OKI

Oki, Network Solutions for a Global Society

OKI Semiconductor

PEDL60841-01
Issue Date: Oct. 2, 2002

ML60841

Preliminary

USB Host + Device controller LSI

GENERAL DESCRIPTION

The ML60841 is a controller LSI with a device controller and a host controller that conform to the Universal Serial Bus (USB) 1.1. implemented on a single chip. By setting an external pin, the ML60841 can be used interchangeably as either a host controller or a device controller. The host controller section and device controller section support data transfer modes such as a control transfer mode, an interrupt transfer mode, a bulk transfer mode, and an isochronous transfer mode. The host controller section supports one USB port transceiver, can be connected to a USB transceiver LSI, and conforms to the OpenHCI Specification 1.0a.

FEATURES

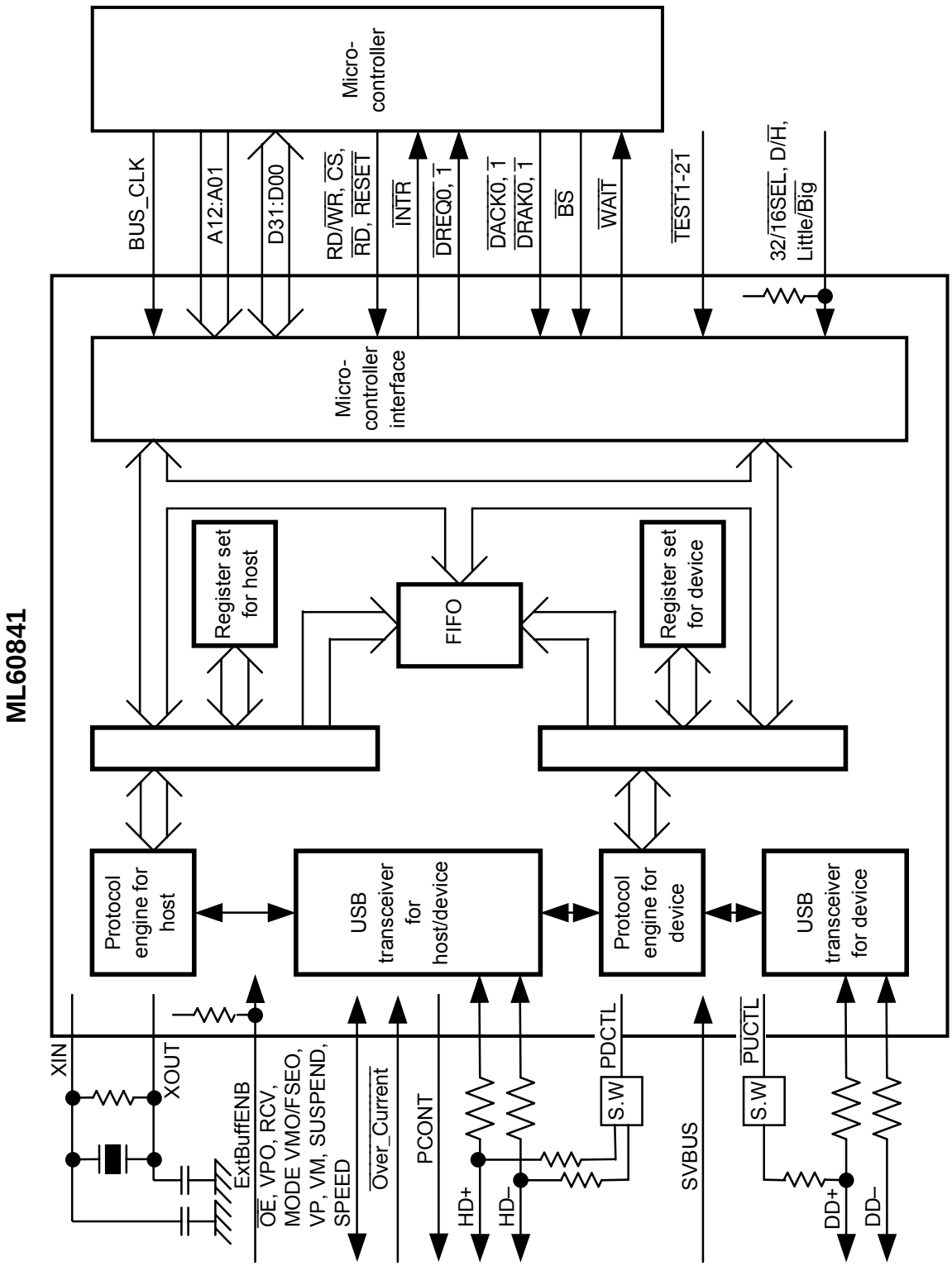
[Common Section]
- USB 1.1 compliant
- Maximum bus clock (BUS_CLK) frequency: 33 MHz
- Allows host/device selection by an external terminal
- 16-/32-bit bus width selectable
- Little/Big endian can be selected
- Common connector use/no use can be selected
- Occupies 0000h to 1FFFh 8K byte space (registers: 0xxxh, internal RAM: 1000h to 1FFFh)
- DMA control of high flexibility.
  - Supports host mode: 1 channel, device mode: 2 channels
  - Supports the following DMA transfer mode
    Transfer size: 32 bits (need to specify the same transfer size as that of the microcontroller side DMA controller)

[Host Controller Section]
- OpenHCI (Open Host Controller Interface) 1.0a compliant
- Supports 4 data transfer types (control transfer, bulk transfer, interrupt transfer, and isochronous transfer)
- Built-in 4-Kbyte RAM
- DMA slave function (1 channel) reduces software load
- Supports 1 USB port (supports full speed (12 Mbps) and low speed (1.5 Mbps))
- Supports SOF generation and CRC5/16 bit generation function

[Device Controller Section]
- Supports full speed (12 Mbps)
- Supports 4 data transfer types (control transfer, bulk transfer, interrupt transfer, and isochronous transfer)
- End points: 5 or 6
  - Control EP : 1 (EP0)
  - Bulk/interrupt EP : 3 (EP1, EP2, and EP3)
  - Isochronous/bulk/interrupt EP : 1 or 2 (EP4 and/or EP5)
- Built-in FIFO for storing data
- Double-layer configuration FIFO of EP1, EP2, EP4, and EP5
- Supports 2-channel DMA slave function (EP1, EP2, EP4, and EP5)
- Supports Suspend and Wakeup functions.

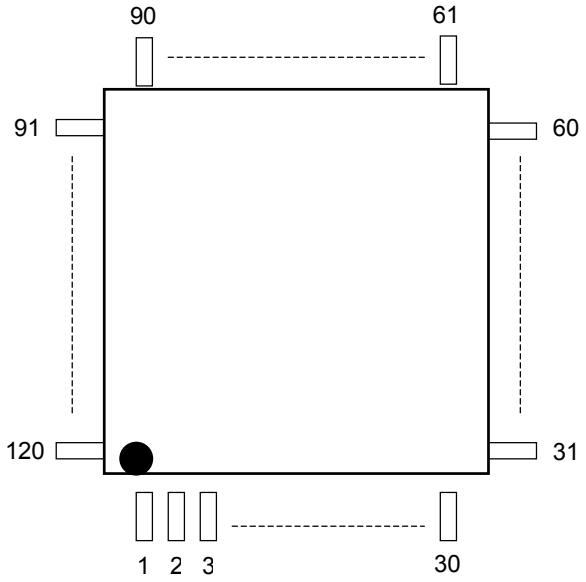
[Others]
- USB port can be connected to a USB transceiver LSI
- 48 MHz crystal oscillator
- 3.3 V single power supply
- Package: 120-pin TQFP (TQFP120-P-1414-0.40-K)
  120-pin BGA (P-TFBGA120-0909-0.65)

BLOCK DIAGRAM



PIN CONFIGURATION

120-pin TQFP (Top View)



120-pin TQFP

<TQFP Pin Layout>

Pin No	I/O	I/O Type	Signal Code	Pin No	I/O	I/O Type	Signal Code	Pin No	I/O	I/O Type	Signal Code
1	I/O	*3	D2	41	I	*2	$\overline{TMD0}$	81	I	*4	$\overline{BS}$
2	I/O	*3	D1	42	O	*5	SUSPEND	82	I/O	*6	TIO19
3	I/O	*3	D0	43	O	*5	SPEED	83	I	*4	BUS_CLK
4	—	—	GND	44	O	*5	MODE	84	—	—	GND
5	—	—	V_{CC}	45	I	*2	$\overline{TMD1}$	85	—	—	V_{CC}
6	I	*1	$\overline{RESET}$	46	O	*5	VPO	86	I/O	*3	D31
7	I/O	*6	TIO00	47	I	*2	$\overline{TMD2}$	87	I/O	*3	D30
8	I/O	*6	TIO01	48	O	*5	VMO/FSEO	88	I/O	*3	D29
9	I	*2	Little/$\overline{Big}$	49	I	*2	$\overline{TMD3}$	89	I/O	*3	D28
10	I	*2	32/$\overline{16SEL}$	50	O	*5	$\overline{OE}$	90	I/O	*3	D27
11	I/O	*6	TIO02	51	I/O	*6	TIO11	91	I/O	*3	D26
12	I/O	*6	TIO03	52	I	*2	D/$\overline{H}$	92	I/O	*3	D25
13	I	*4	A12	53	I	*2	ExtBuffENB	93	I/O	*3	D24
14	I	*4	A11	54	I/O	*6	TIO12	94	—	—	GND
15	I	*4	A10	55	I/O	*6	TIO13	95	—	—	V_{CC}
16	I	*4	A9	56	O	*5	$\overline{PCONT}$	96	I/O	*3	D23
17	I	*4	A8	57	I	*4	$\overline{Over\_Current}$	97	I/O	*3	D22
18	I	*4	A7	58	O	*5	PDCTL	98	I/O	*3	D21
19	I	*4	A6	59	I/O	*6	TIO14	99	I/O	*3	D20
20	I	*4	A5	60	I/O	*6	TIO15	100	I/O	*3	D19
21	I	*4	A4	61	I/O		HD–	101	I/O	*3	D18
22	I	*4	A3	62	I/O		HD+	102	I/O	*3	D17
23	I	*4	A2	63	—	—	GND	103	I/O	*3	D16
24	I	*4	A1	64	—	—	V_{CC}	104	—	—	GND
25	I/O	*6	TIO04	65	I		XIN	105	—	—	V_{CC}
26	I/O	*6	TIO05	66	O		XOUT	106	I/O	*3	D15
27	—	—	GND	67	I/O	*6	TIO16	107	I/O	*3	D14
28	—	—	V_{CC}	68	I/O	*6	TIO17	108	I/O	*3	D13
29	I/O		DD–	69	I/O	*6	TIO18	109	I/O	*3	D12
30	I/O		DD+	70	O	*5	$\overline{INTR}$	110	I/O	*3	D11
31	I/O	*6	TIO06	71	O	*5	$\overline{DREQ0}$	111	I/O	*3	D10
32	I/O	*6	TIO07	72	I	*4	$\overline{DACK0}$	112	I/O	*3	D9
33	O	*5	$\overline{PUCTL}$	73	I	*4	$\overline{DRAK0}$	113	I/O	*3	D8
34	I	*4	SVBUS	74	O	*5	$\overline{DREQ1}$	114	—	—	GND
35	I/O	*6	TIO08	75	I	*4	$\overline{DACK1}$	115	—	—	V_{CC}
36	I	*4	RCV	76	I	*4	$\overline{DRAK1}$	116	I/O	*3	D7
37	I/O	*6	TIO09	77	O	*5	$\overline{WAIT}$	117	I/O	*3	D6
38	I	*4	VP	78	I	*4	$\overline{CS}$	118	I/O	*3	D5
39	I/O	*6	TIO10	79	I	*4	RD/$\overline{WR}$	119	I/O	*3	D4
40	I	*4	VM	80	I	*4	$\overline{RD}$	120	I/O	*3	D3

Note: I/O types
*1: TTL Schmitt Input Buffer with 4X Drive,
*2: TTL Input Buffer with 4X Drive/50K Pull Up,
*3: I/O Buffer with TTL Input/4mA Output,
*4: TTL Input Buffer with 4X Drive,
*5: Push Pull Output Buffer with 4mA Drive,
*6: I/O Buffer with TTL Input/4mA Output 50K Pull Up

120-pin BGA (Top View)

A	B	C	D	E	F	G	H	J	K	L	M	N	
D2	D3	D7	GND	D10	D13	VCC	D17	D20	D23	D24	D26	D25	13
D1	D4	D6	D8	D9	D12	D15	D16	D19	D22	GND	D27	D29	12
VCC	D0	GND	D5	VCC	D11	D14	GND	D18	D21	VCC	D30	VCC	11
TIO01	TIO00	$\overline{RES}$ $\overline{ET}$								D28	GND	TIO19	10
TIO02	32/$\overline{16}$ $\overline{SEL}$	Little/ $\overline{Big}$								D31	$\overline{BS}$	RD/ $\overline{WR}$	09
A11	A12	TIO03								BUS_CLK	$\overline{CS}$	$\overline{DRAK1}$	08
A8	A9	A10								$\overline{RD}$	$\overline{DACK1}$	$\overline{DRAK0}$	07
A5	A6	A7								$\overline{WAIT}$	$\overline{DACK0}$	$\overline{INTR}$	06
A2	A3	A4								$\overline{DREQ1}$	TIO18	TIO16	05
TIO05	TIO04	A1								$\overline{DREQ0}$	XOUT	XIN	04
VCC	GND	RCV	TIO10	SUSPEND	$\overline{TMD1}$	VMO/FSEO	TIO11	TIO12	$\overline{Over\_Current}$	TIO7	VCC	GND	03
DD+	DD–	SVBUS	TIO09	VM	SPEED	VPO	$\overline{TMD3}$	D/$\overline{H}$	TIO13	PDCTL	HD+	HD–	02
TIO06	TIO07	$\overline{PUCTL}$	TIO08	VP	$\overline{TMD0}$	MODE	$\overline{TMD2}$	$\overline{OE}$	ExtBuffENB	$\overline{PCONT}$	TIO14	TIO15	01

1-pin mark

120-pin BGA

<BGA Pin Layout>

Pin No	I/O	I/O Type	Signal Code	Pin No	I/O	I/O Type	Signal Code	Pin No	I/O	I/O Type	Signal Code
A01	I/O	*6	TIO06	D02	I/O	*6	TIO09	K13	I/O	*3	D23
A02	I/O		DD+	D03	I/O	*6	TIO10	L01	O	*5	$\overline{\text{PCONT}}$
A03	—	—	V_{CC}	D11	I/O	*3	D5	L02	O	*5	PDCTL
A04	I/O	*6	TIO05	D12	I/O	*3	D8	L03	I/O	*6	TIO17
A05	I	*4	A2	D13	—	—	GND	L04	O	*5	$\overline{\text{DREQ0}}$
A06	I	*4	A5	E01	I	*4	VP	L05	O	*5	$\overline{\text{DREQ1}}$
A07	I	*4	A8	E02	I	*4	VM	L06	O	*6	$\overline{\text{WAIT}}$
A08	I	*4	A11	E03	O	*5	SUSPEND	L07	I	*4	$\overline{\text{RD}}$
A09	I/O	*6	TIO02	E11	—	—	V_{CC}	L08	I	*4	BUS_CLK
A10	I/O	*6	TIO01	E12	I/O	*3	D9	L09	I/O	*3	D31
A11	—	—	V_{CC}	E13	I/O	*3	D10	L10	I/O	*3	D28
A12	I/O	*3	D1	F01	I	*2	$\overline{\text{TMD0}}$	L11	—	—	V_{CC}
A13	I/O	*3	D2	F02	O	*5	SPEED	L12	—	—	GND
B01	I/O	*6	TIO07	F03	I	*2	$\overline{\text{TMD1}}$	L13	I/O	*3	D24
B02	I/O		DD–	F11	I/O	*3	D11	M01	I/O	*6	TIO14
B03	—	—	GND	F12	I/O	*3	D12	M02	I/O		HD+
B04	I/O	*6	TIO04	F13	I/O	*3	D13	M03	—	—	V_{CC}
B05	I	*4	A3	G01	O	*5	MODE	M04	O		XOUT
B06	I	*4	A6	G02	O	*5	VPO	M05	I/O	*6	TIO18
B07	I	*4	A9	G03	O	*5	VMO/FSEO	M06	I	*4	$\overline{\text{DACK0}}$
B08	I	*4	A12	G11	I/O	*3	D14	M07	I	*4	$\overline{\text{DACK1}}$
B09	I	*2	32/$\overline{\text{16SEL}}$	G12	I/O	*3	D15	M08	I	*4	$\overline{\text{CS}}$
B10	I/O	*6	TIO00	G13	—	—	V_{CC}	M09	I	*4	$\overline{\text{BS}}$
B11	I/O	*3	D0	H01	I	*2	$\overline{\text{TMD2}}$	M10	—	—	GND
B12	I/O	*3	D4	H02	I	*2	$\overline{\text{TMD3}}$	M11	I/O	*3	D30
B13	I/O	*3	D3	H03	I/O	*6	TIO11	M12	I/O	*3	D27
C01	O	*5	$\overline{\text{PUCTL}}$	H11	—	—	GND	M13	I/O	*3	D26
C02	I	*4	SVBUS	H12	I/O	*3	D16	N01	I/O	*6	TIO15
C03	I	*4	RCV	H13	I/O	*3	D17	N02	I/O		HD–
C04	I	*4	A1	J01	O	*5	$\overline{\text{OE}}$	N03	—	—	GND
C05	I	*4	A4	J02	I	*2	D/$\overline{\text{H}}$	N04	I		XIN
C06	I	*4	A7	J03	I/O	*6	TIO12	N05	I/O	*6	TIO16
C07	I	*4	A10	J11	I/O	*3	D18	N06	O	*5	$\overline{\text{INTR}}$
C08	I/O	*6	TIO03	J12	I/O	*3	D19	N07	I	*4	$\overline{\text{DRAK0}}$
C09	I	*2	Little/$\overline{\text{Big}}$	J13	I/O	*3	D20	N08	I	*4	$\overline{\text{DRAK1}}$
C10	I	*1	$\overline{\text{RESET}}$	K01	I	*2	ExtBuffENB	N09	I	*4	RD/$\overline{\text{WR}}$
C11	—	—	GND	K02	I/O	*6	TIO13	N10	I/O	*6	TIO19
C12	I/O	*3	D6	K03	I	*4	$\overline{\text{Over\_Current}}$	N11	—	—	V_{CC}
C13	I/O	*3	D7	K11	I/O	*3	D21	N12	I/O	*3	D29
D01	I/O	*6	TIO08	K12	I/O	*3	D22	N13	I/O	*3	D25

Note: I/O types
*1: TTL Schmitt Input Buffer with 4X Drive,
*2: TTL Input Buffer with 4X Drive/50K Pull Up,
*3: I/O Buffer with TTL Input/4mA Output,
*4: TTL Input Buffer with 4X Drive,
*5: Push Pull Output Buffer with 4mA Drive,
*6: I/O Buffer with TTL Input/4mA Output 50K Pull Up

PIN DESCRIPTIONS

Description of each pin is given below.
Pins having "–" attached to their symbols are "active low" pins and those having nothing attached are "active high" pins.

Symbol	Pin Count	I/O	Description
XIN, XOUT	2	—	Input/output pins for externally connecting a crystal oscillator.
BUS_CLK	1	I	Input pin for synchronizing clock from a microcontroller. Maximum frequency: 33 MHz
Little/$\overline{\text{Big}}$	1	I	Input pin for specifying data bus specification of microcontroller interface. With an internal 50 kΩ pull-up resistor. "L": Big endian, "H": Little endian
D/$\overline{\text{H}}$	1	I	Input pin for selecting host/device function. With an internal 50 kΩ pull-up resistor. "L": Host, "H": Device Enabled only after the $\overline{\text{RESET}}$ signal is active (this signal is disabled after setting by an internal register).
$\overline{\text{PCONT}}$	1	O	USB bus power supply control (ON/OFF) output pin in host mode. "L": ON, "H": OFF
$\overline{\text{Over\_Current}}$	1	I	USB bus power supply trouble (overcurrent etc.) informing input pin in host mode. "L": abnormal, "H": normal
ExtBuffENB	1	I	External USB transceiver enable control input pin. With an internal 50 kΩ pull-up resistor. "L": External USB transceiver disable (not used) "H": External USB transceiver enable (used) When using this LSI in the device controller mode with this pin in the enabled state ("H"), set the bit 00 of the common connector used/not used select register (address: 04h) to common connector used "1" state. The operation cannot be guaranteed if the common connector not used setting "0" is made. See the descriptions of the ConSel register for details.
SVBUS	1	I	USB bus power supply sense input pin. It is necessary to provide an input to this pin that disables the USB power supply sense except only when the power supply of the USB interface is truly valid. "L": USB Interface VBUS disabled, "H": USB interface VBUS enabled
HD+, HD–	2	I/O	USB bus input/output pins for host/device. Since this pin will be in the input state during the device controller mode when the common connector not used setting (bit 00 is "0" in the common connector used/not used select register (address: 04h) and when the external USB transceiver used setting (ExtBuffEN = "H") is made, tie the input level of this pin to "H" or "L".
DD+, DD–	2	I/O	USB bus input/output pins exclusively for device. When the host controller mode has been set, or when the external USB transceiver used setting (ExtBuffEN = "H") has been made, tie the input level of this pin to "H" or "L" since this pin will be in the input state.
$\overline{\text{PUCTL}}$	1	O	Output pin for pull-up control of USB bus input/output pins in device mode. "L": pull-up enable, "H": pull-up disable
PDCTL	1	O	Output pin for pull-down control of USB bus input/output pins in host mode. "H": pull-down enable, "L": pull-down disable
D31:D16	16	I/O	Input/output pins for data bus. An invalid data will be output when read out in the 16-bit mode. Also, since these pins will be in the input state at all times other than reading out, keep the input levels of these pins tied to "H" or "L".
D15:D0	16	I/O	Input/output pins for data bus. Enabled in 16-bit mode.

