To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

RENESAS

MOS INTEGRATED CIRCUIT
μPD17P23

SINGLE-CHIP MICROCOMPUTER BUILT-IN EPROM, PRESCALER, PLL FREQUENCY SYNTHESIZER, IF COUNTER AND LCD DRIVER

μPD17P23 is a 4-bit CMOS microcomputer for digital tuning, containing EPROM, a prescaler which can work within the range of up to 150 MHz, a PLL frequency synthesizer, IF counter and a LCD driver (1/2 duty, 1/2 bias) in one chip.

The CPU can perform the 4-bit parallel addition/subtraction (AD and SU instructions), logical operation (EXL instruction, etc.), plural bit tests (TMT instruction, etc.), and set/reset of carry F/F (STC instruction, etc.), and it also has a timer function. The computer is outlined by 64-pin QFP and it contains various I/O ports controlled by powerful instructions (IN and OUT instructions), serial interfaces (serial I/O and shift clock), 6-bit A/D converter, and frequency/duty variable pulse port (CGP: Clock Generator Port).

The device contains a 16-bit frequency counter so that it can be used to select any specific station by counting intermediate frequency in FM/AM tuners.

The analog characteristics (PLL) of μPD17P23 are different from the μPD1723 cases. Using device really should be evaluated about time constance. Operating Frequency of prescaler is 200 MHz MAX. in μPD1723, but it is 150 MHz MAX. in μPD17P23.

FEATURES

- 4-bit microcomputer for digital tuning
- Containing a prescaler (two modulus prescaler: 150 MHz MAX.), the PLL frequency synthesizer, IF counter and the LCD drivers.
- Single 5 V ± 10 % power supply
- Low power consumption CMOS
- Easy data memory (RAM) backup (by the CE pin)
- Program memory (ROM): 16 bits x 2040 steps
- Data memory (RAM): 4 bits x 256 words
- 94 types of powerful instruction set (all by one-word instruction)
- Instruction execution time: 33.3 μsec (with 4.5 MHz crystal resonator)
- Various addition/subtraction instructions (addition: 12 types, subtraction: 12 types)
- Powerful composite judge instructions (AIS ≒ AIN)
- Storage to storage data transfer in a same row address
- Indirect transfer between registers (MVRD and MVRS instructions)
- Sixteen powerful general registers (on RAM space)
- Three stack levels
- Containing a LCD driver (1/2 duty, 1/2 bias, frame frequency = 100 Hz, driving voltage = 5 V ± 10 %, 56 segments MAX.)
- Containing the PLA (Programmable Logic Array) for indications (LCD pattern)
- Clock can be stopped by the CKSTP instruction (Supply current = 10 μA or less)
- Eight I/O ports (PA_0 to PA_3: 1-bit I/O setting, PC_0 to PC_3: 4-bit I/O setting)

NEC cannot assume any responsibility for any circuits shown or represent that they are free from patent infringement.

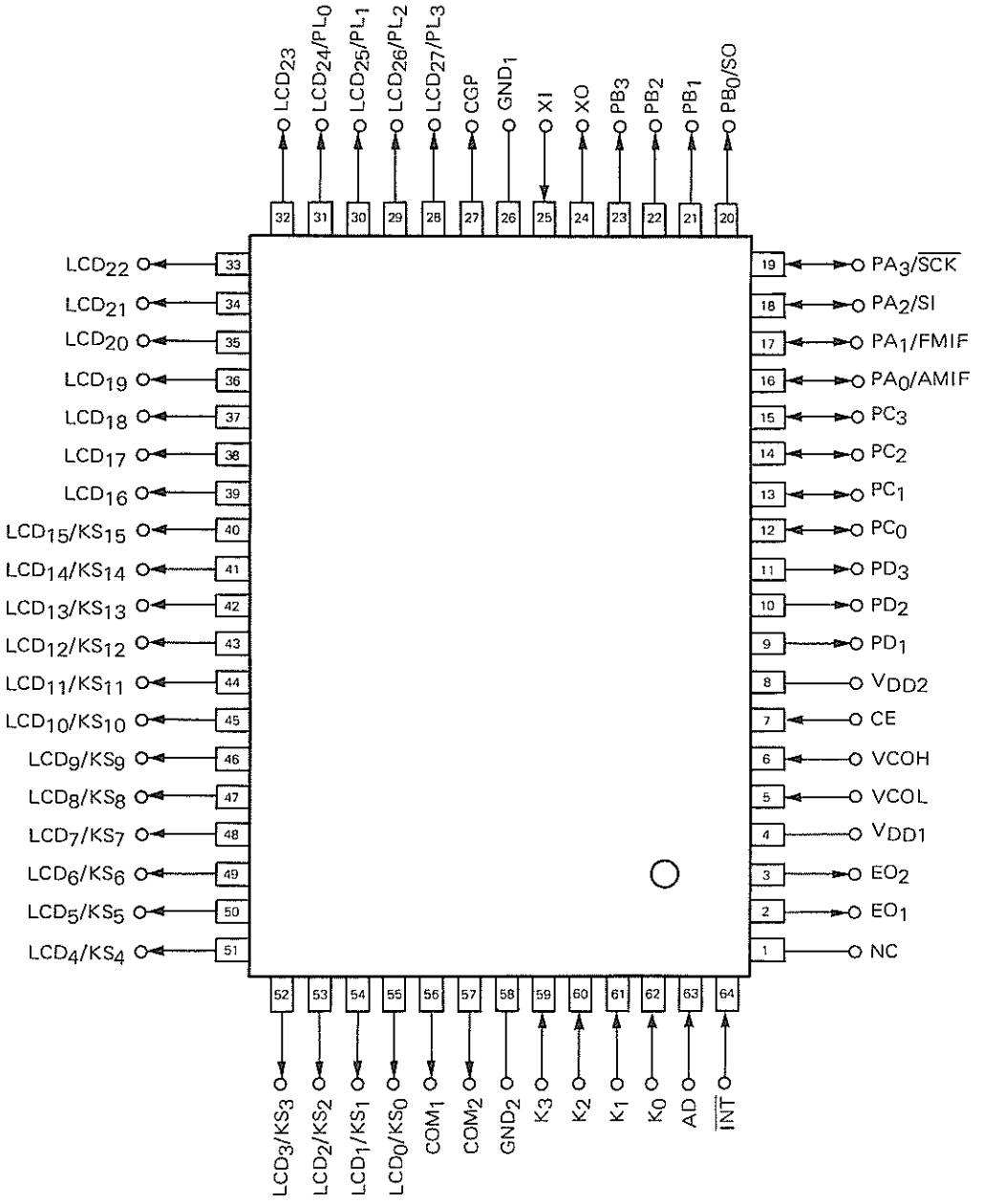
© NEC Corporation 1989

- 12 output ports (PB_0 to PB_3, PD_1 to PD_3, CGP, PL_0 to PL_3: for CMOS output, note that ports PL_0 to PL_3 are also used as a LCD segment pin.)
- Serial interface ($PA_3/\overline{SCK}$: shift clock, PA_2/SI: serial input, PB_0/SO: serial output, 8-bit I/O, SIO instruction)
- 6-bit A/D converter ($V_{ref} = V_{DD}$, sequential comparison method by a program: TADT and TADF instructions)
- CGP (Clock Generator Port) is provided (Output of 64 divided-frequencies based on 180 kHz or 18 kHz; Duty on 2.69 kHz varying into 64 steps)
- Providing key source output pins (LCD_0/KS_0 to LCD_{15}/KS_{15}) which are also used as LCD segment pins
- Input ports used only for key input (K_0 to K_3)
- Powerful I/O instructions (IN and OUT instructions)
- Test on status of input and output ports (TPT and TPF instructions)
- Edge trigger interrupt function (by $\overline{INT}$ pin)
- IF counter is provided. (16 bits, Gate time: 1 ms, 4 ms, 8 ms, and infinity, Maximum input frequency: FMIF = 20 MHz, AMIF = 5 MHz)
- Containing a timer F/F (setting: every 125 ms. Clock function can be easily set.)
- Providing the interval pulse output (internal output)
  (pulse at every 5 ms (200 Hz, duty 60 %); Being tested by TIP instruction.)
- Test on locked condition of the PLL (TUL instruction)
- Transfer of data of dividing ratio and dividing method and reference frequency to PLL by one instruction (PLL instruction)
- Frequency input pins are provided for both AM and FM.
  (Maximum input frequency: VCOL pin = 30 MHz, VCOH pin = 150 MHz)
- Two types of frequency dividing methods (pulse swallowing method and direct method) can be selected using a program.
- Two independent Error Out output (EO_1 and EO_2 pins)
- Seven different reference frequencies can be selected using a program.
  (1, 5, 6.25, 9, 10, 12.5, and 25 kHz)
- Built-in mask ROM: μPD1723

