FUJITSU

# MB86614A

# IEEE 1394 SERIAL BUS CONTROLLER (400Mbps)

# SPECIFICA TION

**Provisional**

Edition 3.1

## 1.1. Overview

This section will explain the outline of the MB86614A.

Fujitsu MB86614 is a high performance 1394 serial bus controller device conforming to IEEE 1394 standard (IEEE std 1394-1995), and a part of P1394a draft. It has two 1394 interface ports with differential transceiver and comparator for the network under 1394 cable circumstance. The 1394 data rate achieves maximum 400Mbps.

The MB86614 device integrates both physical-layer block and link-layer block into a single-chip to realize the down-sizing and low power consumption for 1394 serial bus applications.

The device has a 16-bit Asynchronous data port, can automatically separates the header section from the data section and packetizes it for the continuity of data transfer.

The device also has the Write request/Read request Chain functions to improve the performance of the system.

The MB86614A has a Cycle timer resister in it to realize the Cycle master function.

The device can run as a repeater node with cable power supply when the system power is turned off.

## 1.2. Features

This section will explain the features of the MB86614A.

- Complies with IEEE 1394 high-performance serial bus standard (IEEE std 1394-1995) and a part of P1394a draft
  Automatic generation and correspondence of Short Bus reset
  Automatic response at Ping packet transmission/receive
- Integrates Physical layer and Link layer into a single-chip
- 2 cable ports
- 100/200/400Mbps data transfer speed
- +3.3V Single Power Supply (5V I/F for digital input)
- Internal clock generation by on-chip PLL
- Low power consumption and forced sleep state with instruction from MPU
- Automatic separation of received-header from data and their packetization for transmission
- Cycle master function
- 32-bit CRC generation & check
- General-purpose system interface ports (16-bit MPU/DMA common bus interface)
- **Write request/Readrequest Chain transfer functions**
- Supports 6-pin cable
- Package: 80-Pin Plastic LQFP, 112-pin plastic Fine pitch BGA (FBGA)

## 1.3. Block Diagram

This section will show the block diagram of the MB86614A.

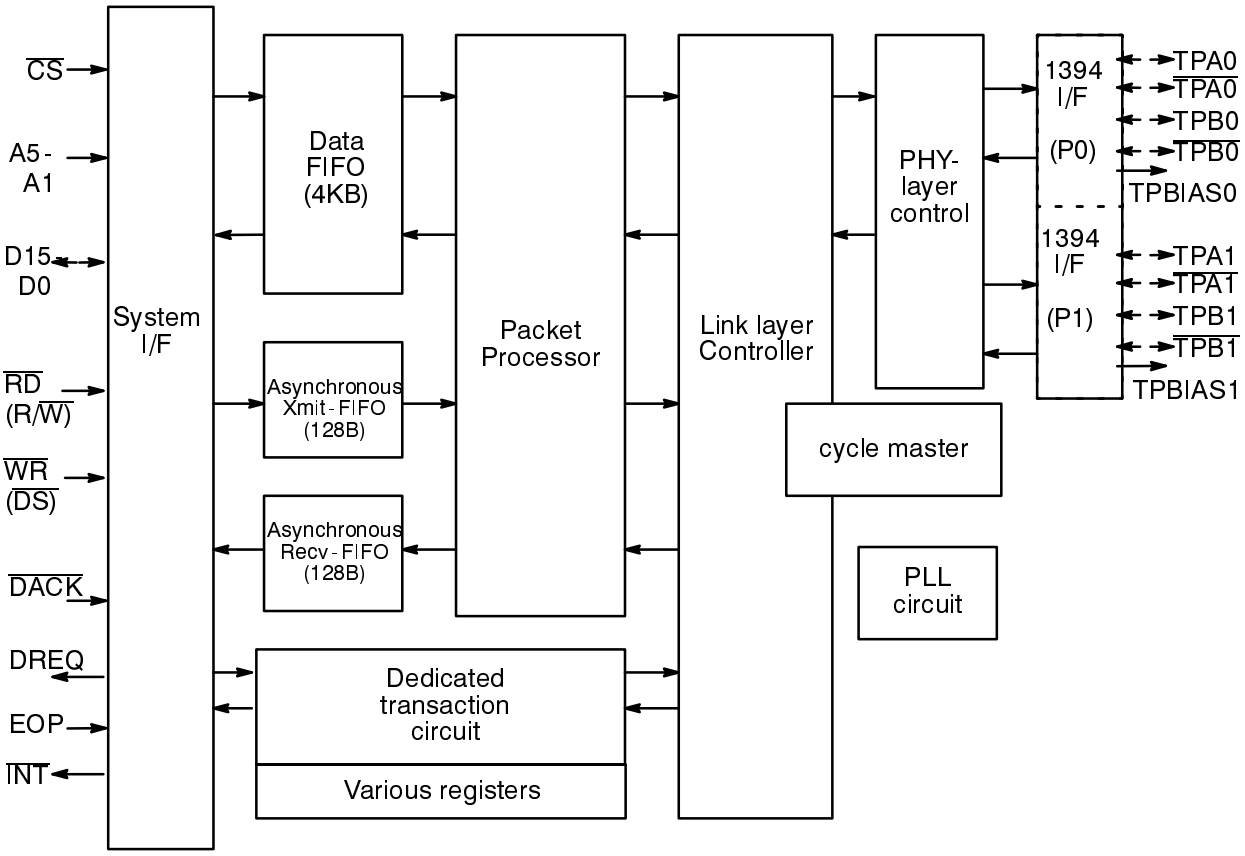



**Fig. 1.1 Block Diagram**

## 1.4. Outline of Each Block Function

<Physical layer>

- Supports asynchronous transfer under IEEE 1394 cable circumstance
- Maximum data transfer speed is 393.216Mbps.
- Two ports for analog transceiver/receiver, with bus status monitor, initialization, speed signaling, arbitration and encode/decode functions for data transmission/reception.

<Link layer>

- Controls generation and transfer of standard packet in compliance with IEEE 1394
- Generates and checks 32-bits CRC for data and header
- Incorporates 32-bits cycle timer register and cycle master function

<Transmission/Reception FIFO>

- Incorporates a 4KB Transmit/Receive - FIFO (for asynchronous transmit, bank switching 1, 2, or 4 available) for asynchronous data.
- Incorporates two 128B FIFOs for independent asynchronous transmit and receive.

<Packet Process>

- For transmission, it packetizes the header, data, and CRC sections. CRC is automatically generated and added.
- For reception, it isolates the bus packet into the header and data sections, and discards the CRC section.

<Various Registers>

- This section is composed of the LSI control registers, 1394 parameter setting registers, and so on.

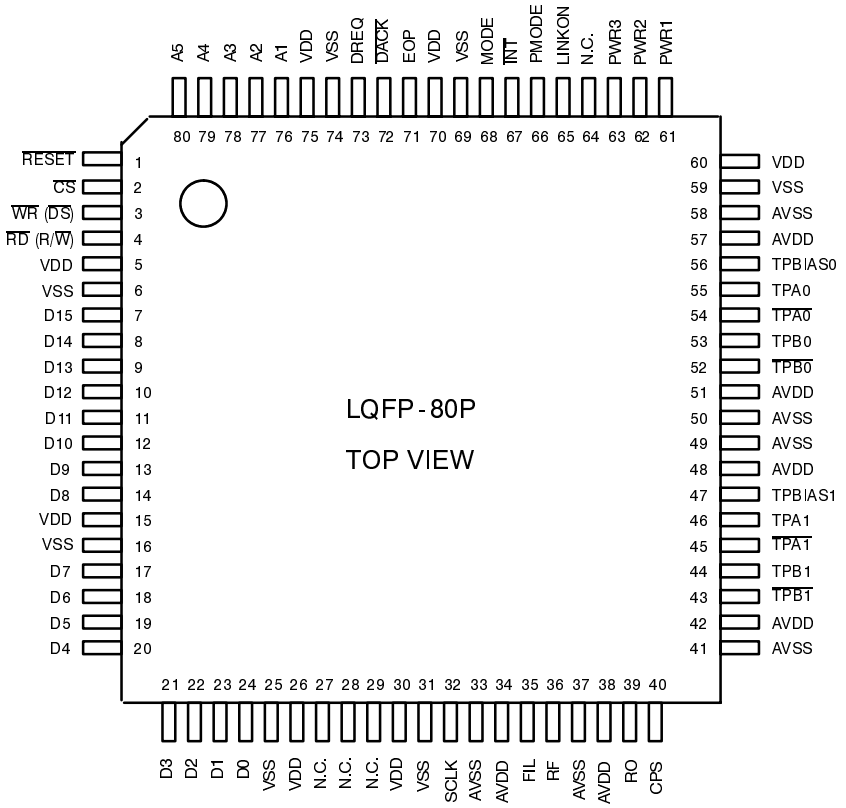
<PLL circuit>

- Generates internal operation clock of 49.152MHz and transfer clock of 393.216MHz. The reference clock applied is 24.576MHz.

### 1.5. Pin Assignment

The figure below shows the pin assignment of MB86614A for 112-Pin FBGA-112.

**MB86614 Pin Assignment - 80-Pin LQFP**



## MB86614 Pin Assignment - 112-Pin FBGA

| A | B | C | D | E | F | G | H | J | K | L | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| N.C. | N.C. | N.C. | N.C. | TPA0 | $\overline{TPB0}$ | $AV_{DD}$ | TPB1 | N.C. | N.C. | N.C. | 11 |
| N.C. | PWR1 | $V_{DD}$ | $V_{SS}$ | TPBIAS0 | TPB0 | $AV_{SS}$ | $\overline{TPA1}$ | $AV_{DD}$ | $AV_{SS}$ | N.C. | 10 |
| N.C. | PWR2 | PWR3 | N.C. | $AV_{SS}$ | $\overline{TPA0}$ | $AV_{SS}$ | TPA1 | $\overline{TPB1}$ | CPS | N.C. | 9 |
| N.C. | LINKON | N.C. | PMODE | N.C. | $AV_{DD}$ | $AV_{DD}$ | TPBIAS1 | N.C. | $AV_{DD}$ | $AV_{SS}$ | 8 |
| $\overline{INT}$ | MODE | $V_{SS}$ | $V_{DD}$ | FBGA-120 TOP VIEW | | | R0 | RF | $AV_{DD}$ | $AV_{SS}$ | 7 |
| EOP | $\overline{DACK}$ | DREQ | A1 | | | | FIL | N.C. | SCLK | $V_{SS}$ | 6 |
| $V_{SS}$ | $V_{DD}$ | A2 | A4 | | | | $V_{DD}$ | N.C. | N.C. | N.C. | 5 |
| N.C. | A3 | N.C. | $V_{SS}$ | D12 | $V_{SS}$ | D5 | $V_{DD}$ | $V_{SS}$ | D0 | N.C. | 4 |
| N.C. | A5 | $\overline{WR}$ | $V_{DD}$ | D13 | D9 | D7 | N.C. | D1 | D2 | N.C. | 3 |
| N.C. | $\overline{RESET}$ | $\overline{CS}$ | $\overline{RD}$ | D14 | D10 | $V_{DD}$ | D6 | D4 | D3 | N.C. | 2 |
| N.C. | N.C. | N.C. | N.C. | D15 | D11 | D8 | N.C. | N.C. | N.C. | N.C. | 1 |

Pin 1

## 1.6. Pin List

This section will show the pin assignment of MB86614A.

### 1.6.1. 80-Pin LQFP

The pin list for 80-Pin LQFP will be shown below.

| Pin No. | I/O | 80-MPU I/F Mode | 68-MPU I/F Mode | Pin No. | I/O | 80-MPU I/F Mode | 68-MPU I/F Mode |
|---|---|---|---|---|---|---|---|
| | | Pin Name | Pin Name | | | Pin Name | Pin Name |
| 1 | ID | $\overline{RESET}$ | $\overline{RESET}$ | 24 | ID/O | D0 | D0 |
| 2 | ID | $\overline{CS}$ | $\overline{CS}$ | 25 | - | $V_{SS}$ | $V_{SS}$ |
| 3 | ID | $\overline{WR}$ | $\overline{DS}$ | 26 | - | $V_{DD}$ | $V_{DD}$ |
| 4 | ID | $\overline{RD}$ | R/$\overline{W}$ | 27 | - | N.C. | N.C. |
| 5 | - | $V_{DD}$ | $V_{DD}$ | 28 | - | N.C. | N.C. |
| 6 | - | $V_{SS}$ | $V_{SS}$ | 29 | - | N.C. | N.C. |
| 7 | ID/O | D15 | D15 | 30 | - | $V_{DD}$ | $V_{DD}$ |
| 8 | ID/O | D14 | D14 | 31 | - | $V_{SS}$ | $V_{SS}$ |
| 9 | ID/O | D13 | D13 | 32 | ID | SCLK | SCLK |
| 10 | ID/O | D12 | D12 | 33 | - | $AV_{SS}$ | $AV_{SS}$ |
| 11 | ID/O | D11 | D11 | 34 | - | $AV_{DD}$ | $AV_{DD}$ |
| 12 | ID/O | D10 | D10 | 35 | O | FIL | FIL |
| 13 | ID/O | D9 | D9 | 36 | O | RF | RF |
| 14 | ID/O | D8 | D8 | 37 | - | $AV_{SS}$ | $AV_{SS}$ |
| 15 | - | $V_{DD}$ | $V_{DD}$ | 38 | - | $AV_{DD}$ | $AV_{DD}$ |
| 16 | - | $V_{SS}$ | $V_{SS}$ | 39 | O | R0 | R0 |
| 17 | ID/O | D7 | D7 | 40 | I | CPS | CPS |
| 18 | ID/O | D6 | D6 | 41 | - | $AV_{SS}$ | $AV_{SS}$ |
| 19 | ID/O | D5 | D5 | 42 | - | $AV_{DD}$ | $AV_{DD}$ |
| 20 | ID/O | D4 | D4 | 43 | I/O | $\overline{TPB1}$ | $\overline{TPB1}$ |
| 21 | ID/O | D3 | D3 | 44 | I/O | TPB1 | TPB1 |
| 22 | ID/O | D2 | D2 | 45 | I/O | $\overline{TPA1}$ | $\overline{TPA1}$ |
| 23 | ID/O | D1 | D1 | 46 | I/O | TPA1 | TPA1 |

(Continued)

| Pin No. | I/O | 80-MPU I/F Mode | 68-MPU I/F Mode |
|---|---|---|---|
| | | Pin Name | Pin Name |
| 47 | O | TPBIAS1 | TPBIAS1 |
| 48 | - | $AV_{DD}$ | $AV_{DD}$ |
| 49 | - | $AV_{SS}$ | $AV_{SS}$ |
| 50 | - | $AV_{SS}$ | $AV_{SS}$ |
| 51 | - | $AV_{DD}$ | $AV_{DD}$ |
| 52 | I/O | $\overline{TPB0}$ | $\overline{TPB0}$ |
| 53 | I/O | TPB0 | TPB0 |
| 54 | I/O | $\overline{TPA0}$ | $\overline{TPA0}$ |
| 55 | I/O | TPA0 | TPA0 |
| 56 | O | TPBIAS0 | TPBIAS0 |
| 57 | - | $AV_{DD}$ | $AV_{DD}$ |
| 58 | - | $AV_{SS}$ | $AV_{SS}$ |
| 59 | - | $V_{SS}$ | $V_{SS}$ |
| 60 | - | VDD | VDD |
| 61 | ID | PWR1 | PWR1 |
| 62 | ID | PWR2 | PWR2 |
| 63 | ID | PWR3 | PWR3 |
| 64 | - | N.C. | N.C. |
| 65 | O | LINKON | LINKON |
| 66 | ID | PMODE | PMODE |
| 67 | O | $\overline{INT}$ | $\overline{INT}$ |
| 68 | ID | MODE | MODE |
| 69 | - | $V_{SS}$ | $V_{SS}$ |
| 70 | - | $V_{DD}$ | $V_{DD}$ |
| 71 | ID | EOP | EOP |
| 72 | ID | $\overline{DACK}$ | $\overline{DACK}$ |
| 73 | ID | DREQ | DREQ |
| 74 | - | $V_{SS}$ | $V_{SS}$ |
| 75 | - | $V_{DD}$ | $V_{DD}$ |
| 76 | ID | A1 | A1 |
| 77 | ID | A2 | A2 |
| 78 | ID | A3 | A3 |
| 79 | ID | A4 | A4 |
| 80 | ID | A5 | A5 |

Note
I : Input
ID : Input pull down with resistance
O : Output
I/O : Input/Output (two-way)
ID/O : Input pull down with resistance /Output pull down in two-way

### 1.6.2. 112-Pin FBGA

The pin list for 112-Pin FBGA will be shown.

| Pin No. | Ball No. | I/O | 80-MPU I/F Mode | 68-MPU I/F Mode | Pin No. | Ball No. | I/O | 80-MPU I/F Mode | 68-MPU I/F Mode |
|---|---|---|---|---|---|---|---|---|---|
| | | | Pin Name | Pin Name | | | | Pin Name | Pin Name |
| 1 | A1 | - | N.C. | N.C. | 24 | G4 | ID/O | D5 | D5 |
| 2 | B1 | - | N.C. | N.C. | 25 | J1 | - | N.C. | N.C. |
| 3 | B2 | ID | $\overline{RESET}$ | $\overline{RESET}$ | 26 | J2 | ID/O | D4 | D4 |
| 4 | C1 | - | N.C. | N.C. | 27 | H3 | - | N.C. | N.C. |
| 5 | C2 | ID | $\overline{CS}$ | $\overline{CS}$ | 28 | K1 | - | N.C. | N.C. |
| 6 | C3 | ID | $\overline{WR}$ | $\overline{DS}$ | 29 | L1 | - | N.C. | N.C. |
| 7 | D1 | - | N.C. | N.C. | 30 | L2 | - | N.C. | N.C. |
| 8 | D2 | ID | $\overline{RD}$ | R/$\overline{W}$ | 31 | K2 | ID/O | D3 | D3 |
| 9 | D3 | - | $V_{DD}$ | $V_{DD}$ | 32 | L3 | - | N.C. | N.C. |
| 10 | D4 | - | $V_{SS}$ | $V_{SS}$ | 33 | K3 | ID/O | D2 | D2 |
| 11 | E1 | ID/O | D15 | D15 | 34 | J3 | ID/O | D1 | D1 |
| 12 | E2 | ID/O | D14 | D14 | 35 | L4 | - | N.C. | N.C. |
| 13 | E3 | ID/O | D13 | D13 | 36 | K4 | ID/O | D0 | D0 |
| 14 | E4 | ID/O | D12 | D12 | 37 | J4 | - | $V_{SS}$ | $V_{SS}$ |
| 15 | F1 | ID/O | D11 | D11 | 38 | H4 | - | $V_{DD}$ | $V_{DD}$ |
| 16 | F2 | ID/O | D10 | D10 | 39 | L5 | - | N.C. | N.C. |
| 17 | F3 | ID/O | D9 | D9 | 40 | K5 | - | N.C. | N.C. |
| 18 | G1 | ID/O | D8 | D8 | 41 | J5 | - | N.C. | N.C. |
| 19 | G2 | - | $V_{DD}$ | $V_{DD}$ | 42 | H5 | - | $V_{DD}$ | $V_{DD}$ |
| 20 | F4 | - | $V_{SS}$ | $V_{SS}$ | 43 | L6 | - | $V_{SS}$ | $V_{SS}$ |
| 21 | G3 | ID/O | D7 | D7 | 44 | K6 | ID | SCLK | SCLK |
| 22 | H1 | - | N.C. | N.C. | 45 | J6 | - | N.C. | N.C. |
| 23 | H2 | ID/O | D6 | D6 | 46 | L7 | - | $AV_{SS}$ | $AV_{SS}$ |

(Continued)

| Pin No. | Ball No. | I/O | 80-MPU I/F Mode | 68-MPU I/F Mode | Pin No. | Ball No. | I/O | 80-MPU I/F Mode | 68-MPU I/F Mode |
|---|---|---|---|---|---|---|---|---|---|
| | | | Pin Name | Pin Name | | | | Pin Name | Pin Name |
| 47 | K7 | - | $AV_{DD}$ | $AV_{DD}$ | 70 | G8 | - | $AV_{DD}$ | $AV_{DD}$ |
| 48 | H6 | O | FIL | FIL | 71 | F11 | I/O | $\overline{TPB0}$ | $\overline{TPB0}$ |
| 49 | J7 | O | RF | RF | 72 | F10 | I/O | TPB0 | TPB0 |
| 50 | L8 | - | $AV_{SS}$ | $AV_{SS}$ | 73 | F9 | I/O | $\overline{TPA0}$ | $\overline{TPA0}$ |
| 51 | K8 | - | $AV_{DD}$ | $AV_{DD}$ | 74 | E11 | I/O | TPA0 | TPA0 |
| 52 | H7 | O | R0 | R0 | 75 | E10 | O | TPBIAS0 | TPBIAS0 |
| 53 | L9 | - | N.C. | N.C. | 76 | F8 | - | $AV_{DD}$ | $AV_{DD}$ |
| 54 | K9 | I | CPS | CPS | 77 | E9 | - | $AV_{SS}$ | $AV_{SS}$ |
| 55 | J8 | - | N.C. | N.C. | 78 | D11 | - | N.C. | N.C. |
| 56 | L10 | - | N.C. | N.C. | 79 | D10 | - | $V_{SS}$ | $V_{SS}$ |
| 57 | L11 | - | N.C. | N.C. | 80 | E8 | - | N.C. | N.C. |
| 58 | K11 | - | N.C. | N.C. | 81 | C11 | - | N.C. | N.C. |
| 59 | K10 | - | $AV_{SS}$ | $AV_{SS}$ | 82 | C10 | - | $V_{DD}$ | $V_{DD}$ |
| 60 | J11 | - | N.C. | N.C. | 83 | D9 | - | N.C. | N.C. |
| 61 | J10 | - | $AV_{DD}$ | $AV_{DD}$ | 84 | B11 | - | N.C. | N.C. |
| 62 | J9 | I/O | $\overline{TPB1}$ | $\overline{TPB1}$ | 85 | A11 | - | N.C. | N.C. |
| 63 | H11 | I/O | TPB1 | TPB1 | 86 | A10 | - | N.C. | N.C. |
| 64 | H10 | I/O | $\overline{TPA1}$ | $\overline{TPA1}$ | 87 | B10 | ID | PWR1 | PWR1 |
| 65 | H9 | I/O | TPA1 | TPA1 | 88 | A9 | - | N.C. | N.C. |
| 66 | H8 | O | TPBIAS1 | TPBIAS1 | 89 | B9 | ID | PWR2 | PWR2 |
| 67 | G11 | - | $AV_{DD}$ | $AV_{DD}$ | 90 | C9 | ID | PWR3 | PWR3 |
| 68 | G10 | - | $AV_{SS}$ | $AV_{SS}$ | 91 | A8 | - | N.C. | N.C. |
| 69 | G9 | - | $AV_{SS}$ | $AV_{SS}$ | 92 | B8 | O | LINKON | LINKON |

(Continued)

| Pin No. | Ball No. | I/O | 80-MPU I/F Mode | 68-MPU I/F Mode |
|---|---|---|---|---|
| | | | Pin Name | Pin Name |
| 93 | C8 | - | N.C. | N.C. |
| 94 | D8 | ID | PMODE | PMODE |
| 95 | A7 | O | $\overline{INT}$ | $\overline{INT}$ |
| 96 | B7 | ID | MODE | MODE |
| 97 | C7 | - | VSS | VSS |
| 98 | D7 | - | VDD | VDD |
| 99 | A6 | ID | EOP | EOP |
| 100 | B6 | ID | $\overline{DACK}$ | $\overline{DACK}$ |
| 101 | C6 | O | DREQ | DREQ |
| 102 | A5 | - | VSS | VSS |
| 103 | B5 | - | VDD | VDD |
| 104 | D6 | ID | A1 | A1 |
| 105 | C5 | ID | A2 | A2 |
| 106 | A4 | - | N.C. | N.C. |
| 107 | B4 | ID | A3 | A3 |
| 108 | D5 | ID | A4 | A4 |
| 109 | A3 | - | N.C. | N.C. |
| 110 | B3 | ID | A5 | A5 |
| 111 | C4 | - | N.C. | N.C. |
| 112 | A2 | - | N.C. | N.C. |

Note
I : Input
ID : Input pull down with resistance
O : Output
I/O : Input/Output (two-way)
ID/O : Input pull down with resistance /Output pull down in two-way

***PRELIMINARY***

## 1.7. Pin Functions

This section will explain the function of pins on the MB86614A.

### 1.7.1. 1394 interface

The 1394 interface of the MB86614A will be explained.

| Nameof pin | I/O | Function |
|---|---|---|
| TPA0 | I/O | TPA positive signal of cable port 0 |
| $\overline{\text{TPA0}}$ | I/O | TPA negative signal of cable port 0 |
| TPB0 | I/O | TPB positive signal of cable port 0 |
| $\overline{\text{TPB0}}$ | I/O | TPB negative signal of cable port 0 |
| TPA1 | I/O | TPA positive signal of cable port 1 |
| $\overline{\text{TPA1}}$ | I/O | TPA negative signal of cable port 1 |
| TPB1 | I/O | TPB positive signal of cable port 1 |
| $\overline{\text{TPB1}}$ | I/O | TPB negative signal of cable port 1 |
| TPBIAS0 | O | Reference voltage output pin for common voltage for cable port 0 common voltage |
| TPBIAS1 | O | Reference voltage output pin for common voltage for cable port 1 common voltage |
| R0 | O | Load resistance connection pin (Ground the pin via a 5.1kΩ. resistor) |

## 1.7.2. System interface

The system interface of the MB86614A will be explained.

| Nameof pin | I/O | Function |
|---|---|---|
| $\overline{CS}$ | I | Chip Select signal Input pin for MPU to select this device as an I/O device |
| A5 to A1 | I | Address input pins for selecting internal registers. |
| D15 to D0 | I/O | 16-bit data input/output pins. (MSB:D15, LSB:D0) |
| $\overline{RD}$ (R/$\overline{W}$) | I | In 80-system mode: Input pin of read strobe signal for output data to data bus from this device<br>In 68-system mode: Input pin of control signal R/W for output and input data from/to this device |
| $\overline{WR}$ ($\overline{DS}$) | I | In 80-system mode: Input pin for Write strobe signal for input data on data bus to this device<br>In 68-system mode: Input pin of $\overline{DS}$ signal output when the data bus is enabled |
| DREQ | O | DMA Request signal output pin. This signal requests for DMA transfer between the MB86614A device and a host memory. |
| $\overline{DACK}$ | I | DMA acknowledge signal input pin. This input signal allows the DMA access with a DMAC device. |
| EOP | I | End of Process signal input pin from a DMAC device.Signal to stop the Chain transfer (continuous packet transfer).<br>This is not a pin for stopping the DMA transfer I/F.<br>The value is maintained once this pin is in active state.<br>Do not make this pin active except when the device is in the continuous transfer period. |
| $\overline{INT}$ | O | Interrupt output pin. |

## 1.7.3. Others

The system interface of the MB86614A will be explained.

| Nameof pin | I/O | Function |
|---|---|---|
| SCLK | I | System Clock input pin for internal PLL component. (24.576MHz) |
| RF | O | Ground this pin with a 5.1 kΩ resistor. |
| FIL | O | Filter circuit connection pin. |
| $\overline{\text{RESET}}$ | I | System Reset signal input pin. When no system power is present on the device, input "0" to this pin. |
| MODE | I | MPU Mode select pin. When "0" : 80-system mode, and when "1" : 68-system mode |
| PWR1 - PWR3 | I | Power_Class field setting pins for self_ID packet transmission at the cable power operation. While the device is operating with the system power, not with the cable power, Power_Class field setting is not influenced with these pin settings but the internal register settings as described in Section 2 Physical Register #4 in this document. |
| LINKON | O | Link-on Packet detection pin. At the reception of link-on packet, this pin outputs "H" signal. Then, this pin outputs "L" when PMODE="H". When this signal is not used, please leave this pin open. |
| PMODE | I | Operation mode input pin to determine the operating environment (system power supply or cable power supply [halt condition of MPU I/F]).<br>Input "0" to this pin when the system power is OFF.<br>Input "1" when the cable power is OFF or system power is ON. |
| CPS | I | Input pin to detect the availability of cable power supply.<br>Cable power supply: 8 to 30V message to decrease the voltage with an external resistance.<br>The pin determines the availability of cable power supply (1.2 V or more must be applied to the CPS pin) for operation.<br>This pin is not used when cable power supply is not used. In this case, connect it to the GND. |
| AVDD | - | Analog power supply pin |
| AVSS | - | Analog GND pin |
| VDD | - | Digital power supply pin |
| VSS | - | Digital GND pin |
| N.C. | - | No connection pins. Do not connect anything to these pins. |

## 2. Internal Registers

The following lists the MB86614A internal registers. These registers are accessible only in 16-bit.

| Adr (hex) | Address | | | | | When Write | When Read |
|---|---|---|---|---|---|---|---|
| | A5 | A4 | A3 | A2 | A1 | | |
| 00 | 0 | 0 | 0 | 0 | 0 | mode-control register | ← |
| 02 | 0 | 0 | 0 | 0 | 1 | (Reserved) | flag & status register |
| 04 | 0 | 0 | 0 | 1 | 0 | instruction-fetch register | ← |
| 06 | 0 | 0 | 0 | 1 | 1 | interrupt-mask register | interrupt-code register |
| 08 | 0 | 0 | 1 | 0 | 0 | (Reserved) | Receive acknowledge display register |
| 0A | 0 | 0 | 1 | 0 | 1 | A-Buffer Data Port (Transmit) | A-Buffer Data Port (Receive) |
| 0C | 0 | 0 | 1 | 1 | 0 | D-Buffer Data Port (Transmit) | D-Buffer Data Port (Receive) |
| 0E | 0 | 0 | 1 | 1 | 1 | (reserved) | (reserved) |
| 10 | 0 | 1 | 0 | 0 | 0 | (reserved) | (reserved) |
| 12 | 0 | 1 | 0 | 0 | 1 | Xmit-Async-des-ID setting register | (reserved) |
| 14 | 0 | 1 | 0 | 1 | 0 | Xmit-Async-PKT-param setting register | Rcv-Async-PKT-param display register |
| 16 | 0 | 1 | 0 | 1 | 1 | Xmit-Async data-length setting register | Rcv-Async data-length display register |
| 18 | 0 | 1 | 1 | 0 | 0 | Xmit-Async ex-tcode setting register | Rcv-Async ex-tcode display register |
| 1A | 0 | 1 | 1 | 0 | 1 | Xmit-Async source-bus-ID setting register | Rcv-Async source-bus-ID display register |
| 1C | 0 | 1 | 1 | 1 | 0 | Xmit-Async rcode setting register | Rcv-Async rcode display register |
| 1E | 0 | 1 | 1 | 1 | 1 | Xmit-Async des-offset setting register (upper) | Rcv-Async des-offset display reg. (upper) |
| 20 | 1 | 0 | 0 | 0 | 0 | Xmit-Async des-offset setting register (mid.) | Rcv-Async des-offset display reg. (mid.) |
| 22 | 1 | 0 | 0 | 0 | 1 | Xmit-Async des-offset setting register (lower) | Rcv-Async des-offset display reg. (lower) |
| 24 | 1 | 0 | 0 | 1 | 0 | Chain sum-data-length setting reg. (upper) | Chain remained data byte counter (upper) |
| 26 | 1 | 0 | 0 | 1 | 1 | Chain sum-data-length setting reg. (lower) | Chain remained data byte counter (lower) |
| 28 | 1 | 0 | 1 | 0 | 0 | Chain Xmit-des-ID setting register | Ping Time Monitor register |
| 2A | 1 | 0 | 1 | 0 | 1 | Chain Xmit-des-offset setting reg. (upper) | (reserved) |
| 2C | 1 | 0 | 1 | 1 | 0 | Chain Xmit-des-offset setting reg. (mid.) | (reserved) |
| 2E | 1 | 0 | 1 | 1 | 1 | Chain Xmit-des-offset setting reg. (lower) | (reserved) |

| Adr (hex) | Address | | | | | When Write | When Read |
|---|---|---|---|---|---|---|---|
| | A5 | A4 | A3 | A2 | A1 | | |
| 30 | 1 | 1 | 0 | 0 | 0 | Chain Xmit-data-length setting register | Rcv-packet transfer speed display register |
| 32 | 1 | 1 | 0 | 0 | 1 | Chain-Retry setting register | Cycle-timer-monitor display reg. (upper) |
| 34 | 1 | 1 | 0 | 1 | 0 | (reserved) | Cycle-timer-monitor display reg. (lower) |
| 36 | 1 | 1 | 0 | 1 | 1 | (reserved) | (reserved) |
| 38 | 1 | 1 | 1 | 0 | 0 | PHY/LINK register address setting register | ← |
| 3A | 1 | 1 | 1 | 0 | 1 | PHY/LINK register access port (Write) | PHY/LINK register access port (Read) |
| 3C | 1 | 1 | 1 | 1 | 0 | (reserved): access prohibited | (reserved): access prohibited |
| 3E | 1 | 1 | 1 | 1 | 1 | (reserved) : access prohibited | (reserved) : access prohibited |

Notes:

1) Address 3Ch and 3Eh are assigned for the deivce internal test purpose. Do not access these addresses.

