1. General Description

KL5C80A16 is a high speed microcontroller developed with the state-of-the-art 0.8μm CMOS process. As CPU core it uses KC82, which is binary compatible with Zilog's Z80, and contains MMU to expand address space. The KC82 is a next generation 8-bit CPU core which executes instructions about four times faster than the Z80 (10MHz) and offers higher performance than typical 16-bit CPUs. In addition to the CPU core, KL5C80A16 contains a high speed DMA controller, an interrupt controller, a 16-bit high performance timer/counter, an asynchronous serial interface (UART), 32 parallel ports, and a DRAM controller providing all requirements for high performance and compact system. It has also low power dissipation and is suitable for application to portable devices.

Features

1) Binary compatible with the Z80 CPU,4 times faster than Z80 at the same clock rate
2) Built-in MMU which expands address space to 1MB
3) Two channels of high speed UART (Universal Asynchronous Receiver Transmitter)
4) Two channels of clock synchronous serial I/Os
5) Four high performance 16-bit timer/counters with 8-bit prescaler
6) Sixteen internal/external interrupts (flexible priority), one non-maskable interrupt
7) 32 parallel ports
8) DRAM controller which can be connected to a DRAM chip directly
9) On-chip external memory chip select circuit
10) On-chip crystal oscillator buffer
11) Maximum operating frequency 10MHz
12) Low power dissipation

The KL5C80A16's CPU core (KC82) uses synchronous internal bus. Its internal I/Os includes a bus interface suitable for the KC82's synchronous bus so that they are connected directly. When connecting I/O and memory to KL5C80A16, it is required to convert KC82's synchronous bus signals to asynchronous bus signals which can be input to ordinary memories and I/Os. For this purpose KL5C80A16 contains a circuit (external bus interface unit), which outputs Read/Write signals (EMRD_, EMWR_, EIORD_, EIOWR_) appropriate for external asynchronous buses. Unlike Read/Write signals of internal bus cycle on the synchronous bus, these Read/Write signals of external bus cycle are strobe signals which can be directly connected to external memories. In this Manual the KC82's synchronous bus is referred to as internal bus cycle and the external asynchronous bus converted by the external bus interface unit as external bus cycle.

All bus cycles in the figures of the chapter "KC82 CPU" are shown in terms of internal bus cycle, that is, the operations of KC82 in that chapter are described as bus cycles inside KL5C80A16 are described. Therefore, a wait state may be inserted by the external wait input or wait state controller when accessing external memories or I/Os in the same operations as shown in that chapter. Refer to the chapter on External Bus Interface Unit for more information on external bus cycle.

The external bus interface unit contains a wait state controller to enable an efficient connection with external memories. When accessing external memory or I/O, this controller automatically inserts wait states according to System Control Register, and generates various external accesses such as external memory access and external I/O access. The controller is designed to connect two kinds of memories with different access time efficiently, such as high speed SRAM and EPROM, by controlling 1MB internal physical memory divided in two. The external bus interface unit contains a DRAM controller circuit. This DRAM controller consists of a refresh counter, a timing circuit, and a row/column address multiplexer. It can be connected to the DRAM directly and the system can be compact. By bank configuration of a part of the address space, large size DRAM can be used.

In addition to the system control register, an external input BFMOD is provided to set the mode. When BFMOD input is "H" and a Bug Finder adapter is connected to the external pin BFSIO, our simple debugging tool (Bug Finder) is started just after reset. If RAM is connected in place of external ROM, a program can be downloaded from a personal computer to the RAM and debugging operation can be performed on this RAM just like an ICE. The Bug Finder adapter is sold by several vendors. Please ask about them to local agents.

2. Block Diagram

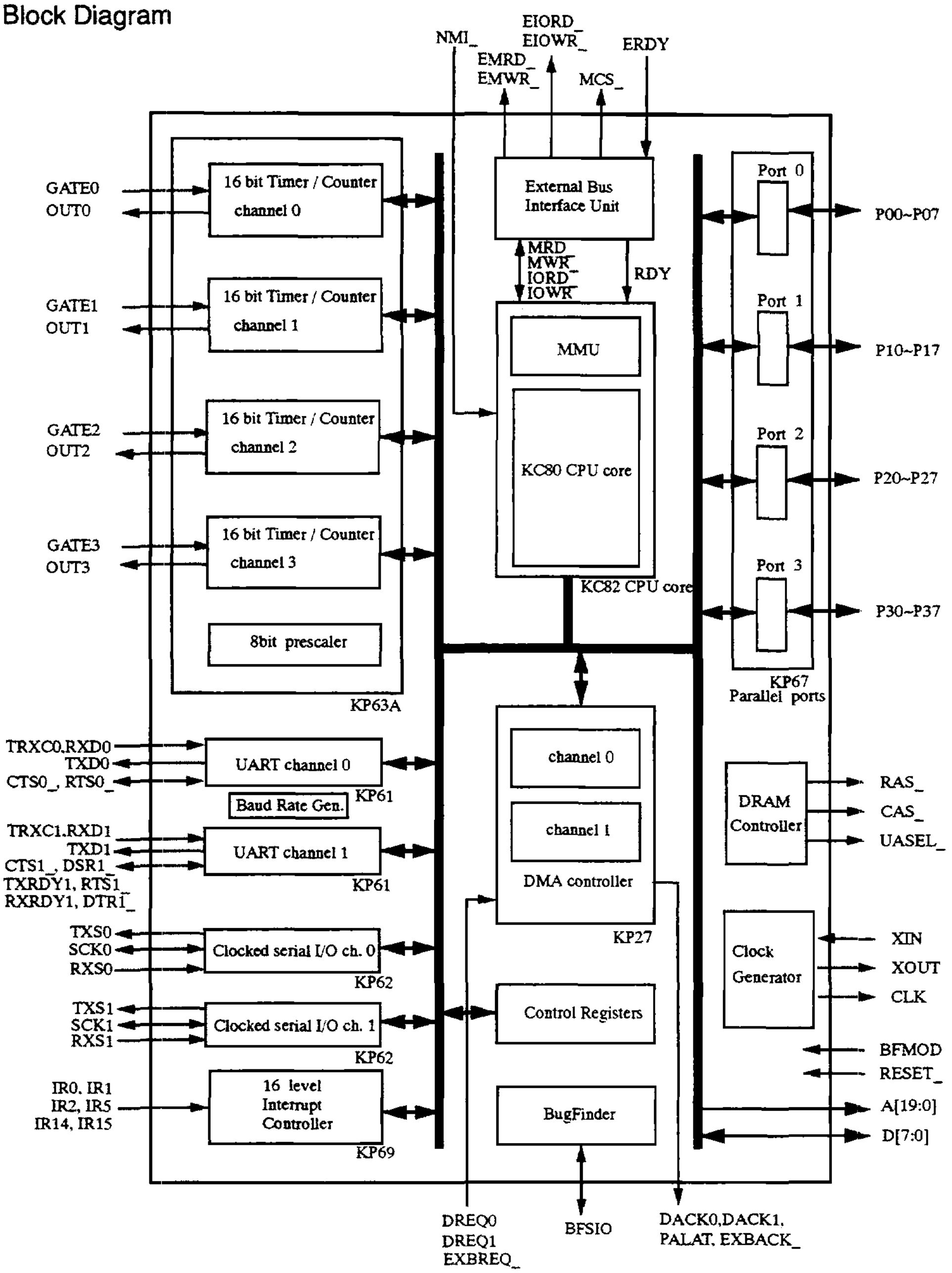


Fig. 2-1 KL5C80A16 Block Diagram

3. Pin Description

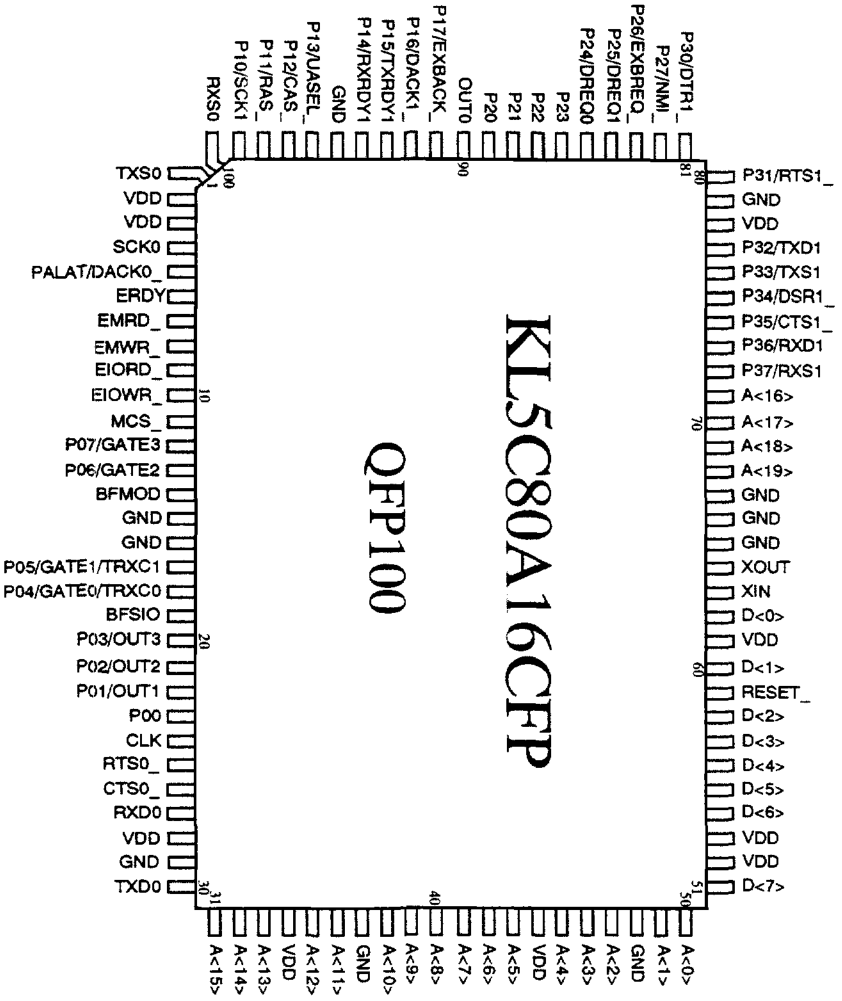


Figure 3-1. Pin Configuration (Top View)

Some KL5C80A16 pins are parallel port pins and multiplexed pins.
Please refer to the pin configuration on the previous page for the information on multiplexing.
Please refer to chapter 12 for these multiplexed pin controls.

Table 3-1. Dedicated Pins

pin name	I/O	description
GND	power	supply Connect 0V.
VDD	power	supply Connect 5V.
EMRD_	O	External memory read signal. This signal is obtained by converting the KC82's internal bus signal MRD_ for external bus by the external bus interface unit.
EMWR_	O	External memory write signal. This signal is obtained by converting the KC82's internal bus signal MWR_ for external bus by the external bus interface unit.
EIORD_	O	External I/O device read signal. This signal is obtained by converting the KC82's internal bus signal IORD_ for external bus by the external bus interface unit. This signal remains "H" in internal I/O access.
EIOWR_	O	External I/O device write signal. This signal is obtained by converting the KC82's internal bus signal IOWR_ for external bus by the external bus interface unit. This signal remains "H" in internal I/O access.
CLK	O	Outputs the internal clock generated by crystal oscillator to external circuit.
BFSIO	I/O	Dedicated port for Bug Finder debugging tool. Leave open when the Bug Finder is not used.
BFMOD	I	Input for setting the mode. Sets the KL5C80A16's operation mode.
ERDY	I	External wait request input. Wait cycles are inserted, when ERDY is L.
A[19:0]	O	Address output. CPU's A[19:0] are connected.
D[7:0]	I/O	External data bus.
RESET_	I	Reset input. The KL5C80A16 is reset when this signal goes "L".
NMI_	I	Non-Maskable interrupt input. Falling edge trigger. This input accepts the non-maskable interrupt. The priority of this input is higher than maskable interrupt but lower than BREQ_. As soon as current instructions execution is completed, it jumps to 0066H and executes the interrupt service routine regardless of the status of interrupt enable flag. Please refer to the chapter 7 for the bus request priority of internal DMA.
XIN	For crystal oscillator	Connects an external crystal oscillator to the built-in clock generator. Use a clock of twice the system clock frequency. Input a clock of twice the system clock frequency when a crystal oscillator is not used.
XOUT	For crystal oscillator	Connects an external crystal oscillator to the built-in clock generator. Use a clock of twice the system clock frequency.
RAS_	O	DRAM controller RAS_ output. When DRAMs are connected, this pin is connected to RAS_ of DRAM. For more information, see chapter 4.

pin name	I/O	description
CAS_	O	DRAM controller CAS_ output. When DRAMs are connected, this pin is connected to CAS_ of DRAM. For more information, see chapter 4.
UASEL_	O	DRAM controller upper address selecting output. This output "L" indicates the lower address bus outputs row address. This is used when the ICE is used.
MCS_	O	External RAM chip select output. This output goes "L" when address = C0000H ~ FFFFFH.
GATE3~0	I	GATE input to timer/counter channel 3~0. For more information, see chapter 9.
OUT3~0	O	OUT output from timer/counter channel 3~0. For more information, see chapter 9.
DREQ0 DREQ1	I	DMA request signal input pin. Input DMA request signals to each channel. The polarity can be selected, H or L. For more information, see chapter 7.
DACK0_ DACK1_	O	DMA request acknowledge signal output pin During DMA transfer, these pins output acknowledge signal to the device subject to DMA transfer. "L" means DMA request is accepted. For more information, see chapter 7.
EXBREQ_	I	External device bus request signal input pin. When an external device is connected, its bus request signal is input. Active low. For more information, see chapter 7.
EXBACK_	O	External device bus request acknowledge signal output pin. When an external device is connected, its bus request acknowledge signal input is connected to this output. For more information, see chapter 7.
PALAT	O	I/O address latch signal output. For more information, see chapter 7.
TRXC0 TRXC1	I	UART transmission reception clock input pin. Input the clock signal which controls baud rate at data transmission and reception. For more information, see chapter 8.
TXD0 TXD1	O	UART transmission data output pin. This pin outputs serial transmission data. For more information, see chapter 8.
RXD0 RXD1	I	UART reception data input pin. This pin inputs serial data to be received. For more information, see chapter 8.

pin name	I/O	description
CTS0_	I	UART clear to send signal input pin. For more information, see chapter 8.
DSR1_	I	UART data set ready signal input pin. For more information, see chapter 8.
RTS0_ RTS1_	O	UART request to send signal output pin. For more information, see chapter 8.
DTR1_	O	UART data terminal ready signal output pin. For more information, see chapter 8.
TXRDY1	O	UART transmission ready signal output pin. For more information, see chapter 8.
RXRDY1	O	UART reception ready signal output pin. For more information, see chapter 8.
SCK0 SCK1	I/O	Serial clock input/output pin for Clocked Serial I/O. For more information, see chapter 10.
RXS0 RXS1	I	Reception data input pin for Clocked Serial I/O. For more information, see chapter 10.
TXS0 TXS1	O	Transmission data output pin for Clocked Serial I/O. For more information, see chapter 10.
P00~P07	I/O	P0 port of parallel port. For more information, see chapter 11.
P10~P17	I/O	P1 port of parallel port. For more information, see chapter 11.
P20~P27	I/O	P2 port of parallel port. For more information, see chapter 11.
P30~P37	I/O	P3 port of parallel port. For more information, see chapter 11.

The numbers after pin names such as 0, 1, 2, and 3 indicate channel0, channel1, channel2, and channel3 respectively.

BFMOD pin has pull-down resistor. RESET_ pin has pull-up resistor. About those characteristics (current value), see chapter 15.

4. External Bus Interface Unit

4. 1 Outline

The external bus interface unit includes a wait state controller, and converts internal bus cycles to external bus cycles inserting specified number of wait states. The number of wait states is specified by bit 7 ~ 4 of SCR4 (System Control Register 4). This unit works only at the access of external I/O and external memory. The Read/Write signal of external bus stays "H" at the access of internal I/Os. SCR4 becomes 00H on reset.

SCR4(I/O address = 1FH)

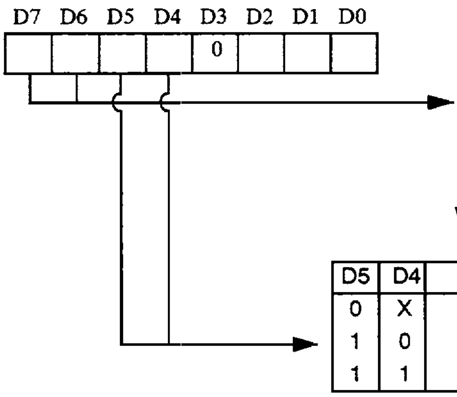


Table 4-1 External I/O wait control

D7	D6	external I/O
0	0	1 wait (3 clock/bus cycle)
0	1	2 wait (4 clock/bus cycle)
1	0	3 wait (5 clock/bus cycle)
1	1	4 wait (6 clock/bus cycle)

In case of 4 wait, starting edge of EIORD_/EIOWR_ is delayed with 1/2 clock.

Table 4-2 External memory wait control

D5	D4	address: 00000H~7FFFFH	address: 80000H~FFFFFH
0	X	1 wait	1 wait*
1	0	1 wait	0 wait*
1	1	0 wait	0 wait*

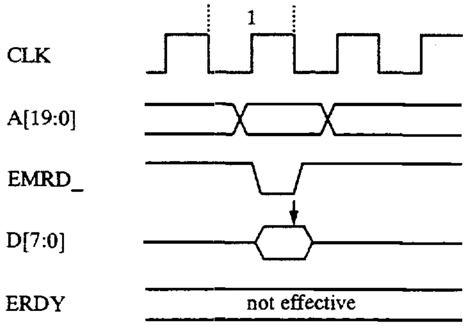
*In case that DRAM is used, the unit does not follow this table.

4.2 General description

The KL5C80A16's CPU core (KC82) uses synchronous system bus. Its internal I/Os include a bus interface suitable for the KC82's synchronous bus so that they are connected directly. When connecting I/O and memory to KL5C80A16, the external bus interface unit converts KC82's synchronous bus Read/ Write signals (MRD_, MWR_, IORD_, IOWR_) to external asynchronous bus Read/Write signals (EMRD_, EMWR_, EIORD_, EIOWR_) described later. The external bus interface unit includes a wait state controller, and converts internal bus cycles to external bus cycles inserting specified number of wait states. The number of wait states is specified by bit 7~4 of SCR4. The external bus interface unit adds a wait request from the built-in wait state controller to an external wait request input from ERDY, and send them to the CPU.

This controller is designed to connect two kinds of memories with different access time efficiently, such as high speed SRAM and EPROM, by controlling 1MB physical memory space divided in two. It should be noted that a external wait signal (ERDY) is ignored in the external memory access without wait cycle. This external bus interface unit also includes a DRAM controller. Three different memories with different access times, such as fast SRAM, DRAM and normal EPROM, can be connected directly. DRAM access cycles through the integrated DRAM controller is always three clock cycle. See section 4.4 for more detail.

4.3 External CPU bus cycles



0 wait or 1 wait can be selected for memory access cycle. However, ERDY input is neglected in case of 0 wait setting. On resetting, this setting is initialized to 1 wait.

Figure 4-1 External memory read with 0 wait setting

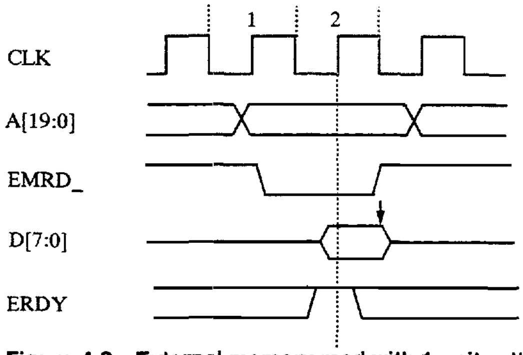
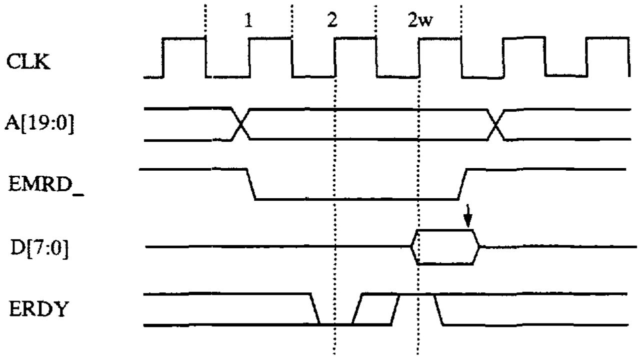
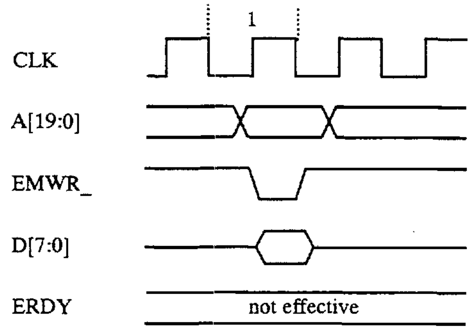


Figure 4-2 External memory read with 1 wait setting



2w: The wait cycle requested by external wait request signal, ERDY

Figure 4-3 External memory read with 1 wait setting and 1 wait cycle added



0 wait or 1 wait can be selected for memory access cycle. However, ERDY input is neglected in case of 0 wait setting. On resetting, this setting is initialized to 1 wait.

Figure 4-4 External memory write with 0 wait setting

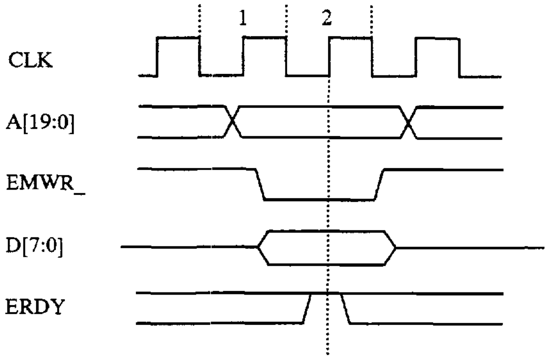
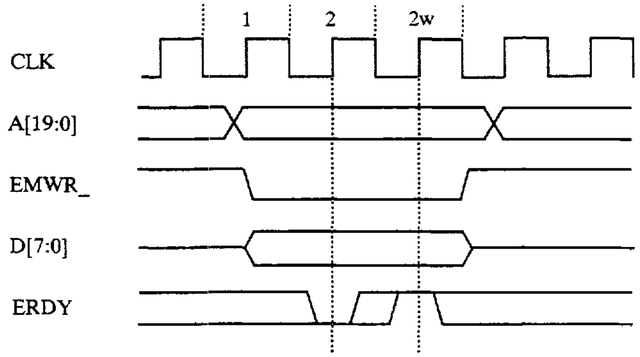
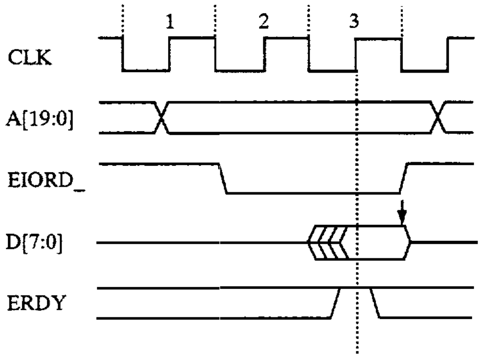


Figure 4-5 External memory write with 1 wait setting



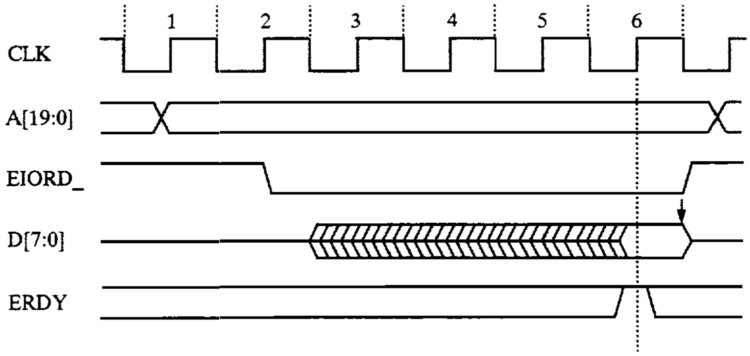
2w: The wait cycle requested by external wait request signal, ERDY

Figure 4-6 External memory write with 1 wait setting and 1 wait cycle added



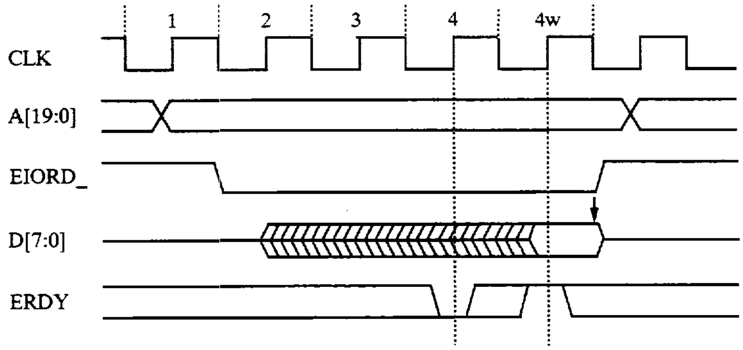
From 1 wait to 4 wait can be selected for external I/O access cycle. In case of 1 wait setting, external I/O devices are read in 3 clock cycle. In case of 2 or 3 waits setting, additional 1 or 2 wait cycles are inserted affer cycle 2 respectively. Regardless of SCR setting, internal I/O access is 0 wait. EIORD_ signal is kept "H" during internal I/O read cycles.

Figure 4-7 External I/O read with 1 wait setting



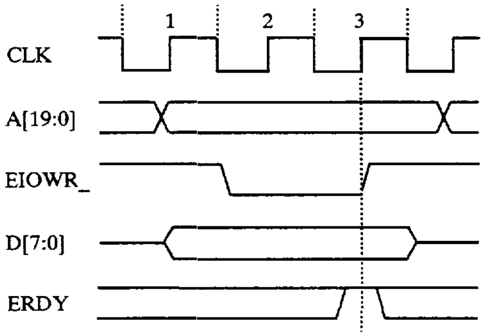
The I/O read cycle with 4 wait cycles is different from other I/O read cycle. Its falling edge of EIORD_ delayed with half clock cycle to other external I/O read cycles. Use this cycle when the external I/O devices with large setup time is connected.

Figure 4-8 External I/O read with 4 wait setting



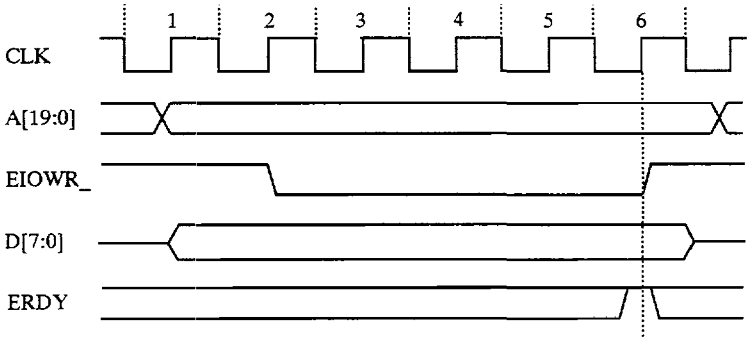
4w: The wait cycle requested by external wait request signal, ERDY

Figure 4-9 External I/O read with 2 wait setting and 1 wait cycle added



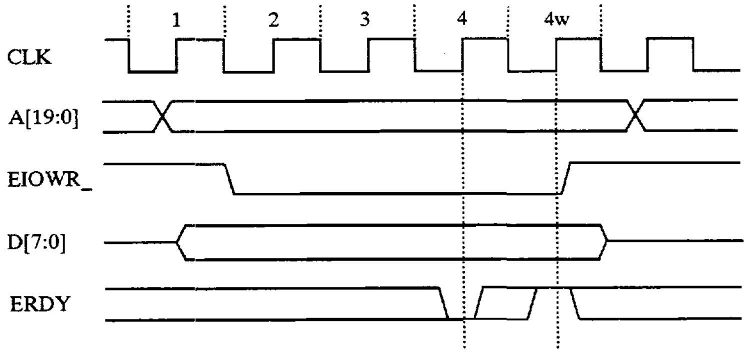
From 1 wait to 4 wait can be selected for external I/O access cycle. In case of 1 wait setting, external I/O devices are written in 3 clock cycle. In case of 2 or 3 waits setting, additional 1 or 2 wait cycles are inserted after cycle 2 respectively. Regardless of SCR setting, internal I/O access is 0 wait. EIOWR_ signal is kept "H" during internal I/O write cycles.

Figure 4-10 External I/O write with 1 wait setting



The I/O write cycle with 4 wait cycles is different from other I/O write cycle. Its falling edge of EIOWR_ delayed with half clock cycle to other external I/O write cycles. Use this cycle when the external I/O devices with large setup time is connected.

Figure 4-11 External I/O write with 4 wait setting



4w: The wait cycle requested by external wait request signal, ERDY

Figure 4-12 External I/O write with 2 wait setting and 1 wait cycle added

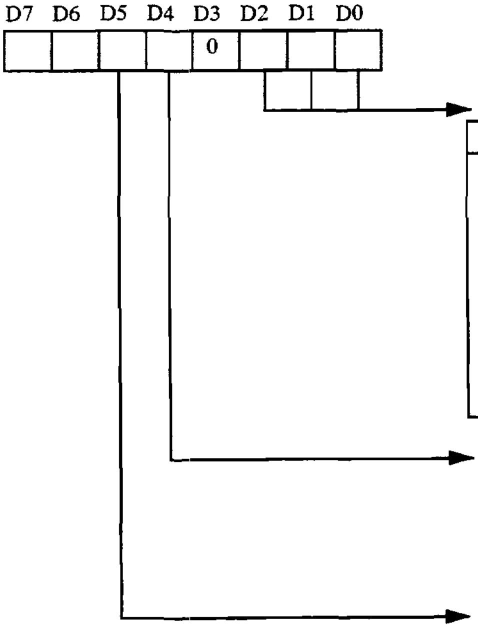
4.4 DRAM controller bus timings

A DRAM controller is integrated on KL5C80A16. The DRAM controller has a refresh counter, RAS/ CAS timing function and DRAM bank register. Large size DRAMs can be connected directly with this DRAM controller. The DRAM controller is controlled through SCR3 and SCR0. The address space for DRAMs is 80000H ~ BFFFFH. C0000H ~ FFFFFH can be used as SRAM area, or the upper most bank of DRAM banks can be mapped to C0000H ~ FFFFFH depending on D5 of SCR3.

The DRAM controller integrates a multiplexer that multiplexes DRAM row address and column address. Connect DRAM address signals as follows.

256K byte (no bank configration)	: Connect A16, A7~A0 to DRAMs
1M byte extended (4 banks)	: Connect A17,A16, A7~A0 to DRAMs.
4M byte extended (16 banks)	: Connect A18~A16, A7~ A0 to DRAMs.

SCR3(I/O address = 1EH)



Refresh Rate

D2	D1	D0	rifles interval
0	0	0	every 128 clock
0	0	1	every 88 clock
0	1	0	every 64 clock
0	1	1	every 44 clock
1	0	0	every 32 clock
1	0	1	every 22 clock
1	1	0	every 16 clock
1	1	1	every 11 clock

When D5 = 1, use C0000H ~ FFFFFH as
0: SRAM area
1: the upper most bank of DRAM banks
(When D5 = 0, this bit is meaningless.)

Use 80000H ~ BFFFFH as
0: SRAM area.
Pin 96 works as P13.
Pin 97 works as P12.
Pin 98 works as P11.
1: DRAM area.
Pin 96 works as UASEL_.
Pin 97 works as CAS_.
Pin 98 works as RAS_.
(UASEL_ is used only in debugging purpose.)

SCR0(I/O address = 1BH)

The address space that can be used for DRAMs is limited to 80000H ~ BFFFFH. However, DRAM area bank mechanism that controlled by SCR0 supports up to 4 M byte of DRAMs.

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0				

DRAM bank number
When 80000H ~ BFFFFH is used as DRAM area, this register specifies the bank number.
Use necessary bits from LSB (D<1:0> or D<3:0>), and specify the bank numbers from 0 up to 15. For example,
1 M byte extended: Use two chips of 4 M byte DRAM with four bit I/Os. Use two bit(D<1:0>) and control four banks.
4 M byte extended: Use two chips of 16 M byte DRAM with four bit I/Os. Use four bit(D<3:0>) and control sixteen banks.

Figure 4-13 shows DRAM refresh timing. The DRAM controller supports CAS before RAS refresh. The refresh cycle goes parallel with bus cycles other than DRAM read/write (read/write to memories other than DRAM, I/O read/write, CPU idle cycle). The states of address bus, EMRD_, EMWR_, EIORD_, EIOWR_, D<7:0>, and ERDY vary and depend on the bus cycle that goes in parallel with the refresh cycle. In case that the DRAM refresh cycle and DRAM read/write cycle occur simultaneously, the DRAM refresh cycle goes and CPU goes into a waiting state. In case that the DRAM refresh is requested while DRAM read/write cycle is going, the DRAM refresh cycle starts after the on-going DRAM read/write cycle ends. On the other hand, in case that the DRAM read/write cycle is requested while the DRAM refresh is going, the CPU waits until the on-going DRAM refresh cycle ends.

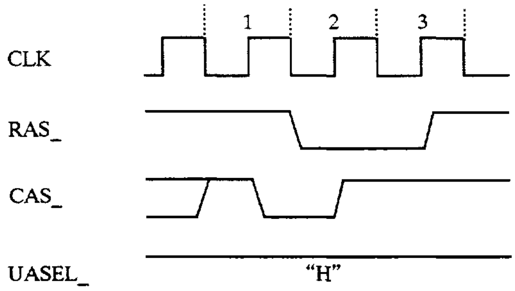
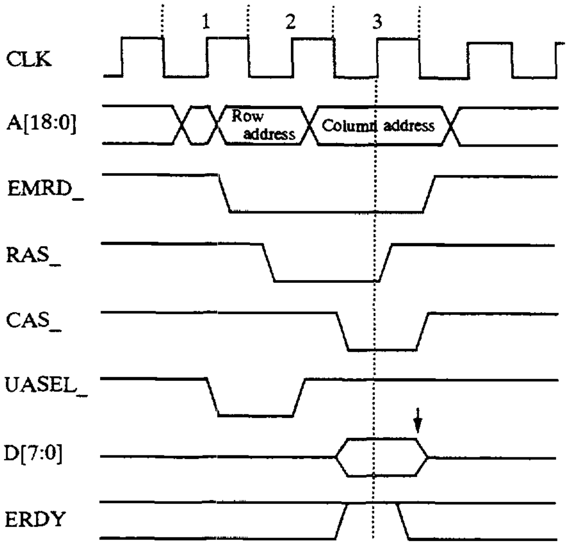
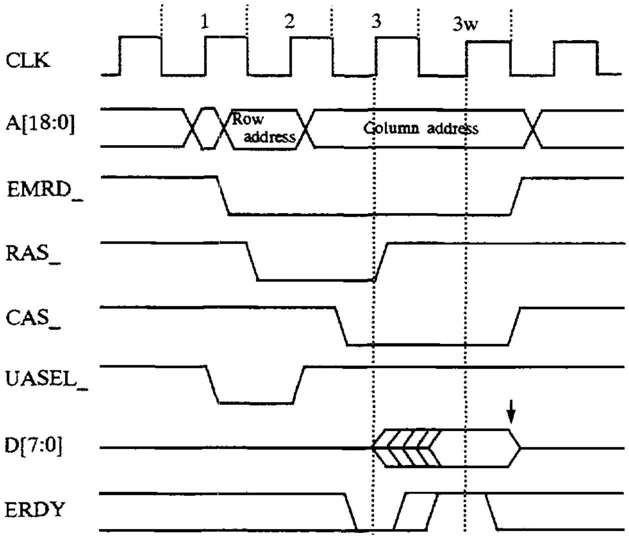


Figure 4-13 DRAM refresh



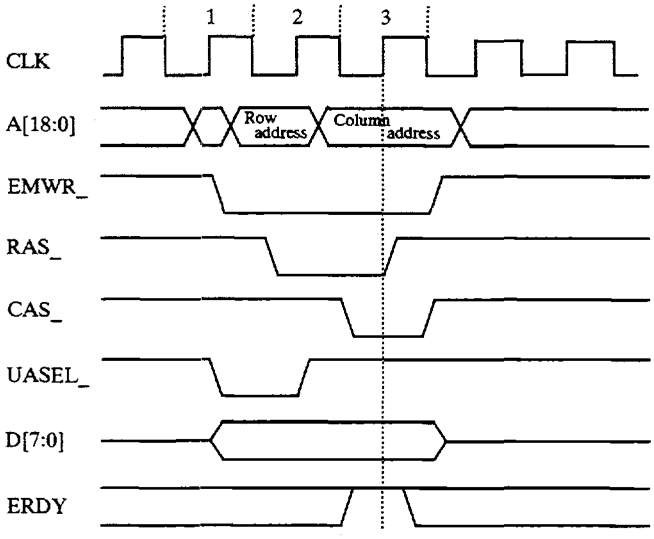
Regardless of SCR setting (memory access wait setting), DRAM read cycle is always 3 clock bus cycle. In case that DRAM access time is not adequate, add wait cycles by ERDY input. ERDY input is always effective regardless of SCR setting.

Figure 4-14 DRAM read cycle with no wait added



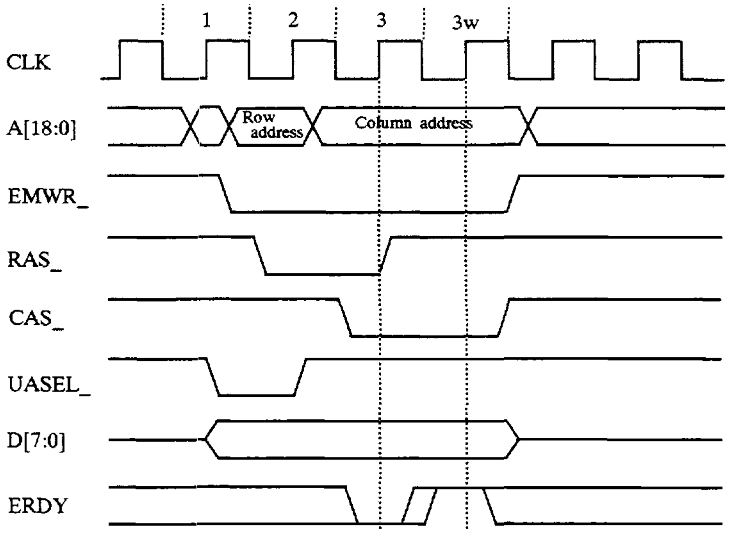
3w: The wait cycle requested by external wait request signal, ERDY

Figure 4-15 DRAM read cycle with 1 wait added



Regardless of SCR setting (memory access wait setting), DRAM write cycle is always 3 clock bus cycle. In case that DRAM access time is not adequate, add wait cycles by ERDY input. ERDY input is always effective regardless of SCR setting.

Figure 4-16 DRAM write cycle with no wait added



3w: The wait cycle requested by external wait request signal, ERDY

Figure 4-17 DRAM write cycle with 1 wait added

4.5 DMA controller external bus timings

The external bus interface unit converts DMA transfer cycles by the integrated DMA controller into external bus cycles. The memory to memory transfer cycle is converted in the same way as two independent memory accesses. Refer to memory access or DRAM controller timings about external bus cycles of the memory to memory transfer. See chapter 7 for more details of the DMA controller.

The Basic timings of DMA controller external bus cycles

The integrated DMA controller outputs I/O address on A19 ~ A0 in cycle 1 of DMA transfer. In cycle 2, it outputs memory address, and asserts EMRD_ and EIOWR_, or EMWR_ and EIORD_. (In case of DMA transfer between the internal I/O and memory, EIORD_ and EIOWR_ don't go active.) Then the data is transferred from the memory to the I/O, or the I/O to the memory directly. If the I/O subject to DMA transfer is an external device, there are two ways to select the device. The one is to latch the address output in cycle 1 with an external circuit at the falling edge of PALAT_ signal, and keep and provide it for external I/O devices. Another way is to select the chip select inputs of the external I/O devices directly using DMA request acknowledge signals (DACK0_, DACK1_ outputs) with some external circuits.

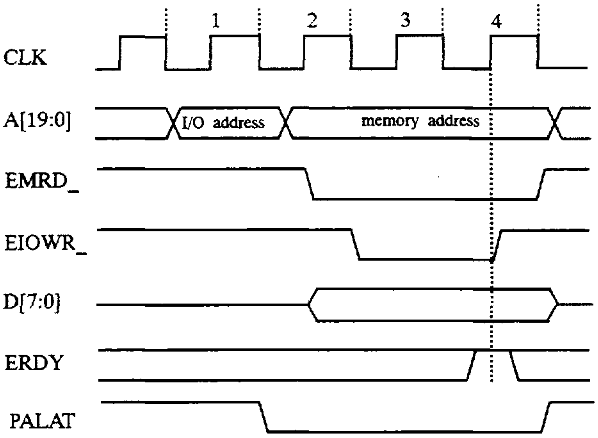
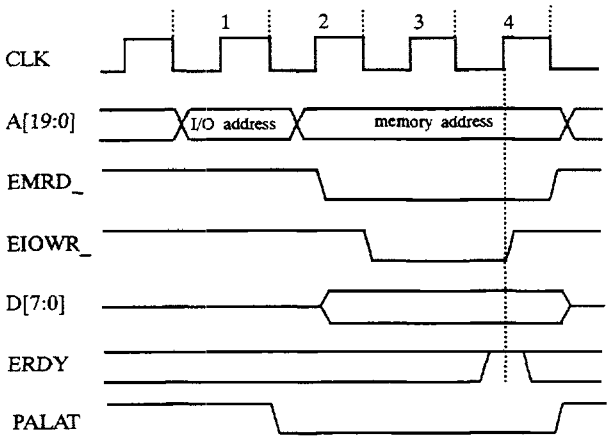


Figure 4-18 Example of DMA transfer (memory to external I/O, memory 0 wait access I/O 1 wait access setting)



The clock cycles needed for DMA transfer from memories except DRAM to external I/Os depend on wait settings of DMA transfer source and destination. In case of DMA transfer with source memory access 0 wait and destination I/O 1 wait, DMA transfer cycle is 4 clock cycle as shown in Figure 4-19. In case of DMA transfer with source memory access 1 wait, a wait cycle is added after cycle 2. In case of DMA transfer with destination I/O access 2, 3 or 4 waits, 1, 2 or 3 wait cycles are added after cycle 3. The Figure 4-20 shows the example of with source memory access 1 wait and destination I/O 2 wait.

Figure 4-19 DMA transfer (memory to external I/O, memory 0 wait access, I/O 1 wait access setting)

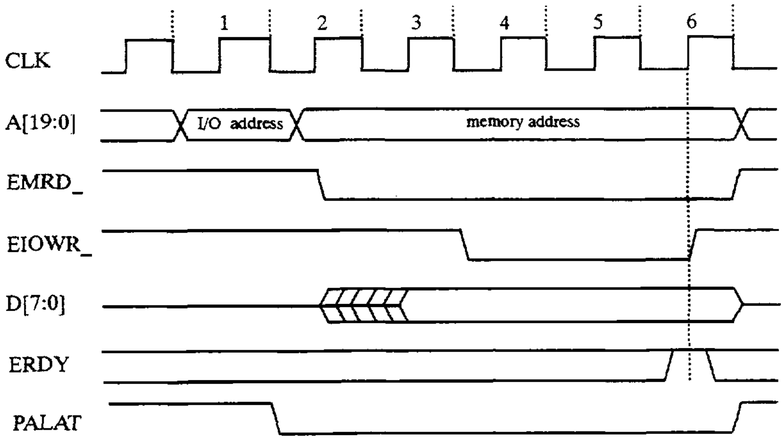


Figure 4-20 DMA transfer (memory to external I/O, memory 1 wait access, I/O 2 wait access setting)

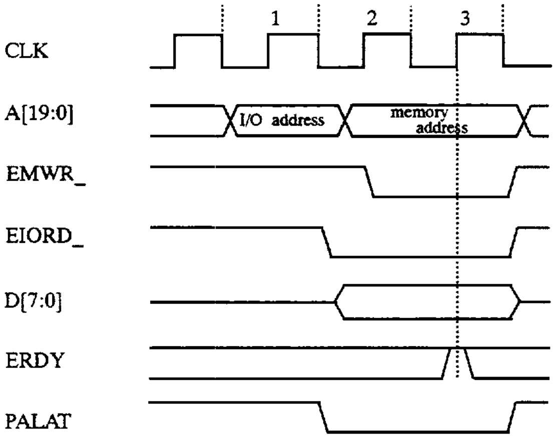


Figure 4-21 DMA transfer (external I/O to memory, memory 0 wait access, I/O 1 wait access setting)

The clock cycles needed for DMA transfer from external I/Os to memories except DRAM depend on wait settings of DMA transfer source and destination. In case of DMA transfer with source I/O 1 wait and destination memory access 0 wait, DMA transfer cycle is 3 clock cycle as shown in Figure 4-21. In case of DMA transfer with source I/O access 2, 3 or 4 waits, 1, 2 or 3 wait cycles are added after cycle 2. In case of DMA transfer with destination memory access 1 wait, a wait cycle is added after cycle 3. The Figure 4-22 shows the example of with source I/O 2 wait and destination memory access 1 wait.