ORDERING INFORMATION

Order Code	Package
μPD17P23GF-3BE	64-pin plastic QFP (bent lead)

PIN CONFIGURATION (Top View)



NC : No Connection

PIN DESCRIPTION

PIN NO.	SYMBOL	PIN NAME	DESCRIPTION	OUTPUT TYPE
1	NC	No-Connection	This pin is not connected to an internal chip so that it cannot be used. However, it can be used for GND, OPEN. and V_{DD} connections.	–
2 3	EO_1 EO_2	Error Output	Error output pin for PLL. If divided frequency of a local oscillator (VCO output) is higher than the reference frequency, high level signals sent from these pins and if it is lower than the reference frequency, low level signal is output from them. If the two frequencies are the same, it becomes a floating. This output is sent to the external LPF (Low Pass Filter), then is output to a varactor diode via the LPF. Since EO_1 and EO_2 output same wave form, either one can be selected.	CMOS 3-states
4 8	V_{DD1} V_{DD2}	Power Supply	These are device power pins which supplies the voltage of 5 V ± 10 % while the device is in operation. The voltage can be lowered to 2.5 V when the internal data memory (RAM) is provided (when the CKSTP instruction is executed). If the voltage of 0 to 4.5 V is supplied to this pin, the device is reset and the program starts from Address 0. (See the TIMER F/F in Section 1.5). Pin 4 and pin 8 must be connected in the same electric potential condition. V_{DD1} (4 pin) is a device power pin for analog (PLL, A/D converter, $\overline{INT}$ and CE), V_{DD2} (8 pin) is a device power pin for digital (CPU, LCD driver and IF counter).	–
5	VCOL	AM Local Oscillation Signal Input	This inputs the AM local oscillation output (VCO output) in the range of 0.50 to 30 MHz (0.3 V_{p-p} minimum). This pin becomes active when the direct frequency dividing method is selected. When the direct frequency dividing method is selected, the dividing ratio varies from 16 to $(2^{12} - 1.)$ When the pulse swallowing method is selected, the AM pin automatically, becomes pull-down state (GND). The output have to be cut by a capacitor to be input because an AC amplifier is provided in the device.	Input
6	VCOH	FM Local Oscillation Signal Input	Local oscillation output (VCO output) in the range of 10 to 150 MHz (0.3 V_{p-p} minimum) is input to this pin. The dividing ratio when the pulse swallowing method is selected varies from 1024 to $(2^{17} - 1)$. The FM pin becomes pull-down state (GND) automatically when the direct dividing method is selected. Since an AC amplifier is contained, output should be cut by a capacitor to be input. (See Section 2.4, PLL REGISTOR, for details.)	Input

PIN NO.	SYMBOL	PIN NAME	DESCRIPTION	OUTPUT TYPE
7	CE	Chip Enable	Device selection signal input pin. This pin is set to high level when the device performs normal operation (This pin is set to low level when the device is not used). The PLL becomes disable state while this pin is in low level. However, input of 134 μs or less cannot be accepted. If the CKSTP instruction is executed while the CE pin is in the low level, the internal clock generator and the CPU halt their motions and the memory holding state can be obtained in the low current consumption state (10 μA or less). (CKSTP instruction is only effective when the CE is in the low level. When the CE is in the high level, it does the same operation as that of the NOP instruciton.) At this time the display output (from LCD_0 to LCD_{27}, COM_1, and COM_2) automatically become the display-off mode (low level). If the CE pin is changed from the low level to high level, the device is reset and the program starts from address zero. (See 1.5 the TIMER F/F.) At this time, the I/O ports (Port A and Port C) become the input mode.	Input
9 to 11	PD_1 to PD_3	Port D	These pins are 3-bit output ports. (See Notes 1 and 3.)	CMOS push-pull
12 to 15	PC_0 to PC_3	Port C	These pins are 4-bit I/O ports. Any of these become an output port when OUT, SPB, or RPB instruction is executed for the Port C. Any of these become an input port when IN instruction is executed for the port. (Refer to Notes 1, 2 and 3.)	CMOS push-pull
16 17 18 19	PA_0/AMIF PA_1/FMIF PA_2/SI PA_3/SCK	Port A	These pins are 4-bit I/O ports. 1-bit I/O assignment can be done at these ports. The assignment is specified by the contents of address 1FH in the data memory BANK0 (RAM). These ports can be used as a serial interface in the SIO instruction. At this time, pin PA_3 functions as a $\overline{SCK}$ (Shift Clock) pin. Likewise, pin PA_2 and pin PB_0 function as a SI (Serial Input) pin and a SO (Serial Output) pin, respectively. When the device is reset (V_{DD} = Low → High, CE = Low → High) and the CKSTP instruction is executed, these ports become the input mode. When PA_3 is used as the $\overline{SCK}$ pin, PA_3 has to be pulled up using a resistor. If the pin is used without the above operation, shift clock is not output correctly. PA_0 and PA_1 are used as a pin for frequency measurement. PA_0 and PA_1 work as the AMIF pin and the FMIF pin, respectively. The maximum input frequency for the AMIF pin is 5 MHz and that for the FMIF is 20 MHz. If the AMIF pin is selected, input signals are directly input to the IF counter. If the FMIF pin is selected, input signals are input to the IF counter via a 1/2 frequency divider. (See Notes 1 and 2.)	CMOS push-pull
20 to 23	PB_0/SO to PB_3	Port B	These pins are 4-bit output ports. Pin PB_0 is used as a SO (Serial Output) pin by executing the SIO instruction (See Notes1 and 3.)	CMOS push-pull
24 25	XO XI	X'tal	These pins are connection pins for crystal resonator. Connect a 4.5 MHz crystal resonator and adjust the oscillator frequency (4.5 MHz) by monitoring the LCD driver waveform.	CMOS Input

PIN NO.	SYMBOL	PIN NAME	DESCRIPTION	OUTPUT TYPE
26 58	GND_1 GND_2	Ground	Ground pin of the device. (Note) pin 26 and pin 58 are not connected in the chip so that there is no need to supply voltage to both pins. Power should be supplied to any one pin to operate the device.	—
27	CGP	Clock Generator Port	CGP (Clock Generator Port) or one-bit output port (PG_2) can be selected by the program. When this pin is used as the CGP, two functions, VDP (Variable Duty Pulse) function and SG (Signal Generator) function, can be selected by the program. The VDP function can continuously output 2.69 kHz of pulses and can change the duty of the pulses into 64 steps using the program. The SG function can output the frequency divided into 64 steps (duty 50 %) after varying it by the program where the reference frequency is 18 kHz or 180 kHz. When the device is reset (V_{DD} = Low → High, CE = Low → High) or when the CKSTP instruction is executed, the CGP pin becomes the low level. (See the Note 1.)	CMOS push-pull
28 to 55	LCD_{27}/PL_3 to LCD_0/KS_0	LCD Segment Output	These pins are segment signal output pins for the LCD panel. Up to 56 dots of display can be made by using the matrix of COM_1 and COM_2. Output to these pins can be done by executing the LCDD instruction. Contents of any address in the data memory (RAM) specified by the first operand of the LCDD instruction is output to the column of the LCD matrix specified by the second operand. (See LCD DRIVER in Section 8.) If an even numbered column is specified at the time, the contents specified by the first operand is output to these pins through the segment PLA (Programmable Logic Array). The segment PLA can generate 32 types of patterns. (See PLA in Section 10.) When pins from LCD_{27}/PL_3 to LCD_{24}/PL_0 are not used as LCD segment signal output pins, they can be used as 4-bit output ports. Pins from LCD_{15}/KS_{15} to LCD_0/KS_0 can be used as the signal source of key matrix. In this case, display data and key source signals are output from these pins using the time sharing method. (Note) Low level signals are automatically output (Display Off Mode) when power is applied (V_{DD} = Low → High) and when the CKSTP instruction is executed.	CMOS push-pull
56 57	COM_1 COM_2	LCD Common Output	These pins are output pins of the common signals for the LCD panel. Up to 56 dots of display can be made by using the matrix of LCD_0 to LCD_{27}. Three values (GND, 1/2 V_{DD}, and V_{DD}) are output in every 5 ms with 100 Hz of frequency. When $\pm V_{DD}$ of potential difference arises between these pins and the LCDs from LCD_0 to LCD_{27}, the segment will be indicated. (See LCD DRIVER in Section 8.) (Note) Low level is automatically output (Display Off Mode) when power is applied (V_{DD} = Low → High) and when the CKSTP instruction is executed.	CMOS push-pull

