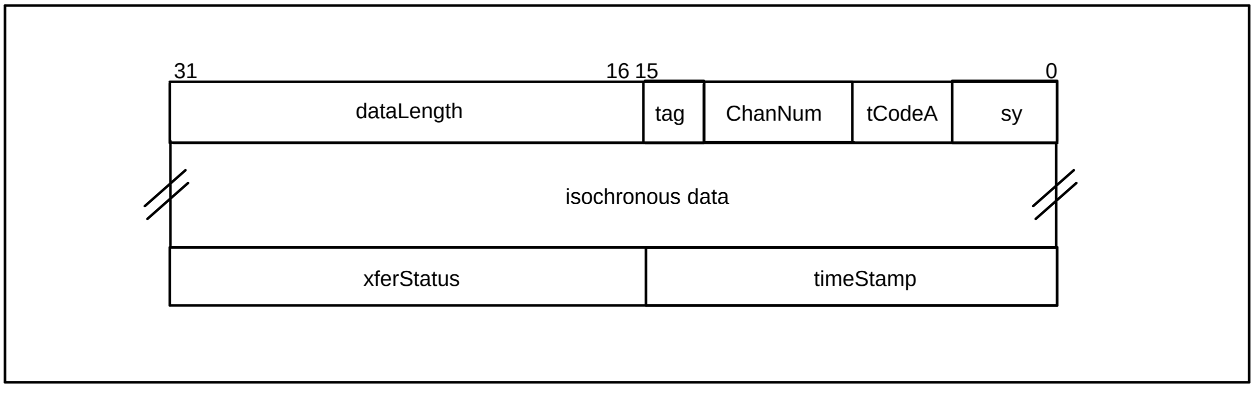


Fig. 4.37 isochronous receive packet Format (buffer- fill Mode with header/trailer)

7.3.1 How to access EEPROM

EEPROM can be accessed by using the address space of Open HCI register space field.
Data storing to EEPROM should be done in the following procedures.

1) Write ”00”b at EEPCMD.EEPCmd and ”11”b at the bit 5- 4 in EEPAddr field. Then change the write- operation to ”enable”.

2) The following three settings should be done.

a) Set ”01b” into the EEPCMD.EEPDone field.
b) Set the EEPROM address where the data is stored into the bit 5- 0 of the EEPAddr field.
c) Set the data to be written into the EEPDATA register.

3) Wait until the EEPCMD.EEPDone bit is set.

4) Write ”00”b at EEPCMD.EEPCmd field and ”00”b at the bit 5- 4 in EEPAddr field. Then change the write- operation to ”disable”.

In case EEPCMD.EEPDone bit has not set, set EEPCMD.EEPReset bit to write again.
Reading out EEPROM data can be done as follows;

1) Write ”10”b at EEPCMD.EEPCmd field and set EEPROM address where the data is stored to EEPAddr field.

2) Wait until the EEPCMD.EEPDone bit is set.

3) EEPROM data in the address set at 1) above are displayed on EEPDATA register.

Writing ”AA559966”h and ”669955AA”h continuously to KeyLocation register makes the EEPCMD and EEPDATA registers available.