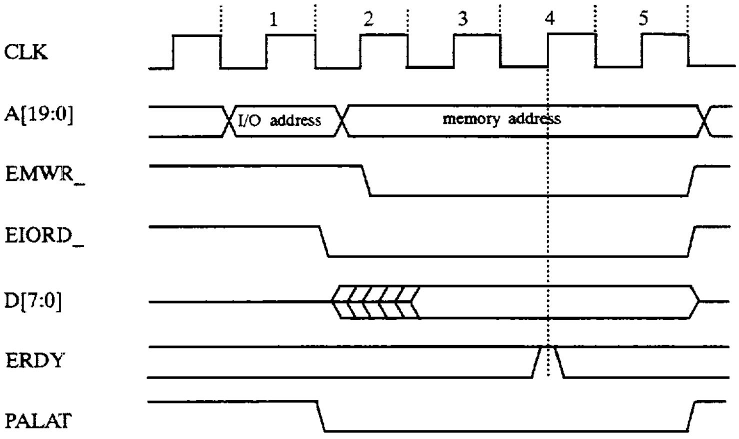
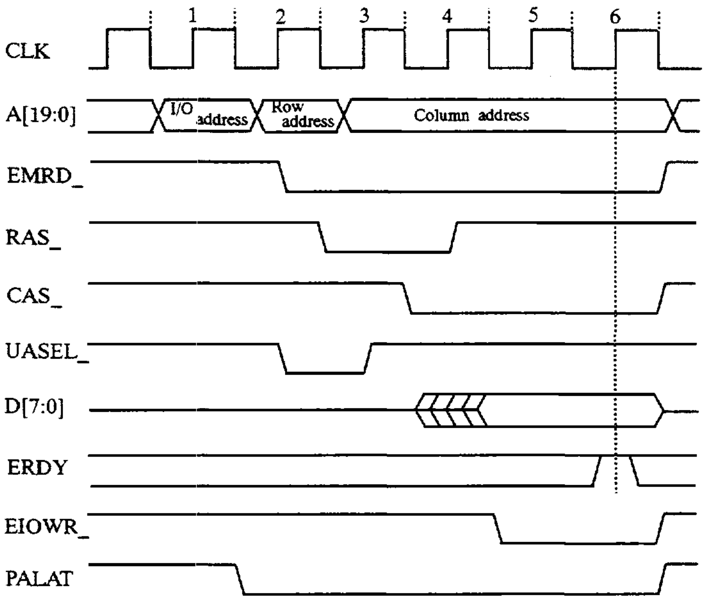
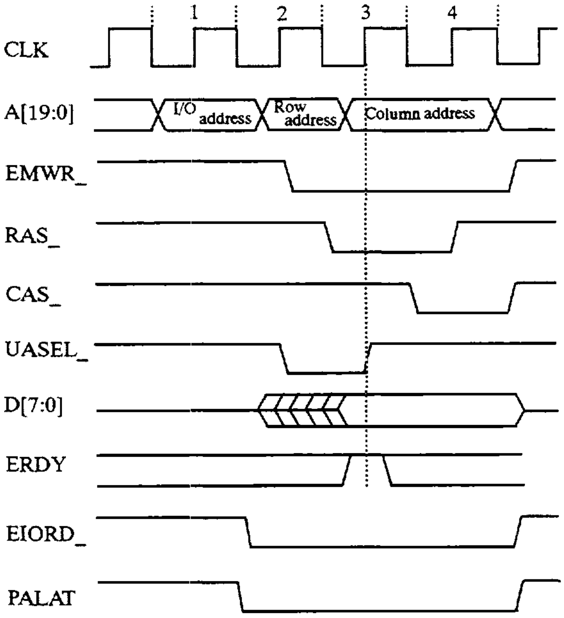


Figure 4-22 DMA transfer (external I/O to memory, memory 1 wait access, I/O 2 wait access setting)



The clock cycles needed for DMA transfer from DRAM to external I/Os depend on wait settings of DMA transfer destination I/O wait setting on SCR4. In case of DMA transfer with destination I/O 1 wait, DMA transfer cycle is 6 clock cycle as shown in Figure 4-23. In case of DMA transfer with destination I/O access 2, 3 or 4 waits, 1, 2 or 3 wait cycles are added after cycle 5. Wait setting for memory access doesn't affect the clock cycles needed for DMA transfer from DRAM.

Figure 4-23 DMA transfer (DRAM to external I/O, I/O 1 wait access setting without wait cycles by ERDY)



The clock cycles needed for DMA transfer from external I/Os to DRAM depend on wait settings of DMA transfer source I/O wait setting on SCR4. In case of DMA transfer with source I/O 1 wait, DMA transfer cycle is 4 clock cycle as shown Figure 4-24. In case of DMA transfer with source I/O access 2, 3 or 4 waits, 1, 2 or 3 wait cycles are added after cycle 3. Wait setting for memory access doesn't affect the clock cycles needed for DMA transfer from DRAM.

Figure 4-24 DMA transfer (external I/O to DRAM, I/O 1 wait access setting without wait cycles by ERDY)

During DMA transfer between internal I/O and memories, external I/O read/write signals (EIORD_, EIOWR_) are kept "H", and only memory read/write signals are asserted. Then I/O access cycle doesn't appear, and only memory access cycle seems to appear for external devices. Figure 4-25 and 4-26 show the examples of 0 wait memory read and write. The memory access cycle which appears on each case depends on wait setting of memory access and types of memories.

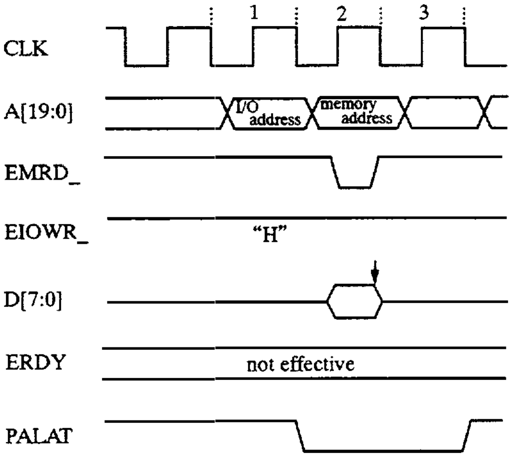


Figure 4-25 DMA transfer (memory to internal I/O, memory access 0 wait setting)

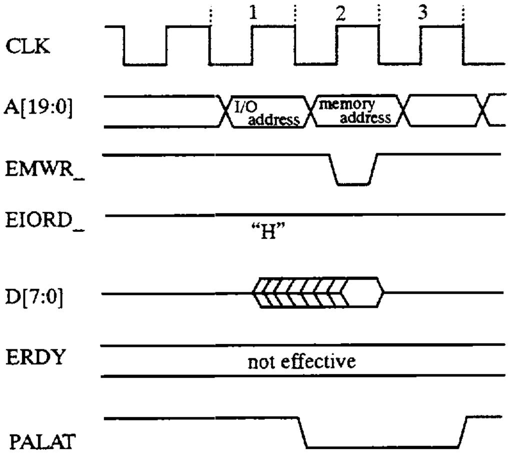


Figure 4-26 DMA transfer (internal I/O to memory, memory access 0 wait setting)

5. KC82 CPU

5.1 General description

KC 82 is the CPU core which is object compatible with Zilog's Z80 CPU with MMU that extends the address space up to 1M byte. But its internal circuit is totally different. KC82 has RISC like architecture, 16 bit internal data path and synchronous bus, and offers higher performance than typical 16 bit CPU.

Feature

KC82 has the following features:

1)KC82 executes instructions 4 times faster than Zilog Z80 CPU at the same clock rate.

ex : instruction	Z80	KC82
LD r, r'	4 clocks	1 clock
ADD HL, ss	11 clocks	1 clock

2)With interface macro cell, it can be used with the Z80 peripheral and ordinary RAMs.

3)158 instructions and fully compatible with the Z80 CPU at object code level.

5.2 Block Diagram

The following figures 5-1 has the block diagram of the KC82.

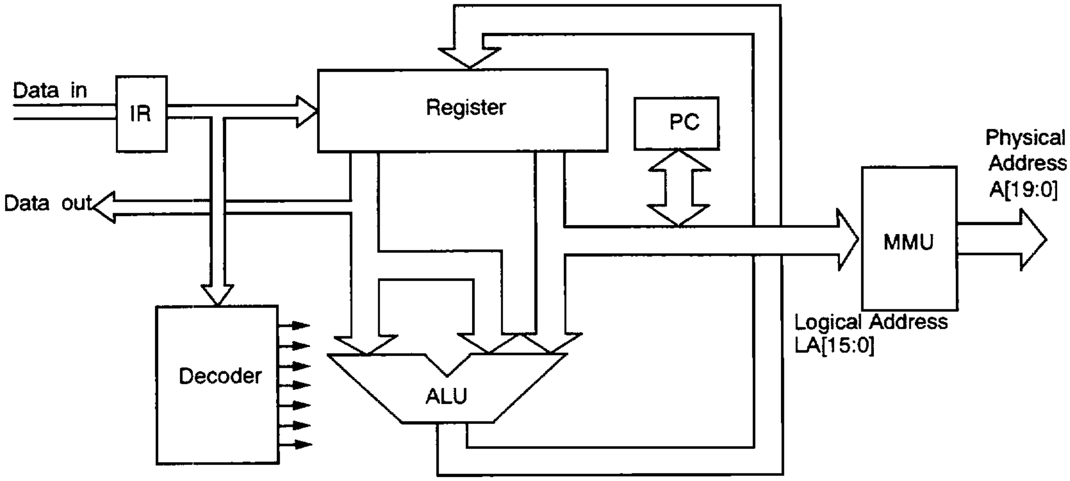


Figure 5-1. KC82 Block Diagram

5.3 CPU internal pins description

signal name	I/O	description
MRD_	O	Memory read. Active Low. This signal goes active while reading data from memory.
MWR_	O	Memory write. Active Low. This signal goes active while writing data to memory.
IORD_	O	I/O read. Active Low. This signal goes active for minimum of two clock cycles while reading data from I/O device.
IOWR_	O	I/O write. Active Low. This signal goes active for minimum of two clock cycles while writing data to I/O device.
WAIT_	I	Wait input. Active Low. This signal is to notify the CPU that the peripherals or memory is not ready for data transfer. While this signal stays active, the CPU is placed in the wait state. This input is connected to the external pin, ERDY through the wait state controller in the external bus interface unit.
INT_	I	Maskable interrupt input. Active Low. This input accepts the interrupt from peripherals. If the interrupt enable flag of the CPU is set, and BREQ_ is inactive, the CPU completes the execution of the current instruction, and starts interrupt service. This input is connected to the INT_ output of the interrupt controller.
IACK_	O	Interrupt Acknowledge output. Active Low. This signal indicates that the CPU acknowledges the interrupt, requesting interrupt vector or instruction from the I/O device. It stays active at minimum of two clock cycles. This output is connected to the interrupt controller. The CPU reads the interrupt vector synchronously with this signal.
EOI_	O	End of interrupt signal output. Active Low. This signal goes active at refetch of RETI instruction (code ED 4D). This output is connected to the interrupt controller.
LA [15:0]	O	Logical Address output. It outputs the physical address A[19:0] expanded by the MMU in KC82.

5.4 Register architecture

5.4.1 Special purpose registers

Program Counter (PC)

Program Counter holds the address of the next instruction. The next instruction is fetched from memory address Program Counter indicates.

Stack Pointer (SP)

Stack Pointer holds the current top address of stack area in RAM.

Index registers (IX, IY)

Index registers hold base address for indexed addressing. There are two index registers. The one is IX, and the other IY.

Interrupt page address register (I)

Interrupt page address register holds indirect upper 8 bit address for indirect jump in Mode 2.

Memory refresh register (R)

This register automatically increments by one on instruction fetch. Bit D7 of this register will not change by increments on D6-D0.

Accumulator (A, A'), and Flag register (F, F')

There are two 8 bit accumulators (A, A') and two flag registers. Accumulator holds the result of arithmetic and logical operation. Flag register holds status of 8 bit or 16 bit operation. The instruction, EX AF, AF' exchanges A, F and A', F'.

5.4.2 General purpose registers

There are two sets of general purpose registers. Each of them can be used as an 8 bit register independently (B, C, D, E, H, L or B', C', D', E', H', L'). Paired registers (BC, DE, HL, BC', DE' and HL') can be used as a 16 bit register. The instruction, EXX exchanges B,C, D, E, H, L and B',C', D', E', H', L'.

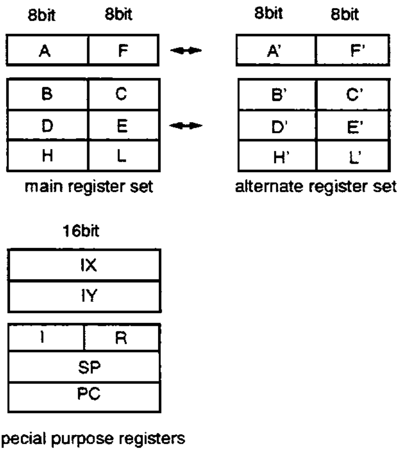


Figure 5-3. KC82 register architecture.

5.5 Flags

Bit assignment of the Flag register.

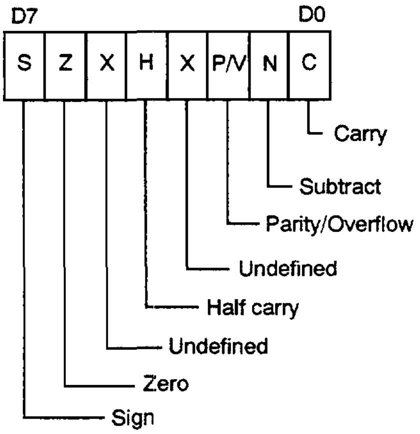


Figure 5-4. Flag register

The function of each bit in the Flag register is defined as below.

Carry flag (C)

Carry flag is set by carry produced by Accumulator from the MSB at execution of add instructions, subtract instructions, shift and rotate instructions, etc.

Zero flag (Z)

Zero flag is set if the operation result is zero, when add instructions, subtract instructions, logical instructions including INC, DEC, DAA instructions. This also occurs on block I/O instructions, rotate and shift instructions, string search instructions, bit test instructions and data detection of block search instructions.

Sign flag (S)

Sign flag is set when the result of sign number operations is negative.

Parity/Overflow flag (P/V)

Parity/Overflow flag has two functions. It indicates the parity on logical operations, and also the overflow on arithmetic operations. The overflow means that the content of Accumulator is less than -128 in 2's complement, or larger than 127. It is also set by block search instructions, block I/O instructions, LD A, R instruction and LD A, I instruction.

Half carry flag (H)

Half carry flag is set when the carry or borrow from lower 4 bits is produced.

Subtract flag (N)

Subtract flag is set on subtract instructions. KC82 checks if the previous instruction is addition or subtraction with this flag on DAA instruction.

5.6 Functional description and timing

In this section, functional description and timing are described. Note that the address bus in the timing diagrams is the logical address, LA[15:0], to describe easily. The logical address LA[15:0] is converted to the physical address by MMU.
All bus cycles are described using the internal bus. See the chapter 4 for more information of the external bus.

5.6.1 Basic operation (Instruction cycle)

Basic operation of the KC82 can be divided into the following 5 cycles.

1)Opcode fetch cycle - Fetches opcode of an instruction from memory.
2)Memory read cycle - reads data from memory
3)Memory write cycle - writes data to memory.
4)I/O read cycle - reads data from I/O device
5)I/O write cycle - writes data to I/O device

Follows are the description of the each machine cycles.

Opcode fetch cycle

As shown figure 5-5, KC82 fetches one byte of an instruction in a clock cycle unless there is a wait. During this cycle, MRD_ and M1_ go active. The difference between memory read cycle is whether if M1_ goes active or not. This cycle is minimum of one clock cycle and appears at the end of the instruction execution sequence.

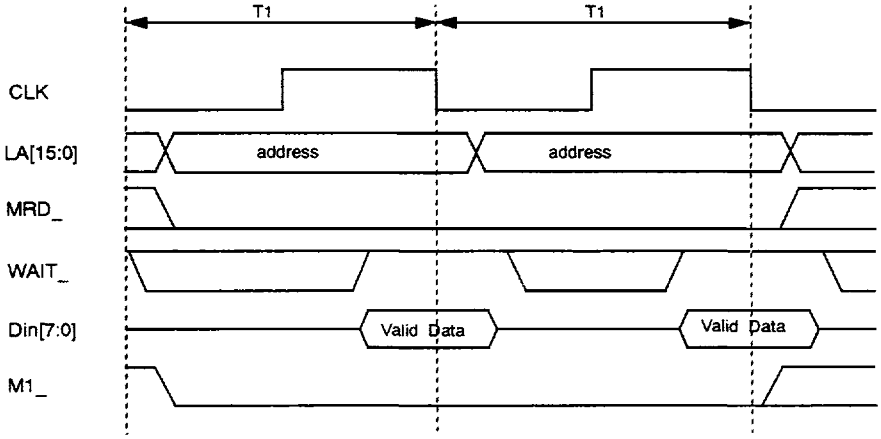


Figure 5-5. Opcode fetch cycle (Internal cycle)
(0 wait state, two contiguous reads.)

Internal bus memory read cycle

The difference between opcode fetch cycle is whether if M1_ goes active or not. This cycle is minimum of 1 clock cycle.

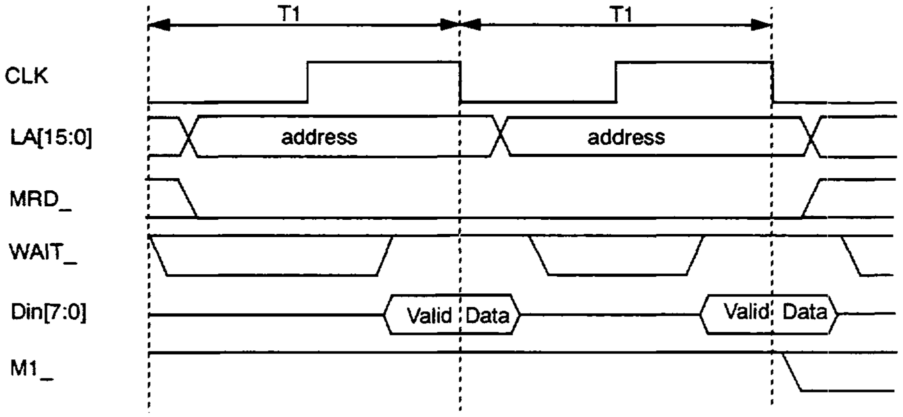


Figure 5-6. Memory read cycle (0 wait)
(0 wait state, two contiguous reads.)

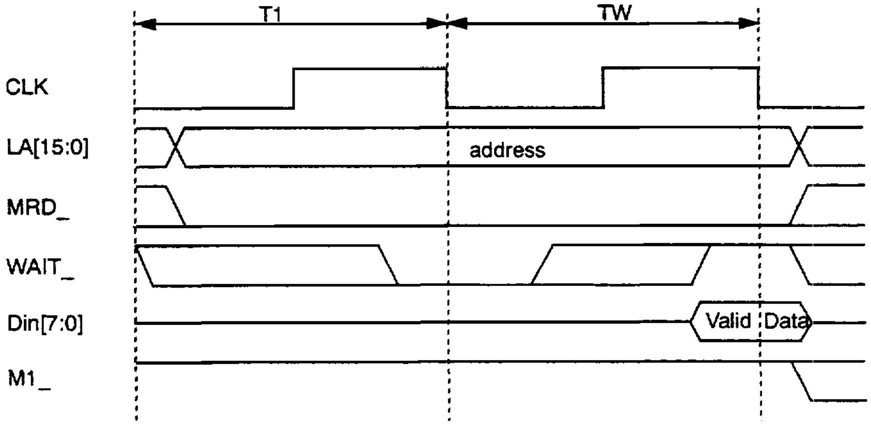


Figure 5-7. Memory read cycle (1 wait)
(One read with 1 wait state)

Internal bus memory write cycle

This cycle is minimum of 1 clock cycle.

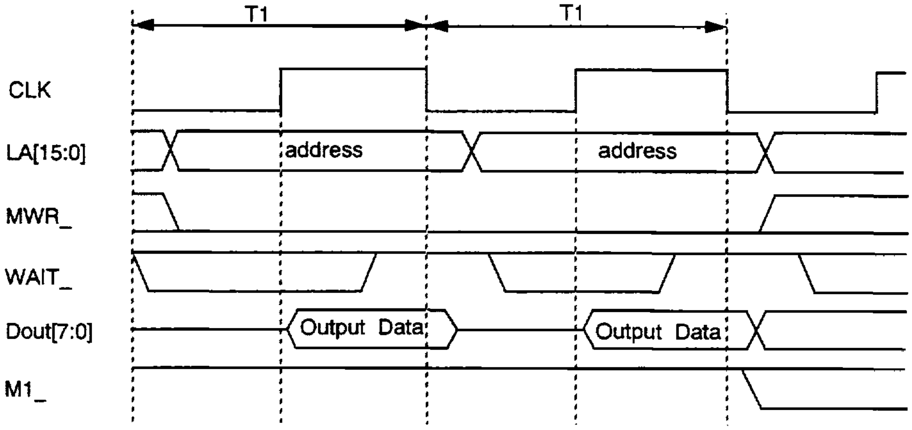


Figure 5-8. Memory write cycle (0 wait)
(0 wait state, two contiguous writes.)

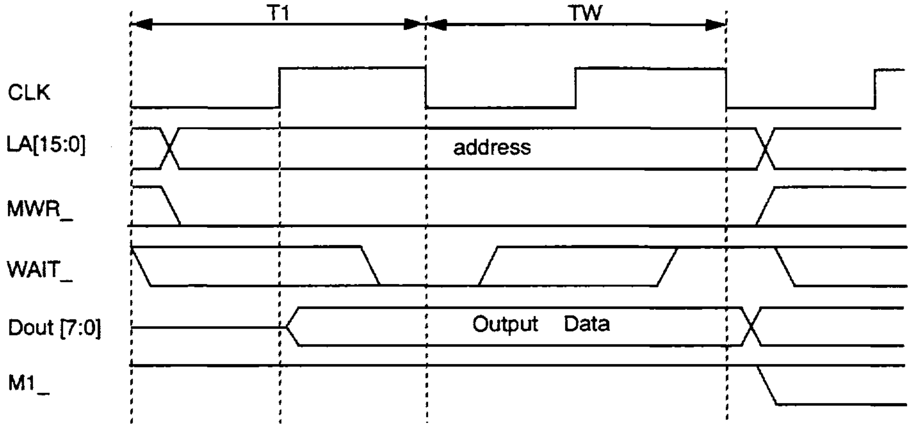


Figure 5-9. Memory write cycle (1 wait)
(One write with 1 wait state)

Internal bus I/O read cycle

This cycle is minimum of 2 clock cycles.

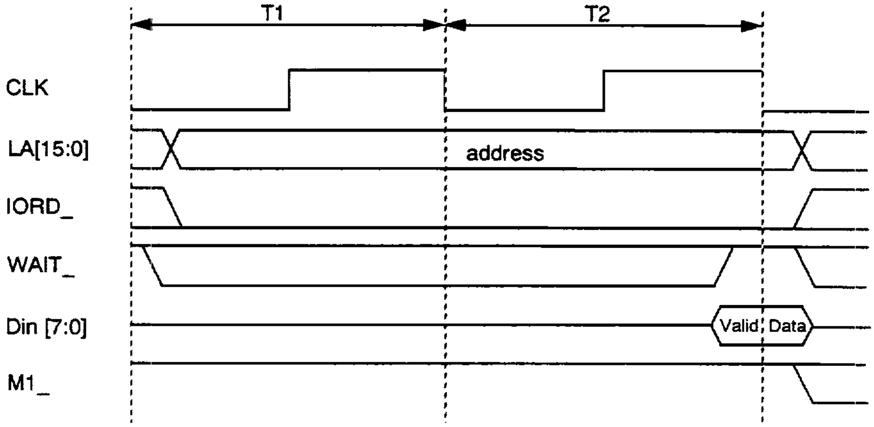


Figure 5-10. I/O read cycle (0 wait)

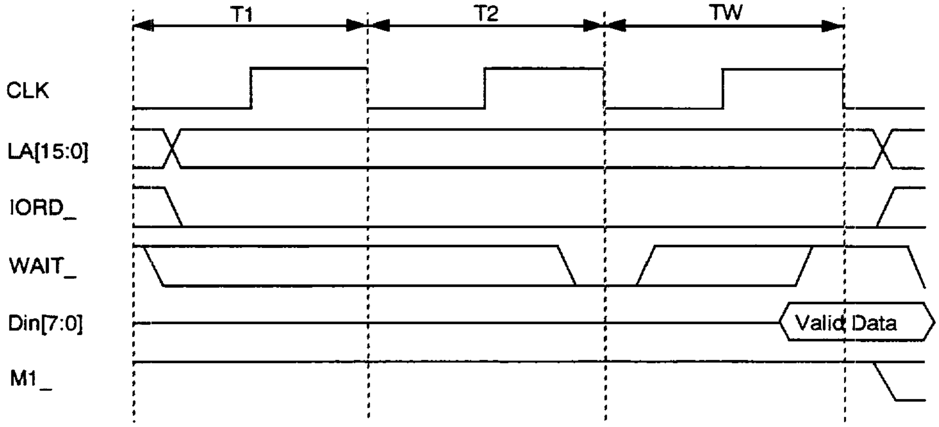


Figure 5-11. I/O read cycle (1 wait)

Internal bus I/O write cycle

This cycle is minimum of 2 clock cycles.

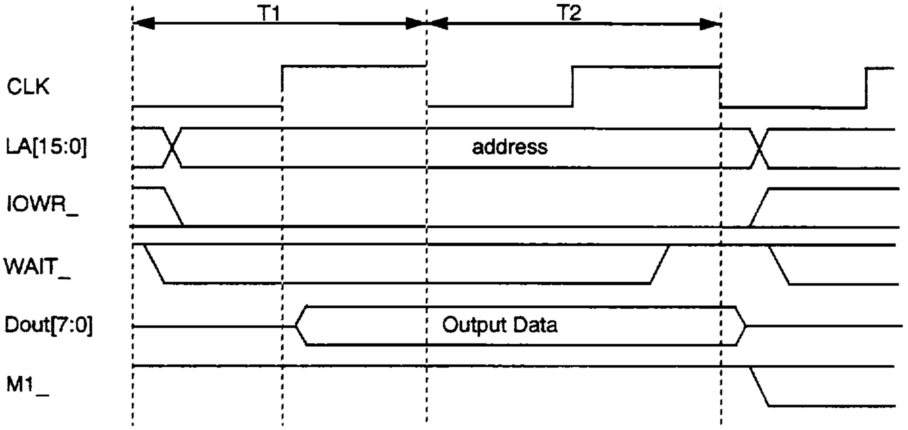


Figure 5-12. I/O write cycle (0 wait)

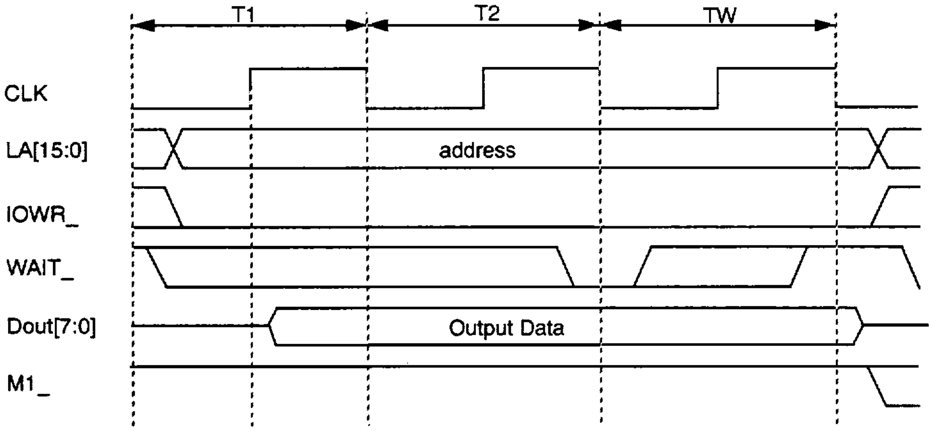


Figure 5-13. I/O write cycle (1 wait)

Instruction Prefetch cycle

The KC82 has the prefetch cycle at the end of instruction execution cycle all the time. Figure5-15 shows that KC82 executes the instruction sequence indicated in Figure5-14 as an example.

◁—▷ (referred as "arrow" hereinafter) in the Figure 5-15 denotes the prefetch cycle. The arrow ① part prefetches the instruction at n+3 (ADD A,D), the arrow ② part is for prefetching instruction at n+4 (opcode of JP instruction, C3h), the arrow ③ part is for prefetching 77h at 1000h (LD [HL], A instruction). As shown, these cycles appear at the end of execution of current instruction.

address	mnemonic	code
n	LD A, [1234h]	3A 34 12
n+3	ADD A, D	82
n+4	JP 1000h	C3 00 10
·	·	
·	·	
1000h	LD [HL], A	77
·	·	
1234h		5A

Figure 5-14. Source code for Figure5-15

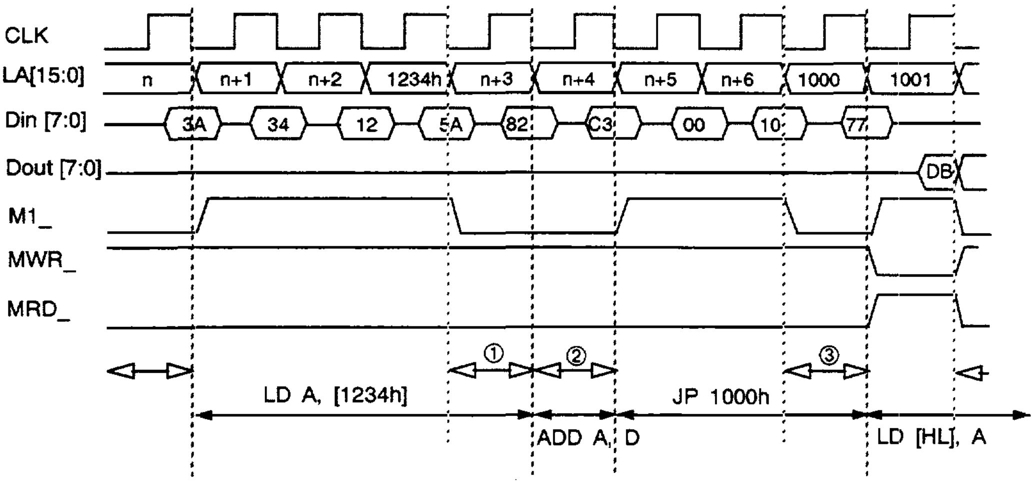


Figure 5-15. Prefetch cycle (◁—▷ denotes the prefetch cycle. Assumes memory location 1000h has 77h, memory location 1234h has 5Ah, and D register has 81H.)

Special case for prefetch. (Discard of the prefetched instruction)

For example, under normal condition, the prefetched instruction, at address 1000h, data 77h at ③ in Figure 5-15 on execution of the instruction, JP 1000h is held and execute correctly. However, on interrupt the prefetched instruction will be discarded and refetch on return from an interrupt service routine. On interrupt recognition, PC is decremented by one before saved onto stack. On RETI or RETN instruction the PC will be loaded with the address of the discarded instruction. Figure 5-19 has the timing diagram for the case of receiving interrupt in mode 1 when executing LDIR instruction. In the figure, the data "EDh" prefetched at ⑥, is discarded and then go to interrupt service. On return, the CPU resume from the prefetch of this "EDh". For the bus request cycle, the data "EDh" prefetched at ⑥ prior to the bus request will be kept, and on bus release the CPU starts to prefetch from the next address to the "EDh" as shown in the Figure 5-18.

5.6.2 Bus release (Bus request/acknowledge cycle)

Under normal operation, the CPU holds the control of address bus, control bus (MRD_, MWR_, IORD_ and IOWR_). However, if there is an external bus request (BREQ_="L"), address bus goes to high-impedance state, all the interrupts are disabled, BACK_ turns to "L", and the CPU releases the bus control to the external device. By using this feature, data transfer without CPU intervention can be done. Figure 5-17 shows basic timing for the bus request cycle. This cycle continues while BREQ_ stays "L".

Note that bus request is accepted only at the end of instructions except the instructions which have iteration, such as block transfer instructions, block search instructions and iterating IN/OUT instructions. In a special case, for the block move instructions the acceptance of a bus request occurs on every iteration of an execution cycle, not the end of the instruction. As shown in the Figure 5-18, BREQ_ is asserted at point ④ during the execution, the bus is released at the last execution cycle of an instruction ⑤.

5.6.3 Interrupt and timing

The KC82 can handle the following two kinds of interrupts.

1)maskable interrupt on INT_

2)Non-maskable interrupt on NMI_

2) has the higher priority over 1). If both request made simultaneously, 2) will be accepted over 1).

Maskable interrupt

The EI instruction enables interrupts, and the DI instruction disables interrupts. The control of interrupt is implemented using two flip-flops (IFF1 and IFF2). Figure 5-16 has the state table for these flip-flops.

events	IFF1	IFF2	
reset	0	0	
DI instruction	0	0	
EI instruction	1	1	
INT accepted	0	0	
NMI accepted	0	-	
RETN instruction accepted	IFF2	-	IFF2 is copied into IFF1
LD A, I instruction	-	-	IFF2 is copied into P/V
LD A, R instruction	-	-	IFF2 is copied into P/V

Figure 5-16. State diagram for IFF1 and IFF2
("-" denotes remain unchanged.)

The maskable interrupt will be accepted if all of the following conditions are met:

1)Both IFF1 and IFF2 are set. (The EI instruction and RETN instructions change the status of these flags after the execution of the following instruction. So if there are EI instruction followed by DI instruction, the interrupt request will not be accepted.)

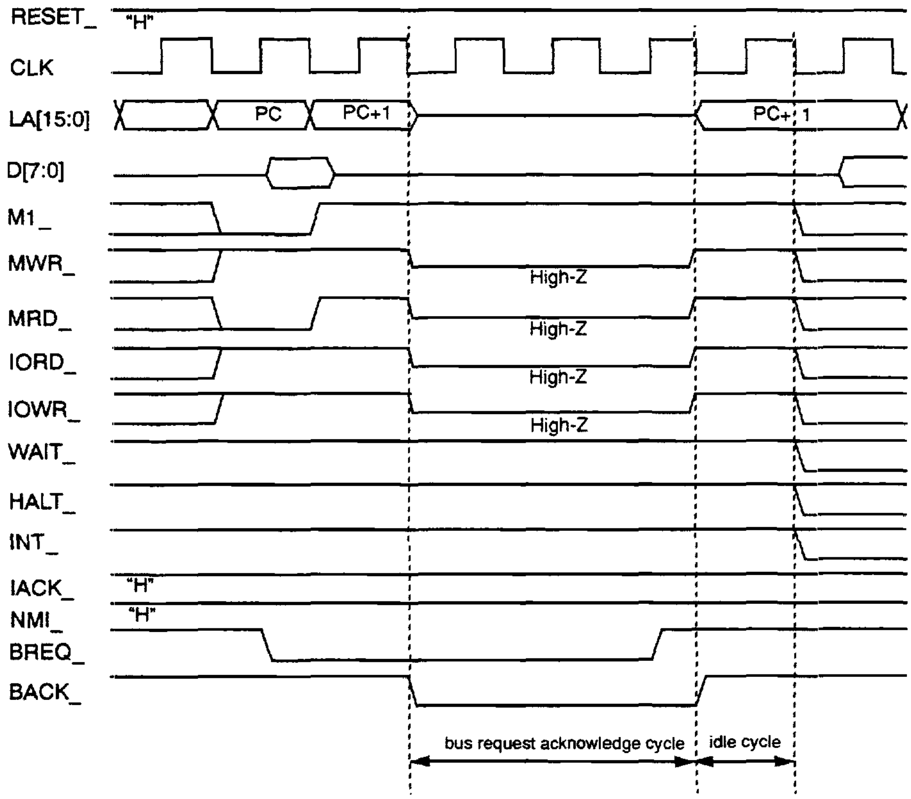


Figure 5-17. Bus request timing

2)BREQ_ is inactive ("H"). (When BREQ_ goes inactive, interrupt will be accepted after execution of an instruction.)

3)No NMI_ falling edge has been detected.

The KC82 has the dedicated signal, IACK_ for the external devices to acknowledged that the interrupt has been accepted, while the Z80 CPU signals using M1_ and IORQ_ combination for that purpose.

Maskable Interrupt modes

The KC82 has the following three different interrupt handling modes, and interrupt sequence is different by each modes. The following descriptions are the operations in each mode. Use the mode 2 usually.

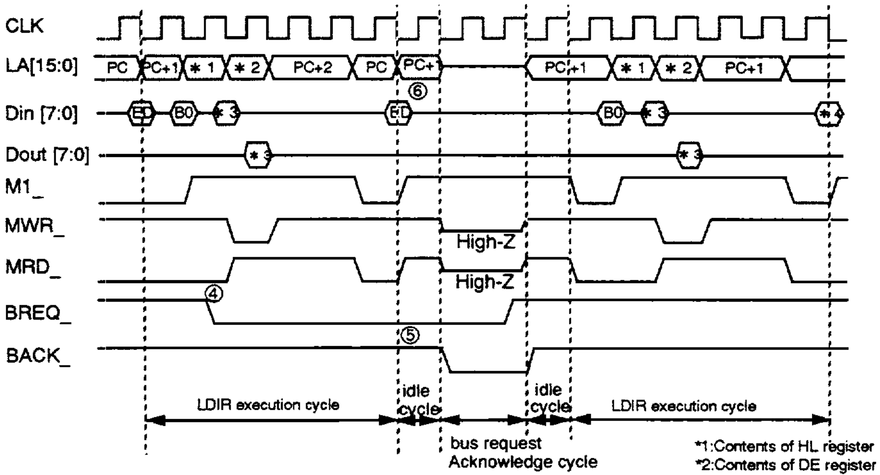
1)Mode 0

This is the default mode and set to this on reset automatically.

Also, executing "IM 0" instruction sets the CPU to this mode. In this mode, the CPU executes the instruction read during interrupt acknowledge cycle.

Usually the instruction to be used in this mode is RST instructions, or CALL instruction.

Figure 5-20 and 5-21 show each timing for RST and CALL instructions. For the all interrupt modes, if INT_ goes active during execution of a mode change instruction (IM 0~2), the new mode become effective right after these instructions.



*1:Contents of HL register
*2:Contents of DE register
*3:Data pointed by (HL)
*4:Next instruction

Figure 5-18. Bus request accepting timing

(The figure has the case for BREQ_ goes active at ④, accepting it at instruction boundary ⑤. Note that the data fetched at ⑥ will be kept.)

2)Mode 1

The "IM 1 instruction" sets the CPU in this mode. In this mode, the CPU saves the contents the PC (Program Counter) onto the stack, ignores the data read during the interrupt acknowledge cycle, and executes "RST 38H" instruction internally. Figure 5-22 has the timing for Mode 1 interrupt.

3)Mode 2

The "IM 2 instruction sets" the CPU to this mode. In this mode, as shown in the Figure 5-23-A, the 8-bit wide vector with 0 in LSB read during the interrupt acknowledge cycle as a lower byte, and the content of the I register as a upper byte form up a pointer which points an entry in a table of address for the interrupt service routines. After forming up the 16-bit address, the CPU fetches the table location and gets the start address, saves the return address onto the stack, then jumps to the service routine. Figure 5-23-B has the timing diagram for Mode 2 interrupt acknowledge cycle.

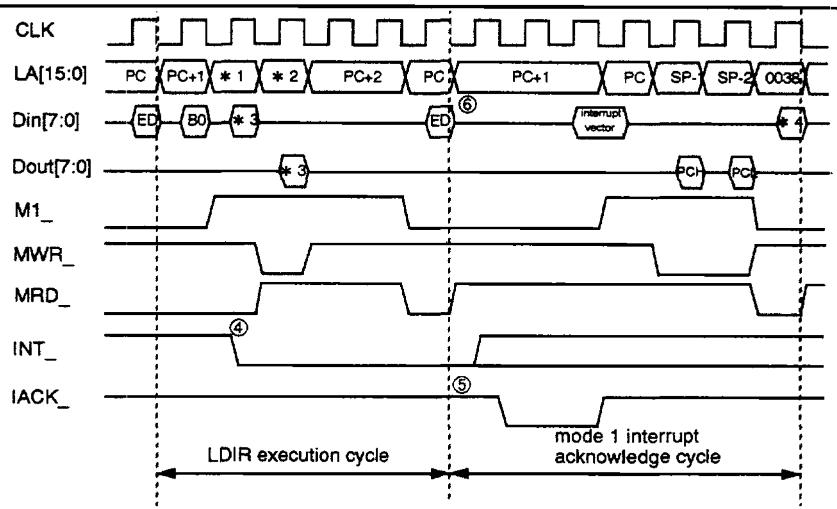
Timing on interrupt acceptance

The Figure 5-19 show the interrupt acknowledge timing. In this figure, an interrupt is accepted during block move instruction. ④ shows that when an interrupt request occurs, the interrupt will not be accepted until the prefetch cycle of the last instruction, ⑤. In this case, the data "EDh" fetched at ⑥ will not be kept by the CPU, and it will be refetched on the return from an interrupt.

Non-maskable interrupt

Non-maskable interrupt is an interrupt which can not be masked by software. The falling edge of the signal NMI_ is latched internally, and checked on the clock falling edge of the last cycle of every instruction. Non--maskable interrupt takes place if BREQ_ is inactive. If BREQ_ goes inactive, NMI_ will be accepted after execution of an instruction.

When a non-maskable interrupt is accepted, the CPU saves the contents of the PC onto the stack, and jumps to 0066h. Return from non-maskable service is done by "RETN" instruction. Figure 5-24 shows the timing for NMI_ acknowledge cycle.



*1:Contents of HL register.
*2:Contents of DE register.
*3:Data pointed by (HL).
*4:Contents of 0038H.

Figure 5-19 .Interrupt acknowledge cycle in mode 1

(The figure shows the case when the interrupt occurred at ④,accepting it at the instruction boundary ⑤. This case restarts from the fetch of ED at ⑥ on the return from an interrupt.)

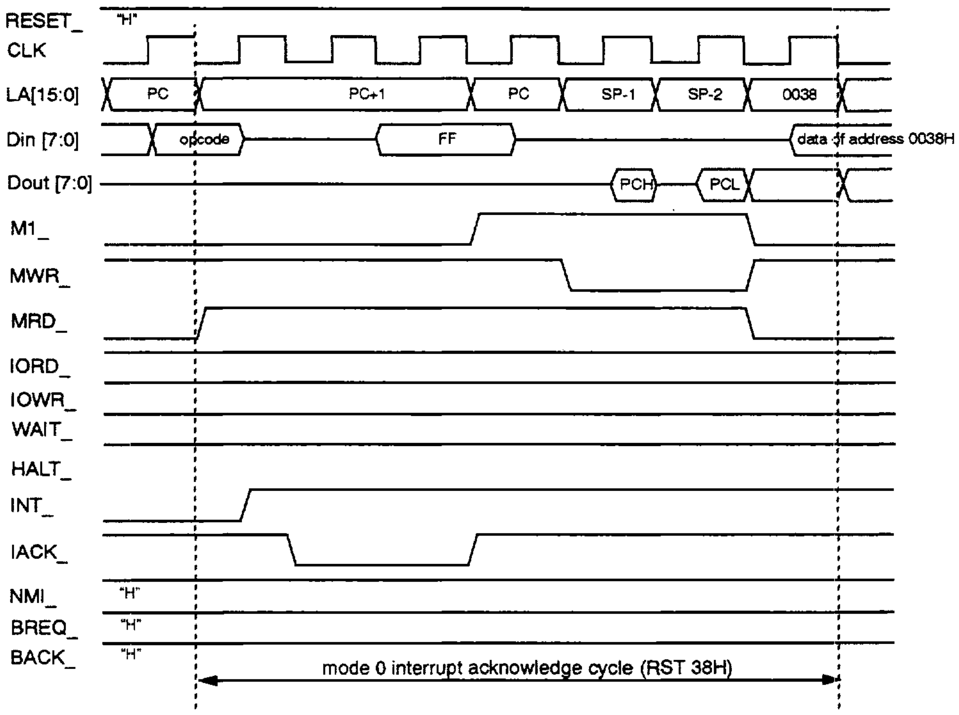


Figure 5-20. Mode 0 timing

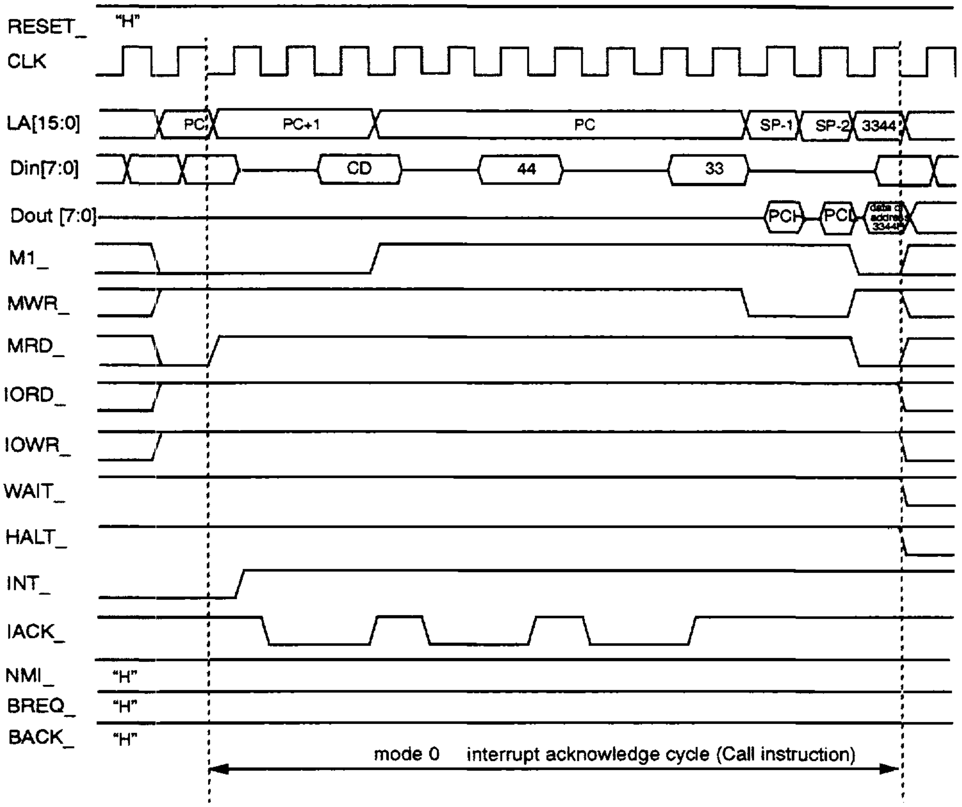


Figure 5-21. Mode 0 timing

Enable/disable of interrupts

Enable/disable of an interrupt is controlled by two flip-flops - IFF1 and IFF2. Refer to Figure 5-16 on the state diagram for these flip-flops. As shown, maskable interrupt will be accepted when both IFF1 and IFF2 set to 1. After reset and execution of DI instruction maskable interrupt is disabled, because both of these flip-flops are reset. An interrupt can be accepted while both IFF1 and IFF2 are set to 1 by execution of an EI instruction. The reason there are two flip-flops is to memorize the status of EI/DI condition with IFF2 on acceptance a Non-maskable interrupt. For example, if both IFF1 and IFF2 are '1' (maskable interrupt is enabled) on acceptance a non-maskable interrupt, in a non-maskable interrupt acknowledge cycle IFF1 is cleared to 0 and maskable interrupts are disabled. At this stage, IFF2 still holds "1" and the content of IFF2 is copied back into IFF1 on RETN instruction so that maskable interrupts could be enabled automatically again.