Symbol	Pin Count	I/O	Description
32/$\overline{16SEL}$	1	I	Input pin for selecting 32-bit/16-bit data bus width. With an internal 50 kΩ pull-up resistor. "L": 16-bit, "H": 32-bit
A12:A1	12	I	Input pins for address from microcontroller. (For accessing an internal 4 KB RAM and an internal register)
$\overline{CS}$	1	I	Input pin for chip select signal from microcontroller.
RD/$\overline{WR}$	1	I	Input pin for RD/$\overline{WR}$ signal from microcontroller.
$\overline{RD}$	1	I	Input pin for $\overline{RD}$ signal from microcontroller.
$\overline{INTR}$ (*)	1	O	Output pin for interrupt request to microcontroller.
$\overline{WAIT}$	1	O	Output pin for wait request to microcontroller. Set the microcomputer so that this signal is sampled on the falling edge of the "BUS_CLK" signal.
$\overline{RESET}$	1	I	Reset input pin.
$\overline{DREQ0}$ (*)	1	O	Output pin for DMA request to microcontroller (used when HOST and Device are operating).
$\overline{DACK0}$ (*)	1	I	Input pin used to indicate that DMA cycling is in progress (used when HOST and Device are operating). When not used, tie the input level of this pin to the inactive level.
$\overline{DRAK0}$ (*)	1	I	Input pin used to indicate that DMA has received the "$\overline{DREQ0}$ signal" (used when HOST and Device are operating). When connecting to a microcontroller without the "$\overline{DRAK}$" pin, connect to the "$\overline{DACK}$" pin. When not used, tie the input level of this pin to the inactive level.
$\overline{DREQ1}$ (*)	1	O	Output pin for DMA request to microcontroller (used when Device is operating).
$\overline{DACK1}$ (*)	1	I	Input pin used to indicate that DMA cycling is in progress (used when Device is operating). When not used, tie the input level of this pin to the inactive level.
$\overline{DRAK1}$ (*)	1	I	Input pin used to indicate that DMA has received the "$\overline{DREQ0}$ signal" (used when Device is operating). When connecting to a microcontroller having no "$\overline{DRAK}$" pin, connect to the "$\overline{DACK}$" pin. When not used, tie the input level of this pin to the inactive level.
$\overline{BS}$	1	I	Input pin for bus start ($\overline{BS}$) signal from microcontroller.
$\overline{OE}$ (**)	1	O	External USB transceiver input/output selector output pin. "L": transmit mode (ML60841 to USB), "H": receive mode (USB to ML60841)
MODE (**), VPO (**), VMO/FSEO (**)	3	O	Pins for output to an external USB transceiver. The ML60841 outputs the status shown in the following Result column in combinations of 3 signal lines, as shown below. MODE VPO VMO/FSEO Result 0 0 0 Logic "0" 0 0 1 SE0# 0 1 0 Logic "1" 0 1 1 SE0# 1 0 0 SE0# 1 0 1 Logic "0" 1 1 0 Logic "1" 1 1 1 Illegal code
RCV (**)	1	I	Input pin for receive data from an external USB transceiver. When not used, tie the input level of this pin to "H" or "L".

(*) Active level can be changed by setting register.
(**) Signal lines for connecting the host/device common USB ports to an external USB transceiver.

Symbol	Pin Count	I/O	Description
VP (**), VM (**)	2	I	Input pins for external USB transceiver. The ML60841 operates by judging the status shown in the following Result column in combinations of 2 signal lines, as shown below. VP VM Result 0 0 SE0# 0 1 Low speed 1 0 Full speed 1 1 Error When not used, tie the input level of this pin to "H" or "L".
SUSPEND (**)	1	O	Output pin used to notify an external USB transceiver that the SUSPEND status will be entered when the USB bus is not used. "L": specify non-SUSPEND, "H": specify SUSPEND
SPEED (**)	1	O	Output pin used to notify an external USB transceiver of the USB bus speed. "L": low speed, "H": full speed
TIO00 to 19	20	I/O	Input/output pins for testing (normally left unconnected).
$\overline{TMD0}$ to $\overline{3}$	4	I	Input pins for testing. With an internal 50 kΩ pull-up resistor (normally left unconnected).
V_{CC}	7	—	V_{CC}
GND	7	—	GND
Total	120		

(**) Signal lines for connecting the host/device common USB ports to external USB transceiver.

REGISTER

The registers of the ML60841 are divided mainly into common registers, host controller registers and device controller registers.

Approximate categorywise map of addresses is as follows.

Category	Offset	Remarks
Common register	000h–0FFh	
Host controller register	100h–2FFh	OpenHCI control register included
Device controller register	200h–3FFh	
Host controller memory	1000h–1FFFh	Internal 4 KB RAM

Mapping of each register is shown on next page onward.

The meanings of "R", "W", and "R/W" are the following in the register descriptions given hereafter.

R: Only read operation is valid. Unless specifically described in the descriptions of the different registers, there will be no effect on the operations even when a "0" or a "1" is written to the concerned part.

W: Only write operation is valid. Unless specifically described in the descriptions of the different registers, an uncertain data will be returned when the concerned part is read out.

R/W: Both read and write operations are valid.

In the following description of registers, operation cannot be guaranteed when an unspecified address is accessed. Therefore, control the LSI operation so that such an address will not be accessed. Moreover, since operation cannot be guaranteed when a "1" is written in an unspecified field of specified registers, write a "0" in an unspecified field when writing in other fields.

Also, if the device controller registers (registers 300h to 3FFh) are accessed during operations as a host controller, or if the host controller registers (registers 100h to 2FFh) and host controller memories (1000h to 1FFFh) are accessed during operations as a device controller, the operations of the function being executed currently (host controller function in the former case and device controller function in the latter case) cannot be guaranteed and hence controls should be implemented so as not to make such access.

The meaning of "Don't Care" and "X" is as follows.

When writing: There is no affect on the operation of the ML60841 by writing either "0" or "1".

When reading: Undefined data is read.

In the ML60841, it is possible to specify combinations of big-endian or little-endian and 16-bit bus width or 32-bit bus width as the data bus specifications. Depending on these combinations, there are some restrictions on the register (memory) accesses of different blocks. Implement controls to observe the following restrictions. The operations cannot be guaranteed if these restrictions are not observed.

① When using a 16-bit bus width, carry out one set of the two accesses to the offset address and offset address +2 when accessing the host controller registers (including the OpenHCI operation register) and the host controller memories.
The data is written to the concerned register or memory when the 32-bit data becomes ready within the ML60841.

② When using a 16-bit bus width, write data of bits 15:0 to the offset address of a common register in the case of both the big-endian and the little-endian modes. In other words, when using a 16-bit bus width, carry out writes using instructions that handle 16-bit data.

Types and Mapping of Common Registers

Offset	Symbol	Name
000h	HostDevSel	Host/Device select register
004h	ConSel	Common connector use/no use select register
008h	PolModeSel	Pin polarity and USB transceiver mode select register
00Ch	DmaMode0	DMA0 mode setting register
010h	DmaMode1	DAM1 mode setting register
014h	HdVbusIntStt	Interrupt status register for a status change in the D/$\overline{H}$ pin and VBUS
018h	HdVbusIntMask	Interrupt mask register for a status change in the D/$\overline{H}$ pin and VBUS
01Ch	SrstPDownCtl	Software reset/Power down control register

- HostDevSel Register (000h)

Bits	Field	Reset	R/W
31:02	Don't care	X	X
01	Indicates that the ML60841 is under internal processing when changing the D/$\overline{H}$ pin setting or bit 00 set value. 0: Not under internal processing, 1: Under internal processing	0b	R
00	Specifies Host or Device function. 0: Host, 1: Device	Note	R/W

Note) Depends on the status of the D/$\overline{H}$ pin.

[Description]

By writing in bit 00, host/device selection is possible irrespective of the D/$\overline{H}$ pin status. For example even if D/$\overline{H}$ pin is at device setting, it is possible to operate as host by writing "0000h" in this register. However, after writing in this register, change of D/$\overline{H}$ pin setting is ignored. (An interrupt is generated due to a change in the state of the D/$\overline{H}$ pin the status changes.)

When change occurs in value set in bit 00, the ML60841 resets both the host block and the device block internal to the LSI. During reset, bit 01 gets set at "1" and then becomes "0" when reset is completed and operation becomes possible. Therefore, after change in setting, it is necessary for the microcontroller to execute initialization. Start initialization after bit 01 has become "0". See "Precautions in control of ML60841" for details related to the timing of starting the initialization processing.

- ConSel Register (004h)

Bits	Field	Reset	R/W
31:01	Don't care	X	X
00	Specifies common connector use/no use. 0: Common connector not used 1: Common connector used	0b	R/W

[Description]

When common connector is used, the operation becomes as follows.

- In Host mode: the HD+ and HD– pins are designated for host.
- In Device mode: the HD+ and HD– pins are designated for device.

When common connector is not sued, the operation becomes as follows.

- In Host mode: the HD+ and HD– pins are designated for host.
- In Device mode: the DD+ and DD– pins are designated for device.

Set this bit to "1" to select the common connector used state when the external transceiver used setting has been made (ExtBuffEN= "H") when using the device controller operations. The operations cannot be guaranteed when the common connector not used setting "0" is made here. The relationship between the setting of this bit and the USB buffer status is shown below.

Operating mode	Pin name	This bit = "0" (Common connector not used)		This bit = "1" (Common connector used)	
		ExtBuffENB="L"	ExtBuffENB="H"	ExtBuffENB="L"	ExtBuffENB="H"
Host controller	HD+/HD–	Valid	Power down state	Valid	Power down state
	DD+/DD–	Power down state	Power down state	Power down state	Power down state
	$\overline{OE}$, VMO/FSEO, VPO, RCV, VM, VP, SUSPEND (External USB buffer pins)	Unused state*	Valid	Unused state*	Valid
Device controller	HD+/HD–	Power down state	Illegal specification	Valid	Power down state
	DD+/DD–	Valid	Illegal specification	Power down state	Power down state
	$\overline{OE}$, VMO/FSEO, VPO, RCV, VM, VP, SUSPEND (External USB buffer pins)	Unused state*	Illegal specification	Unused state*	Valid

* The unused state is the following:
  - Input pin: Tied to "H" or "L" level
  - Output pin: Inactive level output (excepting the MODE, SPEED and SUSPEND pins)
  - SUSPEND pin: "H" level output

- PolModeSel register (008h)

Bits	Field	Reset	R/W
31:08	Don't care	X	X
07	Specifies the mode signal to an external USB transceiver. 0: MODE0, 1: MODE1	0b	R/W
06:04	Don't care	X	X
03	Specifies the polarity of the INTR signal. 0: Active Low, 1: Active High	0b	R/W
02	Specifies the polarity of the DACK signal. 0: Active Low, 1: Active High	0b	R/W
01	Specifies the polarity of the DRAK signal. 0: Active Low, 1: Active High	0b	R/W
00	Specifies the polarity of the DREQ signal. 0: Active Low, 1: Active High	0b	R/W

[Description]

Bit 07 sets the value ("0" = "L" or "1" = "H") output to the MODE pin. Bits 03 to 00 set active level of each pin ($\overline{\text{INTR}}$; DACK0, DACK1; DRAK0, DRAK1; DREQ0, DREQ1). Setting "0" makes the level active low and setting "1" makes it active high. Two each of "DACK0/1", "DRAK0/1" and "DREQ0/1" are provided for each DMA channel, but by setting this register the two get set to the same active level.

- DmaMode0 Register (00Ch), DmaMode1 Register (010h)

Bits	Field	Reset	R/W
31:04	Don't care	X	X
03:02	Specifies the data bit length for one time of "$\overline{\text{DREQ}}$". 11: Illegal specification 10: 32 bits 1 transfer/32 bits, 2 transfers/16 bits 01: Illegal specification 00: Illegal specification Set "10" in this field. Operation is not guaranteed if a value other than "10" is set.	10b	R/W
01	Don't care.	X	X
00	Don't care.	X	X

[Description]

The settings of bits 03:02 and the operations are as follows:

- When the data bus width is 32 bits ("32/$\overline{16}$ $\overline{\text{SEL}}$" pin is "H")

In one DMA transfer, 32 bits of data are transferred by asserting "$\overline{\text{DREQ}}$" once.

- When the data bus width is 16 bits ("32/$\overline{16}$ $\overline{\text{SEL}}$" pin is "L")

16 bits of data are transferred twice by asserting "$\overline{\text{DREQ}}$" once in one DMA transfer. The address in this case will have to be made 'n+2' during the second transfer if the address during the first transfer was 'n' (which should be a multiple of 4).

For example, when transferring data to or from the host controller register FIFO ACC (20Ch), it is necessary to specify the addresses repeatedly in the sequence 20Ch → 20Eh → 20Ch → 20Eh → ….. 20Eh → 20Ch → 20Eh (the starting address is 20Ch and the ending address is 20Eh). Similarly, when transferring data to or from the device controller EP1 transmit/receive FIFO register (388h), it is necessary to specify the addresses

repeatedly in the sequence 388h → 38Ah → 388h → 38Ah → …. → 388h → 38Ah (the starting address is 388h and the ending address is 38Ah).

- HdVbusIntStt Register (014h)

Bits	Field	Reset	R/W
31:10	Don't care	X	X
09	Indicates the status of the SVBUS pin. 0: VBUS inactive, 1: VBUS active	Note 1	R
08	Indicates the status of the D/$\overline{H}$ pin. 0: Host, 1: Device	Note 2	R
07:02	Don't care	X	X
01	Indicates the status change of the SVBUS pin. 0: No change, 1: Change Clear this bit by writing a "1" in it.	0b	R/W
00	Indicates the status change of the D/$\overline{H}$ pin. 0: No change, 1: Change Clear this bit by writing a "1" in it.	0b	R/W

(Note 1) Goes into the status of the SVBUS pin.
(Note 2) Goes into the status of the D/$\overline{H}$ pin.

[Description]

Bit 09 reflects the status of the SVBUS pin.

Bit 08 reflects the status of the D/$\overline{H}$ pin. The status may not agree with the current operating mode (Host/Device), because this bit only reflects the D/$\overline{H}$ pin status even if setting is changed by the HostDevSel register. For example, when the D/$\overline{H}$ pin is at "H" level (Device setting) and "0 (Host function specified)" is written in bit 00, this bit 08 indicates "1 (Device)", but the ML60841 is in the Host operating mode. To know the current operating mode, read the HostDevSel register.

Bit 01 gets set to "1" when the status of the SVBUS pin changes. At this time, if the bit corresponding to the HdVbusIntMask register is interrupt enable, an interrupt occurs. This bit can be cleared by writing a "1" in it.
Bit 01 gets set to "1" when the D/$\overline{H}$ pin status changes. At this time, if the bit corresponding to the HdVbusIntMask register is interrupt enable, an interrupt occurs. This bit can be cleared by writing a "1". When a change occurs in the state of the SVBUS pin before this bit is cleared (that is, multiple causes of this interrupt are present), all the interrupt causes will be cleared when a "1" is written to this bit (the causes are not queued). For example, if the state of the SVBUS pin has changed several times before this bit has been cleared, the value of bit 09 after the interrupt may become the same as the value of bit 09 before the interrupt.
Therefore, it is necessary that the microcomputer reads the value of bit 09 and confirms the status of SVBUS.

Bit 00 will be set to "1" when there is a change in the state of the D/$\overline{H}$ pin. An interrupt is generated at this time if the interrupt has been enabled in the corresponding bit of the HdVbusIntMask register. When a "1" has been set in this bit, this bit can be cleared by writing a "1". When a change occurs in the state of the D/$\overline{H}$ pin before this bit is cleared (that is, multiple causes of this interrupt are present), all the interrupt causes will be cleared (the causes are not queued) when a "1" is written to this bit.
For example, if the state of the D/$\overline{H}$ pin has changed several times before this bit has been cleared, the value of bit 08 after the interrupt may become the same as the value of bit 08 before the interrupt.
Therefore, it is necessary that the microcomputer reads the value of bit 08 and confirms the status of D/$\overline{H}$.

- HdVbusIntMask Register (018h)

Bits	Field	Reset	R/W
31:02	Don't care	X	X
01	Specifies whether to generate an interrupt or not when the SVBUS pin status changes. 0: Interrupt disable 1: Interrupt enable	0b	R/W
00	Specifies whether to generate an interrupt or not when the D/$\overline{H}$ pin status changes. 0: Interrupt disable 1: Interrupt enable	0b	R/W

[Description]

Bits 00 and 01 set whether or not to generate an interrupt when the status of bits 01 and 00 of the HdVbusIntStt register changes to "1".

- SRstPDownCtl Register (01Ch)

Bits	Field	Reset	R/W
31:02	Don't care	X	X
01	When a "1" is written to this bit, internal processings are executed and the power down state is entered into within a maximum of 3 μs. Writing a "0" in this bit has no meaning (that is, has no effect on the operation). It is necessary to give an input of the "$\overline{\text{RESET}}$" signal in order to recover from the power down mode.	0b	W
00	A reset of the host controller is instructed by writing a "1" to this bit. Writing a "0" in this bit has no meaning (that is, has no effort on the operation).	0b	W

[Description]
When a "1" is written to bit 01, after the completion of the write access cycle, power down processings are executed within the ML60841 and the power down state is entered into within a maximum of 3 μs after writing the "1". In concrete terms, the power down state is the following:

- The oscillator circuit ("XIN" and "XOUT" pins) has stopped operating and the clock signal is not supplied to the internal circuits of the ML60841.
- The clock signal supplied to the microcomputer ("BUS_CLK" pin) is not supplied to the internal circuits of the ML60841.
- The receiver part of the USB transceiver ("HD+", "HD–", "DD+", and "DD–" pins) goes into the inactive state. In this state, the ML60841 does not detect even when there is any change in the status of the USB bus.
- The suspend output (an "H" level output at the "SUSPEND" pin) is given to the external USB transceiver.

In the power down state, since the USB transceiver will be in the disabled state, it will not be possible to verify changes in the status of the USB bus.
Therefore, it is possible that no matching is made with the operations of the other equipment connected to the USB interface bus. Carry out the writing of a "1" in this bit when no other devices have been connected to the USB interface, and it is permissible to stop the operations completely. In concrete terms, write a "1" to this bit only when the power down mode has been instructed intentionally upon an operator intervention.

Regarding all pins other than the above, the input or output states are not being controlled after the completion of the access cycle in which a "1" was written to this bit. Give a definite level input "H" or "L" to the input pins (including input/output pins in the input state).
All subsequent operations cannot be guaranteed if a "1" is written to this bit while the ML60841 is operating (during USB communication, DMA transfer, etc.)

After writing a "1" in this bit, it is necessary to input the "$\overline{\text{RESET}}$" signal in order to return from the power down state. The operations cannot be guaranteed if the ML60841 is accessed during the period immediately after writing a "1" in this bit until a "$\overline{\text{RESET}}$" input is given. See the section on the "Precautions in Control of the ML60841" for details of the processing after the "$\overline{\text{RESET}}$" input is given.

When a "1" is written in the bit 00, the ML60841 resets the host controller section within the LSI. In concrete terms, all the registers in the host controller section will be cleared to their initial values. The common section and the device controller section will not be reset. (It is possible to carry out a software reset of the device controller section by writing a "1" in bit 00 of the system control register.)
During this reset operation, the bit 01 of the HostDevSel register is set to "1" and is reset to "0" when the resetting has been completed and normal operations can be started. Therefore, it is necessary for the microcomputer to execute the initialization processing after bit 01 has become "0".

In addition, upon this resetting, the ML60841 resets the USB bus.
Therefore, the microcomputer will also have to carry out initializations of all the devices connected to the USB bus.

Types and Mapping of Host Controller Registers - 1

Offset	Symbol	Name
200h	HOST CTL	Host Control register
204h	STT/TRNS CNT	Status, RD/WR FIFO transfer length register
208h	Host data transfer REQ	Host data transfer request register
20Ch	FIFO ACC	FIFO access register
210h	RAM ADR	Internal RAM address setting register

For details concerning the registers mentioned below, refer to the section of NOTES ON THE ML60841 CONTROL.

- HOST CTL Register (200h)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
31:04	Don't care	X	X	X
03	Specifies DMA transfer. 0: Enables DMA transfer, 1: Disables DMA transfer	1b	R/W	R
02	Host data transfer interrupt mask 0: No mask, 1: Mask	1b	R/W	R
01	OpenHCI core interrupt mask 0: No mask, 1: Mask	1b	R/W	R
00	Don't care	X	X	X

- STT/TRNS CNT Register (204h)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
31:27	Don't care	X	X	X
26:24	Don't care	0b	R	R/W
23:19	Don't care	X	X	X
18:16	Don't care	0b	R	R/W
15:02	Don't care	X	X	X
01 (*)	Indicates an interrupt request from the host core. 0: No interrupt request, 1: Interrupt request	0b	R/W	R/W
00 (*) (**)	Indicates an interrupt request relating to host data transfer. 0: No interrupt request, 1: Interrupt request	0b	R/W	R/W

(*) Cleared by writing a "1" from the microcontroller.
(**) DREQ is generated when interrupt clear is triggered and the HOST CTL register bit 03 is "0 (DMA)".