2) MPU Access Recovery Time:

- For consecutive write opration to regular registers:
  There must be 25ns (45ns in sleep mode) or longer interval between a rising edge of first $\overline{WR}$ signal and a falling edge of next $\overline{WR}$ signal.

- For consecutive read operation to regular registers:
  There must be 25ns (45ns in sleep mode) or longer interval between a rising edge of first $\overline{RD}$ signal and a falling edge of next $\overline{RD}$ signal.

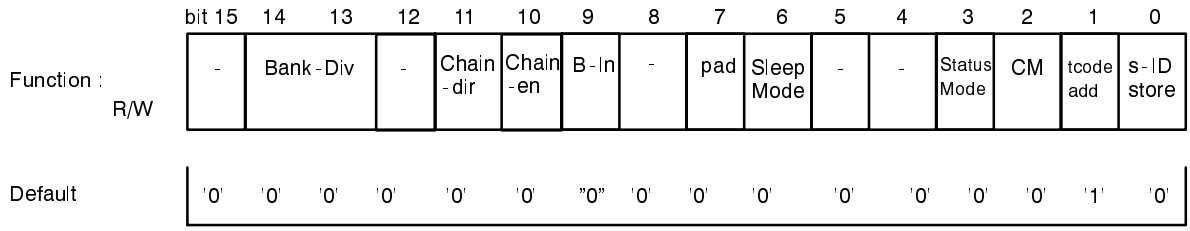
- For read-after-write operation to regular registers:
  There must be 80ns (160ns in sleep mode) or longer interval between a rising edge of $\overline{WR}$ signal and a falling edge of $\overline{RD}$ signal.

- For write operation to PHY/LINK register access port after writing to PHY/LINK register address setting register:
  There must be 100ns or longer interval between rising edges of $\overline{WR}$ signals. (This operation never happens in sleep mode because the internal PLL stops in sleep mode.)

- - For read operation from PHY/LINK register access port after writing to PHY/LINK register address setting register:
  There must be 100ns (200ns in sleep mode) or longer interval between a rising edge of $\overline{WR}$ signal and a falling edge of $\overline{RD}$ signal.

## 2.1 mode-control register (read/write): Address 00h

This register sets the operating mode and automatic operation of the MB86614A.



| | bit 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Function : R/W | - | Bank-Div | | - | Chain -dir | Chain -en | B-In | - | pad | Sleep Mode | - | - | Status Mode | CM | tcode add | s-ID store |
| Default | '0' | '0' | '0' | '0' | '0' | '0' | "0" | '0' | '0' | '0' | '0' | '0' | '0' | '0' | '1' | '0' |

[UNUSED] : R/W
Bits 15, 12, 8, 5 and 4 are unused bits. "0" is read out from these bits. For write operation of this register, always set "0" to these bits.

[ Bank-Div ] : R/W
This field specifies the bank organization of data buffer:
"00" : 1-bank
"01" : 2-bank
"1X" : 4-bank

[Chain-dir]: R/W
"0" at this bit uses Read chain for the chain transfer.
"1" at this bit uses Write chain for the chain transfer.

[ Chain_en ] : R/W
This bit enables the chain transfer.
"0" at this bit does not use the data buffer for the chain transfer.
"1" at this bit uses the data buffer for the chain transfer. This bit retains "1" even if the chain transfer is suspended due to an error detected, EOP signal asserted, or "chain suspend" instruction issued.

[ B_In ] : R/W
"0" at this bit stores a write request for data block packet in the asynchronous receive-FIFO, as well as the case of other asynchronous packet.
But "1" at this bit does not store a write request for data block packet in the asynchronous receive-FIFO. Instead, it is stored in 4KB transmit/receive-FIFO when the FIFO is used for asynchronous packet transfer.

Note: When using the chain transfer, please set "0" to this bit. Only Write request for data payload is stored because the chain is set when receiving Read response for data block.

[pad] : R/W
Padding enable bit.
"0" at this bit pads '0' data to received data that does not reach the specified data-length in header.
"1" at this bit pads '1' data to received data that does not reach the specified data-length in header.

Note: This bit implies the chip, either way, always contains a packet with correct data-length.

[Sleep mode]: R/W
When ”0” is set: The 1394 port goes to the sleep state after issuing Sleep instruction.
The value at the register, which indicates the state of connection to the 1394, is invalid.
When ”1” is set: The 1394 port does not go to the sleep state after issuing Sleep instruction.
The value at the register, which indicates the state of connection to the 1394, is valid.

Note: Register: Bias bit on Physical register #8 and #9.

[Status mode]: R/W
When ”0” is set: Bit 12 of Flag & status register displays ISO cycle.
When ”1” is set: Bit 12 of Flag & status register displays D-buffer not all empty.

[CM] : R/W
When ”0” is set, the device does not function as a ”cycle master”. When ”1” is set, the device can function as a ”cycle master”.

Note: Refer to Section 2.2 in this document for the function of Bit 12 on the flag & status register. Also the interrupt report of Chain operation depends on this bit as described in Section 3.2 in this document.

Note: The condition to be the cycle master is when this bit being contains ”1” after bus reset to set RHB bit and the device became a root node. So, even if this bit is set to ”1”, the device does not become a cycle master when it failed to become a root node. ’cmstr’ bit in flag & register indicates whether or not the device is being the cycle master.

[tcode add] : R/W
When ”0” is set: For Send-operation, it incorporates the values in tcode section for the corresponding registers (Xmit-ISO-PKT-header setting register and Xmit-ASYNC-PKT-param setting register) into the packet header.
When ”1” is set: For Send-operation, it automatically incorporates the tcode corresponding to the instruction issued into the packet header. For details, please see Section 4.7 in this document.

[s-IDstore] : R/W
When ”0” is set: It functions as ”No self-ID packet receipt”. When the chip receives a self-ID packet during the bus reset process it discards the received packet.
When ”1” is set: The chip stores the self-ID packet in the asynchronous receive-FIFO. Section 5.3 in this document further describes the operational flows for self-ID receive.

## 2.2 flag & status register (read): Address 02h

This resister indicates the status and data access request of the device.

| | bit 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Function : R | IPC busy | tran ready | tran busy | D-buff all empty | A-Tx buff empty | A-Rx buff empty | D-buff empty | A-Tx buff avail | A-Rx buff avail | D-buff avail | Int-insert | sleep | data req | recv busy | cmstr | INT |
| Default | '0' | '0' | '0' | '0' | '1' | '1' | "1" | '1' | '1' | '1' | '0' | '0' | '0' | '0' | '0' | '0' |

[IPC busy]: R
"0" at this bit indicates that the chip is ready for receiving an instruction.
"1" at this bit indicates that the chip is busy and not ready for receiving an instruction.

[tran ready] : R
"0" at this bit indicates that the bus reset is in-progress or in forced-sleep state so no packet transfer possible.
"1" at this bit indicates that the bus reset has been completed and now ready to transfer packets.

[tran busy] : R
"0" at this bit indicates that the chip is not transmitting or receiving a packet.
"1" at this bit indicates that the chip is transmitting or receiving a packet.

[D-buff all empty] : R (ISO cycle)
Status mode 0: "0" at this bit indicates that the chip is not in the isochronous cycle.
Status mode 1: "0" at this bits indicates that no data is in the bank of all D-Buff. (no data is in all banks of D-Buffer.)
Status mode 0: "1" at this bit indicates that the chip is in the isochronous cycle.
Status mode 1: "1" at this bits indicates that data is in the bank of D-Buff.

[A-Tx buff-empty] : R
"0" at this bit indicates that the asynchronous transmit-FIFO is not empty.
"1" at this bit indicates that the asynchronous transmit-FIFO is empty.

[A-Rx buff-empty] : R
"0" at this bit indicates that the asynchronous receive-FIFO is not empty.
"1" at this bit indicates that the asynchronous receive-FIFO is empty.

[D-buff-empty] : R
"0" at this bit indicates that the transmit/receive-FIFO has no bank with empty.
"1" at this bit indicates that the transmit/receive-FIFO has the bank with empty.

[A-Tx buff-avail] : R
"0" at this bit indicates that the asynchronous transmit-FIFO has 1 packet data already and so no more packet can be stored.
"1" at this bit indicates that the asynchronous transmit-FIFO has a space available to store packet.

[A-Rx buff-avail] : R
”0” at this bit indicates that the asynchronous receive-FIFO has 1 packet data already and so no more packet can be stored.
”1” at this bit indicates that the asynchronous receive-FIFO has a space available to store packet.

[D-buff-avail] : R
”0” at this bit indicates that all the banks of transmit/receive-FIFO have 1 packet data each and no more packet can be stored.
”1” at this bit indicates that at least 1 bank still has a space available to store packet.

[int-reset]
At 1, it indicates that the device is releasing from the internal reset state. At 0, it indicates that the device is in internal reset state.

Note: After the powered-on, the chip retains the internal reset state until the on-chip PLL is locked. The PHY and Link layers do not start operating in this state.

[sleep]
At 1, it indicates that the device is in forced-sleep state. At 0, it indicates that the device is not in forced-sleep state.

[data-req]
”0” at this bit indicates that the asynchronous receive-FIFO does not contain any data.
”1” at this bit indicates that the asynchronous receive-FIFO contains any data.

This bit is used for reading out self-ID packet contained in the asynchronous receive-FIFO.

[recv-busy]
At 0, it indicates that the device is not in busy mode. At 1, it indicates the device is in busy mode because of an asynchronous/self-ID packet received or ’start busy mode’ instruction (03h) issued.

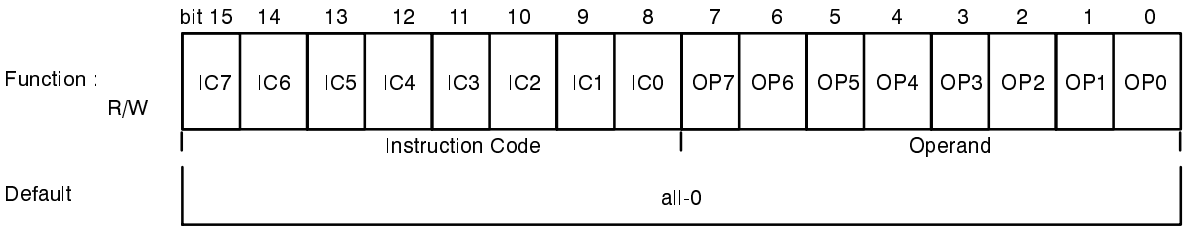
Note: When an asynchronous packet is received while this bit indicates ”1” , the chip transmits ’ack_busy_X’. But, one exception is that the chip is receiving a read response for data block packet during the chain transfer. In this case there is no relationship between the packet receipt and the busy mode indicated by this bit.

[cmstr]
”0” at this bit means the device is not a cycle master.
”1” at this bit means the device is a cycle master.

[INT]
Interrupt flag bit.
”0” at this bit means the device does not hold interrupt(s).
”1” at this bit means the device holds interrupt(s).

## 2.3 Instruction-fetch register (read/write): Address 04h

This register is composed of ”instruction code” and ”Operand”, used for writing the instruction to the device. See Chapter 3 for details of instruction codes and operands.



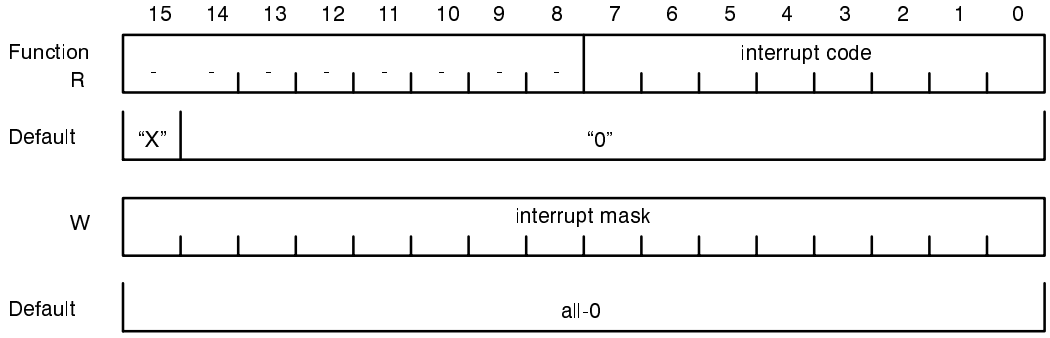

Note: Before writing an instruction code, please check the IPC busy bit of flag&status register and make sure it indicates ”0”.

[instruction-code] :Bits 15 to 8
Specify an instruction code as listed in Chapter 3.

[operand] : Bits 7 to 0
Specify operand for the instruction. If no operand instruction, write ”0” at this field.

## 2.4 Interrupt coderegister(read) /Interrupt mask register(write) : Address 06h

Interrupt code register indicates the interrupt reported by the device.
Refer to Section 4 for the details of interrupt code.
Interrupt mask register controls the mask of each interrupt factor generated by the device.
It is controlled with flags by bit. Refer to Section 4 for the details of interrupt factors to be masked.




[UNUSED] : Bits 15 to 8
Undefined value is read out from these bits at the read operation. ”0” data is read out from this field. However, “X” data is read out from bit 15 in interrupt code register.

[interrupt code (read)]
Stores 8-bit code for each interrupt factor (This is a FIFO type (8-byte) register, and saves up to 8 interrupts generated by the chip.)

[interrupt mask (write)]
Specifies mask of each interrupt factor. The interrupt code masked by setting with this register is not stored in the interrupt-code register and INT signal is not asserted.

## 2.5 Receive-acknowledge display register (read): Address 08h

This register indicates the acknowledge packet received to the chip itself.
This register must be read out after reporting "Asynchronous packet sent" interrupt (61h).

Interrupt code register indicates the interrupt reported by the device.
Refer to Section 4 for the details of interrupt code.
Interrupt mask register controls the mask of each interrupt factor generated by the device.
It is controlled with flags by bit. Refer to Section 4 for the details of interrupt factors to be masked.

| | bit 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Function : R | - | - | - | - | - | - | - | - | AC3 | AC2 | AC1 | AC0 | AP3 | AP2 | AP1 | AP0 |
| | | | | | | | | | Received-acknowledge | | | | Received-ack-parity | | | |
| Default | all-'0' | | | | | | | | | | | | | | | |

[UNUSED] : Bits 15 to 8
"0" is read out from this field.

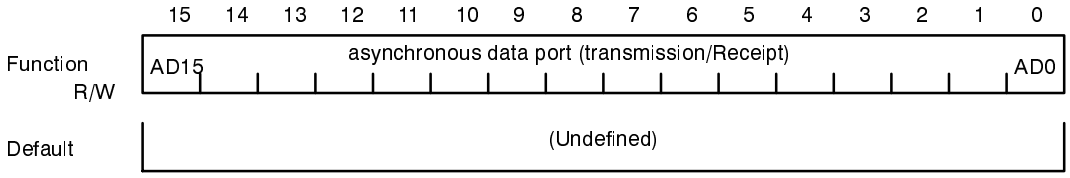
[received-acknowledge ] : Bits 7 to 4
This field indicates the code section of the received acknowledge packet. (MSB:AC3, LSB: AC0) (MSB:bit7, LSB: bit4)

[received-ack-parity]
This field indicates the parity section of the received acknowledge packet (MSB:AP3, LSB:AP) (MSB:bit3, LSB: bit0)

When an acknowledge code could not be received within a specified time, the indication of this register retuns to the initial value "0000h", and the chip reports "acknowledge missing" interrupt (91h).

## 2.6 A-Buffer Data Port (Receive)(read)/(Transmit) (write) : Address 0Ah

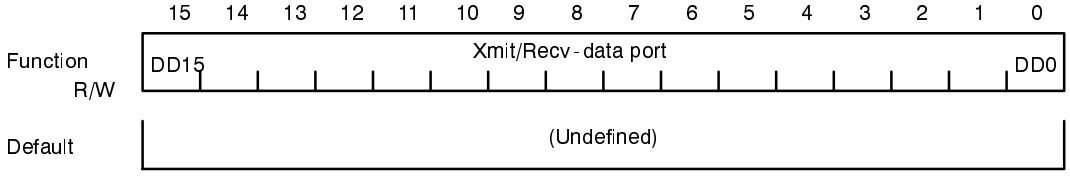
This is a data-port register for accessing the asynchronous receive- or transmit-FIFO.
The read data can be read out in the order of the receipt of 1394 bus-packet (MSB: 1st read). MSB: AD15, LSB: AD0
The write data is sent out as 1394 bus-packet in the order of written (MSB: 1st write). MSB: AD15, LSB: AD0




[asynchronous data-port ] : Bits 15 to 0
For Read operation, these are the receive-FIFO read out ports for Asynchronous transfer. (MSB:bit15, LSB: bit0)
For Write operation, these are the transmit-FIFO write ports for Asynchronous transfer. (MSB:bit15, LSB: bit0)

## 2.7 D-Buffer Data Port (Receive)(read)/(Transmit) (write) : Address 0Ch

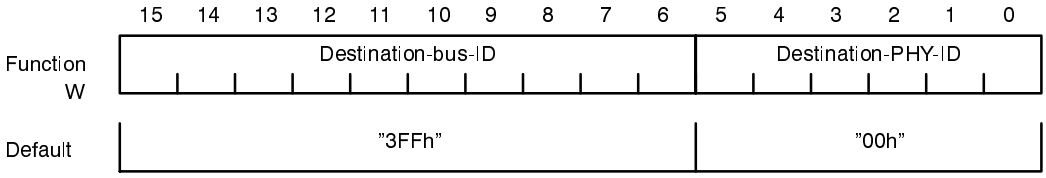
This is a data-port register for accessing the transmit/receive-FIFO.
It accesses to the Xmit/recv-data port without the DMA transfer and can write/read out the transmitting/received data.
The data written in this port is sent out as 1394 bus-packet in order of written (MSB: 1st write). MSB: DD15, LSB: DD0
The read data is deda out l order the receipt with 1394 bus-packet (MSB: 1st read). MSB: DD15, LSB: DD0




[Xmit/Recv-data port ] : Bits 15 to 0
For Read operation, these are the Xmit/recv-FIFO read out ports. (MSB:bit15, LSB: bit0)
For Write operation, these are the Xmit/recv-FIFO write ports. (MSB:bit15, LSB: bit0)

## 2.8 Xmit-ASYNC-Destination-ID setting register (write) : Addr:12h

This register sets the destination ID field of header at Asynchronous packet transmission.

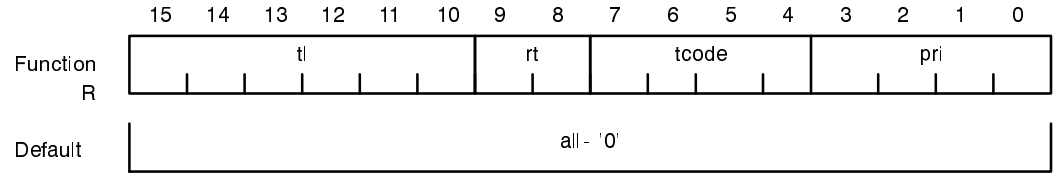


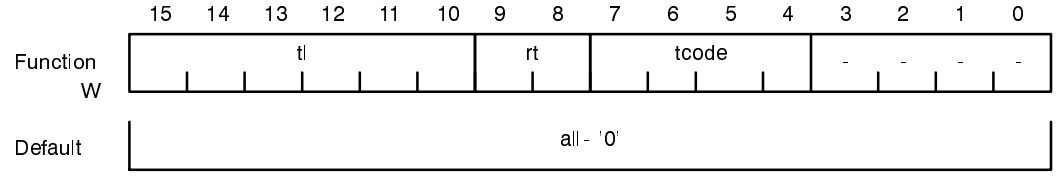

[Destination-bus-ID]
Set the destination bus-ID for transmit-asynchronous packet header (MSB:bit15, LSB:bit6) Set the destination-bus-ID.

[Destination-PHY-ID]
Set the destination physical-ID for transmit-asynchronous packet header (MSB:bit5, LSB:bit0) Set the destination-physical-ID.

## 2.9 Xmit-/Recv-Asynchronous-packet-parameteregister (write/read):Addr:14h

Xmit-Asynchronous-packet-parameter register sets the tl, rt, tcode and pri fields of header at Asynchronous packet transmission.
Recv-Asynchronous-packet-parameter register indicates the tl, rt, tcode and pri fields of header at received Asynchronous packet.




[tl]
For read-operation, it indicates the tl for receive-asynchronous packet header (MSB:bit15, LSB:bit10)
For write-operation, set the tl for transmit-asynchronous packet header (MSB:bit15, LSB:bit10)

[rt]
For read-operation, it indicates the rt(retry code) for receive-asynchronous packet header (MSB:bit9, LSB:bit8).
For write-operation, set the rt(retry code) for transmit-asynchronous packet header (MSB:bit9, LSB:bit8)

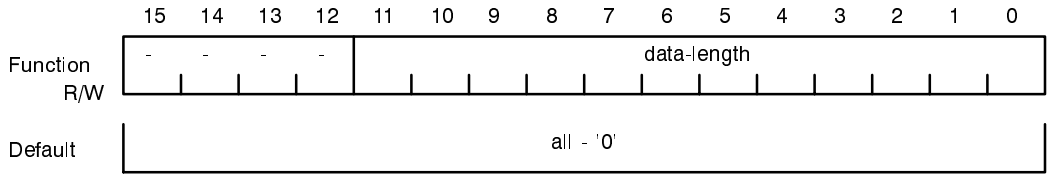
[tcode]
For read-operation, it indicates the tcode(transaction code) for receive-asynchronous packet header (MSB:bit7, LSB:bit4).
For write-operation, set the tcode(transaction code) for transmit-asynchronous packet header (MSB:bit7, LSB:bit4). These bits setting is invalid when "1" is set in the tcode-add bit (bit 1 of the mode-control register:addr00h).

The tcode (transaction code) defines the packet form and processing. If a tcode that can not be recognized is received, the header and data are not stored, and no interrupt is reported.

[pri] : Bits 3 to 0
For read-operation, it indicates the pri field for receive-asynchronous packet packet header (MSB:bit3, LSl:bit0).
For write-operation, this is a reserved field. Always write '0' at this field. (The pri field of transmit-asynchronous packet header is always fixed with '0h'.

## 2.10 Xmit-/Recv-Asynchronous-data-lengthregister (write/read): Address 16h

Xmit-Asynchronous-data-length register sets the data-length field of header at asynchronous packet transmission.
Recv-Asynchronous-data-length register indicates the data-length field of received asynchronous packet header.




[UNUSED] : Bits 15 to 12
Reserved field.
For read-operation, it always indicates '0h'.
For write-operation, always write '0h' to this field.

[data-length] : Bits 11 to 0
For read-operation, it indicates the lower 12-bit of data-length for receive-asynchronous packet header (MSB:bit11, LSB:bit0)
For write-operation, set the lower 12-bit of data-length for transmit-asynchronous packet header (MSB:bit11, LSB:bit0)

Note 1: When the actual data-length of packet received was shorter than the value that is written in the header of the packet, the chip pads "0" or "1" data according to the pad bit setting in mode-control register to the received packet and stores it in the transmit/receive-FIFO or asynchronous receive-FIFO. Then, it reports "Data length short error (Asynchronous)" interrupt (31h).

Note 2: When the actual data-length of packet received was longer than the value that is written in the header of the packet, the chip stores the data in the FIFO up to the data-length specified by the data-length field and discards the remained data. Then, it reports "Data length long error (Asynchronous)" interrupt (32h).

Note 3 : If the transmit data is stored in the asynchronous transmit-FIFO, the data-length that can be specified in this field is 0 to 128-byte.

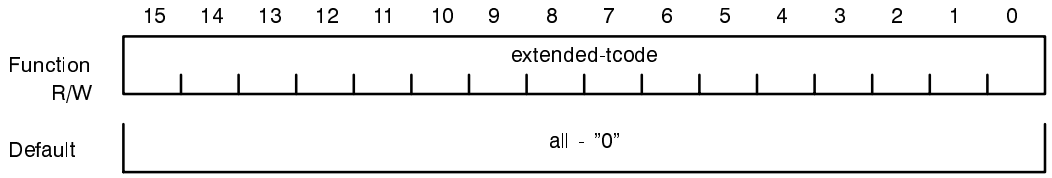
Note 4 : If the transmit data is stored in the transmit/receive-FIFO, the data-length that can be specified in this field is 1 to 2048-byte regardless of the bank configuration.

Note 5 :The chip makes the packetization to the data as specified in this field to read out the data from the FIFO.

Note 6 : If this field contains a small value that does not reach a quadlet level, the chip pads '0' data until quadlet and generates a packet.

## 2.11 Xmit-/Recv-Async extended-tcode register (write/read): Address 18h

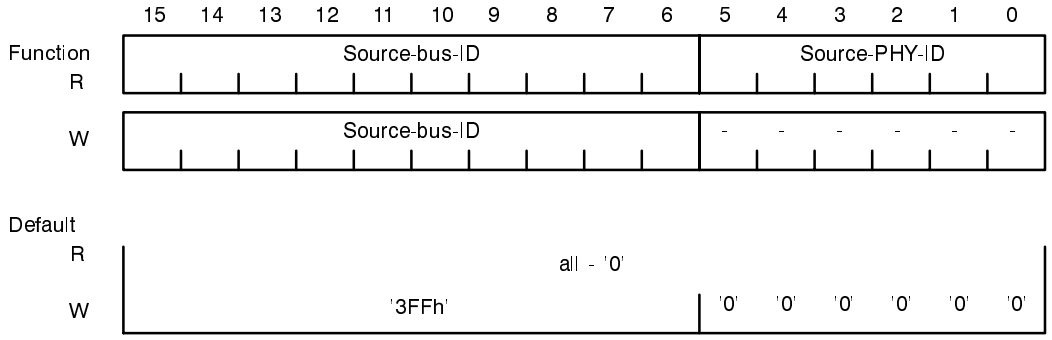
Xmit-Async extended-tcode register sets the ex-tcode field at asynchronous packet transmission.
Recv-Async extended-tcode register indicates the ex-tcode field of received asynchronous packet header.




[extended-tcode]
For read-operation, it indicates the extended-tcode for receive-ASYNC-packet header (MSB:bit15, LSB:bit0)
For write-operation, set the extended-tcode for send-ASYNC-packet header (MSB:bit15, LSB:bit0) The extended-tcode field is valid only for packet transmission on lock-request and lock-response.
For other asynchronous packet transmission, 0000h is specified in the packet irrespective of the register value.

## 2.12 Xmit-ASYNC-source-bus-ID/Receive-ASYNC-source-ID register (write/read) : Address 1Ah

Xmit-ASYNC-source-bus-ID register sets the source-bus-ID field at asynchronous packet header transmission.
Receive-ASYNC-source-ID register indicates the source-ID field of received asynchronous packet header.

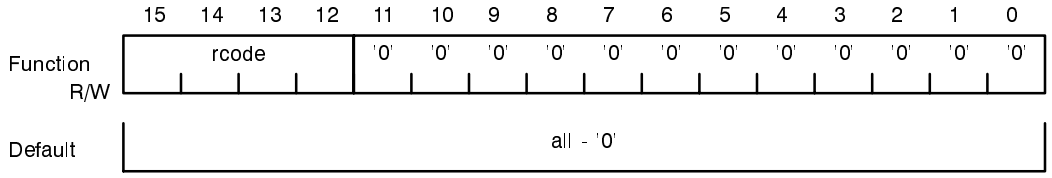



[Source-bus-ID] :
For read-operation, this field indicates the source-bus-ID for received asynchronous packet header (MSB:bit 15, LSB:bit6).
For write-operation, set the source-bus-ID for transmit-asynchronous packet header (MSB:bit 15, LSB:bit6). Specify the source-bus-ID to this field.

[Source-PHY-ID]
For read-operation, it indicates the source-physical-ID for received asynchronous packet header (MSB:bit5, LSB:bit0).
For write-operation, set "0" to this field. (The chip always sets the current physical ID held by the PHY layer internally.)

## 2.13 Xmit-/Receive-Async rcode register (write/read): Address 1Ch

Xmit-Async rcode register sets the rcode field at asynchronous packet header transmission.
Receive-Async rcode register indicates the rcode (response code) field of received asynchronous packet header.

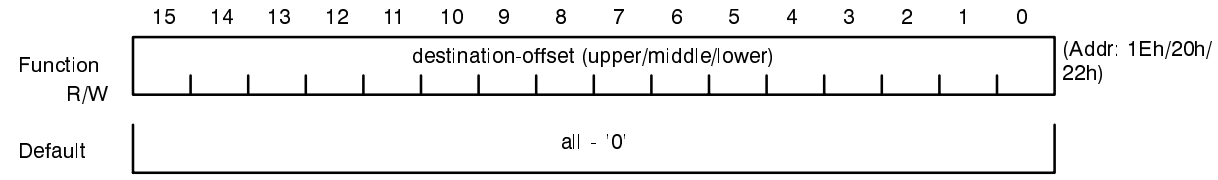



[rcode]
For read-operation, this field indicates the rcode (response code) for received asynchronous packet header (MSB:bit15, LSB: bit12).
For write-operation, set the rcode (response code) for transmit-asynchronous packet header (MSB:bit15, LSB: bit12).

[UNUSED] : bits 11 to 0
Read out value from this field is '0'. For write-operation, always set "0" to this field.

## 2.14 Xmit-/Receive-Async destination-offset register (u/m/l) (write/read): Addr: 1Eh/20h/22h

Xmit-Async destination-offset register sets the destination-offset at asynchronous packet header transmission.
Receive-Async destination-offset register indicates the destination-offset of received asynchronous packet header.



[destination-offset]

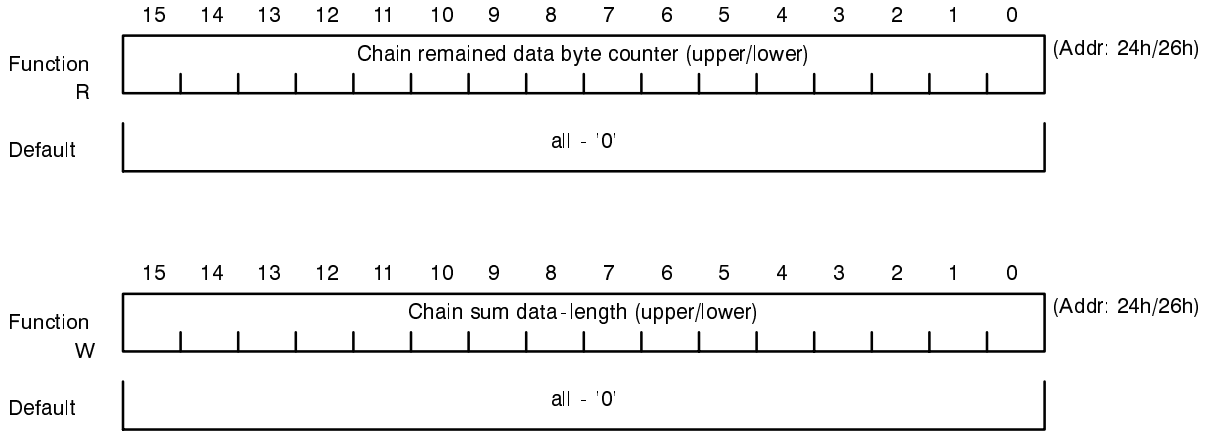
For read-operation, it indicates the destination-offset for receive-asynchronous packet header (MSB:bit15, LSB:bit0).

For write-operation, set the destination-offset for transmit-asynchronous packet header (MSB:bit15, LSB:bit0). set the lower 48-bit of destination node address for the request packet.

## 2.15 Chain remained data byte counter (u/l) (read) / Chain sum data-length setting register(u/l) (write) : Addr: 24h/26h

Chain remained data byte counter register indicates the remained data byte transferred to an host side in chain mode.

Chain sum data-length setting register is used for setting a sum of data-length to be transferred by chain with 'asynchronous chain' instruction.




For read operation, this register functions as a counter that indicates the remained byte count for data transferred to an host side in chain mode.

For write operation, this register is used for setting a sum of data-length to be transferred by chain with 'asynchronous chain' instruction.

[Chain remained data byte]

This field indicates the remained byte count for data transferred to an host side in chain mode. (MSB:bit15, LSB:bit0).

Note: This counter in decrement every packet transferred to a host side.

[Chain sum data-length]

This field specifies a sum of data-length transferred in chain mode. (MSB:bit15, LSB:bit0).

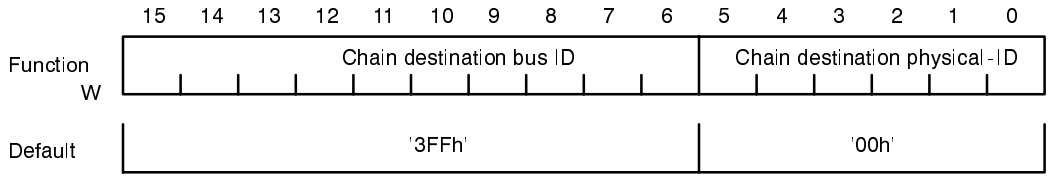
Notes 1. The data-length range that can be set is from 1h to FFFFFFFFh bytes.