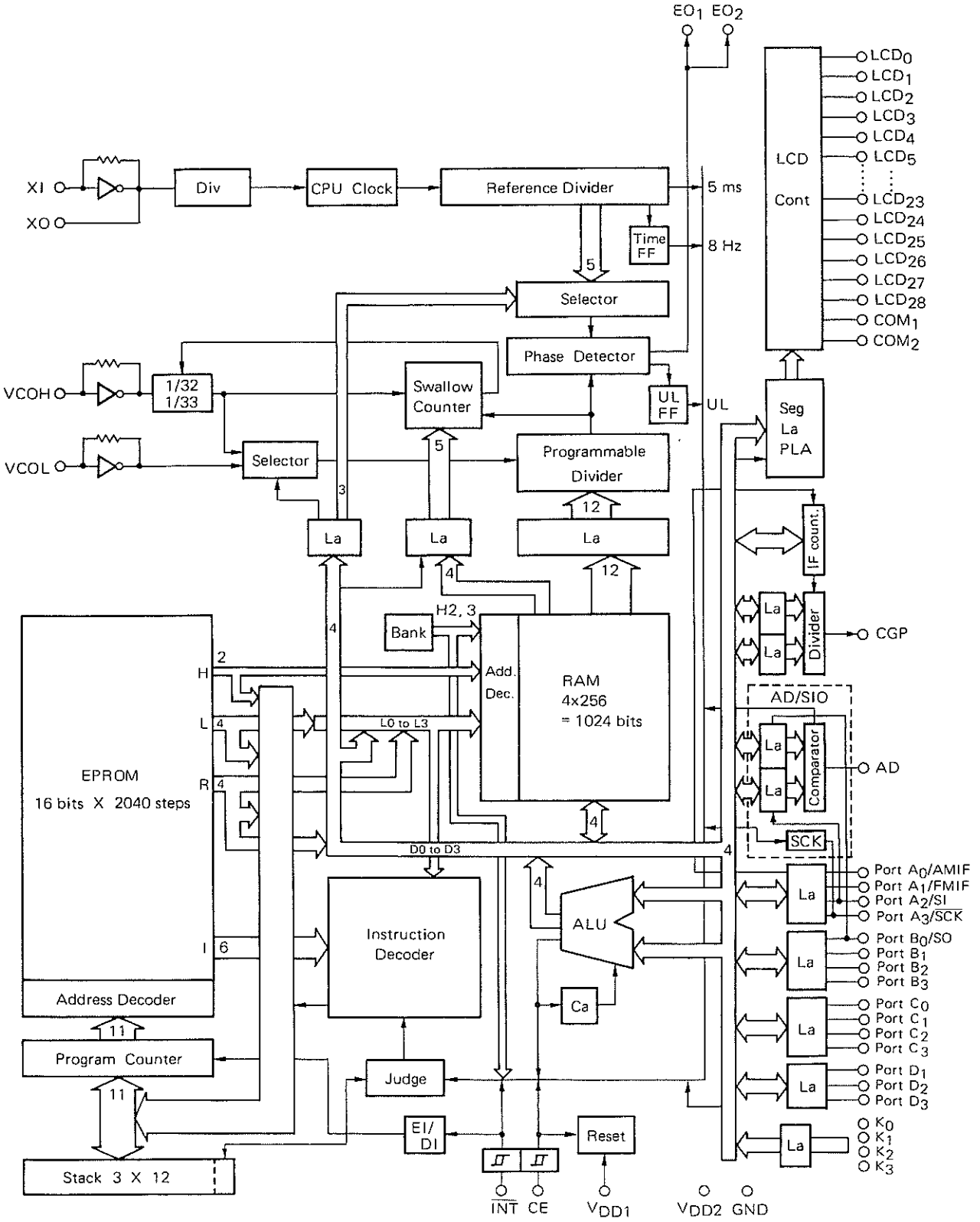
PIN NO.	SYMBOL	PIN NAME	DESCRIPTION	OUTPUT TYPE
59 to 62	K_0 to K_3	Key-return Signal Input	These are 4-bit input ports used for key matrix input. When the KIN instruction or the KI instruction is executed, the state of these pins are stored in the data memory (RAM) specified by the operand of the instruction. Segment pins of the LCD (LCD_{15} to LCD_0) are especially used as key return signal source. The state of key latch can be read at the time if the KIN instruction or the KI instruction is executed.	Input
63	AD	A/D converter Input	This is an A/D (Analog/Digital) converter input pin. A 6-bit A/D converter using the sequential comparison method by the program is provided in the device. Reference voltage of the A/D converter is V_{DD} (5 V ± 10 %).	Input
64	$\overline{INT}$	Interrupt	This is the input pin of interrupt request signals, which are issued at the fall edge of a signal applied to this pin. When the interrupt request is accepted, the program flow unconditionally jumps to Address 1.	Input

Note 1: PA_0 corresponds to the lowest-order bit of a register or an operand data and PA_3 to the highest-order bit under the port operation instructions (IN, OUT, SPB, and RPB instructions). It goes to port B or port C as well.

Note 2: All the I/O ports (Port A and Port C) turn to the input mode when the device is reset (V_{DD} = Low to High, CE = Low to High) or when the CKSTP instruction is executed.

Note 3: Output-only ports (Port B and Port D) output incorrect data when power is applied (V_{DD} = Low to High). Therefore, they should be initialized by the program. When the CE pin changes from Low to High and when the CKSTP instruction is executed, the contents of output data does not change from the last one. Therefore, in this case, initializing should be done if necessary.

BLOCK DIAGRAM



CONTENTS

1. CPU 11
1.1 PROGRAM COUNTER (PC) 11
1.2 STACK REGISTER (SR) 11
1.3 PROGRAM MEMORY (ROM) 11
1.4 DATA MEMORY (RAM) 14
1.5 TIMER F/F (T F/F) 16
1.6 INTERVAL PULSE (ITP) 19
1.7 UNLOCK F/F (UL F/F) 19
1.8 CARRY F/F (C F/F) 19
1.9 BANK F/F (B F/F) 20
1.10 INT F/F AND INTE F/F 20
1.11 STATUS WORD 21

2. PLL 22
2.1 REFERENCE FREQUENCY GENERATOR (RFG) 22
2.2 PHASE DETECTOR (ϕ-DET) 22
2.3 PROGRAMMABLE DIVIDER (P/D) 22
2.4 PLL REGISTER 23
2.5 SETTING OF PLL INFORMATION 27