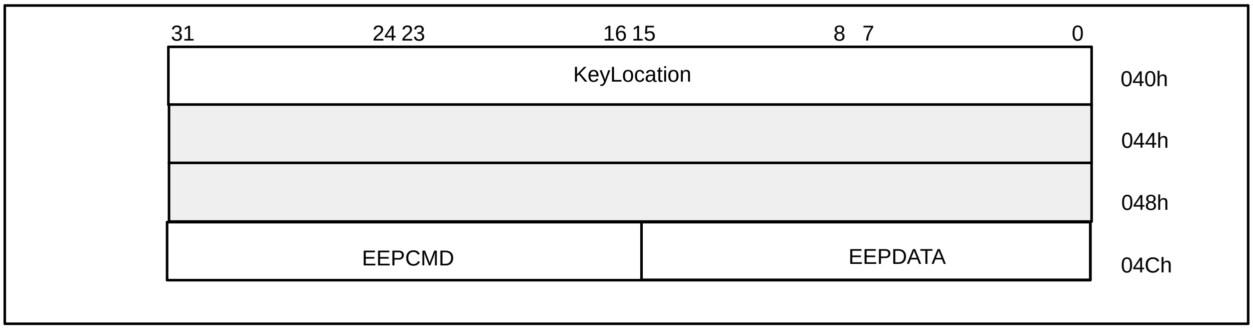


Fig. 7.5 EEPROM Register

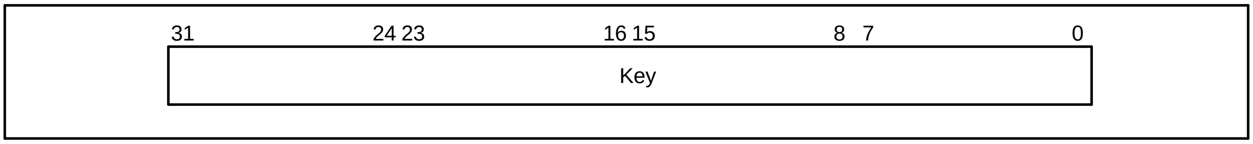


Fig. 7.6 KeyLocation Register Map

Bit size	Field Name	rwcu	reset	description
31:0	Key	w	000000h	Writing ”AA559966”h and ”669955AA”h continuously to KeyLocation register makes EEPCMD and EEPDATA registers available. 1st : ”AA559966”h 2nd :”669955AA”h

Bit size	Field Name	rwcu	reset	description
15	PME_status	rcu	0b	This bit indicates ”1” when the PME# is asserted.
14:13	data_scale	rw	00b	This bit is used to set the data range indicated in the Data register. Refer to the 7.4.6 ”Data Register” for the details.
12:9	data_select	rw	0h	This bit is used to change the windows in the Data register. Refer to the 7.4.6 ”Data Register” for the details.
8	PME_en	rw	undefined	Setting ”1” at this bit enables the PME#. Setting ”0” at this bit does not assert the PME#.
1:0	PowerStatus	rw	00b	This bit sets the state of the power management to be performed. ”00b” : D0 ”01b” : Prohibit to set. ”10b” : D2 ”11b” : D3hot