2. Upon the receipt of 'asynchronous chain' instruction, the chip transmits 'read request for data block packet' that is set to Chain Xmit-data-length setting register (30h) and repeats this action until all the data specified in this field are read out.

## 2.16 Ping Time Monitor register (read) / Chain Xmit-destination-ID setting register (write) : Address 28h

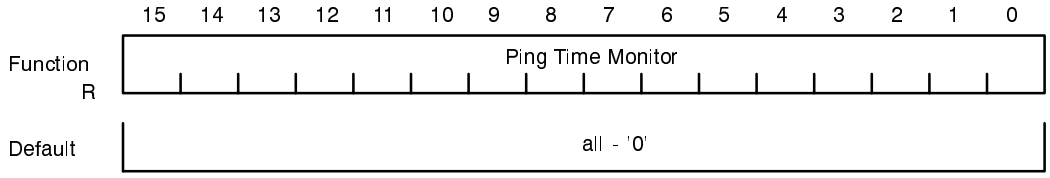
Ping Time Monitor register displays the ping time that indicates the time until a self node receives the response.

Chain Xmit-destination-ID setting register specifies the destination IDs for bus-ID and physical-ID described in header area of Read request for data block packet automatically transmitted by chain mode data transfer.




[Chain destination bus ID] : bits15 to 6
This field specifies the destination bus-ID described in the packet header for 'read request for data block'. (MSB :bit15, LSB:bit6)

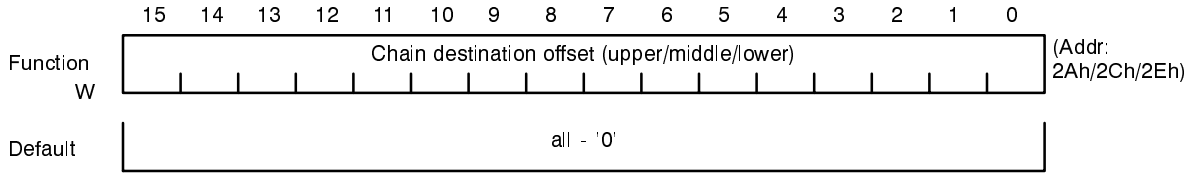
[Chain destination physical-ID] : bits 5 to 0
This field specifies the destination physical-ID described in the packet header for 'read request for data block'. (MSB :bit5, LSB:bit0)




[Ping Time Monitor] : bits15 to 0
This field displays the ping time that indicates the time until a self node receives the response.
1-bit is in 20ns unit. (Precisely, 2 times of 24.576MHz).

## 2.17 Chain Xmit-destination offset setting register (u/m/l)(write) : Address 2Ah/2Ch/2Eh

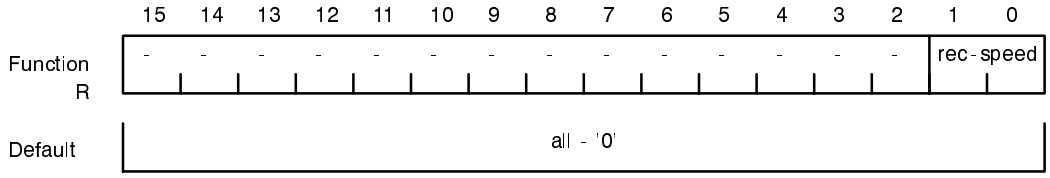
This register specifies the destination-offset for the first 'read request for data block'.




[Chain destination offset] : bits15 to 0
This field specifies the destination offset described in the packet header for 'read request for data block'. (MSB :bit15, LSB:bit0)
Note: Set the address for page-boundary of Initiator's memory in this field. If the starting point of data field indicated by ORB is not in the boundary, the host side should have all the responsibility to read out the data until the page boundary and execute the chain transfer.

## 2.18 Recv-packettransfer speed display register (read): Address 30h

This register indicates the transfer speed of received asynchronous packet.




[UNUSED]: bits 15 to 2
This field always indicates '0'.

[rec-speed]: bits 1 and 0
This field indicates the transfer speed of received asynchronous packet.
It displays the value for each packet other than the followings:

- Read response packet at Read chain
- Write response packet at Write chain

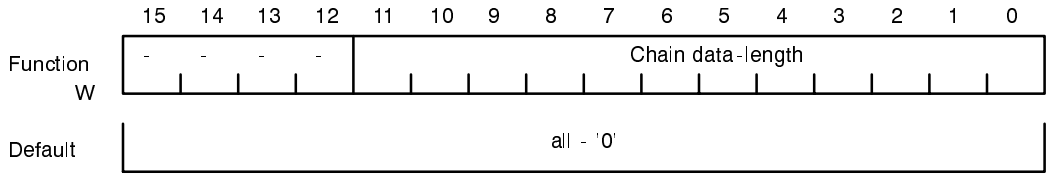
Transfer speed
s100 (Bit1: 0, Bit0: 0)
s200 (Bit1: 0, Bit0: 1)
s400 (Bit1: 1, Bit0: 0)
Reserved (Bit1: 1, Bit0: 1)

Notes:
1. The register does not change the value when it receives the PHY packet or Cycle-start packet.
2. Refer to the table below when an error with an interrupt code is detected when receiving packet.

When an error occurs:
Data length short error (31h): Data change
Data length long error (32h): Data change
Packet format error (33h): No change
Data end not found (34h): Data change
Header CRC error (35h): No change
Data CRC error (36h): Data change
Data length invalid (37h): Data change

## 2.19 Chain Xmit-data-length setting register (write) : Address 30h

This register specifies the data-length field of packet header for 'read request for data block' packet which is automatically transmitted in the chain transfer.

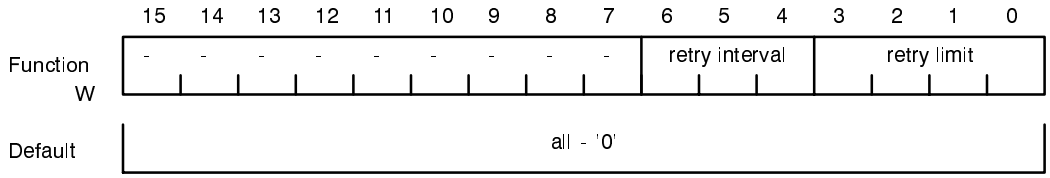



[UNUSED]: Bits 15 to 12
Always write '0' to this field.

[Chain data-length] : bits11 to 0
This field specifies the lower 12-bit of data-length field of packet header for 'read request for data block' packet which is automatically transmitted in the chain transfer. (MSB: bit11, LSB: bit0)
Notes 1.Set a measure or multiple of the memory page size owned by an initiator node within 1 to 2048-byte.
2.When the sum data-length is not equal to the multiple of this register, the value of this data-length field is autumatically calculated since the last packet length is the fraction.

## 2.20 Chain-Retry setting register (write): Address 32h

This register specifies the time interval and the maximum number of retry automatically performed when an acknowledge 'ack_busy_X' is received in response to 'read request for data block' packet automatically transmitted in the chain transfer.




[UNUSED]: bits 15 to 7
Always write '0' to this field.

[retry-interval]: bits 6 to 4
This field specifies the approximate time interval of retry in the chain transfer. Refer to the list of 8.29b shown below.

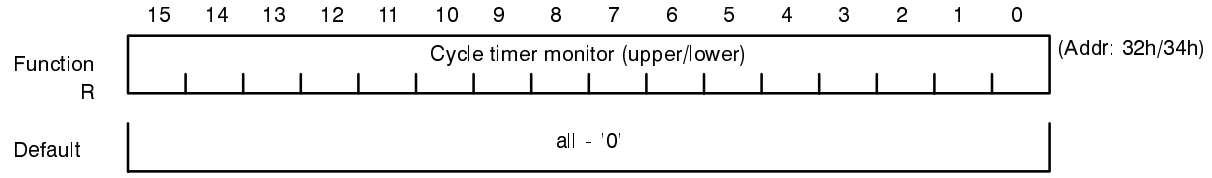
[retry-limit]: bits 3 to 0
This register specifies the maximum number of retry automatically performed when an acknowledge 'ack_busy_X' is received in response to 'read request for data block' packet automatically transmitted in the chain transfer. (MSB: bit3, LSB: bit0)

Notes 1. Retry number that can be set is 0 to 15.
2. When 'ack_pending' is not received even after retrying as number specified, the chip stops the chain transfer in a half way and reports 'Retry time out' interrupt (F6h).

Time interval
Immediately (Setting value: 000)
Approx. 125 us (Setting value: 001)
Approx. 250 us (Setting value: 010)
Approx. 500 us (Setting value: 011)
Approx. 1 ms (Setting value: 100)
Approx. 2 ms (Setting value: 101)
Approx. 4 ms (Setting value: 110)
Approx. 8 ms (Setting value: 111)

## 2.21 Cycletimer monitor register (u/l)(read): Address 32h/34h

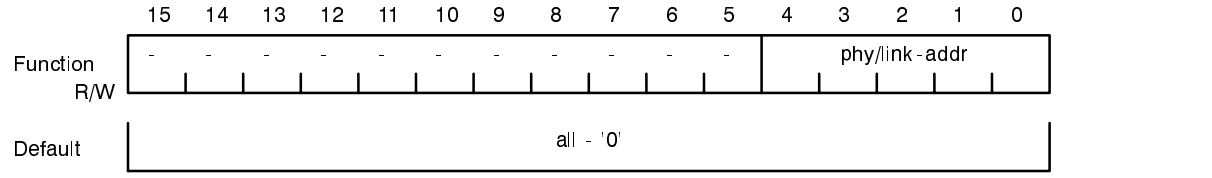
This register indicates the on-chip cycle-timer register value.




[Cycle timer monitor] : bits15 to 0
This field indicates the on-chip cycle-timer register value. (MSB:bit15, LSB:bit0)
Note : Reading out the lower word (at address 34h) activates latching the upper word (32h) and reading out the upper word releases the latch. Therefore, a combination of reading '34h' first and then '32h' is required.

## 2.22 PHY/LINK register address setting register (read/write): Address 38h

This register specifies the address for the indirect access to the internal PHY/LINK registers. The PHY/LINK register at the address selected by this register can be accessed through the PHY/LINK register access port.

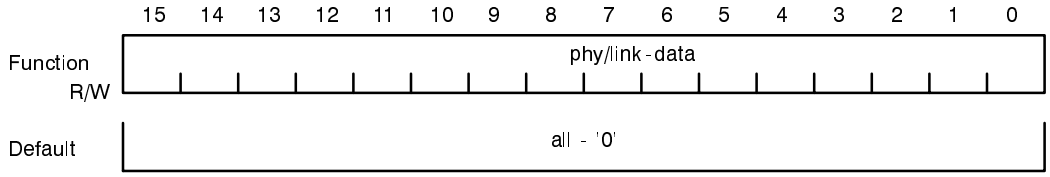



[ UNUSED ] : Bits 15 to 5
For read operation, '0' is read out from these bits.
For write operation, always write "0" at these bits.

[ phy/link-addr. ] : Bits 4 to 0
This field specifies the address for PHY/LINK register to be accessed. (MSB:bit4, LSB:bit0)

## 2.23 PHY/LINK register access port (read/write): Address 3Ah

This port is used for the indirect access to the internal PHY/LINK registers. The register at address selected by the PHY/LINK register address setting register can be accessed through this port.




[ phy/link-data] : Bits 15 to 0
For read operation, contents of register selected by 'phy/link-addr' are read out. (MSB:bit15, LSB:bit0)
For write operation, specify the data written to the register that 'phy/link-addr' selects.

## 2.24 PHY/LINK Register Functional Description

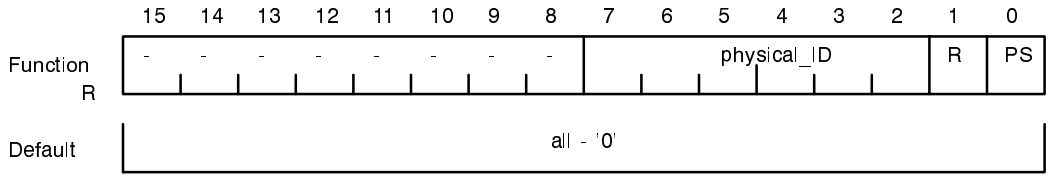
The PHY/LINK Register specifies the address with the PHY/LINK register address setting register (address 38h) to enable reading out/writing to several registers through the PHY/LINK register access port.

The list below shows the PHY/LINK registers on the MB86614A.
Refer to the paragraphs 2.33.2 to 2.33.10 for their details.

| phy/link-addr | Register Name |
|---|---|
| 00h | Physical register #0 |
| 02h | Physical register #1 |
| 04h | Physical register #2 |
| 06h | Physical register #3 |
| 08h | Physical register #4 |
| 0Ah | Physical register #5 |
| 0Ch | Physical register #6 |
| 0Eh | Physical register #7 |
| 10h | Physical register #8 |
| 12h | Physical register #9 |
| 14h | reserve |
| 16h | reserve |
| 18h | Link register #0 |

### 2.24.1 Physical register #0 (read): phy/link-addr: 00h

This register indicates the parameters related to the connection with the 1394 bus.




[UNUSED] : bits 15 to 8
This unused field indicates '0'.

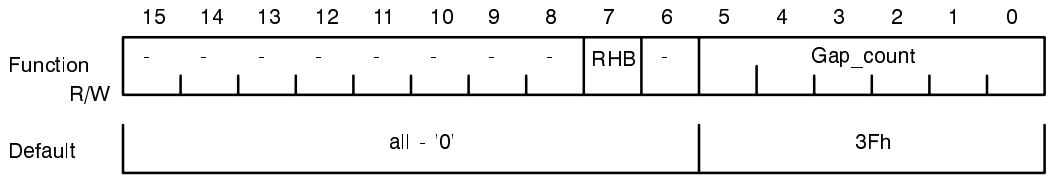
[physical_ID]
This field indicates the number of self node determined in the self-identify process in bus reset. (MSB:bit7, LSB:bit2) This field is valid after the bus reset completed.

[ R ]
'0' at this bit indicates that the self node is not a root.
'1' at this bit indicates that the self node is a root.

[ PS ]
'0' at this bit indicates that the cable power being supplied to the node is below the specification.
'1' at this bit indicates that the cable power being supplied to the node is beyond the specification.
This bit is valid only when a comparator circuitry to detect the cable power is externally connected with the chip. Also the threshold level in the specification is 8V.

### 2.24.2 Physical register #1 (read/write): phy/link-addr: 02h

This register specifies the gap count value and setting to be a root.
It also indicates the current gap count value and state to be a root.
Writing operation to this register is allowed only when the node is 'bus manager' or when it is 'isochronous resource manager' if 'bus manager' does not exist.



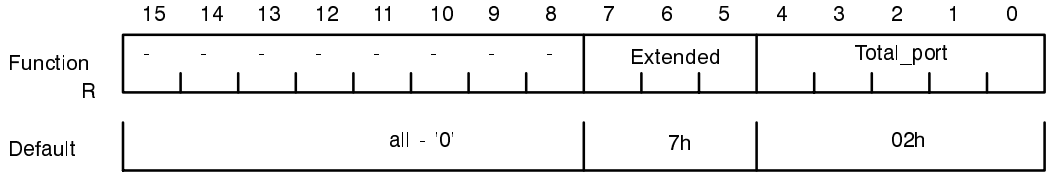

[UNUSED] : bits 15 to 8 and 6
For read operation, this unused field indicates '0'.
For write operation, always write '0' at this field.

[RHB]
When '0' is specified to this field, the chip does not intend to be a root in the next bus reset.
When '1' is specified, the chip intends to be a root in the next bus reset.
Note: This field is also set upon receipt of a PHY configuration packet automatically.

[Gap_count]
For read operation, this field indicates the current gap_count value. (MSB:bit5, LSB:bit0)
For write operation, set a gap_count value (MSB:bit 5, LSB:bit0)
Note: This field is also set upon receipt of a PHY configuration packet automatically. Also the gap_count value returns to the default again with twice bus reset.

### 2.24.3 Physical register #2 (read): phy/link-addr: 04h

This register indicates the fixed value of the physical layer parameters.



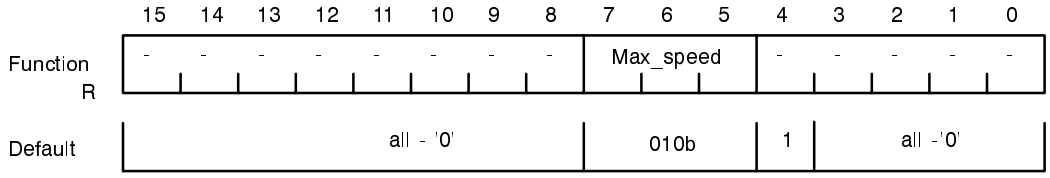

[UNUSED] : bits 15 to 8
This unused field indicates '0'.

[Extended]
This field indicates '7h' which means that the self node has an extended PHY register map.

[Total_port]
This field indicates '02h' which means the self node has 2 ports.

### 2.24.4 Physical register #3 (read): phy/link-addr: 06h

This register indicates the maximum transfer speed (fixed) of the physical layer.



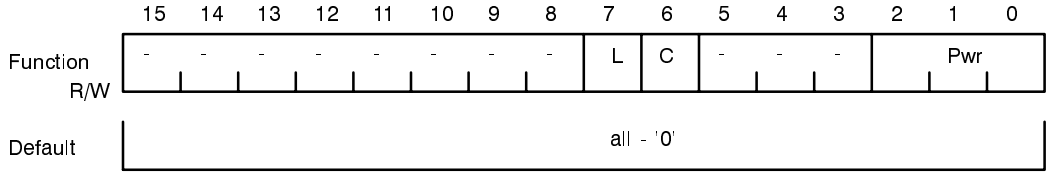

[UNUSED] : bits 15 to 8 and 3 to 0
This unused field indicates '0'.

[Max_speed]
This field indicates the maximum transfer speed supported by the PHY. '010b' which means s400 rate is read out.

[UNUSED]: bit 4
This unused field indicates '1'.

### 2.24.5 Physical register #4 (read/write): phy/link-addr: 08h

This register specifies the parameters required for the self_ID packet.




[UNUSED] : bits 15 to 8 and 5 to 3
For read operation, these unused fields indicate '0'.
For write operation, set '0' at these fields.

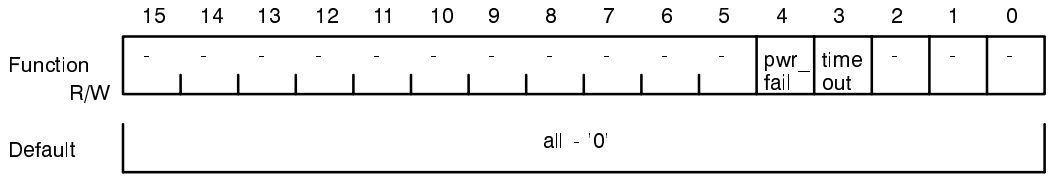
[ L ]
This field specifies the L bit value (link_active field) for self-ID packet that is transmitted automatically at the system powered operation.
Note: For the cable powered operation, the L value for self-ID packet is always '0' regardless of this bit setting.

[ C ]
This field specifies the C bit value (contender field) for self-ID packet that is transmitted automatically at the system powered operation.
Note: For the cable powered operation, the C value for self-ID packet is always '0' regardless of this bit setting.

[ Pwr ]
This field specifies the pwr bit value (power_class field) for self-ID packet that is transmitted automatically at the system powered operation.
Note: For the cable powered operation, the pwr value for self-ID packet is always in accordance with the PWR1 to PWR3 pin levels regardless of this bit setting.

### 2.24.6 Physical register #5 (read/write): phy/link-addr: 0Ah

This register indicates the current state of 1394 bus cable and detection of arbitration time out. Writing operation clears the bit value.



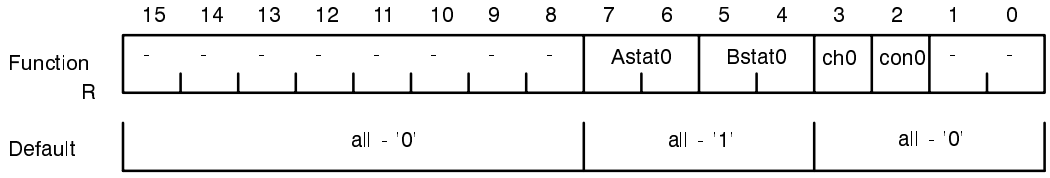
[UNUSED] : bits 15 to 5 and 2 to 0
For read operation, these unused fields indicate '0'.
For write operation, set '0' at these fields.

[ Pwr_fail ]
'0' at this bit indicates that the cable power satisfies the specification.
'1' at this bit indicates that a cable power level is failed because it does not satisfy the specification.
When '1' is written at this bit, the bit value is cleared with '0'.

[ Timeout ]
'0' at this bit indicates that no timeout is detected in the arbitration state machine.
'1' at this bit indicates that the timeout is detected in the arbitration state machine.
When '1' is written at this bit, the bit value is cleared with '0'.

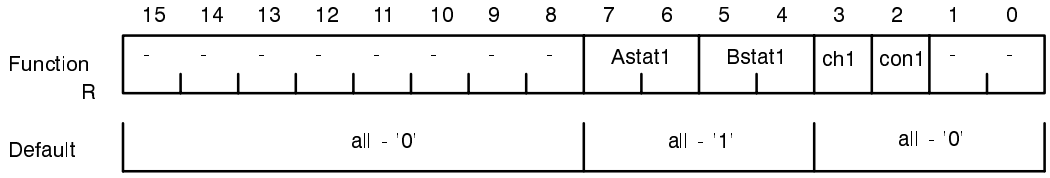
### 2.24.7 Physical register #6 (read): phy/link-addr: 0Ch

This register indicates the state of connection between the 1394 bus signals and other nodes at 1394 port 0 (for #6) and port 1 (for #7) respectively.




### 2.24.8 Physical register #7 (read): phy/link-addr: 0Eh

This register indicates the state of connection between the 1394 bus signals and other nodes at 1394 port 0 (for #6) and port 1 (for #7) respectively.




[UNUSED] : bits 15 to 8, 1, and 0
These unused fields indicate '0'.

[ AstatN ]
This field indicates the TPA line state of Port N. (where N = 0 or 1) (MSB:bit7, LSB:bit6)
00 : invalid
01 : '1'
10 : '0'
11 : 'Z'

[ BstatN ]

This field indicates the TPB line state of Port N. (where N = 0 or 1) (MSB:bit5, LSB:bit4)

00 : invalid
01 : '1'
10 : '0'
11 : 'Z'

[ chN ]

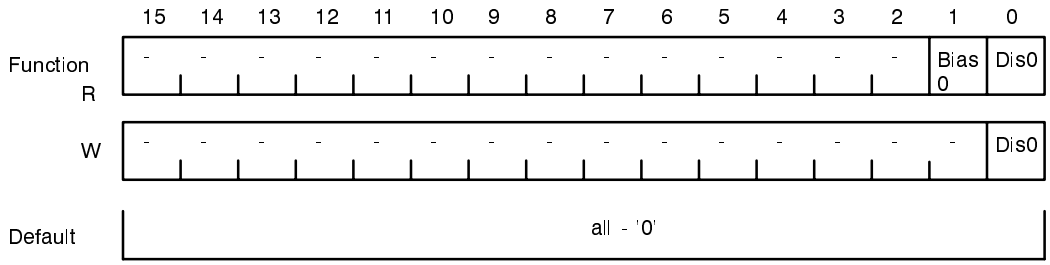
'0' at this bit indicates that the port N is a parent port. (where N=1 or 0)
'1' at this bit indicates that the port N is a child port. (where N = 1 or 0)

[ conN ]

'0' at this bit indicates that no cable is connected with port N. (where N=1 or 0)
'1' at this bit indicates that a cable is connected with port N. (where N = 1 or 0)

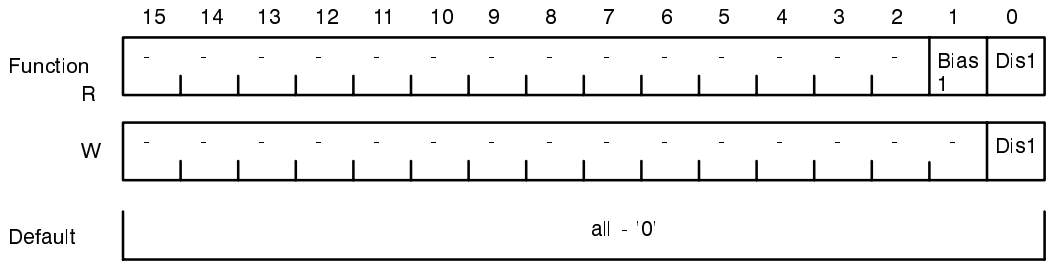
### 2.24.9 Physical register #8 (read/write): phy/link-addr: 10h

This register indicates the enable/disable control, state of port and bias detection at 1394 port 0 (for #8) and port 1 (for #9) respectively.




### 2.24.10 Physical register #9 (read/write): phy/link-addr: 12h

This register indicates the enable/disable control, state of port and bias detection at 1394 port 0 (for #8) and port 1 (for #9) respectively.




[UNUSED] : bits 15 to 2

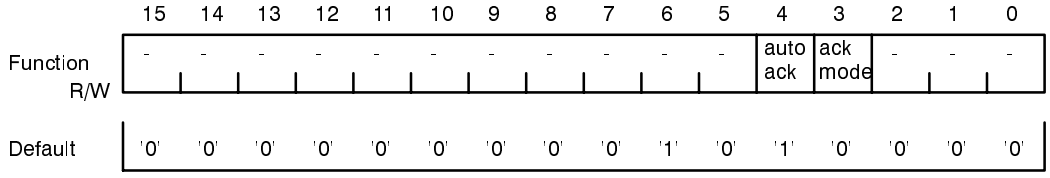
For read operation, this unused field indicates '0'.
For write operation, always write '0' at this field.

[ BiasN ]
'0' at this bit indicates that no bias voltage is detected in the port N. (where N=1 or 0)
'1' at this bit indicates that a bias voltage is detected in the port N. (where N = 1 or 0)
For writing at the register, please set '0' to this field.

[ DisN ]
'0' at this bit enables the port N. (where N=1 or 0)
'1' at this bit disables the port N. (where N = 1 or 0)

### 2.24.11 Link register #10 (read/write): phy/link-addr: 18h

This register selects the transmission mode of the acknowledge packet when this chip receives the asynchronous packet other than the 'broadcast'.




[UNUSED] : bits 15 to 5, 2 to 0
For read operation, this unused field indicates '0'.
For write operation, always write '0' at this field.

[ auto_ack ]
'0' at this bit does not transmit the acknowledge packet automatically upon a receipt of asynchronous packet.
'1' at this bit automatically transmits the acknowledge packet upon a receipt of asynchronous packet.
The code of acknowledge packet automatically transmitted in response to a receipt of normal asynchronous packet is in accordance with the 'ack_mode' bit setting. The code for erroneous asynchronous packet varies with the error type.

Note: This bit must be "1" for the chain transfer.

[ auto_mode ]
This bit is valid only when auto_ack bit is '1'.
'0' at this bit transmits 'ack_pending' acknowledge packet for the all request packets. It also transmits 'ack_complete' packet for the all response packets.
'1' at this bit transmits 'ack_pending' packet for the all read request packets and lock request packet. It also transmits 'ack_complete' packet for all the write request packets and response packets.

## 3. INSTRUCTIONS

This chapter will explain the instruction code and corresponding operations.

The following tables lists the MB86614A instruction codes. Writing the instruction code and operand shown below to the instruction-fetch register enables the execution of instruction corresponded. Refer to 3.1 for the details of their functions.

| Nameof Instruction | Code (hex) | Operand | | |
|---|---|---|---|---|
| Start sleep | 01 | | | |
| Remove sleep | 02 | | | |
| Start busy mode | 03 | | | |
| Remove busy mode | 04 | | | |
| Bus reset | 11 | | | |
| Link reset | 12 | | | |
| Link init | 13 | | | |
| Send physical configuration packet with force-root | 21 | Remote physical ID | | |
| Send physical configuration packet with gap-count | 22 | Gap-count value | | |
| Send link-on packet | 23 | Remote physical ID | | |
| Send ping packet | 24 | Remote physical ID | | |
| Asynchronous write request for data quadlet | 30 | | FIFO select | Speed Code |
| Asynchronous write request for block payload | 31 | | FIFO select | Speed Code |
| Asynchronous write response for split transaction | 32 | | | Speed Code |
| Asynchronous read request for data quadlet | 33 | | | Speed Code |
| Asynchronous read request for block payload | 34 | | | Speed Code |
| Asynchronous read response for data quadlet | 35 | | FIFO select | Speed Code |
| Asynchronous read response for data block payload | 36 | | FIFO select | Speed Code |
| Lock request | 37 | | FIFO select | Speed Code |
| Lock response | 38 | | FIFO select | Speed Code |
| Asynchronous chain | 39 | | | Speed Code |

(Continued)

| Nameof Instruction | Code (hex) | Operand |
|---|---|---|
| Chain suspend | 61 | |
| Data-FIFO init | 63 | FIFO select code |
| DMA Tx | 71 | |
| DMA Rx | 72 | |
| DMA start (Chain) | 73 | |

## 3.1 Functional Descriptions of Instructions

Functions of several instruction codes will be described.

### 3.1.1 Start sleep (01h)

This code makes the MB86614A 'sleep' state. All the functions of driver/receiver on 1394 port stop and the power of the cable is turned OFF. In addition, the internal PLL operation also stops. The internal registers are still accessible. No interrupt is generated for this instruction. Therefore, please check the execution status of this instruction with "sleep" bit (bit 4 of flag&status register at addr 02h).

### 3.1.2 Remove sleep (02h)

This code resumes the chip from the sleep state. No interrupt is generated for this instruction. Therefore, please check the execution status of this instruction with "sleep" bit (bit 4 of flag&status register at addr 02h).

### 3.1.3 Start busy mode (03h)

Upon the receipt of asynchronous packet, the device shifts to the busy mode. This code sends the acknowledge of 'ack_busy_X' for busy in response to the receipt of normal asynchronous packet. The received-packet is not stored.

### 3.1.4 Remove busy mode (04h)

This code releases the busy mode set by the start busy mode instruction or busy mode activated by receiving normal asynchronous packet or self-ID packet.

### 3.1.5 Bus reset (11h)

This code starts the bus reset. It initializes the PHY layer and processes the bus reset (after BUS-RESET until self-identify complete.) For the information, the operation of the gap-count-disable flag is done automatically.

Notes:
1) When this instruction is issued under no cable port connected for Port-0 and Port-1 either, the device reports "bus occupancy violation" interrupt (14h). But please ignore this interrupt.

2) When this instruction is issued during the transmission, receipt, or repeat operation, the instruction is retained.

### 3.1.6 Link reset (12h)

This code resets the Link layer. The link layer status changes from any operational modes to reset state and stops the operation. All the pending processes and sub-actions are discarded and the broadcast packet is only received. This state is released by the Link init instruction (13h).

### 3.1.7 Link init (13h)

This code initializes the Link layer. The link layer is ready for sending/receiving all packets that MB86614A can be received.

### 3.1.8 Send physical configuration packetwith force-root (21h)

When the operands mentioned below are specified, the configuration packet is transferred to set the parameter to the remote node Physical layer. This instruction is used for making the force-root bit of other node "false" to make the force-root bit of a node which is matched with the remote Physical-ID. In IEEE1394 standard, this is possible only for the bus-manager or Isochronous-resource-manager when no bus-manager is existing.

| Bit | Operand | Meaning |
|---|---|---|
| 7, 6 | reserved | Always specify '0' at these bits. |
| 5 to 0 | remote Physical-ID | Please specify the Physical-ID of the node that sets the force-root bit to "true". |

### 3.1.9 Send physical configuration packetwith gap-count (22h)

When the operands mentioned below are specified, the configuration packet is transferred to set the parameter to the remote node Physical layer. This instruction sets the all the node's gap-count value at the operand's gap-count. In IEEE1394 standard, this is possible only for the bus-manager or Isochronous-resource-manager when no bus-manager is existing.

| Bit | Operand | Meaning |
|---|---|---|
| 7, 6 | reserved | Always specify '0' at these bits. |
| 5 to 0 | gap-count | The new value is set for the gap-count of all the nodes. (MSB: 5, LSB: 0) |

### 3.1.10 Send link-on packet(23h)

When the following operand is designated, the Link-on packet is transmitted to the remote node Physical layer

| Bit | Operand | Meaning |
|---|---|---|
| 7, 6 | reserved | Always specify '0' at these bits. |
| 5 to 0 | remote-Physical-ID | Specify the Physical-ID of the counterpart's node receiving the Link-on packet. (MSB: 5, LSB: 0) |

### 3.1.11 Send ping packet(24h)

When the following operand is designated, the ping packet is transmitted to the remote node Physical layer

| Bit | Operand | Meaning |
|---|---|---|
| 7, 6 | reserved | Always specify '0' at these bits. |
| 5 to 0 | remote-Physical-ID | Specify the Physical-ID of the counterpart's node receiving the ping packet. (MSB: 5, LSB: 0) |

### 3.1.12 Asynchronous write request for dataquadlet(30h)

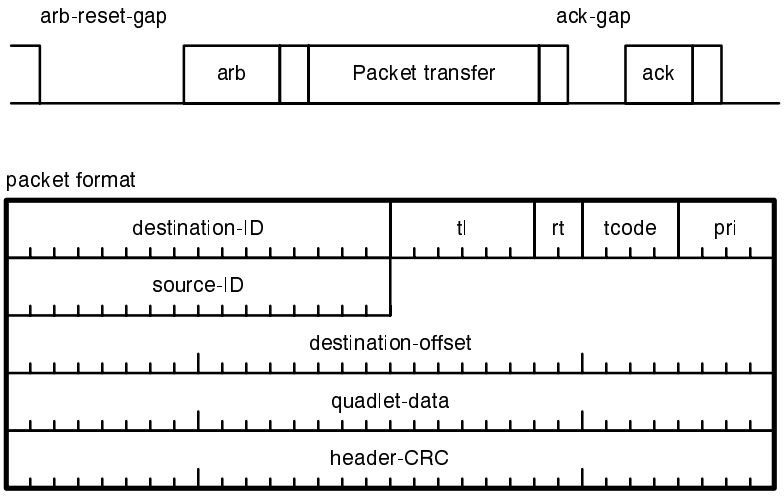
An instruction for data write of 1 quadlet by asynchronous transfer.
The data of one quadlet is incorporated in the header for transfer. By pre-setting the transfer byte number, destination (destination-ID) and destination offset at the stipulated register and buffer, the packet format is automatically generated and transferred.