- Host Data Transfer REQ Register (208h)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
31:02	Indicates the transfer start address when an interrupt relating to host data transfer is generated.	0h	R	R/W
01	Don't care	X	X	X
00	Indicates the direction of transfer when an interrupt relating to host data transfer is generated. 1: MCU (MEM) to ML60841 0: ML60841 to MCU (MEM)	0b	R	R/W

- FIFO ACC Register (20Ch)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
31:00	FIFO access data	X	R/W	R/W

Specify the address of this register as the target address during DMA transfer.
The operations cannot be guaranteed if there is an access from the microcomputer to the FIFO ACC register during DMA transfer.

- RAM ADR Register (210h)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
31:12	Sets the leading address for the internal RAM of the system.	0	R/W	R
11:00	Don't care	X	X	X

[Description]

This register is activated by write from a microcontroller. If write from the microcontroller is not executed, all transfer request addresses from the host core are judged to be outside the internal RAM address range (becomes an interrupt factor with regard to host data transfer).

Therefore, even when it is used as the same value as the initial value "0", it is necessary to write a "0".

Types and Mapping of Host Controller Registers - 2

Conforming to OpenHCI specifications, the registers control the Host controller functions of the ML60841.

- USB HOST Operational Registers Summary

Offset	Register Name	Offset	Register Name
100h	HcRevision	12Ch	HcBulkCurrentED
104h	HcControl	130h	HcDoneHead
108h	HcCommandStatus	134h	HcFmInterval
10Ch	HcInterruptStatus	138h	HcFmRemaining
110h	HcInterruptEnable	13Ch	HcFmNumber
114h	HcInterruptDisable	140h	HcPeriodicStart
118h	HcHCCA	144h	HcLSThreshold
11Ch	HcPeriodCurrentED	148h	HcRhDescriptorA
120h	HcControlHeadED	14Ch	HcRhDescriptorB
124h	HcControlCurrentED	150h	HcRhStatus
128h	HcBulkHeadED	154h	HcRhPortStatus

The outline is given below. See “OpenHCI (Release: 1.0a)” for details..

- HcRevision Register (100h)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
7:0	Revision (REV)	10h	R	R

- HcControl Register (104h)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
10	Remote Wakeup Enable (RWE)	0b	R/W	R
9	Remote Wakeup Connected (WC)	0b	R/W	R/W
8	Interrupt Routing (IR)	0b	R/W	R
7:6	Host Controller Functional State (HCFS)	00b	R/W	R/W
5	Bulk List Enable (BLE)	0b	R/W	R
4	Control List Enable (CLE)	0b	R/W	R
3	Isochronous Enable (IE)	0b	R/W	R
2	Periodic List Enable (PLE)	0b	R/W	R
1:0	Control Bulk Service Ratio (CBSR)	00b	R/W	R

- HcCommandStatus Register (108h)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
17:16	Scheduling Overrun Count (SOC)	00b	R	R/W
3	Ownership Change Request (OCR)	0b	R/W	R/W
2	Bulk List Filled (BLF)	0b	R/W	R/W
1	Control List Filled (CLF)	0b	R/W	R/W
0	Host Controller Reset (HCR)	0b	R/W	R/W

Note: Always set the HCR bit of bit 0 to “0”. Operation is not guaranteed if set to “1”. To execute software reset, use the SRstPDownCtl register.

- HcInterruptStatus Register (10Ch)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
30	Ownership Change (OC)	0b	R	R
6	Root Hub Status Change (RHSC)	0b	R/W	R/W
5	Frame Number Overflow (FNO)	0b	R/W	R/W
4	Unrecoverable Error (UE)	0b	R/W	R/W
3	Resume Detected (RD)	0b	R/W	R/W
2	Start of Frame (SF)	0b	R/W	R/W
1	Writeback Done Head (WDH)	0b	R/W	R/W
0	Scheduling Overrun (SO)	0b	R/W	R/W

Note: The OC bit of bit 30 is a read only bit fixed to “0”. Therefore this bit does not change even when bit 3 (OCR bit) of the HcCommand Status register is set to “1”. Moreover, no interrupt is generated even when bit 8 (IR bit) of the HcControl register is set to “1”.

- HcInterruptEnable Register (110h)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
31	Master Interrupt Enable (MIE)	0b	R/W	R
30	Ownership Change (OC)	0b	R/W	R
6	Root Hub Status Change (RHSC)	0b	R/W	R
5	Frame Number Overflow (FNO)	0b	R/W	R
4	Unrecoverable Error (UE)	0b	R/W	R
3	Resume Detected (RD)	0b	R/W	R
2	Start of Frame (SF)	0b	R/W	R
1	Writeback Done Head (WDH)	0b	R/W	R
0	Scheduling Overrun (SO)	0b	R/W	R

- HcInterruptDisable Register (114h)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
31	Master Interrupt Enable (MIE)	0b	R/W	R
30	Ownership Change (OC)	0b	R/W	R
6	Root Hub Status Change (RHSC)	0b	R/W	R
5	Frame Number Overflow (FNO)	0b	R/W	R
4	Unrecoverable Error (UE)	0b	R/W	R
3	Resume Detected (RD)	0b	R/W	R
2	Start of Frame (SF)	0b	R/W	R
1	Writeback Done Head (WDH)	0b	R/W	R
0	Scheduling Overrun (SO)	0b	R/W	R

- HcHCCA Register (118h)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
31:8	Host Controller Communications Area (HCCA) Base Address	0h	R/W	R

- HcPeriodCurrentED Register (11Ch)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
31:4	Period Current ED (PCED) Base Address	0h	R	R/W

- HcControlHeadED Register (120h)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
31:4	Control Head ED (CHED) Base Address	0h	R/W	R

- HcControlCurrentED Register (124h)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
31:4	Control Current ED (CCED) Base Address	0h	R/W	R/W

- HcBulkHeadED Register (128h)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
31:4	Bulk Head ED (BHED) Base Address	0h	R/W	R

- HcBulkCurrentED Register (12Ch)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
31:4	Bulk Current ED (BCED) Base Address	0h	R/W	R/W

- HcDoneHead Register (130h)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
31:4	Done Head ED (DH) Base Address	0h	R	R/W

- HcFmInterval Register (134h)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
31	Frame Interval Toggle (FIT)	0b	R/W	R
30:16	FS Largest Data Packet (FSMPS)	0h	R/W	R
13:0	Frame Interval (FI)	2EDFh	R/W	R

- HcFmRemaining Register (138h)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
31	Frame Remaining Toggle (FRT)	0b	R	R/W
13:0	Frame Remaining (FR)	0h	R	R/W

Note: Writing in this register is enabled. If written in this register, the operation is not guaranteed.

- HcFmNumber Register (13Ch)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
15:0	Frame Number (FN)	0h	R	R/W

Note: Writing in this register is enabled. If written in this register, the operation is not guaranteed.

- HcPeriodicStart Register (140h)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
13:0	Periodic Start (PS)	0h	R/W	R

- HcLSThreshold (144h)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
11:0	LS Threshold	628h	R/W	R

- HcRhDescriptorA Register (148h)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
31:24	Power On to Power Good Time (POTPGT)	32h	R/W	R
12	No Overcurrent Protection (NOCP)	0b	R/W	R
11	Overcurrent Protection Mode (OCPM)	0b	R/W	R
10	Device Type (DT)	0b	R	R
9	No Power Switching (NPS)	0b	R/W	R
8	Power Switching Mode (PSM)	0b	R/W	R
7:0	Number Downstream Ports (NDP)	1h	R	R

- HcRhDescriptorB Register (14Ch)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
17	Port Power Control Mask (PPCM)	0b	R/W	R
1	Device Removable (DR)	0b	R/W	R

- HcRhStatus Register (150h)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
31	Clear Remote Wakeup Enable (CRWE)	0b	W	R
17	Overcurrent Indicator Change (OCIC)	0b	R/W	R/W
16	Local Power Status Change (LPSC)	0b	R/W	R
15	Device Remote Wakeup Enable (DRWE)	0b	R/W	R
1	Overcurrent Indicator (OCI)	0b	R	R/W
0	Local Power Status (LPS)	0b	R/W	R

- HcRhPortStatus Register (154h)

Bits	Field	Reset	HCD (Host Controller Driver)	HC (Host Controller)
20	Port Reset Status Change (PRSC)	0b	R/W	R/W
19	Port Overcurrent Indicator Change (OCIC)	0b	R/W	R/W
18	Port Suspend Status Change (PSSC)	0b	R/W	R/W
17	Port Enable Status Change (PESC)	0b	R/W	R/W
16	Connect Status Change (CSC)	0b	R/W	R/W
9	Low-speed Device Attached (LSDA)	0b	R/W	R/W
8	Port Power Status (PPS)	0b	R/W	R/W
4	Port Reset Status (PRS)	0b	R/W	R/W
3	Port Overcurrent Indicator (POCI)	0b	R/W	R/W
2	Port Suspend Status (PSS)	0b	R/W	R/W
1	Port Enable Status (PES)	0b	R/W	R/W
0	Current Connect Status (CCS)	0b	R/W	R/W

Memory for Host Controller

Offset	Description	Reset	HCD (Host Controller Driver)	HC (Host Controller)
1000h –1FFFh	4 KB (1K double word) HCCA, ED/TD buffer memory	X	R/W	R/W

This RAM must be accessed in 32 bits unit. When accessing this RAM by a 16-bit wide bus, it is necessary, for example, to continuously access two accesses of 1000h and 1002h as one set.

Types and Mapping of Device Controller Registers - 1

Category	Offset	R/W	Offset Address Low-Order 2 Bits			
			11	10	01	00
FIFO	380h	W	EP0 transmit FIFO (EP0TXFIFO)			
	384h	R	EP0 receive FIFO (EP0RXFIFO)			
	388h	R/W	EP1 transmit/receive FIFO (EP1FIFO)			
	38Ch	R/W	EP2 transmit/receive FIFO (EP2FIFO)			
	390h	R/W	EP3 transmit/receive FIFO (EP3FIFO)			
	394h	R/W	EP4 transmit/receive FIFO (EP4FIFO)			
	398h	R/W	EP5 transmit/receive FIFO (EP5FIFO)			
Common	300h	R	wValue•bRequest•BmRequest Type setup register (SETUP0)			
	304h	R	wLengtht•wIndex setup register (SETUP1)			
	308h	R/W			Device address register (DVCADR)	
	30Ch	R			Frame number register (FRAME MSB)	
	310h	R/W			Interrupt status register (INTSTAT)	
	314h	R/W			Interrupt enable register (INTENBL)	
	318h	R/W			System control register (SYSCON)	
DMA	320h	R/W			DMA0 control register (DMA0CON)	
	324h	R/W			DMA1 control register (DMA1CON)	

Notes:

1. Access to unspecified address sections is meaningless.
2. Regarding access to FIFO/registers:
   - In the case of 32 bits: Access to address specified in the above table with 32-bit width. For example 388h for EP1.
   - In the case of 16 bits: Access to address specified in the above table with 16-bit width. By making the second access to the first access address +2h, the operation becomes the same as in the case of 32 bits. For example, in the case of EP1, by making the first access to 388h and second to 38Ah the operation becomes the same as accessing once 388h in the case of 32-bit address.

Types and Mapping of Device Controller Registers - 2

Category	Offset	R/W	Offset Address Low-Order 2 Bits			
			11	10	01	00
EP support	330h	R			EP1 payload register (EP1PLD) EP1 configuration register (EP1CONF)	
	334h	R/W			EP1 payload configuration register (EP1PLDCONF)	
	338h	R/W			EP2 payload configuration register (EP2PLDCONF)	
	33Ch	R/W			EP3 payload configuration register (EP3PLDCONF)	
	340h	R/W	EP4 payload configuration register (EP4PLDCONF)			
	344h	R/W	EP5 payload configuration register (EP5PLDCONF)			
	348h	R/W			EP0 receive byte count status register (EP0RXCNTSTAT)	
	34Ch	R/W			EP1 receive byte count status register (EP1RXCNTSTAT)	
	350h	R/W			EP2 receive byte count status register (EP2RXCNTSTAT)	
	354h	R/W			EP3 receive byte count status register (EP3RXCNTSTAT)	
	358h	R/W	EP4 receive byte count status register (EP4RXCNSTAT)			
	35Ch	R/W	EP5 receive byte count status register (EP5RXCNSTAT)			
	360h	R/W			EP0 transmit data byte count register (EP0TXCNT)	
	364h	R/W			EP1 transmit data byte count register (EP1TXCNT)	
	368h	R/W			EP2 transmit data byte count register (EP2TXCNT)	
	36Ch	R/W			EP3 transmit data byte count register (EP3TXCNT)	
	370h	R/W			EP4 transmit data byte count register (EP4TXCNT)	
	374h	R/W			EP5 transmit data byte count register (EP5TXCNT)	
	378h	R/W			ISO mode select register (ISO MODE)	

Note: Access to unspecified address sections is meaningless.

Types and Mapping of Device Controller Registers - 3

Category	Offset	R/W	Offset Address 2 Lower Bits			
			11	10	01	00
EP support	39Ch	R/W			EP0 control register (EP0CONT)	
	3A0h	R/W			EP1 control register (EP1CONT)	
	3A4h	R/W			EP2 control register (EP2CONT)	
	3A8h	R/W			EP3 control register (EP3CONT)	
	3ACh	R/W			EP4 control register (EP4CONT)	
	3B0h	R/W			EP5 control register (EP5CONT)	

Note: Access to unspecified address sections is meaningless.

- EP0 Transmit FIFO Register (380h)

Bits	Field	Reset	USB Reset	R/W
31:00	EP0 Transmit data (EP0TXFIFO)	X	X	W

[Description]
The EP0 send (transmit) data can be written by writing into this register. The send data to be transmitted to the host is stored in EP0TXFIFO in the data stage during a control read transfer. When the ML60841 requests an EP0 transmit packet ready interrupt, the microcomputer writes the send data in the EP0 transmit FIFO register. It is possible to write packets of data successively by executing a series of write operations.
The EP0 transmit FIFO gets cleared under the following conditions.
- When an ACK is received from the host in response to data transmission from EP0.
- When a setup packet is received.

- EP0 Receive FIFO Register (384h)

Bits	Field	Reset	USB Reset	R/W
31:00	EP0 Receive data (EP0RXFIFO)	X	X	R

[Description]
The receive data from the host is stored in EP0RXFIFO in the data stage during a control write transfer. When the ML60841 requests an EP0 receive packet ready interrupt, the microcomputer reads the receive data by reading the EP0 receive FIFO register. The data within a packet can be read out successively by executing a series of read operations.
The EP0 receive FIFO gets cleared under the following conditions.
- When the microcomputer resets the EP0 receive packet ready status.
- When a setup packet is received.
- When the microcomputer writes a “0” in the stall bit.

- EP1 Transmit/Receive FIFO Register (388h)

Bits	Field	Reset	USB Reset	R/W
31:00	EP1 Transmit/receive data (EP1FIFO)	X	X	R/W

[Description]
The EP1 transfer direction is specified by the setting made in the EP1 configuration register EP1CONF. The FIFO address of EP1 is the same in the transmit direction as in the receive direction.
When EP1CONF (bit 07) = 0, EP1 has the receive direction and EP1FIFO becomes a read only register.
When EP1CONF (bit 07) = 1, EP1 has the transmit direction and EP1FIFO becomes a write only register.

- EP2 Transmit/Receive FIFO Register (38Ch)

Bits	Field	Reset	USB Reset	R/W
31:00	EP2 Transmit/receive data (EP2FIFO)	X	X	R/W

[Description]
The EP2 transfer direction is specified by the setting made in the EP2 configuration register EP2CONF. The FIFO address of EP2 is the same in the transmit direction as in the receive direction.
When EP2CONF (bit 07) = 0, EP2 has the receive direction and EP2FIFO becomes a read only register.
When EP2CONF (bit 07) = 1, EP2 has the transmit direction and EP2FIFO becomes a write only register.

- EP3 Transmit/Receive FIFO Register (390h)

Bits	Field	Reset	USB Reset	R/W
31:00	EP3 Transmit/receive data (EP3FIFO)	X	X	R/W

[Description]
The EP3 transfer direction is specified by the setting made in the EP3 configuration register EP3CONF. The FIFO address of EP3 is the same in the transmit direction as in the receive direction.
When EP3CONF (bit 07) = 0, EP3 has the receive direction and EP3FIFO becomes a read only register.
When EP3CONF (bit 07) = 1, EP3 has the transmit direction and EP3FIFO becomes a write only register.

- EP4 Transmit/Receive FIFO Register (394h)

Bits	Field	Reset	USB Reset	R/W
31:00	EP4 Transmit/receive data (EP4FIFO)	X	X	R/W

[Description]
The EP4 transfer direction is specified by the setting made in the EP4 configuration register EP4CONF. The FIFO address of EP4 is the same in the transmit direction as in the receive direction.
When EP4CONF (bit 07) = 0, EP4 has the receive direction and EP4FIFO becomes a read only register.
When EP4CONF (bit 07) = 1, EP4 has the transmit direction and EP4FIFO becomes a write only register.

- EP5 Transmit/Receive FIFO Register (398h)

Bits	Field	Reset	USB Reset	R/W
31:00	EP5 Transmit/receive data (EP5FIFO)	X	X	R/W

[Description]
In the ML60841, depending on the setting in the system control register (SYSCON), it is possible to select the 5EP mode, with the number of EP's being five or the 6EP mode in which the number of EP's is six. In the 5EP mode, EP0 to EP4 will be present but EP5 will not be present. All of EP0 to EP5 will be present in the 6EP mode.
The EP5 transfer direction is specified by the setting made in the EP5 configuration register EP5CONF. The FIFO address of EP5 is the same in the transmit direction as in the receive direction.
When EP5CONF (bit 07) = 0, EP5 has the receive direction and EP5FIFO becomes a read only register.
When EP5CONF (bit 07) = 1, EP5 has the transmit direction and EP5FIFO becomes a write only register.

- wValue•bRequest•BmRequest Type Setup Register (300h)

Bits	Field	Reset	USB Reset	R/W
31:24	wValueMSB setup data During the setup stage of a control transfer, the 8 bytes of setup data sent from the host are received automatically by the ML60841 and the fourth byte of that setup data is stored in this register. This is the higher order byte of a two-byte data.	0	0	R
23:16	wValueLSB setup data During the setup stage of a control transfer, the 8 bytes of setup data sent from the host are received automatically by the ML60841 and the third byte of that setup data is stored in this register. This is the lower order byte of a two-byte data.	0	0	R
15:08	bRequest setup data During the setup stage of a control transfer, the 8 bytes of setup data sent from the host are received automatically by the ML60841 and the second byte of that setup data is stored in this register. The content of the request code has been stipulated in section 9.3 of the USB standard as well as in other related standards.	0	0	R
07:00	bmRequest Type setup data During the setup stage of a control transfer based on a request from the host, the 8 bytes of setup data sent from the host are received automatically by the ML60841 and stored in eight registers including this register. The formats of these data bytes have been defined in Section 9.3 of the USB standard. 07: Data transfer direction (1: Device → Host, 0: Host → Device) 06, 05: Type (00: Standard, 01: Class, 10: Vendor, 11: Reserved) 04:00: Receive side definition (00: Device, 01: Interface, 02: End point, 03: Others, 04-31: Reserved)	0	0	R

- wLength•wIndex Setup Register (304h)

Bits	Field	Reset	USB Reset	R/W
31:24	wLengthMSB data During the setup stage of a control transfer, the 8 bytes of setup data sent from the host are received automatically by the ML60841 and the eighth byte of that setup data is stored in this register. This is the higher order byte of a two-byte data.	0	0	R
23:16	wLengthLSB data During the setup stage of a control transfer, the 8 bytes of setup data sent from the host are received automatically by the ML60841 and the seventh byte of that setup data is stored in this register. This is the lower order byte of a two-byte data.	0	0	R
15:08	wIndexMSB data During the setup stage of a control transfer, the 8 bytes of setup data sent from the host are received automatically by the ML60841 and the sixth byte of that setup data is stored in this register. This is the higher order byte of a two-byte data.	0	0	R
07:00	wIndexLSB data During the setup stage of a control transfer, the 8 bytes of setup data sent from the host are received automatically by the ML60841 and the fifth byte of that setup data is stored in this register. This is the lower order byte of a two-byte data.	0	0	R

- Device Address Register (308h)

Bits	Field	Reset	USB Reset	R/W
31:07	Don't care	X	X	X
06:00	The microcomputer writes in this register the device address specified by the host in the SET_ADDRESS request. Thereafter, the specified address of the token from the host is judged by the ML60841, and operates so that only the token packets transmitted to this device address are processed by the device. The bit D7 is fixed at "0" and even if a "1" is written to it, it will have no validity.	0	0	R/W

- Frame Number Register (30Ch)

Bits	Field	Reset	USB Reset	R/W
31:11	Don't care	X	X	X
10:00	The ML60841 writes this automatically when the start of frame (SOF) packet is sent from the host.	0	0	R

- Interrupt Status Register (310h)

Bits	Field	Reset	USB Reset	R/W
31:16	Don't care	X	X	X
15	EP0 Transmit packet ready interrupt status (ep0tx_pry)	0b	0b	R
14	EP0 Receive packet ready interrupt status (ep0rx_pry)	0b	0b	R
13	EP5 Packet ready interrupt status (ep5_pry)	0b	0b	R
12	EP4 Packet ready interrupt status (ep4_pry)	0b	0b	R
11	EP3 Packet ready interrupt status (ep3_pry)	0b	0b	R
10	EP2 Packet ready interrupt status (ep2_pry)	0b	0b	R
09	EP1 Packet ready interrupt status (ep1_pry)	0b	0b	R
08	Setup ready interrupt status (stup_ry)	0b	0b	R
07:05	Don't care	X	X	X
04	Device awake state interrupt status (awake)	0b	0b	R/Reset
03	Device suspend state interrupt status (suspend)	0b	0b	R/Reset
02	USB Bus reset deassert interrupt status (busrst_des)	0b	0b	R/Reset
01	USB Bus reset assert interrupt status (busrst_ass)	0b	0b	R/Reset
00	SOF Interrupt status (sof)	0b	0b	R/Reset

Note 1: When the EP4 and EP5 configuration registers have been set for isochronous transfer, the EP4 and EP5 packet ready interrupt statuses will always be kept fixed at "0".