3. PORT 29
3.1 PORT A 31
3.2 PORT B AND PORT D 33
3.3 PORT C 34
3.4 PORT L 35

4. CGP 36
4.1 PG #2 THROUGH MODE 37
4.2 VDP MODE 37
4.3 SG MODE 38

5. SERIAL I/O 40
5.1 SHIFT MODE REGISTER (SMR) 40
5.2 PRESETTABLE SHIFT REGISTER (PSR) 46
5.3 SHIFT CLOCK COUNTER (SCC) 46
5.4 SHIFT CLOCK GENERATOR (SCG) 47
5.5 APPLICATION EXAMPLE OF SERIAL I/O 48

6. A/D CONVERTER 50
6.1 OPERATING PRINCIPLES 50
6.2 CONSTRUCTION OF D/A CONVERTER 50
6.3 EXAMPLE OF A/D CONVERSION PROGRAM 54

7. IF COUNTER 56
7.1 CONFIGURATION OF IF COUNTER 56
7.2 IF COUNTER CONTROL WORD (IFCW) 57
7.3 GATE SIGNAL 58
7.4 ERROR 59
7.5 EXAMPLE OF IF COUNTER DATA CALCULATIONS 59

8. LCD DRIVER 60
8.1 DIGIT 60
8.2 LCD CONTROL WORD 68

9. KEY MATRIX CONFIGURATION 70
9.1 THE WAY OF USING STANDARD PORT AS KEY SOURCE SIGNAL OUTPUT 70
9.2 THE WAY OF USING LCD SEGMENT PIN AS KEY SOURCE SIGNAL OUTPUT 70
9.3 KEY SOURCE DECODER 71
9.4 KEY LATCH F/F 73
9.5 PROGRAM EXAMPLE USING KEY SOURCE DECODER 73

10. PLA (PROGRAMMABLE LOGIC ARRAY) 76
10.1 COMPOSITION OF SEGMENT PLA 76
10.2 PATTERN EXAMPLE OF SEGMENT PLA 81
10.3 EXAMPLE OF PLA PROGRAM 83

11. INSTRUCTION 84
11.1 INSTRUCTION SET 84
11.2 INSTRUCTIONS 85

12. INPUT/OUTPUT CIRCUIT OF PIN 88

13. ELECTRICAL CHARACTERISTICS 90

14. PACKAGE DIMENSIONS 92

15. SUPPORT TOOLS 94

1. CPU

1.1 PROGRAM COUNTER (PC)

The program counter is composed of a 11-bit binary counter, and addresses the program memory (EPROM), that is, the program. (Note 1)

PC_{10}	PC_9	PC_8	PC_7	PC_6	PC_5	PC_4	PC_3	PC_2	PC_1	PC_0

11 bits

The counter is generally increased by one every time one instruction has been executed, and it is loaded with the address as designated by the operand of a jump instruction or a subroutine call instruction when it is executed. (Note 2) When any skip instruction (e.g. ADS, TMT or RTS instructions) is executed, it designates the address of a command ensuing to the skip instruction, regardless of the content of skip condition. If the said condition requires the skip, the instruction subsequent to the skip instruction is considered NOP (no-operation). In other words, the address of next instruction can be designated after the execution of NOP.
When an interruption request is accepted **Address 1 is loaded unconditionally.**

Note 1: The program counter of each of μPD1701, μPD1703, μPD1704, μPD1705, μPD1710, μPD1711, μPD1716, μPD1720 and μPD1730 (in the series having the ROM capacity less than 1K steps) is composed of 10 bits only.

Note 2: Since the operand of the JMP instruction consists of 10 bits, there are two types of JMP instructions in μPD17P23 and PC_{10} is set or reset depending on the operation code. These two JMP instructions have the same mnemonic code (JMP) so that the identity of these two instructions are judged by the assembler automatically. (See the Program Memory in Section 1.3) The CAL instruction is only one type. When the CAL instruction is executed, PC_{10} is reset.

1.2 STACK REGISTER (SR)

The stack register consisting of 3 x 12 bits stores the return address (11 bits), which is the value of adding one to the contents of the program counter when a subroutine call instruction is executed or when an interruption request is accepted, and the result of a judgement (1 bit) if any instruction having the skip function has been executed upon the acceptance of the interruption request. The contents of the stack register is loaded into the progam counter by the execution of a return instruction (RT or RTS), and the original program flow is restored.

The stack register is used both for the subroutine call and the interrupt so that if one level is used for the interrupt, the remaining stack register which can be used for a subroutine call becomes two levels.

1.3 PROGRAM MEMORY (EPROM)

The program memory (EPROM), consisting of 16 bits x 2040 steps, stores programs within the address range from 000H to 7F7H.

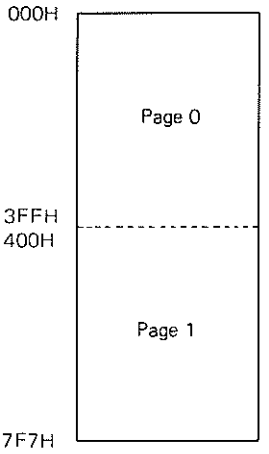


Fig. 1-1 EPROM Configuration

The page concept applies to this EPROM of μPD17P23; **Page 0 ranges from the EPROM addresses 000H to 3FFH and Page 1 from 400H to 7F7H.**

The head address of a subroutine should be placed within Page 0 when a program is created; no subroutine having its head address within Page 1 can be called either from Page 0 or Page 1 (see Notes to Use of CAL instruction). The JMP instruction, which is described in the assembler, can use any address between 000H to 7F7H in the same description (JMP ADDR) without being conscious of the page, provided that the patch correction needs be made for debug as the operation code of JMP instruction differs between Page 0 and Page 1 (see Note to Use of JMP instruction).

The following must be noted in the use of CAL and JMP instructions as the page concept (discrimination between Page 0 and Page 1) applies to EPROM of μPD17P23.

EPROM of μPD17P23 can be written at one time only.

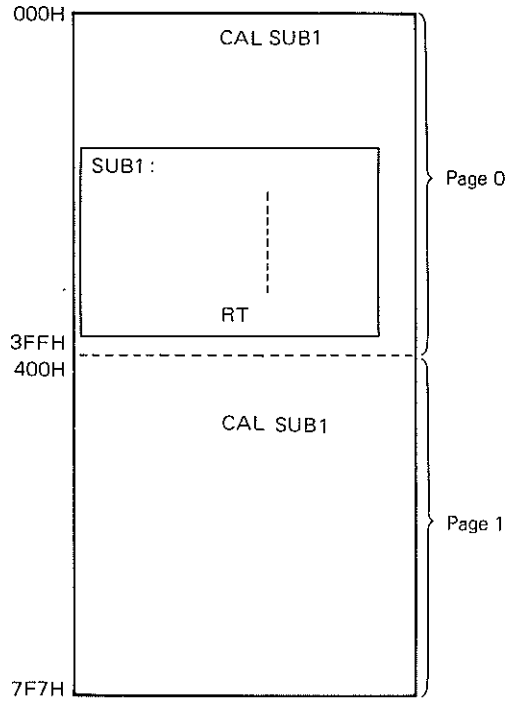
When writing to μPD17P23, must be used the EPROM programmer only for writing to μPD17PXX (See 15. SUPPORT TOOLS)

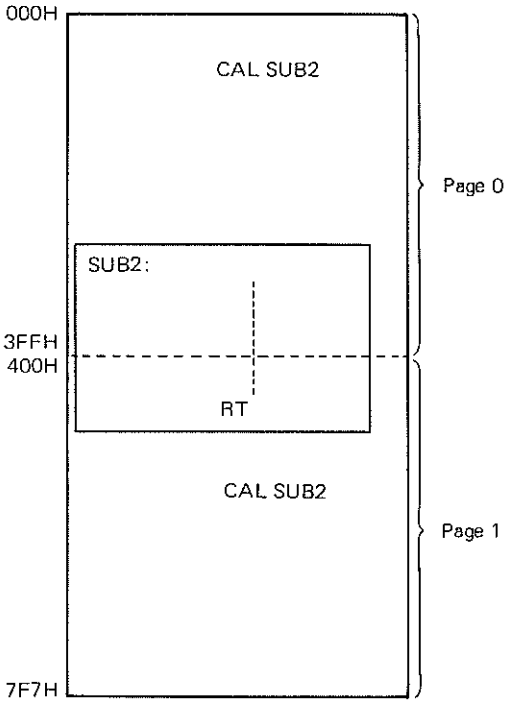
The family code must be specified "P19" at assembling. (The object data of μPD1723 is different to that of μPD17P23.)