This instruction gets in the arbitration by detecting the arb-reset-gap, generating and transferring the packet, and receiving the acknowledge packet.

Interrupt is reported when the packet is transmitted or the acknowledge is received. When an error occurs, the interrupt is reported and the operation is terminated.

The received acknowledge is stored in the receive-acknowledge display register as mentioned in section 2.5.

The transmit-data should be pre-set in the asynchronous transmission-FIFO or transmit/receive-FIFO.




Note: If the destination-ID is set as "broadcast", the operation is terminated not waiting for receiving the acknowledge.

| Bit | Operand | Meaning |
|---|---|---|
| 7 to 4 | reserved | Always specify '0' at these bits. |
| 3 | FIFO select | Specify the buffer to store the data:<br>0 = asynchronous transmit-FIFO<br>1 = transmit/receive-FIFO |
| 2 to 0 | speed code | Specify the transmission speed code (MSB: bit2, LSB:bit0)<br>000 = S100<br>001 = S200<br>010 = S400<br>others = reserved (invalid) |

### 3.1.13 Asynchronous write request for datablock payload(31h)

An instruction that transmits block data using the asynchronous data write command.
It automatically generates the packet format and transmits the packet data by presetting the necessary parameters and data in the register and buffer respectively.

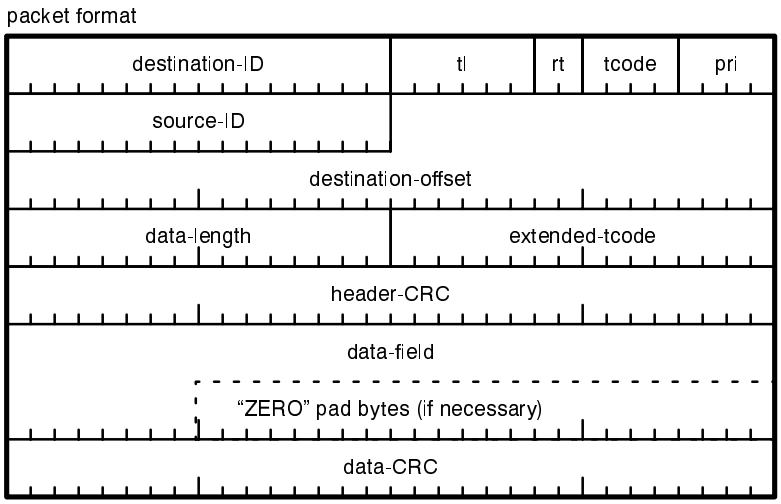
This instruction gets in the arbitration by detecting the arb-reset-gap, generating and transferring the packet, and receiving the acknowledge packet.

Interrupt is reported when the packet is transmitted and the acknowledge is received. When an error occurs, the interrupt is reported and the operation is terminated.

The received acknowledge is stored in the receive-acknowledge display register as mentioned in section 2.5.

The transfer data is stored in the asynchronous transmit-FIFO or transmit/receive-FIFO.

The data-length value does not reach at a quadlet size, zero (0) data is automatically padded until the quadlet size. (In this case, no report is made to the MPU) But the data-length (requested transmit byte count) does not change.




Note: If the destination-ID is set as ”broadcast”, the operation is terminated not waiting for receiving the acknowledge.

| Bit | Operand | Meaning |
|---|---|---|
| 7 to 4 | reserved | Always specify '0' at these bits. |
| 3 | FIFO select | Specify the buffer to store the data:<br>0 = asynchronous transmit-FIFO<br>1 = transmit/receive-FIFO |
| 2 to 0 | speed code | Specify the transmission speed code (MSB: bit2, LSB:bit0)<br>000 = S100<br>001 = S200<br>010 = S400<br>others = reserved (invalid) |

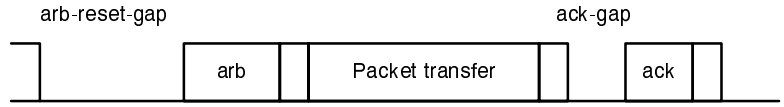
### 3.1.14 Asynchronous write response for split transaction (32h)

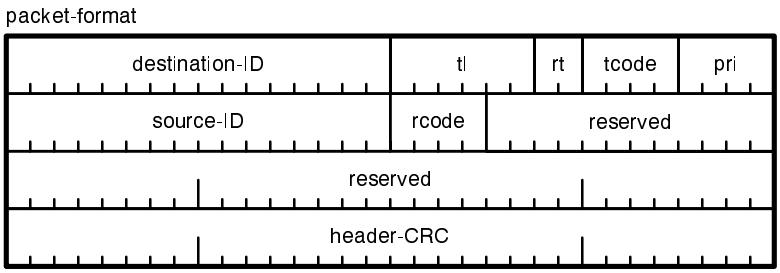
Command for responding to data write request by Asynchronous transfer. It transfers 'ack_pending' acknowledge packet in response to the request received and a packet of split transaction completed.
It automatically generates the packet format and transmits the packet data by presetting the necessary parameters in the register.

This instruction gets in the arbitration by detecting the arb-reset-gap, generating and transferring the packet, and receiving the acknowledge packet.

Interrupt is reported when the packet is transmitted and the acknowledge is received. When an error occurs, the interrupt is reported and the operation is terminated.

Because the parameters necessary for the transfer are all set in the register, the async FIFO for transmission and D-buffer for transmission/receipt are not used in this instruction.
The received acknowledge is stored in the receive-acknowledge display register as mentioned in section 2.5.




| Bit | Operand | Meaning |
|---|---|---|
| 7 to 3 | reserved | Always specify '0' at these bits. |
| 2 to 0 | speed code | Specify the transmission speed code (MSB: bit2, LSB:bit0)<br>000 = S100<br>001 = S200<br>010 = S400<br>others = reserved (invalid) |

### 3.1.15 Asynchronous read request for data quadlet (33h)

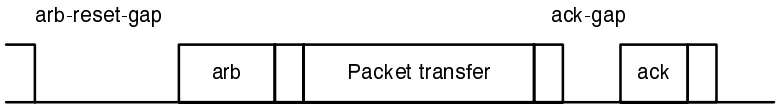
Data read request instruction of 1 quadlet by asynchronous transfer.
This instruction inquiries for reading out the 1 quadlet data. Presetting the transfer parameters (such as destination-ID and transfer offset) in the specified registers automatically generates and transfers the packet.

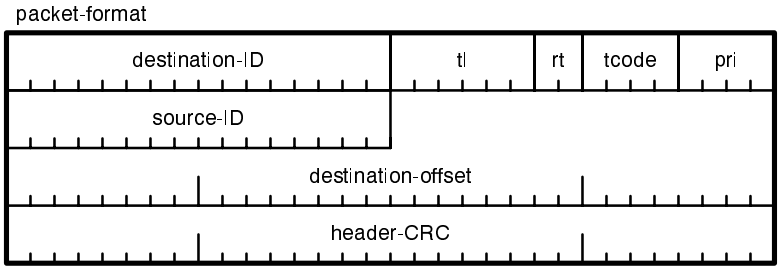
This instruction gets in the arbitration by detecting the arb-reset-gap, generating and transferring the packet, and receiving the acknowledge packet.

Interrupt is reported when the packet is transmitted and the acknowledge is received. When an error occurs, the interrupt is reported and the operation is terminated.

The received acknowledge is stored in the receive-acknowledge display register as mentioned in section 2.5.

Because the parameters necessary for the transfer are all set in the register, the FIFO is not used in this instruction.




| Bit | Operand | Meaning |
|---|---|---|
| 7 to 3 | reserved | Always specify '0' at these bits. |
| 2 to 0 | speed code | Specify the transmission speed code (MSB: bit2, LSB:bit0)<br>000 = S100<br>001 = S200<br>010 = S400<br>others = reserved (invalid) |

### 3.1.16 Asynchronous read request for data block payload(34h)

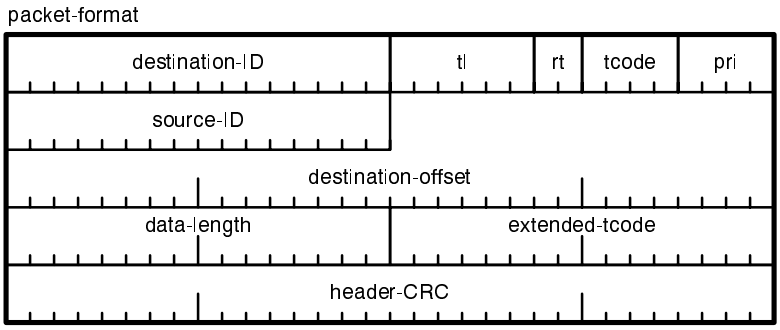
Command for the data request of the data block by asynchronous transfer. It requests the read out of the data block.
Presetting the transfer parameters (such as destination-ID and transfer offset) in the specified registers automatically generates and transfers the packet.

This instruction gets in the arbitration by detecting the arb-reset-gap, generating and transferring the packet, and receiving the acknowledge packet.

Interrupt is reported when the packet is transmitted and the acknowledge is received. When an error occurs, the interrupt is reported and the operation is terminated.

The received acknowledge is stored in the receive-acknowledge display register as described in section 2.5.

Because the parameters necessary for the transfer are all set during the instruction or by the register, the asynchronous transmit-FIFO and data transmit/receive-FIFO are not used in this instruction.




| Bit | Operand | Meaning |
|---|---|---|
| 7 to 3 | reserved | Always specify '0' at these bits. |
| 2 to 0 | speed code | Specify the transmission speed code (MSB: bit2, LSB:bit0)<br>000 = S100<br>001 = S200<br>010 = S400<br>others = reserved (invalid) |

### 3.1.17 Asynchronous readresponse for dataquadlet(35h)

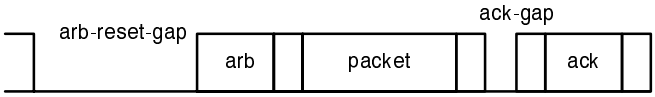
This instruction performs the response to the asynchronous read request. The response data is 1 quadlet. Presetting the transfer parameters (such as destination-ID and transfer offset) in the specified registers automatically generates and transfers the packet.

This instruction gets in the arbitration by detecting the arb-reset-gap, generating and transferring the packet, and receiving the acknowledge packet.

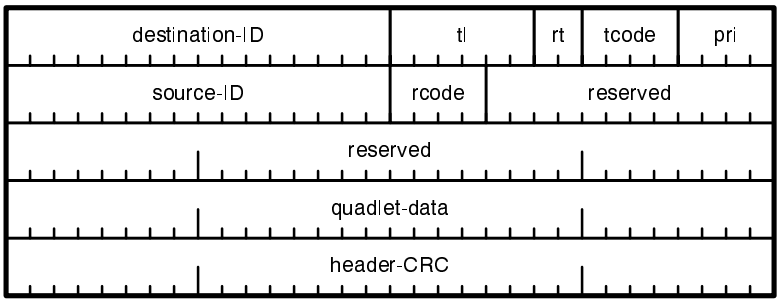
Interrupt is reported when the packet is transmitted and the acknowledge is received. When an error occurs, the interrupt is reported and the operation is terminated.

The received acknowledge is stored in the receive-acknowledge display register as described in section 2.5.

The transmit-response data should be stored in the asynchronous transmission-FIFO or data transmit/receive-FIFO in advance.




Packet format




| Bit | Operand | Meaning |
|---|---|---|
| 7 to 4 | reserved | Always specify '0' at these bits. |
| 3 | FIFO select | Specify the buffer to store the data:<br>0 = asynchronous transmit-FIFO<br>1 = transmit/receive-FIFO |
| 2 to 0 | speed code | Specify the transmission speed code (MSB: bit2, LSB:bit0)<br>000 = S100<br>001 = S200<br>010 = S400<br>others = reserved (invalid) |

### 3.1.18 Asynchronous read response for data block payload(36h)

This instruction performs the response to the asynchronous read request. The response data is data-block. Presetting the transfer parameters (such as destination-ID and transfer offset) in the specified registers automatically generates and transfers the packet.

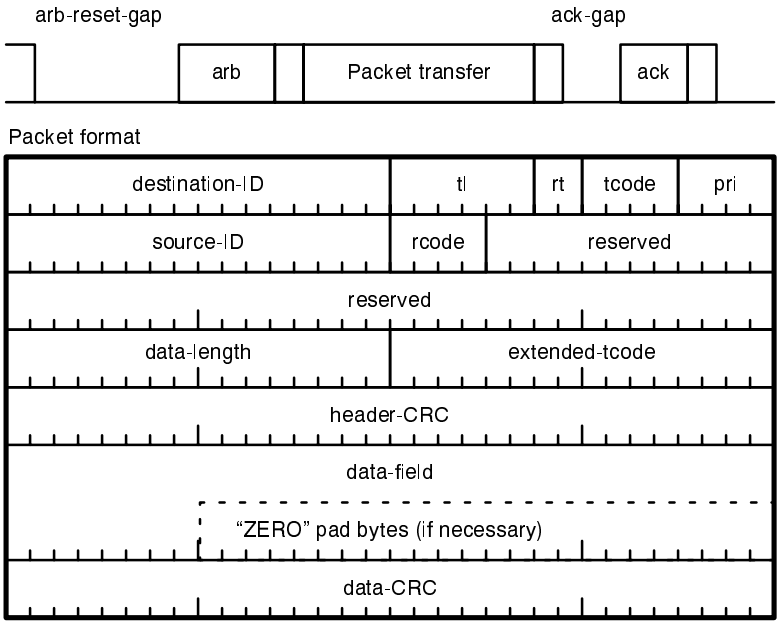
This instruction gets in the arbitration by detecting the arb-reset-gap, generating and transferring the packet, and receiving the acknowledge packet.

Interrupt is reported when the packet is transmitted and the acknowledge is received. When an error occurs, the interrupt is reported and the operation is terminated.

The received acknowledge is stored in the receive-acknowledge display register as described in section 2.5.

The transmit-response data should be stored in the asynchronous transmission-FIFO or data transmit/receive-FIFO in advance.

When the requested transfer byte count does not reach quadlet, it pads '0' data. (No report is made to an host MPU for padding.) Data-length (the requested transfer byte count) itself does not change.




| Bit | Operand | Meaning |
|---|---|---|
| 7 to 4 | reserved | Always specify '0' at these bits. |
| 3 | FIFO select | Specify the buffer to store the data:<br>0 = asynchronous transmit-FIFO<br>1 = transmit/receive-FIFO |
| 2 to 0 | speed code | Specify the transmission speed code (MSB: bit2, LSB:bit0)<br>000 = S100<br>001 = S200<br>010 = S400<br>others = reserved (invalid) |

### 3.1.19 Lock request (37h)

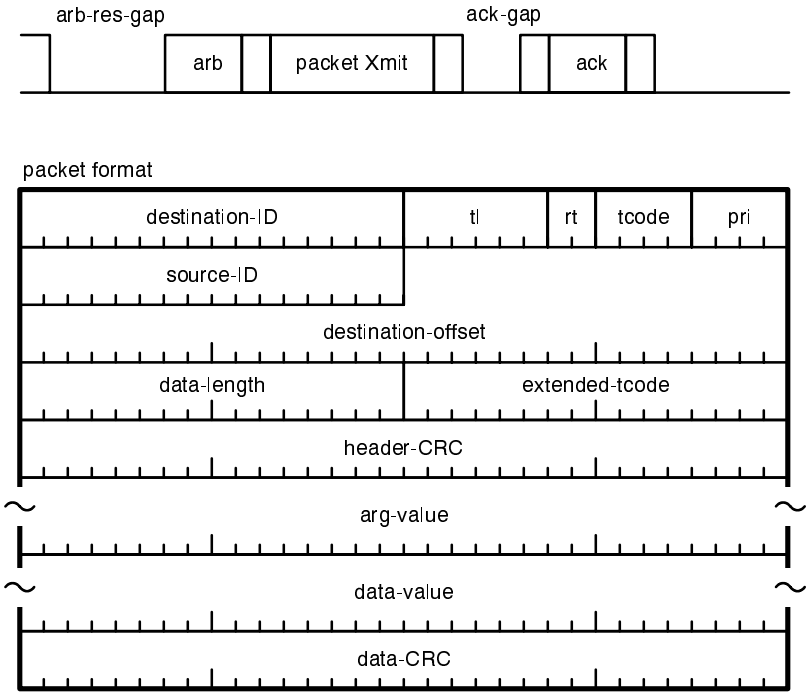
This instruction is used when requesting lock access to a specified destination address.
Presetting the transfer parameters (such as destination-ID and transfer offset) in the specified registers automatically generates and transfers the packet.

This instruction gets in the arbitration by detecting the arb-reset-gap, generating and transferring the packet, and receiving the acknowledge packet.

Interrupt is reported when the packet is transmitted and the acknowledge is received. When an error occurs, the interrupt is reported and the operation is terminated.

The received acknowledge is stored in the receive-acknowledge display register as described in section 2.5.

Please store "arg-value" and "data-value" to be transmitted into the asynchronous transmit-FIFO or transmit/receive-FIFO in advance. (The storage order is 1) "arg-value" and 2) "data-value".)




| Bit | Operand | Meaning |
|---|---|---|
| 7 to 4 | reserved | Always specify '0' at these bits. |
| 3 | FIFO select | Specify the buffer to store the data:<br>0 = asynchronous transmit-FIFO<br>1 = transmit/receive-FIFO |
| 2 to 0 | speed code | Specify the transmission speed code (MSB: bit2, LSB:bit0)<br>000 = S100<br>001 = S200<br>010 = S400<br>others = reserved (invalid) |

### 3.1.20 Lock response (38h)

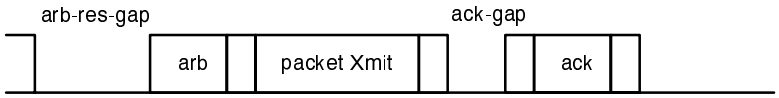
The lock response packet for replying to a lock request packet is transmitted with this instruction.
Presetting the transfer parameters (such as destination-ID and transfer offset) in the specified registers automatically generates and transfers the packet.

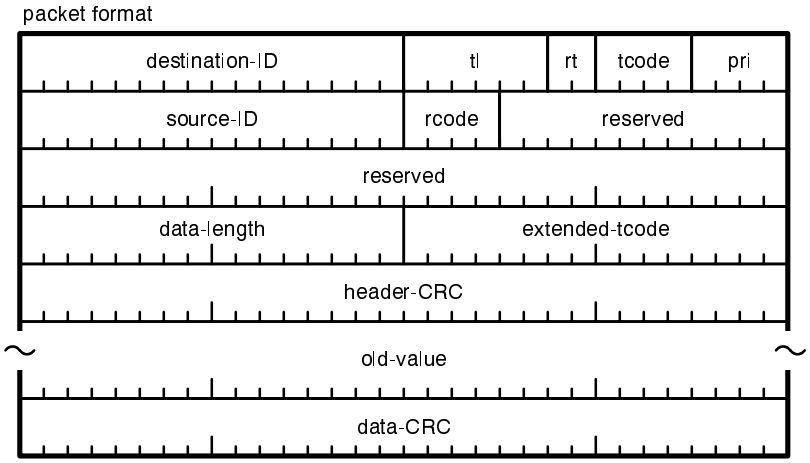
This instruction gets in the arbitration by detecting the arb-reset-gap, generating and transferring the packet, and receiving the acknowledge packet.

Interrupt is reported when the packet is transmitted and the acknowledge is received. When an error occurs, the interrupt is reported and the operation is terminated.

The received acknowledge is stored in the receive-acknowledge display register as described in section 2.5.

Please store "old-value" to be transferred into the asynchronous transmit-FIFO or transmit/receive-FIFO in advance.




| Bit | Operand | Meaning |
|---|---|---|
| 7 to 4 | reserved | Always specify '0' at these bits. |
| 3 | FIFO select | Specify the buffer to store the data:<br>0 = asynchronous transmit-FIFO<br>1 = transmit/receive-FIFO |
| 2 to 0 | speed code | Specify the transmission speed code (MSB: bit2, LSB:bit0)<br>000 = S100<br>001 = S200<br>010 = S400<br>others = reserved (invalid) |

### 3.1.21 Asynchronous chain (39h)

Presetting the transfer parameters (such as destination-ID and transfer offset in the specified registers automatically performs the following cycle until all the data set in the Chain sum data-length setting register (refer to 2.23) are completely transferred.
The chain-dir bit of the mode-control register (refer to 2.1) selects the type of chain transfer (read or write).

• Read chain
1) Generate and transmit 'Read request for data block' packet.
2) Receive 'ack_pending' acknowledge packet.
3) Receive 'Read response for data block' packet and store it in the transmit/receive-FIFO.
4) Transmit 'ack_complete' acknowledge packet.

• Write chain
1) Store the data to be transmitted into the transmit/receive-FIFO.
2) Generate and transmit 'Write request for data block' packet.
3) Receive 'ack_pending' acknowledge packet.
4) Receive 'Write response for data block' packet.
5) Transmit 'ack_complete' acknowledge packet.
The processes 4) and 5) are not necessary when 'ack_complete' acknowledge packet is received at 3).

If the acknowledge packet for request packet is "ack_busy_X" was received, the retry sequence for transmitting the packet requested and intends to transmit it repeatedly according to the 2.20 Chain-retry setting register. If "ack_busy_X" is still received after the retry sequence, the chain transfer stops and reports Relay time out (Chain) interrupt (F6h). In this case, the transmit/receive-FIFO is not cleared.

If an error occurred when receiving the acknowledge/response packet, the chain transfer stops and reports "Fatal error detected (Chain)" interrupt (F7h). In this case, the transmit/receive-FIFO is not cleared.

While chain transfer is executed, the following interrupts are reported. So, please set the interrupt masks in advance.
a) After the acknowledge packet is received:
"Asynchronous packet send" (61 h)
b) After the response packet is received:
"Asynchronous packet received (data buffer FIFO)" (C1 h)
c) After one write transaction is completed (at Write chain):
"One transaction complete (Write chain)" (C3h)
d) After the acknowledge packet is transmitted:
"Acknowledge send" (A1 h)
e) After all the data set in 2.15 Chain sum data-length setting register are completely transferred:
"Chain transaction complete" (C2 h)

Interrupt reports can be switched at the chain transfer of 1 transaction.

Note: The status mode bit (bit 3) of mode control register (refer to 2.1) can switch the interrupt report in the chain transfer of one transction.

When this instruction is issued, some registers must be set properly as follows:
1) mode-control register : B-in='0'
2) Link register #0 : auto-ack='1'

| Bit | Operand | Meaning |
|---|---|---|
| 7 to 3 | reserved | Always specify '0' at these bits. |
| 2 to 0 | speed code | Specify the transmission speed code (MSB: bit2, LSB:bit0)<br>000 = S100<br>001 = S200<br>010 = S400<br>others = reserved (invalid) |

| Status mode | Operation |
|---|---|
| 0 | The 'chain transaction complete' interrupt is not reported at the chain transfer of one transaction.Use the 'asynchronous packet received (data buffer FIFO)' (Ch1) at read chain transfer or 'one transaction complete' (Ch3) at write chain transfer to determine the completion of transfer. |
| 1 | The 'chain transaction complete' interrupt is reported at the chain transfer of one transaction. |

### 3.1.22 Chain suspend (61h)

This instruction suspends the chain transfer in-progress.
It does not clear the FIFO automatically.
Upon the completion of this instruction, the chip reports "Chain stop" (F5h) interrupt.

### 3.1.23 Data-FIFO init (63h)

This instruction clears the contents of a FIFO specified in the operand. The operand for FIFO is defined as follows:

| Bit | Operand | Meaning |
|---|---|---|
| 7 to 0 | FIFO select code | Specify the FIFO (MSB: bit7, LSB:bit0)<br>11h = Asynchronous Receive-FIFO<br>12h = Asynchronous Transmit-FIFO<br>21h = Transmit/Receive-FIFO<br>others = reserved (invalid) |

### 3.1.24 DMA Tx (71h)

This instruction activates the DMA interface to store a transmit-asynchronous packet in the transmit/receive-FIFO.
When 'B-sel' bit of mode control register is set to '0', this instruction keeps asserting the DREQ signal until all the data as specified in 2.10 "Xmit-asynchronous data-length setting register" are written in the FIFO.

### 3.1.25 DMA Rx (72h)

This instruction activates the DMA interface to read out a transmit-asynchronous packet from the transmit/receive-FIFO.
This instruction keeps asserting the DREQ signal until all the data of 1 packet stored in the FIFO is read out.

### 3.1.26 DMA start (Chain) (73h)

This instruction is used for the chain transfer in packet-per-packet via the DMA transfer.

When '0' is set at the Chain-dir bit (bit 11) of mode-control register (address 00h), the data of received 'read response for data block' packet is transferred.
This instruction activates the DMA transfer every time when 1 packet of the 'read response for data block' is stored in the transmit-receive-FIFO. It keeps asserting the DREQ signal until all the data of 1 packet stored in the FIFO is read out.

When '1' is set at the Chain-dir bit (bit 11) of mode-control register (address 00h), the data of 'write request for data block' packet to be transmitted is transferred.
This instruction activates the DMA transfer every time when the space for 1 packet of the 'write request for data block' is available. It keeps asserting the DREQ signal until the transmit-receive-FIFO stores the data for 1 packet.

This instruction must be issued prior to the "asynchronous chain" instruction (39h).

This instruction also requires the category M in interrupt masking to be masked, and setting of the Chain sum data-length setting register (refer to 2.15).

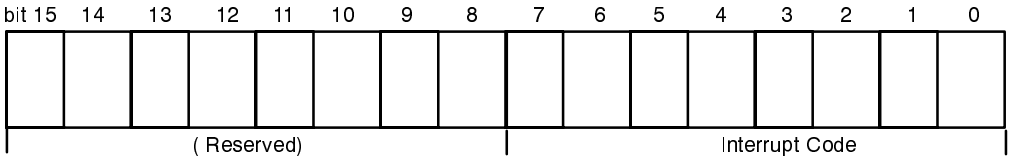
# 4. INTERRUPT

This chapter will explain the interrupt registers, interrupt masks and interrupt codes.

## 4.1 Interrupt Register and Interrupt Mask Register

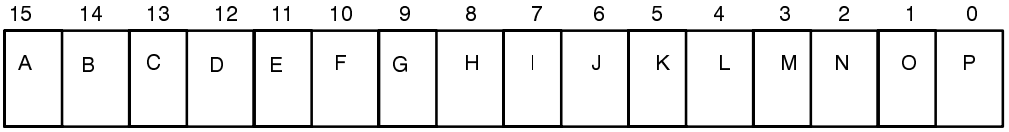
The interrupt code is indicted on the register. The masked code with the interrupt mask register is not displayed. (not reflected with INT signal pin either)
The register is a FIFO-type and stores up to 8 codes. Once the number of interrupts exceeds 8, the code is overflowed and the overflown interrupt is discarded.

- Interrupt Register :




This register masks the interrupt which reported by this device in accordance with the categories shown below.
When '1' is set to the corresponding bit to each category, the above listed interrupt is masked.

- Interrupt Mask Register :




Cat. A : Bus Reset Process
Cat. B : Bus Error
Cat. C : Asynchronous Packet Received
Cat. D : Asynchronous Packet Receive Error
Cat. E : (Reserved)
Cat. F : (Reserved)
Cat. G : Asynchronous Packet Sent
Cat. H : (Reserved)
Cat. I : (Reserved)
Cat. J : Acknowledge Packet Receive Error
Cat. K : Acknowledge Packet Sent
Cat. L : Cycle Start Packet
Cat. M : Asynchronous Packet Receive (data buffer)
Cat. N : PHY Packet Sent
Cat. O : PHY Packet Received
Cat. P : Instruction Check / Response

## 4.2 List of Interrupt Codes & Interrupt Mask

The list below shows the interrupt codes the MB86614A reports and categories of interrupt masks.

| Nameof Interrupt | Code | Mask Category |
|---|---|---|
| Protocol Error | 00 | A |
| Loop Detected | 02 | |
| Self-ID packet error | 04 | |
| Bus Reset Completed | 05 | |
| Bus Reset Detected | 06 | |
| Bias Change Detected | 07 | |
| Isochronous Cycle Too Long | 13 | B |
| Bus Occupancy Violation | 14 | |
| Data Buffer FIFO Full | 17 | |
| Asynchronous Packet Received (asynch FIFO) | 21 | C |
| Asynchronous Packet Received (no data) | 22 | |
| Asynchronous Packet Received (destination_offset="FFFFF0001008h") | 23 | |
| Asynchronous Packet Received (destination_offset="FFFFF0001010h") | 24 | |
| Data Length Short Error (Asynchronous) | 31 | D |
| Data Length Long Error (Asynchronous) | 32 | |
| Packet Format Error (Asynchronous) | 33 | |
| Data End Not Found (Asynchronous) | 34 | |
| Header CRC Error (Asynchronous) | 35 | |
| Data CRC Error (Asynchronous) | 36 | |
| Data Length Invalid (Asynchronous) | 37 | |
| Asynchronous Packet Sent | 61 | G |
| Acknowledge Missing | 91 | J |
| Acknowledge Sent | A1 | K |
| Cycle Start Packet Received | B1 | L |
| Cycle Start Packet Sent | B2 | |

| Nameof Interrupt | Code | Mask Category |
|---|---|---|
| Asynchronous Packet Received (data buffer FIFO) | C1 | M |
| Chain Transaction Completed | C2 | - |
| One Transaction Completed (write chain) | C3 | M |
| Physical Configuration Packet Sent | D1 | N |
| Link On Packet Sent | D2 | |
| Ping Packet Sent | D3 | |
| Physical Configuration Packet Received | E1 | O |
| Link On Packet Received | E2 | |
| Self-ID Packet Received | E3 | |
| Busy Mode Started | F1 | P |
| Busy Mode Stopped | F2 | |
| Instruction Abort (send packet data-length) | F3 | |
| Instruction Abort (state invalid) | F4 | |
| Chain Stopped | F5 | |
| Retry Timeout (Chain) | F6 | |
| Fatal Error Detected (Chain) | F7 | |

## 4.3 Interrupt Description

This section describes each interrupt code per its category.