Note 2: The status bit becomes "1" when an interrupt corresponding to the bits 04 to 00 occurs. This status will be cleared when a "1" is written to that status bit itself.

- Interrupt Enable Register (314h)

Bits	Field	Reset	USB Reset	R/W
31:16	Don't care	X	X	X
15	EP0 Transmit packet ready interrupt enable	0b	State before reset is maintained	R/W
14	EP0 Receive packet ready interrupt enable	0b		R/W
13	EP5 Packet ready interrupt enable	0b		R/W
12	EP4 Packet ready interrupt enable	0b		R/W
11	EP3 Packet ready interrupt enable	0b		R/W
10	EP2 Packet ready interrupt enable	0b		R/W
09	EP1 Packet ready interrupt enable	0b		R/W
08	Setup ready interrupt enable	1b		R/W
07:05	Don't care	000b		X
04	Device awake state interrupt enable	0b		R/W
03	Device suspend status interrupt enable	0b		R/W
02	USB Bus reset deassert interrupt enable	0b		R/W
01	USB Bus reset assert interrupt enable	0b		R/W
00	SOF Interrupt enable	0b		R/W

[Description]
When the 5EP mode has been selected (bit 02 is "1" in the system control register), the EP5 packet ready interrupt will not be generated irrespective of the value of bit 13 (EP5 packet ready interrupt enable).
Similarly, in this case, even bit 13 of the interrupt status register (EP5 Packet ready interrupt status) will always be "0".

- System Control Register (318h)

Bits	Field	Reset	USB Reset	R/W
31:08	Don't care	X	X	X
07	Instructs a clock stop of the device controller section 0: clock not stopped, 1: clock stopped.	0b	State before reset is maintained	R/W
06:05	Don't care	X	X	X
04	A remote wake-up is executed by writing a "1" to this bit. This bit itself will remain "0".	0b	State before reset is maintained	W
03	Specifies the status of the $\overline{\text{PUCTL}}$ pin. 0: $\overline{\text{PUCTL}}$ pin invalid (the pin is at "H" level) 1: $\overline{\text{PUCTL}}$ pin valid (the pin is at "L" level)	0b		R/W
02	Specifies the EP mode 0: 6EP mode, 1: 5EP mode	0b		R/W
01	Specifies the power down mode 0: Power not saved by suspending, 1: Power saved by suspending	0b		R/W
00	A software reset of the device controller section can be executed by writing a "1" to this bit. This bit itself will remain "0".	0b		W

[Description]
When a "1" is written to bit 07, the clock of the device control section stops after the write access cycle has been completed. In addition, the USB buffer goes into the power down mode. Therefore, it may not be possible to achieve matching with the host side since it will not be possible to recognize any changes in the USB bus status (such as a token from the USB host, etc.).
Make the clock stop specification when it is permissible to stop all the operations of a device controller such as when the host has not been connected, and in concrete terms, only when an operator intervenes and explicitly gives instruction to stop the clock.
When transition to the suspend mode is specified while the ML60841 is operating (during DMA transfer or during USB communication), the operations thereafter cannot be guaranteed.
It is necessary to input the "$\overline{\text{RESET}}$" signal in order to recover from the clock stop state initiated due to writing a "1" in this bit. The operation cannot be guaranteed if the ML60841 is accessed during the period immediately after writing a "1" in this bit until the "$\overline{\text{RESET}}$" signal is input. See the section "Precautions in Control of ML60841" for details of the processing after the "$\overline{\text{RESET}}$" signal has been input.

The difference between bit 07 and bit 01 is the following.
As has been explained above, bit 07 instructs the stop of the clock supply to the device controller block. When this bit is set to "1", the ML60841 immediately starts the clock stop processing.
On the other hand, bit 01 specifies whether or not to go into the power saving mode when the suspend state of the USB bus is detected.
When this bit has been set to "1", the processing for transition to the power saving mode (*) will be started when the suspend state of the USB bus is detected. Even when this bit has been set to "1", the processing for transition to the power saving mode will not be started as long as the suspend state of the USB bus has not been detected. When this bit has been reset to "0", the processing for transition to the power saving mode will not be made even when the suspend state of the USB bus has been detected.

*: The power save mode is equivalent to the clock stopped state that is initiated by writing a "1" to bit 07 excepting that some of the circuits in the device controller section for detecting a resume command from the USB host and some of the circuits in the USB transceiver will be operating. Therefore, the power consumption will be reduced more when a "1" is set in bit 07.

- DMA0 Control Register (320h), DMA1 Control Register (324h)

Bits	Field	Reset	USB Reset	R/W
31:08	Don't care	X	X	X
07	Don't care	X	X	X
06:05	Specifies the EP that is a target of DMA transfer. *1, *2 0: EP1, 1: EP2, 2: EP4, 3: EP5	00b	State before reset is maintained	R/W
04:03	Don't care	X	X	X
02	When the EP that is specified as the target of DMA transfer is configured as bulk transfer or interrupt transfer, insertion or no insertion of DMA byte count is specified. *1, *3 0: No insertion of byte count 1: Insertion of byte count at the top of transfer data	0b	State before reset is maintained	R/W
01	Don't care	X	X	X
00	DMA enable *1 0: DMA disable, 1: DMA enable	0b	State before reset is maintained	R/W

*1) Complete setting of bits 06, 05, 02, 00 before arrival of token packet to EP1 to EP5 and do not change after that. Operation is not guaranteed if the setting is changed.

*2) If the EP setting values for DMA channels 0 and 1 are the same, DREQ0, DRACK0 and DACK0 become equivalent to DREQ1, DRACK1 and DACK1 respectively.

*3): Although DMA transfers can be made in the bulk, interrupt, and isochronous transfer modes, the DMA byte count insertion function using this bit is only valid in the bulk and interrupt transfer modes.
The operations cannot be guaranteed when this bit is set to "1" for isochronous transfers.

- EP0 Payload & Configuration Register (330h)

Bits	Field	Reset	USB Reset	R/W
31:16	Don't care	X	X	X
15:07	EP0 Payload data (EP0PLD) Write "20h" to the bMaxPacketSize byte of the device descriptor since the FIFO of the EP0 in the ML60841 is of 32 bytes. The maximum packet size of the EP0PLD is fixed to 32 bytes. If a packet exceeding 32 bytes is received, the stall bit of the EP0 status register is asserted and the stall is returned to the Host.	20h	20h	R
07:05	Don't care	X	X	X
04	Configuration bit: The configuration bit of the EP0 becomes "1" at USB bus reset. When this bit is "1", the data transmitted from the Host to this endpoint is received and it becomes possible to transmit data from this endpoint to the Host. When this bit is "0", no response is made to the transaction with this EP as target. The microcontroller cannot write to this bit.	0	1	R
03:02	Don't care	X	X	X
01:00	These bits indicate D transfer type, and fixed to "00b" since in the ML60841 the EP0 is determined as control transfer. The microcontroller cannot write in these bits.	0	0	R

- EP1/2/3 Payload & Configuration Registers (334h/338h/33Ch)

Bits	Field	Reset	USB Reset	R/W
31:15	Don't care	X	X	X
14:07	Maximum packet size: Make the microcomputer write in this register the value of wMaxPacketSize in the end point descriptor selected by the Set_Configuration request from the host. This specifies the packet size in bytes for all packets other than short packets. When the EP is being used for reception, if a data packet with a number of bytes exceeding the maximum packet size set in this register is received, the receive packet ready status will not be asserted, but a stall handshake is returned to the host by setting the stall bit in the EOP. When the EP is being used for transmission, the transmit packet ready status bit will be set automatically at the completion of writing data by the DMA controller equal to the maximum packet size set in this register. The content of this register is "Don't care" during transmissions that do not use DMA transfer.	0	0	R/W
07	Transfer direction (0: Receive, 1: Transmit)	0	0	R/W
06:05	Don't care	X	X	X
04	Configuration bit: Write a "1" in this bit from the microcomputer during the status stage of control transfer when a Set Configuration request is received from the host requesting to make that EP active. Data transmission and reception can be made between the host and the EP when this bit is "1". When this bit is "0", there will be no responses to the transactions made targeting that EP.	0	1	R
03:02	Don't care	X	X	X
01:00	These are bits indicating the type of transfer, and are fixed at 00b since control transfer is the only type of transfer allowed for EP0 in the ML60841. The microcomputer cannot write to these bits.	0	0	R

- EP4 Payload & Configuration Register (340h)

Bits	Field	Reset	USB Reset	R/W
31:24	Don't care	X	X	X
23:16	EP4 Packet maximum size MSB	00h	00h	R/W
05:08	EP4 Packet maximum size LSB	00h	00h	R/W
07	EP4 Transfer direction	0b	0b	R/W
06:05	Don't care	00b	00b	X
04	EP4 Configuration	0b	0b	R/W
03:02	Don't care	00b	00b	X
01:00	EP4 Transfer type	00b	00b	R/W

[Description]
Make the microcomputer write a "1" in bit 04 (EP4 Configuration) during the status stage of control transfer when a Set Configuration request is received from the host requesting that EP4 be made active.
When this bit 04 is "1", data transmission and reception can be made between the host and EP4. When this bit is "0", there will be no responses to the transactions made targeting EP4.
However, when the 5EP mode has been selected (bit 02 = 1 in the system control register), there will be no responses to the transactions made targeting EP4 irrespective of the value set in bit 04 (EP4 Configuration). In other words, in this case, the operations will be the same as when a "0" has been set in this bit even if a "1" has been written in this bit.

- EP5 Payload & Configuration Register (344h)

Bits	Field	Reset	USB Reset	R/W
31:24	Don't care	X	X	X
23:16	EP5 Packet maximum size MSB	00h	00h	R/W
05:08	EP5 Packet maximum size LSB	00h	00h	R/W
07	EP5 Transfer direction	0b	0b	R/W
06:05	Don't care	00b	00b	X
04	EP5 Configuration	0b	0b	R/W
03:02	Don't care	00b	00b	X
01:00	EP5 Transfer type	00b	00b	R/W

[Description]
Make the microcomputer write a "1" in bit 04 (EP5 Configuration) during the status stage of control transfer when a Set Configuration request is received from the host requesting that EP5 be made active.
When this bit 04 is "1", data transmission and reception can be made between the host and EP5. When this bit is "0", there will be no responses to the transactions made targeting EP5.
However, when the 5EP mode has been selected (bit 02 = 1 in the system control register), there will be no responses to the transactions made targeting EP5 irrespective of the value set in bit 04 (EP5 Configuration). In other words, in this case, the operations will be the same as when a "0" has been set in this bit even if a "1" has been written in this bit.

- EP0 Received Byte Count & Status Register (348h)

Bits	Field	Reset	USB Reset	R/W
31:16	Don't care	X	X	X
15:08	The ML60841 automatically counts the number of bytes in the packet being received. While the counting is made up to the number of bytes equal to the maximum packet size specified in the payload register in the case of full packets, the byte count will fall short of that maximum packet size in the case of short packets. The microcomputer refers to this value and reads the data of one packet from the EP0 Receive FIFO. EP0RXCNT is cleared under the following conditions: 1. When the microcontroller resets the EP packet ready bit. 2. When a setup packet is received. 3. When the microcontroller writes a "0" in the stall bit.	00h	00h	R
07:06	Don't care	X	X	X
05:04	EP0 stage 00: Default stage 01: Data stage 10: Data stage completed state	0b	0b	R
03	Don't care	X	X	X
02	Setup ready: This bit is set automatically when a setup packet for storing 8 bytes of setup data in the setup registers arrives correctly, and the EP0 Receive FIFO gets locked. If INTENBL1 (0) has been asserted, the $\overline{INTR}$ signal is asserted automatically when this bit is set. Write a "1" in this bit when the microcomputer completes reading the 8 bytes of setup data. By doing so, the setup ready will be reset and also the $\overline{INTR}$ pin will be deasserted. In the case of a control write transfer, the EP0 packet ready bit is reset at the same time thereby releasing the lock on the EP0 Receive FIFO so that packet reception can be made by EP0 during the data stage. Note that the value of this register will not change even if a "0" is written in this bit.	0b	0b	R/Reset
01	EP0 Transmit pocket ready	0b	0b	R/Set
00	EP0 Receive packet ready	0b	0b	R/Reset

[Description]

• EP0 Transmit packet ready bit (bit 01)

This bit can be read by the microcontroller. In addition, this bit can be set to "1" by writing a "1" in this bit 01. The conditions of asserting and deasserting this bit are as follows.

Bit name	Assert condition	Operation when asserted
EP0 Transmit packet ready (bit 01)	When the microcontroller sets this bit	Data transmission can be made from EP0.

Bit name	Deassert condition	Operation when deasserted
EP0 Transmit packet ready (bit 01)	1. When an ACK is received from the host in response to data transmission from EP0. 2. When a setup packet is received.	EP0 is locked. That is, an NAK is returned automatically when an IN token arrives from the host.

• EP0 Receive packet ready bit (bit 00)
This bit can be read by the microcontroller. In addition, this bit can be reset to "0" by writing a "1" to this bit 00. The assert and deassert conditions of this bit are as follows.

Bit name	Assert condition	Operation when asserted
EP0 Receive packet ready (bit 01)	1. When data is received in EP0 and is stored in the FIFO. 2. When a setup packet is received during a control read or a control write transfer.	EP0 is locked. That is, an NAK is returned automatically when a data packet arrives from the host.

Bit name	Deassert condition	Operation when deasserted
EP0 Receive packet ready (bit 01)	1. When the microcontroller resets this bit (by writing a "1" in it). 2. When the microcontroller rests the setup ready bit during a control write transfer.	Can be received by EP0.

- EP1, 2 Receive Byte Count Registers / EP1, 2 Status Registers (34Ch, 350h)

Bits	Field	Reset	USB Reset	R/W
31:16	Don't care	X	X	X
15:08	The number of bytes in the packets received at EP1 and EP2 is counted and the value is indicated in these bits. Although the counting is made only up to the number of bytes equal to the maximum packet size specified in the payload register in the case of full packets, the count will be short of that maximum packet size in the case of short packets. The microcontroller refers to this value and reads one packet of data from the EP1/2 Receive FIFOs. This register will not be valid when the transfer direction of the EPs is set to transmission. EP1, 2 RXCNT are cleared under the following conditions. 1. When an OUT token for the EP is received. 2. When the microcontroller resets the EP receive packet ready bit. 3. When the microcontroller writes a "0" in the stall bit.	00h	00h	R
07:02	Reserved. (For write, write a "0".)	00h	00h	R/W
01	This bit is valid only when EP1 and EP2 are set to bulk or INT transfer. It indicates whether valid data is stored in the internal FIFO of this LSI or not. For details, refer to the description below.	0b	0b	R/SET
00	This bit is valid only when EP1 and EP2 are set to bulk or INT transfer. It indicates whether valid data is stored in the internal FIFO of this LSI or not. For details, refer to the description 2 below.	0b	0b	R/RST

Note: Write a "0" in bit 00 when writing in bit 01, and write a "0" in bit 01 when writing in bit 00. Operation is not guaranteed if a value other than "0" is written.

[Description 1]
Bit 01 (transmit packet ready) can be read and set. By writing a "1", this bit can be set to "1".
The assert/deassert conditions of this bit are shown below. EP1 and EP2 have FIFO of two layers (layers A and B). The transmit packet ready bit also individually exists in layer A and layer B and this LSI automatically executes switching of these two layers.

	Condition	Operation
Assert	When the microcontroller sets bits of both layer A and layer B.	Transmitting from EP1 and EP2 are possible when either layer A or layer B is asserted. (Data is transmitted with regard to IN token.)
Deassert	When ACK is received from the host with regard to the transmit data of either layer A or layer B.	Lock EP1 and EP2 when transmit data is not ready in either layer A or layer B. (NACK reply for IN token.)

Detailed example is shown below.

		Layer A FIFO (64B)	Layer B FIFO (64B)	Layer A FIFO Packet Ready	Layer B FIFO Packet Ready	EP1, 2 Transmit Packet Ready Bit
1	Layers A and B FIFO both empty			×	×	×
2	Microcontroller writes data in layer A FIFO.	■		×	×	×
3	Data write for 1 packet is completed.	■		O	×	×
4	Transmit layer A FIFO data to USB bus and write next packet data to layer B FIFO.	■	■	O	×	×
5a	When data write to layer B FIFO is completed before layer A FIFO becomes empty.	■	■	O	O	O
5b	When layer A FIFO becomes empty before data write to layer B FIFO is completed.		■	×	×	×
6	From 5a : layer A FIFO became empty. From 5b : layer B FIFO became full.		■	×	O	×
7	Start transmitting layer B FIFO data to USB bus.		■	×	O	×

O: Assert ×: Deassert

[Description 2]
Bit 00 (Receive packet ready bit) can be read by the microcontroller. Further, it is possible to reset this bit to "0" by writing a "1" to it. The conditions of asserting and deasserting this bit are as follows. EP1 and EP2 have two separate layers of FIFOs, and even the receive packet ready bits are provided independently for layer A and layer B. The switching between these two layers is done automatically by the ML60841.

	Condition	Operation
Assert	When an error-free packet is received in either layer A or layer B.	The microcontroller can read the EP1, 2 receive FIFOs. EP1 and EP2 are locked if packets of data have been received in both layer A and layer B.
Deassert	When the microcontroller resets (by writing a "1") the bits of both layer A and layer B.	EP1 and EP2 can receive data when one of the bits of layer A and layer B has been reset.

- EP3 Receive Byte Count & Status Register (354h)

Bits	Field	Reset	USB Reset	R/W
31:14	Don't care	X	X	X
13:08	The ML60841 automatically counts the number of packets being received. While the counting is made up to the number of bytes equal to the maximum packet size specified in the payload register in the case of full packets, the byte count will fall short of that maximum packet size in the case of short packets. The microcontroller refers to this value and reads the data of one packet from the EP3 Receive FIFO. This register will not be valid when the EP3 transfer direction is set to transmission. EP3RXCNT is cleared under the following conditions: 1. When an OUT token is received for EP3. 2. When the microcontroller resets the EP3 receive packet ready bit. 3. When the microcontroller writes a "0" in the stall bit.	00h	00h	R
07:02	Don't care	X	X	X
01	Receive packet ready	0b	0b	R/SET
00	Transmit packet ready	0b	0b	R/RST

[Description]
This register becomes valid only when the corresponding EP has been set to bulk or interrupt transfer.

• EP3 Receive packet ready bit (bit 00)
This bit can be read by the microcontroller. In addition, this bit can be set to "0" by writing a "1" to this bit 00. The conditions of asserting and deasserting this bit are as follows:

Bit name	Assert condition	Operation when asserted
EP3 Receive packet ready (bit 00)	When an error-free packet is received.	EP3 is locked.

Bit name	Deassert condition	Operation when deasserted
EP3 Receive packet ready (bit 00)	When the microcontroller resets this bit (by writing a "1" to it).	EP3 can receive packets.

• EP3 Transmit packet ready bit (bit 01)
This bit can be read by the microcontroller. In addition, this bit can be set to "1" by writing a "1" in this bit 01. The conditions of asserting and deasserting this bit are as follows.

Bit name	Assert condition	Operation when asserted
EP3 Transmit packet ready (bit 01)	When the microcontroller sets this bit	EP3 can transmit packets.

Bit name	Deassert condition	Operation when deasserted
EP3 Transmit packet ready (bit 01)	When an ACK is received from the host in response to data transmission from EP3.	EP3 is locked.