Notes to Use of CAL Instruction

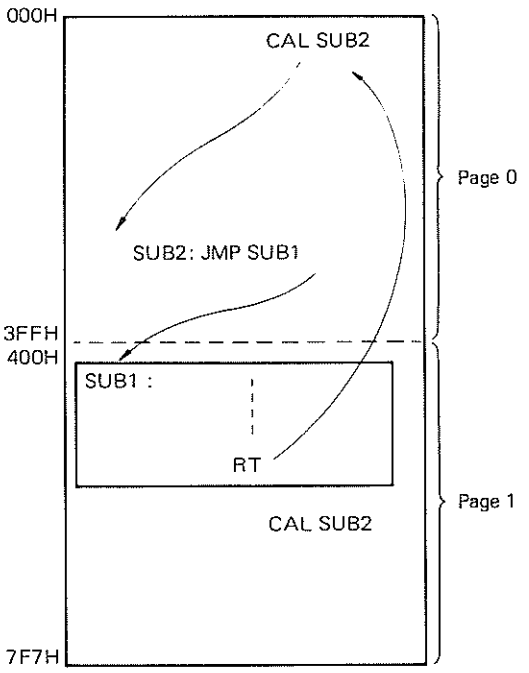
When the CAL instruction is used, its call address or the head address of subroutine must be set within Page 0 (000H to 3FFH); the subroutine whose head address is set within Page 1 (400H to 7F7H) cannot be called, provided that the return address (RT or RTS instruction) can be placed within Page 1.

(Example 1) Head address of a subroutine in page 0.





The return address (RT or RTS instruction) can be placed either in Page 0 or Page 1 so far as the head address of subroutine is held within Page 0; the CAL instruction can be used then without being conscious of the page concept. The following technique is however effective if the placing of head address of any subroutine within Page 0 is prevented by programming:



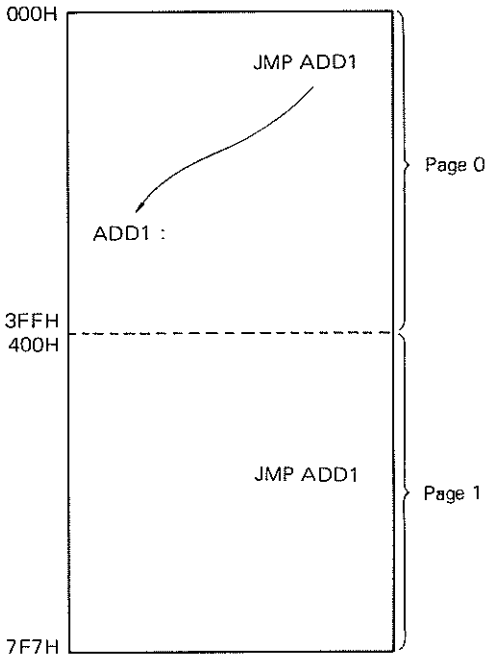
In this technique the actual subroutine (SUB 1) is called through a JMP instruction by storing the JMP instruction in page 0.

Notes to Use of JMP Instruction

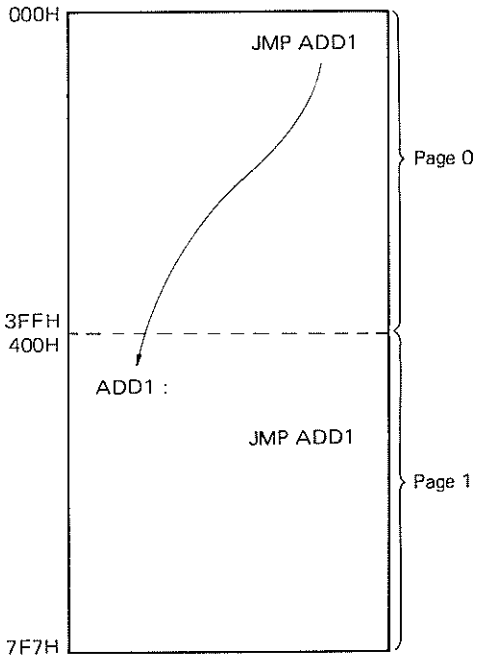
The JMP instruction can be used in the same description between the ROM addresses 000H and 7F7H without being conscious of the page concept insofar as it is described in the assembler. Note that the operation code of JMP instruction differs between Page 0 (000H to 3FFH) and Page 1 (400H to 7F7H). The operation code of JMP instruction within Page 0 is "06" or "02" if it is within Page 1.

The assembler of μPD1700 Series can automatically refer to the jump address and convert the operation codes.

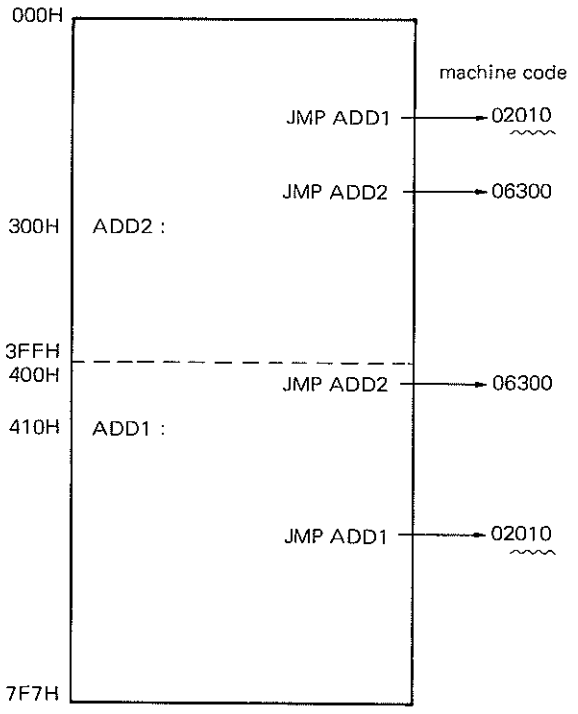
Operation Code 06
(Jump address in page 0)



Operation Code 02
(Jump address in page 1)



The programmer is however required to convert the operation codes "06" and "02" when the patch correction takes place in debugs. The address also needs to be converted if the jump address of JMP instruction is beyond 400H (operation code 02); the address is then increased one by one from 400H, which is then considered 000H. Thus the address 7F7H turns to 3F7H.



To effect the patch correction of JMP 400H described in the assmebler, for instance, input 02000 and turn JMP 000H to 06000.

1.4 DATA MEMORY (RAM)

The RAM, consisting of 4 bits x 256 (4 x 64) words, is generally used to store data. These 256 words are divided into two groups; BANK0, BANK1, BANK2 and BANK3 contains 64 words each. If data processing is done in each BANK, a BANK specification (BANK0 and BANK1 instructions, etc.) has to be done in advance to process data.

Addresses from 00H to 0FH in BANK0 are called a general register and it is used for operations and transfer with memory. It is also used as a regular memory. (If it is used as a register, a BANK specification is not required. However, if it is used as a memory, BANK0 must be specified.)

Information (divided frequency, reference frequency and dividing method) needed for the PLL control can be set via the RAM. Total of 17 bits, consisting of 4 bits by 4 words (N's words) in the addresses of 00H, 10H, 20H, and 30H plus the most significant bit (N_F bit) of a general register, are allocated to the setting of divided frequencies. Similarly, N's words and any one word (4 bits) of the RAM except for the one which contains the N_F bit are allocated to the setting of the reference frequency (control words). These bits are transferred to the PLL register by a PLL instruction.

Address 1FH in BANK0 is called PAIO word and is used for the I/O assignment of the Port A (PA_0 to PA_3).