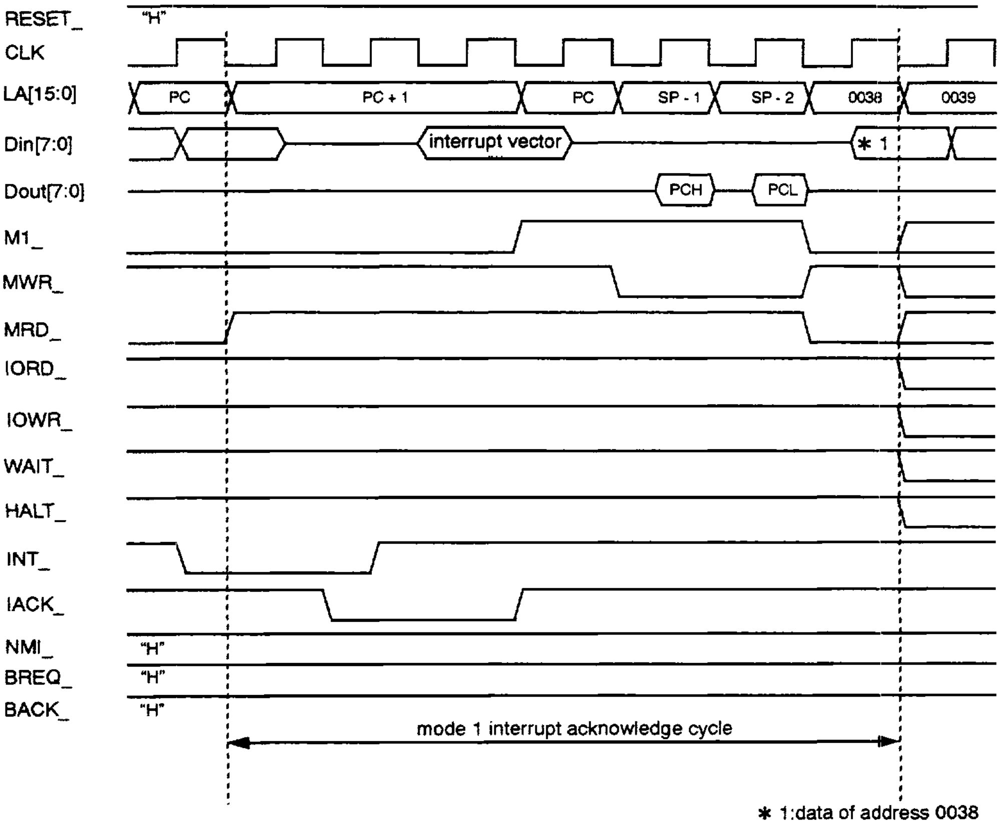


Figure 5-22. Mode 1 timing

5.6.4 Halt enter and exit

By executing the HALT instruction, the CPU enters into HALT cycle. In this cycle, the CPU continues to execute NOP instruction internally. The CPU exits from this cycle by either reset, or interrupts (Non-maskable interrupt or interrupt with IFF set). Figure 5-25 shows timing for the case of acceptance an interrupt in mode 2 during HALT cycle. Then the CPU exits from the HALT cycle. HALT cycle made up with two separate bus cycles; the first cycle is idle cycle and the second cycle is opcode fetch cycle. The address lines holds the next address to the HALT instruction during this cycle. The instruction fetched during the second opcode fetch cycle will not be read by the KC82. The INT_ input is sampled on the falling CLK edge of the second cycle, and forces HALT_ back to "H".

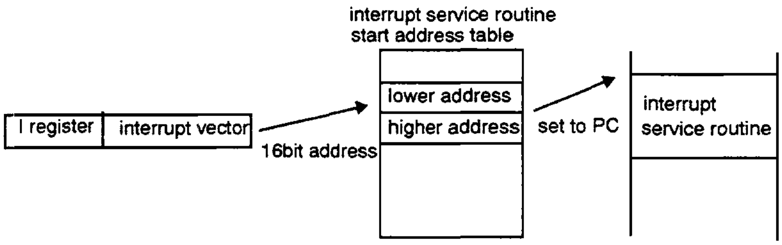
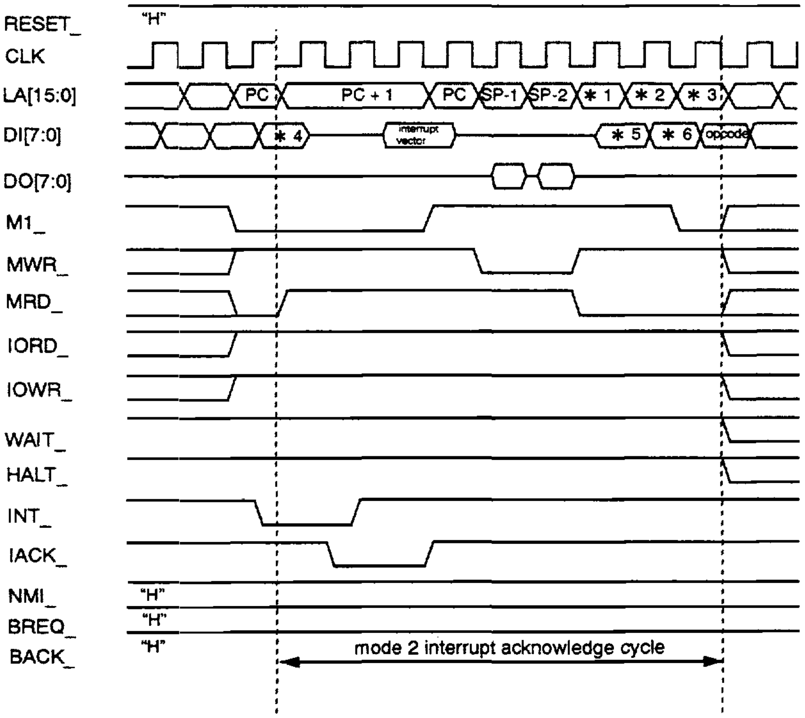


Figure 5-23-A. Mode 2 interrupt



*1:Address for the interrupt service routine address table.

*2:Address + 1 for the interrupt service routine address table.

*3:Starting address for the interrupt service routine.

*4:Fetched data here will be refetched on return.

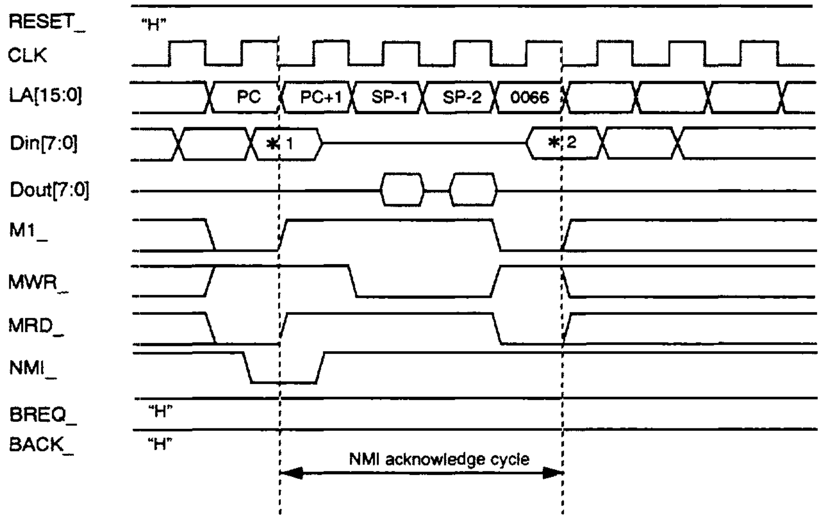
*5:Low address byte for the interrupt service routine.

*6:High address byte for the interrupt service routine.

Figure 5-23-B. Mode 2 timing

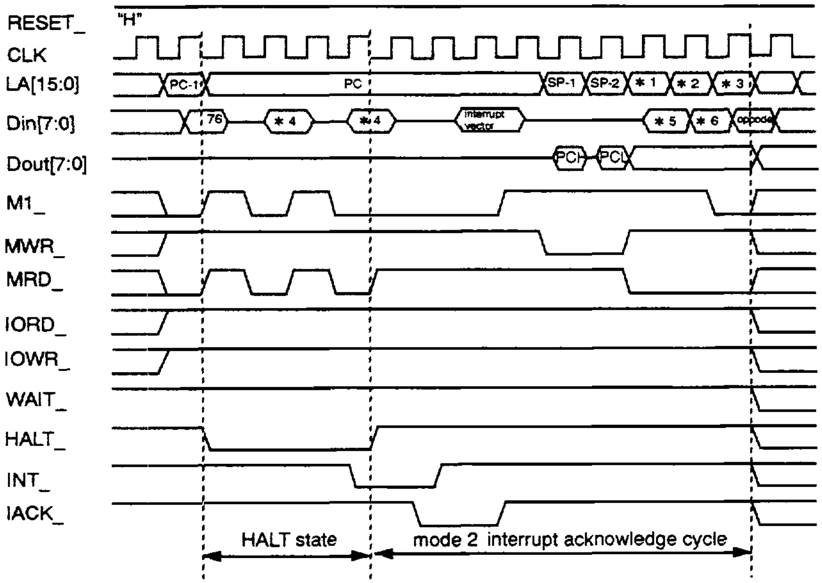
5.6.5 Reset timing

By keeping RESET_ input "L" for the minimum 3 clock cycles, the KC82 is reset. During reset cycle, AEN_ goes "H". When RESET_ goes inactive ("H"), opcode fetch cycle will be initiated on the falling edge of 3rd clock, and starts from 0000h. The interrupt mode is set to Mode 0. IFF1, IFF2 flags and I, R registers are reset.



*1:Fetched data here will be refetched on return.
*2:Data at 0066h.

Figure 5-24. Non-maskable interrupt timing



*1:Address for the interrupt service routine address table.
*2:Address + 1 for the interrupt service routine address table.
*3:Starting address for the interrupt service routine.
*4:Address of the following instruction to the HALT.
*5:Low address byte for the interrupt service routine.
*6:High address byte for the interrupt

Figure 5-25. HALT Exit (Mode 2)

5.7 MMU

5.7.1 General description

The MMU block is the circuit which converts the KC82 16 bit logical address, LA[15:0], to 20 bit physical address, A[19:0]. However, the MMU converts addresses only in case of memory access. It does not affect addresses in case of I/O access. The MMU is consisted of the following registers and a physical address calculation unit. The operation of the physical address calculation unit is described in the later section.

Table 5-1. Register Architecture

Name		Number of Bits	Read / Write
Boundary/Base Register	1 (BBR1)	8bit	R/W
〃	2 (BBR2)	〃	R/W
〃	3 (BBR3)	〃	R/W
〃	4 (BBR4)	8bit*	R/W
Base Register	1 (BR1)	8bit	R/W
〃	2 (BR2)	〃	R/W
〃	3 (BR3)	〃	R/W
〃	4 (BR4)	〃	Read Only

*Upper 2 bit is Read Only

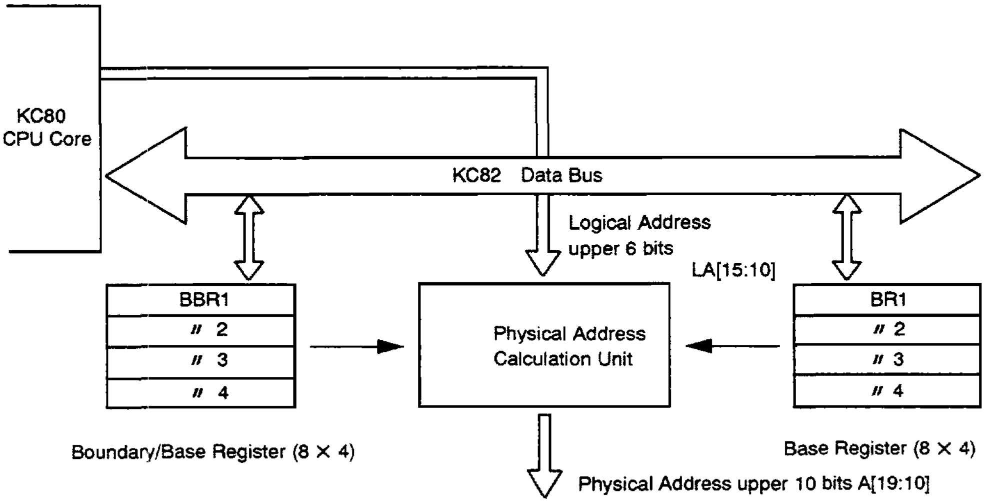


Figure 5-26. Block diagram

5.7.2 Description of registers

Eight 8 bit registers are included in the MMU block. These 8 registers hold four 10 bit data and four 6 bit data which are necessary for the MMU block.

Boundary/base registers (BBR1 ~ BBR4)

The upper 2 bits of these registers (A1<1:0> ~ A4<1:0>) are used to hold base addresses in the physical address space with the 8 bits of base registers. The lower 6 bits are logical boundary addresses (B1<5:0> ~ B4<5:0>). They are used as the boundary addresses to divide the logical space into five regions. The upper 2 bits of BBR4 are fixed, and the contents don't change when data is written on it.

Base registers(BR1 ~ BR4)

These registers compose 10 bit physical address bases (A1<9:0> ~ A4<9:0>) with the upper 2 bits of boundary/base resisters. They are used to hold base addresses in the physical address space. The content of BR4 is fixed to F0H.

Logical boundary address and physical address base

The MMU block divides the logical address space into five regions, and each region is mapped to the physical address space. The MMU block needs five logical boundary addresses (B0 ~ B4) and five physical address bases (A0 ~ A4). The content of A0 is fixed to 000H, and that of B0 is fixed to 00H. The other data are assigned in boundary/base registers and base registers as shown in Figure 5-27.

I/O Address	Register Name	bit7 6	5 0
00H	BBR1	A1<1:0>	B1<5:0>
01H	BR1	A1<9:2>	
02H	BBR2	A2<1:0>	B2<5:0>
03H	BR2	A2<9:2>	
04H	BBR3	A3<1:0>	B3<5:0>
05H	BR3	A3<9:2>	
06H	BBR4	A4<1:0> 00B Fixed	B4<5:0>
07H	BR4	A4<9:2> F0H Fixed	

Figure 5-27. MMU Register Mapping

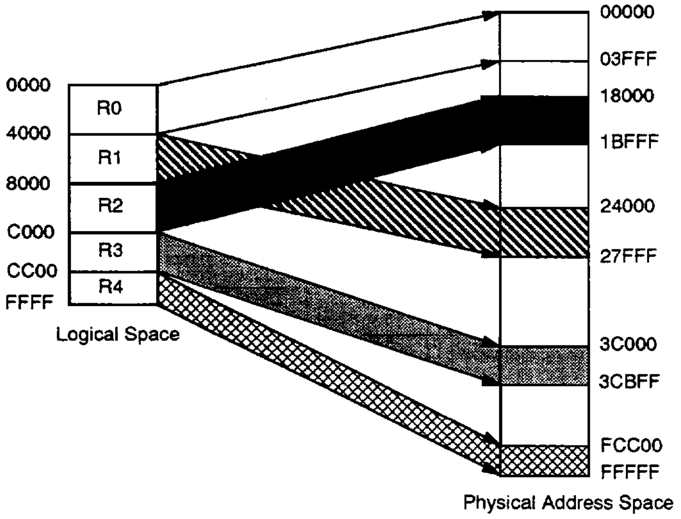
5.7.3 Calculation of physical address by MMU

The logical address space is divided into five regions (R0 ~ R4) as shown in Figure 5-28. The five regions are decided with logical boundary address (B0<5:0> ~ B4<5:0>). The lower limit address of Rn region is (Bn+1)*400H, and the higher limit address of Rn region is (B(n+1)+1)*400H-1. (The R0's lower limit address is always 00000H and the R4's higher limit address is always FFFFFH.) These regions in the physical space are created by adding the physical address base (10 bits) of divided each region and the upper 6 bits of logical address. (See Figure 5-29.) The division of logical space is decided by comparison of the upper 6 bits of logical address and the logical boundary address (B0 ~ B4). Setting of 1K byte unit is possible. The lowest value of the R1 region's lower limit address is 0400H and the content of A0 is fixed to 000H. Therefore, the lowest 1K byte of logical space (0000H ~ 03FFH), which is belong to R0 region, is always mapped to 00000H ~ 003FFH in the physical space.

An example is shown in Figure 5-28. For example, bit data are set as following;

B0 = 00H (Fixed)
B1 = 0FH
B2 = 1FH
B3 = 2FH
B4 = 32H
A0 = 000H (Fixed)
A1 = 080H
A2 = 040H
A3 = 0C0H
A4 = 3C0H (Fixed)

The relationship of five logical regions and physical regions is followed.



Region R0	Logical Address 0000H ~ 3FFFH	Physical Address 00000H ~ 03FFFH
Region R1	Logical Address 4000H ~ 7FFFH	Physical Address 24000H ~ 27FFFH
Region R2	Logical Address 8000H ~ BFFFH	Physical Address 18000H ~ 1BFFFH
Region R3	Logical Address C000H ~ CBFFH	Physical Address 3C000H ~ 3CBFFH
Region R4	Logical Address CC00H ~ FFFFH	Physical Address FCC00H ~ FFFFFH

Figure 5-28. Example of Relationship of Logical Address and Physical Address

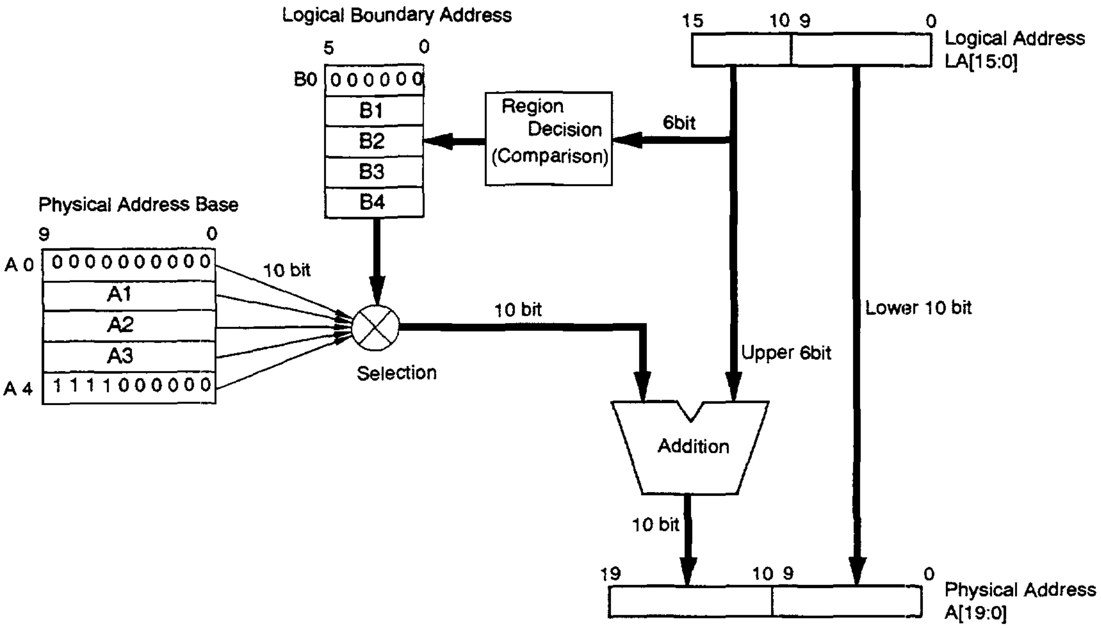


Figure 5-29. Physical Address Calculation

5.7.4 MMU mechanism

(1) Memory space

The MMU converts the address when the CPU accesses memory space. This is divided into following cases.

1.Instruction fetch
2.Read or write to memory space by instruction
3.Instruction fetch to interrupt restart address
4.Start address table in mode 2 interrupt

(2) I/O space

When the CPU accesses I/O space, the MMU does not convert the address and outputs the logical address. In this case, 0H is put on upper 4 bits of the address.

(3) DMA controller

The MMU does not convert the address which the DMA outputs.

5.7.5 Reset

The registers are initialized when they are reset as following.

B0	=	00H (Fixed)
B1	=	3FH
B2	=	3FH
B3	=	3FH
B4	=	3FH
A0	=	000H (Fixed)
A1	=	000H
A2	=	000H
A3	=	000H
A4	=	3C0H (Fixed)

When they are reset, there is only R0 region in the logical address space. 64K byte of the logical address space is mapped to the lowest 64K byte in the physical address space.

5.7.6 Precautions on the use of the MMU

(1)When the logical boundary address is set as B0 < B1 < B2 < B3 < B4, all regions are valid. When the logical boundary addresses are set in opposite order or equally, the region whose region number is bigger is valid. The region whose region number is smaller is disappears. For example, if B1 >= B2 is set, R1 region disappears.

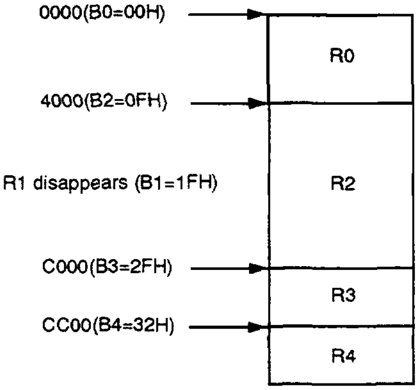


Figure 5-30. Example when B1 ≧ B2

(2)When data is written in MMU registers (boundary/ base registers, base registers), the set value is valid from the next bus cycle of I/O write cycle in which data is written in MMU registers.

(3)There is no hardware restriction for the use of each region(R0 ~ R4). However, the following assignments are recommended.

R0: resident program region (common program, interrupt vector, etc.)
R1: program bank window
R2: data bank window 1 (source)
R3: data bank window 2 (destination)
R4: resident data region (stack etc.)

(4)This MMU circuit occupies the following I/O addresses. Note that these I/O addresses cannot be assigned for user's I/O.

Table 5-2 I/O. Mapping MMU circuit uses.

I/O Address	Register Name
00H	BBR1
01H	BR1
02H	BBR2
03H	BR2
04H	BBR3
05H	BR3
06H	BBR4
07H	BR4
08H ~ 0FH	Reserved For Kawasaki Steel corp.

(5)The logical boundary address of R1 is decided by the R1 logical boundary address (B1). Because the lower limit address of the R1 region is (B1+1)*400H, the lowest value of the R1 region lower limit address is 0400H. On the other hand, the lower limit address of the R0 region is 0000H. Therefore, the lowest 1K byte of logical address is always belong to the R0 region. This lowest 1K byte is always mapped to the fixed location, 00000H ~ 003FFH.

(6)Set a value under 3FH to B1 ~ B4. 3FH to B1 ~ B4 is an invalid value and the corresponding region disappears.

6. Interrupt Controller

6.1 General description

The KL5C80A16 includes a KP69 macro cell as interrupt controller. The KP69 is a small-sized interrupt controller developed exclusively for microcontroller based on our CPUs (KC80 or KC82). The KP69 can support 16 levels of interrupt request inputs for the KC80 or KC82's Mode 2 interrupt. Each interrupt request can be set to the HIGH or LOW priority group, and within each group a higher number of interrupt request input level has higher priority. The edge/level operation and mask condition of each interrupt request input, higher 3 bits of interrupt vector can be programmed.

When the KP69 receives an interrupt request, it determines its mask condition and priority and sends the INT_ signal to the CPU. When the IACK_ signal returns from the CPU, it sends a programmed interrupt vector to the data bus. It also recognizes the end of interrupt service routine with EOI_= "L" from the CPU. This EOI_ signal goes low while executing the RETI instruction. Therefore, if the RETI instruction is placed at the end of interrupt service routine, the KC69 automatically recognizes the RETI instruction and the end of interrupt service routine. The KC69 has also an illegal interrupt detection capability.

Features

- Support KC80/KC82 mode 2 interrupt.
- The priorities of 16 level interrupt request inputs can be programmed.
- Each interrupt request maskable.
- Nested interrupt operation possible.
- The edge/level mode of interrupt request input can be selected.
- Illegal interrupt detection capability
- Automatically detects the execution of the CPU's RETI instruction and the end of interrupt service routine.

6.2 Block diagram

The following is the block diagram of KC69. The KC69's IACK_ input, EOI_ input and INT_ output are internally connected to the CPU's IACK_ output, EOI_ output and INT_ input respectively. The signals shown in the table are connected to IR[15:0].

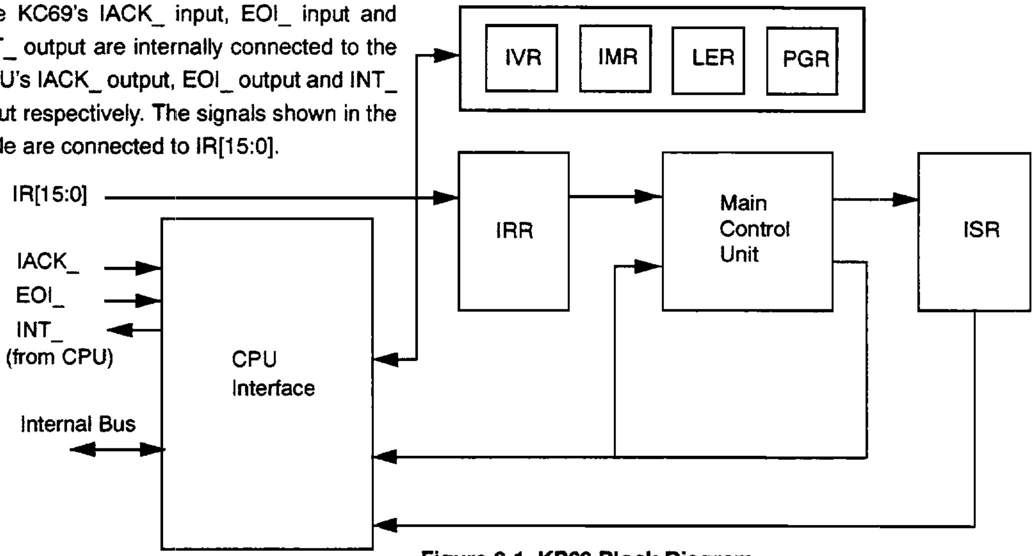


Figure 6-1. KP69 Block Diagram

Table 6-1. KL5C80A16 Interrupts

	Source of interrupt request
IR[15]	External input (P23)
IR[14]	External input (P22)
IR[13]	Timer/Counter channel 1 interrupt
IR[12]	Timer/Counter channel 0 interrupt
IR[11]	UART break detection/error detection interrupt (channel 0)
IR[10]	USART RXRDY output (channel 0)
IR[9]	USART TXRDY output (channel 0)
IR[8]	Clocked Serial I/O (channel 0)
IR[7]	DMTC1 of DMA controller
IR[6]	DMTC0 of DMA controller
IR[5]	UART break detection/error detection interrupt (channel 1)/External input P21
IR[4]	UART RXRDY output (channel 1)
IR[3]	UART TXRDY output (channel 1)
IR[2]	Clocked Serial I/O (channel 1)/External input P20
IR[1]	Timer/Counter channel 3 interrupt/External input P21
IR[0]	Timer/Counter channel 2 interrupt/External input P20

6.3 Register architecture and I/O mapping

KP69 contains the following registers.

Table 6-2. I/O Mapping

I/O address	Block	Write cycle	Read cycle
34H	Interrupt controller	LERL／PGRL	ISRL
35H		LERH／PGRH	ISRH
36H		IMRL	IMRL
37H		IVR ／ IMRH	IMRH

IRR (Interrupt Request Register)
(not readable, not writable)
The corresponding bits are set when an interrupt request is generated. When the interrupt request is accepted in the level mode, this register is reset by disappearance of the request. When the interrupt request is accepted in the edge mode, it is reset by the start of interrupt service. The edge detection circuits of all interrupt levels are reset right after the reset.

ISR (In Service Register)
(read only)
When the IACK_ signal is returned, the bits corresponding to the accepted interrupt request are set. They are reset at the end of request service. All bits are reset right after the reset. Each 8 bits can be read.

IVR (Interrupt Vector Register)
(write only)
This register specifies higher 3 bits of interrupt vector of the KC82 Mode 2 interrupt. Writable registers change when data is written into this register.

	D7	D6	D5	D4	D3	D2	D1	D0
ISRH	ISR[15]	ISR[14]	ISR[13]	ISR[12]	ISR[11]	ISR[10]	ISR[9]	ISR[8]

	D7	D6	D5	D4	D3	D2	D1	D0
ISRL	ISR[7]	ISR[6]	ISR[5]	ISR[4]	ISR[3]	ISR[2]	ISR[1]	ISR[0]

ISR[n]	description
0	Out of service
1	In service

	D7	D6	D5	D4	D3	D2	D1	D0
IVR	IVR[7]	IVR[6]	IVR[5]	☒	☒	☒	☒	☒

LER (Level / Edge Register)
(write only)

This register controls the level/edge mode of each interrupt request input (hereinafter referred to as "IR input"). The mode can be set for each IR input. All bits are set to the level mode right after the reset. Be sure to write data to this register before setting IVR. This register is write only.

Note: To use an interrupt of timer/counter, the corresponding IR input should be set to the edge mode.

PGR (Priority Group Register)
(write only)

This register controls priority groups of IR inputs. The priority group can be set for each IR input. There are two kinds of priority groups; "HIGH" and "LOW". All bits are set to LOW right after the reset. Be sure to write data to this register after setting IVR. This register is write only.

	D7	D6	D5	D4	D3	D2	D1	D0
LERH	LER[15]	LER[14]	LER[13]	LER[12]	LER[11]	LER[10]	LER[9]	LER[8]

	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
LERL	LER[7]	LER[6]	LER[5]	LER[4]	LER[3]	LER[2]	LER[1]	LER[0]

LER[n]	description
0	LEVEL mode
1	EDGE mode

	D7	D6	D5	D4	D3	D2	D1	D0
PGRH	PGR[15]	PGR[14]	PGR[13]	PGR[12]	PGR[11]	PGR[10]	PGR[9]	PGR[8]

	D7	D6	D5	D4	D3	D2	D1	D0
PGRL	PGR[7]	PGR[6]	PGR[5]	PGR[4]	PGR[3]	PGR[2]	PGR[1]	PGR[0]

PGR[n]	description
0	"LOW" group
1	"HIGH" group

IMR (Interrupt Mask Register)
(readable, writable)

This register mask IR inputs. The mask state can be set for each IR input. All bits are set to "1" right after the reset. Be sure to write data to this register after setting IVR. This register is write/read enable.

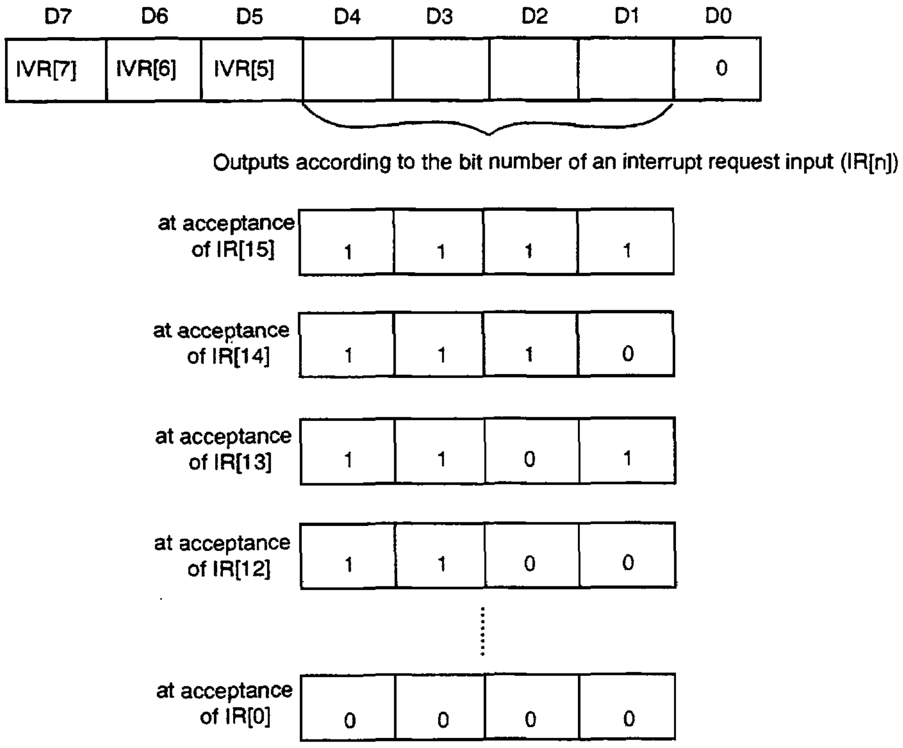
	D7	D6	D5	D4	D3	D2	D1	D0
IMRH	IMR[15]	IMR[14]	IMR[13]	IMR[12]	IMR[11]	IMR[10]	IMR[9]	IMR[8]

	D7	D6	D5	D4	D3	D2	D1	D0
IMRL	IMR[7]	IMR[6]	IMR[5]	IMR[4]	IMR[3]	IMR[2]	IMR[1]	IMR[0]

IMR[n]	description
0	non-masked state
1	masked state

6.4 Interrupt vector output

The following is the output format of interrupt vector.
The Figure 6-2 shows the interrupt vector output as response to IACK_.



D7	D6	D5	D4	D3	D2	D1	D0
IVR[7]	IVR[6]	IVR[5]					0

Outputs according to the bit number of an interrupt request input (IR[n])

at acceptance of IR[15]	1	1	1	1
at acceptance of IR[14]	1	1	1	0
at acceptance of IR[13]	1	1	0	1
at acceptance of IR[12]	1	1	0	0
		⋮		
at acceptance of IR[0]	0	0	0	0

bits 7 to 5: data set in IVR
bits 4 to 1: binary code of accepted interrupt level
bit 0 : always 0

Figure 6-2. Interrupt Vector

6.5 Priorities of interrupt requests

The priorities of interrupt request inputs are set in descending order of bit numbers right after the reset. The priorities can be changed in terms of priority group. There are two kinds of priority groups ("LOW" and "HIGH") and the priority group can beset for each IR. The IRs of "HIGH" group have higher priority over those of "LOW" group. Within each group, an IR with higher bit number has higher priority. The following is an example of IR priorities.

Interrupt request input	Group
IR[15]	LOW
IR[14]	LOW
IR[13]	LOW
IR[12]	HIGH
IR[11]	LOW
IR[10]	HIGH
IR[9]	LOW
IR[8]	HIGH
IR[7]	HIGH
IR[6]	LOW
IR[5]	LOW
IR[4]	HIGH
IR[3]	LOW
IR[2]	HIGH
IR[1]	HIGH
IR[0]	HIGH

⇒

Interrupt request input	Group	Priority
IR[12]	HIGH	Highest
IR[10]	HIGH	↓
IR[8]	HIGH	↓
IR[7]	HIGH	↓
IR[4]	HIGH	↓
IR[2]	HIGH	↓
IR[1]	HIGH	↓
IR[0]	HIGH	↓
IR[15]	LOW	↓
IR[14]	LOW	↓
IR[13]	LOW	↓
IR[11]	LOW	↓
IR[9]	LOW	↓
IR[6]	LOW	↓
IR[5]	LOW	↓
IR[3]	LOW	Lowest

Figure 6-3. Interrupt Request Priority

6.6 Register setting sequence

I/O addresses are assigned to the internal registers of KC69 as shown in Table 6-2. So initialization of registers after clearing the reset is performed in the following order: LER, IVR and IMR (or PGR). It should be noted that only IMR or PGR can be set after setting IVR.

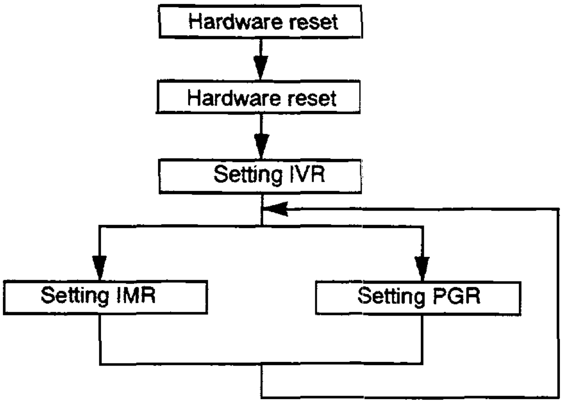


Figure 6-4. Register Setting Sequence

6.7 Readout of registers

Of the KP69 registers, ISR and IMR are always readable. Readout of ISR is required to determine an illegal interrupt.

6.8 Acceptance of interrupt request

There are two modes (level and edge) for the acceptance of interrupt request and they are set by LER.

Level mode

In level mode, "H" level of IR input is recognized as interrupt request.

Edge mode

In edge mode, the rising edge of IR input is recognized as interrupt request. In this case, the interrupt request is held until accepted.

Operation sequence

The following is the operations when an interrupt request is generated at IR[n] pin.

When IR input goes "H" to generate an interrupt request, the corresponding bits of IRR are set. This interrupt request becomes an INT_ signal through the mask conditions determined by IMR and the priority decision by ISR and PGR. The KC82 receives the signal and makes the IACK_ signal "L". When the KP69 receives this IACK_ signal, it outputs an interrupt vector corresponding to the IR[n] where the interrupt request has been generated to set the corresponding bit of ISR and (in edge mode) reset the corresponding bits of IRR. Then the CPU enters the interrupt service routine. When ISR is set, acceptance of an interrupt request in edge mode is enabled.

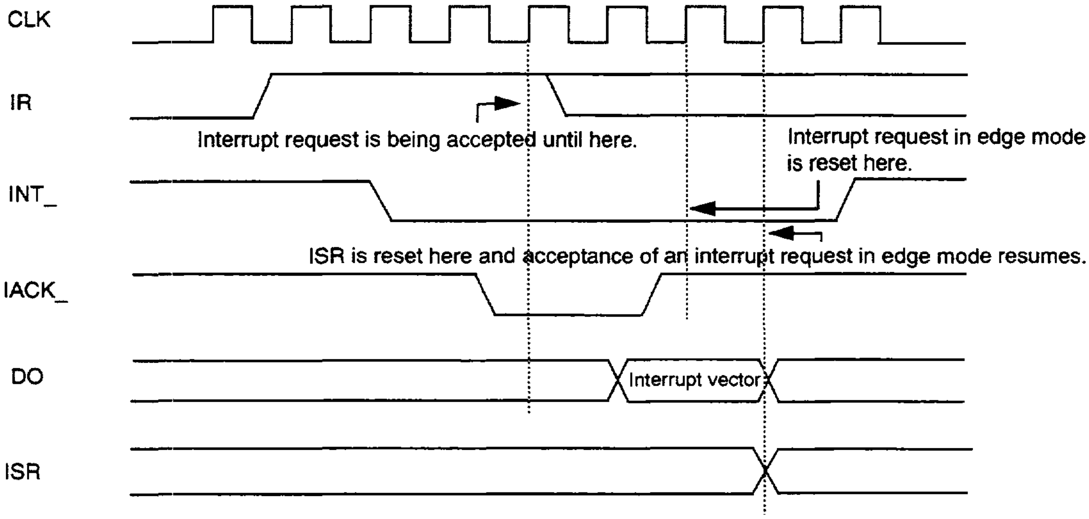

CLK
IR
Interrupt request is being accepted until here.
Interrupt request in edge mode is reset here.
INT_
ISR is reset here and acceptance of an interrupt request in edge mode resumes.
IACK_
DO
Interrupt vector
ISR

Figure 6-5. Interrupt Request Accepting Timing

6.9 End of interrupt

When the EOI_ signal from the KC82 goes "L", the KP69 resets the ISR bits corresponding to the highest priority level of the interrupt being serviced. This terminates the interrupt service of that level.

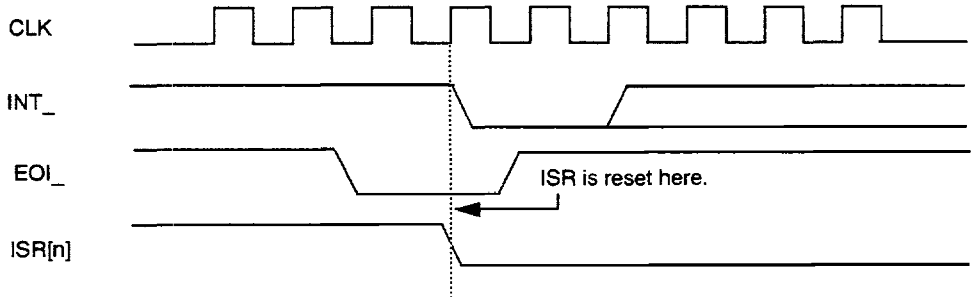


Figure 6-6. Interrupt Terminating Timing

6.10 Nested interrupt operation

The KP69 compares the interrupt level of a new interrupt request with that of the interrupt request being serviced in terms of the predetermined priority. When the interrupt level of a new interrupt request has higher priority than that of the interrupt request being serviced, the KP69 accepts a new interrupt request, otherwise the KP69 rejects it or keeps it wait. This implements a nested interrupt operation. The state of nested interrupts can be obtained by reading ISR.

6.11 Illegal interrupt operation

An illegal interrupt indicates the following state: when an interrupt request input set to the level mode made INT_ signal "L" and the KC82 returned IACK_ as response, the interrupt request has already disappeared and there is no higher interrupt request input than the interrupt request being serviced with the highest priority. In this state the KP69 does not set ISR and outputs an interrupt vector corresponding to IR[0], entering an illegal interrupt operation state which lasts until the EOI_ from the KC82. ISR is not reset by this EOI_. Any interrupt request (legal or illegal) would not be accepted in the illegal interrupt operation state. Therefore, during the IR[0] interrupt service it is necessary to distinguish a normal interrupt request or an illegal interrupt request based on ISR[0]. When the KP69 goes to an illegal interrupt request state during the IR[0] interrupt service, the readout of ISR gives ISR[0]=0. This indicates that ISR[0]=1 represents a legal interrupt request and ISR[0]=0 represents an illegal interrupt request.

6.12 Reset

When the RESET_ goes "L", the following operations are performed.

(1) IMR is set to "FFFFH". (All levels in masked state)
(2) IRR, ISR, LER and PGR are reset to "0000H".
(3) Illegal interrupt operation state is disabled.
(4) IVR is placed in non-set state. (Initialization is required.)

6.13 Precautions

(1) Be sure to use Mode 2 for CPU interrupt mode.
(2) Be sure to place RETI instruction (source code: ED 4D) at the end of interrupt service routine.
(3) To use an interrupt of timer/counter, the corresponding IR input should be set to the edge mode.

7. DMA Controller

7.1 General description

KL5C80A16, as DMA controller, has a KP27 macro cell built-in. KP27 is our original compact but fast programmable DMA controller with two independent DMA channels. In each channel, it is possible to set the transfers of memory → memory, memory → I/O and I/O → memory.

The transfer between memories is started by the software request. For the transfer between the memory and I/O, it is possible to specify the valid polarity of DMA request signal and the transfer mode (single/demand transfer) etc. Although the transfer is executed in the way which outputs two types of address, the high-speed one bus cycle transfer is possible by latching the address, with the address latch signal. Also, it is possible to connect to the device which has DMA functions outside. The bus request signal from the device and the DMA request signals from KP27 to each channel are then arbitrated. This device also has functions to discontinuation and resume the DMA operations through NMI.

When KP27 receives a DMA request or bus request from the external DMA device, it determines the enable condition and priority of each channel, requesting bus to the CPU. After CPU releases the bus, KP27 obtains the bus, and executes the DMA operations or provides the external DMA device with the bus, waiting for the completion of the external DMA device. After the DMA operations are completed, the system bus is returned to CPU.

Features

- Clock synchronous type DMA controller. The maximum clock rate is 10MHz.
- It is possible to access the memory space of 1 Mbyte without MMU.
- It is possible to specify the I/O address in the program because KP27 transfers data by outputting two types of address, source address and destination address.
- It is possible to access in DMA transfer by the DACK_ signal, because the DACK_ signal is outputtable.
- The priority is channel 1 > channel 0.
- In each channel, it is possible to set the transfers of memory → memory, memory → I/O and I/O → memory.
- In setting the transfers of memory → I/O and I/O → memory, it is also possible to specify the single/demand transfer, valid polarity of DREQ and Auto Initialize function.
- Continuing transfer with the maximum 64 K byte is possible.
- One transfer is executed with three clocks. Therefore, the maximum transfer rate is 3.3 Mbyte/sec.
- It is possible to connect to the device which has the DMA functions outside. It is possible to specify the priority of the external DMA device.
- It is possible to discontinue/resume the DMA operations through NMI.
- channel 1 can be used exclusively for UART channel 0 by setting SCR2.

7.2 Block diagram

The following is the block diagram of the DMA controller. See Section 7.3 on next page for each signal in the diagram.

The DMTC1 and DMTC0 outputs are connected to IR[7] and IR[6] of the interrupt controller respectively. The bus signal and the bus approval signal are internally connected to the CPU's BREQ_ input and the CPU's BACK_ output respectively. The NMI_ input is connected to the same signal as the CPU's NMI_ input.

It is possible to connect DREQ0 to the signal input from the pin 85 by setting the parallel port P24 as input. One of such signals as the external signal input from the pin 84, TXRDY0 or RXRDY0 from KP61 can be internally connected to DREQ1 by setting SCR2.(See table 7-1)

By specifying the parallel port P26 in the input direction, the input from the pin 83 can be used as EXBREQ_. The EXBACK_ is then connected to the pin 91 by specifying SCR2.

KP 27 can access the I/O device for DMA transfer by DACK_ signal. The DACK0_ is multiplexed with the PALAT, and is allocated to the pin 5. This switchover is performed by SCR2. It is also possible to output the DACK1_ from the pin 92 by setting the SCR2. See Chapter 12 for particulars.

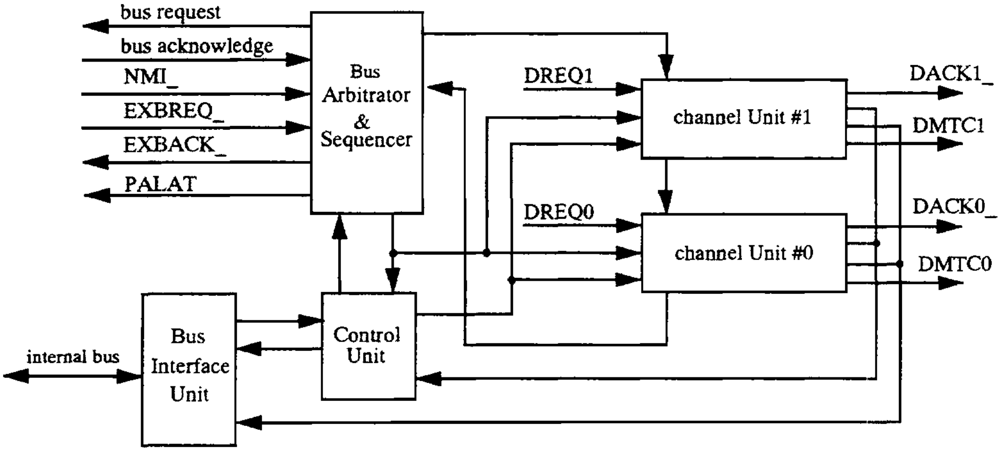


Figure7-1 DMA controller block diagram

Table7-1 DMA request source internally connectable to DMA channel 1

DMA request source	SCR2 D<1:0>	pin92
pin84	00	P16
	01	DACK1_
UART TXRDY0	10	P16
UART RXRDY0	11	P16

7.3 Functions of signals

Pin	I/O	Functions
DREQ0 DREQ1	I	DMA request signal input. The DMA request signal is input for each channel. It is possible to specify the valid polarity with "H (↑)" or "L (↓)."
DACK0_ DACK1_	O	DMA request acknowledge signal output. After receiving DREQ, outputs the acknowledge signal for the I/O device subject to DMA transfer. Active low.
EXBREQ_	I	External device bus request signal input. When connected to the device which has the DMA functions outside, inputs its bus request signal. Active low.
EXBACK_	O	External device bus request acknowledge signal output. When connected to the device which has the DMA functions outside, connects to the bus request acknowledge signal input pin of the device. Active low.
PALAT	O	Latch signal (including peripheral address) output. This signal changes from "H" to "L" just before the value output to the address bus changes from the I/O address to the memory address in DMA transfer between memory and I/O. It then changes from "L" to "H" after two bus cycles. By latching the I/O address to the external latch with this signal, and inputting it to each I/O device, the I/O device subject to transfer can be programmably selected.
DMTC0 DMTC1	O	Terminal count output. This signal indicates that each channel has reached the TC condition. The TC condition is indicated with "H." The channels reach the TC condition, when the transfer of the byte a number of which is previously set is executed. In this signal, DMTC0 and DMTC1 are internally connected to IR[6] and IR[7] of the interrupt controller respectively.

Note: 0 and 1 of the signal name indicate channels 0 and 1 respectively.

7.4 Internal register mapping

The internal register mapping of the KP27 is shown on next page. See Section 7.5 for the functions of each register. BSFF in Table 7-2 is a two bit counter which reads/writes each internal register with 8 bits. For this also, see the final part of Section 7.5.