- EP4, 5 Receive Byte Count & Status Registers (358h, 35Ch)

Bits	Field	Reset	USB Reset	R/W
31:24	Don't care	X	X	X
23:08	The number of bytes in the packets received at EP4 and EP5 is counted and the value is indicated in these bits. Although the counting is made only up to the number of bytes equal to the maximum packet size specified in the payload register in the case of full packets, the count will be short of that maximum packet size in the case of short packets. The microcontroller refers to this value and reads one packet of data from the EP4/5 Receive FIFOs. The lower order 8 bits of the received byte count are stored in this register and the higher order bits are stored in the EP Receive byte counter MSB register. This register will not be valid when the transfer direction of the EPs is set to transmission. EP4, 5 RXCNT are cleared under the following conditions. 1. When an OUT token for the EP is received. 2. When the microcontroller resets the EP Receive packet ready bit. 3. When the microcontroller writes a "0" in the shall bit.	00h	00h	R
07	This bit becomes valid only when EP4, 5 have been set to ISO IN transfer. This bit indicates whether or not an IN token for this EP was sent by the host in the immediately previous frame. 0: An IN token has not been sent for this EP in the immediately previous frame. 1: An IN token has been sent for this EP in the immediately previous frame. For details, see "Description 1" below.	00h	00h	R
06:02	Reserved (Write "0" when writing to these bits)	00h	00h	R/W
01	This bit becomes valid only when EP4 and EP 5 have been set to either bulk or interrupt transfer mode. This bit indicates whether or not valid data has been stored in the internal FIFO of the ML60841. For details, see "Description 2" below.	0b	0b	R/SET
00	This bit becomes valid only when EP4 and EP 5 have been set to either bulk or interrupt transfer mode. This bit indicates whether or not valid data has been stored in the internal FIFO of the ML60841. For details, see "Description 3" below.	0b	0b	R/RST

Note: Write "0" to bit 00 when writing to bit 01 and also write "0" to bit 01 when writing to bit 00. The operation cannot be guaranteed when a value other than "0" is written.

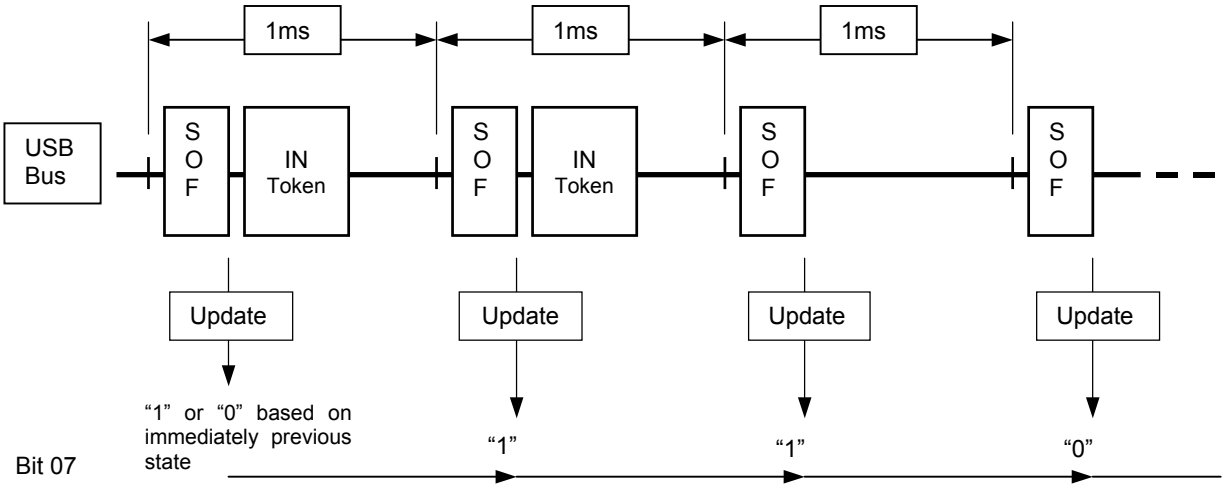
[Description 1]

The bit 07 (previous frame IN token bit) is valid only when EP4 and EP5 have been configured to the ISO IN mode. The value of this bit will be indeterminate when the configuration is other than ISO IN.

The ML60841 judges whether or not an IN token to this EP was sent by the host during the immediately previous frame at the timing of detecting SOF, and updates the value of this bit at that timing.

- This bit is updated to "1" when an IN token has been sent to this EP.
- This bit is updated to "0" when an IN token has not been sent to this EP.

Therefore, the microcontroller can know the correct status concerning the immediately previous frame by reading this bit after an SOF interrupt. See the diagram below for the detailed timings.



See "Precautions in Control of ML60841" for the method of using this bit 07.

[Description 2]
Bit 01 (transmit packet ready) can be read and set. This bit can be set to "1" by writing a "1".
The assert/deassert conditions of this bit are shown below. EP4 and EP5 have FIFO of two layers (layer A and B). The transmit packet ready bit also individually exists for layer A and layer B and this LSI automatically executes switching of these two layers.

	Condition	Operation
Assert	When microcontroller sets both layer A bit and layer B bit.	Transmitting from EP4 and EP5 is possible when asserted either from layer A or layer B. (Data is transmitted with regard to INT token.)
Deassert	When ACK is received from the host with regard to transmit data of either layer A or layer B.	Lock EP4 and EP5 when send data can not be prepared in both layer A and layer B. (NACL reply to IN token.)

Detailed example is shown below.

		Layer A FIFO (64B)	Layer B FIFO (64B)	Layer A FIFO Packet Ready	Layer B FIFO Packet Ready	EP4, 5 Transmit Packet Ready Bit
1	Layers A and B FIFO both empty.			×	×	×
2	Microcontroller writes data in layer A FIFO.	■		×	×	×
3	Data write of 1 packet is completed.	■		O	×	×
4	Transmits layer A FIFO data to USB bus and writes next packet data to layer B FIFO.	■	■	O	×	×
5a	When data write to layer B FIFO is completed before layer A FIFO becomes empty.	■	■	O	O	O
5b	When layer A FIFO becomes empty before data write to layer B FIFO is completed.		■	×	×	×
6	From 5a: Layer A FIFO became empty. From 5b: Layer B FIFO became full.		■	×	O	×
7	Starts transmitting layer B FIFO data to USB bus.		■	×	O	×

O: Assert ×: Deassert

[Description 3]
Bit 00 (Receive packet ready bit) can be read out by the microcontroller. In addition, it is possible to reset this bit to "0" by writing a "1".
The conditions of asserting and deasserting this bit are as given below. EP4 and EP5 each have two separate layers of FIFOs, and also the receive packet ready bit is provided independently for layer A and layer B. The switching between these two layers is made automatically by the ML60841.

	Condition	Operation
Assert	When an error-free packet is received in either layer A or layer B.	The microcontroller can read the EP4, 5 receive FIFOs. EP4 and EP5 are locked if packets of data have been received in both layer A and layer B.
Deassert	When the microcontroller resets (by writing a "1") the bits of both layer A and layer B.	EP4 and EP5 can receive packet data when one of the bits of layer A and layer B has been reset.

- EP0 Transmit Data Byte Count Register (360h)

Bits	Field	Reset	USB Reset	R/W
31:06	Don't care	X	X	X
05:00	EP0TFCnt[5:0]	0h	0h	R/W

It is possible to suppress the transmission of invalid data by setting the number of bytes to be transmitted per packet in this register ("01h" for one-byte transmission). When sending successive packets with the same number of bytes, it is not necessary to set the value in this register each time (since the value set immediately before is retained).
In the PIO mode, write the data in the FIFO for packet data transmission after first writing the number of bytes of transmit data in this register, and then set the transmit PKTRDY bit to "1".

- EP1 Transmit Data Byte Count Register (364h)

Bits	Field	Reset	USB Reset	R/W
31:07	Don't care	X	X	X
06:00	EP1TFCnt[6:0]	0h	0h	R/W

It is possible to suppress the transmission of invalid data by setting the number of bytes to be transmitted per packet in this register ("01h" for one-byte transmission). When sending successive packets with the same number of bytes, it is not necessary to set the value in this register each time (since the value set immediately before is retained).
In the PIO mode, write the data in the FIFO for packet data transmission after first writing the number of bytes of transmit data in this register, and then set the transmit PKTRDY bit to "1".
In the DMA mode, write the data in the FIFO after first writing the number of bytes of transmit data in this register.
See the section "Precautions in Control of ML60841" for details on the method of using this register.

- EP2 Transmit Data Byte Count Register (368h)

Bits	Field	Reset	USB Reset	R/W
31:07	Don't care	X	X	X
06:00	EP2TFCnt[6:0]	0h	0h	R/W

It is possible to suppress the transmission of invalid data by setting the number of bytes to be transmitted per packet in this register ("01h" for one-byte transmission). When sending successive packets with the same number of bytes, it is not necessary to set the value in this register each time (since the value set immediately before is retained).
In the PIO mode, write the data in the FIFO for packet data transmission after first writing the number of bytes of transmit data in this register, and then set the transmit PKTRDY bit to "1".
In the DMA mode, write the data in the FIFO after first writing the number of bytes of transmit data in this register.
See the section "Precautions in Control of ML60841" for details on the method of using this register.

- EP3 Transmit Data Byte Count Register (36Ch)

Bits	Field	Reset	USB Reset	R/W
31:06	Don't care	X	X	X
05:00	EP3TFCnt[5:0]	0h	0h	R/W

It is possible to suppress the transmission of invalid data by setting the number of bytes to be transmitted per packet in this register ("01h" for one-byte transmission). When sending successive packets with the same number of bytes, it is not necessary to set the value in this register each time (since the value set immediately before is retained).
In the PIO mode, write the data in the FIFO for packet data transmission after first writing the number of bytes of transmit data in this register, and then set the transmit PKTRDY bit to "1".

- EP4 Transmit Data Byte Count Register (370h)

Bits	Field	Reset	USB Reset	R/W
31:10	Don't care	X	X	X
09:08	EP4TFCntMSB[1:0]	0h	0h	R/W
07:00	EP4TFCnt[7:0]	0h	0h	R/W

It is possible to suppress the transmission of invalid data by setting the number of bytes to be transmitted per packet in this register ("01h" for one-byte transmission). When sending successive packets with the same number of bytes, it is not necessary to set the value in this register each time (since the value set immediately before is retained).
In the PIO mode, write the data in the FIFO for packet data transmission after first writing the number of bytes of transmit data in this register, and then set the transmit PKTRDY bit to "1".
In the DMA mode, write the data in the FIFO after first writing the number of bytes of transmit data in this register.
See the section "Precautions in Control of ML60841" for details on the method of using this register.

- EP5 Transmit Data Byte Count Register (374h)

Bits	Field	Reset	USB Reset	R/W
31:09	Don't care	X	X	X
08	EP5TFCntMSB[0]	0b	0b	R/W
07:00	EP5TFCnt[7:0]	0h	0h	R/W

It is possible to suppress the transmission of invalid data by setting the number of bytes to be transmitted per packet in this register ("01h" for one-byte transmission). When sending successive packets with the same number of bytes, it is not necessary to set the value in this register each time (since the value set immediately before is retained).
In the PIO mode, write the data in the FIFO for packet data transmission after first writing the number of bytes of transmit data in this register, and then set the transmit PKTRDY bit to "1".
In the DMA mode, write the data in the FIFO after first writing the number of bytes of transmit data in this register.
See the section "Precautions in Control of ML60841" for details on the method of using this register.

- ISO Mode Select Register (378h)

Bits	Field	Reset	USB Reset	R/W
31:06	Don't care	X	X	X
05	This bit will be valid only when EP5 has been configured for ISO IN transfer. This bit specifies timing of layer switching and clearing of the two-layer FIFO for EP5. 0: The FIFO layer switching and clearing are made every time an SOF is received. 1: The FIFO layer switching and clearing are made only when an IN token for EP5 has been sent by the host during the immediately preceding frame. The ML60841 checks this bit and carries out the corresponding operation at the timing when an SOF is received.	0b	0b	R/W
04	This bit will be valid only when EP4 has been configured for ISO IN transfer. This bit specifies the timing of layer switching and clearing of the two-layer FIFO for EP4. 0: The FIFO layer switching and clearing are made every time an SOF is received. 1: The FIFO layer switching and clearing are made only when an IN token for EP4 has been sent by the host during the immediately preceding frame. The ML60841 checks this bit and carries out the corresponding operation at the timing when an SOF is received.	0b	0b	R/W
03:00	Don't care	X	X	X

[Description]
Bits 05 and 04 are valid only when EP4 and EP5 have respectively been configured for ISO IN transfer. These bits specify the timing of switching between the two layers of the 2-layer FIFO and the timing of clearing the FIFO during ISO IN transfers.
The settings of these bits are ignored (that is, they have no effect on the operations) when EP5 and EP4 have been configured for a transfer mode other than the ISO IN transfer mode.
See the section "Precautions in Control of ML60841" for details on the method of using these bits.

- EP0 Control Register (39Ch)

Bits	Field	Reset	USB Reset	R/W
31:05	Don't care	X	X	X
04	Data sequence toggle bit (Transmit): The ML60841 carries out automatically the synchronization based on the data sequence toggle mechanism. Further, write operations to these bits (bit 04 and bit 01) are not valid.	X	X	X
03:02	Don't care	X	X	X
01	Data sequence toggle bit (Receive): The ML60841 carries out automatically the synchronization based on the data sequence toggle mechanism. Further, write operations to these bits (bit 04 and bit 01) are not valid.	0b	0b	R/W
00	Stall bit: The ML60841 automatically sets this bit to "1" during EP0 reception (data stage of a control write transfer) when a packet having a number of bytes exceeding the maximum packet size specified in EP0PLD (or a packet with missing EOP) is received. This bit is cleared to "0" automatically when a setup packet is received, in order to conform to the protocol stall of USB Rev. 1.1.	0h	0h	R/W

- EP1/2/3/4/5 Control Registers (3A0/3A4/3A8/3AC/3B0h)

Bits	Field	Reset	USB Reset	R/W
31:05	Don't care	X	X	X
03	Rate feed back: Valid only in the case of EP3. This bit will remain fixed at "0" in the case of all other EPs.	0b	0b	R/W
02	FIFO Clear: Set this bit at the same time as setting the data sequence toggle bit. Do not set this bit at any other time. This bit is valid only when the EP has been configured for transmission in the EP control register. When a "1" is written to this bit, the transmit FIFO of the corresponding EP will be cleared. (This bit itself will remain "0".)	0b	0b	R/Reset
01	Data sequence toggle bit: A reset is made when a "1" is written in this bit. When initializing the EP, a "1" is written in this bit thereby resetting the data packet toggle bit, and the PID of DATA0 is specified. (This bit too becomes "0".) The synchronization operation thereafter based on the data sequence toggle mechanism will be made automatically.	0b	0b	R/Reset
00	Stall bit: The ML60841 automatically sets this bit to "1" during EP0 reception (data stage of a control write transfer) when a packet having a number of bytes exceeding the maximum packet size specified in EP0PLD (or a packet with missing EOP) is received.	0b	0b	R/W

NOTES ON THE ML60841 CONTROL

Host/Device Setting and Select Control

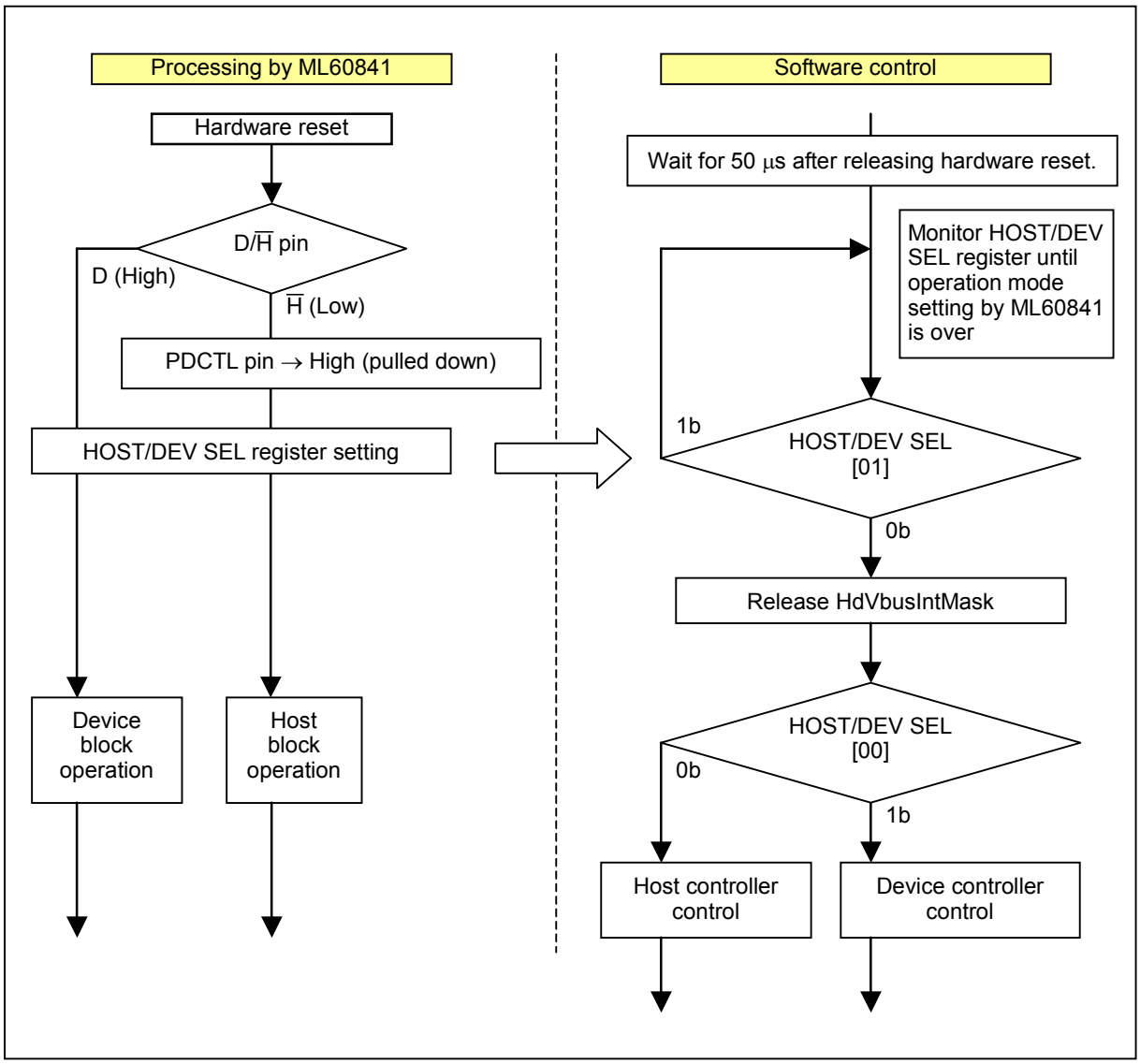
This ML60841 selects Host or Device operation mode according to the status of the D/$\overline{H}$ pin after reset. Therefore, it is necessary for the microcontroller to wait until the setting is over (*). After setting of the operation mode, only write to a register corresponding to the operation mode is guaranteed.

(*) By setting the HOST/DEV SEL register, operation mode can also be set actively from the microcontroller side.

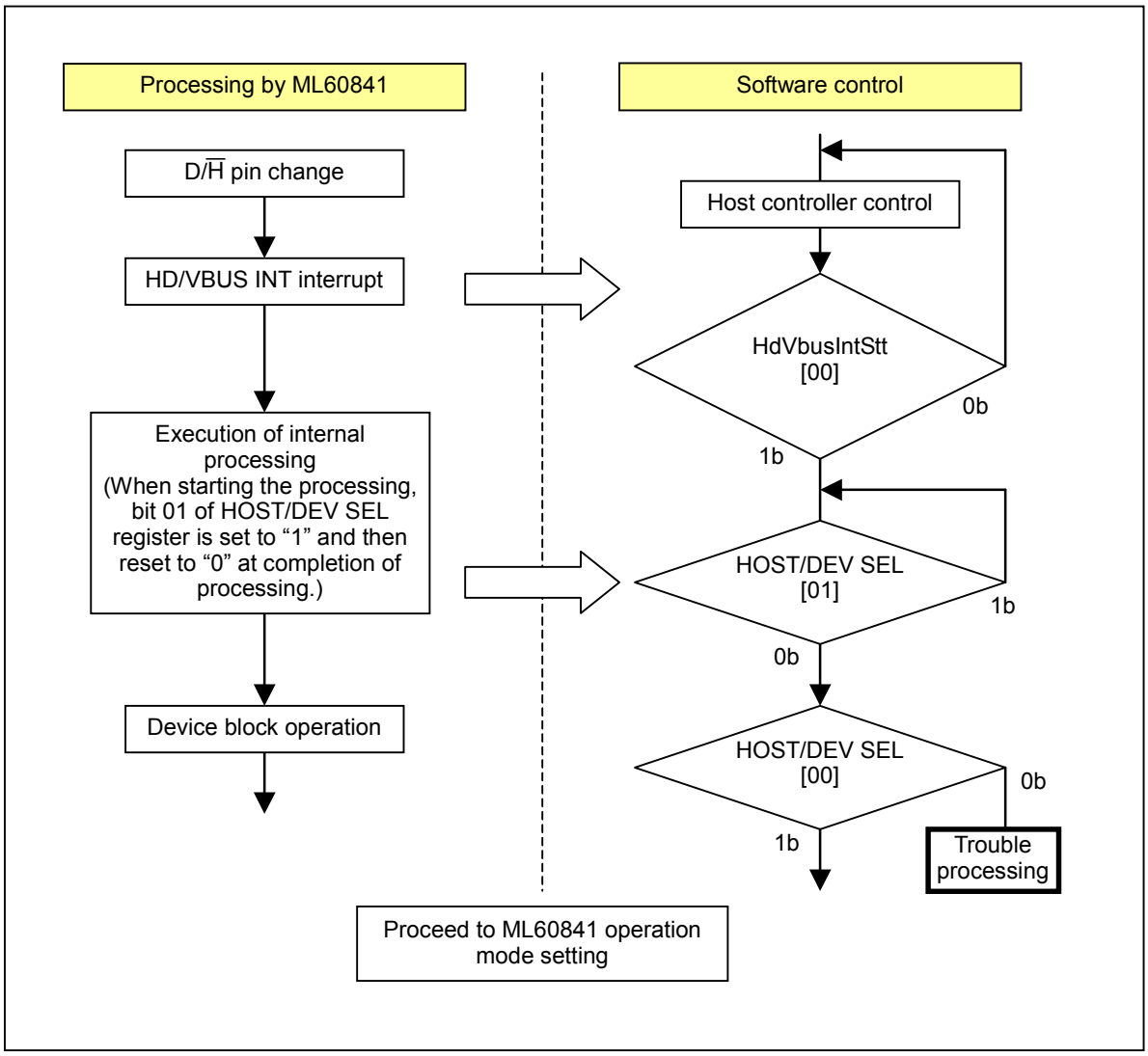
Supply current is reduced by stopping clock to the device control section in the case of Host operation mode setting and clock to the host control section in the case of Device mode setting.