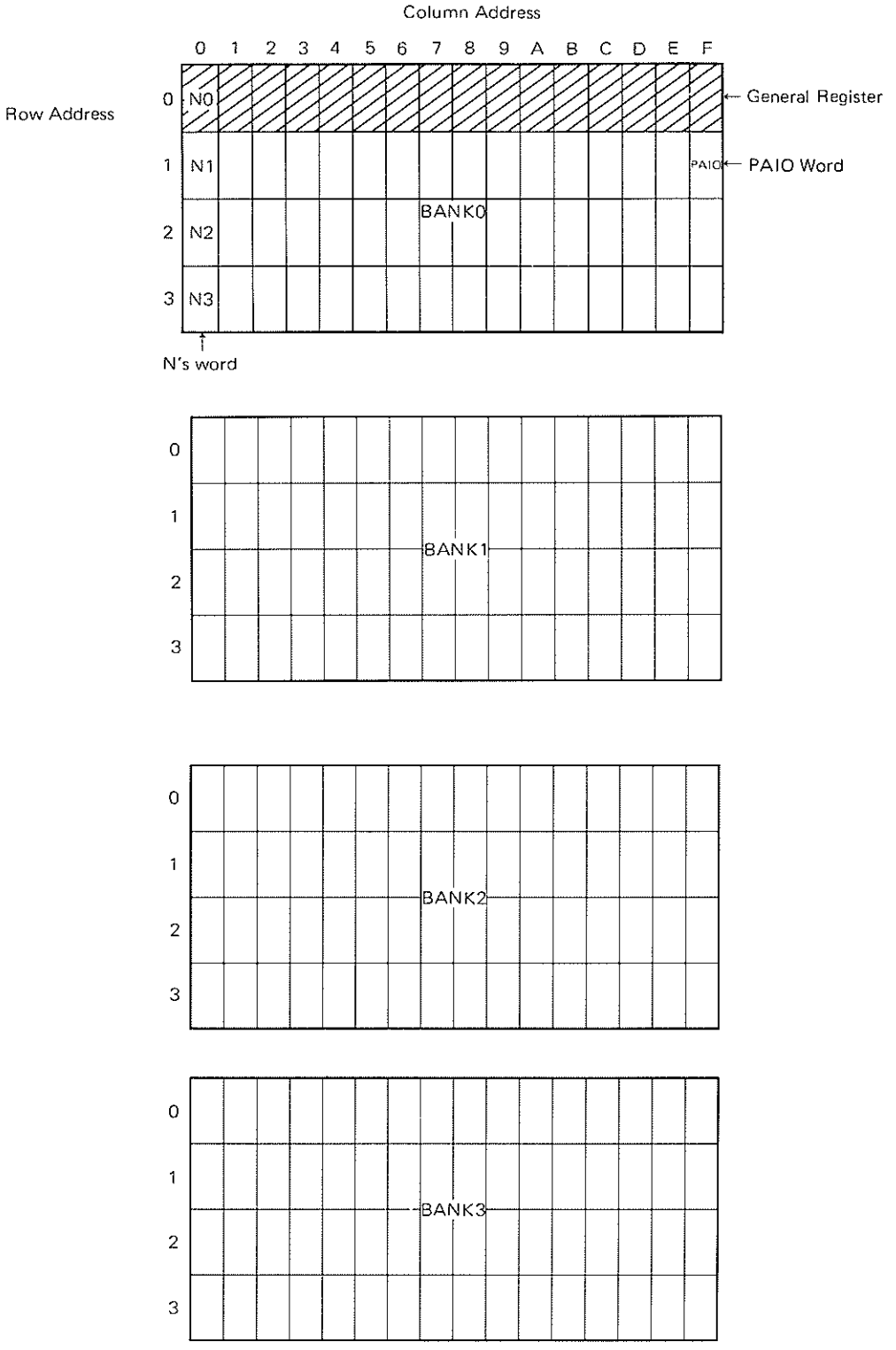


Fig. 1-2 RAM Configuration

Note: The most important point you should keep in mind in the general register operation without BANK0 is that **μPD17P23 does not have operational instructions between the general registers and the immediate data.** For example, expression "AI 00, 1" in a program in BANK0 adds one to the general register which is stored in address 00 in the data memory. This AI instruction is the operation between memory and the immediate data. This instruction is not the operation between the register and the immediate data. If the above instruction is executed when BANK1 is specified, this expression does not add one to the address 00 of the general register but adds one to the address 00 of the data memory in BANK1.

1.5 TIMER F/F

The timer F/F is to be set with an 8 Hz (125 ms) signal and reset by the test timer instruction (TTM instruction). This timer F/F is automatically set every 125 ms so that it can be used as a clock counter (one second by 8 counts) or the mute time counting.

Since the timer F/F can be reset only by the execution of TTM instruction **this instruction must be executed within the period of 125 ms under any circumstance.** When this instruction is executed within a period of 125 ms or more, the timer F/F fails to count up the clock and becomes unable to control the correct time.

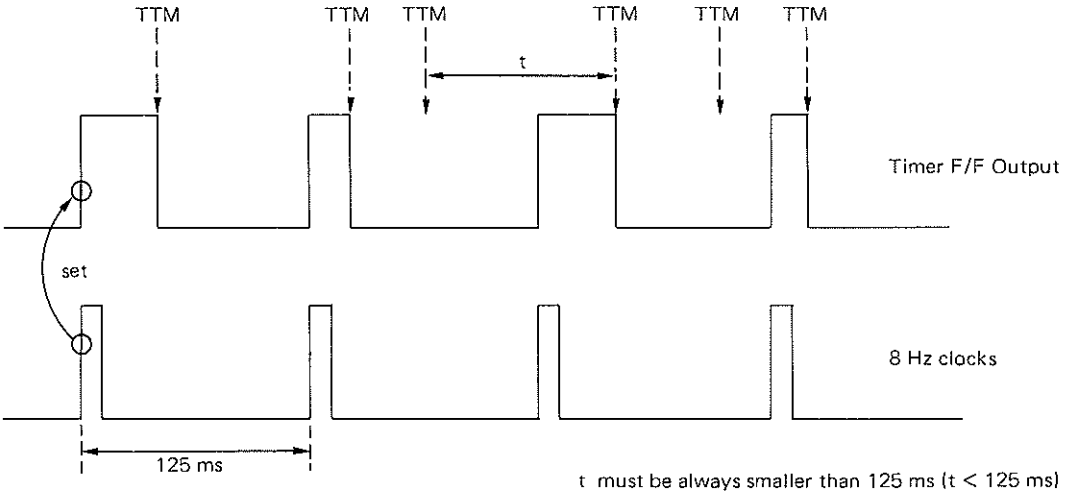


Fig. 1-3 Execution Timing of TTM Instruction

This timer F/F can be also used to judge the detection of power failure. It is reset when the power is turned on (V_{DD} changes from low to high) or it is set again by the execution of CKSTP instruction or when CE changes from low to high. (Note 3) Fig. 1-4 shows the status transition diagram illustrating the afore-mentioned relations.

As apparent from the above illustration, the program starts from the address 0 after the power is turned on (V_{DD} changes from Low to High), no matter what condition the CE pin is held, **while the timer F/F remains reset.** The timer F/F is not set again unless the TTM instruction was once executed (status unable to set the timer F/F). Once the TTM instruction was executed, however, the timer F/F can be set at any time at the intervals of 125 ms each.

If the power is being fed (V_{DD} = high) and the CE pin changes from low to high, the program flow jumps to the address 0 immediately when the timer F/F is set. (Note 4) The program therefore starts from the address **0 while the time F/F remains set.**

As you could understand well from the above explanation, the contents of Timer F/F vary between the time the power failure is recovered (V_{DD} changes from low to high) and the time when the power is continuously fed (V_{DD} =high and CE=low) or when the device is restored from the backed-up condition. Through testing the contents of this timer F/F (i.e. the execution of TTM instruction) it is possible to judge if it is restored from the power failure or from the non-power failure. In other words the power failure can be judged if the execution of TTM instruction, which is **executed within 125 ms from the start of porgram from the address 0,** results in 0 (false) or it can be determined as non-power failure (backed-up condition) if the result of test turns to 1 (true).