Table7-2 I/O register mapping

I/O address	write	read	BSFF
10H	channel 0 B-PAR0 channel 0 B-PAR1 channel 0 B-PAR2	channel 0 C-PAR0 channel 0 C-PAR1 channel 0 C-PAR2	00 01 10
11H	channel 0 B-SAR0 channel 0 B-SAR1 channel 0 B-SAR2	channel 0 C-SAR0 channel 0 C-SAR1 channel 0 C-SAR2	00 01 10
12H	channel 0 B-BCR0 channel 0 B-BCR1	channel 0 C-BCR0 channel 0 C-BCR1	00 01
13H	channel 0 CR	channel 0 SR0 channel 0 SR1	00 01(only for read)
14H	channel 1 B-PAR0 channel 1 B-PAR1 channel 1 B-PAR2	channel 1 C-PAR0 channel 1 C-PAR1 channel 1 C-PAR2	00 01 10
15H	channel 1 B-SAR0 channel 1 B-SAR1 channel 1 B-SAR2	channel 1 C-SAR0 channel 1 C-SAR1 channel 1 C-SAR2	00 01 10
16H	channel 1 B-BCR0 channel 1 B-BCR1	channel 1 C-BCR0 channel 1 C-BCR1	00 01
17H	channel 1 CR	channel 1 SR0 channel 1 SR1	00 01(only for read)

The PAR, SAR and BCR registers width of which is over 8 bits can be classified with the last digit of their names in Table 7-2. "0," "1" and "2" indicate <7:0>, <15:8> and <19:16> of each register respectively. Also, "B" and "C" of "B-PAR0," "C-PAR0," etc. indicate "Base" and "Current" respectively. See the following Section 7.5 for particulars.

The detailed information on CR (Command Register) in Table 7-2 is shown in Table 7-3. KP27 selects the command register subject to write-down according to the bit number on the MSB side of the written data. The corresponding table is as follows. See Section 7.5 for the format of each register.

Table7-3 Command register mapping

I/O address	Data bus	Command registers
13H	D<7> =0 D<7:6> =10	channel 0 CR0 channel 0 CR1
17H	D<7> =0 D<7:6> =10	channel 1 CR0 channel 1 CR1
13H or 17H	D<7:5> =110 D<7:4> =1111	CR2 (Common command register) NCC (Common command register)

7.5 Internal register architecture

KP27 is generally classified into the following two types of internal register.

- Internal registers existing in each channel.
- Common internal registers that don't belong to any channel.

The details of the above two types are described below.

For your reference, KP27 outputs two types of address in DMA transfer. The address which is output first is Primary Address, and that output later is Secondary Address.

In the transfers of memory → I/O and I/O → memory,

PA = I/O address
SA = Memory address, and

In the transfer of memory → memory,

PA = Memory address in the source
SA = Memory address in the destination

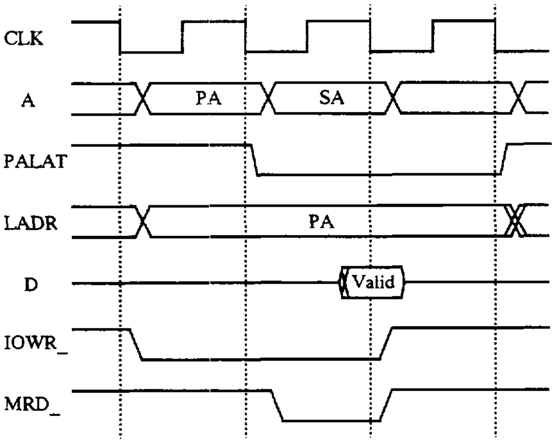


Figure7-2 KP27 basic DMA transfer timing (memory → on-chip I/O)

LADR in Fig. 7-2 shows the signal which latches A (Address) with PALAT. Also, in Fig. 7-2, the I/O access signals and the memory access signals are regarded as the internal bus signals.

●Internal registers existing in each channel

PAR, SAR, BCR, CR0, CR1, SR0 and SR1 are available. PAR, SAR and BCR includes two types, Base and Current.

The base register specifies the initial-values by writing down, and the current register holds the address in action and the rest of byte number subject to transfer after the initial-values are automatically set from the base register. The initial-values from the base register to the current register are simultaneously loaded on PAR, SAR and BCR, when 1) the base register is written down, and 2) CR0 and CR1 (Command Resister the details of which are described later) are written down, and 3) after reaching TC (See 7.6 Section) in case that Auto Initialize (See 7.6 Section) is valid in the transfers of memory → I/O and I/O → memory.

Therefore, when data write is performed on one of three types of base register (B-PAR, B-SAR and B-BCR) for a certain channel, all the data of each base register are then loaded to three types of Current Register (C-PAR, C-SAR and C-BCR) for the channel. Simultaneously, the channel automatically enters disable conditions.

For instance, when write is performed on B-PAR of channel 1, the data of B-PAR, B-SAR and B-BCR of channel 1 are loaded to C-PAR, C-SAR and C-BCR respectively, and channel 1 enters disable conditions. However, this loading occurs only in the same channel. For instance, although the initial-values are written in the base register of channel 1, it has no influence on channel 0.

Also, write down in the common internal registers CR2 and NCC has no influence on any internal register data of any channel.

B-PAR (Base Primary Address Register) (writable, not readable 20 bit)

B-PAR holds the initial-values of Primary Address. The I/O address is set in the transfer of memory → I/O and I/O → memory, and the initial-values of the source memory address is set in the transfer of memory → memory. In case that the DACK_ signal is used for setting the I/O subject to transfer, the data of this register is output to the address bus. Immediately after resetting, all the bits become "0."

	D7	D6	D5	D4	D3	D2	D1	D0
B-PAR2	0	0	0	0				
B-PAR1								
B-PAR0								

C-PAR (Current Primary Address Register)
(not writable, readable 20 bit)

C-PAR keeps the Primary Address in the DMA operation. When B-PAR is specified, the values are simultaneously set. In setting the memory → memory transfer, after the execution of the software request, the device holds the memory address following the memory address which has completed the transfer thus far, namely, the memory address which will perform the data read in the next DMA transfer. Immediately after writing in each base register, CR0 and CR1, the source memory address set at B-PAR (initial-values) is automatically set again. In setting the transfer of memory → I/O and I/O → memory, the I/O address set in B-PAR is held. Immediately after resetting, all the bits become "0."

	D7	D6	D5	D4	D3	D2	D1	D0
C-PAR2	0	0	0	0				
C-PAR1								
C-PAR0								

B-SAR (Base Secondary Address Register)
(writable, not readable 20 bit)

B-SAR holds the initial-values of the Secondary Address. In the transfers of memory → I/O and I/O → memory, the initial-values of the memory address is specified, and, in the transfer of memory → memory, the initial-values of the destination memory address is set. Immediately after resetting, all the bits become "0."

	D7	D6	D5	D4	D3	D2	D1	D0
B-SAR2	0	0	0	0				
B-SAR1								
B-SAR0								

C-SAR (Current Secondary Address Register)
(not writable, readable 20 bit)

C-SAR holds the Secondary Address in the DMA operations.

When B-SAR is specified, the values are simultaneously set. In the transfers of memory → I/O, I/O → memory, and memory → memory, if the data has not reached TC, the device holds the memory address following the memory address which has completed the transfer thus far, namely, the memory address which will perform the access in the next DMA transfer.

In the transfer of memory → memory, the memory address specified at B-PAR (initial-values) is automatically set again, immediately after the data is written down in each base register, CR0 and CR1. In any cases of the transfers of memory → I/O and I/O → memory, it is automatically set again after the Auto Initialize valid time reaches TC, or immediately after the data is written in each register base, CR0 and CR1. Immediately after resetting, all the bits become "0."

	D7	D6	D5	D4	D3	D2	D1	D0
C-PAR2	0	0	0	0				
C-PAR1								
C-PAR0								

B-BCR (Base Byte Count Register)
(writable, not readable 16 bit)

B-BCR sets a total number of transfer byte in the DMA transfer. When a number of byte set in this register is transferred, the device enters TC conditions. Immediately after resetting, all the bits become "0."

	D7	D6	D5	D4	D3	D2	D1	D0
B-BCR1								
B-BCR0								

C-BCR (Current Byte Count Register)
(not writable, readable 16 bit)

C-BCR holds the rest transfer byte number in the DMA transfer.

When B-BCR is specified, the values are simultaneously set. In any case of the transfers of memory → I/O, I/O → memory and memory → memory, when the data has not reached TC, the device maintains the rest of a transfer byte number.

In the transfer of memory → memory, the total of bytes subject to transfer specified at B-BCR are automatically set again, immediately after the data is written down in each base register, CR0 and CR1. During the transfers of memory → I/O and I/O → memory, they are automatically set again in Auto Initialize setting after KP27 reaches TC, or immediately after the data is written down in each base register, CR0 and CR1. Immediately after resetting, all the bits become "0."

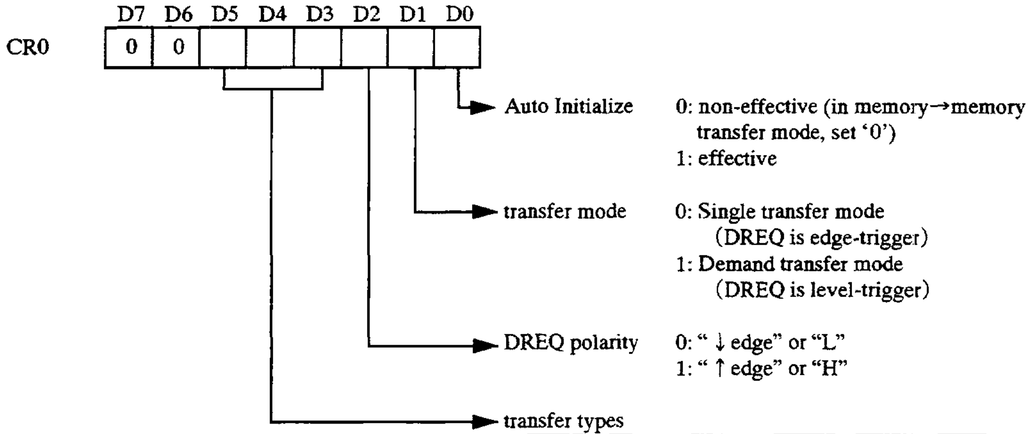
	D7	D6	D5	D4	D3	D2	D1	D0
C-BCR1								
C-BCR0								

CR0 (Command Register 0)
(writable, not readable 8 bit)

CR0 specifies and memorizes the transfer type and mode of each channel. By setting higher two bits of the written data to "00," this register can be selected. The meaning of each bit is shown in the figure of the following page.

When the single transfer is selected by "transfer mode," "DREQ polarity" can select "falling edge" or "rising edge." When the demand transfer is selected, "DREQ polarity" can select "L level" or "H level."

When D<5:4> = 00, D<2:1> is "don't care." Namely, when the transfer of memory → memory by the transfer type, neither DREQ polarity nor the setting of the transfer mode is neglected. Here, the Auto Initialize should be set invalid.



Auto Initialize — 0: non-effective (in memory→memory transfer mode, set '0')
1: effective

transfer mode — 0: Single transfer mode (DREQ is edge-trigger)
1: Demand transfer mode (DREQ is level-trigger)

DREQ polarity — 0: " ↓ edge" or "L"
1: " ↑ edge" or "H"

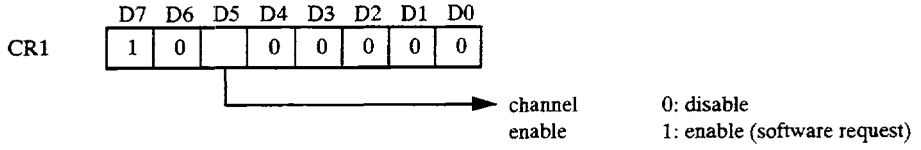
transfer types

D5	D4	D3	transfer types	memory address
0	0	0	memory→memory	increment
0	0	1	memory→memory	decrement
0	1	0	memory→I/O	increment
0	1	1	memory→I/O	decrement
1	0	0	I/O→memory	increment
1	0	1	I/O→memory	decrement
1	1	X	illeagal	———

CR1 (Command Register 1)
(writable, not readable 8 bit Valid bit: D<7:5>)

CR1 specifies and memorizes the DMA transfer enable/disable of each channel.

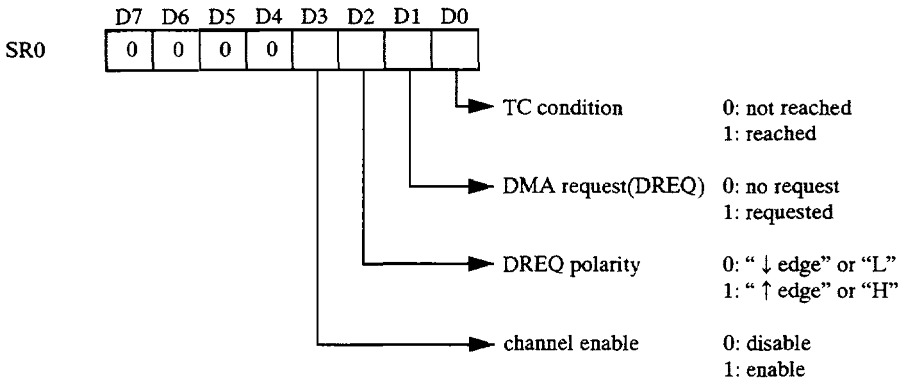
By setting higher 2 bits of the writing data at '10,' this register can be selected. The meaning of each bit is as follows. When the transfer of memory → memory is set by CR0, CR1 becomes the software request register. Namely, the software request is executed by writing down 'A0H' in CR1 of the channel in which the transfer of memory → memory is specified.



channel enable — 0: disable
1: enable (software request)

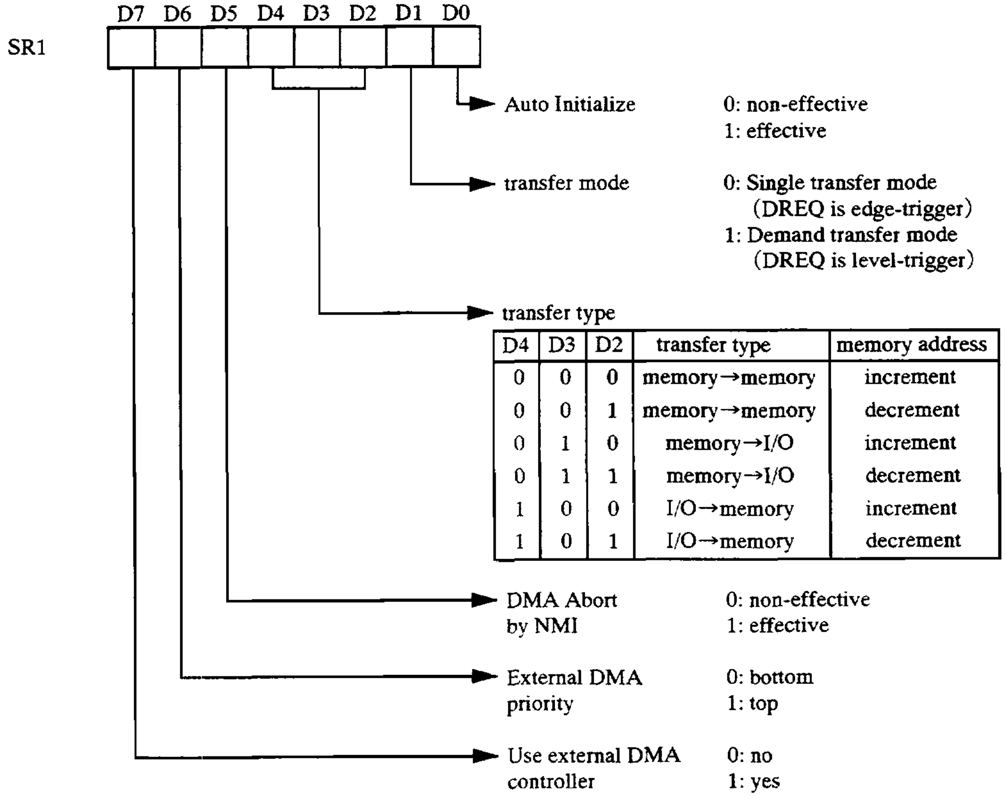
SR0 (Status Register 0)
(not writable, readable 8 bit Valid bit: D <3:0>)

SR0 indicates the status of each channel. '0000' can be read through D<7:4>. From this register, the information on current operation conditions of each channel can mainly be obtained. The meaning of each bit is shown in the figure on the next page. From all the bits '0's are read immediately after resetting.



SR1 (Status Register 1)
(not writable, readable 8 bit)

SR1 indicates the status of each channel and the setting of KP27 operation. The meaning of each bit is shown below. D<4:0> and D<7:5> indicate the data set by CR0 and CR1, and that set by CR2 (described later) respectively. From all the bits '0's are read immediately after resetting.



D4	D3	D2	transfer type	memory address
0	0	0	memory→memory	increment
0	0	1	memory→memory	decrement
0	1	0	memory→I/O	increment
0	1	1	memory→I/O	decrement
1	0	0	I/O→memory	increment
1	0	1	I/O→memory	decrement

●Common internal registers which do not depend on any channel

The following two registers, CR2 and NCC, are the common internal registers which do not depend on any DMA channel. It is possible to first write down either channel 0 or 1.

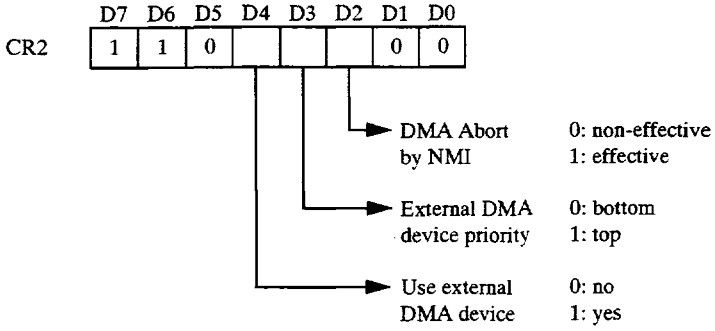
CR2 (Command Register 2)
(writable, not readable 8 bit Valid bit: D<7:2>

Among KP27's programmable operations, CR2 specifies the operation which do not depend on any channel. By setting D<7:5> = '110,' this register can be selected. The meaning of each bit is as follows.
D3's external DMA device can set the priority only when it is set so that the external DMA device can be used in D4.

NCC (NMI Clear Command)
(writable, not readable 8 bit Valid bit: D<7:4>)

When the DMA Abort by NMI_ is set valid by CR2, this command cancels the discontinued conditions, resuming the DMA transfer. The command can be executed by writing down the data in the condition of D<7:0> = 'F0H.'
When the DMA Abort by NMI_ is set invalid, write down of NCC command has no influence on the KP27's operation.

	D7	D6	D5	D4	D3	D2	D1	D0
NCC	1	1	1	1	0	0	0	0



BSFF

KP27 includes the independent BSFF (Byte Select Flip Flop) in channels 0 and 1 respectively. BSFF is a 2 bit counter which determines to access which internal register, when the I/O address allocated with some internal registers, is accessed. BSFF is used in order to access such 16 and 20 bit registers as PAR, SAR and BCR with eight bits.

BSFF of a certain channel is operated according to the following three rules.

◎ When the same type I/O access (I/O read or I/O write) is performed on the same I/O address as that in the previous I/O access (I/O read in both times, or I/O write in both times), increment should be performed on BSFF immediately after the subject access. However, when A[1] = '0' (PAR or SAR) and BSFF = '10,' BSFF = '00' immediately after accessing, and when A[1] = '1' (BCR or SR) and BSFF = '01,' BSFF = '00' immediately after accessing.

◎ When the access is performed on the different I/O address in the same channel as the previous I/O access, or when the defferent type I/O access is performed on the same address as the previous I/O access (first I/O read, second I/O write, vice versa), BSFF is immediately set to '00.' After the execution of access, increment is performed on BSFF.

◎ When I/O write is performed on A[1:0] = '11' (write on CR), BSFF is immediately set to '00.'

Typical setting of KP27's internal register is shown below. Here, BSFF0 and BSFF1 indicate BSFFs of channels 0 and 1 respectively. The ch0 and ch1 stand for channels 0 and 1 respectively.

Case 1: In case of the following instruction sequence, the operations corresponding to KP27's internal register are shown in the following (1) to (3).

```
LD      BC,     FF15H
OUT     (C),    B           ········ (1)
OUT     (C),    B           ········ (2)
OUT     (C),    B           ········ (3)
```

(1) When BSFF1 = '00,' I/O write on A[3:0] = '0101'.
→ ch1_B-SAR0 is written down, thereafter, BSFF1 = '01.'

(2) When BSFF1 = '01,' I/O write on A[3:0] = '0101' again.
→ ch1_B-SAR1 is written down, thereafter, BSFF1 = '10.'

(3) When BSFF1 = '10,' I/O write on A[3:0] = '0101' again.
→ ch1_B-SAR2 is written down, thereafter, BSFF1 = '00.'

By the above (1) to (3), 'FFFFFH' is set at ch1_B-SAR<19:0>.

Case 2: In case of the following instruction sequence, the operations corresponding to KP27's internal register are shown in the following (1) to (6).

```
LD      A,      A0H
OUT     (13H),  A           ········ (1)
XOR     A
OUT     (16H),  A           ········ (2)
IN      A,      (11H)       ········ (3)
IN      A,      (11H)       ········ (4)
LD      A,      80h
OUT     (16H),  A           ········ (5)
IN      A,      (10H)       ········ (6)
```

(1) When BSFF0 = '00,' I/O write with data 'A0H' on A[3:0] = '0011'.
→ ch0_CR1 is written down, thereafter, BSFF0 = '00.'

(2) When BSFF0 = '00', and BSFF1 = '00,' I/O write on A[3:0] = '0110.'
→ ch1_B-BCR0 is written down, thereafter, BSFF0 = '00'and BSFF1 = '01'

(3) When BSFF0 = '00,' and BSFF1 = '01,' I/O read on A[3:0] = '0001'.
→ ch0_C-SAR0 is read out, thereafter, BSFF0 =

'01' and BSFF1 = '01.'

(4) When BSFF0 = '01,' and BSFF1 = '01,' I/O read on A[3:0] = '0001' again.
→ ch0_C-SAR1 is read out, thereafter, BSFF0 = '10' and BSFF1 = '01.'

(5) When BSFF0 = '10,' and BSFF1 = '01,' I/O write on A[3:0] = '0110'.
→ ch1_B-BCR1 is written down, thereafter, BSFF0 = '10' and BSFF1 = '00.'

(6) When BSFF0 = '10,' and BSFF1 = '00,' I/O read on A[3:0] = '0000'.
→ BSFF0 is immediately set to '00,' reading ch0_C-PAR0, thereafter, BSFF0 = '01' and BSFF1 = '00.'

By the above (1), ch0_CR1 is set (ch0 enable). By the above (2) to (6), '8000H' is set on ch1_B-BCR, and ch0_C-SAR<15:0> and ch0_C-PAR<7:0> are read.

Case 3: In case of the following instruction sequence, the operations corresponding to KP27's internal register are shown with (1) to (7).

```
XOR    A
LD     BC,    4014H
LD     DE,    0E08H
OUT    (C),   B         ········ (1)
INC    C
OUT    (C),   A         ········ (2)
OUT    (C),   A         ········ (3)
OUT    (C),   D         ········ (4)
INC    C
OUT    (C),   E         ········ (5)
DEC    C
IN     A,     (C)       ········ (6)
ADD    A,     E
OUT    (C),   A         ········ (7)
```

(1) When BSFF1 = '00,' I/O write on A[3:0] = '0100'.
→ ch1_B-PAR0 is written down, thereafter, BSFF1 = '01.'

(2) When BSFF1 = '01,' I/O write on A[3:0] = '0101'.
→ BSFF1 is immediately set to '00,' writing ch1_B-SAR0. Thereafter, BSFF1 = '01.'

(3) When BSFF1 = '01,' I/O write on A[3:0] = '0101'.
→ ch1_B-SAR1 is written down, thereafter, BSFF1 = '10.'

(4) When BSFF1 = '10,' I/O write on A[3:0] = '0101'.
→ ch1_B-SAR2 is written down, thereafter, BSFF1 = '00.'

(5) When BSFF1 = '00,' I/O write on A[3:0] = '0101'.
→ When BSFF1 = '00' (unchangeable), ch1_B-BCR0 is written down, thereafter, BSFF1 = '10.'

(6) When BSFF1 = '10,' I/O read on A[3:0] = '0101'.
→ BSFF1 is immediately set to '00,' reading ch1_C-SAR0, thereafter, BSFF1 = '01.'

(7) When BSFF1 = '01,' I/O write on A[3:0] = '0101'.
→ BSFF1 is immediately set to '00,' writing ch1_B-SAR0, thereafter, BSFF1 = '01.'

By the above (1), 00040H is set at ch1_PAR<19:0>. By (2) to (4), 'E0000H' is set at ch1_SAR<19:0>. By (5), '0008H' is set at ch1_BCR<15:0>. By (6), ch1_SAR<7:0> is read out, and the calculation results are written back to ch1_SAR<7:0> by (7).

Thus, the data of 20 and 16 bit length can be read and written by continuously performing the I/O read or I/O write into the same address without changing the I/O address when the I/O access is performed on PAR and SAR.

In the same channel, because BSFF is cleared by accessing the I/O address which is different from the previous access, the register can be set with a minimum write down in the initialization.

7.6 Operation of KP27

General description

KP27 is a clock synchronous type DMA controller which operates a clock as a basic unit. The details of the device operation are as follows.

The following explanation refers to Figs. 7-4 through 7-6. In the following figures and explanation, the memory access signals and I/O access signals are assumed as the internal bus signals such as MRD_, MWR_, IORD_, and IOWR_. These signals are converted to the external bus cycle signals by the external bus interface unit. See Chapter 4 for more information of this conversion.

The basic unit of the KP27's operation is hereinafter referred to as "state." The KP27's operation basically consists of six states.

First, the period during which KP27 recognizes the bus acknowledge signal from CPU is state 0. State 0 almost corresponds to the period during which KP27 has no bus. During state 0, KP27 sets the initial-values of each internal register, etc. When DREQ signal is asserted after the channel enable, KP27 requests CPU to release the bus, and waits for the acknowledge .

When the bus request is accepted by CPU, KP27 moves to state 1. During this state, KP27 prepares the transfer of one byte, setting MRD_, MWR_, IORD_ and IOWR_ signals to "H," and DACK_ signal as "L." In continuously transferring plural bytes of data, at state 1 in the second and subsequent transfer, the value to the address bus, indicating how many bytes the channel has transferred since the channel enable, is output.

Following the falling edge of the next system clock, KP27 moves to state 2. During state 2, the data of the internal register C-PAR is output as address, and the access signals (MRD_, IORD_ and IOWR_) to the address are asserted. When the PALAT signal is "L," it should be "H." It should be noted that the data of C-PAR is output to the address bus when the I/O subject to transfer is specified by the DACK_ signal. In this case, it is recommended to specify unused I/O address which has no influence even when it is output to the address bus.

At the next falling edge, KP27 moves to state 3. During state 3, KP27 first sets the PALAT signal to "L." By this signal, the content of C-PAR output to the address bus are latched to the external latch circuit. KP27 thereafter outputs the data of the internal register C-SAR to the address bus, and asserts the access signals (MRD_ and MWR_).

In case that KP27 reaches TC (Terminal Count) by the transfer from state 1 to 3, it will enter the next stage, state 4. In case that KP27 has not reached TC, and is still under transfer, it will return to state 1. In case that KP27 has not reached TC, and the transfer is completed (DREQ negate or single transfer), it will enter state 5.

During state 4, KP27 performs the same operation as that during state 1, however, specifies the TC flag of SR0, setting the TC signal "H." The values then output to the address bus are those set at B-BCR as initial-values.

In case that Auto Initialize (described later) is not specified, or the transfer is complete (DREQ negate or single transfer), state 4 will become state 5 at falling edge of the next system clock. In case that Auto Initialize is specified and KP27 is still under transfer, state 4 returns to state 1 at falling edge of the next system clock.

During state 5, KP27 cancels the request of the bus, releases the bus and sets the DACK_ signal to "H." KP27 also outputs the data to the address bus, indicating how many bytes the channel has transferred since the channel enable. After CPU receives the bus after state 5, KP27 returns to state 0.

The above is the general flow of KP27 operation.

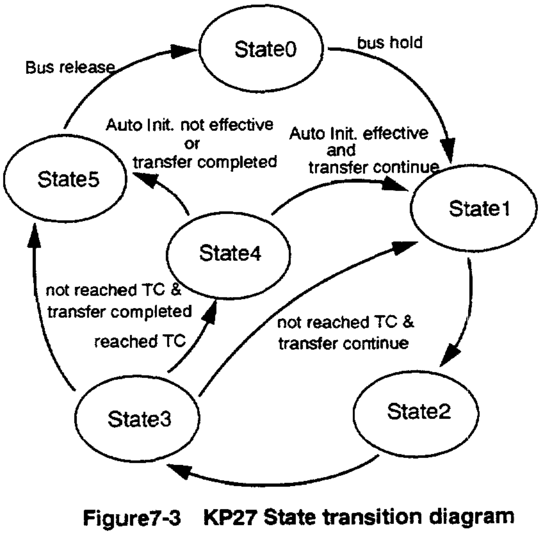

State0
bus hold
Bus release
Auto Init. not effective or transfer completed
Auto Init. effective and transfer continue
State5
State1
State4
not reached TC & transfer completed
reached TC
not reached TC & transfer continue
State3
State2

Figure7-3 KP27 State transition diagram

TC (Terminal Count)

Terminal Count means that the C-BCR register values are counted down to '0000H,' after the execution of transfer of some data. In case that a certain channel reaches TC, KP27 sets the TC signal of the channel to "H," and sets the bit 0 (TC flag) of the SR0 register.

Regardless of validity of Auto Initialize, TC flag is cleared by reading SR0 of the channel. Therefore, even after Auto Initialize, the condition of the TC flag is held until the SR0 read occurs. The TC flag of the channel can also be cleared by writing to the internal register excluding CR2 and NCC.

Auto Initialize

Auto Initialize, when it enters the TC condition, set again the initial-values of each base register at C-PAR, C-SAR and C-BCR of the subject channelso that the channel enable condition can automatically be continued.

For instance, in case that Auto Initialize is not selected in the demand transfer mode, because the channel is disable, the transfer is completed, although DREQ is of valid polarity, when KP27 reaches TC. On the contrary, in case that Auto Initialize is selected, the transfer can be continued because the initial-values of C-PAR, C-SAR and C-BCR are set again after KP27 reaches TC, and the channel continues to be enable, if the DREQ is of valid polarity.

Auto Initialize can be set by setting the bit 0 of the subject channel to '0' (Auto Initialize invalid) or '1' (Auto Initialize valid).

Valid polarity of KP27 DREQ

Setting of the valid polarity of the DMA request signal (DREQ) in each channel is possible by KP27.

It is possible to set the valid polarity at 'falling edge' or 'L,' and at 'rising edge' or 'H' by setting '0' and '1' at bit 2 of CR0 of the subject channel respectively.

In cases that the single transfer mode and demand transfer mode are specified, the above edge of DREQ becomes the DMA request, and the above level of DREQ becomes DMA request respectively.

Transfer mode of KP27

It is possible to specify the following two modes in each channel by KP27.

Setting of the transfer mode is valid when the transfer of memory → I/O or I/O → memory is set as transfer type (refer to the next page.), and neither single nor demand transfer mode cannot be selected when the transfer type of memory → memory is set. In this case, the DMA request is only software request (Refer to the section of CR1, Chapter 7.5), and the DMA transfer is continuously performed until KP27 reaches TC.

· Single transfer mode

This mode performs the DMA transfer on only one byte in response to one DMA request, and is specified by setting bit 1 of CR0 of the subject channel to '0.'

In this mode, the DMA request (DREQ) becomes a edge triggar type. For instance, if channel 0 is enable and if there is a DREQ on channel 0 during DMA transfer on channel 1, DMA of channel 0 is executed after the transfer of channel 1 is completed, because the existence of DREQ has been memorized there.

In the single transfer mode, the operations vary according to states in which the valid edges of DREQ occur.

First, when valid edges have not been available thus far during state 0, the device accepts DREQ (DREQ edge ① in Fig. 7-4), starting the DMA operation.

When valid edges have been available thus far during states 0 to 2 (DREQ edge ②, ③ and ④ in Fig. 7-4), the device cannot accept (memorize) DREQ.

In case that valid edges of DREQ have been available during state 3 (DREQ edge ⑤ in Fig. 7-4), the following transfers, although they are in the single transfer mode, can continuously be performed.

In case that valid edges of DREQ have been available during state 5 (DREQ edge ⑥ in Fig. 7-4), KP27 first releases the bus, however, the bus returns to KP27 again before CPU fetches the instructions, because KP27 requests the bus immediately after one system

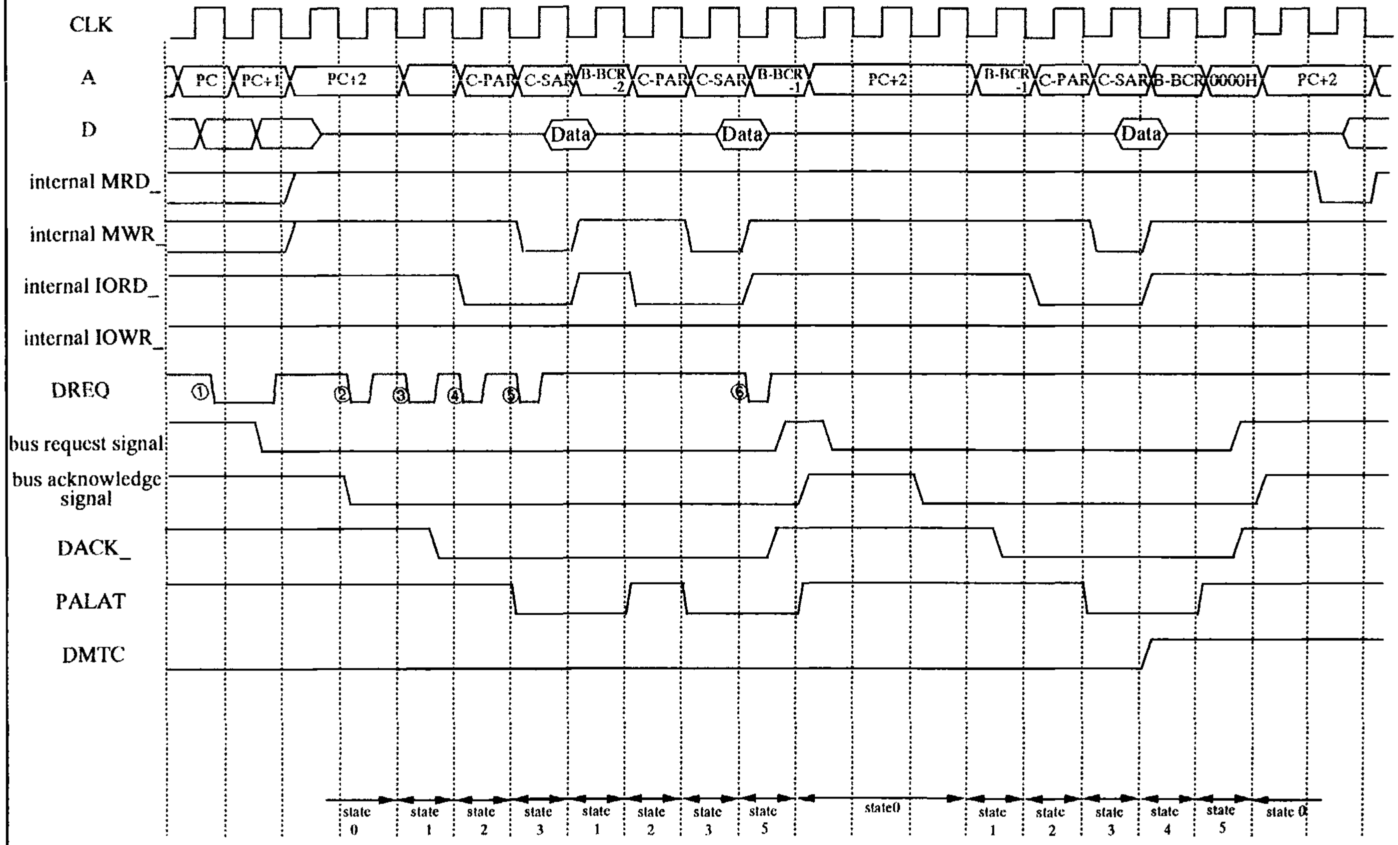


Figure7-4 I/O → memory transfer (single transfer mode, DREQ polarity " ↓ edge")

clock. KP27 then resumes the DMA operation (See Fig. 7-4).

· Demand transfer mode

This transfer mode continuously executes the DMA transfer while the DMA request (DREQ) is of valid polarity. In case that KP27 has reached TC while DREQ is valid, the operations depend on whether Auto Initialize is specified or not (Refer to the section of Auto Initialize). This transfer mode can be specified by setting bit 1 of CR0 of the subject channel to '1.'

In this mode, the DMA request (DREQ) becomes the level recognition. Namely, the DMA transfer of channel 0 cannot be executed if DREQ0 has been kept active until the transfer of channel 1 is completed, although DREQ0 becomes active during the transfer of channel 1.

This mode judges whether or not the DMA transfer should be executed according to DREQ of the valid polarity at falling edge of the system clock which moves to state 1 or 5. Namely, the following transfer is executed if DREQ is then valid, and is not executed if DREQ is then invalid (See Fig. 7-5).

Transfer type of KP27

KP27 can specify the following three transfer types in each channel.

· Transfer of memory → I/O

This type transfers the data from memory to I/O, and is specified by setting D<5:3> of CR0 of the subject channel to '010' (memory address increment) or '011' (memory address decrement). The I/O address cannot be changed.

The I/O address of the data destination and the memory address of the data source are set at B-PAR and B-SAR respectively. A number of transfer byte is set at B-BCR. When each base register is set, the values are simultaneously written into each current register. In case that the I/O subject to transfer is specified by the DACK_ signal, it should be noted that B-PAR requires some unused I/O address on the system.

This transfer is started by the DMA request from the DREQ signal. Here, the single transfer mode and demand transfer mode are available.

In the transfer of memory → I/O, immediately after the transfer is started, the device first asserts the DACK_ signal, changes the PALAT signals to the data-through polarity ("H") while outputting the I/O address specified with the C-PAR values, and starts the I/O write access by asserting the IOWR_ signals. The device further changes the PALAT signals to the data latch polarity ("L"), and through this, latches the I/O address, etc. at the register (latch, etc.). It then outputs the memory address specified by the C-SAR values, asserting the MRD_ signals.

Thus, the device directly writes down the memory-read data into I/O, transferring one bite (See Fig. 7-5).

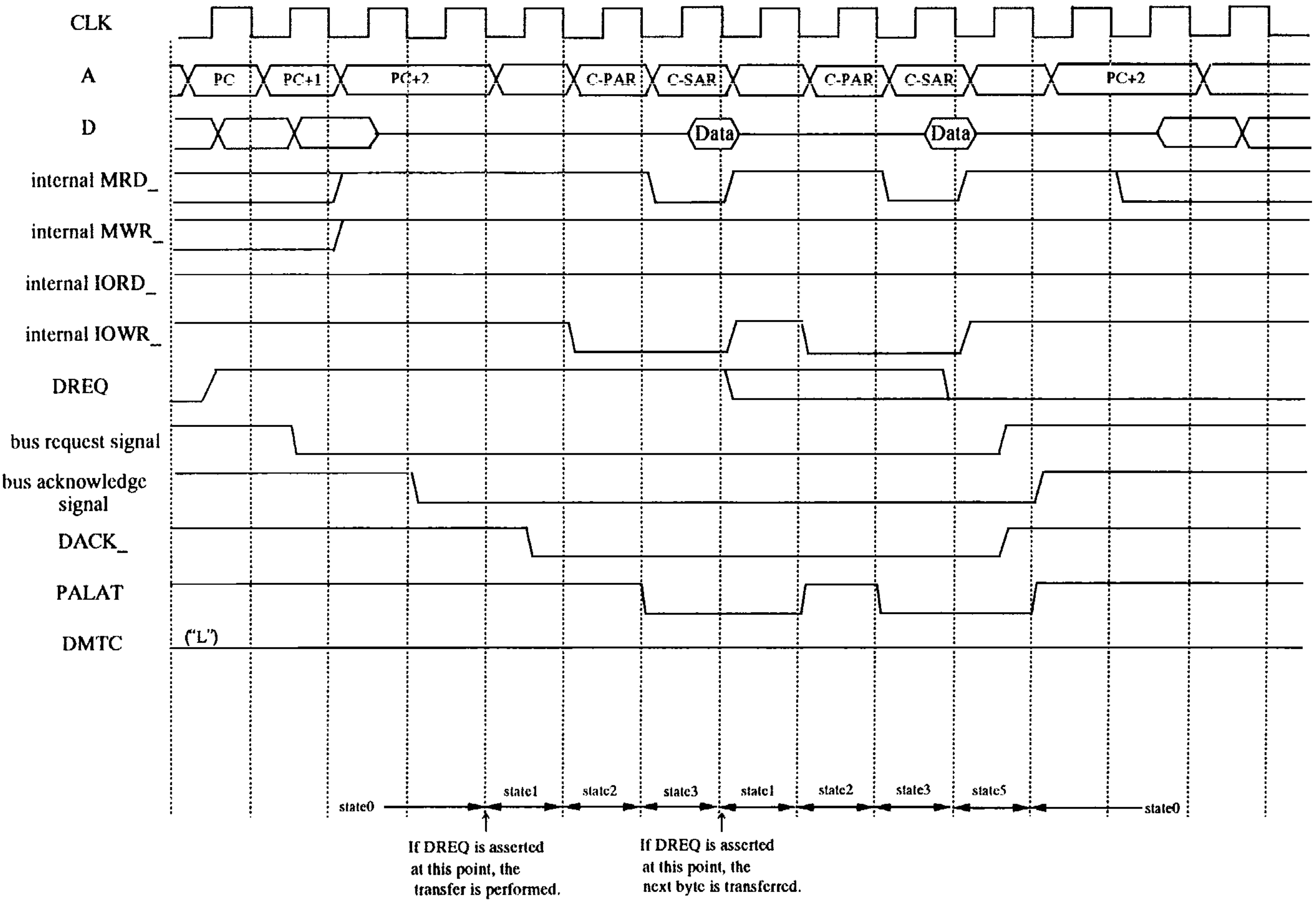
· Transfer of I/O → memory

This type transfers the data from I/O to memory, and is specified by setting D<5:3> of CR0 of the subject channel to '100' (memory address increment) or '101' (memory address decrement). The I/O address cannot be changed.

The I/O address of the data source and the memory address of the data destination are set at B-PAR and B-SAR respectively. A number of transfer byte is set at B-BCR. When each base register is specified, the values are simultaneously written down into each current register. In case that the I/O device subject to transfer is specified by the DACK_ signal, it should be noted that the values set at B-PAR must not overlap the other I/O address.(a unused I/O address on the system)

This transfer is also started by the DMA request from the DREQ signal. The single transfer mode and demand transfer mode are available.

In the transfer of I/O → memory, immediately after the transfer is started, the device first asserts the DACK_ signal, changes the PALAT signals to the data-through polarity ("H") while outputting the I/O address specified with the C-PAR values, and start the I/O read



If DREQ is asserted at this point, the transfer is performed.

If DREQ is asserted at this point, the next byte is transferred.

Figure7-5 memory → I/O transfer (demand transfer mode、 DREQ polarity "H") access by asserting the IORD_ signals. The device further changes the PALAT signals to the data latch polarity ("L"), and through this, latches the I/O address, etc. to the register (latch, etc.). It then outputs the memory address specified with the C-SAR values, asserting the MWR_ signals.

Thus, the device directly writes down the I/O-read data into memory, transferring one bite (See Fig. 7-4. In this figure, the edge timing of the DREQ and the execution of the DMA transfer are also shown. See the section of the single mode transfer.).

- **Transfer of memory → memory**

This type transfers the data from memory to the other address on the memory, and can be specified by setting D<5:3> of CRD in the subject channel at '000' (memory address increment) or '001' (memory address decrement). D <2:1> of CR0, namely, the DREQ polarity and setting of the transfer mode are then neglected. Although Auto Initialize also becomes invalid, D0 of CR0 should be set to '0.'

The memory address in the data transfer source and that in the data transfer destination should be set to B-PAR and B-SAR respectively. A number of transfer byte should be set at B-BCR. These values are simultaneously written down into each current register when they are set at each base register.

The transfer of memory → memory is started only by the software request. The software request is performed by the command-write to CR1. The bus cannot be returned to CPU until it reaches TC, excluding the case in which the transfer is discontinued by NMI. The address can select the increment for both C-PAR and C-SAR, or can select the decrement for both C-PAR and C-SAR.

This type first performs the memory read of the address specified with the C-PAR values when the transfer is started, and, while holding the read data in KP27, transfers one byte by performing the memory write on the address specified with the C-SAR values. (See Fig. 7-6. In this figure, the bus cycle waited by the bus cycle wait signals when the second byte is transferred is also shown).

Channel enable and disable

Each DMA is enabled/disabled by CR1. The channel is enabled by setting CR1 bit 5 to '1,' and is disabled by setting it to '0.' On the contrary, in the transfer of memory → memory, it becomes the software request by setting the bit 5 to '1.'

The channel is disabled when the internal register is written down in the enabled channel. It is therefore necessary to enable the channel again to execute the DMA transfer. Reading of the internal register has no influence on the enable condition. The single transfer request, software request, etc., for the disabled channel are not memorized.

Priority of DMA request and external DMA device

Because KL5C80A16 has two DMA channels, it is possible to arbitrate and perform the DMA transfer request from two different request sources. The priority of two DMA channels is assigned and channel 1 > channel 0.

Therefore, for instance, when DREQ1 of the demand transfer mode goes into an effective polarity during the transfer of memory → memory of channel 0 , the device discontinues the transfer of channel 0, and starts to process DREQ1. On the contrary, when DREQ1 goes out from an effective polarity, the device resumes the transfer of channel 0, and returns the bus to CPU when the channel reaches TC.

KP27 also has functions to connect the device with the DMA function as the external DMA device, arbitrating the bus requests of KP27 DMA channels and the external DMA device which is connected to KP27. This can be realized by connecting the bus request signals of the external DMA device to the EXBREQ_ pin, or connecting EXBACK_ signals to the pin of the bus request recognition signals of the external DMA device. For your reference, effective polarity of both EXBREQ_ and EXBACK_ is "L." EXBREQ_ is of the level recognition.