The Host/Device operation mode settings and the software control method upon mode switching are described below.

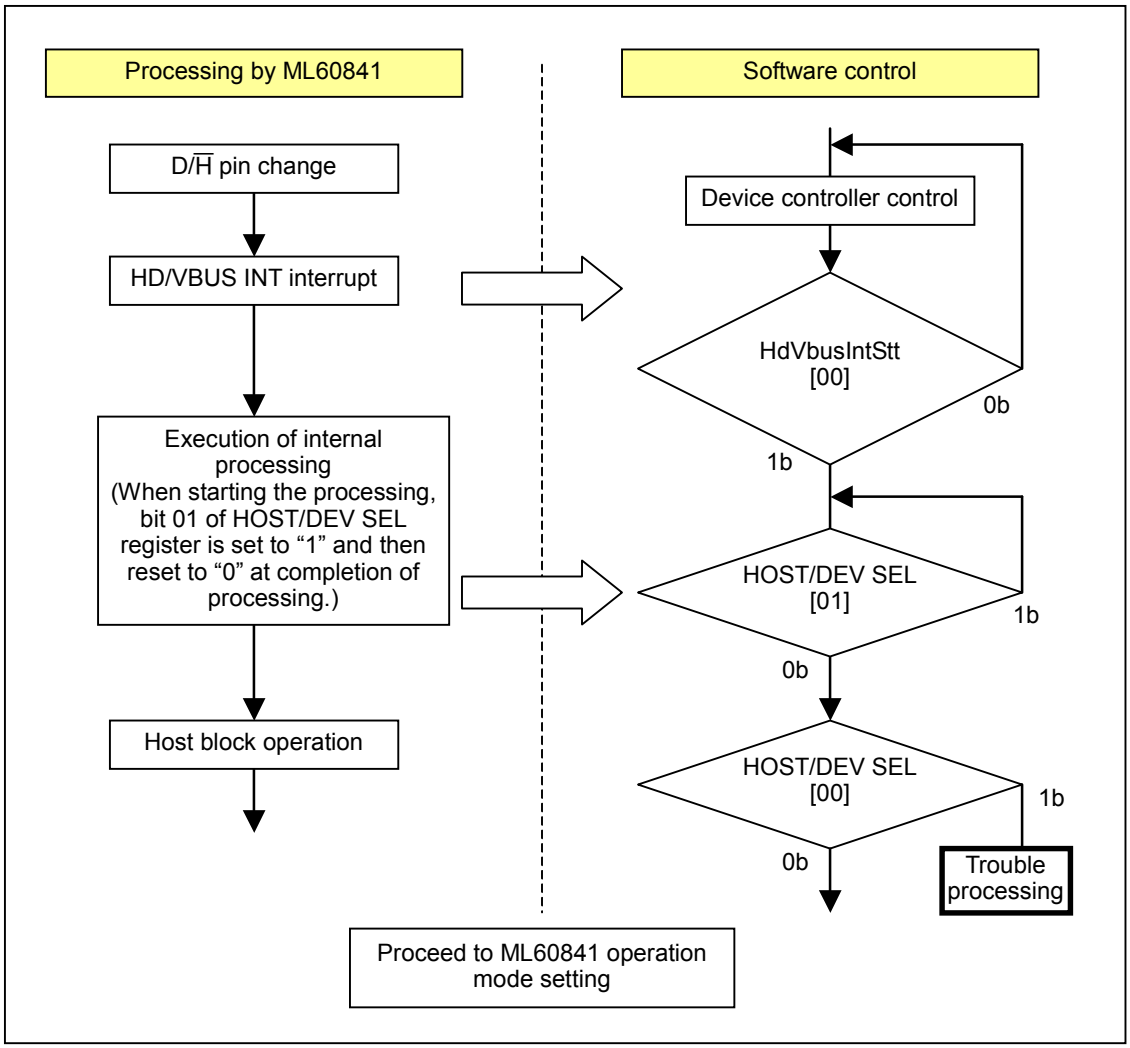
(1) Operation Mode Setting



(2) Shifting to Device operation mode in Host operation



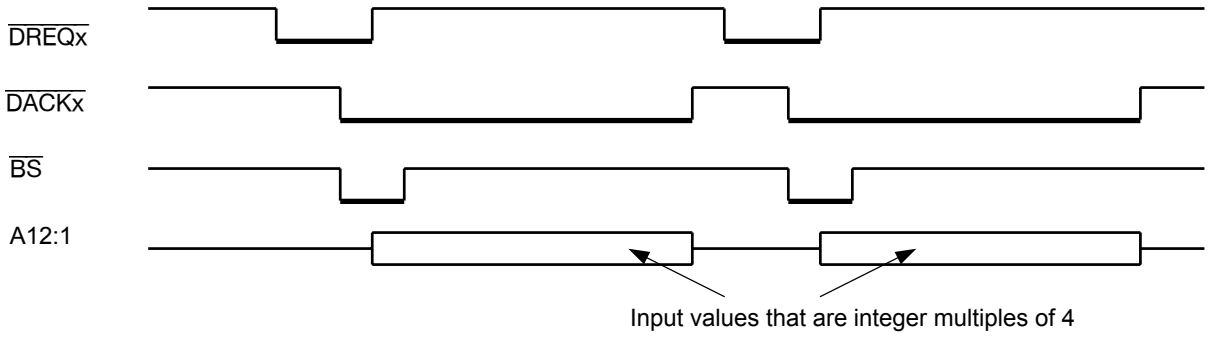
(3) Shifting to Host operation mode in Device operation



Notes on DMA Data Transfer Address

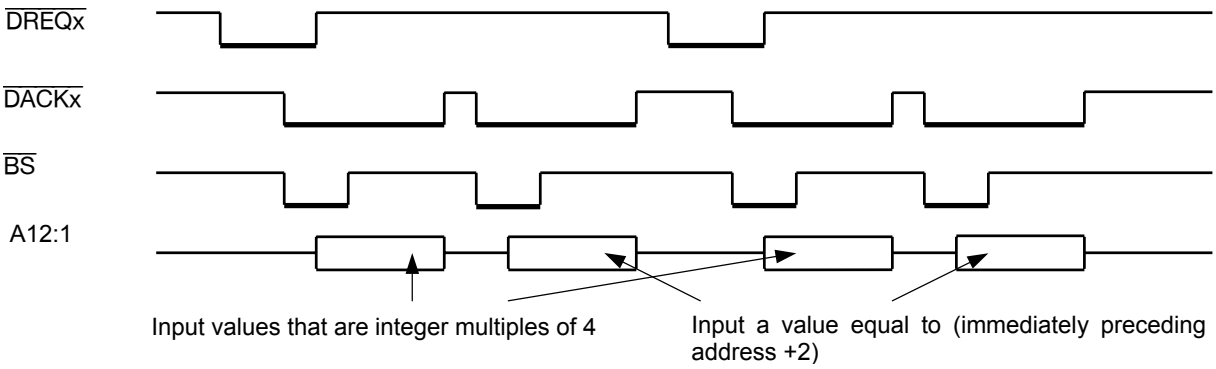
The data bus width in the ML60841 can be set to 16 bits or 32 bits. In addition, it is possible to specify 16 bits or 32 bits as the transfer size per one DREQ in the DMA function.
The method of inputting the address is given below for the different combinations of the data bus width and the transfer size per one DREQ.

(1) Bus width = 32 bits & Transfer size per DREQ = 32 bits



For example, when carrying out data transfer with the FIFO ACC register (20Ch) of the host controller section, it is necessary to repeat the address specification in the sequence 20Ch → 20Ch →…. → 20Ch → 20Ch (the starting address is 20Ch and the ending address is also 20Ch).
Further, when carrying out data transfer with the EP1 transmit/receive FIFO register (388h) of the device controller section, it is necessary to repeat the address specification in the sequence 388h → 388h →…. → 388h → 388h (the starting address is 388h and ending address is also 388h).

(2) Bus width = 16 bits & Transfer size per DREQ =32 bits



For example, when carrying out data transfer with the FIFO ACC register (20Ch) of the host controller section, it is necessary to repeat the address specification in the sequence 20Ch → 20Eh → 20Ch → 20Eh →…. → 20Eh → 20Ch → 20Eh (the starting address is 20Ch and the ending address is 20Eh).
Further, when carrying out data transfer with the EP1 transmit/receive FIFO register (388h) of the device controller section, it is necessary to repeat the address specification in the sequence 388h → 38Ah → 388h → 38Ah →…. → 38Ah → 388h → 38Ah (the starting address is 388h and the ending address is 38Ah).

Notes on the Host Control

As to its host function, the ML60841 complies with OHCI, but possesses special settings and functions apart from OHCI. The special part is described below.

(1) DMA mode 0 register, DMA mode 1 register
These are the setting registers to cope with DMA transfer of microcontroller. They must be set as described below.
As host, only the channel 0 of DMA can be used. Therefore, set the following contents in the DMA mode 0 register. Operation is not guaranteed if DMA channel 1 is used.

- Bits 03, 02
  Specifies data bit length for one time of data transfer (DREQ). Since in this LSI the length is fixed at 32 bits, always set "10". Operation is not guaranteed if a value other than "10" is set.
  These bits correspond to bits TS1 and TS0 of the DMA channel control register of the microcontroller and it is necessary to set the same bit length as the microcontroller.

(2) Host control register
Set DMA/PIO transfer (bit 03) and interrupt factor mask (bits 02, 01) from the host section.
By setting DMA/PIO transfer (bit 03) = 1 (PIO transfer), interrupt is generated if the transfer request address from host core is outside the range of internal RAM address. At this time it is necessary to process the data of double word count indicated in bits 18:16 (write) or bits 26:24 (read) of the STT/TRANS CNT register.

(3) Locations of Descriptor, Data Buffer, etc.
It is necessary to place the end point descriptor (ED), transfer descriptor (TD) and interrupt table (HCCA) required for transfer by OHCI in the internal RAM (4 KB) of the ML60841. Moreover it is also necessary to locate ED and TD at 16-byte boundary (at 32-byte boundary for TD of isochronous transfer) and HCCA at 256-byte boundary.
The buffer for data to be transferred can be located either in the internal RAM of the ML60841 or in external memory area, but the top address should be set on 4-byte boundary.
The ML60841 always compares the address set in the RAM ADR register and transfer request address. If they are the same, they are treated as transfer to the internal RAM of the ML60841; therefore, interrupt for host data transfer is not requested to microcontroller.
Therefore by locating the data buffer in the internal RAM of the ML60841, interrupt requests etc. to microcontroller are minimized and thus burden on the CPU can be considerably reduced.
However, it does not limit the use of the ML60841. It is necessary to suitably locate considering system configuration restraints and burden of microcontroller, etc.

(4) Interrupt Processing
Two types of interrupts, namely interrupt relating to host data transfer and interrupt from host core exist in the host section of the ML60841.
As regards the interrupt relating to host data transfer, the generation conditions are as follows according to the value (DMA/PIO) of bit 03 of the HOST CTL register.

(a) In the case of bit 03 of the HOST CTL register = "0" (DMA)
- When the transfer address is outside the internal RAM address range of the ML60841.
- When either of the following conditions occurs within the internal RAM address range of the ML60841:
  - When continuity (*) of transfer addresses is lost.
    The ML60841 monitors continuity of addresses and generates interrupt when continuity is lost. Therefore, interrupt is always generated at the time of first transfer.
  - When direction of transfer has changed.
    When direction of transfer has changed with regard to the transfer executed just before, that is, when the value of bit 00 of the host data transfer request register has changed.

In such case it is necessary to again set the transfer length (basically the maximum value) and transfer start address with regard to the DMA controller in interrupt processing. Moreover, it is necessary to clear this interrupt factor after completing the setting with regard to the DMA controller.

(b) In the case of bit 03 of the HOST CTL register = "1" (PIO)
When the transfer address is outside the internal RAM address range of the ML60841, interrupt is generated every time a transfer request is generated. However, interrupt is not generated with regard to transfer of double word shown in bits 26:24 (read)/bits 18:16 (write) of the ST/TRANS CTN register at the time of interrupt generation. (Execution of transfer of double word count shown in the said bit in one time of interrupt is possible.)
In interrupt processing it is necessary to write/read data to and from the FIFO ACC register by PIO after clearing of the interrupt factor.

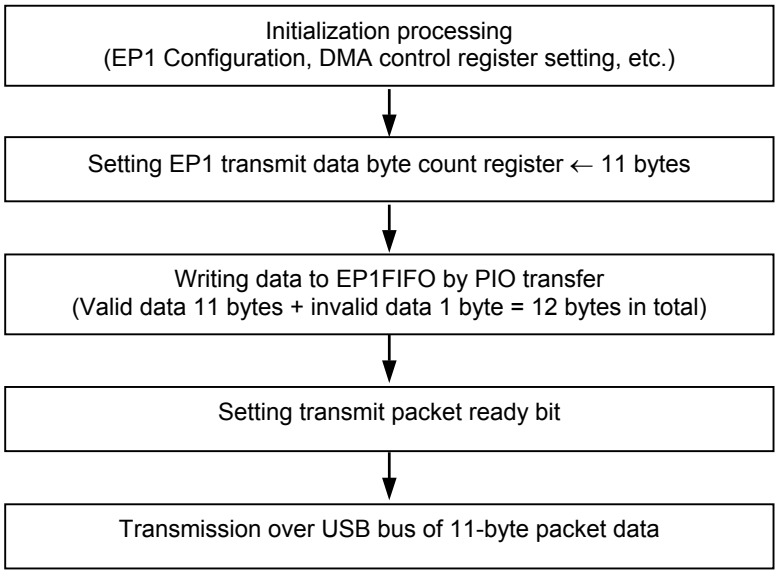
(*) Regarding continuity of addresses
Continuity of addresses means existence of "+4h" address with regard to the just preceding transfer address.
For example when doing 32-bit access by dividing the data of 20h byte up to addresses 00h to 1Fh into 12h byte and Eh byte, since address changes as 00h $\rightarrow$ 04h $\rightarrow$ 08h $\rightarrow$ 0Ch $\rightarrow$ 10h, 10h $\rightarrow$ 14h $\rightarrow$ 18h $\rightarrow$ 1Ch and 10h appears twice in succession, the continuity is lost at this part. Therefore interrupt is generated when starting data transfer of Eh byte.

Notes on Device Control

(1) Packet Data Transmission (ML60841 → USB Host)

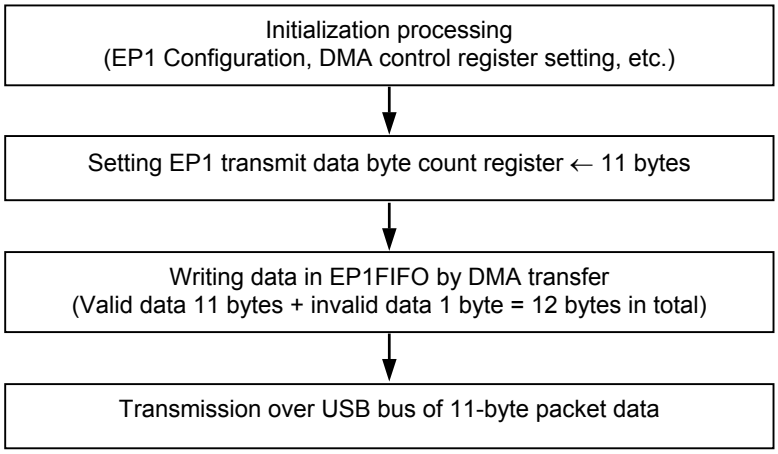
In the ML60841, it is possible to select a 16-bit or 32-bit data bus width for the microcontroller interface.
In addition, data transfer can be made only in the PIO mode with the FIFO for EP0 and EP3 packet data transfer, but can be made in either the PIO mode or the DMA mode with the FIFO for EP1, 2, 4, 5 packet data transfer.
For example, when an 11-byte packet data is written by PIO transfer into a FIFO for packet data transfer using a 16-bit data bus width, the number of bytes written to the FIFO will be 12 bytes. In other words, valid packet data (11 bytes) + invalid packet data (1 byte) will be written in the FIFO.
In the ML60841, the invalid packet data can be suppressed and only valid packet data can be transmitted over the USB bus by setting the number of valid packet data bytes per packet in the EPn transmit data byte count register.
This is explained below using some concrete examples of data transfer using a 16-bit data bus width.

Example 1: When 11 bytes of valid data + 1 byte of invalid data are written to EP1 by PIO transfer.



In Example 1 above, data writes to EP1FIFO are executed six times (12 bytes) with a data bus width of 16 bits.
However, since there are only 11 bytes of valid data, one byte written will be invalid data. Because the number of valid data bytes (11 bytes) is set in the EP1 transmit data byte count register, only the 11 bytes of valid data are transmitted over the USB bus after discarding the one byte of invalid data within the ML60841 after the transmit packet ready bit has been set.
Further, PIO transfer is one in which the target EP for DMA transfer has not been specified in bits 06:05 of the DMA control register, or DMA has been disabled in bit 00 although the target EP for DMA transfer has been specified.

Example 2: When 11 bytes of valid data + 1 byte of invalid data are written to EP1 by DMA transfer



In Example 2 above, data writes to EP1FIFO are executed six times (12 bytes) with a data bus width of 16 bits. However, since there are only 11 bytes of valid data, one byte in the written data will be invalid data. Because the number of valid data bytes (11 bytes) is written in the EP1 transmit data byte count register, after the DMA transfer of 12 bytes has been completed, the one invalid data byte is discarded within the ML60841, and only 11 bytes of valid data are transmitted over the USB bus.
Further, DMA transfer is one in which the target EP for DMA transfer has been specified in bits 06:05 of the DMA control register and also DMA transfer has been enabled in bit 00 of the DMA control register for the EP that has been specified as the target for DMA transfer.

The relationship among the value set in the transmit data byte count register during DMA transfer or PIO transfer, the number of bytes written to the FIFO for transferring packet data, and the number of bytes of packet data transmitted over the USB bus is shown in the table below.

	Number of bytes written to FIFO for packet data transfer	Value set in the transmit data byte count register	Number of bytes in the packet data transmitted over USB bus	Remarks
DMA Transfer	*1	M	M	
PIO Transfer	*2	N	N	

*1: Data transfer is requested by outputting DREQ until completion of the transfer of data including the number of bytes set in the transmit data byte count register (for example, 12 bytes of DMA transfer when the data bus width is 16 bits and M=11, 6 bytes of DMA transfer when M=6 and the data bus width is 16 bits, 12 bytes of DMA transfer when the data bus width is 32 bits and M=11, 8 bytes of DMA transfer when M=6 and the data bus width is 32 bits), the DREQ output is stopped as soon as the required number of bytes has been reached, and packet data containing M bytes of valid data is transmitted over the USB bus.

*2: Packet data is transmitted with N bytes, irrespective of the number of bytes written into the FIFO for packet data transfer, at the time when the packet ready bit is set, where N is the value set in the transmit data byte count register. In other words, if the number of bytes written into the FIFO for packet data transfer is less than N bytes, indeterminate data will be added to make up for the insufficient number of bytes and a packet data with N bytes is transmitted over the USB bus. If the number of data bytes written to the FIFO for packet data transfer is more than N bytes, the excess number of data bytes will be discarded and only N bytes of packet data are transmitted over the USB bus.