**A) Bus Reset Process**

| Code | Interrupt | Description |
|---|---|---|
| 00 | Protocol Error | An error which is not applied to any interrupt category occurred. |
| 02 | Loop Detected | Loop state was detected on the topology.'BUS RESET' is not terminated normally. |
| 04 | Self-ID Packet Error | There was an error or spec violation such as PHY ID was not counted up correctly when self-ID packet was received in self-identity process and so on. After reporting this error, self-ID packet is continuously received.<br><br>There was a logical inversion error while receiving a self-ID packet after transmitting a ping packet in normal transfer mode. In this case, the received packet is discarded. |
| 05 | Bus Reset Complete | The device completed the bus reset process.<br>Check if other error interrupt occurs or not. |
| 06 | Bus Reset Detected | Bus reset was detected from another node.<br>Or, Bus reset was activated by receiving a bus reset instruction. |
| 07 | Bias Change Detected | The chip detected that bus state on a cable port has changed to off state to on state. The chip activates the bus reset process automatically. |

**B) Bus Error**

| Code | Interrupt | Description |
|---|---|---|
| 13 | Isochronous Cycle Too Long | Isochronous cycle exceeded the specified time.<br>This interrupt is reported only when the device is a cycle master. |
| 14 | Bus Occupancy Violation | The node occupied the bus longer than the specified time (max_data_time). |
| 17 | Data Buffer FIFO Full | Because all the banks in transmit/receive-FIFO contained data and no more packet can be received. |

**(C) Asynchronous PacketReceived**

| Code | Interrupt | Description |
|---|---|---|
| 21 | Asynchronous Packet Received (asynchronous FIFO) | Asynchronous packet to the self-node was correctly received in asynchronous receive-FIFO. The data field for 'write request packet for quadlet' and 'read response packet for quadlet' does not exist for the reason of packet configuration. The interrupt for them are reported with this interrupt since the data is stored into the Async FIFO. Accessing the receive-asynchronous header display register finds the contents of received packet. |
| 22 | Asynchronous Packet Received (no data) | Asynchronous packet with a format not containing data to the self-node was correctly received. Accessing the receive-asynchronous header display register finds the contents of received packet. |
| 23 | Asynchronous packet received (destination_offset = "FFFFF0001008h") | This interrupt is reported when the Asynchronous packet to the self-node with Destinaon_offset 'FFFF_F000_1008h' is received. |
| 24 | Asynchronous packet received (destination_offset = "FFFFF0001010h") | This interrupt is reported when the Asynchronous packet to the self-node with Destinaon_offset 'FFFF_F000_1010h' is received. |

**(D) Asynchronous Packet Receive Error**

| Code | Interrupt | Description |
|---|---|---|
| 31 | Data Length Short Error | The data count on the received packet was shorter than the data-length value for the packet header.<br>The chip pads '0' or '1' data into the data according to the pad bit setting bit of mode-control register and stores the padded data in the FIFO. |
| 32 | Data Length Long Error | The data count on the received packet was longer than the data-length value for the packet header.<br>The data count that exceeds the data-length value is not stored in the FIFO. |
| 33 | Packet Format Error | There was a format error on the received packet. Or, there was a spec violation such as "reserved field was not "0" etc.<br>The received packet is discarded. |
| 34 | Data End Not Fount | Data End section did not come to the end of received packet.<br>If the header section is being received, the received packet is discarded. If the data section is being received, the same process as for Interrupt codes, 31 or32, is taken. |
| 35 | Header CRC Error | There was a header CRC error on the received packet.<br>The received packet is discarded. |
| 36 | Data CRC Error | There was a data CRC error on the received packet.<br>The received packet is not discarded. |
| 37 | Data Length Invalid | Although an asynchronous packet that should be stored in the FIFO was received, the data-length was 129-byte or more.<br>Or, although an packet was received and stored in the transmit/receive-FIFO, the data-length was bigger than the size of 1 bank of FIFO.<br>The received packet is discarded.<br>When acknowledge is sent with the asynchronous packet, 'ack_busy_x' is sent. |

**(G) Asynchronous Packet Sent**

| Code | Interrupt | Description |
|---|---|---|
| 61 | Asynchronous Packet Sent | Sending the asynchronous packet by the instruction or the execution of chain transfer was completed. |

**(J) AcknowledgePacketReceiveError**

| Code | Interrupt | Description |
|---|---|---|
| 91 | Acknowledge Missing | The acknowledge packet was not received within the specified time in response to the non-broadcasted asynchronous packet sent out from the self-node. |

**(K) AcknowledgePacketSent**

| Code | Interrupt | Description |
|---|---|---|
| A1 | Acknowledge Sent | The acknowledge packet was sent out. |

**(L) CycleStart Packet**

| Code | Interrupt | Description |
|---|---|---|
| B1 | Cycle Start Packet Received | When the self-node was not a root, it received the cycle start packet correctly.<br>After that, the isochronous cycle activates. Also, at the same time as this interrupt the ISO cycle bit in flag&status register is set to ”1”. |
| B2 | Cycle Start Packet Sent | The self-node was a root and it sent out the cycle start packet. |

**(M) Asynchronous PacketReceived(databuffer)**

| Code | Interrupt | Description |
|---|---|---|
| C1 | Asynchronous Packet Received (data buffer FIFO) | Asynchronous packet to the self-node was correctly received in the transmit/receive-FIFO. Accessing the receive-asynchronous header display register finds the contents of received packet. |
| C2 | Chain Transaction Complete | While the chain transfer is operating, the last 'read/write transaction' was correctly completed. In this case, 'Asynchronous packet received (data buffer FIFO)' interrupt (C1) is not reported. When executing the chain trabsfer of 1 transaction, the output status for this interrupt is selectable. Refer to the bit 3 of mode control register in 2.1. Further, this interruption cannot be masked with interrupt mask register. |
| C3 | One Transaction Complete (write chain) | 'Acknowledge' of 'ack_complete' was received while 'write chain' is executed.<br>'write response' packet was received and corresponding 'acknowledge' is sent.<br>This interruption is generated for each write transaction. |

**(N) PHY PacketSent**

| Code | Interrupt | Description |
|---|---|---|
| D1 | Physical Configuration Packet Sent | The physical configuration packet was sent out. |
| D2 | Link on Packet Sent | The link on packet was sent out. |
| D3 | Ping Packet Sent | The ping packet was sent out. |

**(O) PHY PacketReceived**

| Code | Interrupt | Description |
|---|---|---|
| E1 | Physical Configuration Packet Received | The physical configuration packet was received correctly.<br><br>The result was reflected to 2.24.2. Physical register #1 and the device operation automatically switches to the specified mode. |
| E2 | Link on Packet Received | Link-on packet to the self node was received correctly.<br>In this case, LINKON pin output is also asserted. |
| E3 | Self-ID Packet Received | Self-ID packet was correctly received after transmitting the ping packet.<br>The received packet is stored in the asynchronous receive-FIFO. |

**(P) Instruction Check/Response**

| Code | Interrupt | Description |
|---|---|---|
| F1 | Busy Mode Started | The chip entered the busy mode upon the start busy mode instruction (03h) received. Or, the chip entered the busy mode upon an asynchronous packet received correctly. The chip transmits 'ack_busy_X' acknowledge when the asynchronous packet to the self node was received. |
| F2 | Busy Mode Stopped | The chip was released from the busy mode by the remove busy mode instruction (04h). |
| F3 | Instruction Abort (send packet data-length) | Data corresponding to the packet's data-length were not stored in FIFO when transmitting the packet upon the instruction. |
| F4 | Instruction Abort (state) | The instruction was issued but, since it was an improper invalid instruction for the device condition, it was abort. For example, this result includes the following actions:<br>1) The instruction issued was undefined.<br>2) Although the device was in sleep state, unacceptable instruction like 'remove sleep' (02h) has been issued.<br>3) The instruction issued contained an undefined operand code. |
| F5 | Chain Stopped | Chain transfer stopped by the receipt of "Chain suspend" instruction (61h) or assertion of EOP pin input. |
| F6 | Retry Timeout (Chain) | Chain transfer for 'read request data block' packet was terminated because it retried as specified in the 2.20 Chain Retry Limit setting register but 'ack_busy_X' was still received. |
| F7 | Fatal Error Detected (Chain) | Chain transfer was terminated because a header CRC error, 'ack_missing' state, or bus reset was detected. |

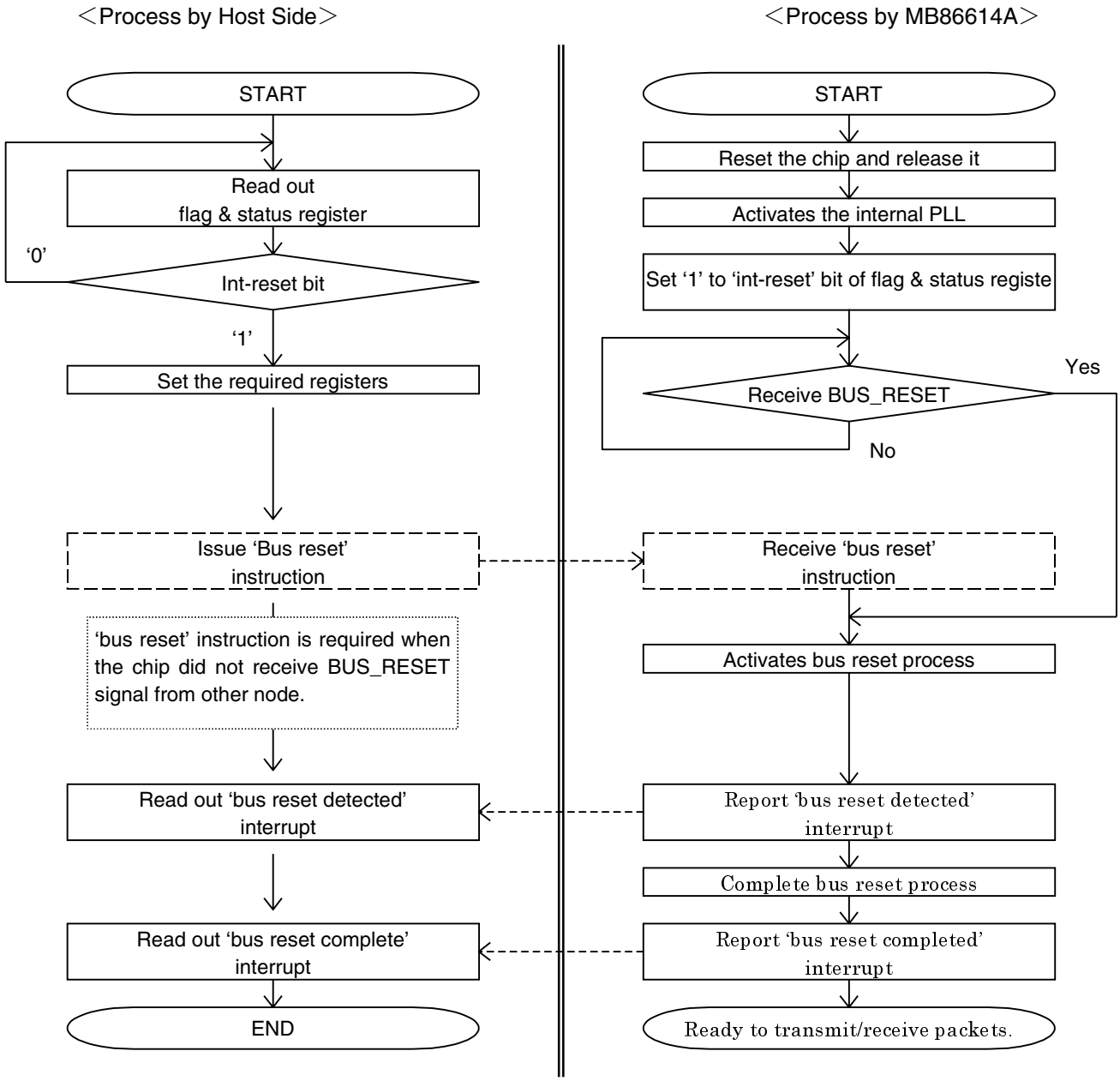
# 5. OPERATION

This chapter will explain the operations of MB86614A.

5.1 Device Initialization under System Powered Operation
5.2 Sleep Operation
5.3 Self-ID Packet Receive
5.4 Asynchronous Packet Transmit and Receive
5.5 Automatic tcode Insertion
5.6 Device Operation under 1394 Cable Power

## 5.1 Device Initialization under System Powered Operation

Before the system is turned OFF, the device does not operate with cable power supply.




Note: This chip automatically executes BUS_RESET when it recognizes the connection at the 1394 port. BUS_RESET, therefore, is completed without issuing an instruction. The flow chart above includes BUS_RESET instruction because the initial value has been set to the internal register.

## 5.2 Sleep Operation

The MB86614A has two types of sleep mode, forced-sleep and 1394 port auto-sleep.

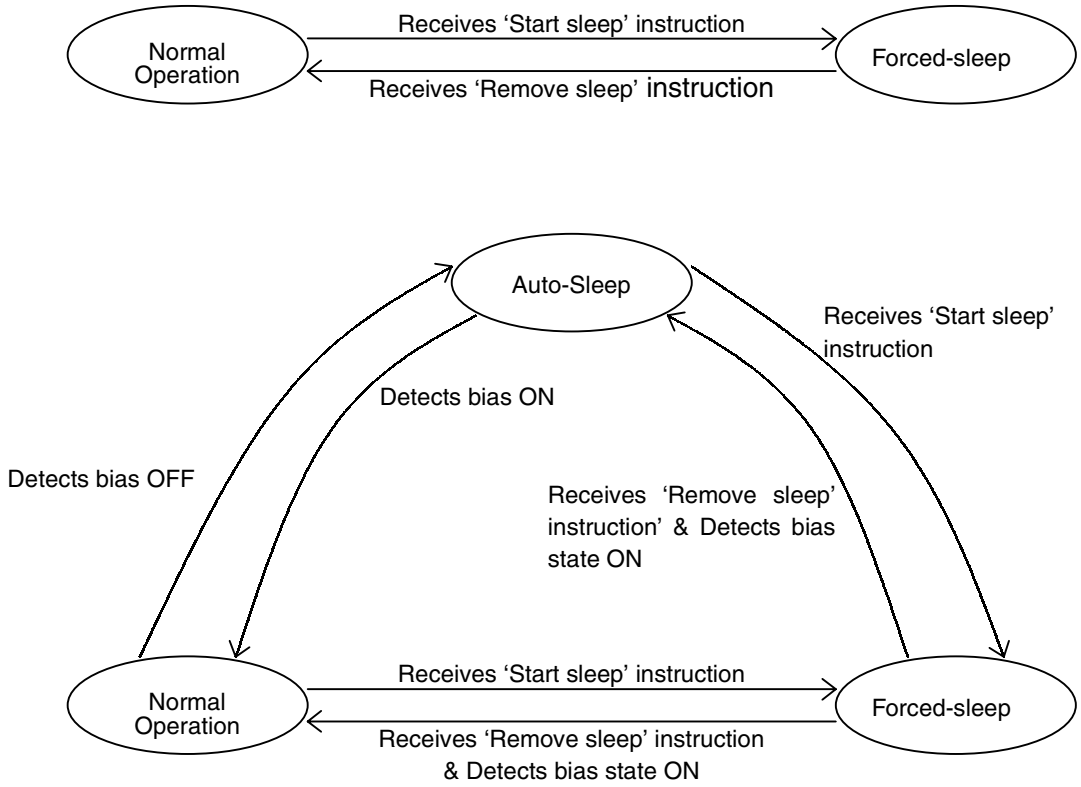
5.2.1 Forced-Sleep
5.2.2 1394 Port Auto-Sleep

### 5.2.1 Forced-Sleep

This sleep mode is activated by the instruction from a MPU (“start sleep”). The only operation is the internal register access and both Link and PHY blocks completely stop the operations. The application layer operates in 24.576 MHz clock. Therefore, the quick access becomes impossible since the Host I/F needs to operate in its half frequency. The 1394 port turned to be “powered-off” status that means the port disconnected electrically. The internal PLL also stops the operation except for a circuit to release from the forced-sleep state. This sleep state can be released with the instruction (“remove-sleep”). Whether or not the device is in sleep state can be referred by the “sleep” bit (bit 4 of flag & status register).

### 5.2.2 1394 Port Auto-Sleep

The MB8661A device watches each port connection and makes the disconnected port “sleep” automatically. The driver and comparator on the disconnected port stop their operations. (Only the bias detection circuit operates.)
The other circuits of PHY layer operate even in this sleep mode. The sleeping circuits automatically wake up by the detection of bias.



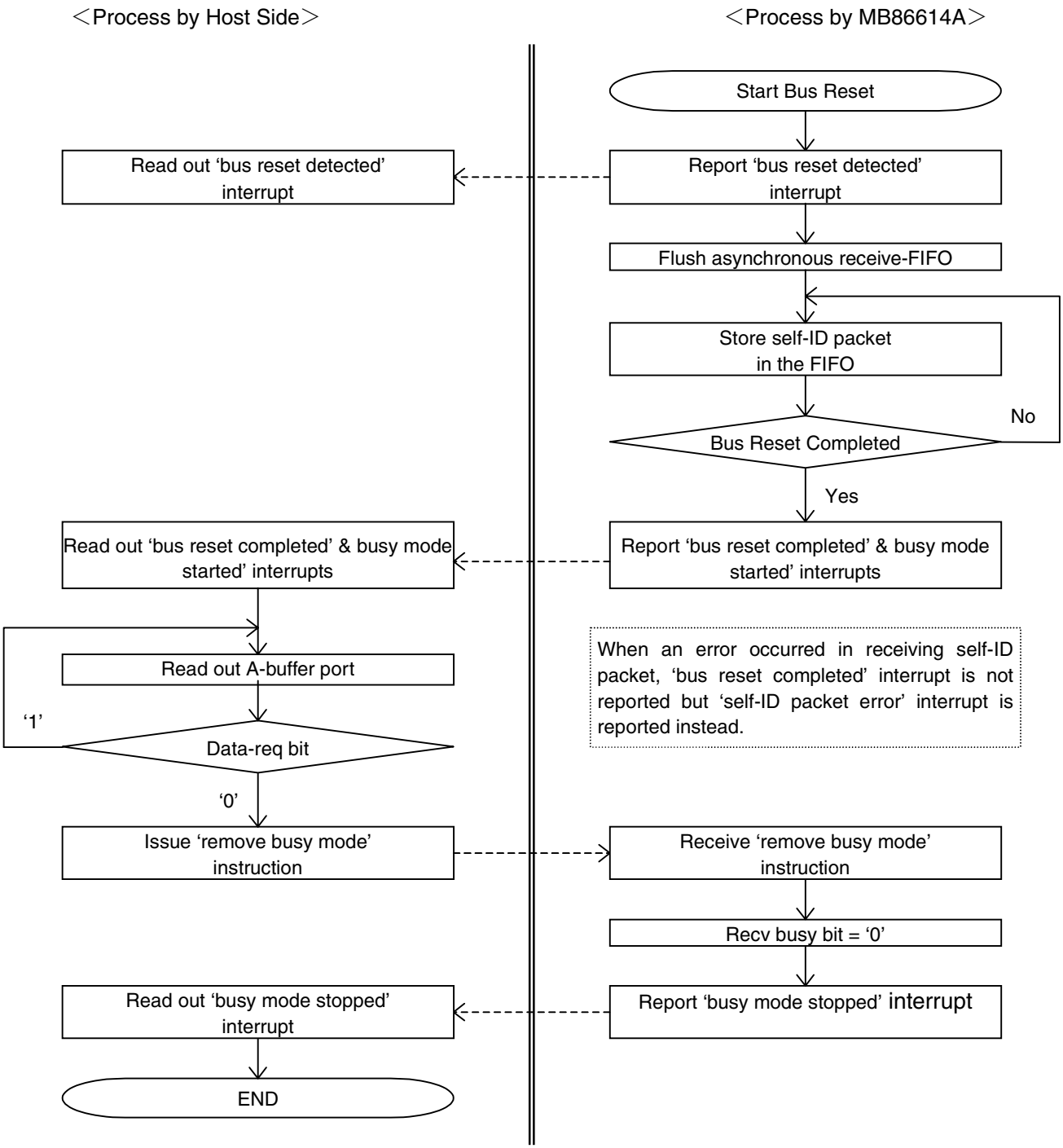
## 5.3 Self-ID Packet Receive

This section will explain the receiving operation of self-ID packet.

5.3.1 Self-ID Packet Receive during Bus Reset Process
5.3.2 Self-ID Packet Receive after Ping Packet Transmit

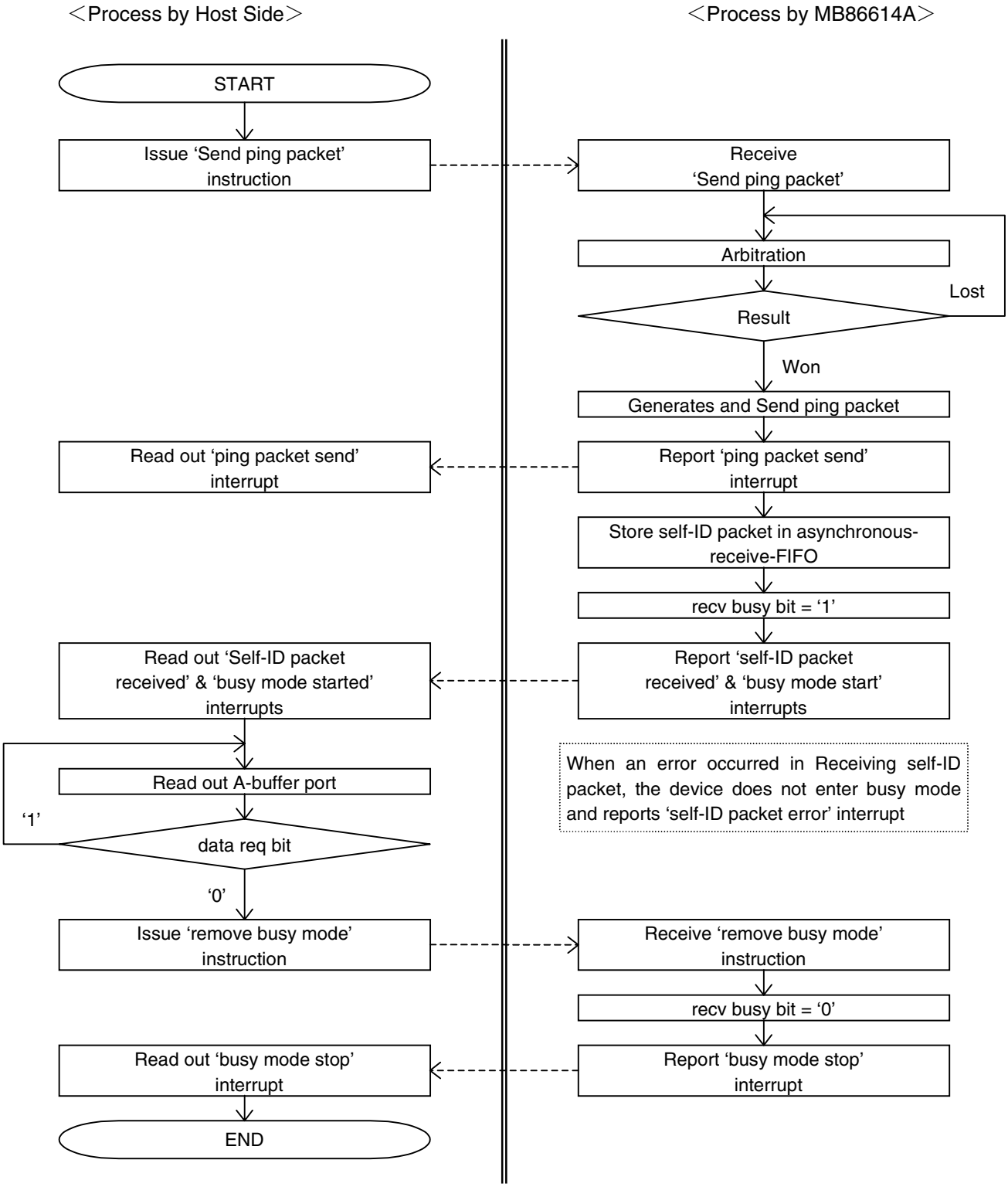
### 5.3.1 Self-ID Packet Receive during Bus Reset Process

The MB86614A device is capable of receiving self-ID packets that each node transmits in the self-identity stage of bus reset process. When '1' is written to the s-ID store bit of mode-control register, mainly the received packet can be stored in the FIFO (and the data removing the logical inverse section is stored in the asynchronous receve-FIFO). When the number of total data exceeds 128 bytes, the overflown data are discarded.
If a Bus reset occurred before the self_ID packets are read out, the packets are discarded even the device is in busy mode, as well as a received data of asynchronous packet.



### 5.3.2 Self-ID Packet Receive after Ping Packet Transmit

Regardless of ‘s-ID store’ bit setting, the device receives self-ID packet after a ping packet transmitted and stores the data removing logical inverse section in the asynchronous receive-FIFO.



## 5.4 Asynchronous Packet Transmit and Receive

This section will explain the transmit/receive operation of asynchronous packet.

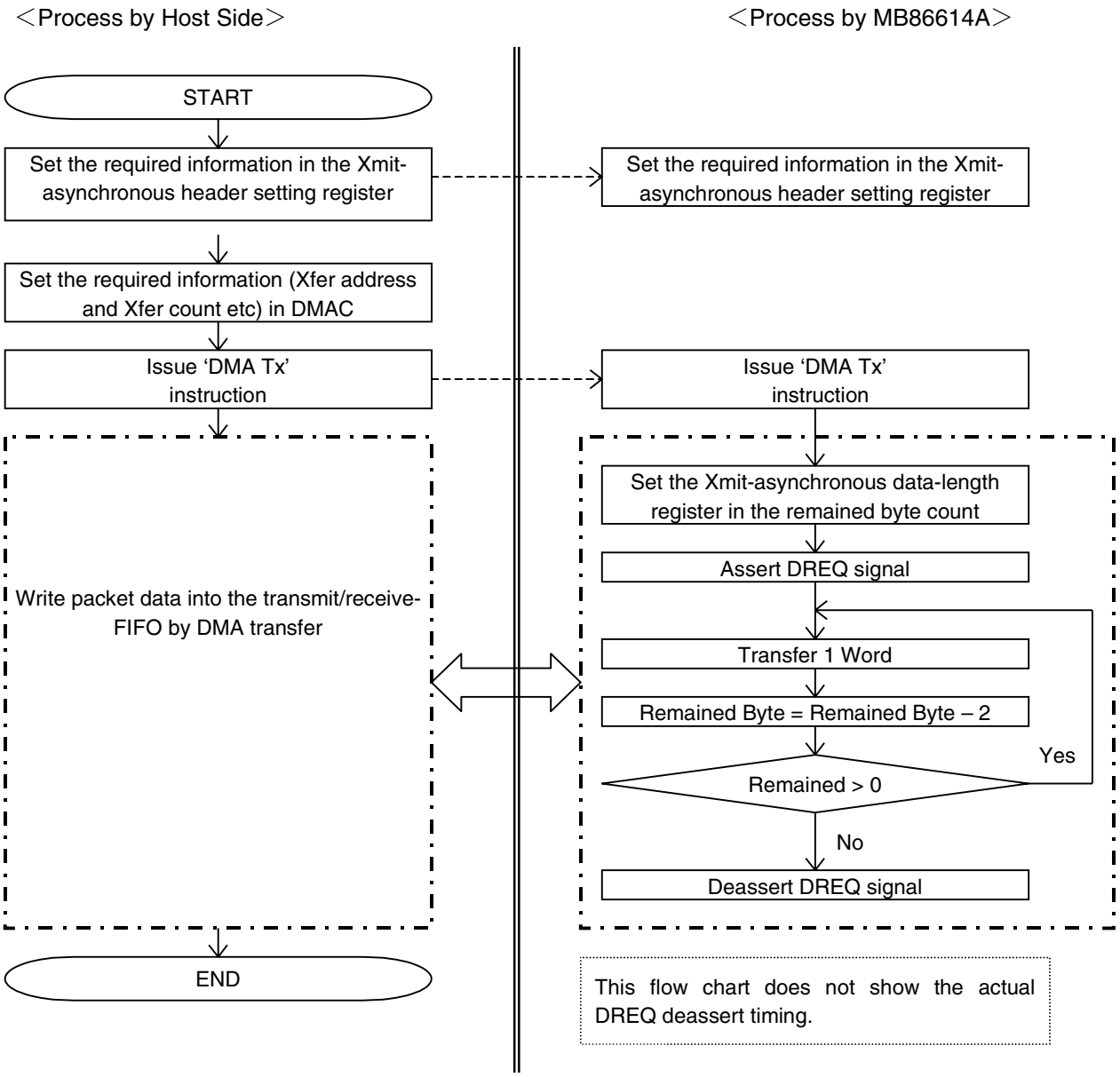
5.4.1 Asynchronous Packet Transmit (using the transmit/receive-FIFO)
5.4.2 Asynchronous Packet Transmit (using the asynchronous transmit-FIFO)
5.4.3 Asynchronous Packet Transmit without data
5.4.4 Chain transfer
5.4.5 Asynchronous packet Receive

### 5.4.1 Asynchronous Packet Transmit (using the transmit/receive-FIFO)

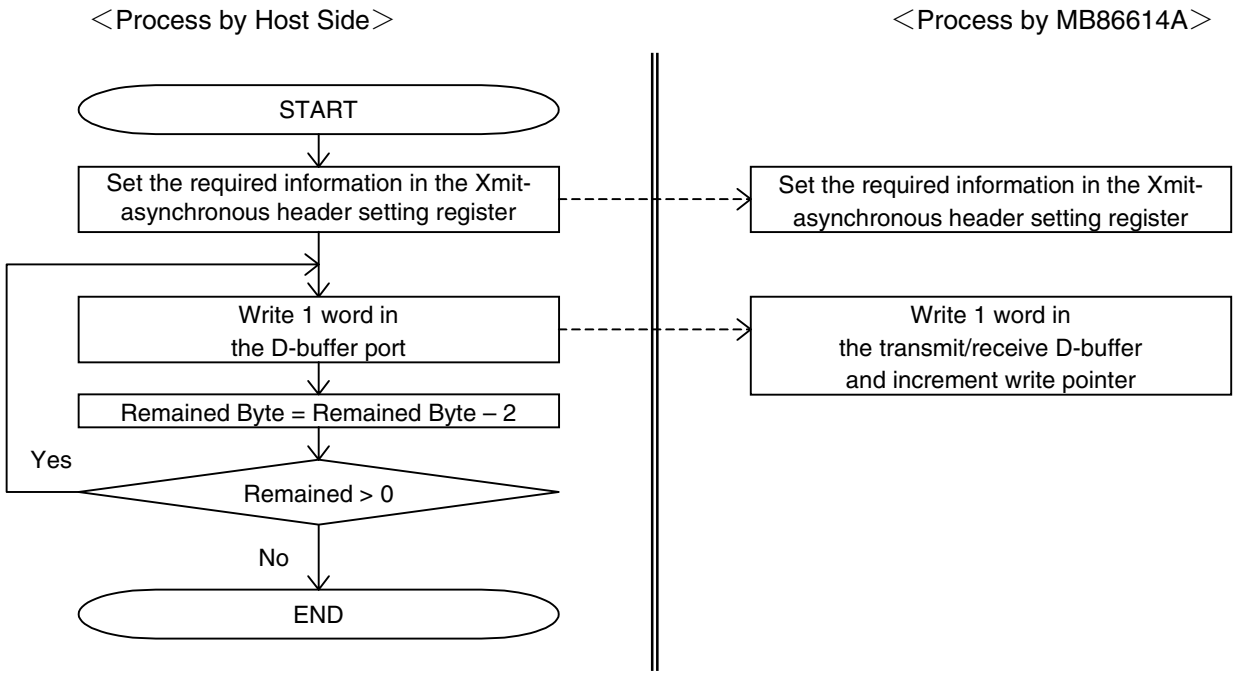
This section will explain the transmitting operation of asynchronous packet with the transmit/receive-FIFO.