It is possible to arbitrate the relations of three DMA

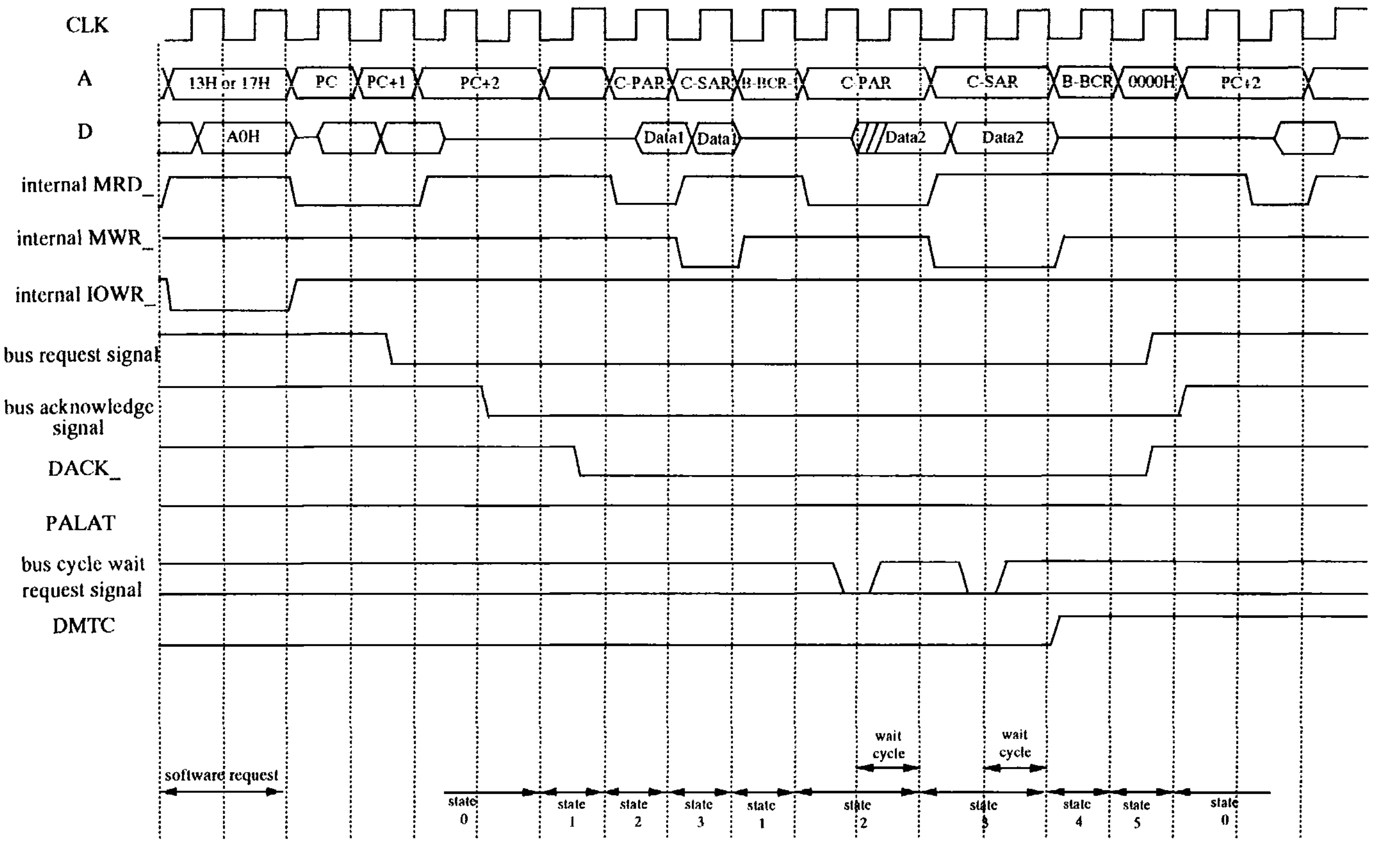
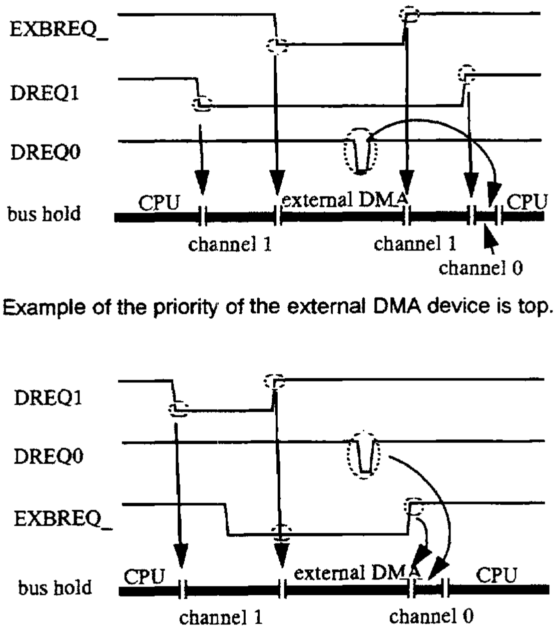


Figure7-6 memory → memory transfer

requests, DREQ0, DREQ1 and EXBREQ_. However, the priority of the external DMA device becomes either higher (top) or lower (bottom) than that of DREQ0 and DREQ1. This setting is performed by D <4:3> of CR2. The external DMA device is set at high priority (top) by setting this to '11,' and is set at low priority (bottom) by setting this to '10.' When the external DMA device is not connected, D4 of CR2 should be set to '0.' The values of D3 are then neglected.



Example of the priority of the external DMA device is top.

Example of the priority of the external DMA device is bottom.

DREQ1: demand transfer, "L" effective
DREQ0: single transfer, "↓"effective

Figure7-7 Examples of bus hold and release in each case of external DMA priority

In case that the priority of the external DMA device is set high (top):

KP27 temporarily discontinues the transfer after the transfer of a certain one byte is completed during state 1, and processes the external DMA, when the external DMA request occurs during processing of DREQ0 or DREQ1. KP27 resumes DREQ after the processing of the external DMA transfer is completed.

KP27 starts to perform DMA request on DREQ signals after the processing of the external DMA is completed, in case that DMA request on DREQ0 or DREQ1 exists during execution of the external DMA and that they are effect (demand transfer mode) or they are memorized (single transfer mode) when the execution of the external DMA is completed.

In case that the priority of the external DMA device is set on the bottom:

KP27 accepts the external DMA if the external DMA exists when the DMA transfer on DREQ0 and DREQ1 are completely finished.

KP27 starts to perform DMA transfer on the valid DREQ after the external DMA transfer is completed when DREQ0 or DREQ1 occurs during processing of the external DMA transfer.

In any case, when processing of the external DMA transfer is started, the bus does not return to neither KP27 nor CPU until it is completed. Therefore, if the external DMA occupies the bus for a long time, adequate CPU and KP27 operations might be prevented.

Discontinuation of DMA by NMI (NMI Abort)

KP27 has function to discontinue the DMA transfer. This function should be set with bit 2 of CR2. The function becomes valid by setting this to '1,' and becomes invalid by setting this to '0.'

In case that the NMI_ signals are used for system emergency, this function prevents the occasion in which the immediate NMI processing is impossible due to the fact that the DMA controller has the bus, although the NMI request occurs.

In case that the falling edge of the NMI signals occurs while KP27 is performing the DMA transfer keeping the bus, KP27 sets the internal NMI flag, returning the bus to CPU at the interval between transfers (a point at which a certain one byte is completed). The internal registers of each DMA channel then hold those values.

KP27, at falling edge of the system clock on the boundary between state 3 and the following state, judges whether or not the bus should be returned to CPU by discontinuing the DMA transfer in order to process NMI. Namely, the DMA transfer is discontinued after the current transfer is completed, when falling edge of the NMI_ signal exists before the above boundary, however, the DMA transfer is discontinued, and the bus is returned to CPU after the transfer of the following one byte is executed, when the NMI falling edge exists after the above boundary. (Fig. 7-8 shows the timing chart from DMA transfer to NMI processing.)

CPU executes the NMI routine after receiving the bus, however, it is absolutely necessary in the last in NMI routine to write down NCC mentioned in Chapter 7.5, because the NMI flags can be set by NMI_ edge not in the DMA transfer when the NMI Abort function is set effective.

KP27 requests the bus again by automatically asserting the bus request signals one bus cycle + 1.5 CLK after falling edge of the clock which performs the I/O write of NCC, and then resumes the DMA transfer which has been discontinued by the NMI_ signals.

It is recommended to use the NMI Abort function by programming the following NMI routine. Fig. 7-9 shows the point at which DMA is resumed based on the program.

```
org     0066H
PUSH    AF                  {F5H}
          .
          .
          .
LD      A,      F0H         {3EH  F0H}
OUT     (13H),  A           {D3H  13H}
POP     AF                  {F1H}
RETN                        {EDH  45H}
```

The characters inside {} is instruction codes.

In this case, because KP27 requests the bus again while CPU is executing the POP AF instruction, CPU, after the execution of the POP AF instruction, releases the bus without execution, while fetching the first byte of the RETN instruction ('EDH'). Therefore, CPU first executes the RETN instruction when the resumed transfer is completed and the bus is returned to CPU.

Also, the NMI Abort is executed again during the resumed transfer, the transfer is discontinued and the bus is returned to CPU. However, the nest of NMI does not occur because CPU then fetches the second byte of the RETN instruction ('45H') whose first byte has been fetched, and enters the NMI routine after the execution of the RETN instruction. (See Fig. 7-10)

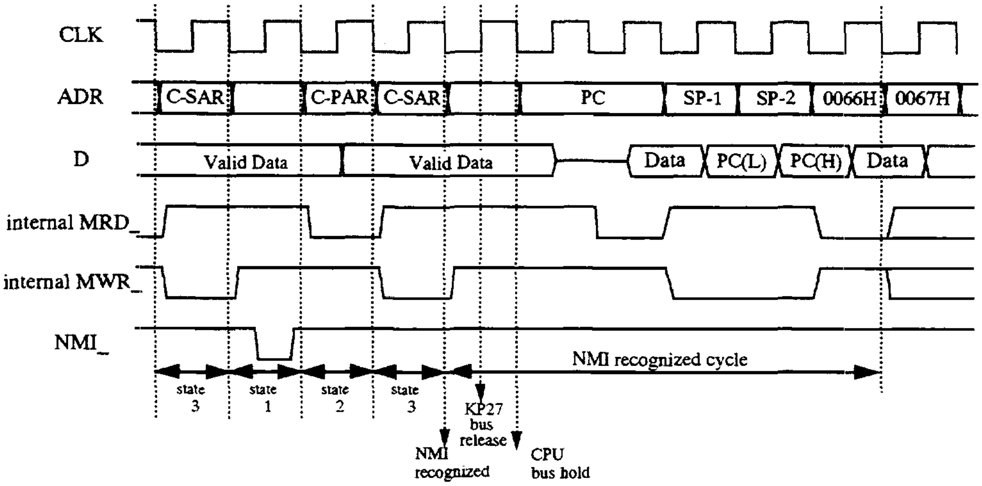


Figure7-8 DMA transfer discontinue → NMI routin start

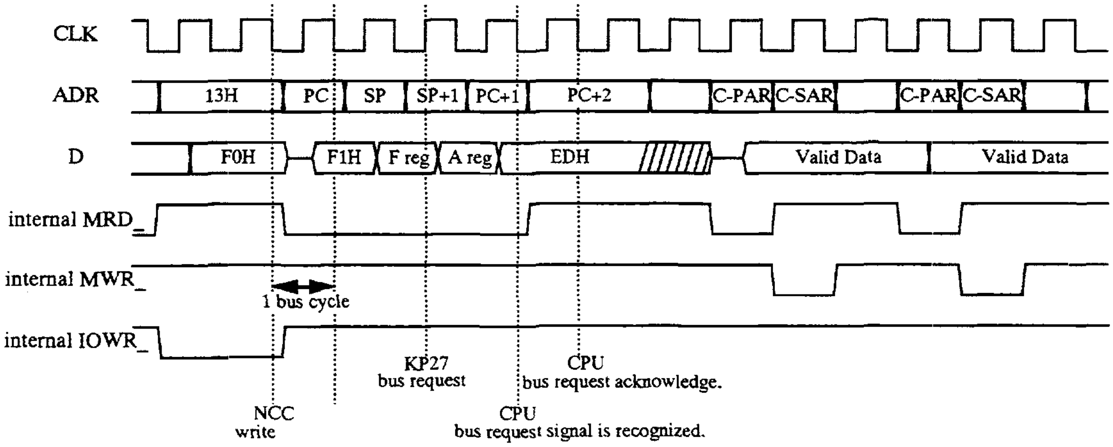


Figure7-9 NMI process routine ends by the program example → DMA transfer restarts.

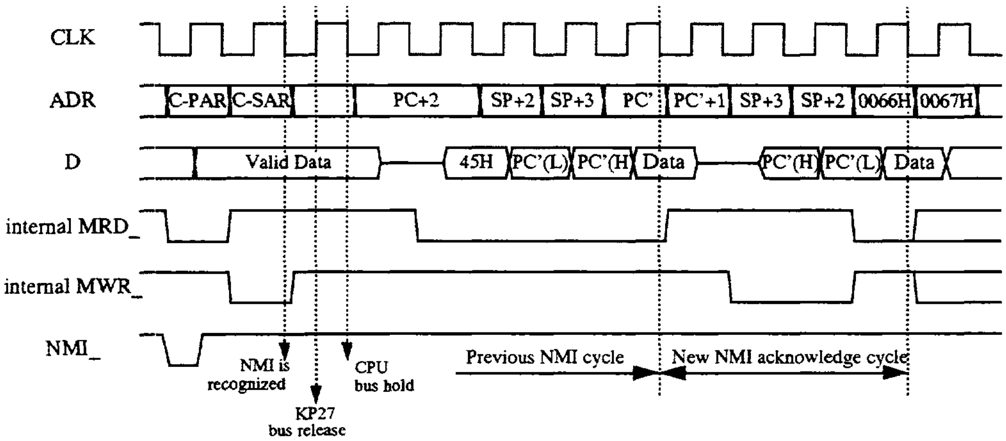


Figure7-10 DMA transfer restarts by the program example → NMI routine starts.(continued from Figure7-9)

Discontinuation by NMI and external DMA

In case that the external DMA device requests the bus while processing NMI which occurred when CPU was a bus master, KP27 recognizes that the EXBREQ_ signals have become valid polarity, requesting CPU to release the bus.

On the other hand, in case that the external DMA device requests the bus while processing NMI which occurred when KP27 was a bus master (during the DMA transfer), the bus moves to the external DMA device even during the execution of the NMI routine, only when the priority of the external DMA device is high (top).

As mentioned above, if the external DMA device once achieves the bus, it returns neither to KP27 nor CPU until the device releases the bus. Therefore, CPU does not execute processing of NMI until the external DMA device releases the bus, even when the NMI request then occurs.

Considering these, designers should take care in using the external DMA device which might occupy the bus for a long time.

Wait in bus cycle

In order to meet the operation with the slow memory and I/O, it is possible to insert waits into the bus cycle by inputting the wait signal into KP27. The bus cycle wait signal is connected to the ERDY signals which are input from the pin 6 of KL5C80A16, and are synchronized with the system clock through the external bus interface unit.

In case that the bus cycle wait signal shows "L" at falling edge of the system clock, the bus cycle which starts at the falling edge of the system clock can be recognized as the wait cycle.

However, the wait is only possible during the period of:

- States 2 and 3 for the transfer of memory → memory
- State 3 for the transfer of memory → I/O and I/O → memory

(See Fig. 7-6)

See Chapter 4 for details of output of external bus signal in case of wait cycles.

Reset

When the RESET_ pin is set to "L" level, KP27 is operated as follows.

(1) All the bits of all the internal registers are cleared to "0." Namely, PAR, SAR and BCR are all set to '00H,' and each item is then set as follows.

transfer type : Memory → memory

Address increment

DREQ polarity : Valid polarity "↓ edge" or "L:"

transfer mode : Single

(Setting of the above two are invalid in the transfer of memory → memory.)

Auto Initialize : Invalid

Each channel : Disable

External DMA : Unused

External DMA priority : Bottom

(This setting is invalid because the external DMA is unused.)

Discontinuation by NMI : Invalid

Therefore, it is necessary to specify each register in order to operate DMA after the reset.

(2) The DMA transfers in operation are all discontinued.

7.7 Precautions

The DACK_ signals of the channel which execute the DMA operation become valid polarity ("L") regardless of channel's transfer type setting. It should also be noted that the I/O address and PALAT signals can be output even when the DACK_ signals are used in order to specify the I/O subject to DMA transfer. The I/O address which has no influence i.e.(the unused I/O address) should be then set at B-PAR.

8. UART

8.1 General Description

The UART on this chip is a KP61 macro cell. In this macro cell, the asynchronous serial port comprises two internal channels which transmit and receive serial data in accordance with instructions from the CPU. Independent interrupt is possible with both two channels when transmitting or receiving, and the transmitting side or receiving side of channel 0 can be connected internally to channel 1 of the internal DMA by specifying SCR2.

Features

- The device is equipped with a buffer register which transmits and receives independently, and performs full-duplex communication.
- Two communication channels are provided.
- Character lengths of 7, 8, and 9 bits can be selected.
- It is possible to select 1 or 2 as the stop bit.
- It is possible to select addition (even number/odd number) or non-addition of a parity bit. However, a parity bit cannot be added when the character length is 9 bits.
- A common Baud Rate generator is provided for channel 1 and 2 as a transmit/receive clock source. It is possible to set the Baud Rate independently for each channel.
- It is possible to select an internal or external transmit/receive clock.The transmit clock and receiving clock are used in common.
- The sampling clock has a clock speed of 16 x the serial data bit rate.
- By using the sleeping mode, communication with multiple micro controllers is possible.

8.2 Block Diagram

A general block diagram of the KP61 is shown in the following figure.

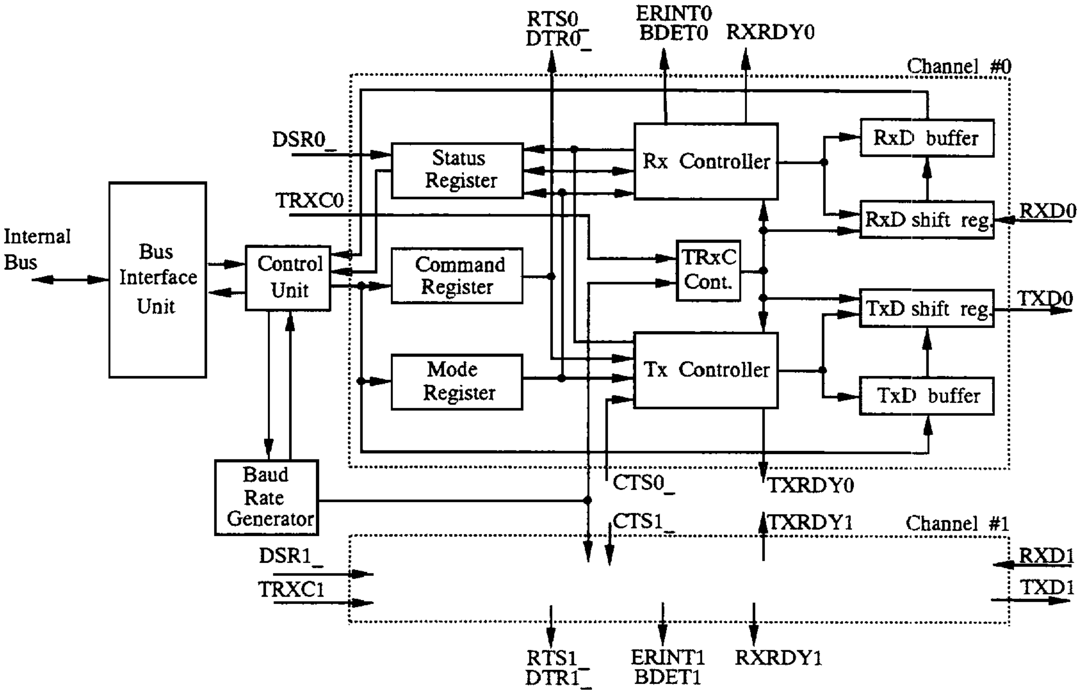


Figure 8.1 Block diagram of KP61

Channel 1 has the same configuration as channel 0 in Figure 8-1.

Among Channel 0 signals, TXRDY0 is connected internally to IR[9] of the interrupt controller, RXRDY0 is connected to IR[10], and ERINT0+BDET0 is connected to IR[11]. DSR0_ is fixed internally to "L", and DTR0_ does not output outside of the KL5C80A16. Of the other channel 0 signals, the external pin of KL5C80A16 is assigned as a dedicated pin.

Among channel 1 signals, TXRDY1 is connected internally to IR[3], and RXRDY1 is connected to IR[4]. It is also possible to connect ERINT1+BDET1 internally to IR[5] by specifying SCR1. All channel 1 signals other than ERINT1 and BDET1 are used in multiplexed with parallel ports. For a detailed description of the use of these signals, refer to Chapter 12.

By specifying SCR2, it is possible to use either TSRDY0 or RXRDY0 of channel 0 as a DMA request for the channel 1 DMA controller. See Chapter 12 for further discussion.

8.3 Pin Description

The pin names 0 and 1 represent channel 0 and 1.

Pin name	I/O	Function
TRXC0 TRXC1	I	transmit/receive clock input pin Pin which inputs x16 clock when transmitting or receiving data. Used as a transmit/receive clock when selecting an external input.
RXD0 RXD1	I	receive data input Pin which inputs external serial data. Received data is sampled three times around the center of the bit, and majority value of sampled data is entered into the receive shift register.
RXRDY0 RXRDY1	O	Receive-ready signal output Outputs "H" indicates data receive is finished and received data can be read. This is reset to "L" by data read out. RXRDY0 is connected to IR[10] internally in the KL5C80A16. Use as DREQ1 for the DMA controller is also possible. (See Chapter 12.) RXRDY1 is connected to IR[4] internally in the KL5C80A16.
TXD0 TXD1	O	Transmit data output Pin which outputs transmit data in serial form. Data transmitting becomes possible by CTS_="L" and transmit enable (Command Register bit 0 = "1"). This pin outputs serial data bit in LSB first order every 16 cycles of the transmit/receive clock at the falling edge of the transmit/receive clock. During transmission, when either CTS_="H" or transmitting is disabled (control register bit 0 = "0"), the pin enters the marking condition (transmit data = "1") following the completion of the transmitting of the data in the transmitting buffer, and the transmitting operation is suspended. "L" is output from TXD on writing of a transmit break command (Command Register bit 3 is set to 1). The break condition continues until the transmit break is canceled (Command Register bit 3 is set to "0").

Pin name	I/O	Function
TXRDY0 TXRDY1	O	Transmit-ready signal output The KP61 has double transmit buffer(TxD shift reg. and TxD buffer) which allows writing next transmit data after beginning transmitting data. TXRDY goes "H" when KP61 is in the transmit enable condition(CTS_="L" and transmit enable bit in Command Register(D0) is "H") and transmit data write is allowed. TXRDY goes "L" at transmit data write and keeps "L" until KP61 goes into the transmit enable condition and transmit data write is allowed. This pin differs from the Status Register pin (TXRDY flag) in the following: TXRDY bit: It goes "H" when transmit data write is allowed. TXRDY pin: It goes "H" when transmit data write is allowed and in the transmit enable condition. TXRDY0 pin signal is connected to IR[9] internally in the KL5C80A16, and can also be used as a DREQ1 of the internal DMA controller. (See Chapter 12) TXRDY1 pin signal is connected to IR[3] internally in the KL5C80A16.
CTS0_ CTS1_	I	Clear-to-send signal input Pin which inputs external transmit request signals. Normally connects modem Clear-to-send signals. If data is present in the transmit shift register under transmit enable condition (Command Register bit 0="1") and CTS_="L", transmit data is output from the TXD pin. Transmission is not discontinued even if a CTS_="H" condition occurs during a data transmission. In this case TXD is into marking condition(TXD keeps "H" state) after the completion of the transmission of data. The reverse of the value of this pin is reflected on bit 6 of Extended Status Register B.
DSR0_ DSR1_	I	Data-set-ready signal input Normally, connects the Data-Terminal-Ready signal of the modem to check the status of the modem, but can also be used as a general purpose input port. The reverse of the value of this pin is reflected on bit 7 (DSR) of the Status Register. DSR0_ is internally pulled down to GND.
RTS0_ RTS1_	O	Request-to-send signal output Normally used as a Request-to-send signal of the modem, but can also be used as a general purpose output port. The reverse of the value set in bit 5 (RTS) of the Command Register is output.

Pin name	I/O	Function
DTR0_ DTR1_	O	Data-terminal-ready signal output Normally used as a Data-Terminal-Ready signal of the modem, but can also be used as a general purpose output port. The reverse of the value set in bit 1 (DTR) of the Command Register is output. Note: DTR0_ is not connected to KL5C80A16 pin.
ERINT0 ERINT1	O	Error interrupt request signal output Outputs "H" on the occurrence of one or more of the following: parity error, framing error, overrun error. This signal can be used as an interrupt request signal.
BDET0 BDET1	O	Break detect signal output Output pin for signal indicating the detection of a break state. KP61 recognize 8 bit '0' in a low after detection of stop bit='0' on TxD pin as break condition and make this signal "H". This state continues until the next time '1' is detected on the RXD.

8.4 Internal register mapping and setting method

A mapping of the KP61 internal register is shown in the following table.

Table 8-1 I/O Register mapping

I/O address	Write cycle	Read cycle
28H	RATE data	RATE data
29H	Reserved	Reserved
2AH	Channel 0 transmit data	Channel 0 receive data/Extended Status A
2BH	Channel 0 mode/command	Channel 0 status/Extended Status B
2CH	Channel 1 transmit data	Channel 1 transmit data/Extended Status A
2DH	Channel 1 mode/command	Channel 1 status/Extended Status B
2EH	Reserved	Reserved
2FH	Reserved	Reserved

Initialization and Read sequence

After reset, the mode should be specified, and the command should be written down according to the following steps, before transmitting and receiving the data by KP61. The read sequence of the Status Registers is also described in the following in detail.

Mode programming

Data Written to Mode/Command address('2BH' and '2DH') followed by hardware reset or Command write software reset is recognized as mode data. Here, the character length, parity, etc. are specified. See Chapter 8.5 for mode register bit definition.

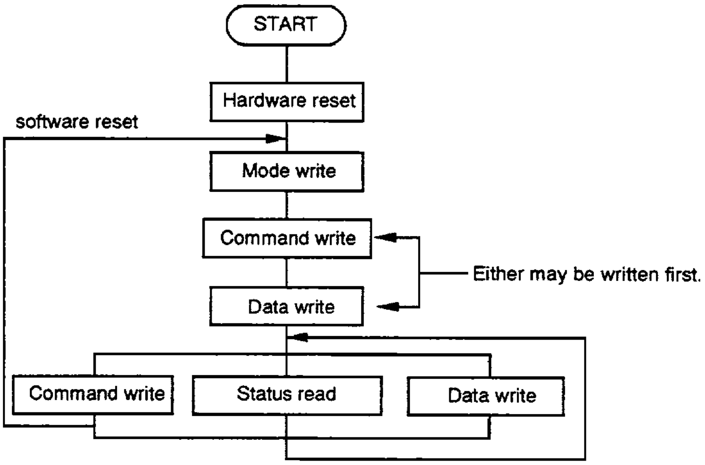


Figure 8-2 initialization and read sequence

Command instruction

Once mode programming is finished, data written to I/O address, '2BH' and '2DH' is recognized as Command data. Transmitting and receiving of the data become possible through this command setting. Three types of Command Register are available. The command should be specified according to the format in Chapter 8.5.

Switchover to readout register

The method to specify registers in read operation is described using channel 0 as example.

After reset, the received data and status of channel 0 can be read out, when the data is read out from 2AH and 2BH of the I/O address. The register which can be read out from 2AH and 2BH of the I/O address becomes the Extended Status Register following the command write with bit 0 = '1' to I/O address 2BH and the setting of the register switchover F/F. See Chapter 8.5 for each format of the Extended Status Register A and B.

The register switchover F/F cannot be cleared by reading out the Extended Status Register A. Therefore, the Extended Status Register B can sequentially be read out. Because the register switchover F/F is cleared when the Extended Status Register B is read out, the received data and Status Register are consequently read out in sequential read. The register switchover F/F can be cleared by write down the register switchover command with bit 0 = '0.'

So with channel 1. Switchover of this readout register is performed between the channels independently. However, it should be noted that the register switchover F/Fs are simultaneously cleared in channels 0 and 1 by readout of the Extended Status B.

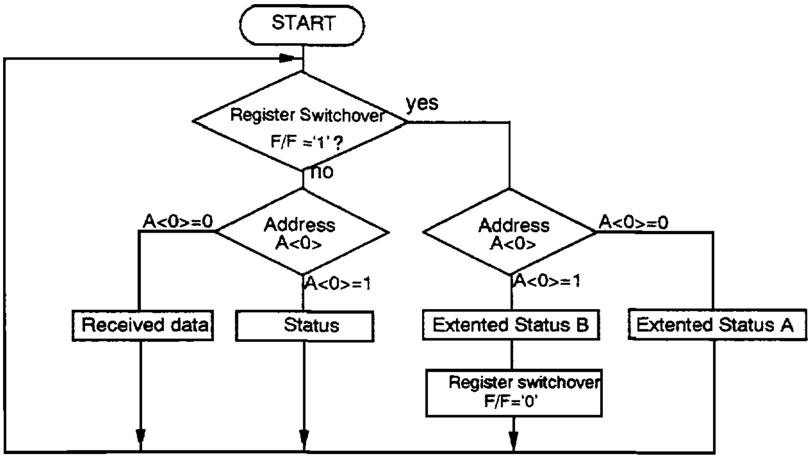
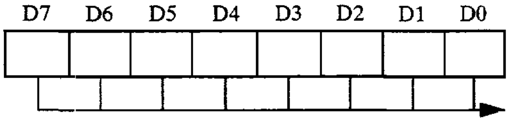


Figure 8-3 Register read sequence

8.5 Internal register architecture

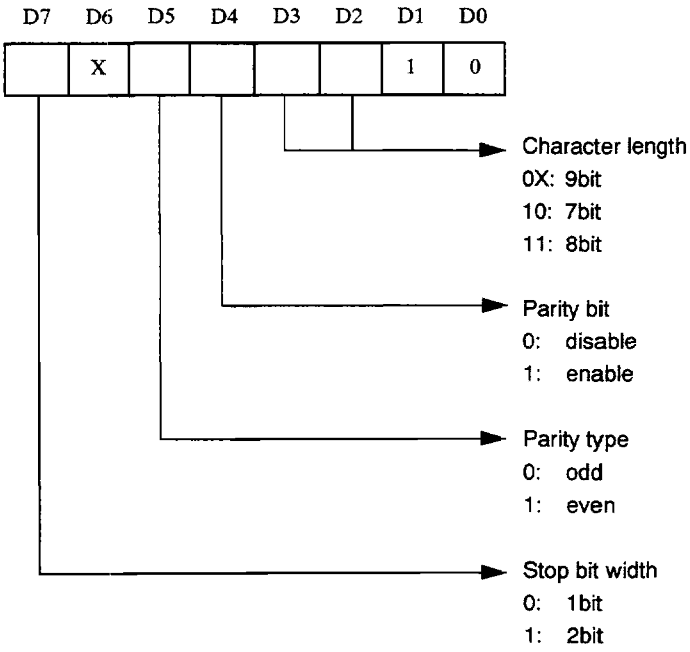
Baud Rate register

This register specifies the parameter RATE <7:0> which determines the operations of the internal Baud Rate generator. Refer to the section, 'Baud Rate generator and selection of transmit/receive clock' in Chapter 8.6.



Baud Rate generator parameter, 'RATE' value

Mode Register



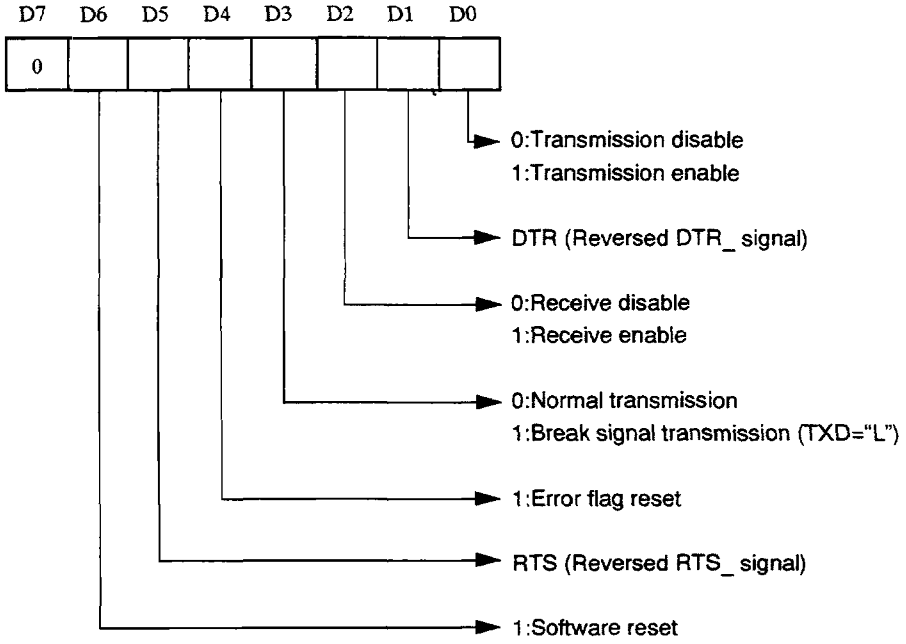
Character length
0X: 9bit
10: 7bit
11: 8bit

Parity bit
0: disable
1: enable

Parity type
0: odd
1: even

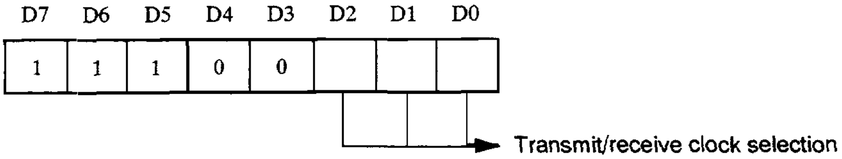
Stop bit width
0: 1bit
1: 2bit

Command Register A



Because D6 and D4 are cleared by completion of operation, it is not necessary to write down '0' to cancel the operation.

Command Register B



D2	D1	D0	Transmit/receive clock
0	0	0	External clock (from TRXC PIN)
0	0	1	BCK<0>
0	1	0	BCK<1>
0	1	1	BCK<2>
1	0	0	BCK<3>
1	0	1	BCK<4>
1	1	0	BCK<5>
1	1	1	BCK<6>

Refer to the section 'Baud Rate generator and selection of transmit/receive clock' in Chapter 8.6 for BCK <6:0>.

Command Register C

D7	D6	D5	D4	D3	D2	D1	D0
1	1	0	0	0	0		

D0 → Transmission data bit 8
(This bit is effective only when the character length is 9 bits.)

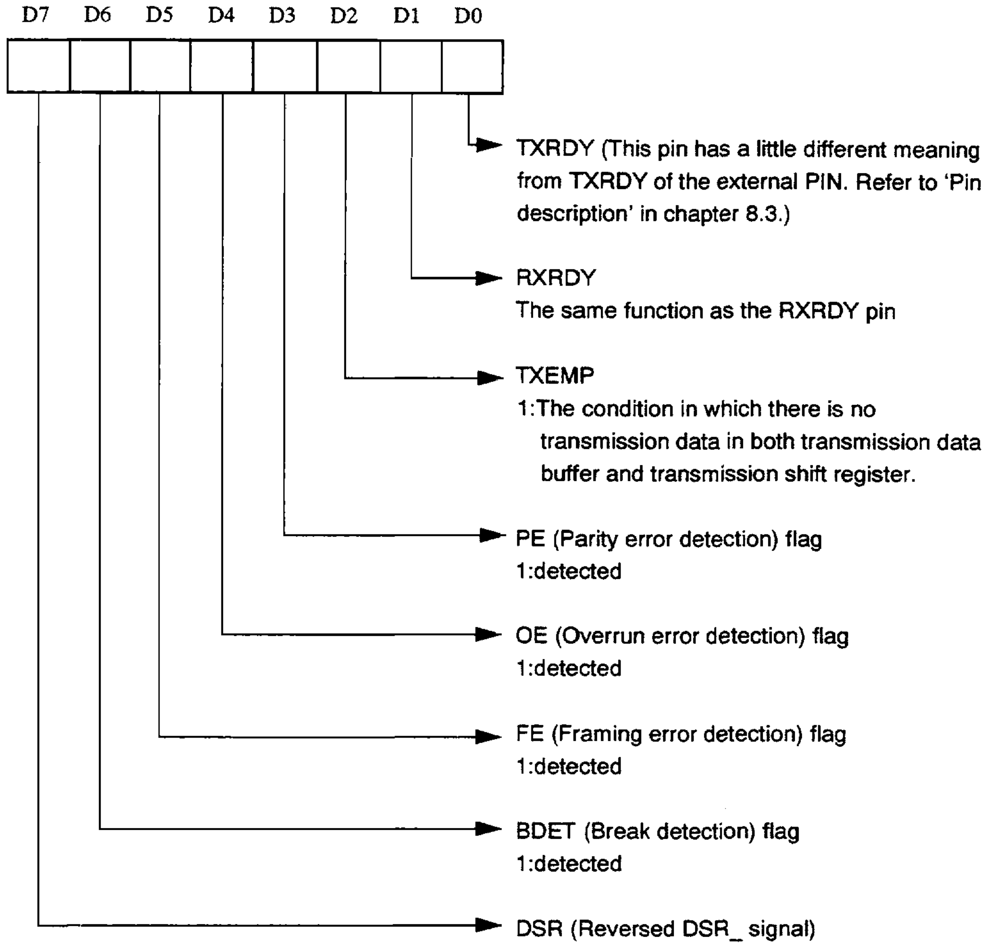
D1 → Sleep mode
0:reset
1:set

Register switchover F/F clear command

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	

D0 → Register switchover F/F
0:clear
1:set

Status Register



TXRDY (This pin has a little different meaning from TXRDY of the external PIN. Refer to 'Pin description' in chapter 8.3.)

RXRDY
The same function as the RXRDY pin

TXEMP
1:The condition in which there is no transmission data in both transmission data buffer and transmission shift register.

PE (Parity error detection) flag
1:detected

OE (Overrun error detection) flag
1:detected

FE (Framing error detection) flag
1:detected

BDET (Break detection) flag
1:detected

DSR (Reversed DSR_ signal)

When the receive is disabled (the Command Register A bit 0 = '0'), PE, OE, and FE flags aren't set. When the error flag is set, the error flag is not cleared even if the receive is disabled. The error flag is cleared only at the error reset (the Command Register A bit 4 is '1') or the hardware/software reset.

Extended Status Register A

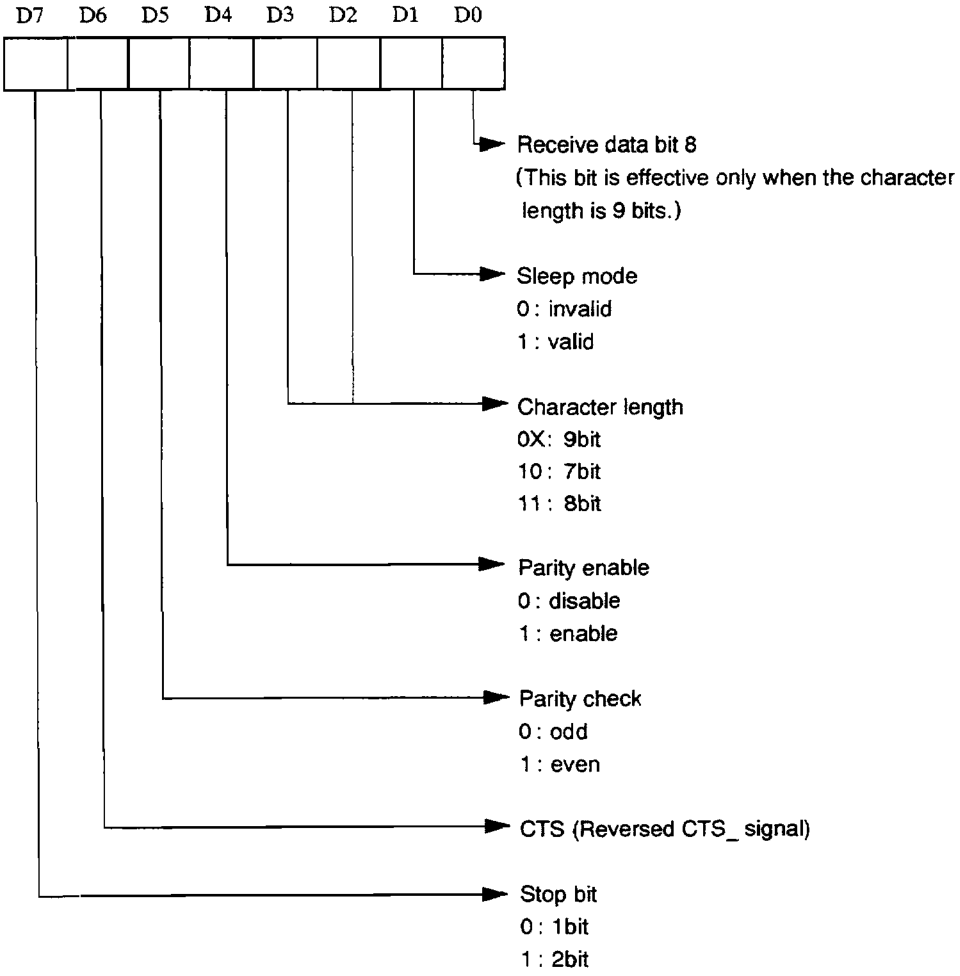
D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0				

Transmit/receive clock selection

D2	D1	D0	Transmit/receive clock
0	0	0	External clock (from PIN)
0	0	1	BCK<0>
0	1	0	BCK<1>
0	1	1	BCK<2>
1	0	0	BCK<3>
1	0	1	BCK<4>
1	1	0	BCK<5>
1	1	1	BCK<6>

Reserved

Extended Status Register B



Receive data bit 8
(This bit is effective only when the character length is 9 bits.)

Sleep mode
0 : invalid
1 : valid

Character length
0X: 9bit
10 : 7bit
11 : 8bit

Parity enable
0 : disable
1 : enable

Parity check
0 : odd
1 : even

CTS (Reversed CTS_ signal)

Stop bit
0 : 1bit
1 : 2bit

8.6 Operations

Transmission

When the transmit data is written down from CPU, the data is set in the transmit data buffer. "H" is continuously output from the TXD pin (marking condition), if the device is not in the break signal transmit condition, and is in the transmit disable condition.

When CTS_ = "L," and the transmit enable (Bit of the Command Register: bit 0 = '1') condition (referred to as 'transmit enable condition'), KP61 puts the data in the transmit data buffer into the transmit shift register, starting to transmit the data. TXRDY flag of the Status Register is also set to '1'. The above operation is reflected on the TXRDY pin.

KP61 checks whether or not transmit is possible every time the data transmit is completed in the transmit shift register. If the subsequent transmit data exists in the transmit data buffer and KP91 is in transmit enable condition, KP61 sets the data at the transmit shift register and start transmitting. While the transmit data does not exist, the transmit operation is not performed, and KP61 holds the marking condition and sets the TXEMP flags of the Status Register to '1'. Moving into transmit disable condition under transmission does not affect the operation. As mentioned above, KP61 checks whether or not the data is transmittable when the transmission is completed, and discontinues the transmission in transmit disable condition.

The TXD output becomes "L" by setting the break condition (bit 3 of the Command Register = '1'). This break condition continues until the transmit break is canceled (Bit 3 of the Command Register should be set to '0'). However, during this period, the transmission and TXRDY output are performed at the same timing as in the usual transmit condition according to the specified character length, etc.

Receive

KP61 receives the data according to the specified mode in the receive enable condition (bit 2 of the Command Register = '1'). The device waits for the start bit when "H" is detected from the RXD pin. Thereafter, RXD is sampled at seventh, eighth and ninth clocks of the transmit/receive clock after RXD = "L". KP61 recognized the occurance of start bit on RXD pin if "L" is detected two times and over (See Fig. 8-7). However, KP61 goes back into the start bit waiting condition again if "L" is detected one time and below.

After recognizing the start bit, KP61 enters the data bit in the receive shift register. So with the recognition of start bit, KP61 samples RXD at seventh, eighth and ninth clocks of the transmit/receive clock in a certain receive bit. The values which are detected two times and over are established as data of the bit.

When the stop bit is detected after receiving the specified number of character bits and parity bit, the receive data is transmitted from the receive shift register to the receive data buffer, and "H" is output from the RXRDY pin while RXRDY flag is set to '1.' The error flag is also set to '1,' when the errors are detected. Its timing is around the center of the first stop bit as shown in Fig. 8-9.

The framing errors can be detected by recognizing the first stop bit as '0.' The overrun errors can be detected at the timing of shifting received data from receive shift register to receive data buffer if previously received data has not been read by CPU. In this case, previous data is overwritten by new one in receive data buffer. Such error detection has no influence on the received operation, although the receive a data after the framing errors cannot be guaranteed.