Care must be also taken to the programming when restoring from the non-power failure (V_{DD}=high and CE changes from low to high) while the clock function of a given program, if provided, needed to be operated (without using the CKSTP instruction) even if CE was low. The porgram flow in this case jumps to the address 0 immediately after the timer F/F is set. It is therefore necessary to update the clock after executing the TTM instruction to detect the power failure (the execution results in finding true). Otherwise the clock delays by 125 ms each whenever the CE pin changes from low to high.

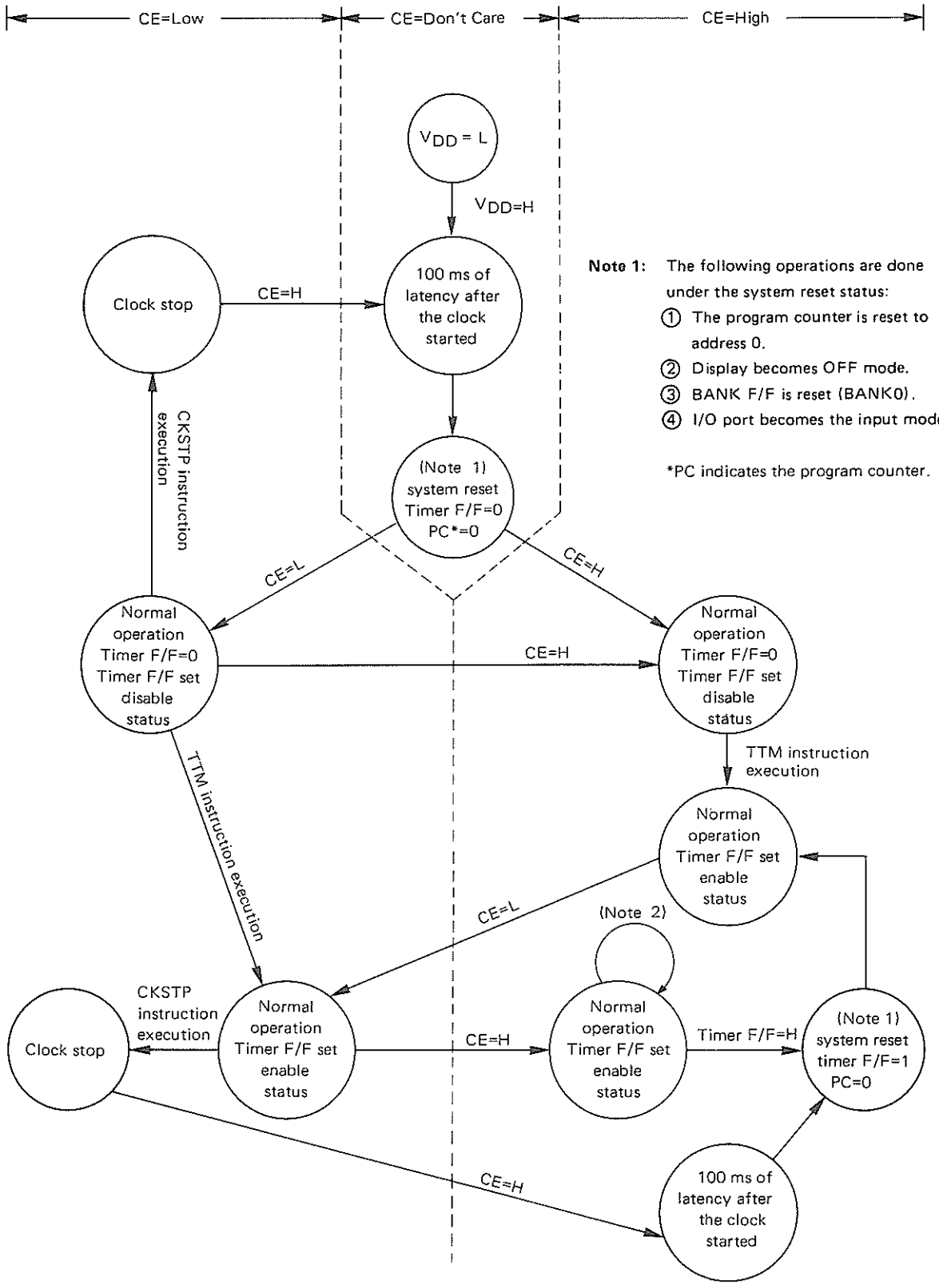


Fig. 1-4 CPU Status Change Diagram by the CE pin

Note 2: It is impossible to come put from this loop if the setting of the timer F/F and the execution of the TTM instruction are done simultaneously. If this is the case, the loop can be broken by the setting of the next timer F/F (after 125 ms), then the program jumps to address 0 after turning the timer F/F to "1". It must be noted if the execution of the TTM instruction is periodic and if the execution cycle happens to coincide with the cycle of the timer F/F setting (125 ms), program cannot be cleared to address 0 permanently.

Note 3: The program starts from the address 0 after the timer F/F was set if CE pin changed from low to high following the execution of CKSTP instruction in case of μPD17P23. The timer F/F is reversely reset and the program starts from the address 0 under μPD1701, μPD1703, μPD1704, μPD1710, μPD1719 and μPD1722. It must be noted when executing the CKSTP instruction that the contents of timer F/F differ between μPD17P23 and the group of μPD1701, μPD1703, μPD1704, μPD1710, μPD1719 and μPD1722.

Note 4: Even if the CE pin changes from low to high level, the porgam flow does not move to address 0 when the setting of the timer F/F and the execution of the TTM instruction overlap. If this is the case, the system judges that the timer F/F is set by the TTM instruction execution and the timer F/F is reset. This point must be kept in mind when a power failure detection is done by the TTM instruction. That is, you should know that the TTM instruction has higher priority than the setting of the timer F/F when they overlap. Therefore, the clock does not become incorrect and a misjudging of a power failure does not occur. However, if the TTM instruction execution and the timer F/F setting happen to coincide in the following program, the program does not jump to address 0 forever. (The timer F/F will not be reset.)

```
A:  TTM
    JMP   B
    JMP   A
B:  ┆
    ┌─────┐ ◄──── 3n Step program
    │     │       (n <1249)
    └─────┘
    JMP   A
```

In this example the program executes the normal TTM instruction, skips the next 'JMP B' instruction because the timer F/F is reset, and executes 'JMP A' instruction. Therefore, it cycles this loop. The cycle is 100 μs. (3 steps) The timer F/F is set once every 125 ms and the operations of B are performed. This operations take (3n + 3) steps (multiple of 100 μs). If the CE pin happens to change from low to high level during the TTM instruction execution in this program, operations in B are done due to the judgement that the timer F/F is set by the TTM instruction. However, the timer F/F is set again (125 ms later) when the next TTM instruction is executed because the time interval from the last TTM execution to the next TTM execution is a multiple of 100 μs. Therefore, the timer F/F will not be reset and the program cycles this endless loop. This problem occurs in a program where the TTM instruction is executed in every 125 ms. **If this is the case, change the program so that the TTM instruction is not executed after 125 ms (3750 steps) from the TTM execution.**

1.6 INTERVAL PULSE (ITP)

The interval pulse is the 60 %-duty pulse, which is output every 5 ms, and it can be tested by the TIP instruction.

The pulse output cannot be reset even by the execution of TIP instruction as no flipflop (F/F) is provided. It is possible to make out the timer of exact multiplex of 5 ms when the TIP instruction is constantly executed to obtain the edges of interval pulses.