5.4.1.1 Flow chart for operation until data is stored in the transmit/receive-FIFO with the DMA transfer
5.4.1.2 Flow chart for operation until data is stored in the transmit/receive-FIFO with the MPU access
5.4.1.3 Flow chart for operation after data is stored in the transmit/receive-FIFO

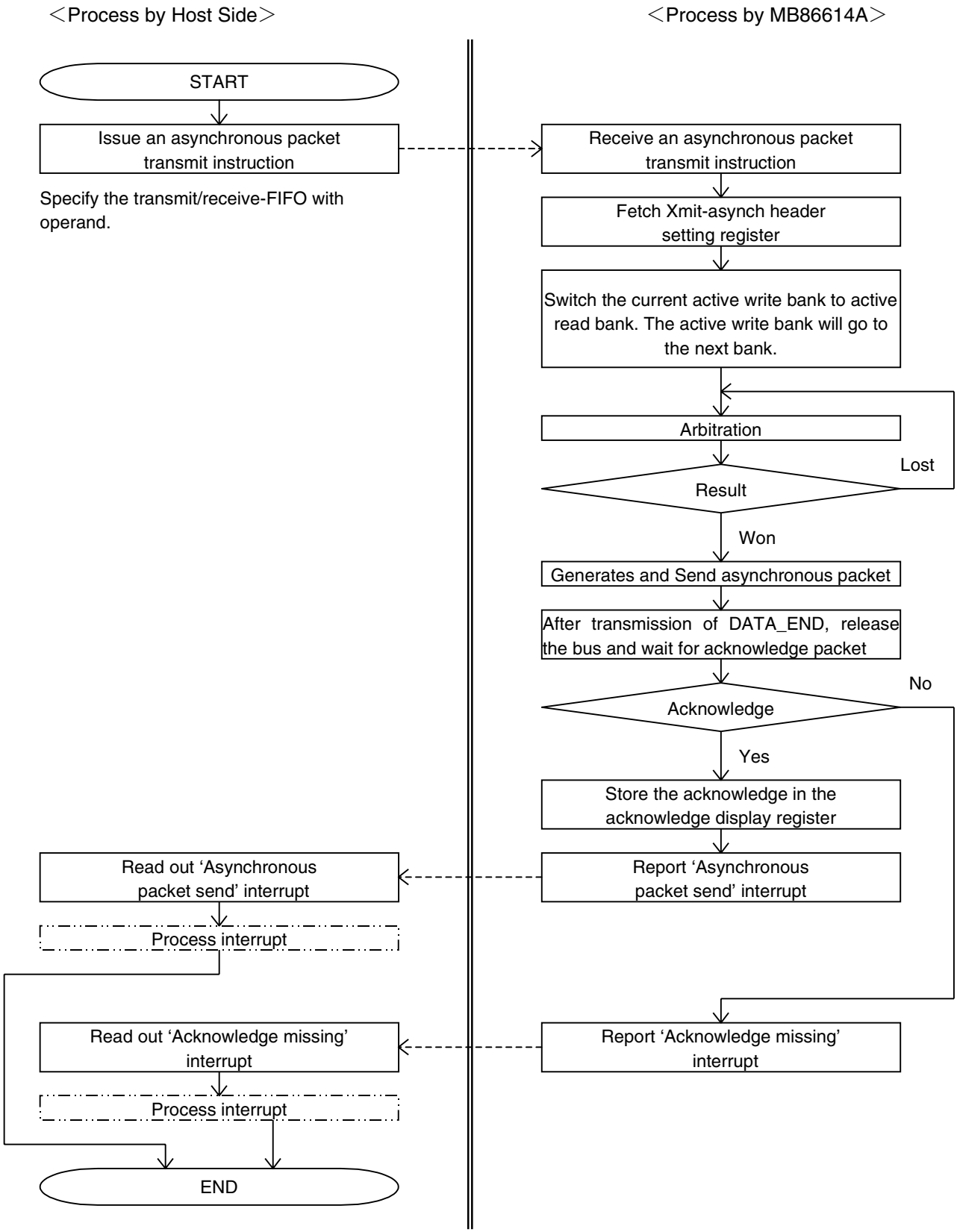
#### 5.4.1.1 Flow chart for operation until data is stored in the transmit/receive-FIFO with the DMA transfer



#### 5.4.1.2 Flow chart for operation until data is stored in the transmit/receive-FIFO with the MPU access



### 5.4.1.3 Flow chart for operation after data is stored in the transmit/receive-FIFO

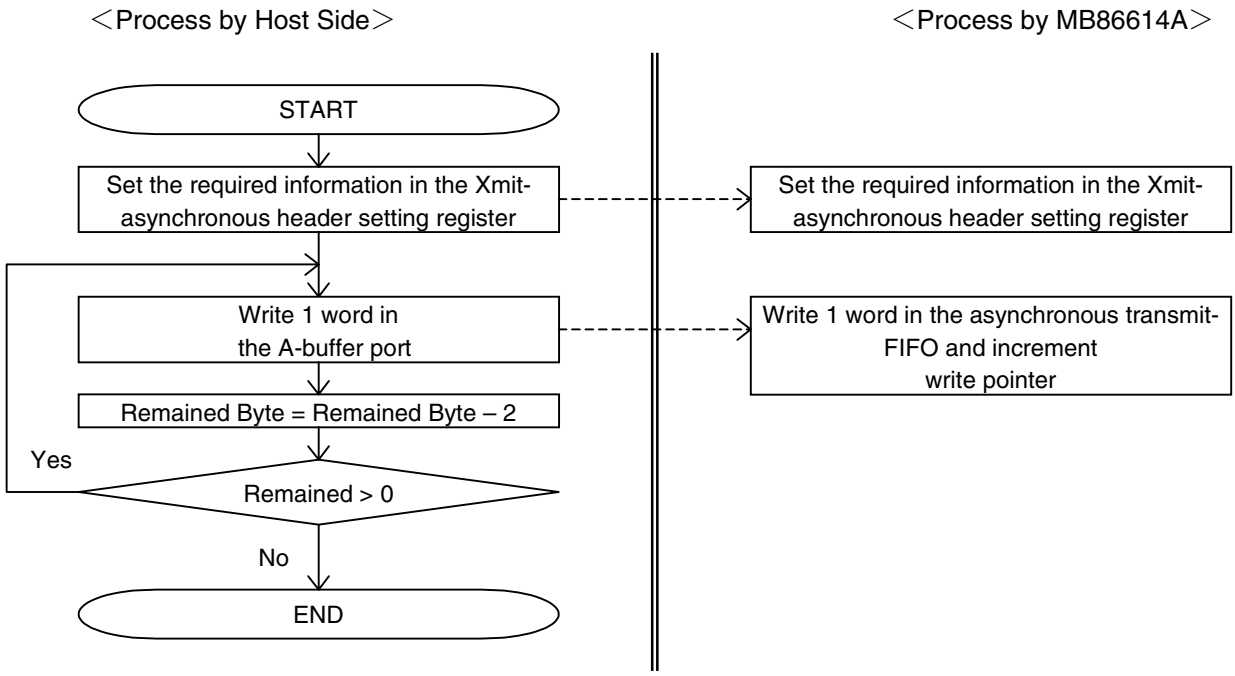


### 5.4.2 Asynchronous Packet Transmit (using the asynchronous transmit-FIFO)

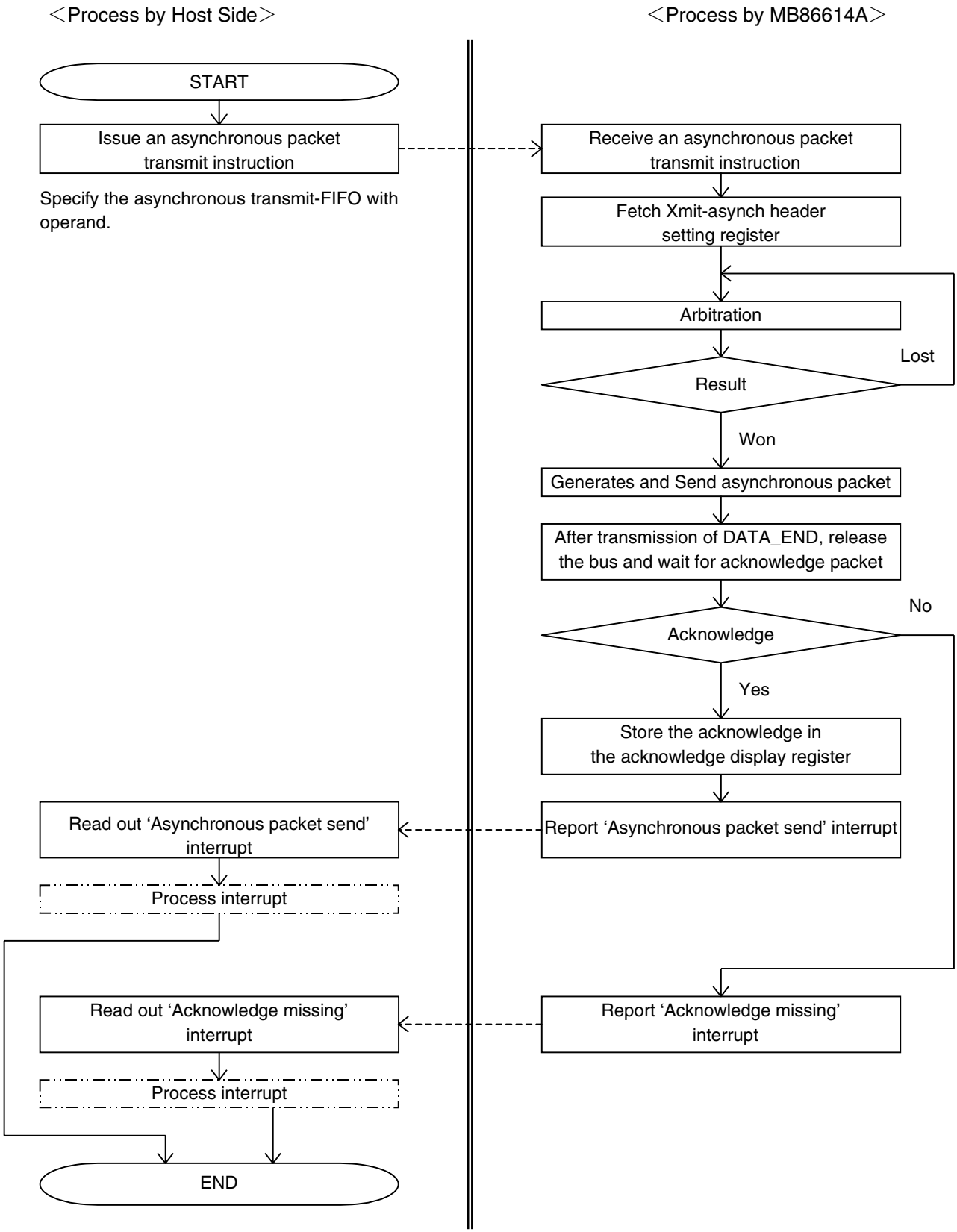
This section will explain the transmitting operation of asynchronous packet with the asynchronous-FIFO.

5.4.2.1 Flow chart for operation until data is stored in the asynchronous-FIFO
5.4.2.2 Flow chart for operation after data is stored in the asynchronous-FIFO

#### 5.4.2.1 Flow chart for operation until data is stored in the asynchronous-FIFO



### 5.4.2.2 Flow chart for operation after data is stored in the asynchronous-FIFO

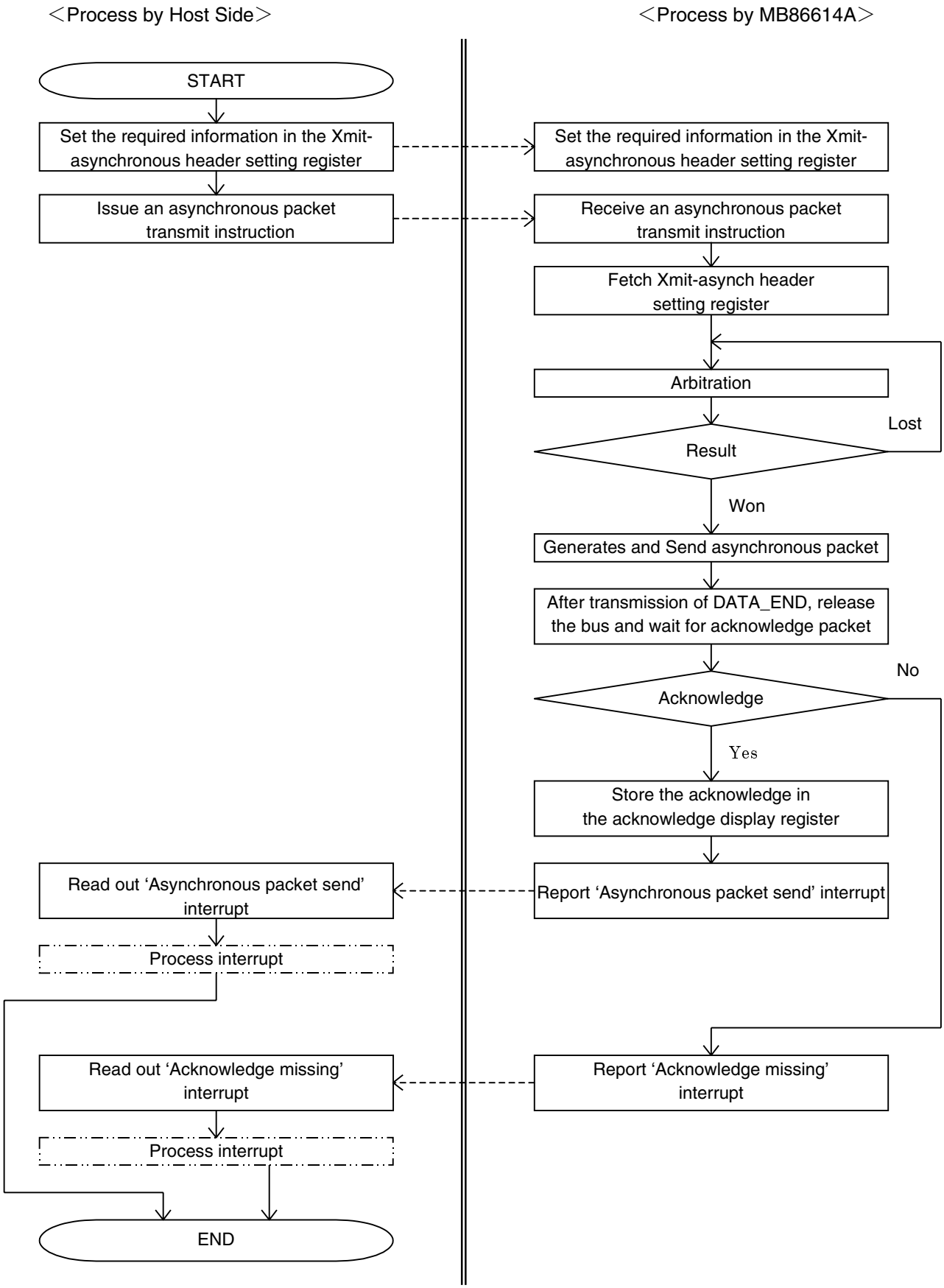


### 5.4.3 Asynchronous Packet Transmit without data

This section will explain the transmitting operation of asynchronous packet with no data.

5.4.3.1 Flow chart for Asynchronous Packet Transmit Operation with no data

### 5.4.3.1 Flow chart for Asynchronous Packet Transmit Operation with no data

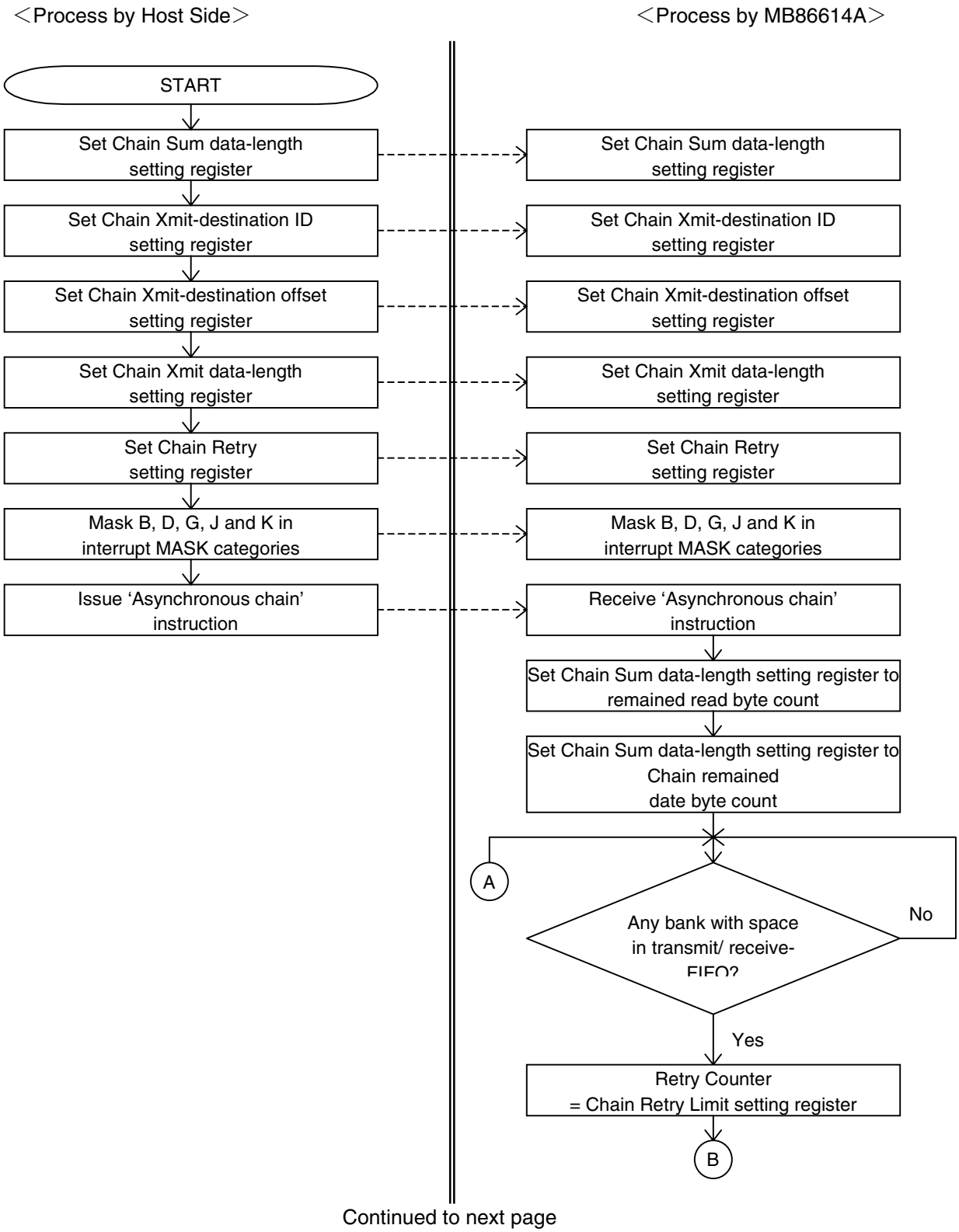


### 5.4.4 Chain Transfer

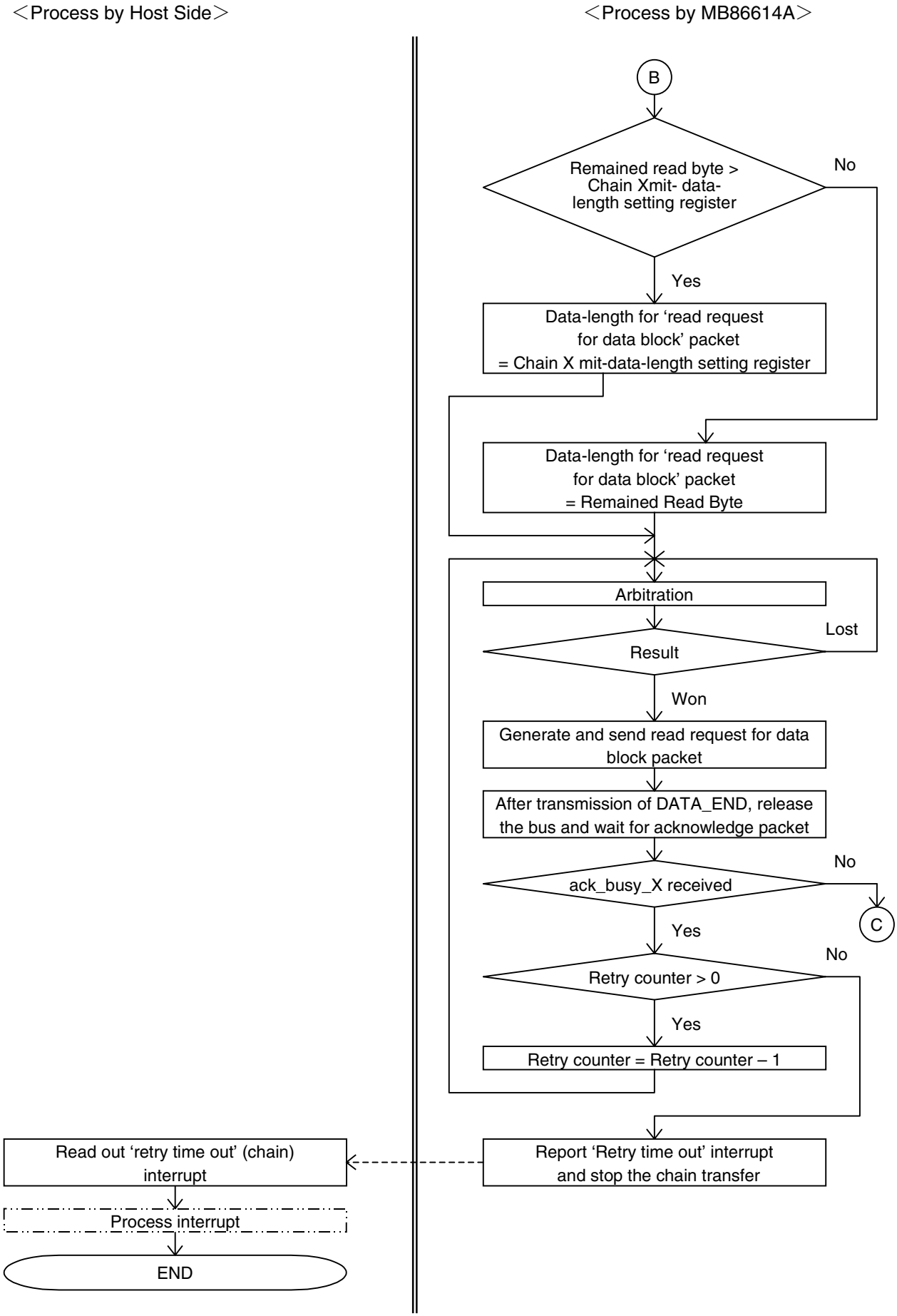
This section will explain the operation of Chain transfer.

5.4.4.1 Flow chart for Read Chain transfer operation
5.4.4.2 Flow chart for Write Chain transfer operation

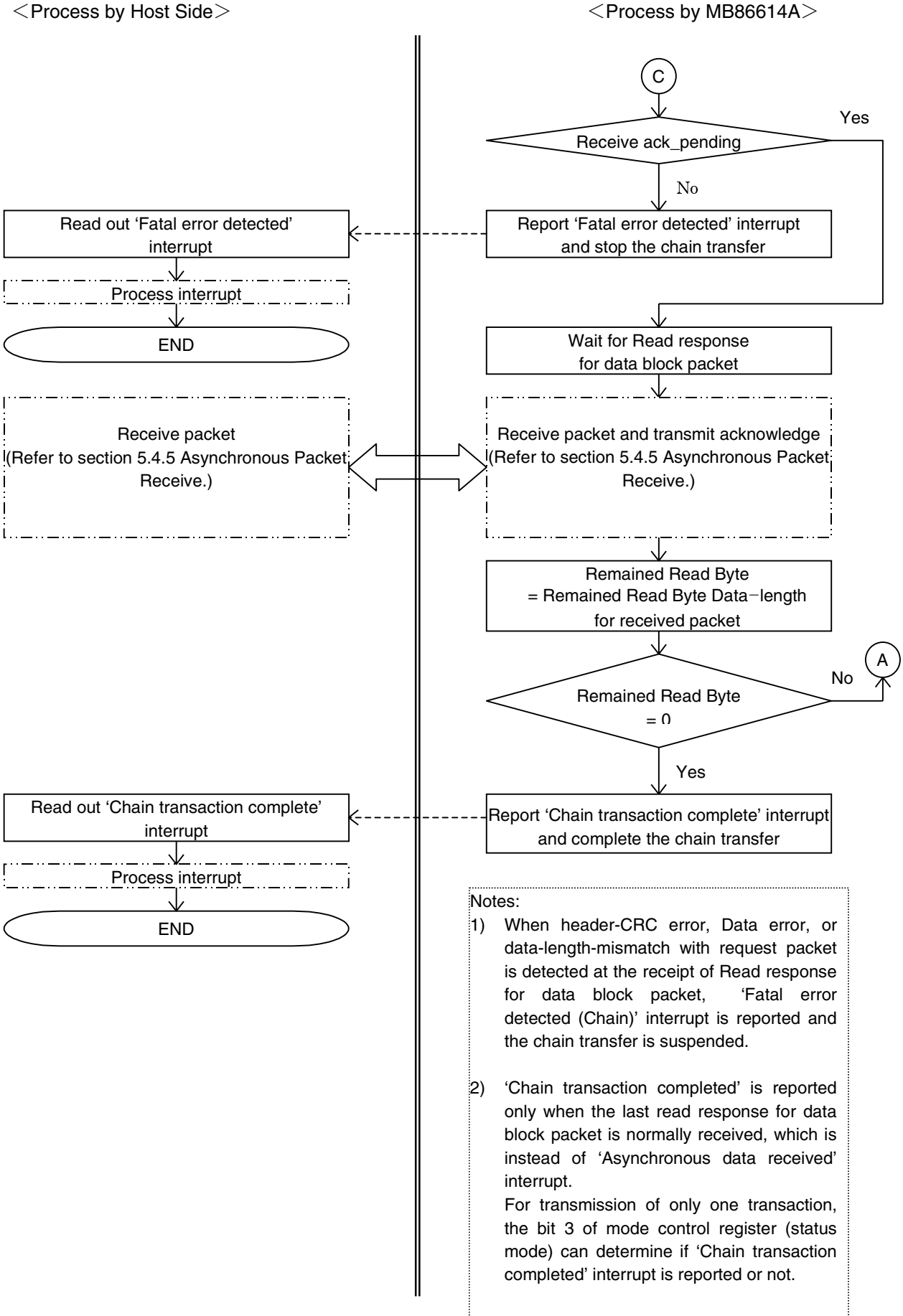
#### 5.4.4.1 Flow chart for Read Chain transfer operation




Continued to next page




Continued to next page

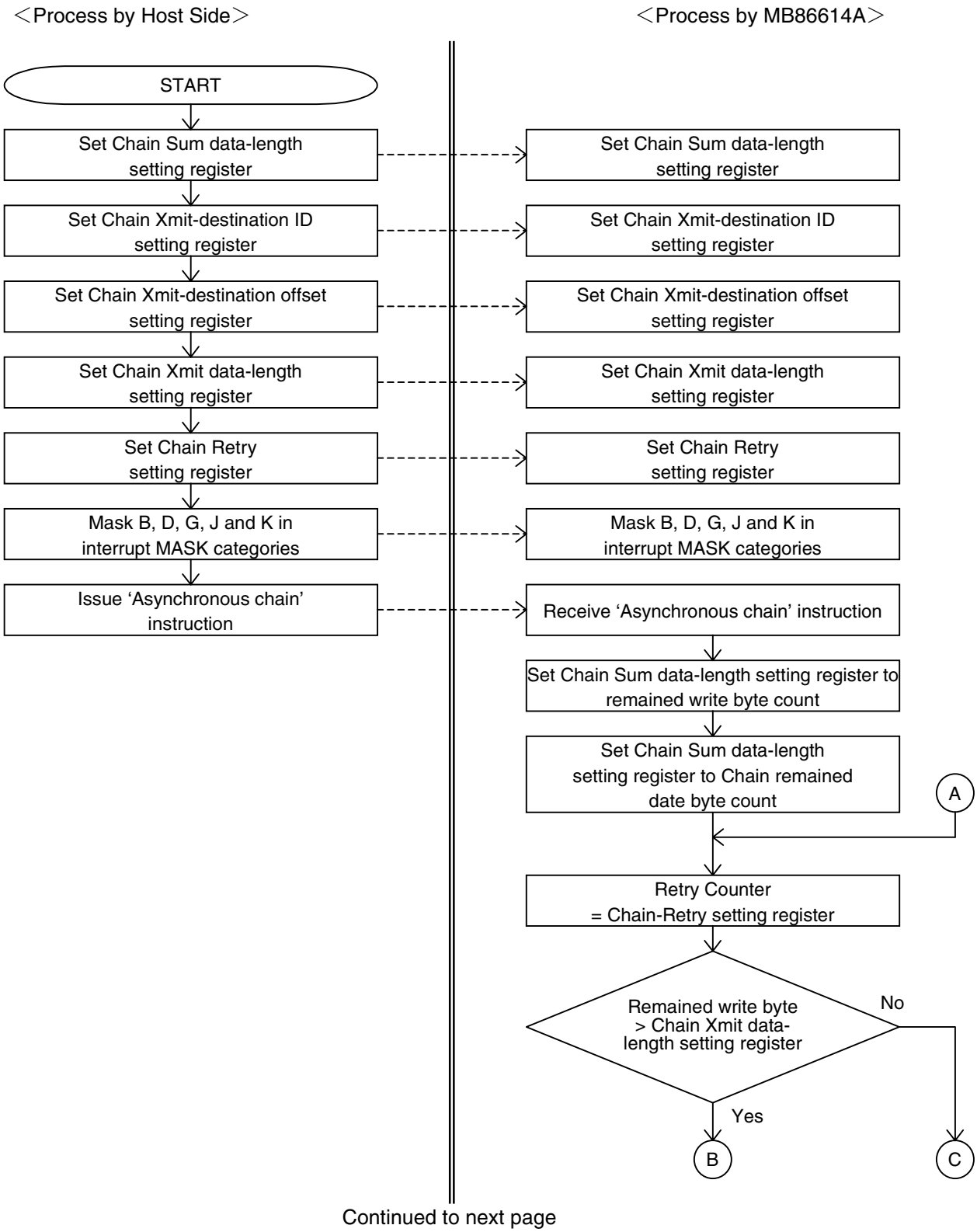



Notes:

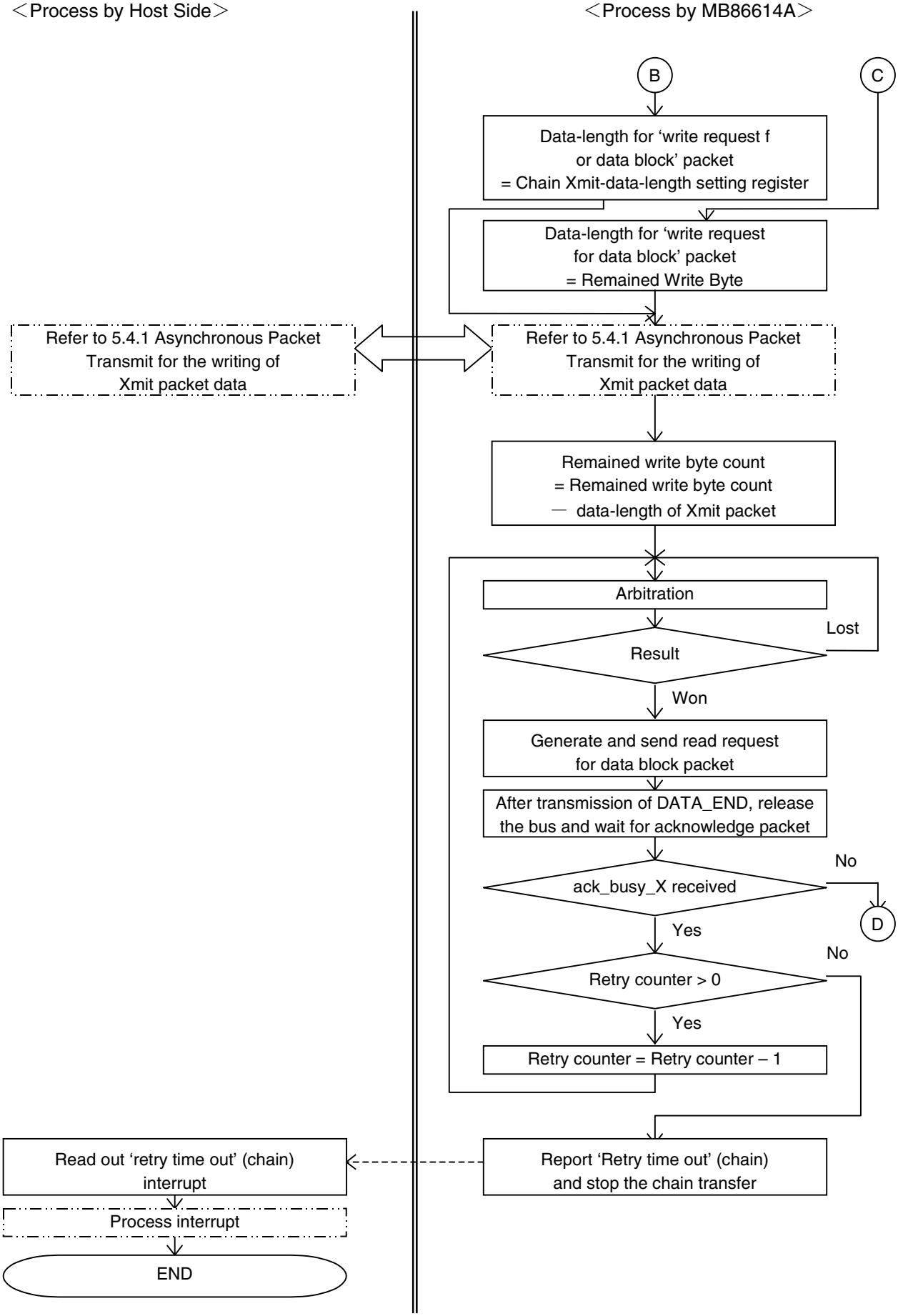
1) When header-CRC error, Data error, or data-length-mismatch with request packet is detected at the receipt of Read response for data block packet, 'Fatal error detected (Chain)' interrupt is reported and the chain transfer is suspended.

2) 'Chain transaction completed' is reported only when the last read response for data block packet is normally received, which is instead of 'Asynchronous data received' interrupt.
   For transmission of only one transaction, the bit 3 of mode control register (status mode) can determine if 'Chain transaction completed' interrupt is reported or not.