The RXRDY flags are cleared to '0' when the received data is read by CPU. The RXRDY pin is also cleared

KP61 does not perform the receive operation in the receive disable condition (bit 2 of the Command Register = '0'). KP61 immediately discontinues the

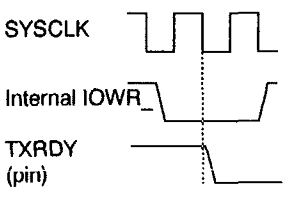


Figure 8-4 The change of TXRDY (pin) by data write

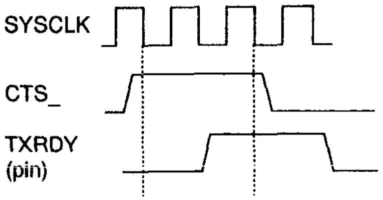


Figure 8-5 The change of TXRDY (pin) by CTS_ signal

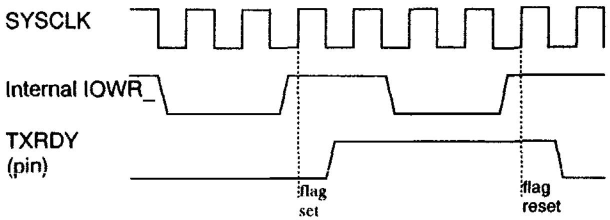


Figure 8-6 The change of TXRDY (pin) by transmission enable (disable) write

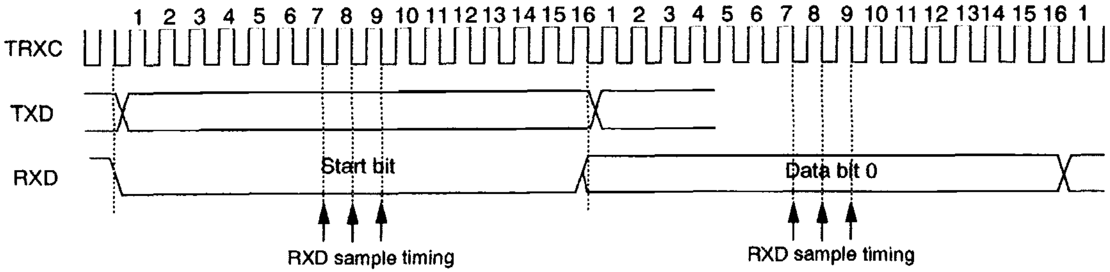


Figure 8-7 Transmit/receive clock and transmit/received data (External TRXC is selected as the transmit/receive clock)

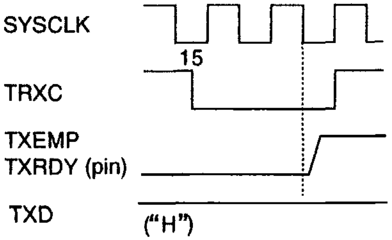


Figure 8-8 Transmit/receive clock and TXEMP, TXRDY signals change timing

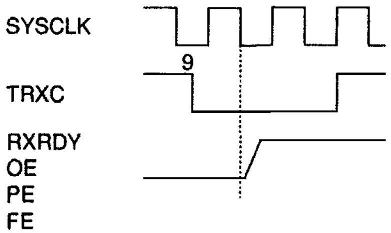


Figure 8-9 RXRDY, error flag set timing while receiving the first stop bit (External TRXC is selected as the transmit/receive clock)

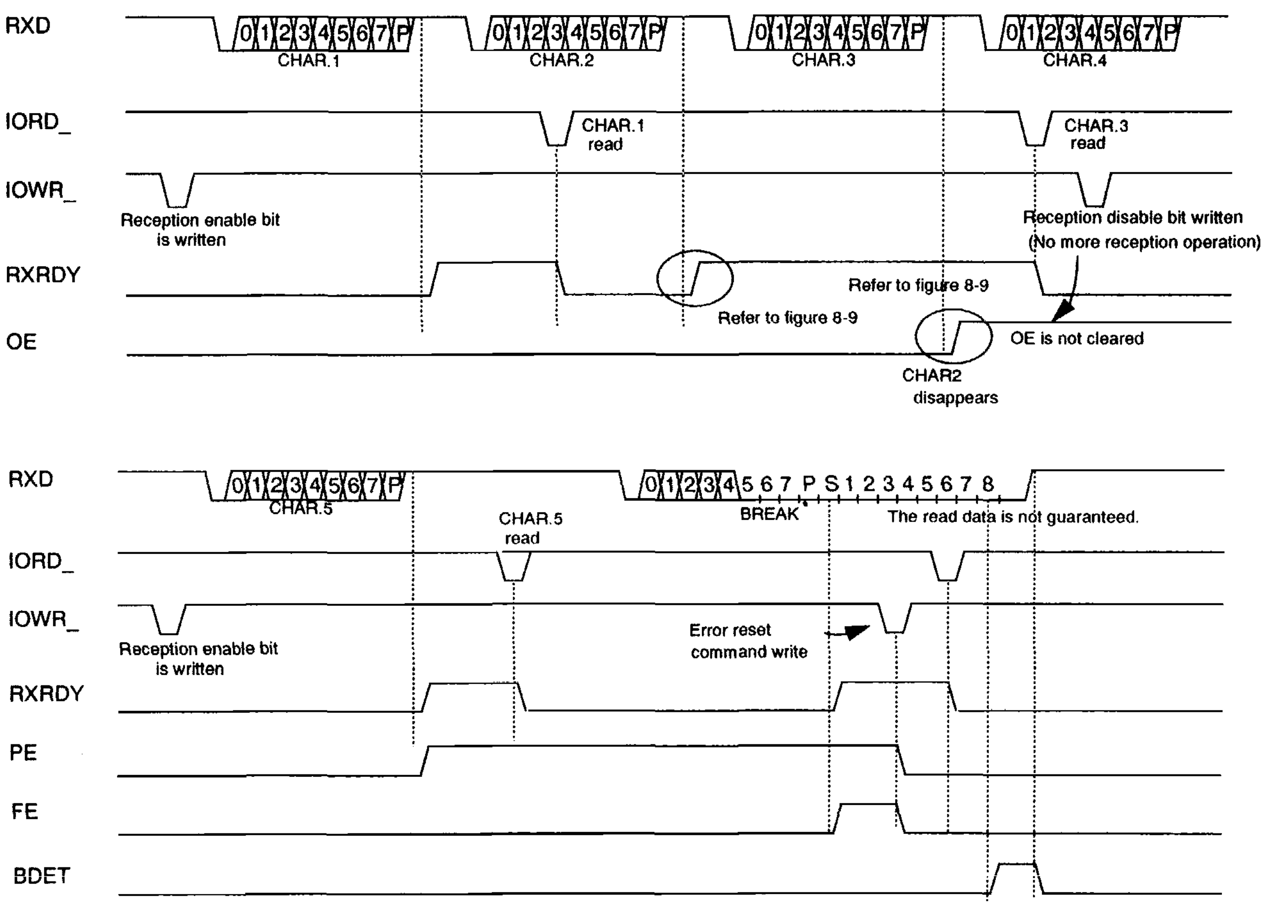


Figure 8-10 Reception operation and flag timing (Example of character length 8 bit, parity bit, two stop bits)

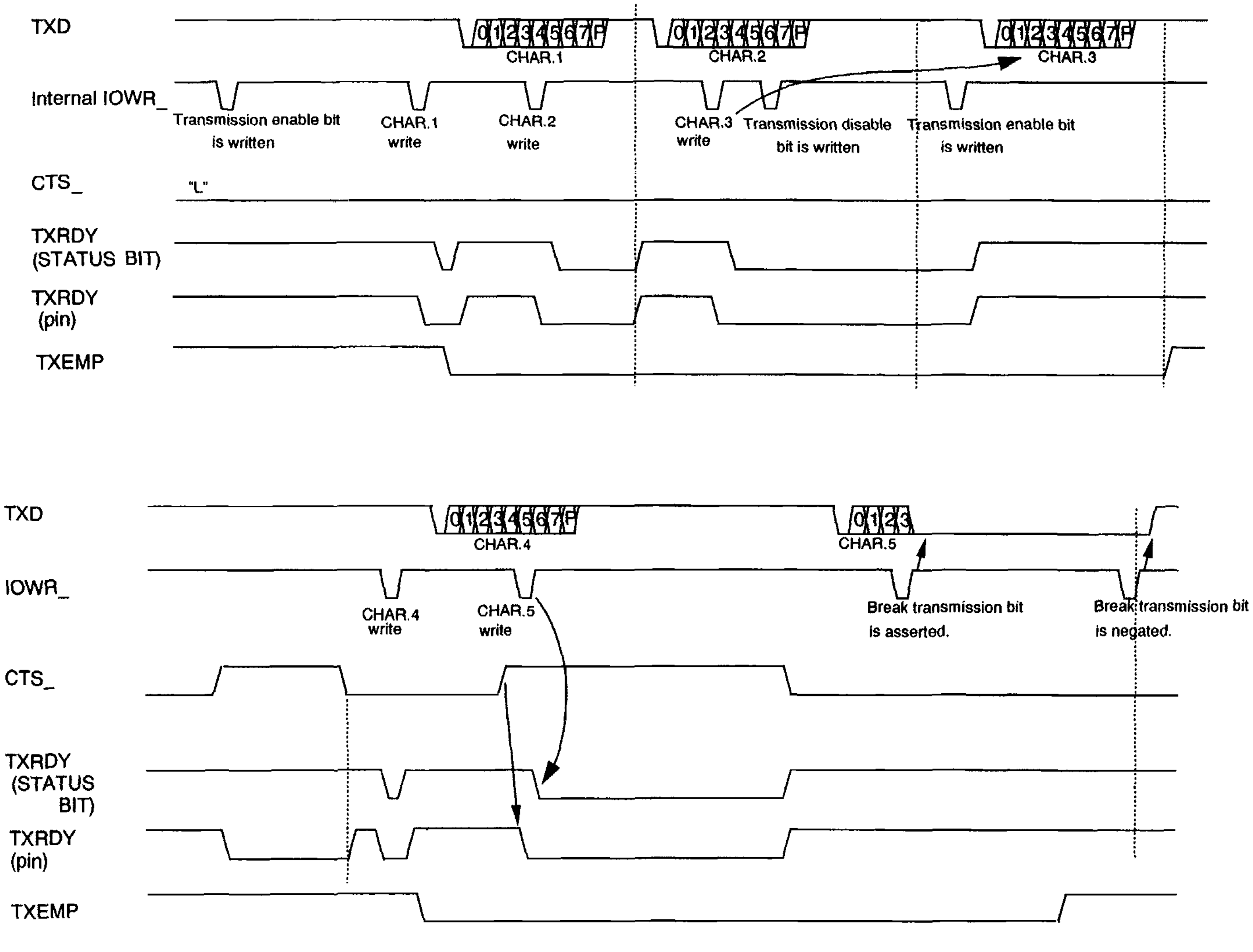


Figure 8-11 Transmission operation and signal change timing (Example of character length 8 bit, parity bit)

receiving operation without waiting for the completion of the data reception in progress. Therefore, no error detections are performed.

KP61 outputs BDET = "H" by recognizing '0' of eight bit after framing error detection on RXD pin as break. This condition continues until KP61 detects 'H' from the RXD pin. Therefore, it is recommended to continuously transmit '0' in a low for at least twenty bits to detect break signal. The break detection cannot be performed in the receive disable condition.

Sleeping mode

KP61 supports the sleeping mode. This mode is used when communication is performed among multiple KL5C80A16 with UART. The sleeping mode becomes valid when character length of nine bits is selected by the Mode Register, and D1 of the Command Register C = '1.' In the sleeping mode, KP61 does not transfer the receive data of the receive shift register to the receive data buffer, when bit 8 of received data equals 0. Neither RXRDY flag nor pin goes 'H'. The sleeping mode can be used as follows.

(1) Set each slave controller to the sleep mode.

(2) Transmits the slave controller selection address from UART of the master controller. Bit 8 of the transmit data should be set to '1.'

(3) Each slave controller judges whether or not the received address indicates itself, and if judged yes, cancels its sleeping mode according to the program.

(4) Transmits the data from UART of the master controller. Bit 8 of the transmit data should be set to '0.' In this operation, the received data on other than specified slave controller cannot be transmitted from the shift register to the receive data buffer because the non-specified slave controllers remain in the sleeping mode.

Communication only with specified slave controller is thus possible.

In the sleeping mode,

- Framing errors can be detected.
- Parity errors cannot be detected.
- Overrun errors is detected if the receive data buffer is never read out while receiving the data of bit 8 = '1' two times.

Baud Rate generator and transmit/receive clock selection

KP61 comprises common internal Baud Rate generator in channels 1 and 2. It is possible to select either external clock which input from the TRXC pin or internal clock obtained by divided frequency of the system clock by the Baud Rate generator as transmit/ receive clock. This is specified by the Command Register B.

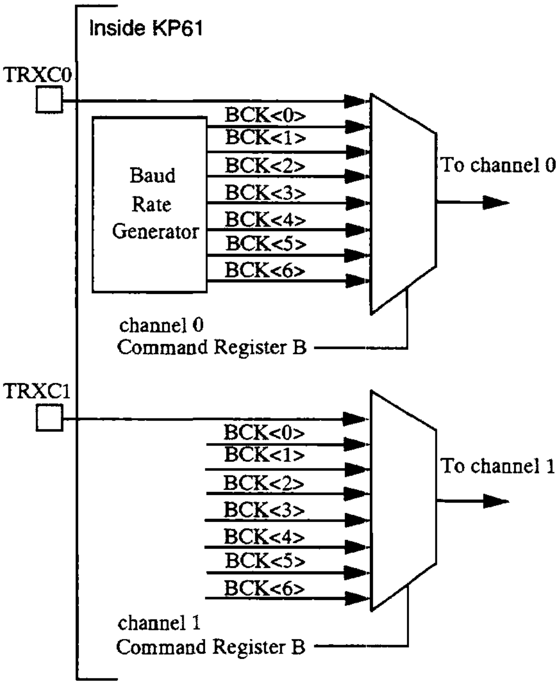


Figure 8-12 Transmit/receive clock selection

The operation of Baud Rate generator can be programmed in 'RATE<7:0>' value of Baud Rate Register.

By setting RATE <7> = 0, a low speed internal transmit/receive clocks' group becomes available, and by setting RATE <7> = 1, a high speed internal transmit/receive clocks' group becomes available. Output of the Baud Rate generator can be obtained as BCK <6:0>. BCK <0> is the fastest, followed by BCK <1> (half of BCK <0>) and BCK <2> (half of BCK <1>) and so on. The internal transmit/receive clock corresponding to 1/64 of BCK <0> can be obtained from the lowest speed BCK <6>.

RATE <6:0> is an operation constant set to the Baud Rate generator. When this value and frequency of the system clock are n (the decimal system) and Fsysclk [Hz] respectively, Baud Rates of the transmit/receive clock obtained from BCK <i> of the Baud Rate generator are as follows.

Table 8-2 Baud Rate calulation

RATE<7>	Baud Rate of transmit/receive clock obtained from BCK <i> (BPS)
0	$\frac{F_{\text{sysclk}}}{(n+1)\times 2^{(i+1)}}\times\frac{1}{16}$
1	$\frac{F_{\text{sysclk}}\times(n+128)}{256\times 2^{(i+1)}}\times\frac{1}{16}$

i: 0~6
n = RATE<6:0>

Table 8.3 shows the Baud Rates obtained by the above formula and 'RATE' values, when the system clock frequency is 10 MHz, 7.159 MHz and 3.932 MHz.

8.7 Reset

KP61 can be initialized to the following conditions by software and hardware reset. However, software reset is only valid on the targeted channel.

Transmit disable condition
Receive disable condition
Each error flag = '0'
TXRDY (flag) = '1' TXRDY (pin) = 'L'
RXRDY (flag) = '0' RXRDY (pin) = 'L'
BDET (flag) = '0' BDET (pin) = 'L'
TXEMP (flag) = '0' (This can be set at '1' by the mode write.)
DTR (flag) = '0' DTR_ (pin) = 'H'
RTS (flag) = '0' RTS_ (pin) = 'H'
DSR (flag) = '0' DSR_ (pin) = 'H'
Register switchover F/F: Clear ('0')

The Baud Rate Register, the transmit/receive clock selection bit (D2, D1 and D0) of Command Register B and the sleeping mode bit (D1) of Command Register C can be reset only by the hardware reset. It should be noted that the register data before the software reset is still valid in this case.

8.8 Precautions

- When each status and data register is read out, the data at which reading is started is read out. Even if the contents of each flags and transmit data buffer change, no influence appears in the read out data.
- When 7 bit is selected for the communication character length, the value of bit 7 of the receive data when read is not guaranteed in the current version of KP61 which is integrated in KL5C80A16.
- Basically, switchover of the read out register is performed between channels independently. However, in case of the current version of KP61 which is integrated on the KL5C80A16, clear of the register switchover F/F by reading of the extention Status Register B is the only case that is valid on both channels.
- Refer to Chapter 3 for signal pins which are multiplexed with the parallel port. For a detailed description of the use of these pins, refer to Chapter 12.

Table 8-3 Baud Rate examples (Baud Rate error is less than 0.5%)

System clock rate (MHz)	RATE<7>	RATE<6:0>	BCK<i> i=0~6	Baud Rate (BPS)	Note
10	1	124	BCK<0>	307200	RATE<7:0>=FCH
			BCK<1>	153600	
			BCK<2>	76800	
			BCK<3>	38400	
			BCK<4>	19200	
			BCK<5>	9600	
			BCK<6>	4800	
	0	64	BCK<0>	4800	RATE<7:0>=40H
			BCK<1>	2400	
			BCK<2>	1200	
			BCK<3>	600	
			BCK<4>	300	
			BCK<5>	150	
			BCK<6>	75	
7.159	1	48	BCK<0>	153600	RATE<7:0>=B0H
			BCK<1>	76800	
			BCK<2>	38400	
			BCK<3>	19200	
			BCK<4>	9600	
			BCK<5>	4800	
			BCK<6>	2400	
	0	92	BCK<0>	2400	RATE<7:0>=5CH
			BCK<1>	1200	
			BCK<2>	600	
			BCK<3>	300	
			BCK<4>	150	
			BCK<5>	75	
			BCK<6>	------	
3.932	1	32	BCK<0>	76800	RATE<7:0>=A0H
			BCK<1>	38400	
			BCK<2>	19200	
			BCK<3>	9600	
			BCK<4>	4800	
			BCK<5>	2400	
			BCK<6>	1200	
	0	50	BCK<0>	2400	RATE<7:0>=1AH
			BCK<1>	1200	
			BCK<2>	600	
			BCK<3>	300	
			BCK<4>	150	
			BCK<5>	75	
			BCK<6>	------	

9. Timer/Counter

9.1 General description

The KL5C80A16's Timer/Counter is a KP63A(16-bit high-performance timer/counter macro cell). The KP63A is asynchronous 16-bit programmable binary timer/counter with 8-bit prescaler designed for KC80 microcomputer. The KP63A consists of four down counters with four operation modes (pulse width modulation (PWM) mode, continuous count mode, single count mode and watchdog timer mode), and its operation can be set by a mode control word from the CPU. Count values can be read out from the bus. A status read command allows you to read out the set mode or status of OUT outputs.

Features

- Built-in 8-bit prescaler. From GATE input external clocks can be got and counted.
- Built-in four 16-bit down counters, each of which enables four operation modes to be set and run
- Stable counter readout with no affects on count operation
- Set mode and status of OUT outputs can be read out.

9.2 Block diagram

The following are the block diagram of Timer/Counter and I/O register mapping.
The interrupt request signal of each channel is connected to the built in interrupt controller as well as external pins.The interrupt request signal of channel 3 is connected to NMI_ input of the CPU.

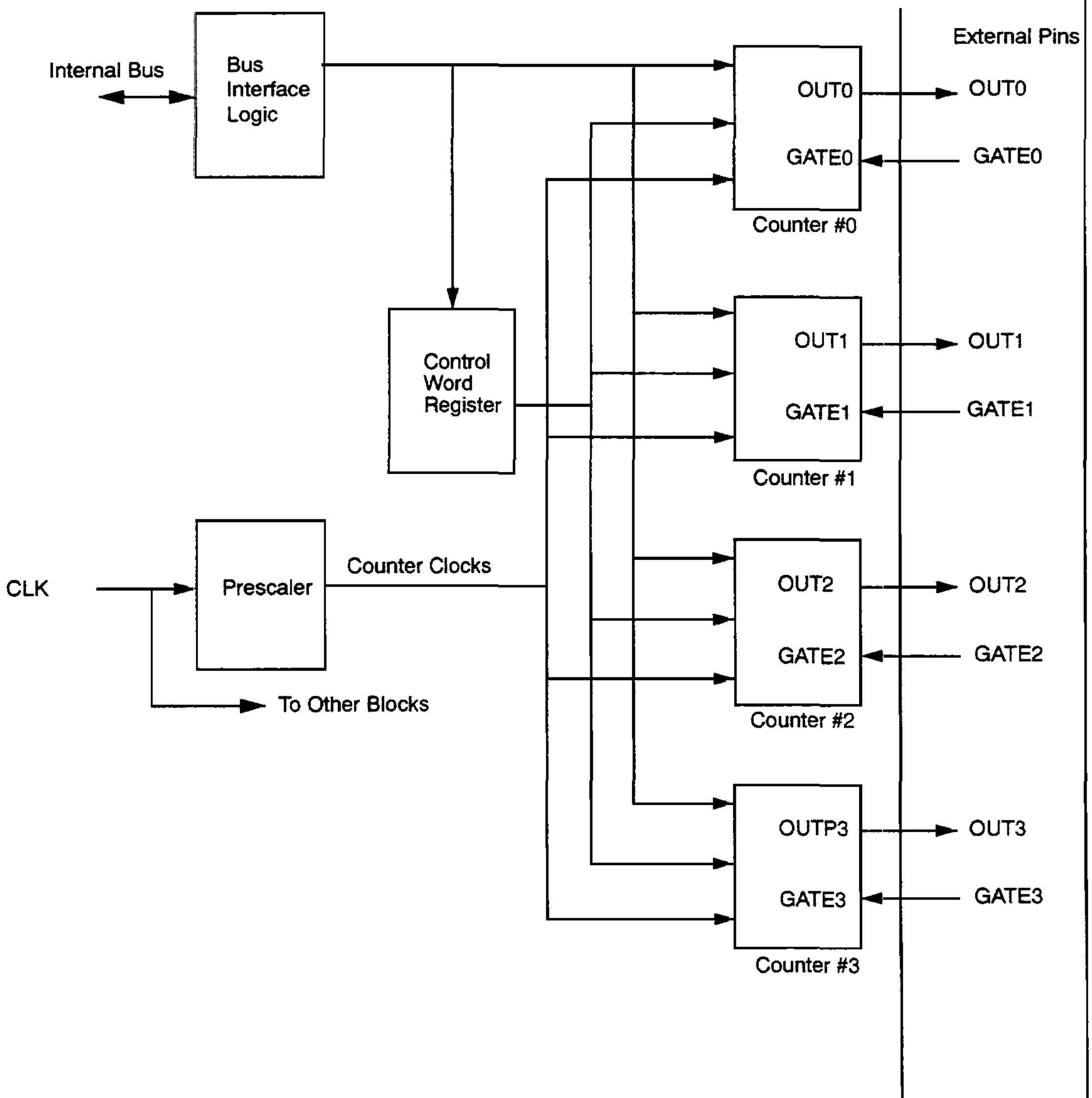


Figure 9-1. Timer/Counter Block Diagram

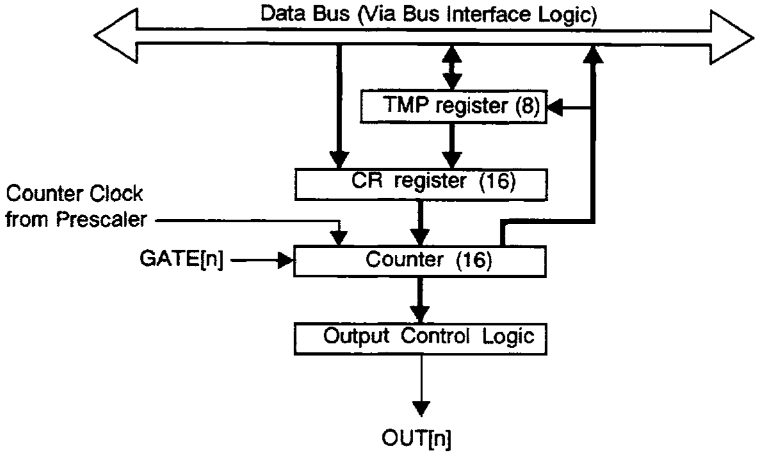


Figure 9-2. Block Diagram of Each Channel Counter Unit

Table 9-1. I/O Register Mapping

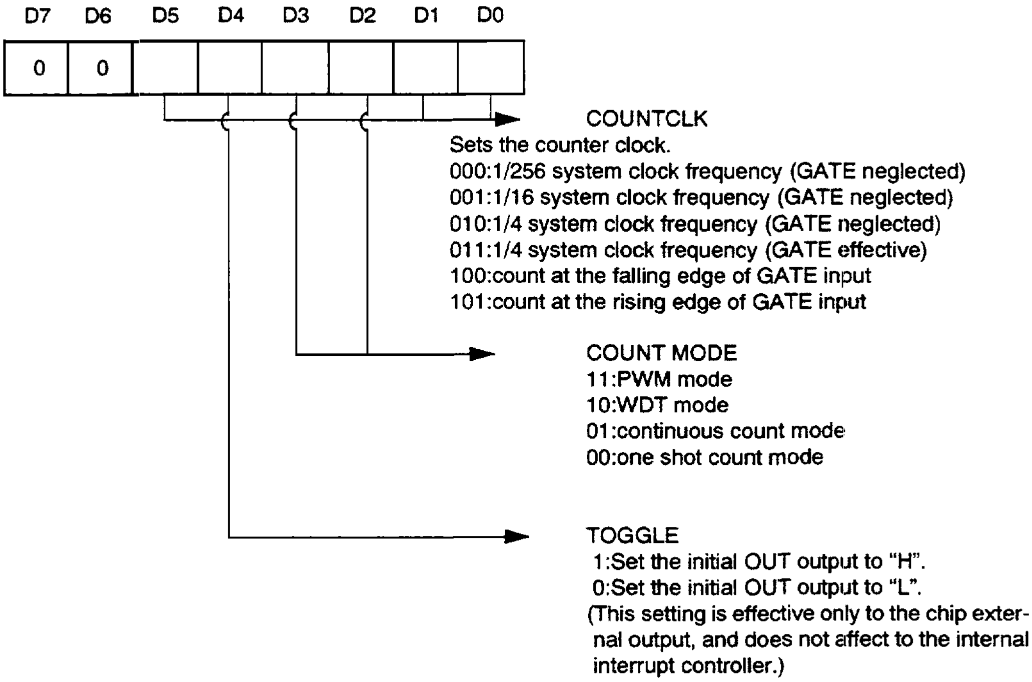
I/O address	Write cycle	Read cycle
20H	channel 0 counter	channel 0 counter
21H	channel 0 control word	channel 0 status
22H	channel 1 counter	channel 1 counter
23H	channel 1 control word	channel 1 status
24H	channel 2 counter	channel 2 counter
25H	channel 2 control word	channel 2 status
26H	channel 3 counter	channel 3 counter
27H	channel 3 control word	channel 3 status

9.3 Pin description

pin name	I/O	description
GATE3~GATE0	I	GATE input It specifies the count operation enable/disable for the counter of each channel. It also works as an external counter clock by set mode.
OUT3~OUT0	O	pulse OUT output It provides frequency divided output or PWM output depending on the set mode. A reset signal makes this output "L" asynchronously with system clock.

9.4 Setting mode

Mode is set by writing a control word into each channel.



COUNTCLK
Sets the counter clock.
000:1/256 system clock frequency (GATE neglected)
001:1/16 system clock frequency (GATE neglected)
010:1/4 system clock frequency (GATE neglected)
011:1/4 system clock frequency (GATE effective)
100:count at the falling edge of GATE input
101:count at the rising edge of GATE input

COUNT MODE
11:PWM mode
10:WDT mode
01:continuous count mode
00:one shot count mode

TOGGLE
1:Set the initial OUT output to "H".
0:Set the initial OUT output to "L".
(This setting is effective only to the chip external output, and does not affect to the internal interrupt controller.)

The prescaler is shared by all channels, but its frequency division rate can be set for each channel.

9.5 Setting an initial count value to the counter

Although the counter is a 16-bit down counter, a data bus is 8 bit wide, so data is written twice for lower byte and higher byte. However, the TMP register is shared for write and read operations, so written data is destroyed if the counter is readout before data write has not been completed.

Lower byte:

D7	D6	D5	D4	D3	D2	D1	D0
7	6	5	4	3	2	1	0

(Numbers indicate data bits.)

Higher byte:

D7	D6	D5	D4	D3	D2	D1	D0
15	14	13	12	11	10	9	8

(Numbers indicate data bits.)

9.6 Counter readout

Counter readout is performed by accessing the address of each counter. Data is read out twice by 8 bits in the order of lower byte and higher byte. A higher byte is read out via the TMP register. As the content of TMP register is copied from the higher byte of the counter when the lower byte is read out, the value in the TMP register would not change if a value in the counter changes during two read cycles. However, the TMP register is shared for write and read operations, so read data is destroyed if the counter is written before data read has not been completed.

Lower byte:

D7	D6	D5	D4	D3	D2	D1	D0
7	6	5	4	3	2	1	0

(Numbers indicate data bits.)

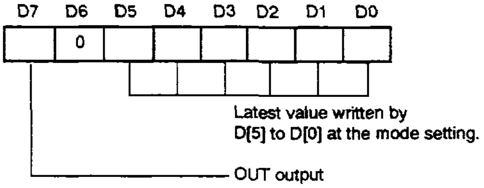
Higher byte:

D7	D6	D5	D4	D3	D2	D1	D0
15	14	13	12	11	10	9	8

(Numbers indicate data bits.)

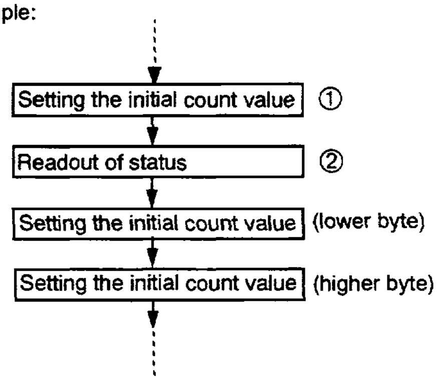
9.7 Status readout

Status register is provided for each channel. The format of the status register is as follows.



Readout of status clears a read/write sequence (see the example below). Therefore, reading out status before setting the initial count value or reading out the counter prevents writing value erroneously.

Example:


Setting the initial count value ①
Readout of status ②
Setting the initial count value (lower byte)
Setting the initial count value (higher byte)

① may be higher or lower byte.
② clears a read/write sequence.

9.8 Operation modes

The following is the description for each operation mode with TOGGLE bit "0".

Continuous count mode

In the continuous count mode, the initial count value is loaded and decremented. When the counter reaches "0", the OUT output changes and the reload and decrement of the initial count value is repeated.
When the mode is set, the OUT pins outputs "L". The count operation is started by loading the initial count value, and the OUT pin outputs "H". Whenever the counter reaches"0", the OUT output toggles.
Reload is performed not by writing the initial value during count, but only when the counter reaches "0".

One shot count mode

In the one shot mode, the initial count value is loaded and decremented. When the counter reaches "0", the OUT output changes and the count operation stops until the initial count value is set again.
When the mode is set, the OUT outputs "L". The count operation is started by setting the initial count value, and the OUT pin outputs "H" until the counter reaches "0". Reload is performed by setting the initial count value.

Watch dog timer (WDT) mode

In the watch dog timer mode, the initial count value cannot be written to the CR register. Write operation to the CR register is recognized as a count start or retrigger.
When the mode is set, the counter stops its operation and the OUT pin outputs "L". The count operation is started by the write operation to the CR register (regardless of the data value at the write operation; one write operation) which causes data to be loaded from the CR register, and the OUT pin outputs "H".
The write operation to the CR register during count causes data to be loaded from the CR register again, and the countdown is performed.
When the counter reaches "0", the OUT pin outputs "L" and then the count operation stops until the write operation to the CR register is performed.
In mode setting, the CR register is not initialized. Perform the following procedures to set the CR register to the initial count value. Set to the one shot count mode, set the CR register to the initial count value, and then return to the WDT mode. Perform the write operation to the CR register, which causes the value set in the CR register to be loaded and starts countdown.
The write operation to the CR register during count causes data to be loaded from the CR register again, and the countdown is performed.
These procedures allows any initial count value to be set in the WDT mode. When this timer/counter is used as watch dog timer, it is required not only to set this mode but also to set by SCR to connect this timer/counter output to the KC82 NMI_ input internally. For more information, refer chapter12.
The watch dog timer mode can be used in all channels, but the only channel 3 can be connected the timer/counter interrupt request to the KC82 NMI_ input in the chip.

Pulse width modulation (PWM) mode

In the pulse width modulation mode, a repetitive pulse with the pulse width and frequency set by the initial count value is output at the OUT pin.
The pulse width is set by the higher byte of CR register, and the pulse frequency set by its lower byte. In the first place, how to set pulse frequency is described.
The pulse frequency is a time from the lower byte of CR register loaded in the counter is decremented independently of the higher byte until its count value becomes "0". That is, it denotes the time of (set value by the first write to the CR register + 1).
On the other hand, the pulse width is a time from the

higher byte of CR register loaded in the counter is decremented independently of the lower byte until its count value becomes "0". That is, it denotes the time of (set value by the second write to the CR register + 1).

As both higher and lower bytes are simultaneously counted down independently, any desired pulse can be obtained from the OUT pin.

When the mode is set, the counter stops and the OUT pin outputs "L". When both higher and lower bytes are set to initial count values and they are loaded, the count operation starts and the countdowns of higher and lower bytes are performed independently. But if the pulse width is greater than or equal to the pulse frequency, the OUT is always "H".

Reload is performed not by writing the initial value during count, but only when the lower byte becomes "0".

(Ex. divide-by-4 frequency, initial count value=0308H)
Pulse width: 4(= 3 + 1) x 4 = 16 system clocks
Pulse frequency: 9(= 8 + 1) x 4 = 36 system clocks

9.9 OUT output and interrupt request

The OUT output is the signal which can be reversed by the mode setting. The following is the description for each operation mode with TOGGLE bit "0". In the continuous mode, it toggles whenever the count value becomes "0". In the one shot mode and the WDT mode, it always outputs "H" during count, and it outputs "L" when the count value becomes "0". In the PWM mode, it outputs signals with desired pulse width and frequency. Therefore, the OUT output can be used for obtaining various pulse signals.

The interrupt request, which is different from the OUT output, always occurs when the counter value becomes "0" regardless of the mode setting.

9.10 Operations

In this section, the operations of Timer/Counter are described referring to the figures.

Countdown timing

The countdown of the timer/counter is made at the falling edge of the system clock by one system clock delay in order of channel 0, channel 1, channel 2 and channel 3.

Countdown timing at each selected counter clock (Figure 9-3)

Figure 9-3 shows the case in which channel 0 and channel 1 are set to counter clock of divide-by-4 frequency without GATE function, channel 2 to that of divide-by-16 frequency, and channel 3 to that of divide-by-256 frequency.

In channel 0 and channel 1, the times from 1 to 1' and from 2 to 2' are four system clocks, the time from 3 to 3' is 16 system clocks, and the time from 4 to 4' is 256 system clocks.

Sampling timing of GATE signal (Figure 9-4-A)

Figure 9-4-A shows the case in which all channels are set to divide-by-4 frequency with GATE function. GATE signal sampling is made at every four system clocks (in Figure, at the rising edge of GATE input sample signal) simultaneously for all channels. Therefore, the polarity of the GATE signal sampled at this time determines whether the countdown is made or not at the next timing. GATE function can be used only at the divide-by-4 frequency rate.

Count timing of the external clock (Figure 9-4-B).

Figure 9-4-B shows the case in which the falling edge of the external count clock is set for channel 0, and the rising edge of the external count clock is set for channel 1, as the external clock. The external count clock is taken from the GATE input. The count edge can be select-

ed whether falling or rising by the mode setting.
Set more than four system clocks of the external count clock H pulse width. The actual counts may delay by 1 ~ 3 clocks from the GATE input edge.

Countdown timing and control word write cycle

The control word write cycle for each channel is reflected at the falling edge of the system clock in order of channel 0, channel 1, channel 2 and channel 3 by one system clock delay like as the timer/counter countdown. Therefore, the time to when the control word write cycle is reflected actually is different in case. The shortest case is one clock and the longest case is four clocks.

Countdown timing and count value write cycle

In the continuous count mode and PWM mode, the count value write cycle for each channel is reflected at the falling edge of the system clock in order of channel 0, channel 1, channel 2 and channel 3 by one system clock delay like as the timer/counter countdown. Therefore, the time to when the count value write cycle is reflected to the OUT output actually is different in case. The shortest case is one clock and the longest case is four clocks.

Operations in the continuous count mode (Figure 9-5)

The Figure 9-5 shows channel 0 in the continuous count mode in which divide-by-4 frequency without GATE function and with TOGGLE bit "0".
When the mode is set, the OUT signal is initialized at the next countdown timing and go "L". When the lower count value "02H" and the higher count value "00H" are written, 0002H is loaded to the counter at the next countdown timing, causing the countdown to start and the OUT output to toggle.
If the initial count value (F0ABH) is reset during count, i.e.between 1 and 2 in the Figure, the count value is set to F0ABH at the next countdown timing after the counter reaches "0", i.e. at the point 2 in the Figure. At the same time the OUT output toggles.

Operations in the one shot count mode (Figure 9-6)

The Figure 9-6 shows channel 0 in the one shot count mode in which divide-by-4 frequency without GATE function and with TOGGLE bit "0".
When the mode is set, the OUT signal is initialized at the next countdown timing and go "L". When the lower count value "02H" and the higher count value "00H" are written, 0002H is loaded to the counter at the next countdown timing, causing the countdown to start and the OUT output to go "H".
If the initial count value (0003H) is reset during count, i.e.between 1 and 2 in the Figure, a retrigger is generated at the next countdown timing (at 2 in the Figure) and the countdown continues from this initial count value. When the initial count value is not reset between 2 and 3, and the counter reaches "0"(at 3 in the Figure), it is cleared to "FFFFH" and the countdown stops. The OUT output goes "L".
When the reset of initial count value (ABCDH) is recognized at the timing from the count value "0" to "FFFFH" as at the point 1,it is recognized as a retrigger and OUT signal does not change.

Operations in the watch dog timer (WDT) mode (Figure 9-7)

The Figure 9-7 shows the watch dog timer (WDT) mode in which divide-by-4 frequency without GATE function and with TOGGLE bit "1".
In the WDT mode, the write operation of initial count value (write to the CR register) is recognized as a retrigger and the value of the initial count value register does not change. To set the initial count value register to a desired value, set the one shot count mode to write a desired value to the CR register, and then set the WDT mode.

The basic operations of the WDT mode are similar to those of the one shot count mode. At first, when the one shot count mode is set, the OUT output goes "H" at the next countdown timing. When the lower count value "03H" and the higher count value "00H" are written, 0003H is loaded to the counter at the next countdown timing, causing the countdown to start and the OUT output to go "L".

Then, if the WDT mode is set (at 1 in the Figure), the OUT output goes "H" at the next countdown timing and the counter has stopped at this moment. When the write operation of count value (any data) is made once,"0003H" in the CR register is loaded to the counter at the next countdown timing, causing the countdown to start for the first time in the WDT mode and the OUT output to go "L" (at 2 in the Figure).

When the count write operation (any data) is made during count, i.e. between 2 and 3 in the Figure, it is recognized as a retrigger and "0003H" is reloaded at the next countdown timing (at 3 in the Figure). Then the countdown starts again. When the count value becomes "0", the OUT pin outputs "H", the count value is cleared to "FFFFH" and the counter stops (at 4 in the Figure).

Then, if the write operation of count value (any data) is made once, "0003H" is loaded to the counter at the next countdown timing, causing the countdown to restart.

Although the OUT output makes unnecessary change for the WDT mode when the one shot count mode is set, the timer/counter channel 3 interrupt request which is connected to NMI_ of CPU internally does not output pulses until the count value becomes "0".

Operations in the pulse width modulation (PWM) mode (Figure 9-8)

The Figure 9-8 shows channel 0 in the pulse width modulation (PWM)mode in which divide-by-4 frequency without GATE function and with TOGGLE bit "0".

The basic operations of the PWM mode are similar to those of the continuous count mode. When the mode is set, the OUT signal is initialized at the next countdown timing and goes "L". When the lower count value (pulse frequency data) "03H" and the higher count value (pulse width data) "01H" are written, 0103H is loaded to the counter at the next countdown timing, causing the countdown to start and the OUT output to toggle.

As shown in the Figure, the countdowns of the higher and lower bytes are made independent and simultaneously. When the higher byte count becomes "0", the OUT output toggles. One countdown operation is from 1 to 2, i.e., from the load of initial count value to when the lower count value becomes "0".

Between 1 and 2 the OUT pin outputs a pulse with (1+1)/fc=2/fc of width and (3+1)/fc=4/fc of frequency (fc: countdown frequency). An interrupt occurs in the internal interrupt controller at 2.

If the initial count value (0204H) is set again during count, "0204H" is loaded to the counter at the next countdown timing after the lower count value becomes "0", i.e. at 2 in the Figure.

If the higher byte is greater than or equal to the lower byte, the OUT pin always outputs "H" as shown between 3 and 4. At the next countdown timing after the lower byte becomes "0" (at 4 in the Figure), the initial count value "0505H" is loaded to the counter and the countdown is performed continuously again.

9.11 Reset

Reset is performed by RESET_= "L" and affects the following features.

(1)Stops the operation of counter. The counter is kept in stop state even after clearing the reset.

(2)Clears a read sequence in readout of register within the counter and a write sequence in write of initial count value.

(3)Clears the CR register within the counter and the counter register to "FFFFH".

(4)Enters the one shot count mode (divide-by-256 frequency counter clock, TOGGLE bit "0"), sets the OUT output to "L", and holds the value after clearing the reset until the mode is set.

9.12 Precautions

The way to stop the count

Setting the mode again in counting stops the count.

Maximum/minimum value of initial count value

Mode	Minimum value	Maximum value
Continuous count	0001H	FFFFH
One shot count	0001H	FFFFH
Pulse width modulation	High order: 01H	High order: FFH
(PWM)	Low order: 01H	Low order: FFH
Watch dog timer	Unable to set*	Unable to set*
(WDT)		

*When it is set again and used in the WDT mode after it has been set to other mode, it operates according to the conditions of the mode in which the initial count value was set.

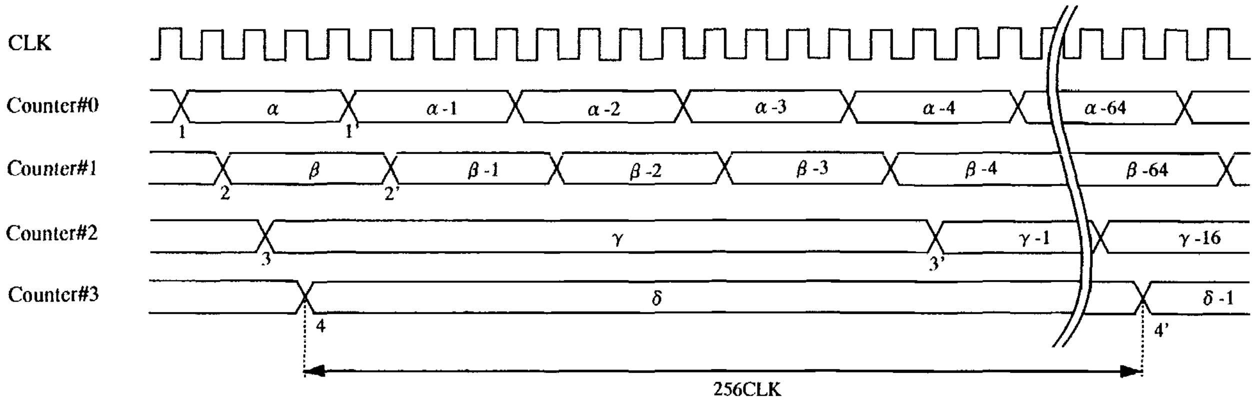


Figure 9-3 Countdown timing (channel 0, 1: without GATE function, channel 2: divide-by-16 frequency, channel3: divide-by-256 frequency)

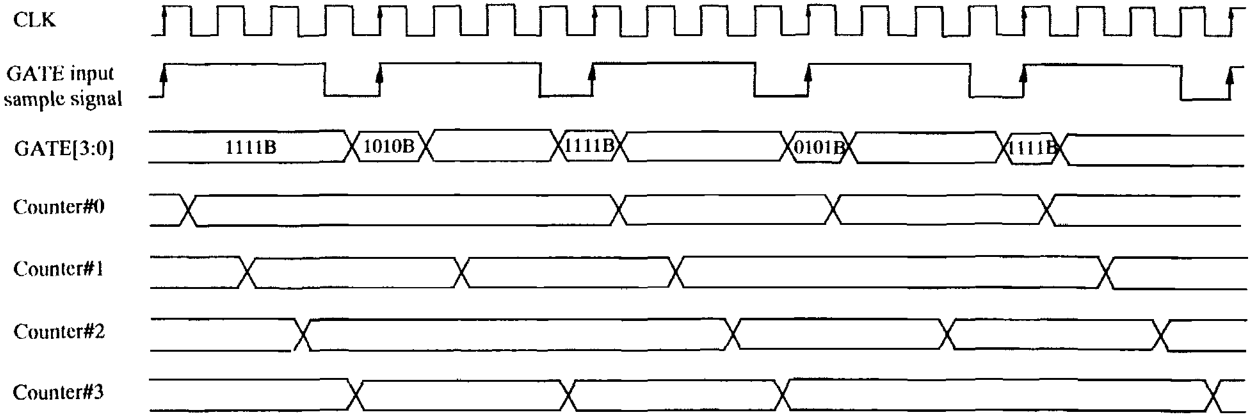


Figure 9-4-A Sample timing of GATE signal (divide-by-4 frequency with GATE function)

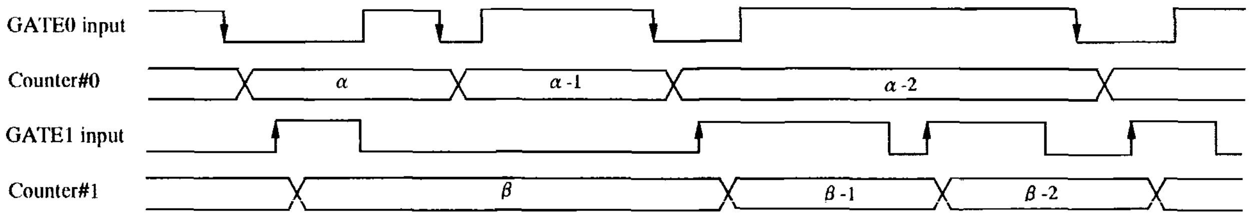


Figure 9-4-B Clock input timing from GATE input (channel 0 is counted at the falling edge of the GATE input, channel 1 is counted at the rising edge.)