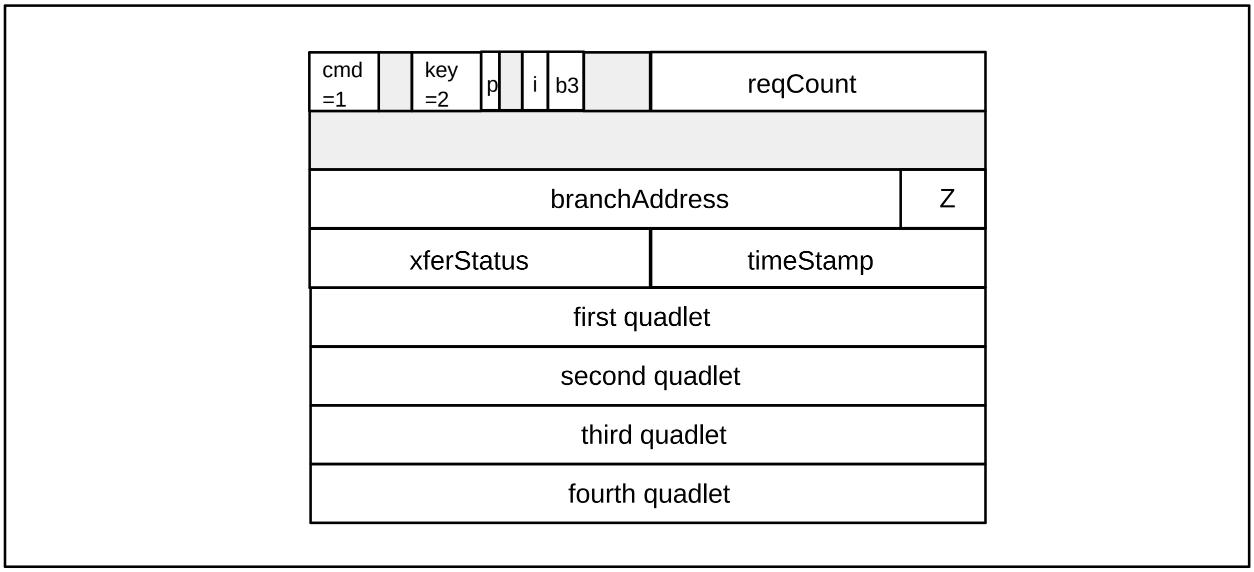


Fig. 8.6 OUTPUT_LAST_Immediate descriptor Format

Bit size	Field Name	description
4	cmd	”0h” means ”MORE” command. ”1h” means ”LAST” command.
3	key	”0h” means ”not immediate” command. ”2h” means ”immediate” command.
1	p	”1b” at this bit means the transmit packet is ping packet. This bit is used to count the time until the selfID packet or acknowledge, or the time until the subaction gap is detected after transmitting a packet. The time is counted in 50MHz and the counted time is indicated in timeStamp field.
2	i	This field is valid only for OUTPUT_LAST and OUTPUT_LAST__Immediate commands. It controls the interrupt event (IntEvent.reqTxComplete or .respTxComplete bit) reported after the descriptor is processed. ”11b” makes to report the interrupt. ”01b” enables the interrupt report only when the acknowledge except for ack_ complete and ack_pending is received. ”00b” does not report the interrupt. ”10b” is an unspecified code.
2	b	Set ”00b” for MORE command. Set ”11b” for LAST command.
16	reqCount	For OUTPUT_MORE and OUTPUT_LAST commands: Set the data byte count to be stored from a host memory into the AT- FIFO. For OUTPUT_MORE_Immediate and OUTPUT_LAST_Immediate commands: Set the byte count of packet header.
32	dataAddress	Set the start address of host memory where data to be stored in AT- FIFO are in.

and dataAddress field contains ”10000000h” which is the start address of host memory where the packet is stored for descriptor- 1 and ”20000000h” for descriptor- 2. Also, the branchAddress field for descriptor- 2 contains ”90000000h” which is the start address of host memory where the context program- 2 to control the host memory for storing the next packet is stored and the Z field contains ”2h” which is a number of descriptor in context program- 2.

The ” i ” field of the INPUT_LAST command is being set to ”11b” so that an interrupt event is reported after completion of isochronous packet receive process.