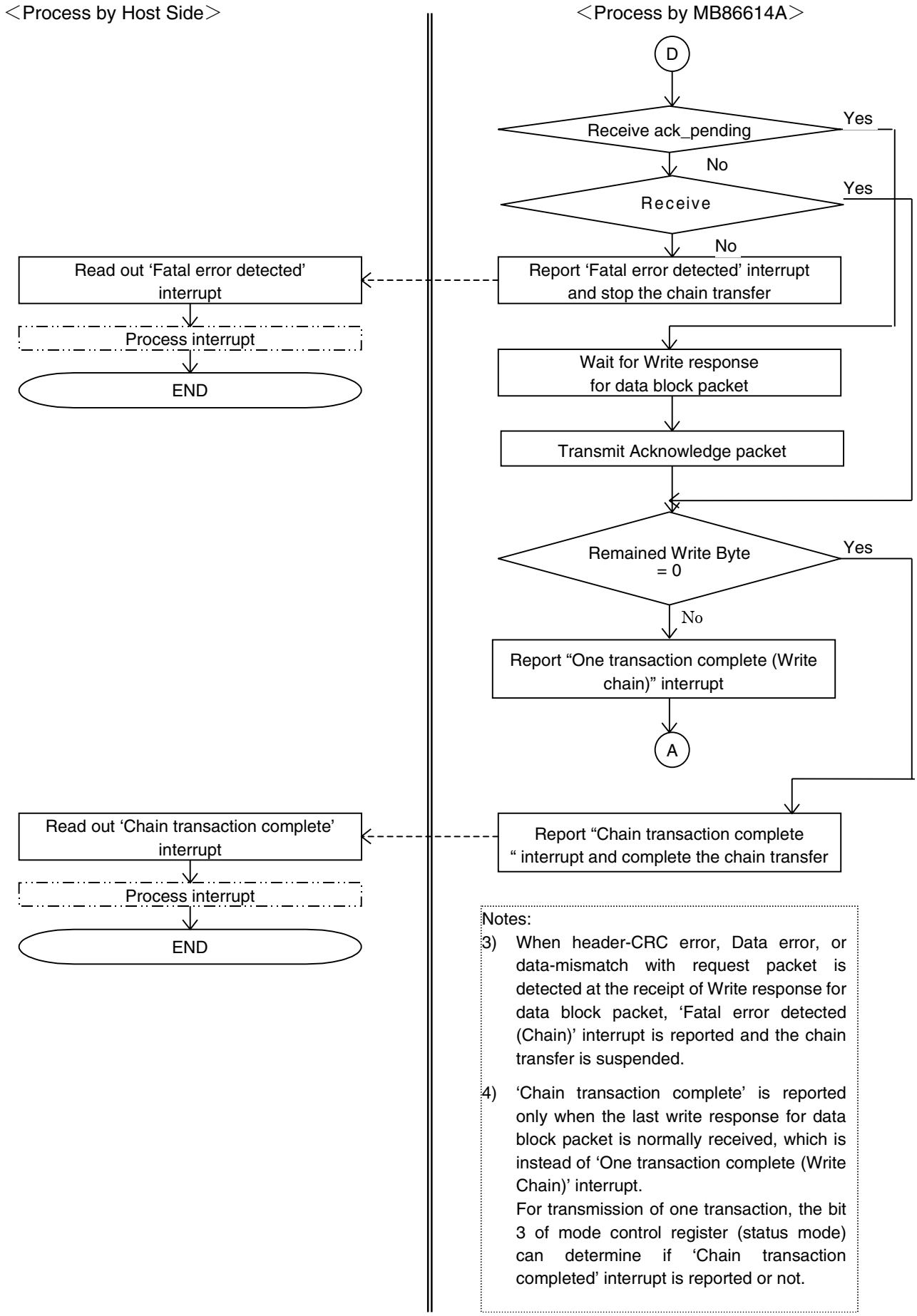
### 5.4.4.2 Flow chart for Write Chain transfer operation




Continued to next page




Continued to next page




Notes:

3) When header-CRC error, Data error, or data-mismatch with request packet is detected at the receipt of Write response for data block packet, ‘Fatal error detected (Chain)’ interrupt is reported and the chain transfer is suspended.

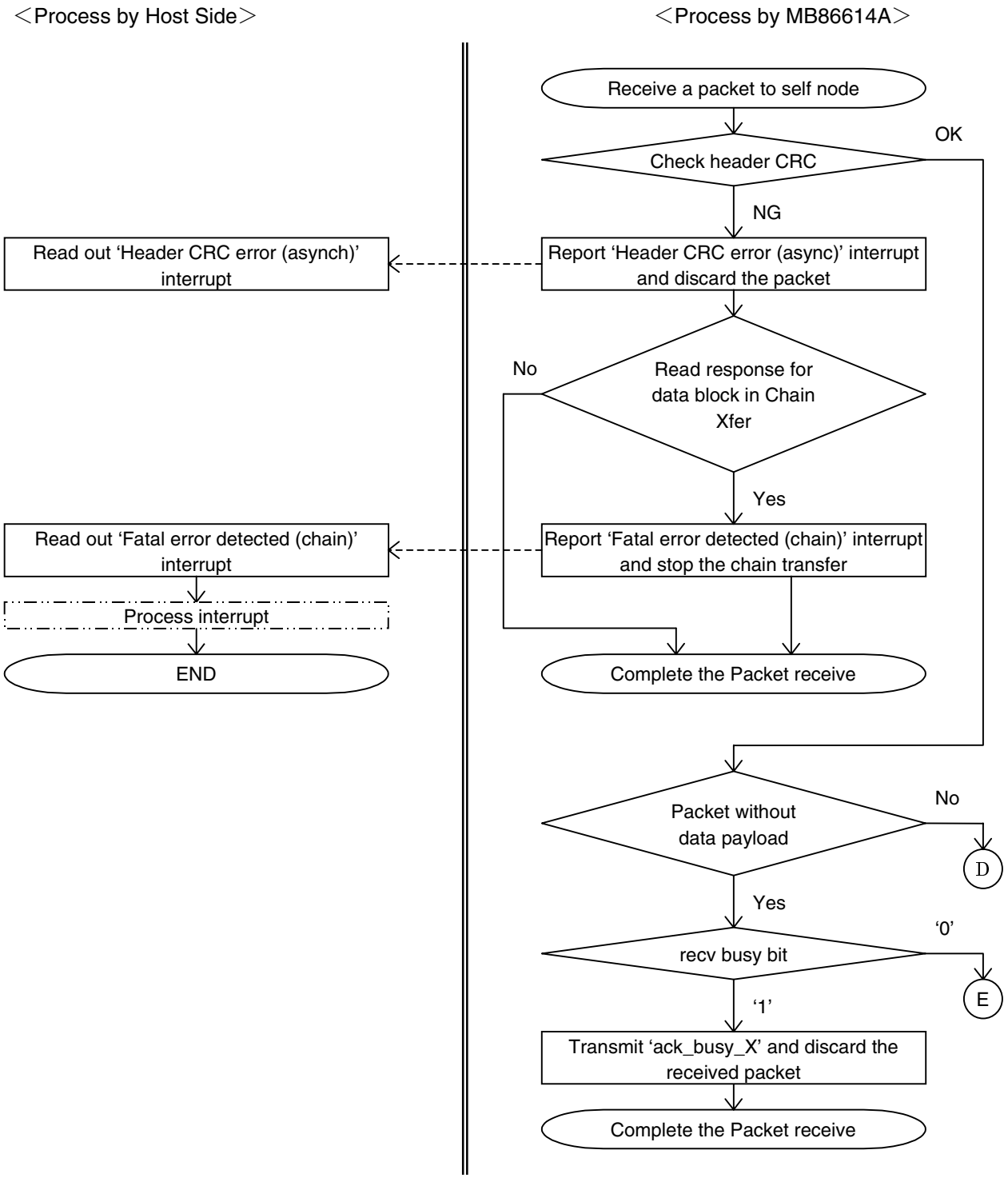
4) ‘Chain transaction complete’ is reported only when the last write response for data block packet is normally received, which is instead of ‘One transaction complete (Write Chain)’ interrupt.
For transmission of one transaction, the bit 3 of mode control register (status mode) can determine if ‘Chain transaction completed’ interrupt is reported or not.

### 5.4.5 Asynchronous packet Receive

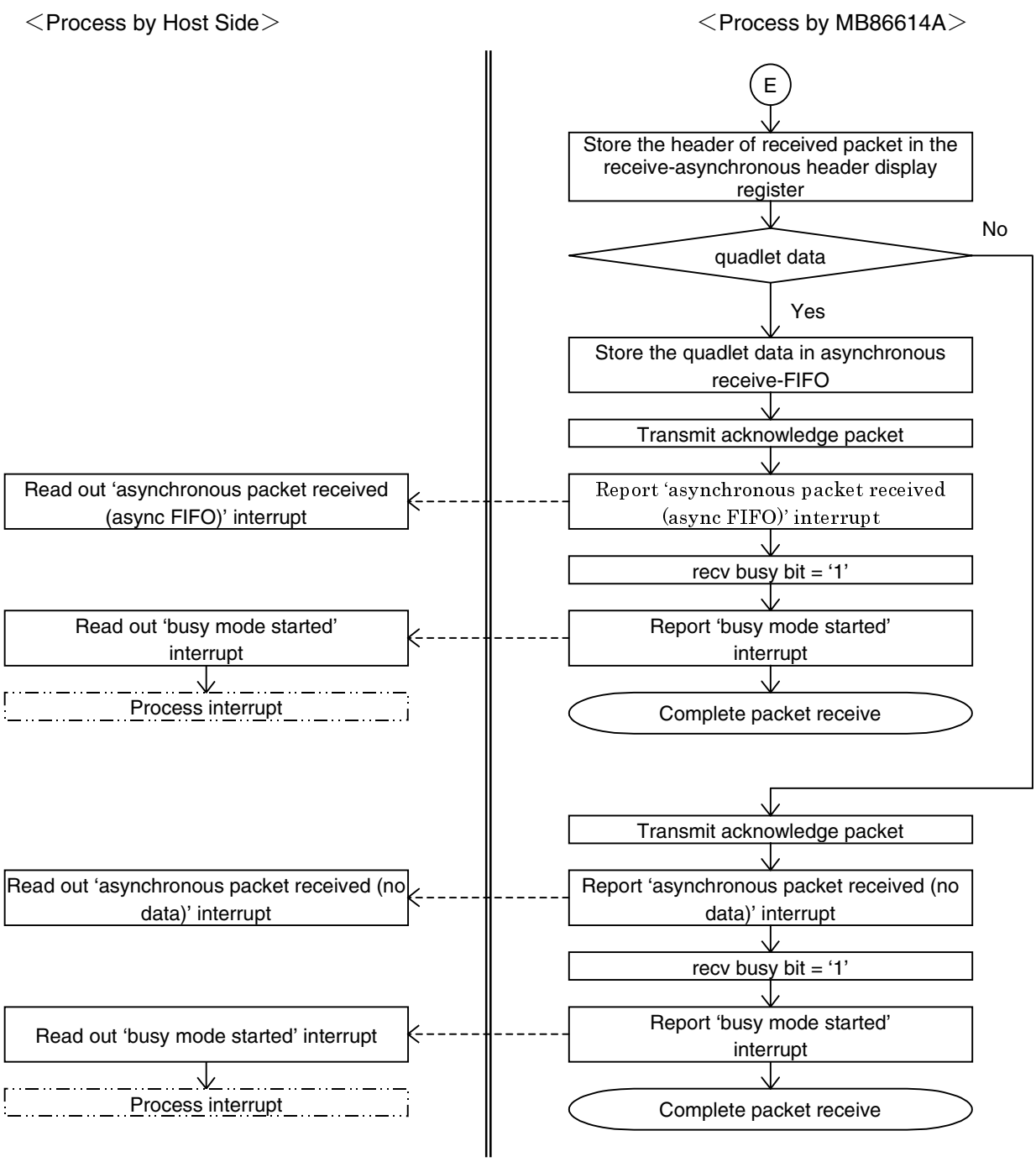
This section will explain the operation of Asynchronous packet receive

5.4.5.1 Flow chart for operation until data is stored in the FIFO
5.4.5.2 Flow chart for operation after data is stored in the FIFO

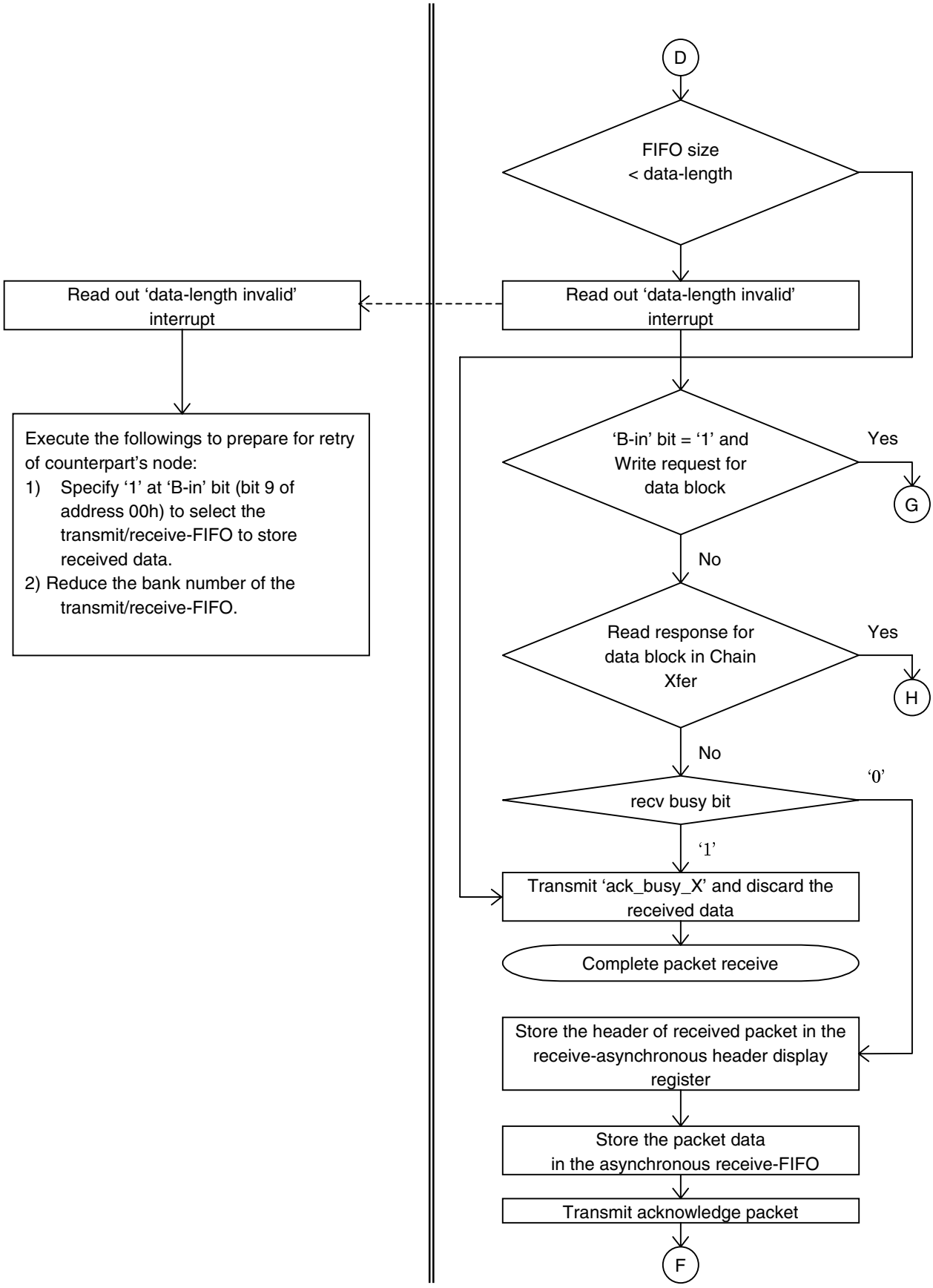
### 5.4.5.1 Flow chart for operation until data is stored in the FIFO



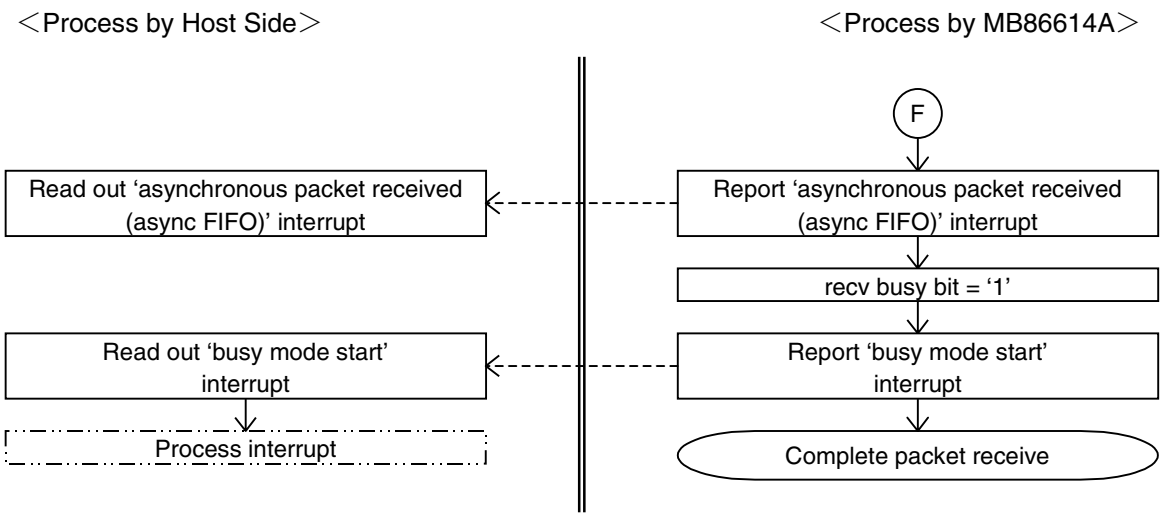

Continued to next page



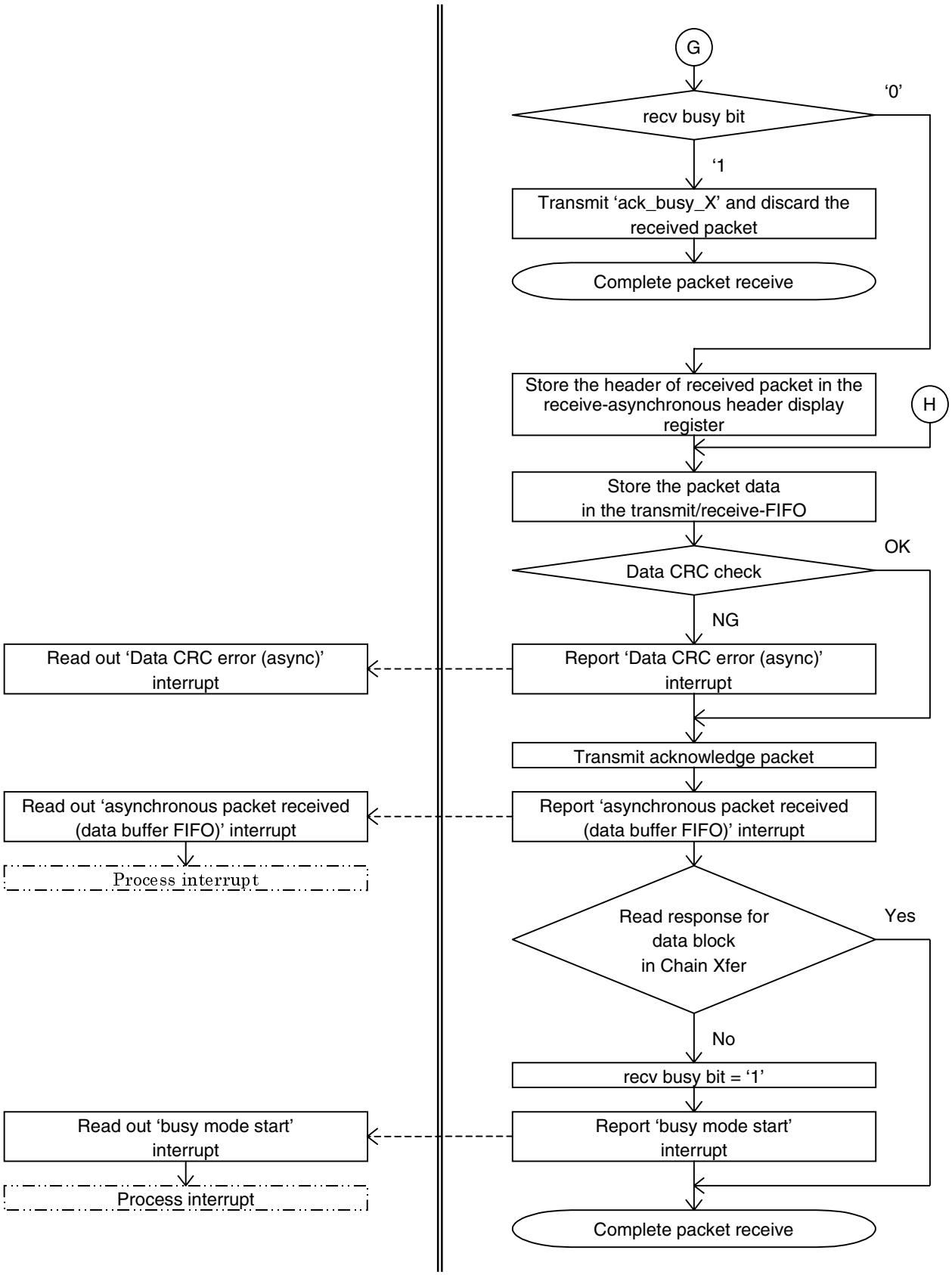

Continued to next page




Continued to next page




Continued to next page


＜Process by Host Side＞
＜Process by MB86614A＞
G
recv busy bit
‘0’
‘1
Transmit ‘ack_busy_X’ and discard the received packet
Complete packet receive
Store the header of received packet in the receive-asynchronous header display register
H
Store the packet data in the transmit/receive-FIFO
Data CRC check
OK
NG
Read out ‘Data CRC error (async)’ interrupt
Report ‘Data CRC error (async)’ interrupt
Transmit acknowledge packet
Read out ‘asynchronous packet received (data buffer FIFO)’ interrupt
Report ‘asynchronous packet received (data buffer FIFO)’ interrupt
Process interrupt
Read response for data block in Chain Xfer
Yes
No
recv busy bit = ‘1’
Read out ‘busy mode start’ interrupt
Report ‘busy mode start’ interrupt
Process interrupt
Complete packet receive

#### 5.4.5.2 Flow chart for operation after data is stored in the FIFO

Data read out process from a host side varies with the type of interrupt at the asynchronous packet receive as follows.

5.4.5.2.1 Flow chart when ‘asynchronous packet received (no data)’ interrupt is reported: int code ‘22h’

5.4.5.2.2 Flow chart when ‘asynchronous packet received (async-FIFO)’ interrupt is reported: int code ‘21h’

5.4.5.2.3 Flow chart when ‘asynchronous packet received (data buffer FIFO)’ interrupt is reported with DMA transfer while not doing Chain transfer: int code ‘C1h’

5.4.5.2.4 Flow chart when ‘asynchronous packet received (data buffer FIFO)’ interrupt is reported with MPU access while not doing Chain transfer: int code ‘C1h’

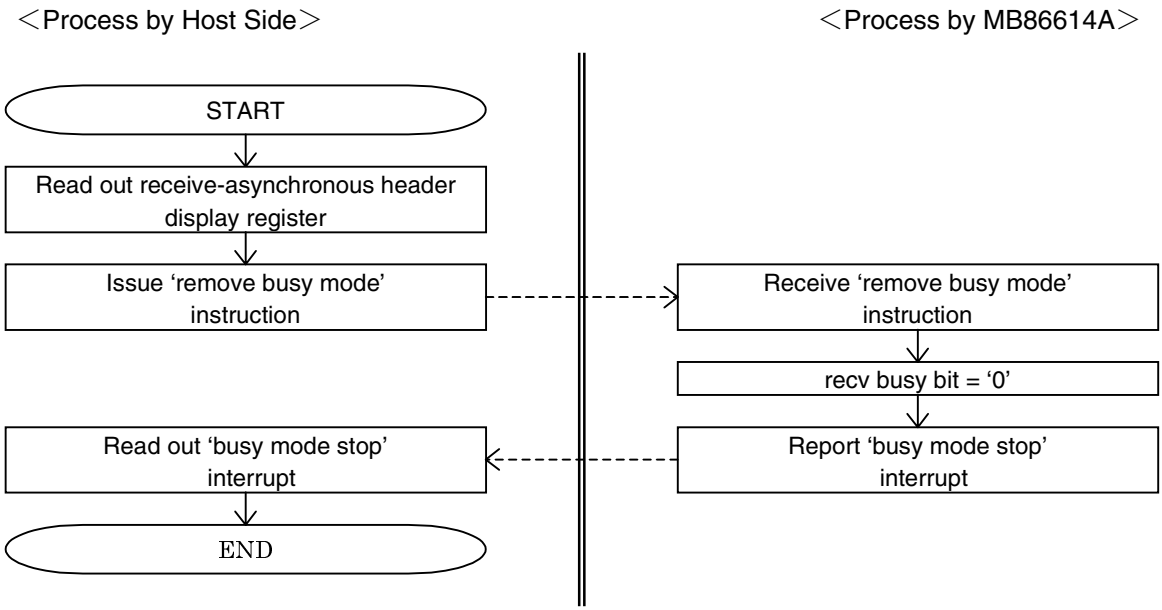
5.4.5.2.5 Flow chart when ‘asynchronous packet received (data buffer FIFO)’ interrupt is reported while doing Chain transfer, that is, when ‘DMA start (chain) is not issued: int code ‘C1h’

5.4.5.2.6 Flow chart when ‘Chain transaction complete’ interrupt is reported while doing Chain transfer, that is, when ‘DMA start (chain)’ is not issued: int code ‘C2h’

5.4.5.2.7 Flow chart when DMA Read is done continuously to issue ‘DMA start (chain)’ instruction before the chain transfer

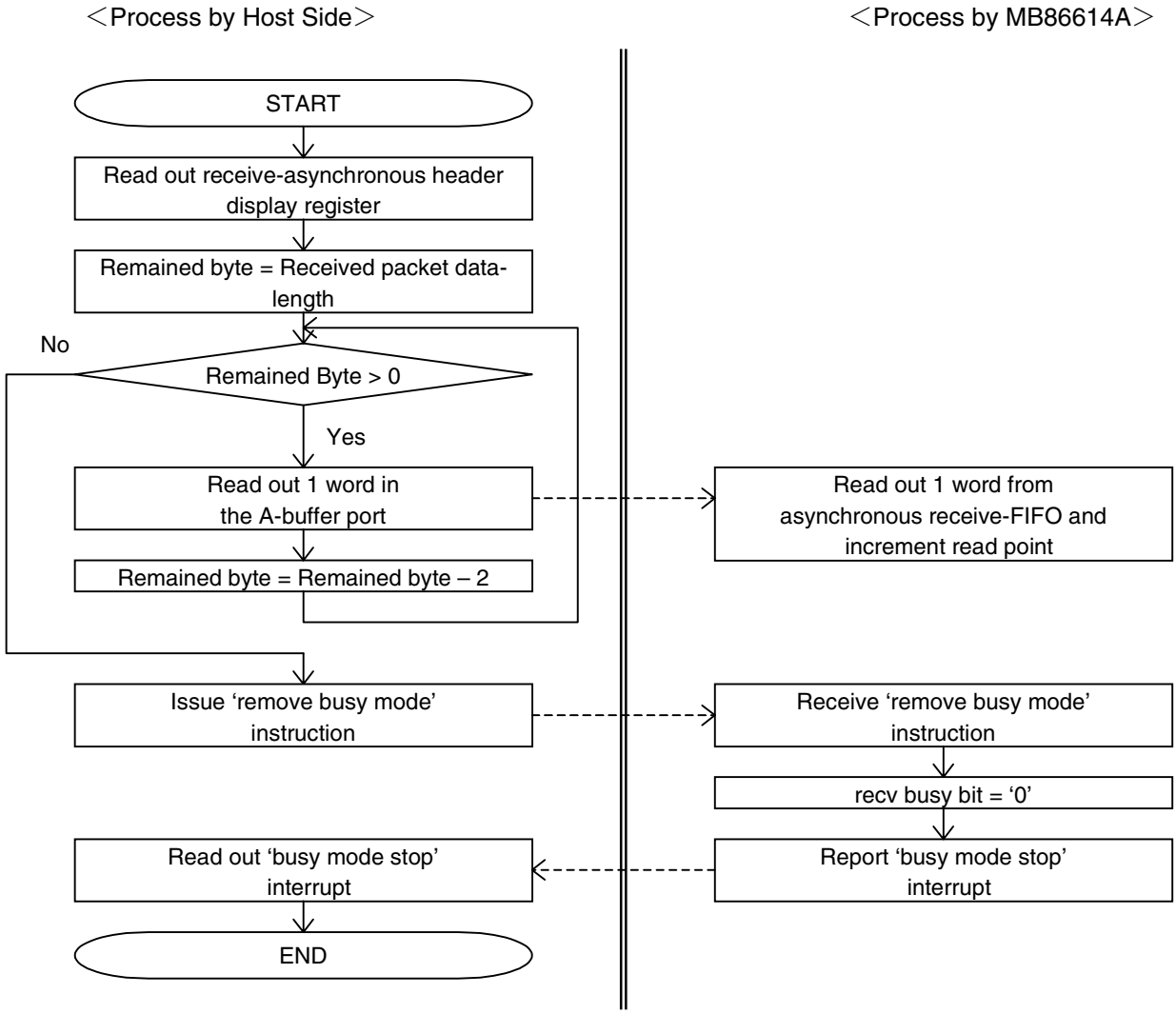
#### 5.4.5.2.1 Flow chart when ‘asynchronous packet received (no data)’ interrupt is reported: int code ‘22h’

The following is a flow chart when ‘asynchronous packet received (no data)’ interrupt is reported.