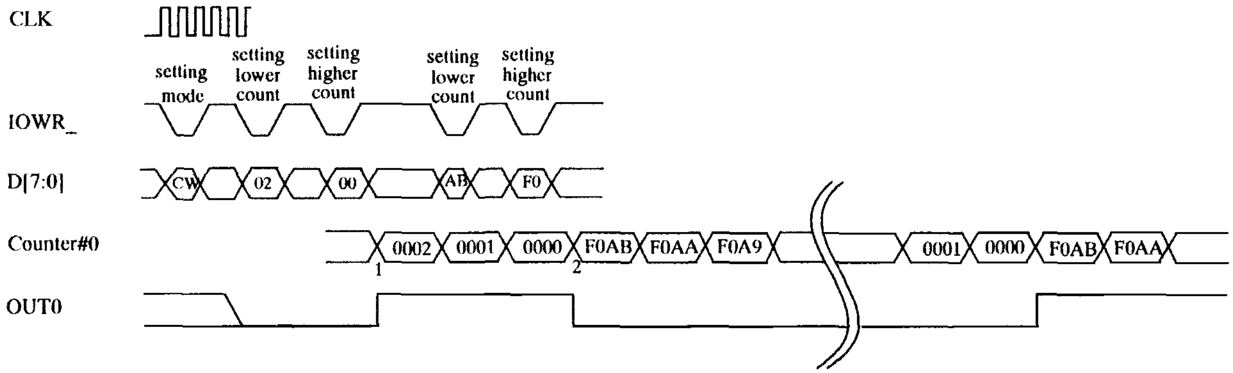


Figure 9-5 Continuous count mode (divide-by-4 frequency without GATE function, with TOGGLE bit "0")

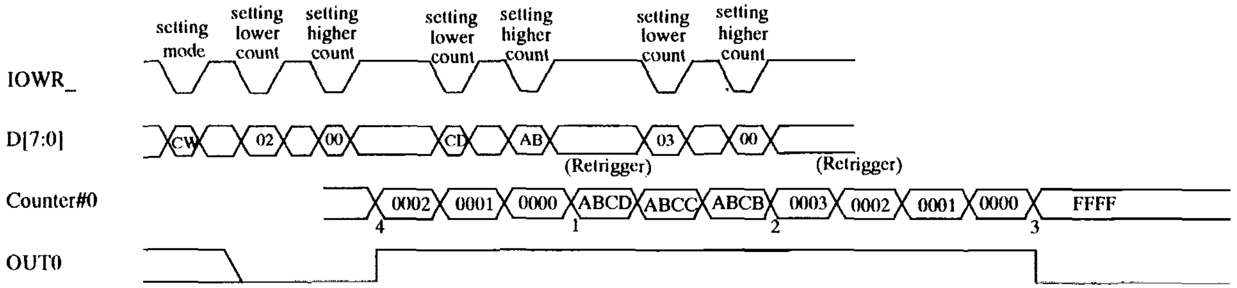


Figure 9-6 One shot count mode (divide-by-4 frequency without GATE function, with TOGGLE bit "0")

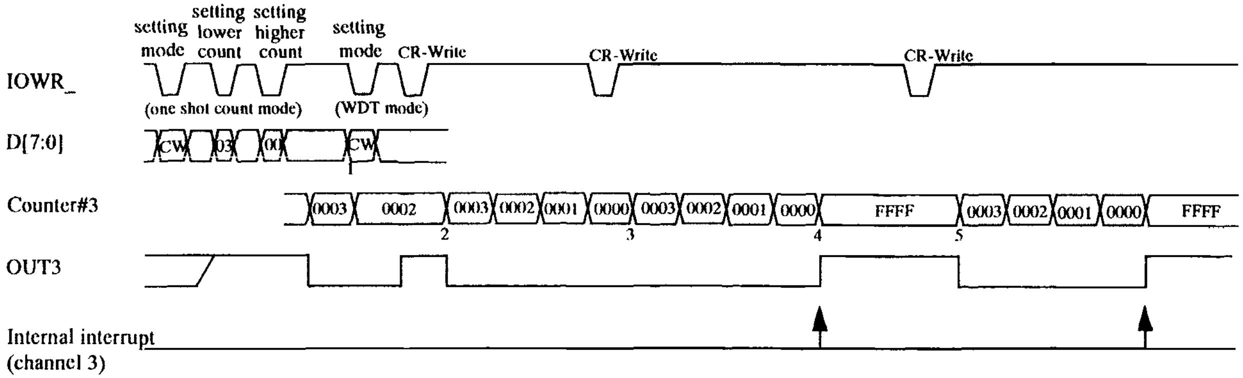


Figure 9-7 WDT mode (divide-by-4 frequency without GATE function, with TOGGLE bit "1")

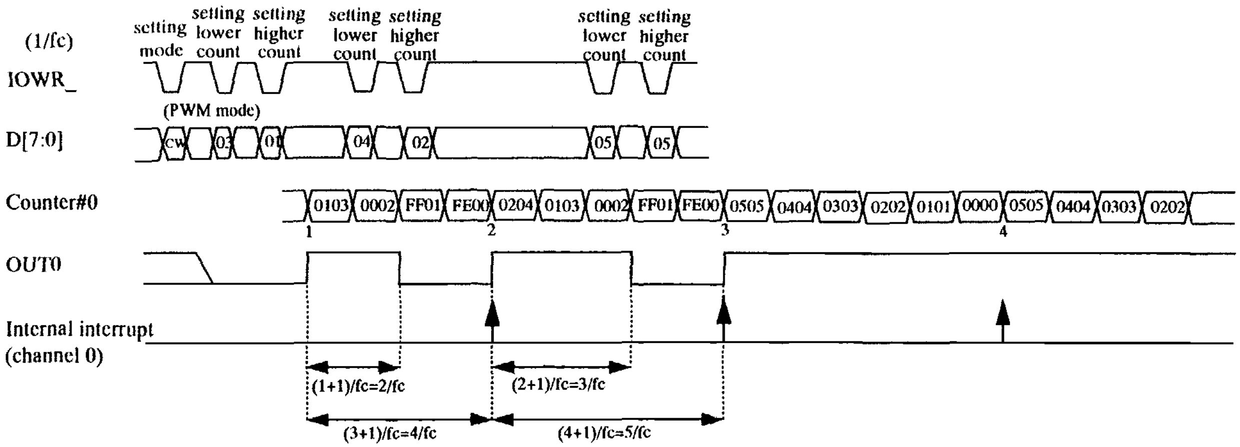


Figure 9-8 PWM mode (divide-by-4 frequency without GATE function, with TOGGLE bit "0")

10. Clocked Serial I/O

10.1 General Description

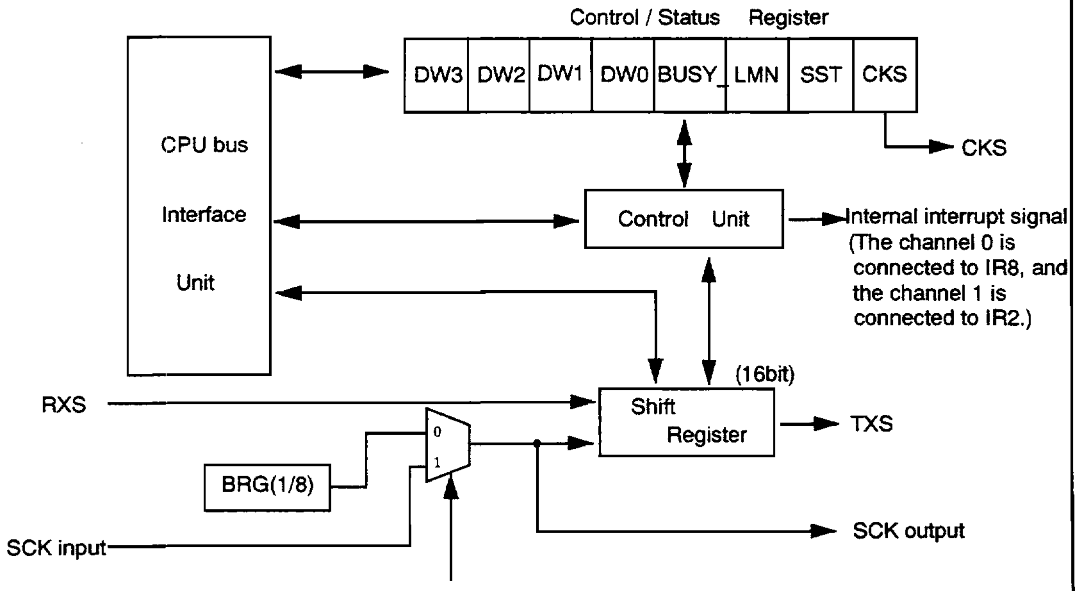
The Clocked Serial I/Os in KL5C80A16 are two channels of KP62 macrocells. They can handle characters whose word length is from 8 bit to 16 bit. The data transmission direction can be selected from LSB first or MSB first. The external clock or internal system clock can be chosen as serial clock. They are best suited for the serial interconnections with single chip MCUs, A/D or D/A converters, and serial ROMs. The description in this chapter is the explanation for one channel, because the two channels of the Clocked Serial I/Os are identical.

10.2 Block Diagram

The block diagram of the Clocked Serial I/O is shown below.

Features

- Half-duplex clocked serial I/Os
- Maximum baud rate is 5 M bps (when the external serial clock is selected, 5MHz)
- Characters length is from 8 bit to 16 bit.
- The data transmission direction can be selected from LSB first or MSB first.
- The external clock or internal system clock divided by 8 can be chosen as serial clock.


Control / Status Register
DW3
DW2
DW1
DW0
BUSY
LMN
SST
CKS
CKS
CPU bus
Interface
Unit
Control Unit
Internal interrupt signal
(The channel 0 is connected to IR8, and the channel 1 is connected to IR2.)
(16bit)
RXS
Shift Register
TXS
0
1
BRG(1/8)
SCK input
SCK output
CKS

The external pin names have the number 0 or 1, which indicate channel 0 or 1 of the Clocked Serial I/Os.

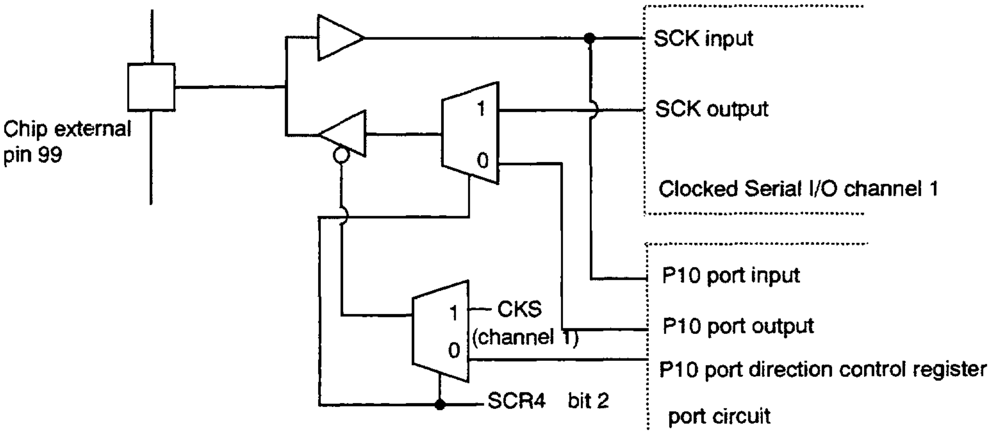
Figure 10-1 Block diagram of the Clocked Serial I/O

10.3 Pin description

Pin Name	I/O	Function
SCK0 SCK1	I/O	Serial clock input When it is set as an input, it is used as an external serial clock input. When it is set as an output, it is used as an output pin of the serial clock generated in the Clocked Serial I/O.
RXS0 RXS1	I	Rtransmit dataeceive data input The pin receive data comes from.
TXS0 TXS1	O	Ttransmission Enable commandransmit data output The pin transmit data goes out.

Channel 1 internal interconnection

The SCK pin of channel 1 is multiplexed with parallel port.
They are interconnected as shown in the diagram below.



10.4 I/O address mapping

The register I/O mapping is indicated in table 10-1.

Table 10-1

I/O address	write	read
30H	channel 0 transmit data	channel 0 receive data
31H	channel 0 mode register/command	channel 0 status
32H	channel 1 transmit data	channel 1 receive data
33H	channel 1 mode register/command	channel 1 status

10.5 Initialization

It is necessary to program mode and issue commands as shown in Fig. 10-2 before transmission/ reception of data starts.

Mode programming

After reset, please set modes as shown in Fig. 10-2. The Clock Serial I/O is always waiting for mode programming after reset. Before programming modes again, make sure that BUSY_ bit of Status Register is 1. Programming modes can be omitted when selecting mode is "external serial clock, 8 bit character length, SST = 0, MSB first".

Mode Register

D0: CKS

By selecting 0 for this bit, the output of B. R. G. (Baud Rate Generator) in The Clock Serial I/O is selected as the serial clock. The output of B. R. G. is the system clock divided by eight. The system clock divided by eight is output from SCK pin.

By selecting 1 for this bit, the input signal from SCK is selected as the serial clock. In The Clock Serial I/O, the input signal from SCK is sampled by the rising edge of system CLK.

D1: SST

By selecting 0 for this bit, the data is transmitted on the falling edge of serial clock, the data is received at the rising edge of serial clock. In this case, if internal system clock is specified as serial clock, the initial value of SCK output is H.

By selecting 1 for this bit, the data is transmit on the rising edge of serial clock, the data is received at the falling edge of serial clock. In this case, if internal system clock is specified as serial clock, the initial value of SCK output is L.

D2: LMN

By selecting 0 for this bit, the first bit of the data transmit is MSB. By selecting 1 for this bit, the first bit of the data transmit is LSB.

D7~D4: DW3~DW0

These bit specified character length. The character length that can be specified is from 8 to 16 bit.

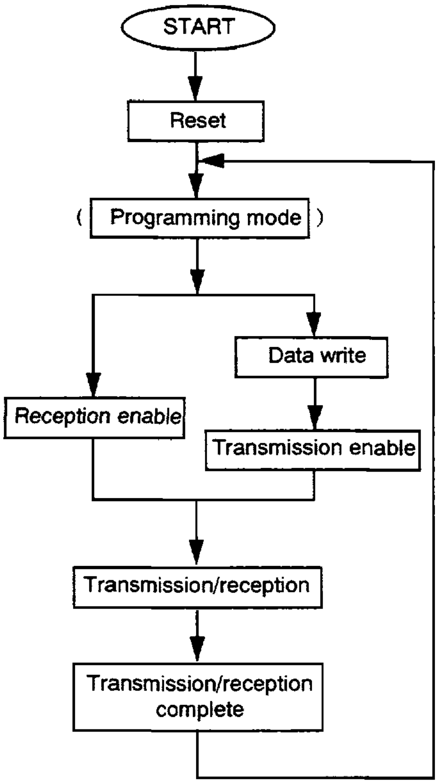
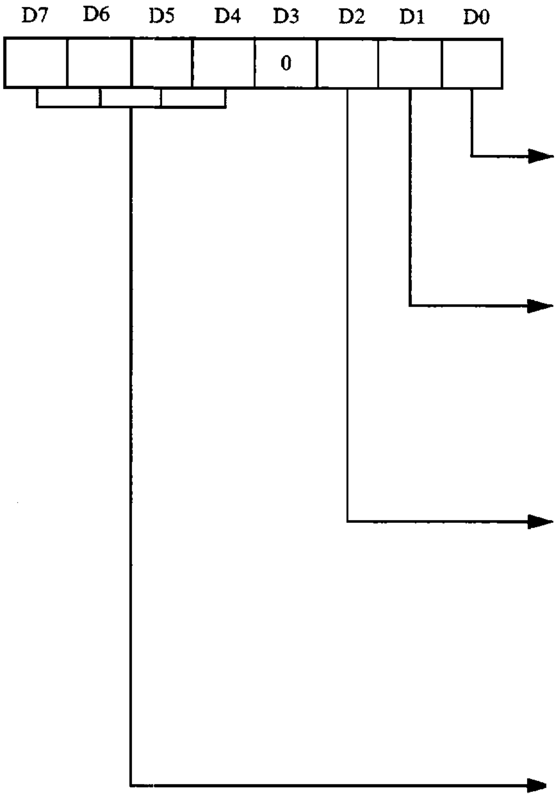


Figure 10-2 Initialization flow

Mode Register



CKS
Serial clock selection bit
0: internal serial clock (the system clock divided by eight)
1: external clock (SCK input)

SST
transmission timing / sample timing

0: ↓ / ↑ SCK initial value is H
when internal clock is selected

1: ↑ / ↓ SCK initial value is H
when internal clock is selected

LMN
transmission direction
0: MSB first
1: LSB first

DW3	DW2	DW1	DW0	Character length
0	1	1	1	8
1	0	0	0	9
1	0	0	1	10
1	0	1	0	11
1	0	1	1	12
1	1	0	0	13
1	1	0	1	14
1	1	1	0	15
1	1	1	1	16

(Note) The bit patterns that is not described here are not guaranteed.

Command Register

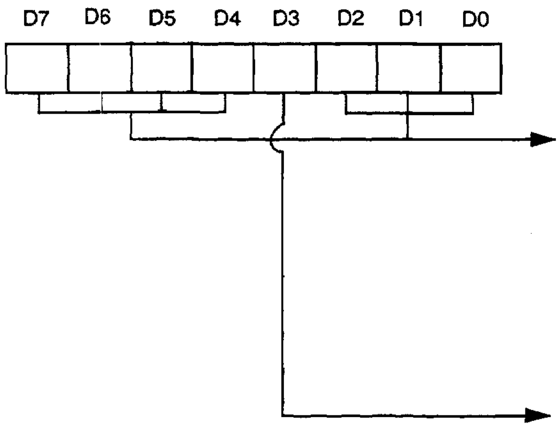
D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0		

D1	D0	command
0	0	Cancel
1	0	Transmission Enable
1	1	Reception enable
0	1	Reserved

(Note) The bit patterns that is not described here are not guaranteed.

Status Register

The status information stored in Status Register is read out from the I/O address 31H for channel 0, 33H for channel 1. The bit assignment of the Status Register is indicated below.



The same as bits correspond to Mode Register
D0: CKS
D1: SST
D2: LMN
D4: DW0
D5: DW1
D6: DW2
D7: DW3
character length

D3: BUSY_
0: under transmission or reception
1: waiting

Commands

There are three commands, Transmission Enable Command, Reception Enable Command, Cancel Command. Transmission Enable Command lets the Clocked Serial I/O begins data transmission. Reception Enable Command lets the Clocked Serial I/O begins data reception. Cancel Command stops data transmission or data reception, even the Clocked Serial I/O is under data transmission or data reception. The Clocked Serial I/O goes into waiting condition after mode programming by Cancel Command. Transmission Enable Command and Reception Enable Command cannot be accepted while the Clocked Serial I/O is transmitting or receiving data.

10.6 Data transmission and reception

Data reception

As shown in Fig. 10-2, in waiting condition (the condition where BUSY_ bit of Status Register is 1 after mode setting) data reception begins at issue of Reception Enable Command. After data reception begins, BUSY_ bit of Status Register turns to be 0. In case that the internal clock is selected as serial clock, the serial clock is output from SCK pin.

After the Clocked Serial I/O received the number of bits of receive data specified in Mode Register at the sampling timing specified in Mode Register, BUSY_ bit of Status Register turns to be 1, an internal interrupt is generated to let the CPU know that reception is over. Then the Clocked Serial I/O goes into the waiting condition. The received data can be read out from data bus in this condition.

Data Transmission

As shown in Fig. 10-2, in waiting condition (the condition where BUSY_ bit of Status Register is 0 after mode setting) write transmit data. The transmit data is accepted only in the waiting condition.

Then data transmission begins at issue of Transmission Enable Command. After data transmission begins, BUSY_ bit of Status Register turns to be 0. In case that the internal clock is selected as serial clock, the serial clock is output from SCK pin. After the Clocked Serial I/O transmit the number of bits of transmit data specified in Mode Register at the transmitting timing specified in Mode Register, BUSY_ bit of Status Register turns to be 1, an internal interrupt is generated to let the CPU know that transmission is over. Now, the Clocked Serial I/O is in waiting condition.

By issuing Transmission Enable Command on completion of data reception the data received just before can be transmit.

Precaution for data transmission and reception

(1) In case that eight bit character length is specified, the data read and date write is limited to the lower byte of the shift register . When the data is written two times erroneously, the first data is lost and the only second data is valid.

In case that the character size which is longer than or equal to nine bit, data read and data write are done with two times access of the upper and lower bytes. Both cases are done in order of the lower bytes and the upper bytes. Pay attention that the register which selects the upper byte or the lower byte is common in write and read. For example, data read after the lower byte of transmit data is written, is from the upper byte. As this register is cleared (selects the lower byte) by reset, mode/command write, and status readout, it is recommended that status read is done before writing transmit data and reading received data.

In case that the character size that is longer than nine bit to sixteen bit, a few bits of upper eight bit of readout data are invalid. For example, in case that ten bit character length is specified, the bits from D7 to D2 of the upper eight bit of read data are invalid.

(2) As there is one shift register for Clocked Serial I/O, transmission and reception are not done at the same time (half-duplex communication). As there is not a buffer register, the transmit data can be written only in

waiting condition.

(3) Data read out while transmitting or receiving, is not guaranteed.

(4) As the content of the shift register is destroyed after transmitting data, set the data again and transmit even in case of transmitting the same data as the previous data.

10.7 Transmission and reception timing

When internal serial clock is selected

The figures 10-3 and 10-4 show the case of mode setting of internal serial clock, twelve bit character length, SST=0, and LSB top. In case of data transmission, after transmit data are written two times, write Transmission Enable Command. Then, BUSY_ bit of Status Register turns to be 0. The initial value of SCK output is H. When SCK output becomes L, LSB bit of the transmit data comes out from TXS output. Every time the serial clock from SCK output goes from H to L, the next bit of transmit data comes out from TXS output. After twelve bit data are transmitted, the serial clock stays H until next transmission. At the end of data transmission, an internal interrupt is generated, BUSY_ bit of Status Register turns to be 1 and the Clocked Serial I/O goes into waiting condition.

Data reception is done in a similar manner except that receive data are sampled at the rising edge of serial clock and shifted into an internal register.

When external serial clock is selected

The figures 10-5 and 10-6 show the case of mode setting of external serial clock, eight bit character length, SST=1, and MSB top. SCK input must be L. In case of data transmission, after transmit data are written one time, write Transmission Enable Command. Then BUSY_ bit of Status Register turns to be 0. The external serial clock is input from SCK input.

Every time the serial clock goes from L to H, the next bit of transmit data are shifted out from TXS output in the order of MSB first. After eight bit data are transmitted, an internal interrupt is generated, BUSY_ bit of Status Register turns to be 1, and the Clocked Serial I/O goes into waiting condition. Set the external serial clock input L after eight times of falling edge of the serial clock or after BUSY_ bit of Status Register turns to be 1.

In case of data reception, write Transmission Enable Command in the waiting condition. (SCK input must be L.) BUSY_ bit of Status Register turns to be 0. Input the serial clock to SCK pin. At the falling edge of the serial clock, the data on RXS pin are sampled and shifted into the internal register. When eight bit data is received, an internal interrupt is generated, BUSY_ bit of Status Register turns to be 1, and the Clocked Serial I/O goes to waiting condition. Set the external serial clock input L after eight times of rising edge of the serial clock or after BUSY_ bit of Status Register turns to be 1.

Precaution on timings

(1) In case of that the external serial clock is selected, the excess pulses of the serial clock are neglected.

(2) In case of that the external serial clock is selected, the pulses of the serial clock input in waiting condition (when BUSY_ bit of Status Register is 1.) are neglected.

(3) In case of that the external serial clock is selected, it is recommended that the serial clock should be started after BUSY_ bit of Status Register turns to be 0.

10.8 Reset

By setting RESET pin L, the mode is set as the external serial clock, eight bit character length, SST = 0, MSB first, and the Clocked Serial I/O goes into the waiting condition.

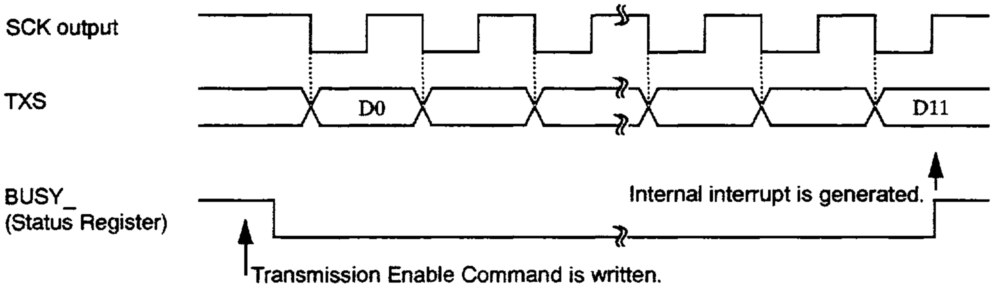

SCK output
TXS
D0
D11
BUSY_
(Status Register)
Internal interrupt is generated.
Transmission Enable Command is written.

Figure 10-3 Transmission timing when internal serial clock, 12 bit character length, SST=0, and LSB top are selected.

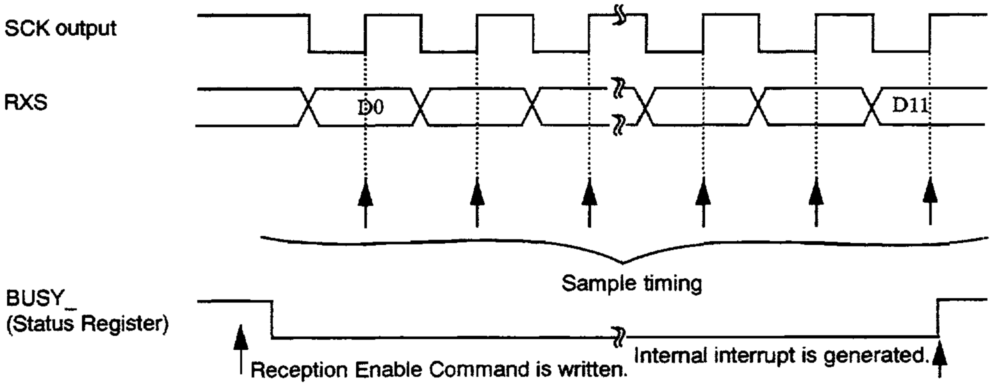

SCK output
RXS
D0
D11
Sample timing
BUSY_
(Status Register)
Reception Enable Command is written.
Internal interrupt is generated.

Figure 10-4 Reception timing when internal serial clock, 12 bit character length, SST=0, and LSB top are selected.

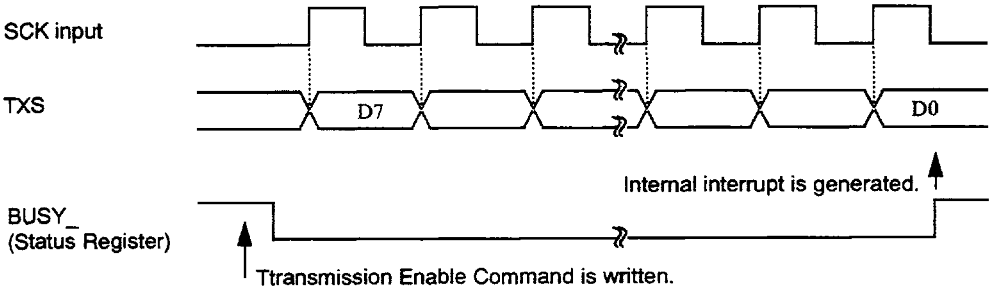

SCK input
TXS
D7
D0
Internal interrupt is generated.
BUSY_
(Status Register)
Ttransmission Enable Command is written.

Figure 10-5 Transmission timing when external serial clock, 8 bit character length, SST=1, and MSB top are selected.

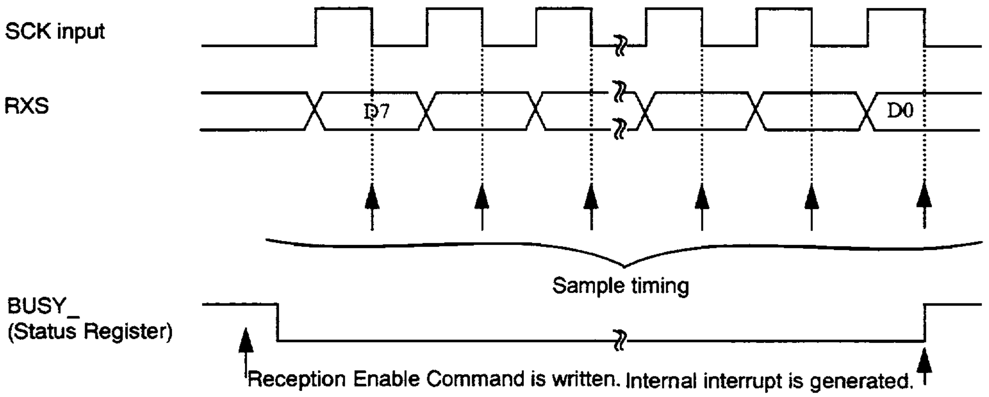

SCK input
RXS
D7
D0
Sample timing
BUSY_
(Status Register)
Reception Enable Command is written.
Internal interrupt is generated.

Figure 10-6 Reception timing when external serial clock, 8 bit character length, SST=1, and MSB top are selected.

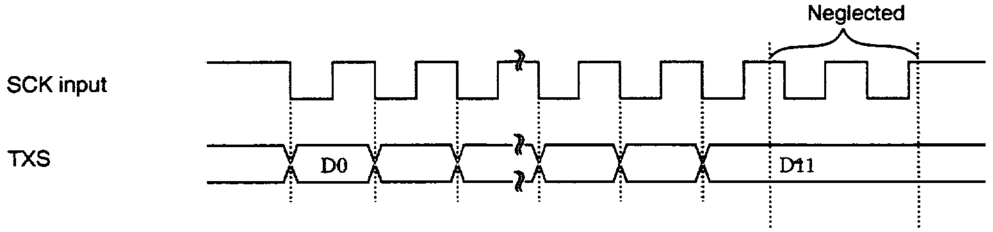


Figure 10-7 Example of the excess pulses of external serial clocks (at transmission)

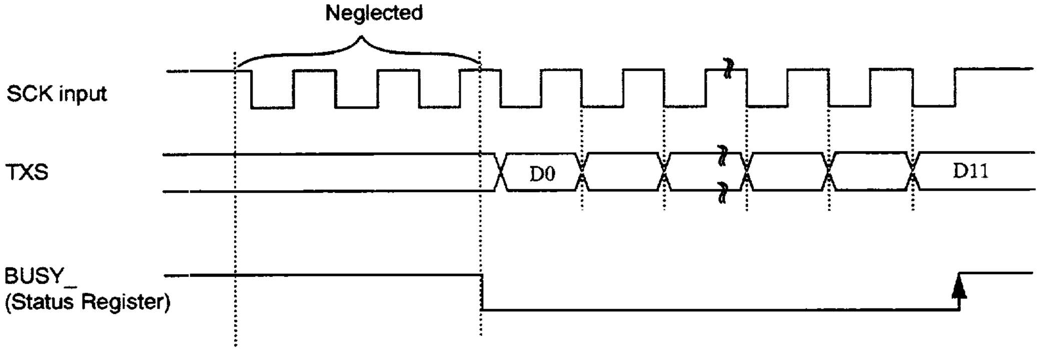


Figure 10-8 Example of external serial clocks which are input while waiting (at transmission)

11. Parallel Ports

11.1 General Description

The parallel ports on KL5C80A16 is KP67 macro cell. It has 32 ports. Each port except Port 0 can be set as input or output by each one bit. This is done by setting of Direction Control Registers. It supports Bit Control Command.

Features:

·The number of ports: 32
·Input/output setting unit: by each port
·Bit set/reset command available

11.2 Block Diagram

The parallel port block diagram and I/O register mapping are shown here.

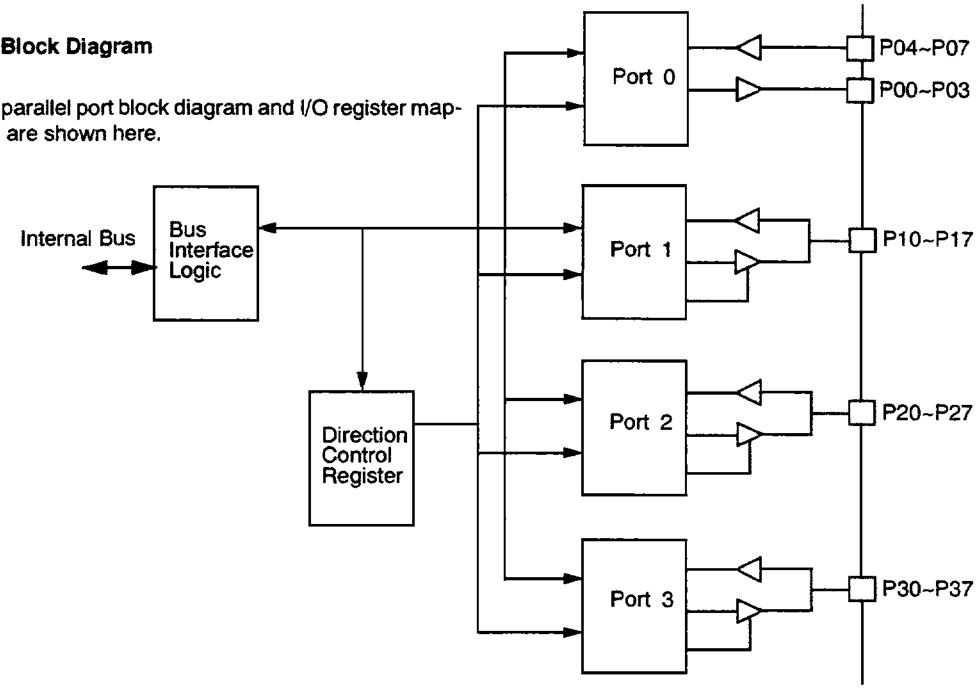


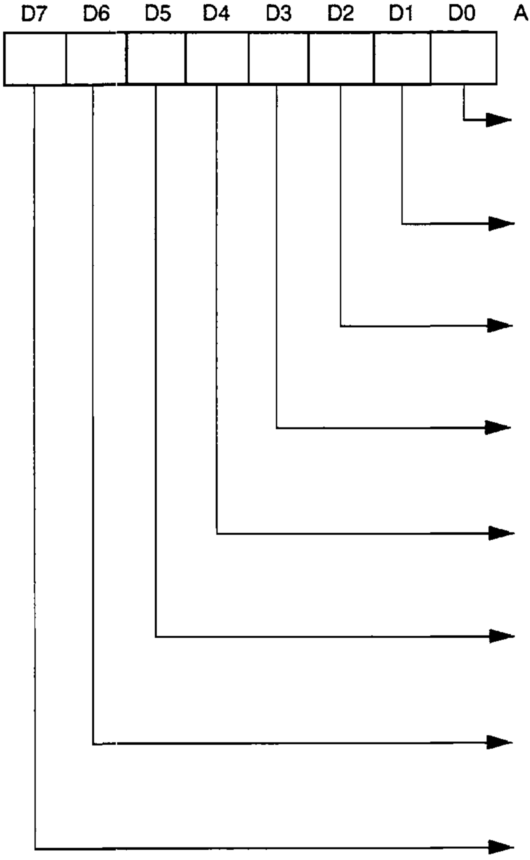
Table 11-1 I/O Register Mapping

I/O	Write	Read
38H	Port 0	Port 0
39H	Bit Control Command	Port 0 Direction Control Register(fixed)
3AH	Port 1	Port 1
3BH	Port 1 Direction Control Register	Port 1 Direction Control Register
3CH	Port 2	Port 2
3DH	Port 2 Direction Control Register	Port 2 Direction Control Register
3EH	Port 3	Port 3
3FH	Port 3 Direction Control Register	Port 3 Direction Control Register

11.3 Input Output Control & Bit Control Command

The upper four bit of port 0 is always input ports, and the lower four bit of port 0 is always output ports. Port 1, 2, 3 can be used as input ports or output ports. Input/ output direction of port 1, 2, 3 is set by Port Direction Control Registers mapped on 3BH, 3DH, 3EH. The port directions of port 1, 2, 3 is controlled by each one port. Bit Control Command can set or reset only one bit of 32 ports.

Port Direction Control Register



A[7:0]= 3BH : Port 1 Direction Control Register
3DH : Port 2 Direction Control Register
3FH : Port 3 Direction Control Register

PnEN[0]
1:Set bit 0 of port n as output
0:Set bit 0 of port n as input

PnEN[1]
1:Set bit1 of port n as output
0:Set bit1 of port n as input

PnEN[2]
1:Set bit 2 of port n as output
0:Set bit 2 of port n as input

PnEN[3]
1:Set bit 3 of port n as output
0:Set bit 3 of port n as input

PnEN[4]
1:Set bit 4 of port n as output
0:Set bit 4 of port n as input

PnEN[5]
1:Set bit 5 of port n as output
0:Set bit 5 of port n as input

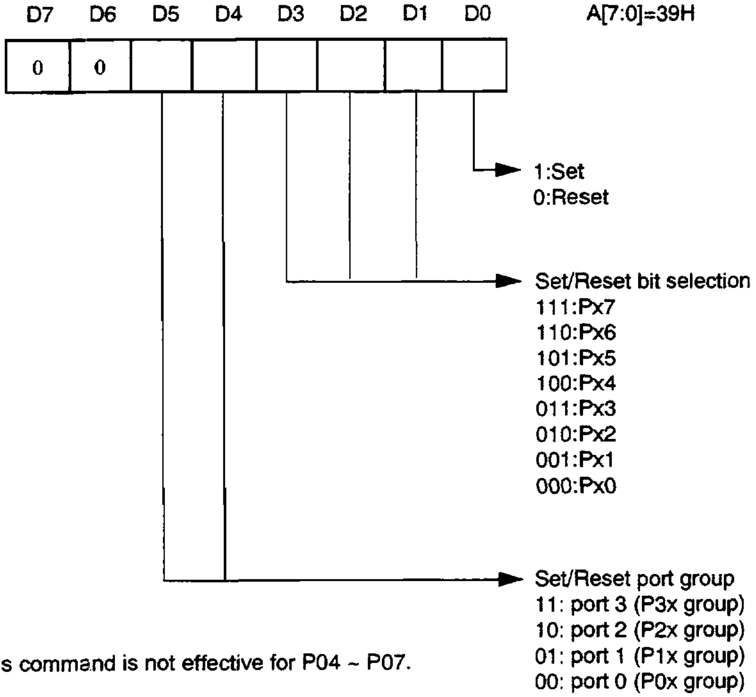
PnEN[6]
1:Set bit 6 of port n as output
0:Set bit 6 of port n as input

PnEN[7]
1:Set bit 7 of port n as output
0:Set bit 7 of port n as input

n=1 (for A[7:0]=3BH)
n=2 (for A[7:0]=3DH)
n=3 (for A[7:0]=3FH)

Bit Control Command

Bit Control Command can set or reset only one bit of 32 ports.



(Note) This command is not effective for P04 ~ P07.

11.4 Input and Output Operation

Input Operation

The data input from the ports set as input ports are read from the data bus.

Output Operation

The data written into the port register are output from the ports set as outputs.

11.5 Reset

On reset, the registers below here are initialized.

Port 3~0	All output registers	reset
Port 1	Direction Control Register	reset
Port 2	Direction Control Register	reset
Port 3	Direction Control Register	reset

11.6 Precaution

The port multiplexed with other signal must be set as an input port when the pin is not used as the port.

12. Operation Mode

The operation mode of this chip is specified by the external input pin, BFMOD and the system control registers (SCR).

12.1 Setting operation mode with the input pins

BFMOD =	1	:	Bug Finder Boot-on-RAM mode
	0	:	normal mode

Normal mode
Use this mode normally.

Bug Finder Boot-on-RAM mode
In the Bug Finder Boot-on-RAM mode, the connection of a Bug Finder adapter to the BFSIO pin allows the Bug Finder to start in external RAM. Bug Finder is a kind of low cost ICE dedicated for KC80 MCU family. For more detail, refer to Bug Finder manuals.

12.2 System control registers

There are five System Control Registers, SCR0 ~ SCR4 for control of KL5C80A16 operation modes.

SCR0 (I/O address= '1BH')

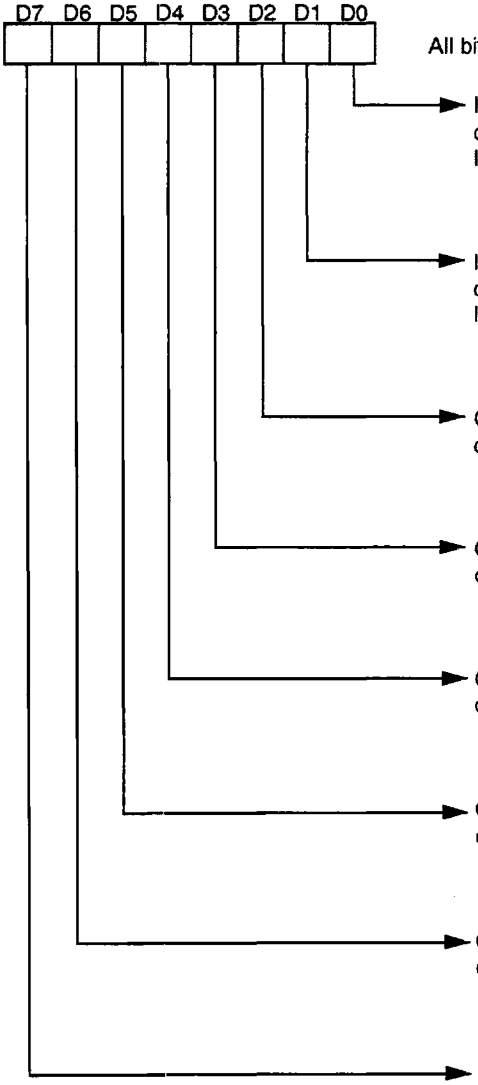
D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0				

All bit is cleared on reset.

D3–D0 → DRAM bank number
Write DRAM bank number here when DRAM is mapped on 80000H~BFFFFH expanded address space.
Use the number of bits needed to specify banks as follows.
Up to sixteen banks can be handled.

1M byte mapped:	Use two chips of 4M bit 4 bit bus DRAM, then use D<1:0> to control four banks.
4M byte mapped:	Use two chips of 16M bit 4 bit bus DRAM, then use D<3:0> to control sixteen banks.

SCR1 (I/O address = 1CH)



All bit is cleared on reset.

In case that the input from the pin 89 is used as IR input of the interrupt controller, connect the following signal to IR input of the interrupt controller.
0 : non-inverted input from pin 89
1 : inverted input from pin 89

In case that the input from the pin 88 is used as IR input of the interrupt controller, connect the following signal to IR input of the interrupt controller.
0 : non-inverted input from pin 88
1 : inverted input from pin 88

Connect the following signal to IR[14] input of the interrupt controller.
0 : non-inverted input from pin 87
1 : inverted input from pin 87

Connect the following signal to IR[15] input of the interrupt controller.
0 : non-inverted input from pin 86
1 : inverted input from pin 86

Connect the following signal to IR[0] input of the interrupt controller.
0 : input from pin 89
1 : the interrupt request from Timer/Counter channel 2*

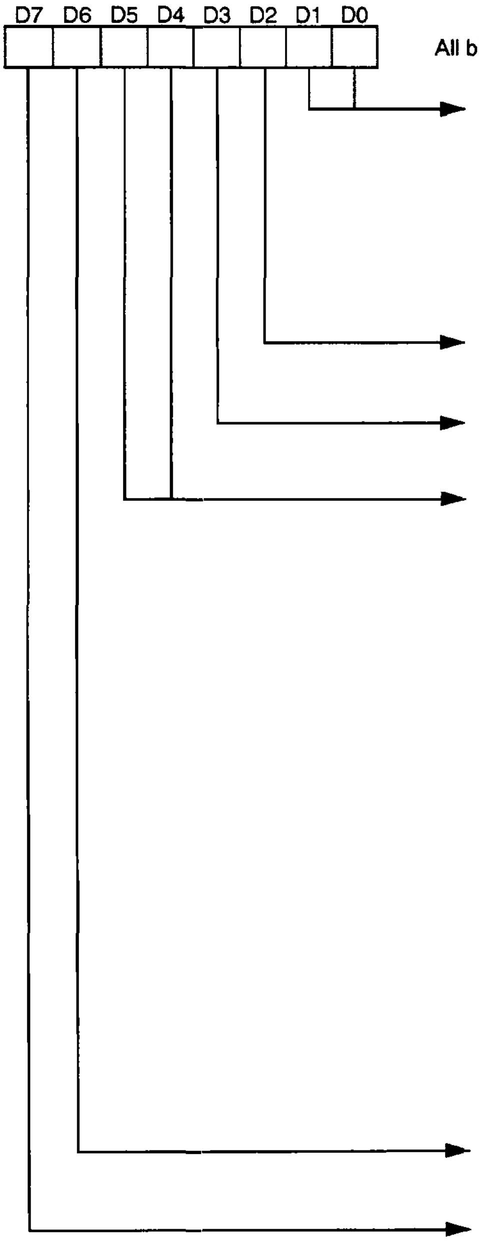
Connect the following signal to IR[1] input of the interrupt controller.
0 : input from pin 88
1 : the interrupt request from Timer/Counter channel 3*

Connect the following signal to IR[2] input of the interrupt controller.
0 : input from pin 89
1 : the interrupt request from Clocked Serial I/O channel 1*

Connect the following signal to IR[5] input of the interrupt controller.
0 : input from pin 88
1 : BDET + ERINT outputs from UART channel 1

*(NOTE) Use the IR input of the interrupt controller with edge mode in this case.

SCR2(I/O address = 1DH)



All bit is cleared on reset.

D1 D0:

00 : Select pin 84 as DREQ1.
Pin 92 functions as P16.
01 : Select pin 84 as DREQ1.
Pin 92 functions as DACK1_.
10 : Select TXRDY0 of UART as DERQ1.
Pin 92 functions as P16.
11 : Select RXRDY0 of UART as DERQ1.
Pin 92 functions as P16.

D2:

0 : Pin 91 functions as P17.
1 : Pin 91 functions as EXBACK_.

D3:

0 : Pin 5 functions as PALAT.
1 : Pin 5 functions as DACK0_.

D5 D4:

00 : Pin 81 functions as P30.
Pin 80 functions as P31.
Pin 77 functions as P32.
"L" is input to CTS1_ of UART.
"L" is input to DSR1_ of UART.

01 : Pin 81 functions as P30.
Pin 80 functions as P31.
Pin 77 functions as TXD1 of UART.
"L" is input to CTS1_ of UART.
"L" is input to DSR1_ of UART.

10 : Pin 81 functions as P30.
Pin 80 functions as RTS1_ of UART.
Pin 77 functions as TXD1 of UART.
Pin 74 functions as CTS1_ of UART.
"L" is input to DSR1_ of UART.

11 : Pin 81 functions as DTR1_ of UART.
Pin 80 functions as RTS1_ of UART.
Pin 77 functions as TXD1 of UART.
Pin 74 functions as CTS1_ of UART.
Pin 75 functions as DSR1_ of UART.

D6:

0 : Pin 94 functions as P14.
1 : Pin 94 functions as RXRDY1 of UART.

D7:

0 : Pin 93 functions as P15.
1 : Pin 93 functions as TXRDY1 of UART.

SCR3(I/O address= 1EH)

D7	D6	D5	D4	D3	D2	D1	D0
				0			

All bit is cleared on reset.

Refresh Rate

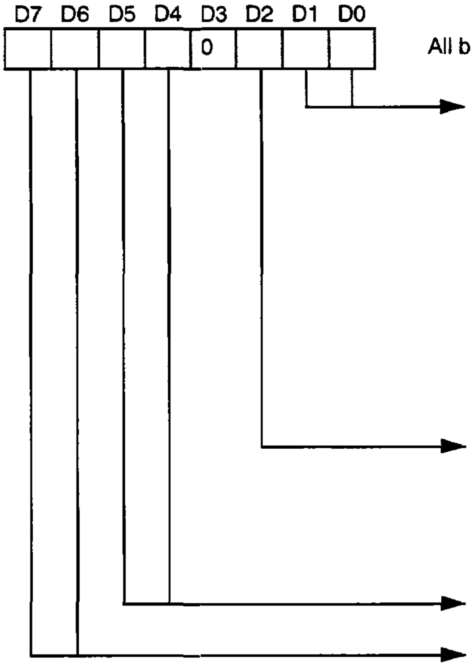
D2	D1	D0	Refresh Interval
0	0	0	128 clock
0	0	1	88 clock
0	1	0	64 clock
0	1	1	44 clock
1	0	0	32 clock
1	0	1	22 clock
1	1	0	16 clock
1	1	1	11 clock

When D5 is '1', address space, C0000H~FFFFFH is
0: used as SRAM area
1: mapped to the highest bank of DRAM area
When D5 is '0', this bit is meaningless.

Address space, 80000H~BFFFFH is used as
0: SRAM area.
Pin 96 functions as P13.
Pin 97 functions as P12.
Pin 98 functions as P11.
1: DRAM area.
Pin 96 functions as UASEL_.
Pin 97 functions as CAS_.
Pin 98 functions as RAS_.

Connect the following signal to NMI_ of CPU.
00 : "H"(NMI_ is always inactive.)
01 : The interrupt request from Timer/Counter channel 3
10 : Pin 82
11 : The interrupt request from Timer/Counter channel 3 + pin 82

SCR4(I/O address = 1FH)



All bit is cleared on reset.

00 : Pin 20 functions as P03.
Pin 21 functions as P02.
Pin 22 functions as P01.
01 : Pin 20 functions as P03.
Pin 21 functions as P02.
Pin 22 functions as OUT1 output of Timer/Counter.
10 : Pin 20 functions as P03.
Pin 21 functions as OUT2 output of Timer/Counter.
Pin 22 functions as OUT1 output of Timer/Counter.
11 : Pin 20 functions as OUT3 output of Timer/Counter.
Pin 21 functions as OUT2 output of Timer/Counter.
Pin 22 functions as OUT1 output of Timer/Counter.

0 : Pin 76 functions as P33.
Pin 99 functions as P10.
1 : Pin 76 functions as TXS1 of Clocked Serial I/O.
Pin 99 functions as SCK1 of Clocked Serial I/O.