(2) ISO IN Transfer Control

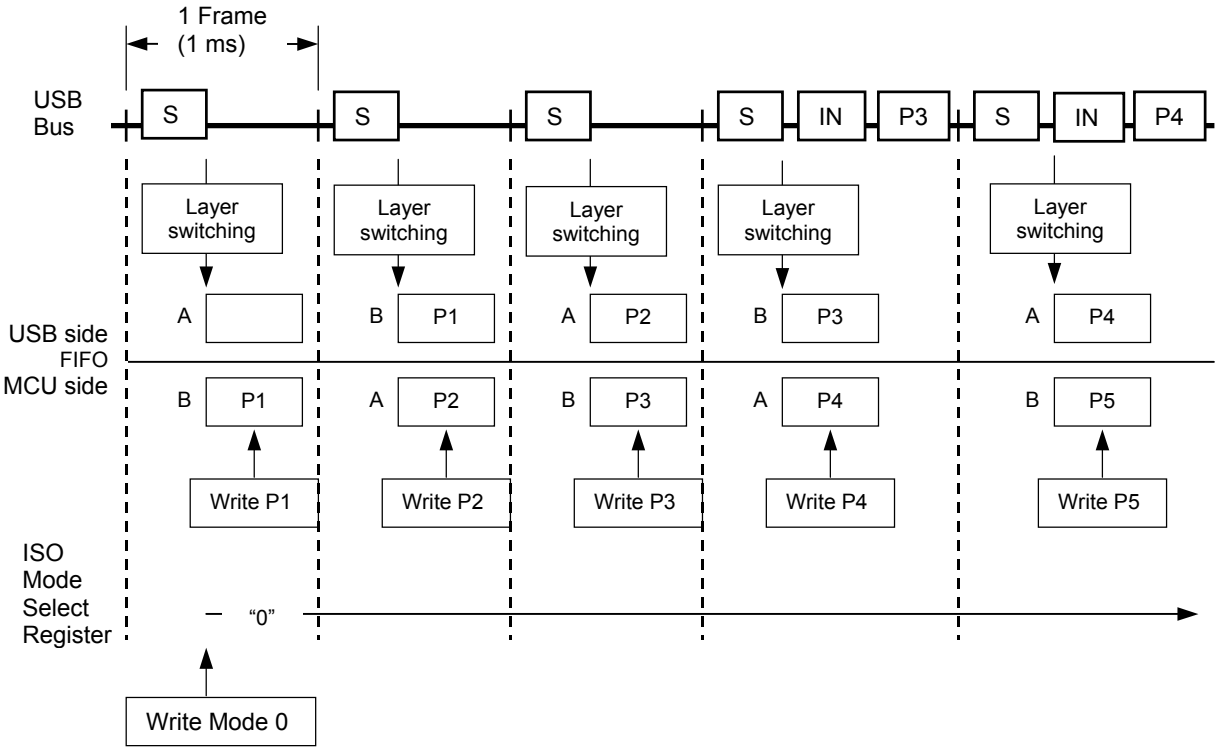
In the ML60841, when an EP has been configured for the ISO IN transfer mode, it is possible to specify the following two types of operation modes:

① In order to transfer data without breaks, the mode in which switching between the two FIFO layers is made every time an SOF is received (Mode 0)

② In order to avoid loss of data, the mode in which the switching between the two FIFO layers is made after the data written in the FIFO has been transmitted over the USB bus (when the SOF detection is made subsequent to receiving the IN token) (Mode 1)

These operating modes can be specified using the ISO mode select register (378h). The operations by the LSI in these two modes are shown below.

Operation equivalent to ML60852A (Corresponding bit is "0" in the ISO mode select register)



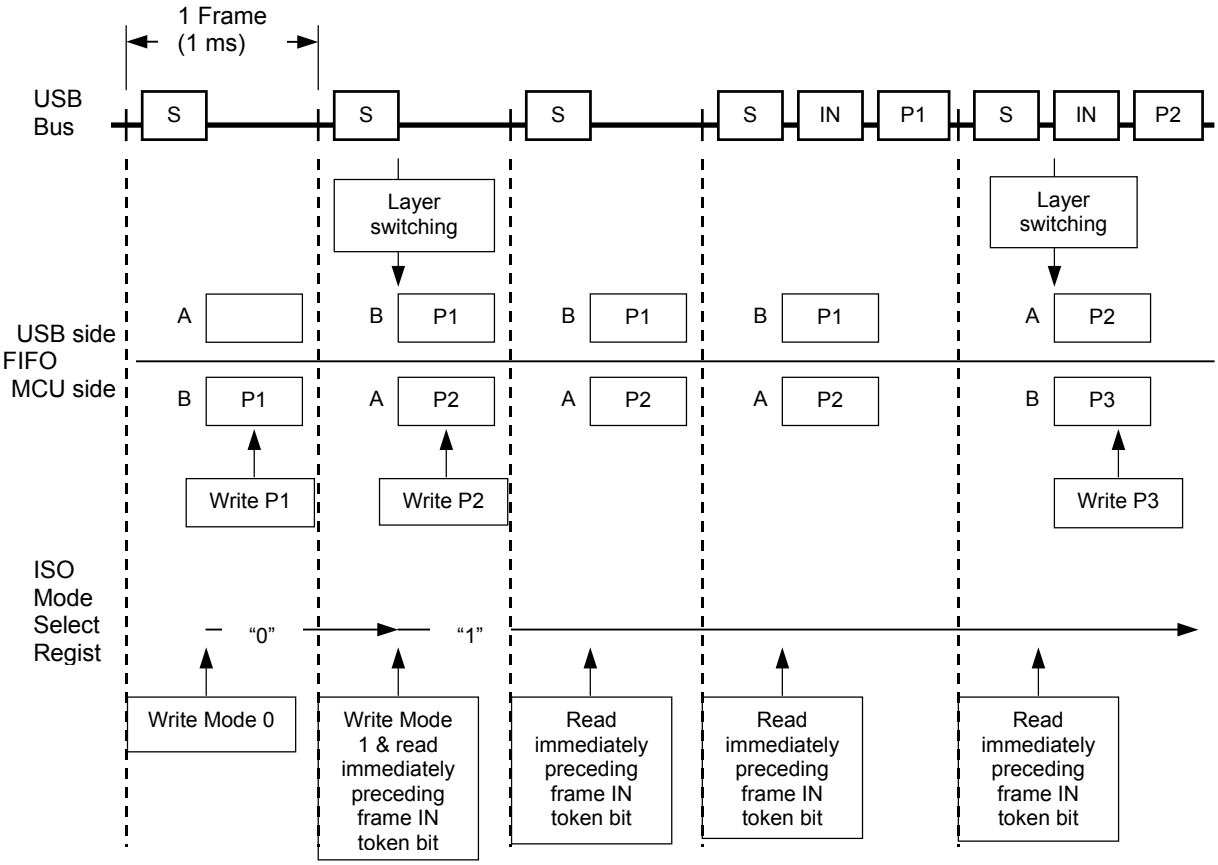
Note: S denotes an SOF, IN denotes an IN token, and Pn denotes data of packet No. n.

The above figure shows the case in which the packet data are written from the MCU side successively in the order P1 → P2 → … P5, and an IN token arrives from the USB host during the frame following the completion of writing packet P3.

The ISO mode select register is set to "0" after the first SOF interrupt, and there is no need to change the setting thereafter.

In this mode (Mode 0), since the layer is switched every time an SOF is received, the packet data prior to the packet data (P3 in the above figure) written during the frame period immediately preceding the IN token reception are not transmitted. The latest data P3 at the time an IN token is received will be transmitted in real time.

Operation of FIFO layer switching after transmitting FIFO data (Corresponding bit is "1" in the ISO mode select register)



Note: S denotes an SOF, IN denotes an IN token, and Pn denotes data of packet No. n.

The above figure shows the case in which the packet data are written from the MCU side successively in the order P1 → P2 → P3, and an IN token arrives from the USB host during the second frame after the completion of writing packet P2.
The ISO mode select register is set to "0" after the first SOF interrupt. By setting to Mode 0 at this time, the ML60841 automatically carries out FIFO layer switching at the next SOF. Mode 1 is set after the SOF interrupt next to the completion of writing the data of packet P1, and thereafter the operation in Mode 1 is specified.
After the specification of Mode 1, the bit 07 (immediately preceding frame IN token bit) of the EPn status register is read out at every SOF interrupt, and since packet data would have been transferred to the USB bus during the preceding frame if this bit has been set to "1", the ML60841 automatically switches the FIFO layer. Therefore, since one of the FIFO layer would be in the free state, the microcontroller writes packet data to the FIFO. Writing packet data to the FIFO will be suspended if this bit has been set to "0".
In this mode (Mode 1), since the FIFO layer is switched at the time of an SOF detection immediately after receiving an IN token from the USB host, the packet data written earlier by the microcontroller in the FIFO (the packet data P1 and P2 in the above figure) will not be lost. However, depending on the relationship between the timings of data writing by the microcontroller and IN token transmission from the USB host, it is possible that the latest data is not transmitted. Therefore, change to Mode 0 in applications that request for the transmission of the latest data.

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Condition	Rating	Unit
Supply voltage	V_{CC}	Ta = 25°C, V_{SS} = 0 V	–0.3 to +4.6	V
Input voltage	V_I		–0.3 to + V_{CC} + 0.3	V
Output current	I_O		–12 to +12	mA
Power dissipation (BGA)	Pb	Ta = 70°C	330	mW
Power dissipation (TQFP)	Pt		300	
Storage temperature	T_{STG}	—	–55 to +150	°C

RECOMMENDED OPERATING CONDITIONS

Parameter	Symbol	Condition	Range	Unit
Supply voltage	V_{CC}	—	3.0 to 3.6	V
Operating temperature	Ta	—	0 to 70	°C
Crystal frequency	f_{OSC}	—	48 ±0.12	MHz

ELECTRICAL CHARACTERISTICS

DC Characteristics (1)

(V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, Ta = 0 to +70°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit	Pin
"H" input voltage	V_{IH}		$V_{CC} \times 0.7$	—	—	V	Note 1
"L" input voltage	V_{IL}		—	—	0.8	V	
"H" input voltage	V_{IH}		$V_{CC} \times 0.8$	—	—	V	Xin
"L" input voltage	V_{IL}		—	—	$V_{CC} \times 0.2$	V	
Schmitt trigger input voltage	Vt+		—	—	$V_{CC} \times 0.7$	V	$\overline{\text{RESET}}$
	Vt–		0.8	—	—	V	
	ΔVt	(Vt+) – (Vt–)	—	0.4	—	V	
"H" output voltage	V_{OH}	I_{OH} = –100 μA	V_{CC} – 0.2	—	—	V	Note 2
		I_{OH} = –4 mA	2.2	—	—	V	
"L" output voltage	V_{OL}	I_{OL} = 100 μA	—	—	0.2	V	Note 3
		IOL = 4 mA	—	—	0.45	V	
"H" input current	I_{IH}	$V_{IH} = V_{CC}$	—	—	10	μA	Note 4
"L" input current	I_{IL}	$V_{IL} = V_{SS}$	–10	—	—	μA	
"H" input current	I_{IH}	$V_{IH} = V_{CC}$	—	—	10	μA	Note 5
"L" input current	I_{IL}	$V_{IL} = V_{SS}$	–200	—	–10	μA	
3-state output leakage current	I_{OZH}	$V_{OH} = V_{CC}$	—	—	10	μA	Note 6
	I_{OZL}	$V_{OL} = V_{SS}$	–10	—	—	μA	
Dynamic supply current	I_{DDO}	$V_{IH} = V_{DD}$ V_{IL} = GND Output open	—	—	50	mA	V_{CC}
Static supply current	I_{DDS}	Ta = 0 to 50°C, Note 7	—	—	100	μA	
		Ta = 50 to 70°C, Note 7	—	—	200		

(Note 1) Applicable to pins corresponding to I/O Type "*2, *3, *4" of list of parameters of PIN CONFIGURATION.

(Note 2) Applicable to pins corresponding to I/O Type "*3, *5" of list of parameters of PIN CONFIGURATION.

(Note 3) Applicable to pins corresponding to I/O Type "*3, *5, *6" of list of parameters of PIN CONFIGURATION.

(Note 4) Applicable to pins corresponding to I/O Type "*4" of list of parameters of PIN CONFIGURATION.

(Note 5) Applicable to pins corresponding to I/O Type "*2" of list of parameters of PIN CONFIGURATION.

(Note 6) Applicable to pins corresponding to I/O Type "*3" of list of parameters of PIN CONFIGURATION.

(Note 7) The condition when 3 μs have elapsed after writing a "1" in bit 01 of the SOFT RESET/POWER DOWN Control Register. All output pins open.

DC Characteristics (2) – USB Port

(V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, Top = 0 to +70°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit	Pin
Differential input sensitivity	V_{DI}	|(D+) – (D–)|	0.2	—	—	V	D+, D–
Differential common mode range	V_{CM}	Including V_{DI}	0.8	—	2.5	V	
Single-ended receiver threshold	V_{SE}		0.8	—	2.0	V	
"H" output voltage	V_{OH}	15 kΩ to GND	2.8	—	3.6	V	
"L" output voltage	V_{OL}	1.5 kΩ to 3.6 V	—	—	0.3	V	
Output leakage current	I_{LO}	0 V < V_{IN} < V_{CC}	–10	—	+10	μA	

AC Characteristics – USB Port Section (Full-Speed)

(V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, Top = 0 to +70°C)

Parameter	Symbol	Condition (Note 1)	Min.	Typ.	Max.	Unit	Pin
Rise time	T_R	C_L = 50 pF	4	—	20	ns	D+, D–
Fall time	T_F	C_L = 50 pF	4	—	20	ns	
Output signal crossover voltage	V_{CRS}		1.3	—	2	V	
Driver output resistance	ZDRV	In normally operating state	28	—	44	Ω	
Data rate	T_{DRATE}	Average bit rate (12 Mbps ±0.25%)	11.97	—	12.03	Mbps	

(Note 1) TR and TF are transition times of 10% point and 90% point of amplitude.

AC Characteristics – USB Port Section (Low-Speed)

(V_{CC} = 3.0 to 3.6 V, V_{SS} = 0 V, Top = 0 to +70°C)

Parameter	Symbol	Condition (Note 1)	Min.	Typ.	Max.	Unit	Pin
Rise time	T_R	C_L = 150 pF	75	—	300	ns	D+, D–
Fall time	T_F	C_L = 150 pF	75	—	300	ns	
Output signal crossover voltage	V_{CRS}		1.3	—	2	V	
Data rate	T_{DRATE}	Average bit rate (1.5 Mbps ±0.25%)	1.4775	—	1.5225	Mbps	

(Note 1) T_R and T_F are transition time of 10% point and 90% point of amplitude.

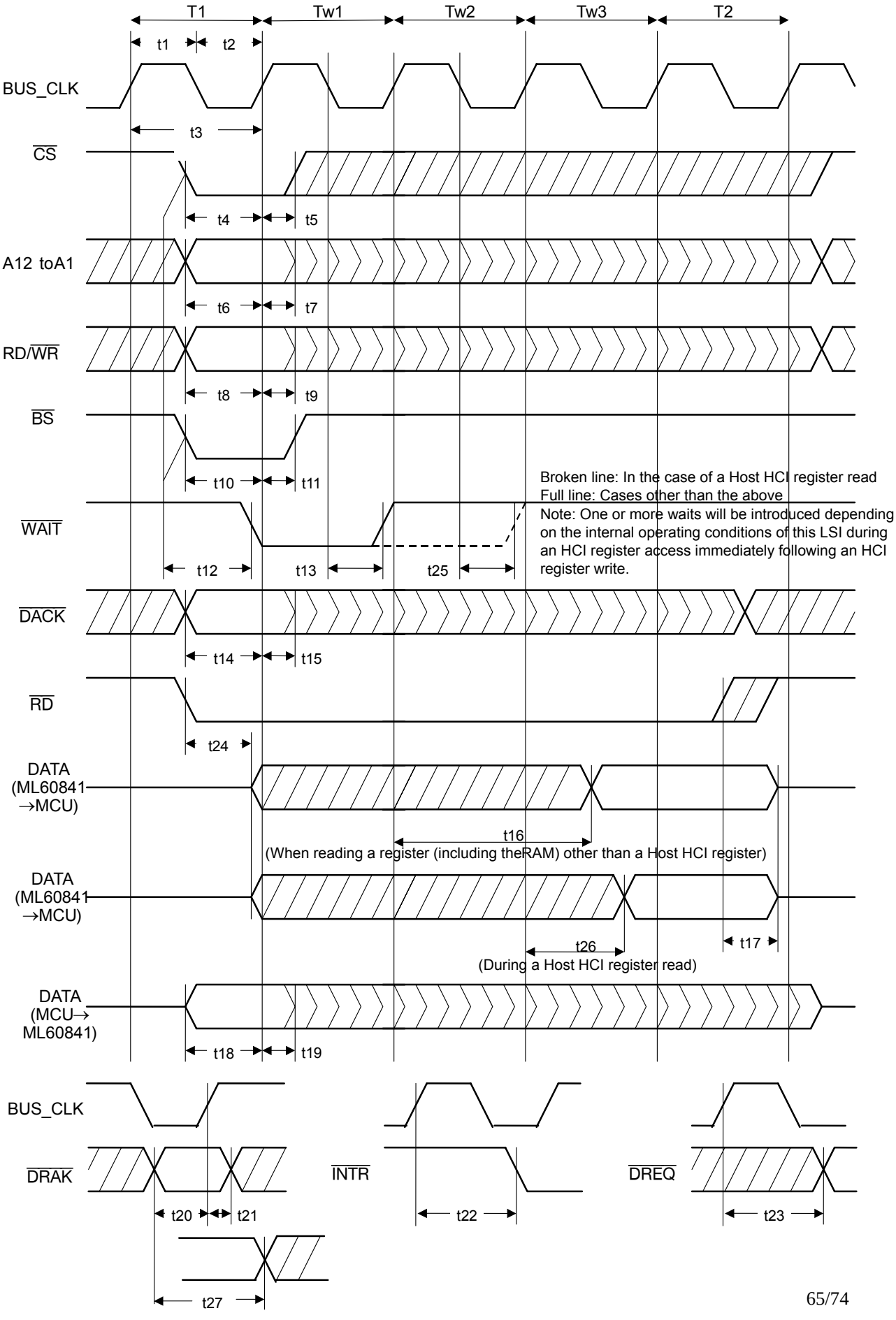
AC Characteristics – Microcontroller Interface Section

(V_{CC} = 3.0 to 3.6V, V_{SS} = 0 V, Ta = 0 to +70°C, Load capacitance 20 pF)

Parameter	Symbol	Min.	Max.	Unit
High level time of BUS_CLK	t1	8		ns
Low level time of BUS_CLK	t2	8		
BUS CLK period	t3	30		
$\overline{CS}$ setup time for rise of BUS_CLK	t4	15		
$\overline{CS}$ hold time for rise of BUS_CLK	t5	10		
A12:A1 setup time for rise of BUS_CLK	t6	15		
A12:A1 hold time for rise of BUS_CLK	t7	t1 + 10		
RD/$\overline{WR}$ setup time for rise of BUS_CLK	t8	15		
RD/$\overline{WR}$ hold time for rise of BUS_CLK	t9	10		
$\overline{BS}$ setup time for rise of BUS_CLK	t10	15		
$\overline{BS}$ hold time for rise of BUS_CLK	t11	0		
Delay time from rise of BUS_CLK till $\overline{WAIT}$ becomes valid	t12		18	
Delay time from rise of BUS_CLK till $\overline{WAIT}$ becomes invalid	t13	4	18	
$\overline{DACKn}$ setup time for rise of BUS_CLK	t14	15		
$\overline{DACKn}$ hold time for rise of BUS_CLK	t15	10		
Delay time from rise of BUS_CLK till D31:D0 (at the time of read) are set up	t16		44	
Delay time from rise of BUS_CLK till D31:D0 (at the time of read) become Hi-Z	t17	0	10	
D31:D0 (at the time of write) setup time for rise of BUS_CLK	t18	13		
D31:D0 (at the time of write) hold time for rise of BUS_CLK	t19	10		
$\overline{DRAKn}$ setup time for rise of BUS_CLK	t20	15		
$\overline{DRAKn}$ hold time for rise of BUS_CLK	t21	0		
Delay time from rise of BUS_CLK till $\overline{INTR}$ becomes valid	t22		18	
Delay time from rise of BUS_CLK till $\overline{DREQn}$ becomes valid	t23		18	
Delay time from both $\overline{CS}$ and $\overline{RD}$ being enabled until D31:D0 are output	t24	0		
Delay time from falling edge of BUS_CLK until $\overline{WAIT}$ becomes disabled (When reading the Host HCI register)	t25	4	18	
Delay time from rising edge of BUS_CLK until valid data is output in D31:D0 (during read) (When reading the Host HCI register)	t26		26	
Delay time from $\overline{DRAK}$ being enabled until $\overline{DREQn}$ becomes disabled	t27		19	

Note 1: The maximum delay time is 10 ns for the delay time (t17) from either $\overline{CS}$ or $\overline{RD}$ becoming disabled until D31:D0 (during read) become high impedance. If $\overline{CS}$ and $\overline{RD}$ disable timings become after the trailing clock rising edge of T2 cycle and if also the access cycle following the read access cycle is a write access cycle, then a bus collision may be caused by this delay time.
Therefore, when the above condition occurs, it is necessary to introduce between the access cycles an idle cycle that is one clock cycle or longer.