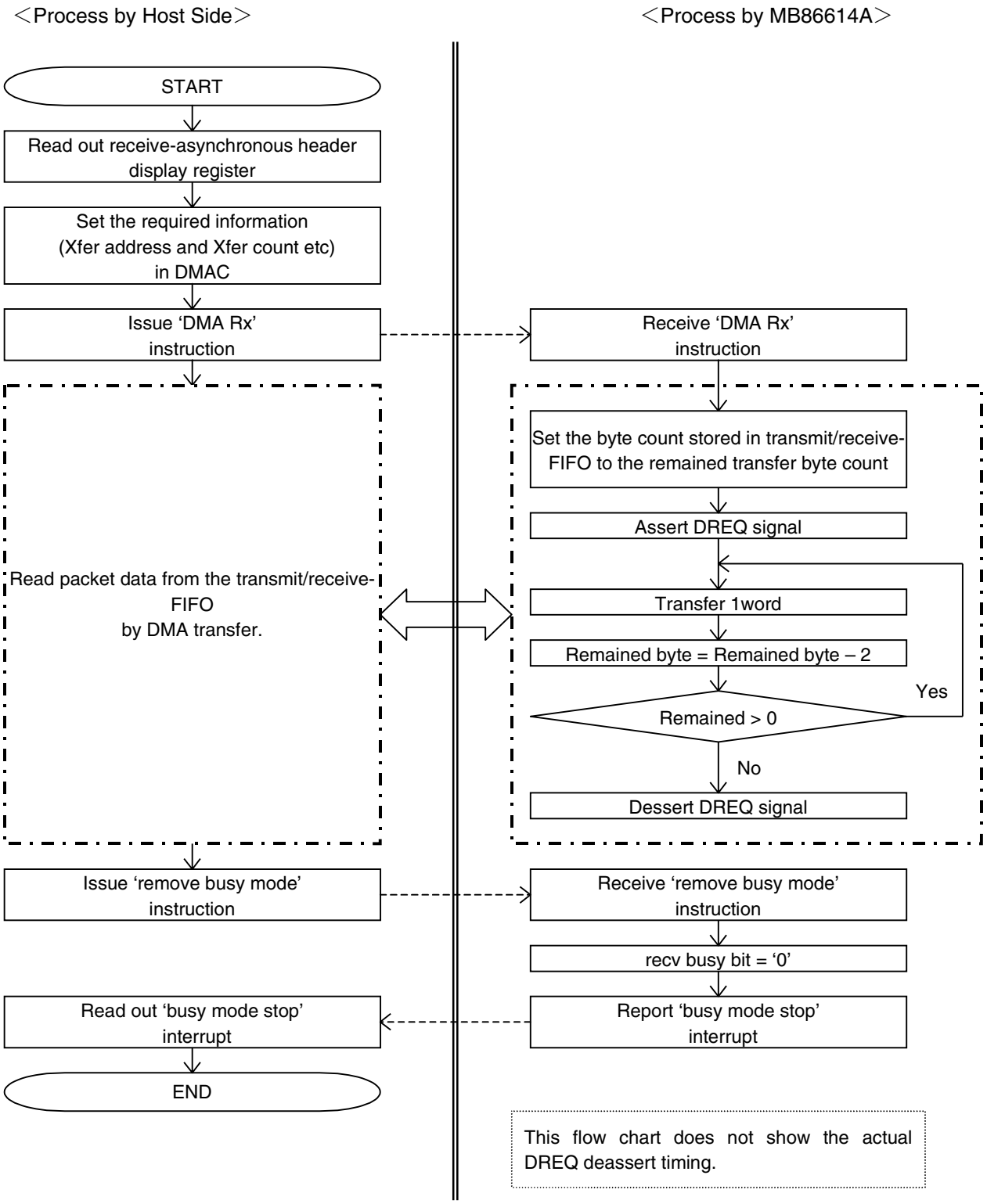


#### 5.4.5.2.2 Flow chart when ‘asynchronous packet received (async-FIFO)’ interrupt is reported: int code ‘21h’

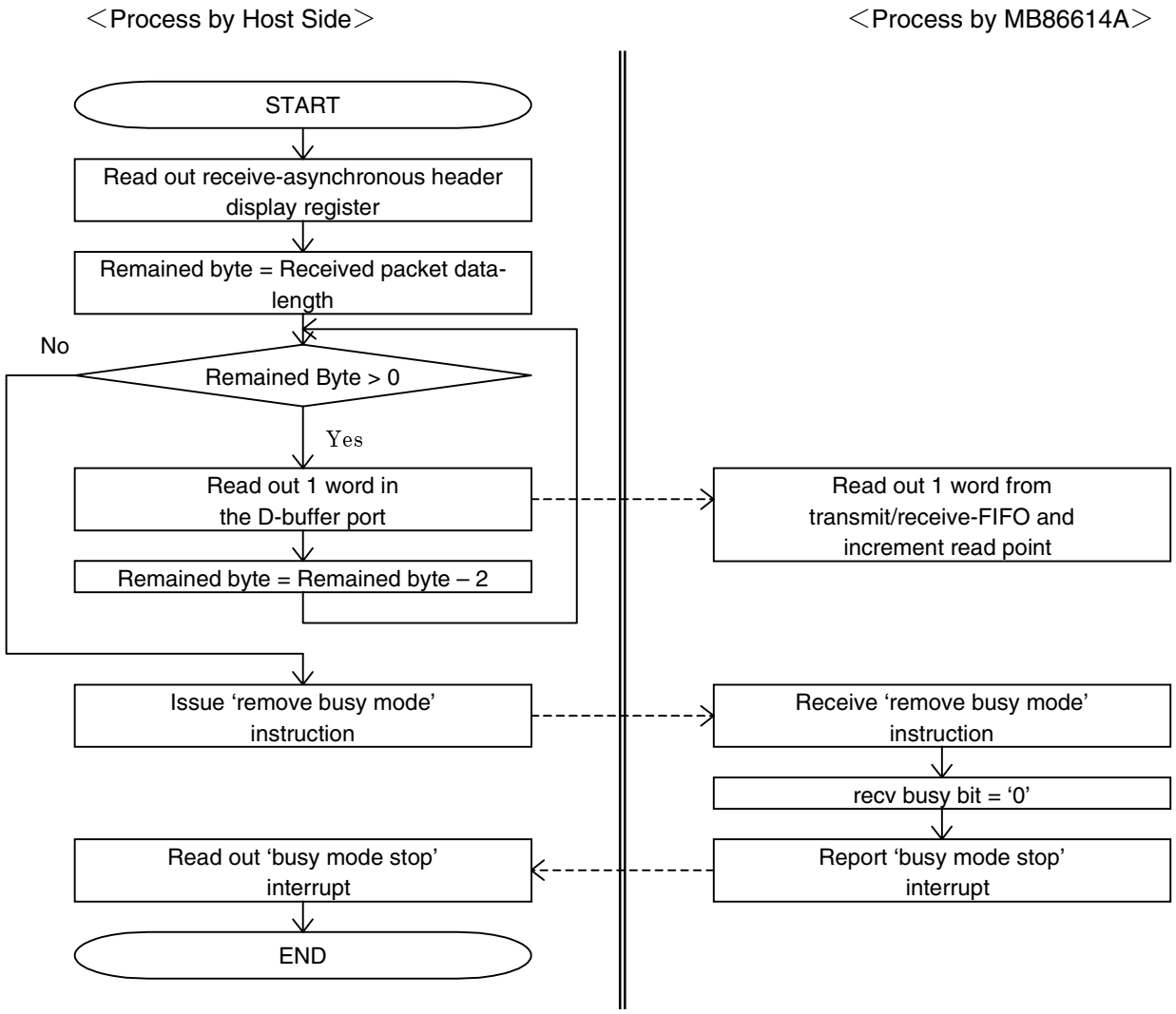
The following is a flow chart when ‘asynchronous packet received’ interrupt is reported (data stored in async-FIFO/21h).



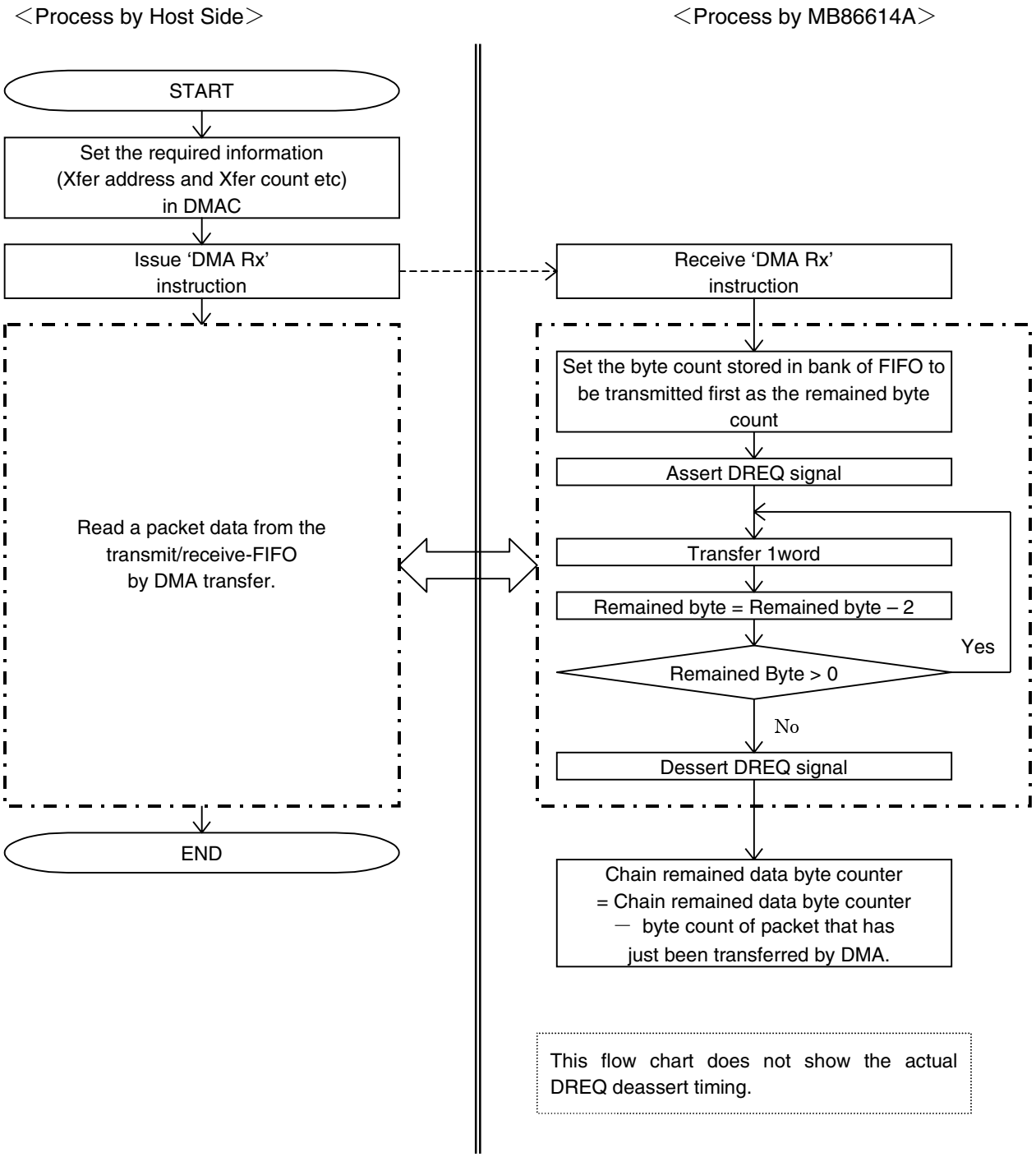
#### 5.4.5.2.3 Flow chart when ‘asynchronous packet received (data buffer FIFO)’ interrupt is reported while not doing Chain transfer (with DMA transfer): int code ‘C1h’



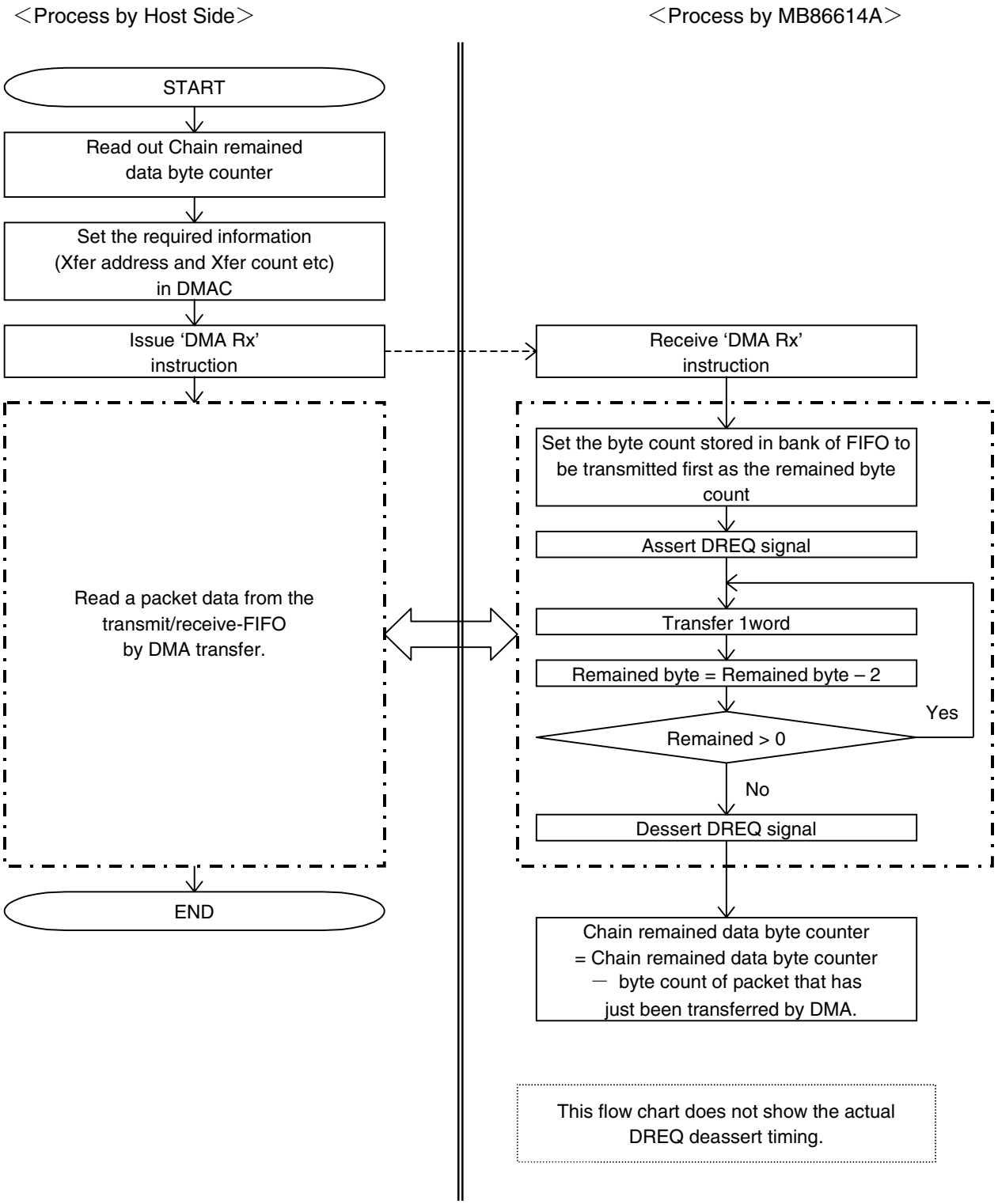
#### 5.4.5.2.4 Flow chart when ‘asynchronous packet received (data buffer FIFO)’ interrupt is reported while not doing Chain transfer (with MPU access): int code ‘C1h’



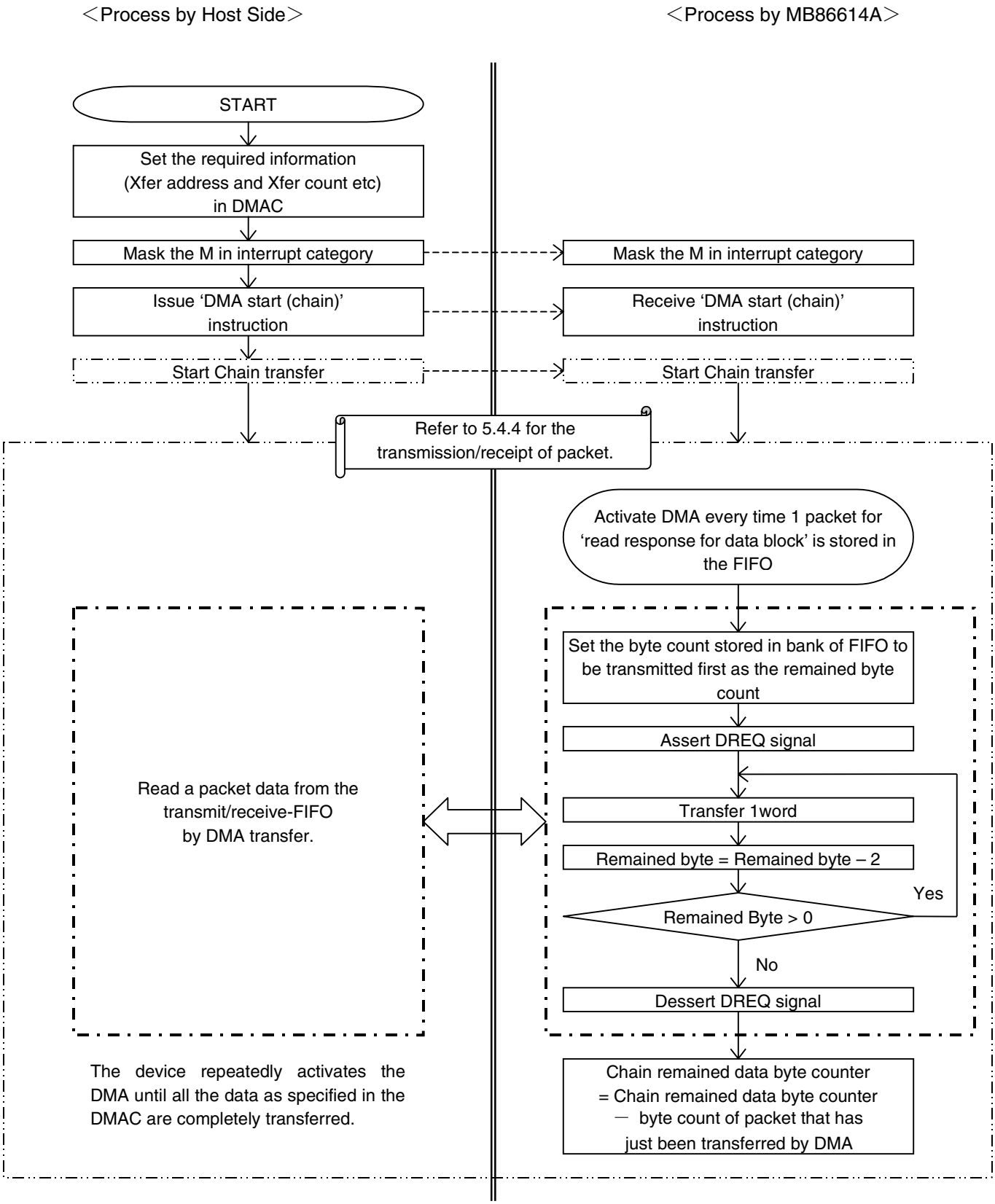
#### 5.4.5.2.5 Flow chart when 'asynchronous packet received (data buffer FIFO)' interrupt is reported while doing Chain transfer, that is, when 'DMA start (chain) is not issued: int code 'C1h'



#### 5.4.5.2.6 Flow chart when ‘Chain transaction complete’ interrupt is reported while doing Chain transfer, that is, when ‘DMA start (chain)’ is not issued: int code ‘C2h’



#### 5.4.5.2.7 Flow chart when DMA Read is done continuously to issue 'DMA start (chain)' instruction before the chain transfer




The continuous DMA transfer ends when the remained data byte count is '0' as a result of normal completion of chain transfer or when the chain transfer is suspended because of an error.

## 5.5 Automatic tcode insertion

For the packet transmission, MB86614A can automatically insert the tcode corresponding to the received instruction into the packet header. This operation is valid when the 'tcode add' bit (bit 1 of mode control register) is set to '1'.

Note:
For the 'read request for data block' packet automatically generated and transmitted in the chain transfer and the cycle start packet transmitted in 125 μs period when the self node is a root, proper tcode is always inserted in the packet header regardless of the 'tcode add' bit setting.

| Instruction | Code | tcode |
|---|---|---|
| Asynchronous write request for data quadlet | 30 h | 0 h |
| Asynchronous write request for block payload | 31 h | 1 h |
| Asynchronous write response for split transaction | 32 h | 2 h |
| Asynchronous read request for data quadlet | 33 h | 4 h |
| Asynchronous read request for block payload | 34 h | 5 h |
| Asynchronous read response for data quadlet | 35 h | 6 h |
| Asynchronous read response for data block payload | 36 h | 7 h |
| Lock request | 37 h | 9 h |
| Lock response | 38 h | B h |

## 5.6 Device Operation under 1394 Cable Power

The MB86614A device still operates under 1394 cable power source even if the system power source turns off as a repeater node. This section describes the device operation with cable power source. The description in this section is applied to the assumption/case that cable power is not being supplied to a host control device to control the MB86614A and it does not operate.

5.6.1 Pin Setting
5.6.2 Function
5.6.3 Device Initialization under Cable Powered Operation

### 5.6.1 Pin Setting

For cable powered device operation, the device pins, as listed below, must be set in accordance with the followings:

| Pin Name | Requirement |
|---|---|
| XRESET | Do not input ‘1’. |
| CPS | Make sure to connect with the cable power source. |
| PMODE | Make sure to switch the signal state to ‘L’ at the cable power ON, and do not switch to ‘H’ until the system power ON or cable power OFF. |
| PWR1-PWR3 | Set the POWER_CLASS area for the self-ID packet transmitted by the chip. |

| Pin Name | Requirement |
|---|---|
| D15 – D0 | Input state of the device.<br>The level is ‘0’ because the pin has a pull down resistance. |
| XINT | The pin outputs ‘0’.<br>This pin indicates ‘0’ output (active) at PMODE = 0, however it does not store an interrupt. |

### 5.6.2 Function

Under the cable power operation, the device can support the following functions:

1) Bus Reset Process (bus reset - tree identify - self identify)
2) Data Repeat
3) Link-on Packet Receive

This section will explain these functions..

5.6.2.1 Bus Reset Process (bus reset - tree identify - self identify)

5.6.2.2 Data Repeat
5.6.2.3 Link-on Packet Receive
5.6.2.4 State of pins under 1394 cable power source operation

#### 5.6.2.1 Bus Reset Process (bus reset - tree identify - self identify)

The device automatically executes bus reset when the 1394 cable is connected to the system of power OFF and the device starts to operate under cable power source.

| Field | 10 | | phy_ID | 0 | L | gap_cnt |
|---|---|---|---|---|---|---|
| VALUE | 1 | 0 | Current PHY_ID | 0 | 0 | Current value |

| sp | | del | | c | pwr | p0 | p1 | p2 | | i | m |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | PWR1-PWR3 setting | Current connection state | | 0 | 0 | * | 0 |

*: 'i' field should indicates '1' when the chip activates the automatic bus reset process. This field indicates '0' when the chip activates the bus reset process upon the receipt of BUS_RESET signal from another node.

#### 5.6.2.2 Data Repeat

Data repeating is possible after the 1394 cable power is supplied and the first bus reset process is completed.

#### 5.6.2.3 Link-on Packet Receive

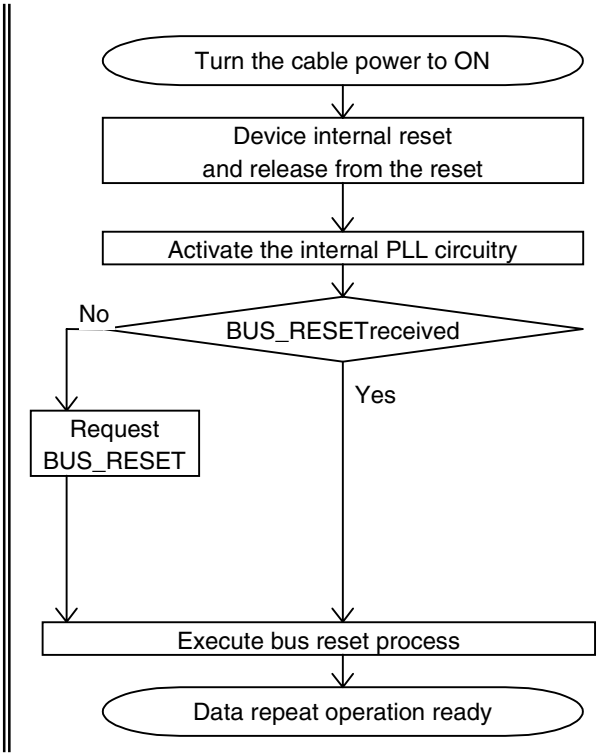
When the device receives the link-on packet for the self-node, it makes the LINKON pin active and reports the receipt of link-on packet. Whether or not the upper layer (link layer or upper) needs to be active after receiving the link-on packet depends on the application system. In order to make the device ready for the transfer, the following process are required:

a) System reset
b) Setting the Physical Register #4
c) Setting the other various registers
d) Bus Reset instruction (instruction code = ‘11h’)

### 5.6.3 Device Initialization under Cable Powered Operation

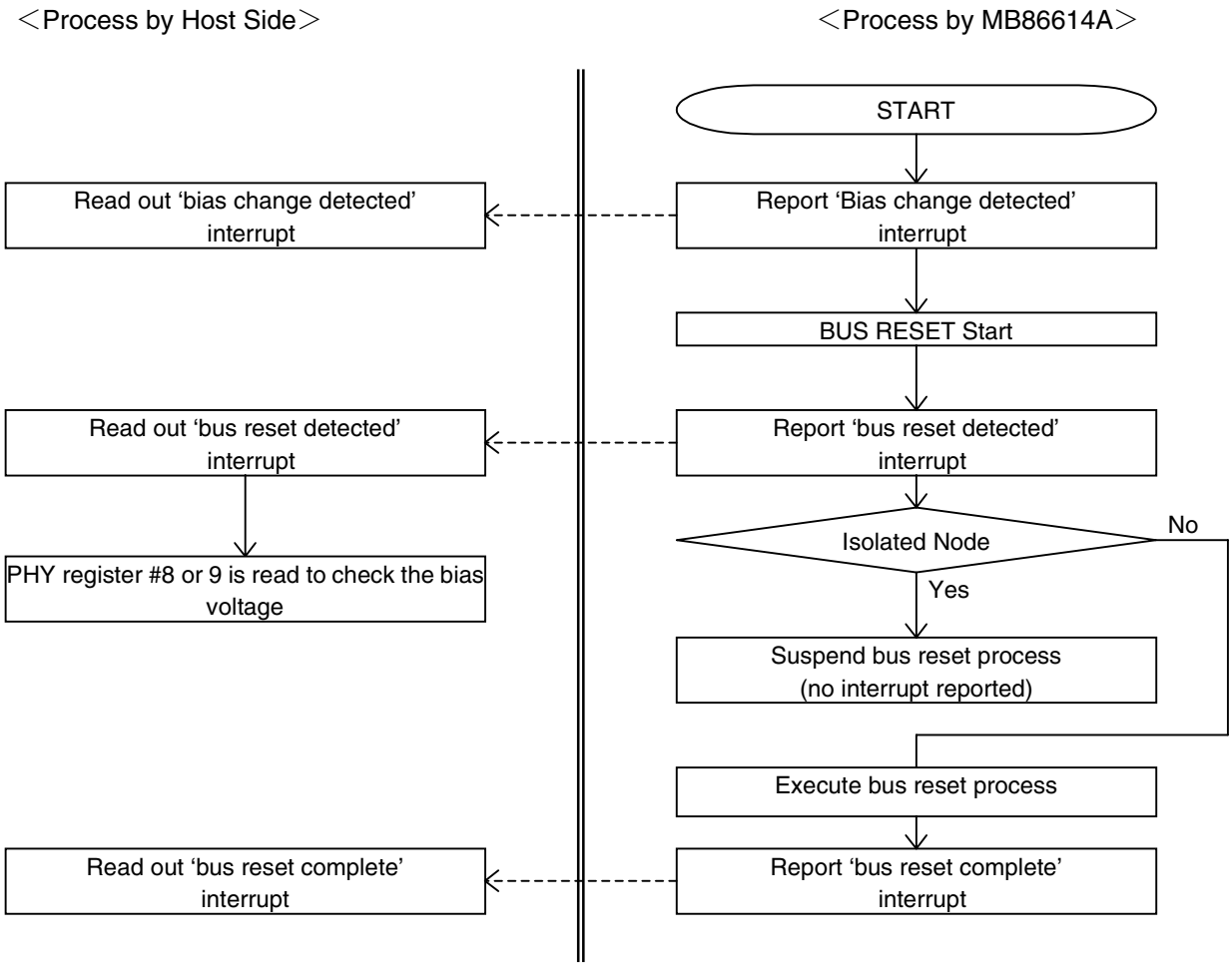
＜Process by Host Side＞

＜Process by MB86614A＞



## 5.7 Operation at ISOLATED NODE

If the connecting state of 1394 cable has changed or the bus of 1394 is in state except IDLE for more than set period, the chip detects an error and the PHY BLOCK automatically operates to generate BUS RESET.
The operation has not completed normally without counterpart.



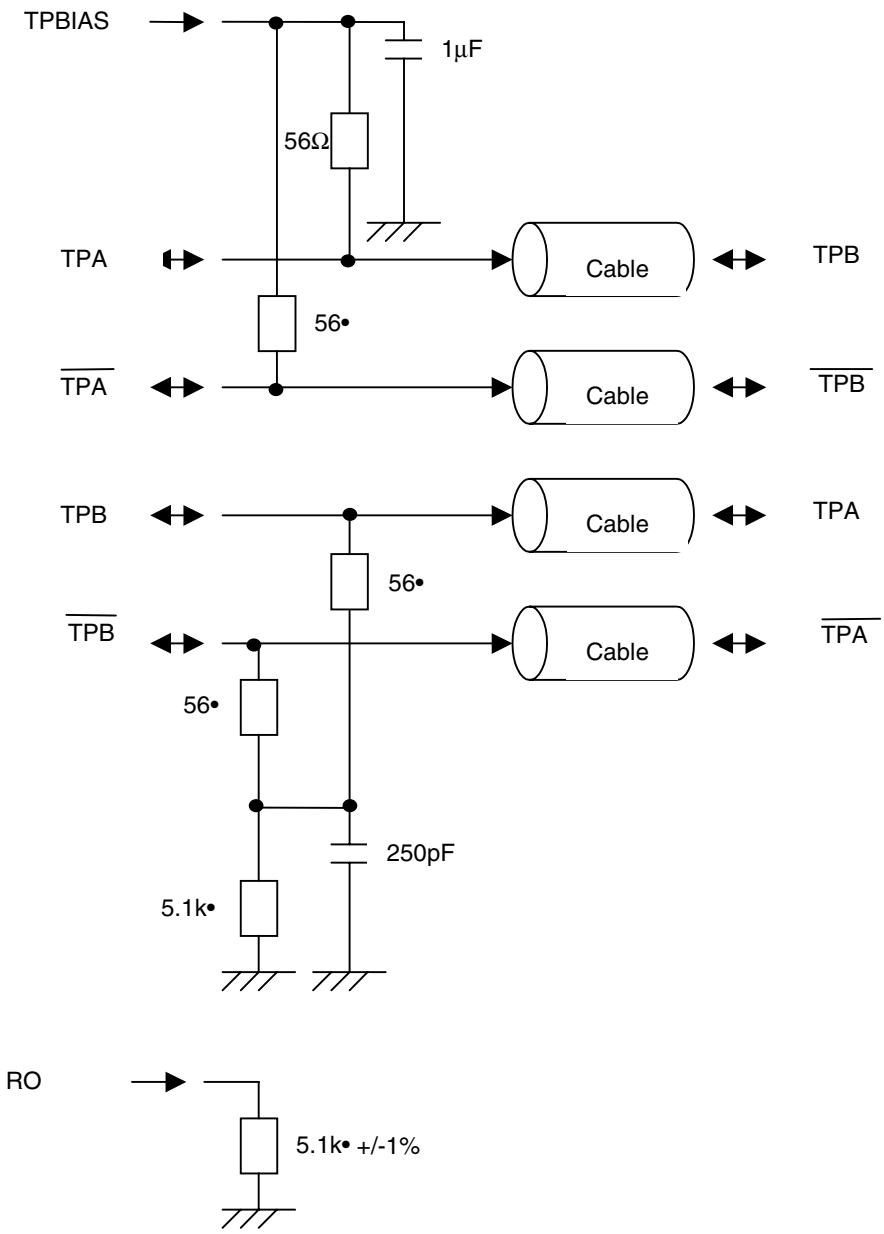
# 6. EXAMPLES OF SYSTEM CONFIGURATION

This chapter will show some examples of system configuration for the chip.

6.1 Recommended Port Connection Diagram (1-Port)
6.2 Recommended Connection Diagram for 1394 Cable Power
6.3 Recommended Connection Diagram for On-chip PLL Loop Filter
6.4 System Configuration

## 6.1 Recommended Port Connection Diagram (1-Port)

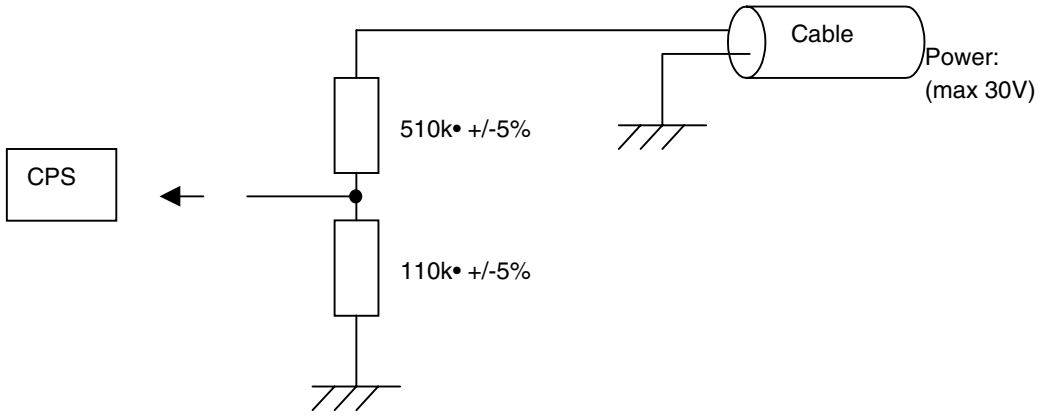
The figure below shows an example of recommended one-port connection.



## 6.2 Recommended Connection Diagram for 1394 Cable Power

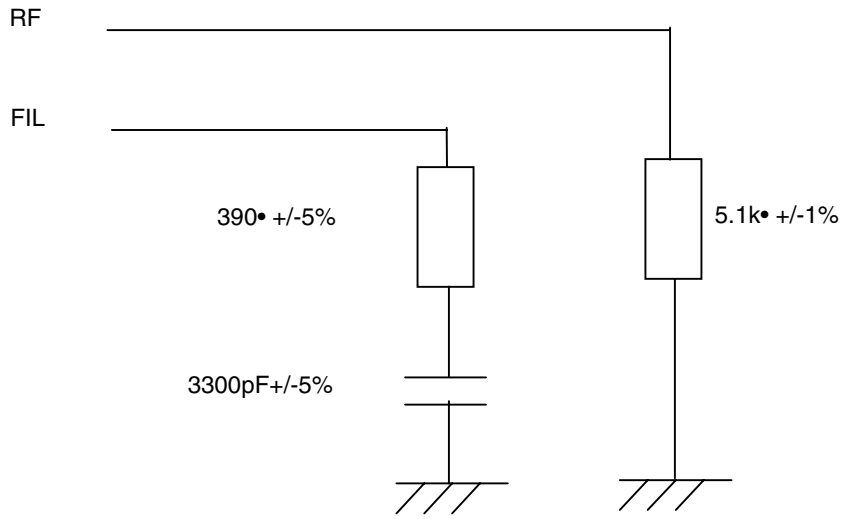
The figure below shows an example of 1394 cable power.

.



## 6.3 Recommended Connection Diagram for On-chip PLL Loop Filter

The figure below shows an example of recommended connection for on-chip PLL loop filter.



## 6.4 System Configuration

The figure below shows an example of system configuration on the chip.
For using a 68-series MPU, input ‘H’ to MODE pin.