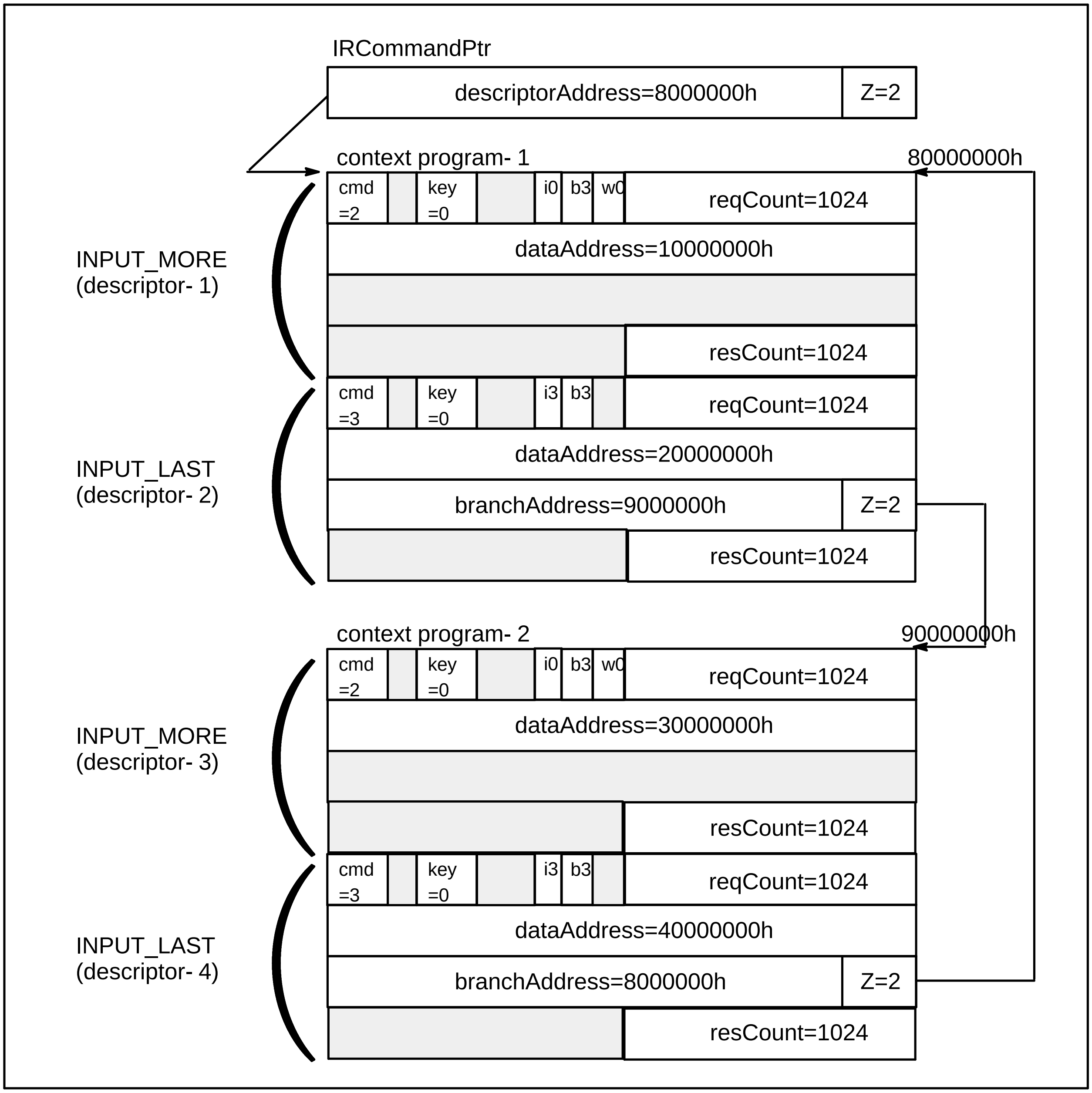


Fig. 8.18 Example of context program Format for isochronous packet Receive

8.4.2. Descriptor

This section describes the format of descriptor processed by IR- CPC. The descriptor(s) that IR- CPC handles are INPUT_MORE command and INPUT_LAST command.

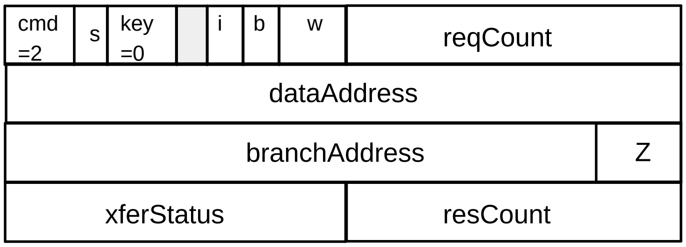


Fig. 8.19 INPUT_MORE descriptor Format

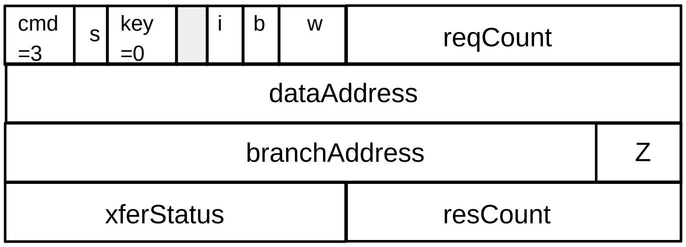


Fig. 8.20 INPUT_LAST descriptor Format

Bit size	Field Name	description
4	cmd	”2h” means ”INPUT_MORE” command. ”3h” means ”INPUT_LAST” command.
1	s	This flag is valid only for packet- per- buffer mode. ”1b” stores the result of execution in the xferStatus and resCount fields. ”0b” does not store the result of execution in the xferStatus and resCount fields.
3	key	Always set ”0h”.
2	i	This field controls the interrupt event (IntEvent.IsochRx bit) reported after the descriptor is processed. ”11b” makes to report the interrupt. ”00b” does not report the interrupt. ”10b” and ”01b” are unspecified codes. In packet- per- buffer mode, only the ” i ” field of INPUT_LAST command is valid.
2	b	For buffer- fill mode: Always set ”11b”. For packet- per- buffer mode : Set ”00b” for INPUT_MORE command. Set ”11b” for INPUT_LAST command.