Memory wait state control (See Table 12-1)

External I/O wait state control (See Table 12-2)

Table 12-1 Memory wait state control

D5	D4	address: 00000H~7FFFFH	address: 80000H~FFFFFH
0	X	1 wait	1 wait*
1	0	1 wait	0 wait*
1	1	0 wait	0 wait*

* In case of use of DRAM, the number of waits inserted depends on DRAM controller in External Bus Interface Unit.

Table 12-2 External I/O wait state control

D7	D6	External I/O
0	0	1 wait (3 clock/bus cycle)
0	1	2 wait (4 clock/bus cycle)
1	0	3 wait (5 clock/bus cycle)
1	1	4 wait (6 clock/bus cycle)

In case of 4 wait, the starting edge of EIORD_/EIOWR_ is delayed by 1/2clock to other cases.

12.3 How to use pins

How to use pin 5

Pin 5 is fixed to output. The function of this pin is selected by SCR2.

How to use pin 12, 13, 17, 18

Pin 12, 13, 17, 18 are fixed to input. Be careful that the same signal from the same external input is input to P07 and GATE3, P06 and GATE2, P05 and GATE1 and TRXC1, P04 and GATE0 and TRXC0. If you do not use one of these functions, disable the function by mode setting in its corresponding block.

For example, if pin 17 is used as TRXC1, use Timer/Counter channel 1 without gate input function, and if pin 18 is used as GATE0, use the baud rate generator to generate the transceiver/receiver clock of UART channel 0.

How to use pin 20, 21, 22

Pin 20, 21, 22 are fixed to output. The functions of these pins are selected by SCR4.

How to use pin 72, 73

The input/output directions of pin 72, 73 are controlled by the port direction control register of the parallel port block, regardless of the system control registers. If pin 72 and 73 are used as RXS1 and RXD1, P37 and P36 should be set as input. In this case, the same external signal is input to each RXS1 and P37, RXD1 and P36. If pin 72 and 73 are used as output ports, RXS1 and RXD1 should not be used. In this case, the signals output from parallel ports are applied to RXS1 and RXD1.

How to use pin 74, 75

The input/output directions of pin 74, 75 are controlled by the port direction control register of the parallel port block, regardless of the system control registers. If pin 74 and 75 are used as CTS1_ and DSR1_, P35 and P34 should be set as input and bit D<5:4> of SCR2 should be set to 10 or 11. In this case, the same external signal is input to each CTS1_ and P35, DSR1_ and P34.

Be careful to use UART channel 1, if pin 74 and 75 are used as output ports, and if D<5:4> of SCR2 is set to 10 or 11, as the signals output from parallel ports are applied to CTS1_ and DSR1_.

How to use pin 76

The input/output direction of pin 76 is controlled by the port direction control register of the parallel port block, regardless of the system control registers. If pin 76 is used as TXS1, P33 should be set as output and bit D2 of SCR4 should be set to 1. In this case, the output from TXS1 is applied to P33. If P33 is readout, the output from TXS1 at this time can be read. (See chapter 11.)

If pin 76 is used parallel ports, the channel 1 of the Clocked Serial I/O should not be used.

How to use pin 77, 80, 81

The input/output directions of pin 77, 80, 81 are controlled by D<5:4> of SCR2. When these pins are used as parallel ports, the input/output directions of pins should be controlled by the port direction control register of the parallel port block.

If these pins are not used as parallel ports, the input/output directions are determined to output direction by SCR2 settings automatically.

How to use pin 82

The input/output direction of pin 82 is controlled by the port direction control register of the parallel port block, regardless of the system control registers. If pin 82 is used as NMI_, P27 should be an input. In this case, if D<7:6> of SCR3 are set to 10, the same signal is input to P27 as NMI_.

If pin 82 is used as parallel port, D<7> of SCR3 should be 0. Then NMI_ is disconnected from pin 82.

How to use pin 83, 84, 85

The input/output directions of pin 83, 84, 85 are controlled by the port direction control register of the parallel port block, regardless of the system control registers. If pin 83, 84, 85 are not used as parallel ports, the corresponding ports should be set as input. In this case, the same external signal is input to each EXBACK_ and P26, DREQ0 and P24. If D<1:0> of SCR2 are set to 00 or 01, the same signal is input to P25 and DREQ1.

If these pins are used as parallel ports, pin 84 should be disconnected from DREQ1 by setting D<1> of SCR2. In addition to it, mode of the DMA controller should be also set in a way that those pin does not affect the functions of the DMA controller.

How to use pin 86, 87, 88, 89

The input/output directions of pin 86, 87, 88, 89 are controlled by the port direction control register of the parallel port block, regardless of the system control registers. If pin 86, 87, 88, 89 are not used as parallel ports, the corresponding ports should be set as input. In this case, the same external signal is input to each IR[15] and P23, IR[14] and P22. In case that IR[5] and IR[1] are connected to pin 88 by setting D7 and D5 of SCR1, the same external signal is input to each IR[5], IR[1] and P21. In case that IR[2] and IR[0] are connected to pin 89 by setting D6 and D4 of SCR1, the same external signal is input to each IR[2], IR[0] and P20. The inverted input these pins can be chosen for these IR inputs depending on D<3:0> of SCR1.

If these pins are used as parallel ports, the setting that doesn't affect the interrupt controller is recommended such that the corresponding IR inputs are masked or disconnected from pins by SCR1 initialization.

How to use pin 91, 92, 93, 94, 96, 97, 98, 99

The input/output directions of pin 91, 92, 93, 94, 96, 97, 98, 99 depends on the system control registers. If they are used as parallel ports, the input/output directions are set by the port direction control register of the parallel port block.

In case that the system control registers are set in a way that they are not used as parallel ports, they are automatically set as outputs except pin 99. As for pin 99, the input/output direction is controlled by the Clocked Serial I/O internal register. If the pin is set as input, the same signal is input to both P10 and SCK1 input of the Clocked Serial I/O.

13. Address Mapping

Internal I/O mapping

Table 13-1. Internal I/O Mapping

I/O address	Block	Write cycle	Read cycle
00H	KC82(MMU)	BBR1 (boundary/base register 1)	BBR1 (boundary/base register 1)
01H		BR1 (base register 1)	BR1 (base register 1)
02H		BBR2 (boundary/base register 2)	BBR2 (boundary/base register 2)
03H		BR2 (base register 2)	BR2 (base register 2)
04H		BBR3 (boundary/base register 3)	BBR3 (boundary/base register 3)
05H		BR3 (base register 3)	BR3 (base register 3)
06H		BBR4 (boundary/base register 4)	BBR4 (boundary/base register 4)
07H		BR4 (base register 4)	BR4 (base register 4)
08H~0FH	reserved for Kawasaki Steel Corp.		
10H	DMA Controller	channel 0 B-PAR	channel 0 C-PAR
11H		channel 0 B-SAR	channel 0 C-SAR
12H		channel 0 B-BCR	channel 0 C-BCR
13H		channel 0 CR	channel 0 SR
14H		channel 1 B-PAR	channel 1 C-PAR
15H		channel 1 B-SAR	channel 1 C-SAR
16H		channel 1 B-BCR	channel 1 C-BCR
17H		channel 1 CR	channel 1 SR
18H	System Control Register	reserved for Kawasaki Steel Corp.	reserved for Kawasaki Steel Corp.
19H		reserved for Kawasaki Steel Corp.	reserved for Kawasaki Steel Corp.
1AH		reserved for Kawasaki Steel Corp.	reserved for Kawasaki Steel Corp.
1BH		SCR0	SCR0
1CH		SCR1	SCR1
1DH		SCR2	SCR2
1EH		SCR3	SCR3
1FH		SCR4	SCR4
20H	Timer/Counter	channel 0 counter	channel 0 counter
21H		channel 0 control word	channel 0 status
22H		channel 1 counter	channel 1 counter
23H		channel 1 control word	channel 1 status
24H		channel 2 counter	channel 2 counter
25H		channel 2 control word	channel 2 status
26H		channel 3 counter	channel 3 counter
27H		channel 3 control word	channel 3 status

NOTE: I/O address decode for those internal I/O is eight bit decode. The upper 8 bits, A15~A8 are neglected.

Table 13-1. Internal I/O Mapping (continued)

I/O address	Block	Write cycle	Read cycle
28H	UART	RATE	RATE
29H		reserved for Kawasaki Steel Corp.	reserved for Kawasaki Steel Corp.
2AH		channel 0 transmission data	channel 0 received data/extended status A
2BH		channel 0 mode/command	channel 0 status/extended status B
2CH		channel 1 transmission data	channel 1 received data/extended status A
2DH		channel 1 mode/command	channel 1 status/extended status B
2EH		reserved for Kawasaki Steel Corp.	reserved for Kawasaki Steel Corp.
2FH		reserved for Kawasaki Steel Corp.	reserved for Kawasaki Steel Corp.
30H	Clocked Serial I/O	channel 0 transmission data	channel 0 received data
31H		channel 0 command/mode	channel 0 status
32H		channel 1 transmission data	channel 1 received data
33H		channel 1 command/mode	channel 1 status
34H	Interrupt Controller	LERL／PGRL	ISRL
35H		LERH／PGRH	ISRH
36H		IMRL	IMRL
37H		IVR／IMRH	IMRH
38H	Parallel Ports	Port0	Port0
39H		bit control command	Port0 direction control register(fixed)
3AH		Port1	Port1
3BH		Port1 direction control register	Port1 direction control register
3CH		Port2	Port2
3DH		Port2 direction control register	Port2 direction control register
3EH		Port3	Port3
3FH		Port3 direction control register	Port3 direction control register

NOTE: I/O address decode for those intemal I/O is eight bit decode. The upper 8 bits, A15~A8 are neglected.

Memory mapping

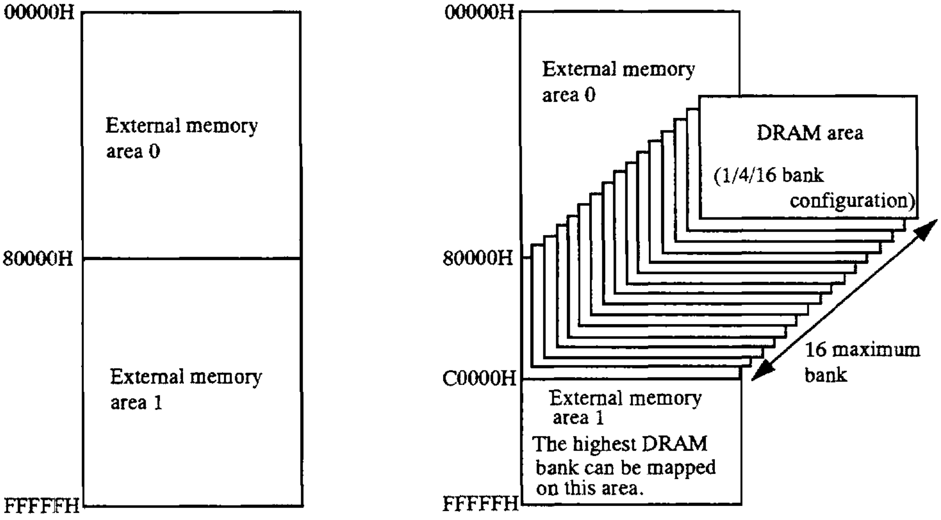
In case that the internal DRAM controller is not used.

External memory area1	Physical address space	80000H~FFFFFH (512Kbyte)
External memory area0	Physical address space	00000H~7FFFFH (512Kbyte)

In case that the internal DRAM controller is used.

External memory area1	Physical address space	C0000H~FFFFFH (256Kbyte) (The highest DRAM bank can be mapped on this area.)
DRAM area	Physical address space	80000H~BFFFFH (256Kbyte) (Three clock bus cycle is always applied regardless of wait setting of the external memory area1.)
External memory area0	Physical address space	00000H~7FFFFH (512Kbyte)

Different wait setting for external memory area 0 and 1 is possible with the system control register.



In case that the DRAM is not used

In case that the DRAM is used

Figure13-1 Memory mapping

14. Oscillation Circuit

14.1 General description

The KL5C80A16 contains an oscillation buffer to generate system clock. The system clock in the chip is a signal which has a divide-by-2 frequency of the signal generated by this oscillation circuit.

14.2 Circuit structure

To generate system clock, the KL5C80A16 can be connected with a crystal oscillator (or ceramic oscillator), a feedback resistor and a condenser as external parts to form an oscillation circuit as shown in the right figure. The constants of external parts are dependent on oscillator, substrate boards and so on to be used. Use the values recommended by oscillator manufacturers for optimum values of external parts constants. The system clock in the chip is a signal which has a divide-by-2 frequency of the signal generated by this oscillation circuit. A frequency divide circuit is built in the chip.

Table 14-1. Oscillation Frequency

Oscillation frequency	Operation power supply voltage
2~20 MHz	5V ± 5%, 5V ± 10%

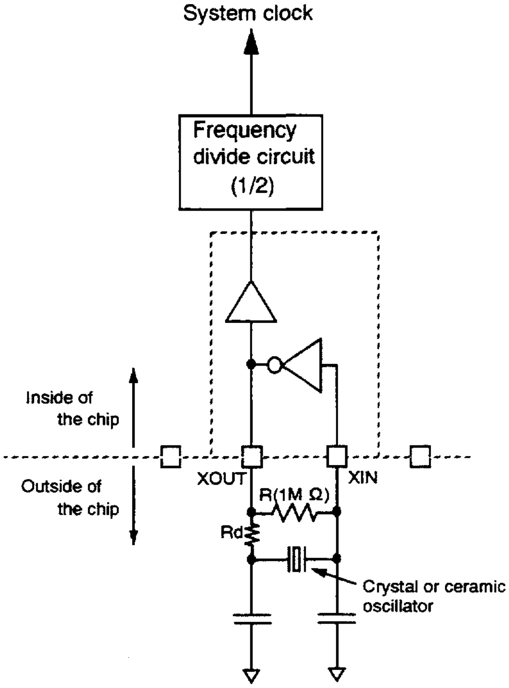


Figure 14-1. Oscillation Circuit

Notes

Be sure to use the CLK pin when the system clock is output. Be sure not to output its signal directly from XIN and XOUT.

Be sure to input a signal with twice the system clock frequency from XIN when the system clock is input from the outside. In this case, make a parasitic capacity of XOUT as low as possible.

Table 14-2. Recommended Range of External Circuit Constants

	Rd	CI, CO
Crystal	100 ~ 800 Ω	5 ~ 30pF
Ceramic	30 ~ 300 Ω	5 ~ 100pF

15. Electrical Characteristics

15.1 Absolute maximum ratings

Table 15-1 Absolute Maximum Ratings (with respect to GND)

Item	Symbol	Rating	Unit
Power supply voltage	V_{DD}	- 0. 6 ～ + 7. 0	V
Input voltage	V_{IN}	- 0. 6 ～ V_{DD} + 0. 6	V
Storage temperature	T_{STG}	- 40 ～ + 125	℃

15.2 D.C. Characteristics (5V ± 10%)

Table 15-2 Recommended Operation Conditions

Item	Symbol	Rating	Unit
Power supply voltage	V_{DD}	4.5 ～ 5.5	V
Ambient temperature	T_A	0 ～ +70	℃

Table 15-3 Electrical Characteristics (under recommended operation conditions)

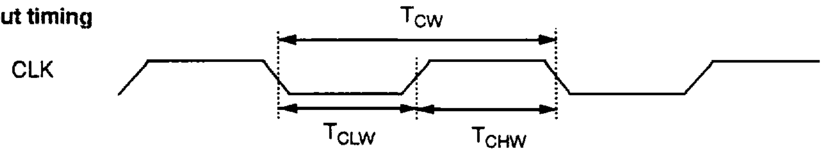
Item	Symbol	Rating			Unit	Test condition
		Min.	Typ.	Max.		
Input voltage	V_{IH}	3. 6		VDD	V	
(all input pins except RESET_)	V_{IL}	GND		1. 4	V	
RESET_ input pin	V_+	2. 4		4. 0	V	
(Schmitt trigger input)	V_-	0. 9		2. 3	V	
	V_h	0. 9			V	
Output voltage	V_{OH}	3. 5			V	I_{OH} = - 4mA or - 6mA
	V_{OL}			0. 4	V	I_{OL} = 4mA or 6mA
Output current	I_{OUT}			± 6	mA	(note-1)
				± 4	mA	(note-2)
Input leakage current	I_{IL}	- 10			μ A	V_{IN} = GND
	I_{IH}			10	μ A	V_{IN} = V_{DD}
Output leakage current	I_{OZ}	- 10		10	μ A	at high impedance output
Pull-up current	I_{PU}	20	95	250	μ A	V_{IN} = GND
Pull-down current	I_{PD}	20	95	250	μ A	V_{IN} = V_{DD}
Standby supply current	I_{DDS}		1. 0*	100	μ A	CLK stop
Power supply current	I_{DDOP}		30*		mA	f (CLK) = 10MHz

* at T_A=25 ℃

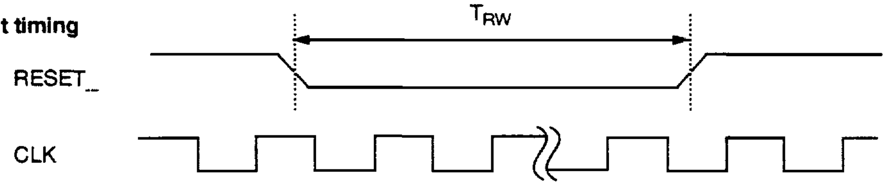
note-1) A[19:0], D[7:0], EMRD_, EMWR_, EIORD_, EIOWR_ and CLK output
note-2) Output pins except pins at note-1

15.3 A.C. Characteristics

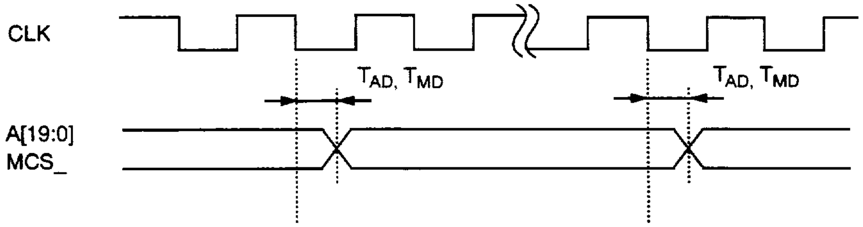
Clock output timing



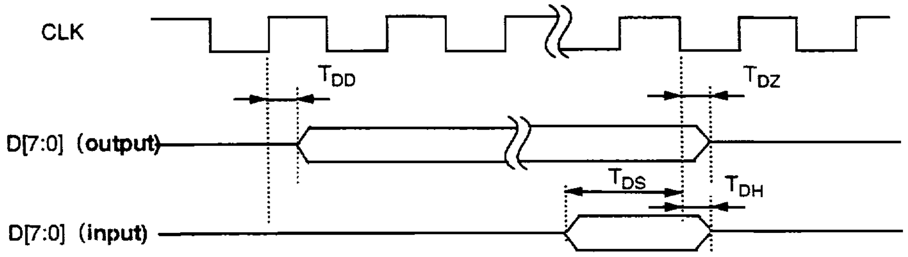
Reset input timing



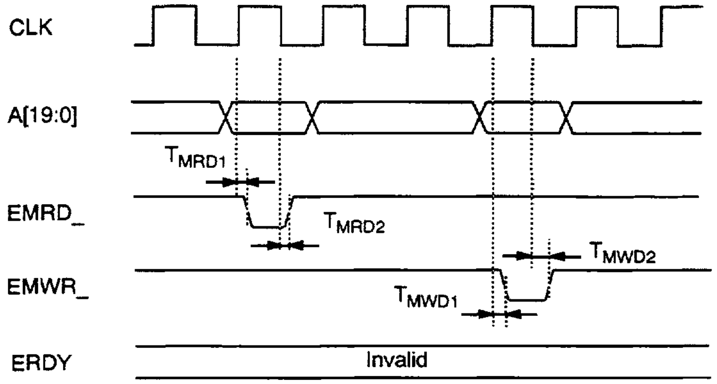
Address output timing



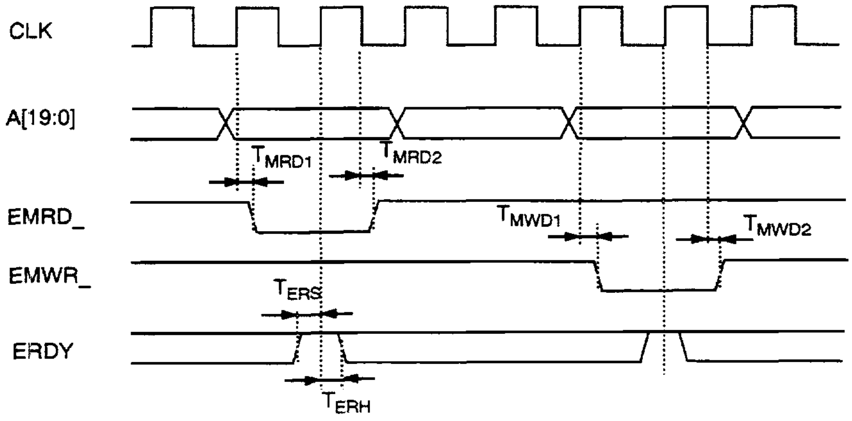
Data I/O timing



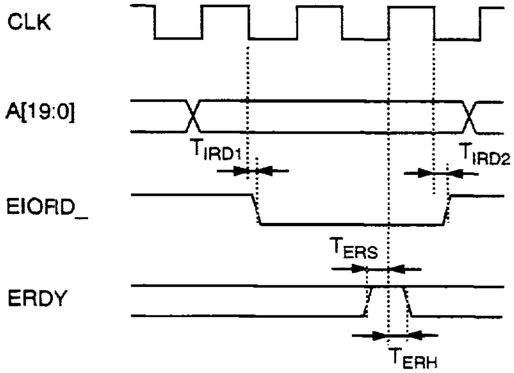
External memory access cycle (0 wait state)



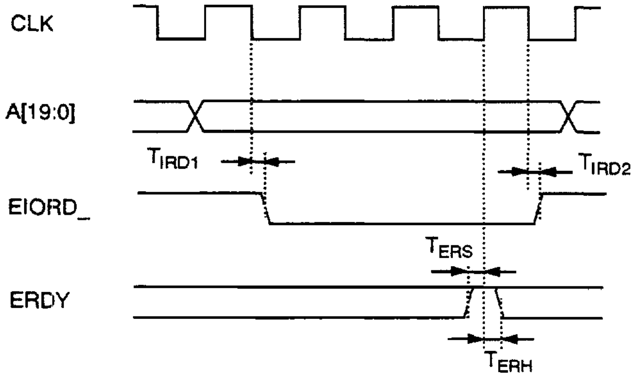
External memory access cycle (1 wait state)



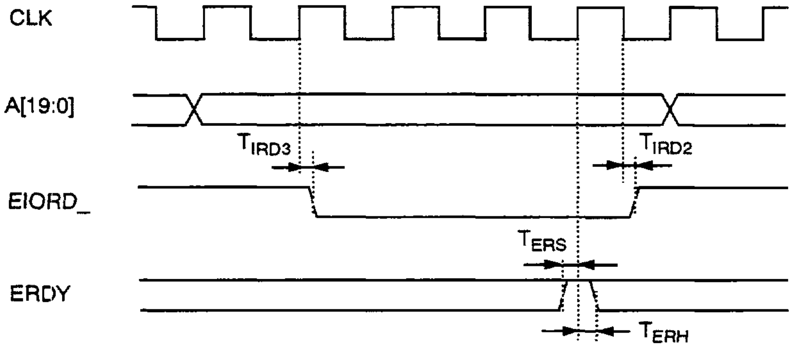
External I/O read cycle (1 wait state)



External I/O read cycle (2 wait states)

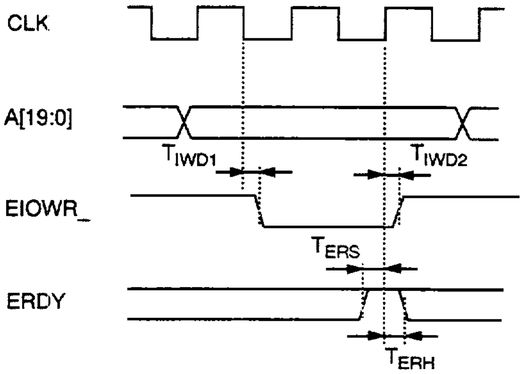


External I/O read cycle (4 wait states)

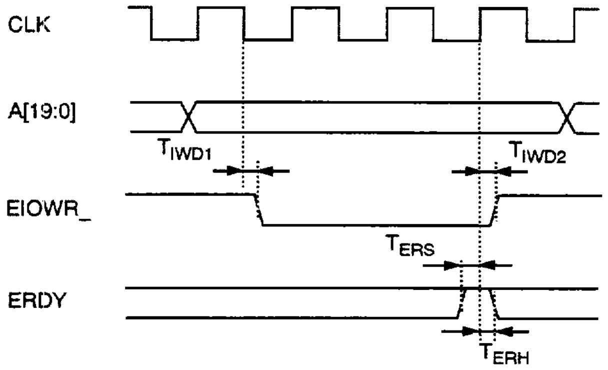


※ In case that 4 wait states external I/O access is selected by SCR4, the starting edge of EIORD_ signal is delayed by 1/2 clock cycle.

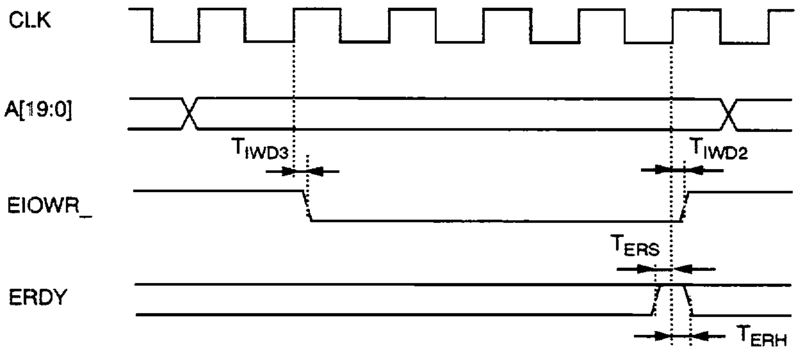
External I/O write cycle (1 wait state)



External I/O write cycle (2 wait states)

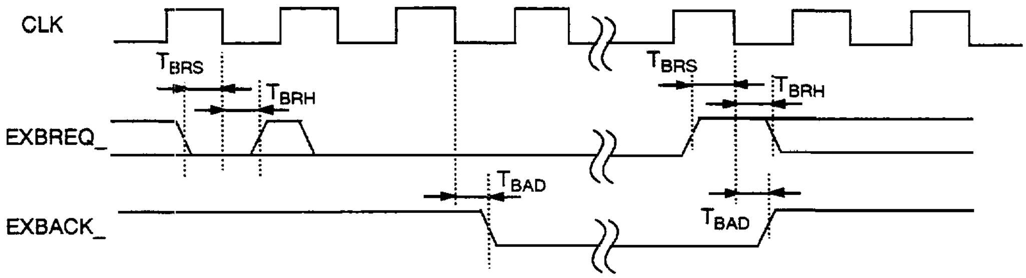


External I/O write cycle (4 wait states)

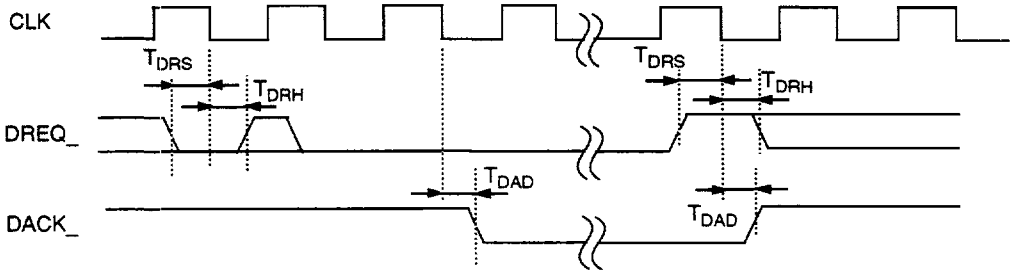


※ In case that 4 wait states external I/O access is selected by SCR4, the starting edge of EIOWR_ signal is delayed by 1/2 clock cycle.

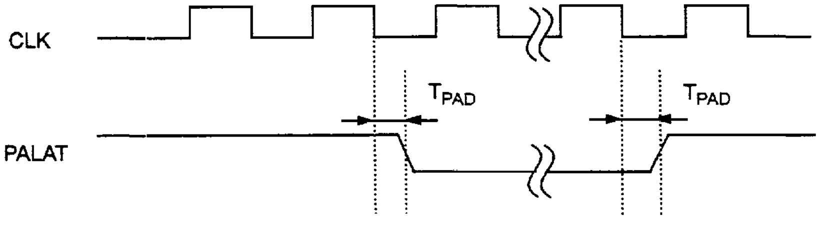
Bus control timing



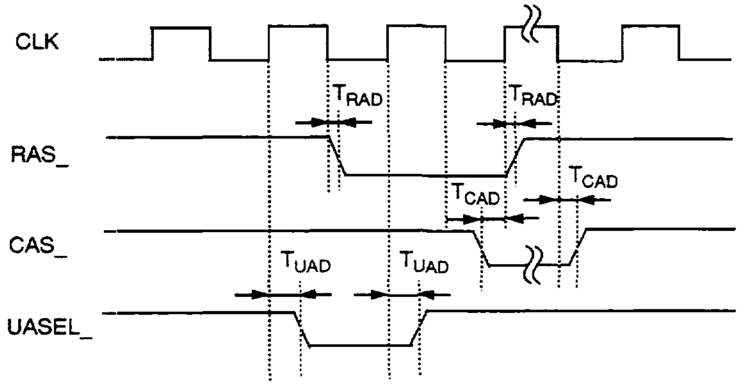
DMA control timing



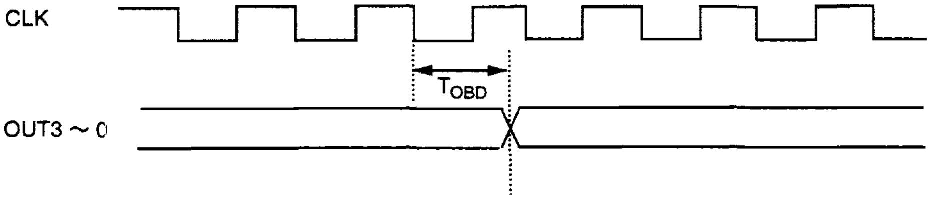
PALAT output timing



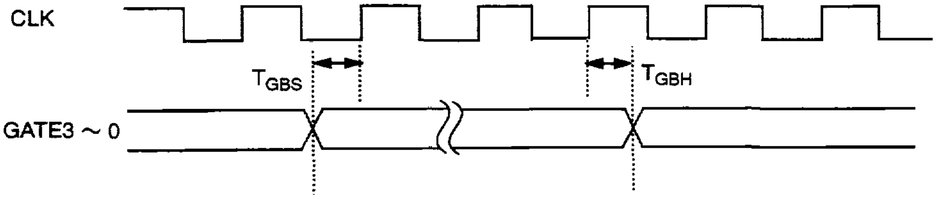
DRAM control timing



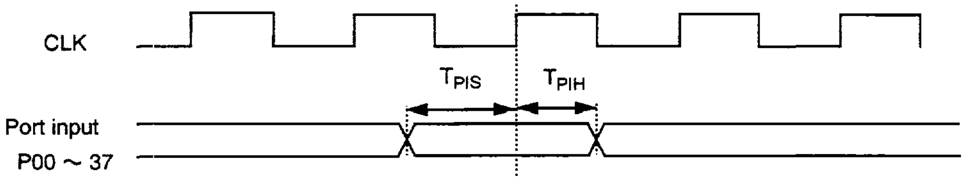
Counter output timing



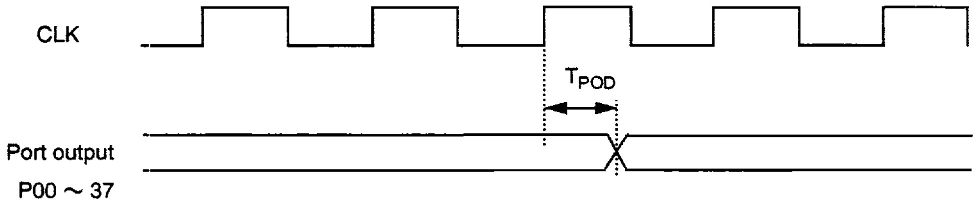
Counter gate timing



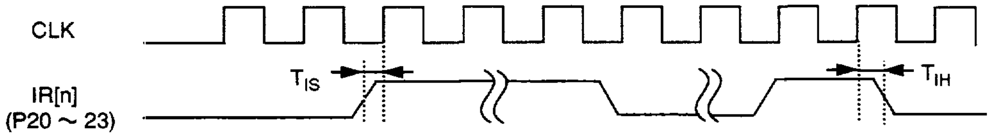
Port input timing



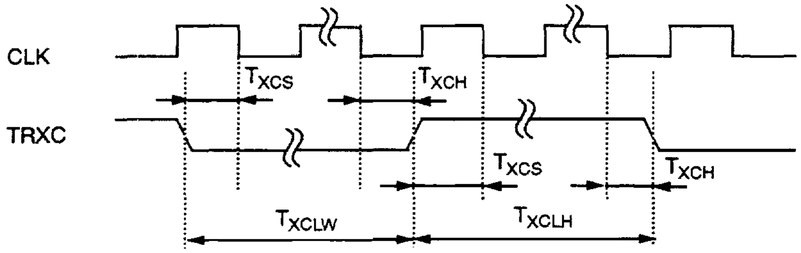
Port output timing



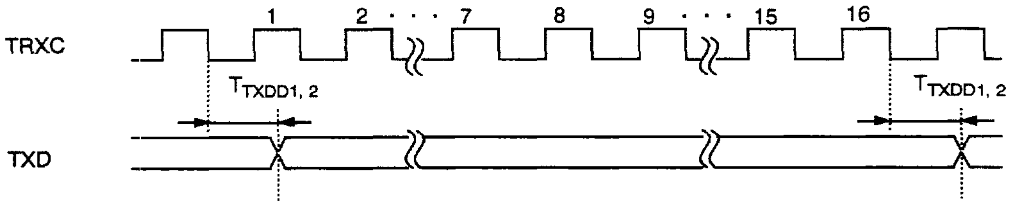
External interrupt input timing



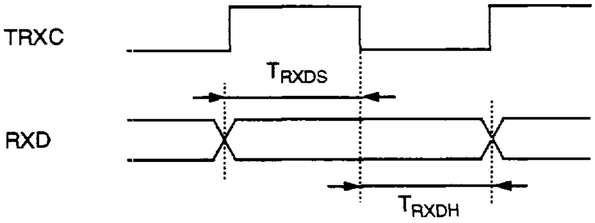
UART transmit/receive clock



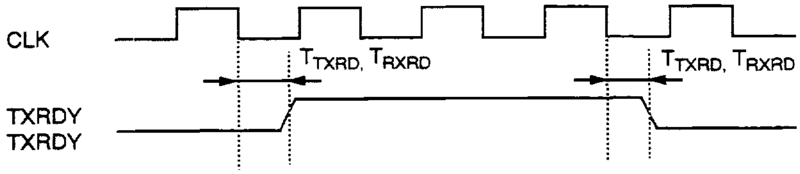
UART transmit clock and transmit data



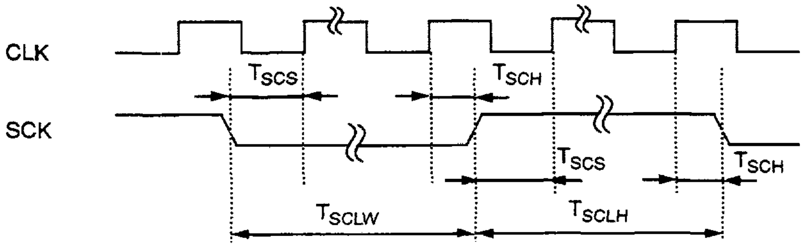
UART receive clock and receive data



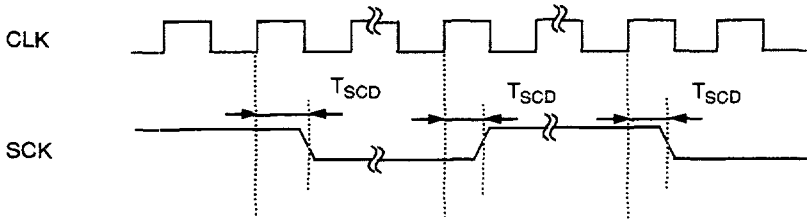
TXRDY, RXRDY output timing



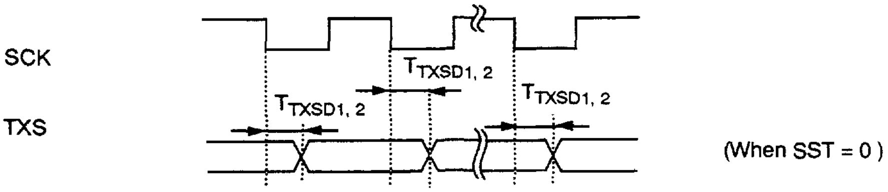
Serial clock input timing



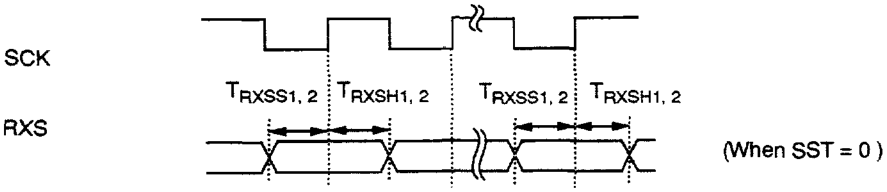
Serial clock output timing



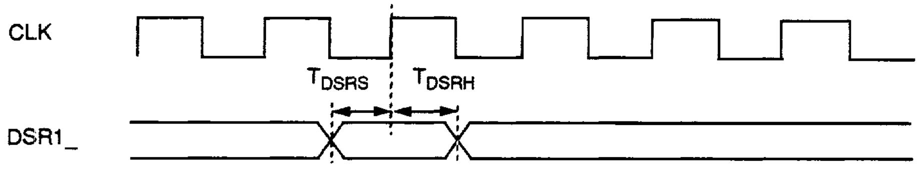
Clocked Serial I/O transmit data timing



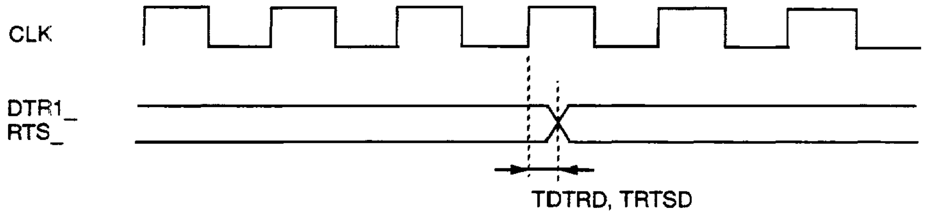
Clocked Serial I/O receive data timing



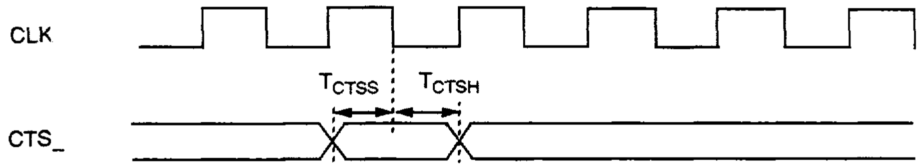
DSR1_ input timing



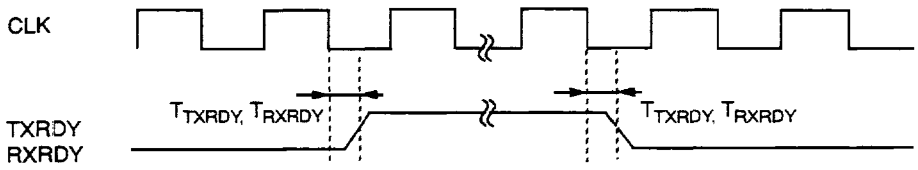
DTR1_ , RTS_ output timing



CTS_ input timing



TXRDT, RXRDY output timing



KL5C80A16 AC Characteristics (5V ± 10%)

No.	Item	MIN	TYP	MAX	Unit
T_{CYC}	XIN cycle time	50. 0			ns
T_{CW}	CLK cycle time	100. 0			ns
T_{CLW}	CLK “L” pulse width		50. 0		ns
T_{CHW}	CLK “H” pulse width		50. 0		ns
T_{RW}	RESET_ pulse width	3			clk
T_{AD}	address delay	10. 0		49. 0	ns
T_{MD}	Chip select delay			52. 0	ns
T_{DD}	CLK→ data output delay			32. 0	ns
T_{DZ}	CLK→ data output off delay	8. 0			ns
T_{DS}	data input set up time	3. 0			ns
T_{DH}	data input hold time	5. 0			ns
T_{MRD1}	CLK rising edge→EMRD_ “L” delay			27. 0	ns
T_{MRD2}	CLK falling edge→EMRD_ “H” delay	7. 0		26. 0	ns
T_{MWD1}	CLK rising edge→EMWR_ “L” delay			27. 0	ns
T_{MWD2}	CLK falling edge→EMWR_ “H” delay	7. 0		27. 0	ns
T_{ERS}	ERDY input set up time	3. 0			ns
T_{ERH}	ERDY input hold time	5. 0			ns
T_{IRD1}	CLK falling edge→EIORD_ “L” delay			30. 0	ns
T_{IRD2}	CLK falling edge→EIORD_ “H” delay	7. 0		25. 0	ns
T_{IRD3}	CLK rising edge→EIORD_ “L” delay			27. 0	ns
T_{IWD1}	CLK falling edge→EIOWR_ “L” delay			31. 0	ns
T_{IWD2}	CLK rising edge→EIOWR_ “H” delay			24. 0	ns
T_{IWD3}	CLK rising edge→EIOWR_ “L” delay			28. 0	ns
T_{BRS}	EXBREQ_ input set up time	3. 0			ns
T_{BRH}	EXBREQ_ input hold time	5. 0			ns
T_{BAD}	EXBACK_ delay			45. 0	ns
T_{DRS}	DREQ_ input set up time	3. 0			ns
T_{DRH}	DREQ_ input hold time	5. 0			ns
T_{DAD}	DACK_ delay			45. 0	ns
T_{PAD}	PALAT delay			40. 0	ns
T_{RAD}	RAS_ delay			31. 0	ns
T_{CAD}	CAS_ delay			31. 0	ns
T_{UAD}	UASEL_ delay			30. 0	ns
T_{GS}	GATE input set up time	3. 0			ns
T_{GH}	GATE input hold time	5. 0			ns

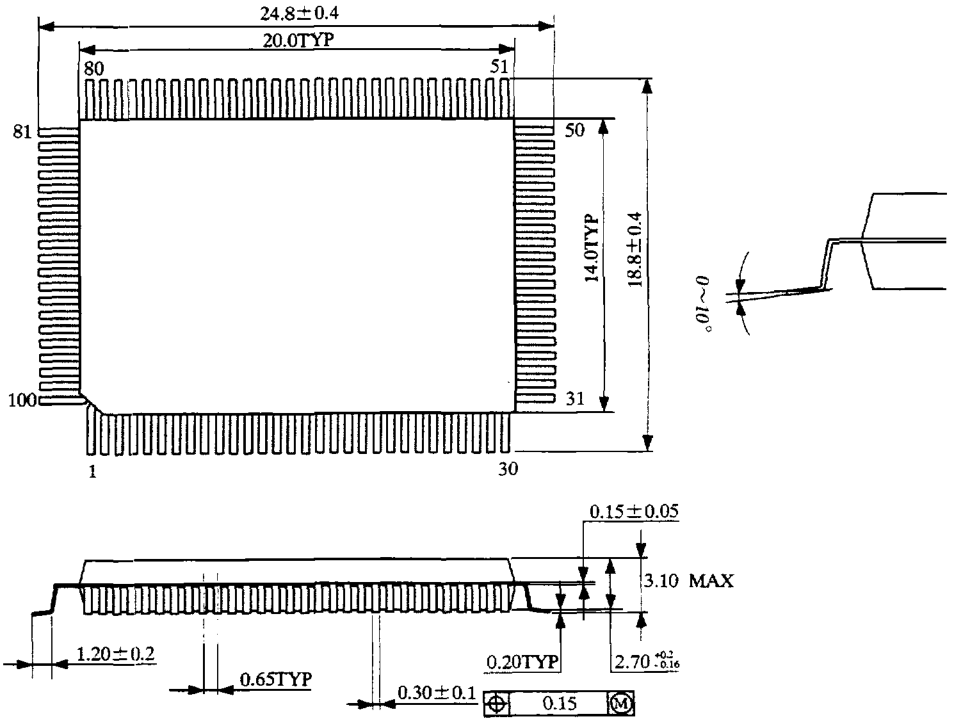
No.	Item	MIN	TYP	MAX	Unit
T_{OD}	Timer/counter OUT output delay			45. 0	ns
T_{PIS}	Port input set up time	5. 0			ns
T_{PIH}	Port input hold time	5. 0			ns
T_{POD}	Port output delay			40. 0	ns
T_{IS}	External interrupt input set up time	3. 0			ns
T_{IH}	External interrupt input hold time	5. 0			ns
T_{NIW}	NMI_ "L" pulse width	20. 0			ns
T_{XCLW}	TRXC "L" pulse width	1			clk
T_{XCHW}	TRXC "H" pulse width	1			clk
T_{XCS}	TRXC input set up time	3. 0			ns
T_{XCH}	TRXC input hold time	5. 0			ns
T_{TXDD1}	TXD delay(for internal transmit/receive clock)			40. 0	ns
T_{TXDD2}	TXD delay(for external clock TRXC)			40. 0	ns
T_{RXDS}	RXD input set up time	3. 0			ns
T_{RXDH}	RXD input hold time	5. 0			ns
T_{DSRS}	DSR1_ input set up time	5. 0			ns
T_{DSRH}	DSR1_ input hold time	5. 0			ns
T_{CTSS}	CTS_ input set up time	3. 0			ns
T_{CTSH}	CTS_ input hold time	5. 0			ns
T_{DTRD}	DTR1_ delay			40. 0	ns
T_{RTSD}	RTS_ delay			40. 0	ns
T_{TXRD}	TXRDY1 delay			40. 0	ns
T_{RXRD}	RXRDY1 delay			40. 0	ns
T_{SCLW}	SCK "L" pulse width	1			clk
T_{SCHW}	SCK "H" pulse width	1			clk
T_{SCS}	SCK input set up time	3. 0			ns
T_{SCH}	SCK input hold time	5. 0			ns
T_{SCD}	SCK delay			40. 0	ns
T_{TXSD1}	TXS delay(for internal transmit/receive clock)			40. 0	ns
T_{TXSD2}	TXS delay(for external clock SCK)			40. 0	ns
T_{RXSS1}	RXS input set up time(for internal clock)	3. 0			ns
T_{RXSH1}	RXS input hold time(for internal clock)	5. 0			ns
T_{RXSS2}	RXS input set up time(for external SCK)	3. 0			ns
T_{RXSH2}	RXS input hold time(for external SCK)	5. 0			ns

Note 1) Output load C_L is 70pf.

2) "clk" in unit is numbers of the system clock.

16. Physical Dimensions

The package of KL5C80A16 is a plastic QFP100. The following is the physical dimensions of QFP100.
"KL5C80A16C " is printed on its package.



Note: All dimensions are in mm.