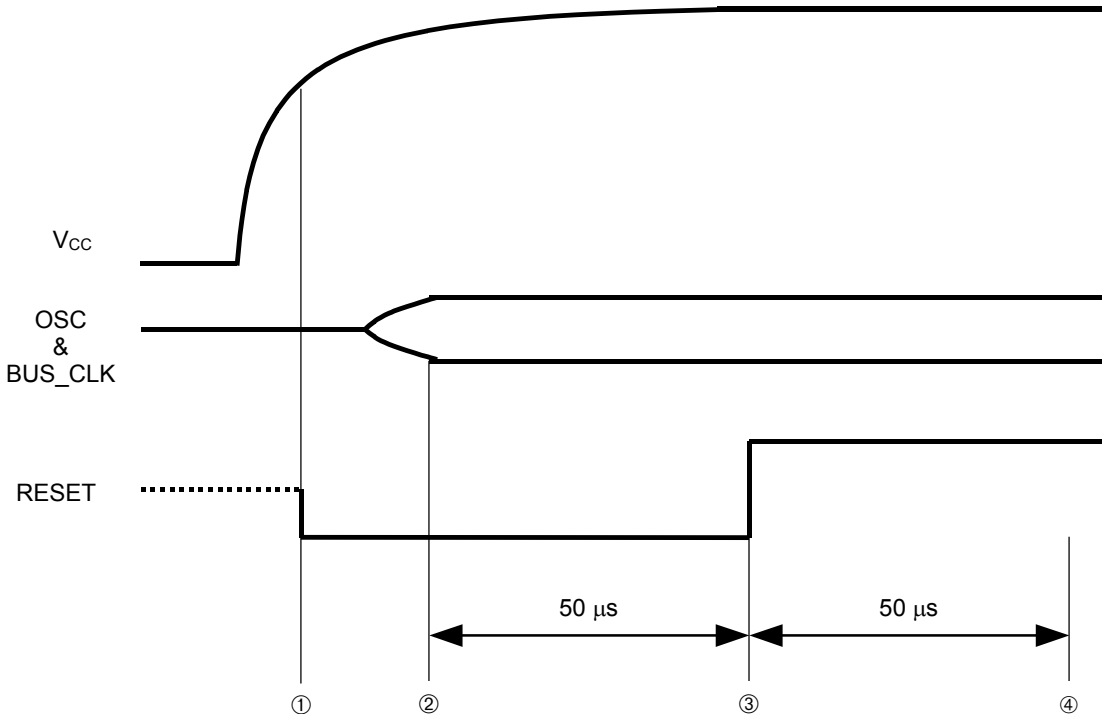
Note 2: The maximum delay time is 44 ns for the delay time (+16) from the rising edge of BUS_CLK until valid data is output in D31:D0 (during read). Introduce a software wait (minimum 3 waits) if the setup time cannot be satisfied at the time of taking in the read data.
The timing specification diagram shown in the next page is the case when 3 wait states are introduced by software.


T1
Tw1
Tw2
Tw3
T2
t1
t2
BUS_CLK
t3
CS
t4
t5
A12 toA1
t6
t7
RD/WR
t8
t9
BS
t10
t11
WAIT
t12
t13
t25
Broken line: In the case of a Host HCI register read
Full line: Cases other than the above
Note: One or more waits will be introduced depending on the internal operating conditions of this LSI during an HCI register access immediately following an HCI register write.
DACK
t14
t15
RD
t24
DATA (ML60841 →MCU)
t16
(When reading a register (including the RAM) other than a Host HCI register)
DATA (ML60841 →MCU)
t26
t17
(During a Host HCI register read)
DATA (MCU→ ML60841)
t18
t19
BUS_CLK
DRAK
t20
t21
t27
INTR
t22
DREQ
t23

TIMING DIAGRAM

Basic Bus Cycle

(1) Reset



① V_{CC} rises and the RESET signal is asserted.

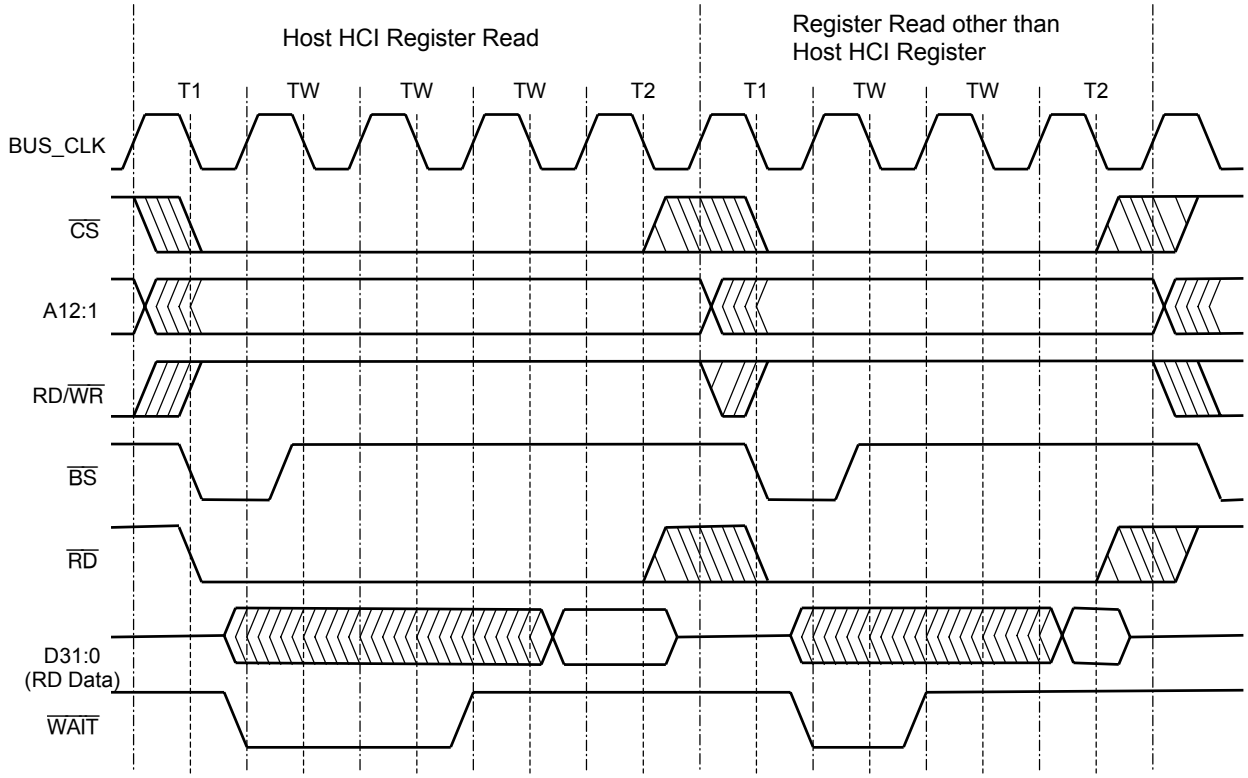
② Oscillation (48 MHz) of a crystal oscillator becomes stable and BUS_CLK is input.
Since oscillation stabilization time (① to ②) of a crystal oscillator differs depending on the crystal oscillator used, take suitable time.

③ RESET signal is deasserted.
Hold the RESET signal active for at least 50 μs (② to ③) for the ML60841 to be initialized.

④ Internal setup of the ML60841 is completed and access from microcontroller becomes possible.
Do not access the ML60841 before step ④.

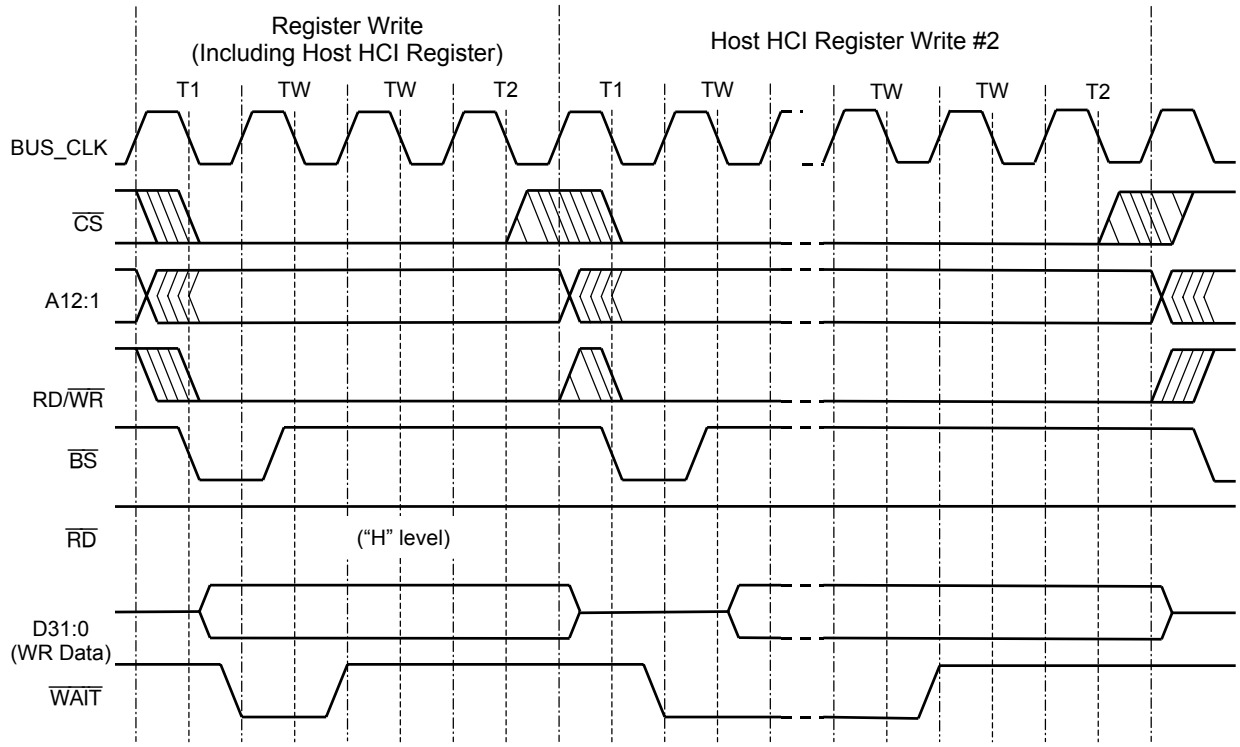
Supply the BUS_CLK (bus clock) signal to the ML60841 not later than step ②.

(2) PIO Read (ML60841 to Microcontroller)



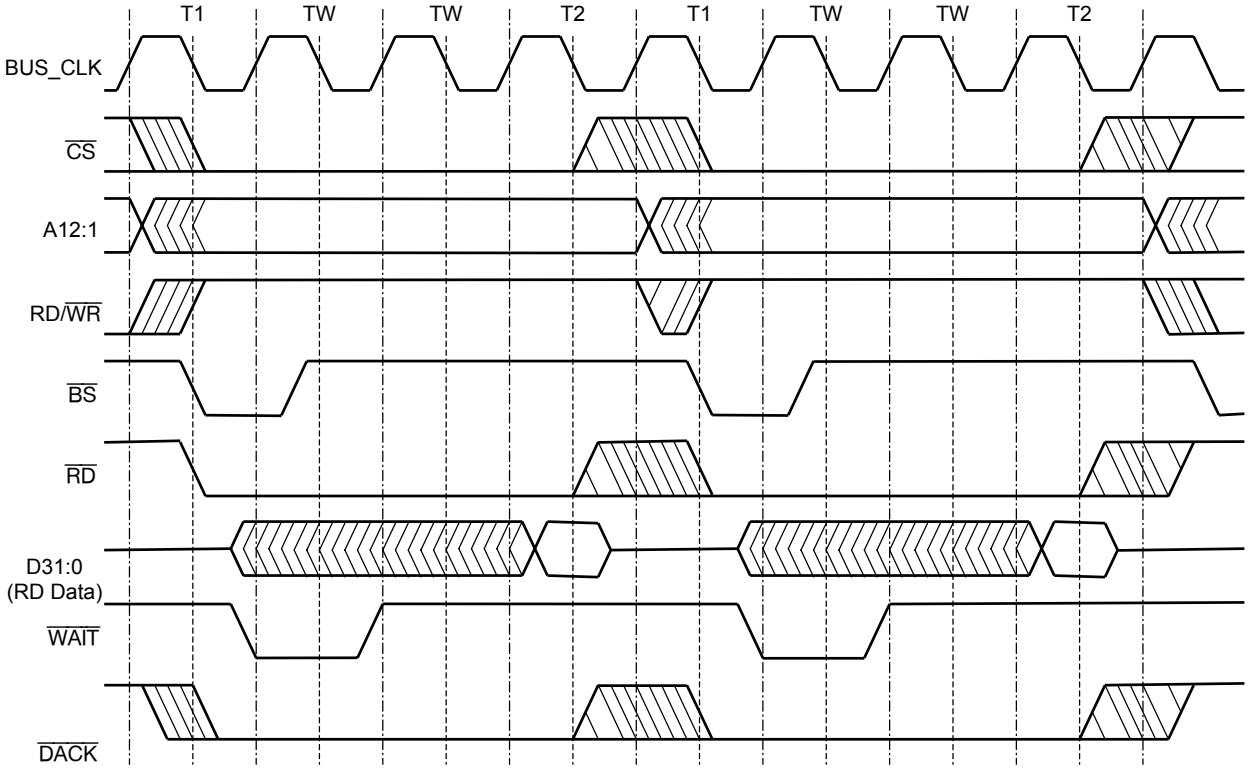
(Note) Regarding access to the HCI register immediately after HCI register write, one or more waits are additionally inserted according to the internal operation status of the ML60841.

(3) PIO Write (Microcontroller to ML60841)



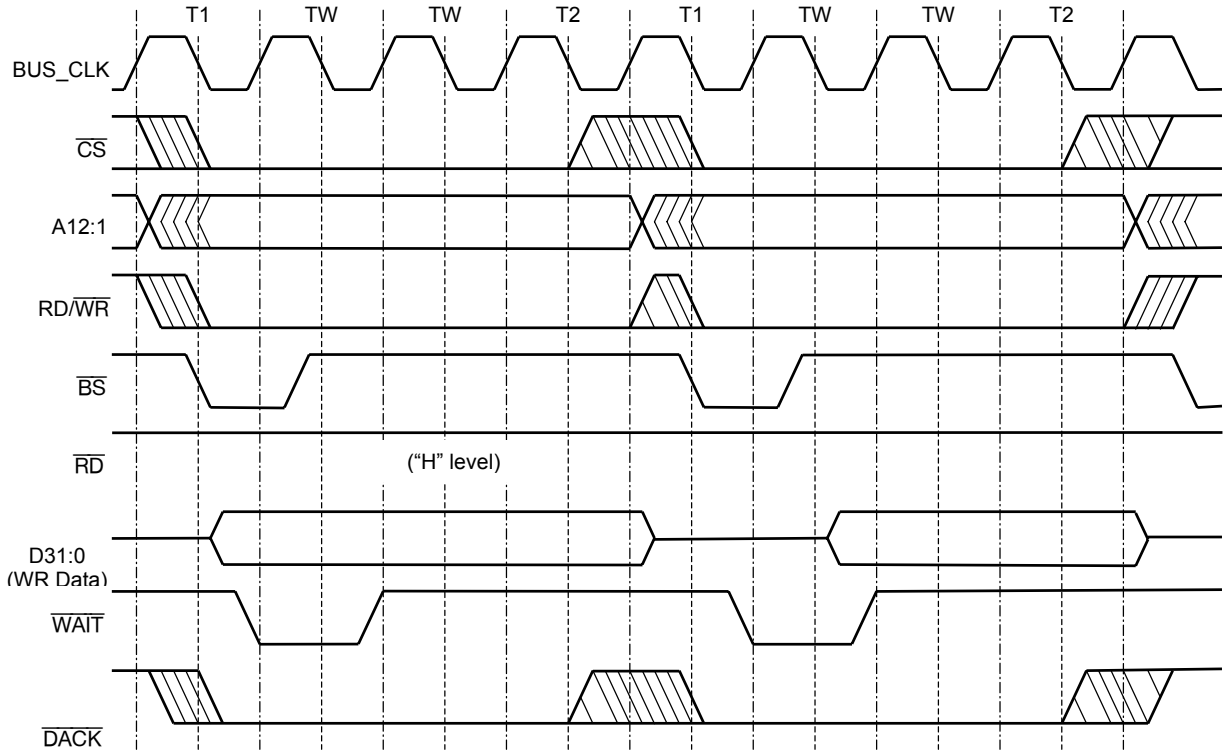
(Note) Regarding access to the HCI register immediately after HCI register write, one or more waits are additionally inserted according to the internal operation status of the ML60841. The Host HCI Reg Write #1 shown in the above diagram represents the case when the wait is not additionally inserted and Host HCI Rer Write #2 represents the case when the wait is additionally inserted.

(4) Dual Address DMA Read (ML60841 to Memory)



(Note) Insert three or more waits at the time of OHCI register read and insert two or more waits when reading a register other than OHCI (including the device controller).

(5) Dual Address DMA Write (Memory to ML60841)

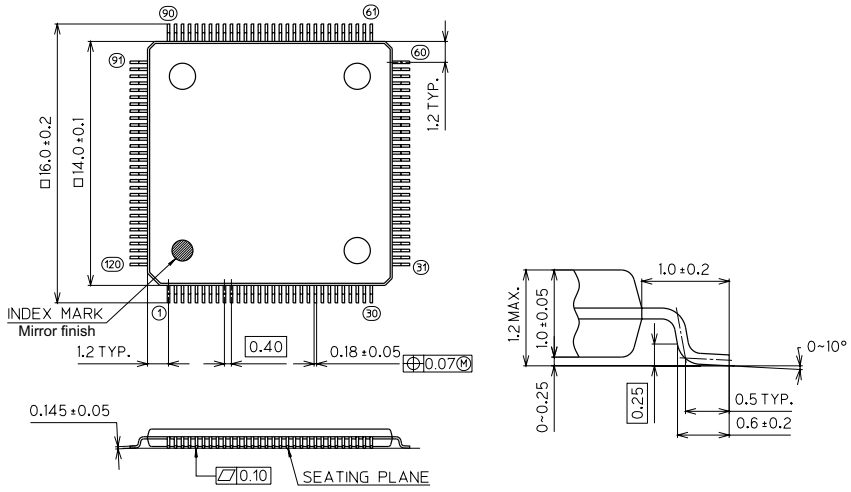


(Note) Insert two or more waits.

PACKAGE DIMENSIONS

(Unit: mm)

TQFP120-P-1414-0.40-K



Package material	Epoxy resin
Lead frame material	42 alloy
Pin treatment	Solder plating (≥5µm)
Package weight (g)	0.55 TYP.
Rev. No./Last Revised	3/Nov. 12, 1998

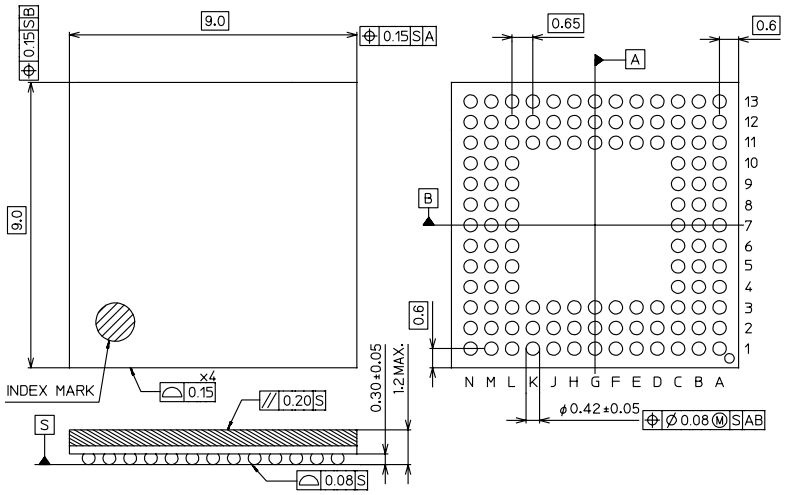
Oki Electric Industry Co., Ltd.

Notes for Mounting the Surface Mount Type Package

The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage.
Therefore, before you perform reflow mounting, contact Oki's responsible sales person for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

(Unit: mm)

P-TFBGA120-0909-0.65



Package material	Epoxy resin
Ball material	Sn/Pb
Package weight (g)	0.16 TYP.
Rev. No./Last Revised	1/Mar. 11, 2002

Oki Electric Industry Co., Ltd.

Notes for Mounting the Surface Mount Type Package

The surface mount type packages are very susceptible to heat in reflow mounting and humidity absorbed in storage.
Therefore, before you perform reflow mounting, contact Oki's responsible sales person for the product name, package name, pin number, package code and desired mounting conditions (reflow method, temperature and times).

REVISION HISTORY

Document No.	Date	Page		Description
		Previous Edition	Current Edition	
PEDL60841-01	Oct. 2, 2002	–	–	Preliminary first edition

NOTICE

1. The information contained herein can change without notice owing to product and/or technical improvements. Before using the product, please make sure that the information being referred to is up-to-date.

2. The outline of action and examples for application circuits described herein have been chosen as an explanation for the standard action and performance of the product. When planning to use the product, please ensure that the external conditions are reflected in the actual circuit, assembly, and program designs.

3. When designing your product, please use our product below the specified maximum ratings and within the specified operating ranges including, but not limited to, operating voltage, power dissipation, and operating temperature.

4. Oki assumes no responsibility or liability whatsoever for any failure or unusual or unexpected operation resulting from misuse, neglect, improper installation, repair, alteration or accident, improper handling, or unusual physical or electrical stress including, but not limited to, exposure to parameters beyond the specified maximum ratings or operation outside the specified operating range.

5. Neither indemnity against nor license of a third party's industrial and intellectual property right, etc. is granted by us in connection with the use of the product and/or the information and drawings contained herein. No responsibility is assumed by us for any infringement of a third party's right which may result from the use thereof.

6. The products listed in this document are intended for use in general electronics equipment for commercial applications (e.g., office automation, communication equipment, measurement equipment, consumer electronics, etc.). These products are not authorized for use in any system or application that requires special or enhanced quality and reliability characteristics nor in any system or application where the failure of such system or application may result in the loss or damage of property, or death or injury to humans.
   Such applications include, but are not limited to, traffic and automotive equipment, safety devices, aerospace equipment, nuclear power control, medical equipment, and life-support systems.

7. Certain products in this document may need government approval before they can be exported to particular countries. The purchaser assumes the responsibility of determining the legality of export of these products and will take appropriate and necessary steps at their own expense for these.

8. No part of the contents contained herein may be reprinted or reproduced without our prior permission.