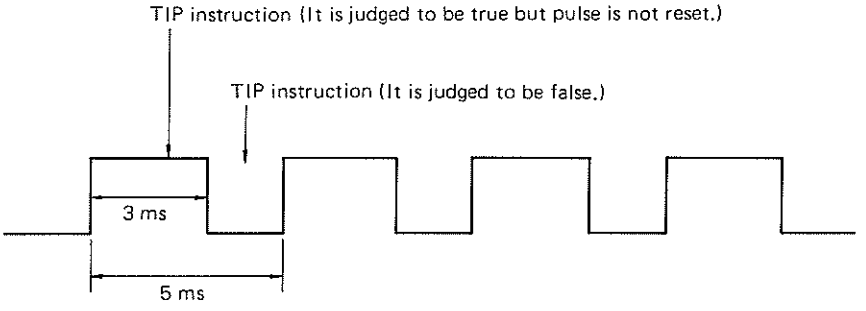


Fig. 1-5 Interval Pulse Timing

1.7 UNLOCK F/F (UL F/F)

The pulses are output from the phase detector (ϕ-DET) at the period of reference frequency (f_r) when the PLL system is not locked or when f_r is not coincident with the divided output frequency of VCO. The unlock F/F is reset by this pulse or it is reset by the execution of TUL instruction. Accordingly, **the period of executing the TUL instruction should be longer than the f_r period**; otherwise the PLL system is considered to be locked even if it is not, and a false operation would be performed. The first TUL instruction should be executed after certain period longer than the f_r period following the execution of PLL instruction.

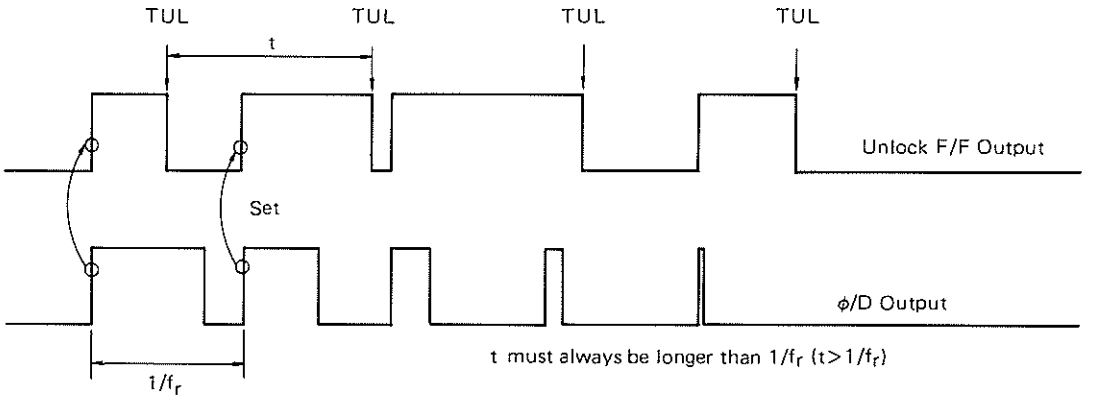


Fig. 1-6 TUL Instruction Execution Timing

1.8 CARRY F/F (C F/F)

The carry F/F is set if the execution of any operation instruction resulted in a carry or a borrow or it is reset if neither carry nor borrow is resulted. The content of this carry F/F remains unchanged unless an operation instruction is executed; it can be further set or reset directly by the carry F/F set/reset instructions (STC and RSC instructions) or by the status word operation instructions (SS and RS instructions).

Note: Even if the interrupt is accepted, the content of carry F/F is not automatically saved.

1.9 BANK F/F (B F/F)

BANK F/F is used for the BANK assignment of data memory (RAM) and for the addressing of the port groups. The 128 words of RAM is divided into BANK0 (64 words) and BANK1 (64 words) and BANK2 (64 words) and BANK3 (64 words). Data processing in each BANK must be done after the BANK assignment (execution of BANK0, BANK1, BANK2 and BANK3 instructions) is carried out. Data processing between BANK0 and BANK1 is done via the general register (address 00 to 0FH in BANK0).

If addresses from 00 to 0FH in BANK0 is used as a general register, the BANK assignment is not required and it can be accessed from each BANKs. However, if it is used as memory, BANK0 should be assigned.

The BANK F/F is used for the addressing of the port groups. Port addressing is done by the two bits in the operand of the instruction and the contents of the BANK F/F. (See the section of 3. PORT.)

The BANK F/F is automatically reset and BANK0 is assigned when power is supplied (V_{DD} = Low → High) for the first time or when the CE pin changes from low to high level (when the device is reset).

Note: Contents of the BANK F/F is not automatically saved even if an interruption is accepted.

1.10 INT F/F AND INTE F/F

The INT F/F is unconditionally set by **the fall edge** of a signal being applied to the $\overline{INT}$ pin. If the internal INTE F/F is in the enable state at the time, an interrupt request can be accepted. If it is in the disable state, the interrupt request cannot be accepted. However, as far as the INT F/F is set, an interrupt request can be accepted as soon as the INTE F/F becomes enable state.

To make the INTE F/F enable state, execute the EI instruction; to make it disable state, execute the DI instruction.

The INTE F/F can be reset by executing the DI or RS instruction which makes the INTE F/F disable state while the INTE F/F is in the DI state. **When an interrupt request is received, DI status is set automatically and program flow is returned to address 1 (interrupt processing routine).** The address of an instruction to be executed after the instruction by which the interrupt request has been made is stored in the stack. Or, the jump address is stored in the stack if the JMP instruction is being executed. If an instruction under execution has the skip function when an interrupt request is made, the evaluation of the skip conditions is also stored in the stack. The contents of the stack as well as the program flow are restored when the RT instruction is executed in the interruption handling routine. The contents of the carry F/F and the BANK F/F should be saved by the program if they are likely to be destroyed by the interruption handling because they are not stored in the stack.

One level of stack is needed for the interruption handling so that the stack level control is required.

The $\overline{INT}$ pin can be used as a regular input port (by using the TITT and TITF instructions). **It should be noted that the $\overline{INT}$ pin becomes TRUE when the low level is input and it becomes FALSE when the high level is input. This is because the $\overline{INT}$ pin is active low.**

Note: The high level should be input to the $\overline{INT}$ pin before being accessed by the program. If it is held under the low level, low level input is always read to be high level until high level input is given. Once the high level is input, the $\overline{INT}$ pin can be operated normally. Therefore, the $\overline{INT}$ pin must be pulled up to the V_{DD} via a register on the applied circuit.

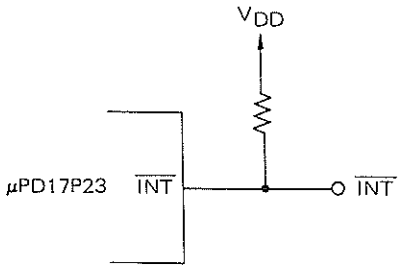


Fig. 1-7 Application Circuit of $\overline{INT}$ Pin

1.11 STATUS WORD

The status word is to split the internal status of device, which must be made known for the program execution of which must be designated without fail, into four bits each, and thereby allows to test, set or reset the status by the program. Two different kinds of the status words are available, status word 1 and 2, both of which are connected any of the folloiwng pins or F/F input/output:

1) Status Word 1 (write only word)

Operation instructions: SS, RS

#3	#2	#1	#0
BANK F/F1	BANK F/F0	Carry F/F	INTE F/F

The status word 1 can be set or reset by the SS, RS or EI instruction.

2) Status Word 2 (read only word)

Operation instructions: TST, TSF

#3	#2	#1	#0
BANK F/F1	BANK F/F0	CE pin	$\overline{INT}$ pin

The content of status word 2 can be judged by the TST, TSF or SBK0 instruction.

2. PLL

2.1 REFERENCE FREQUENCY GENERATOR (RFG)

Seven different kinds of the reference frequencies, viz., 1 kHz, 5 kHz, 6.25 kHz, 9 kHz, 10 kHz, 12. 5 kHz and 25 kHz, can be generated by dividing the external crystal (4.5 MHz). Any kind of the reference frequency can be chosen by the program (data of the control word).

2.2 PHASE DETECTOR (φ-DET)

This circuit detects the difference of phases between the reference frequency (f_r) and the VCO output, which is divided by the programmable divider. The output of this circuit is input into the internal charge pump and the following pulses are then output to the EO_1 and EO_2 pins:

(1) $f_r > f_{OSC}/N$: Low level
(2) $f_r < f_{OSC}/N$: High level
(3) $f_r = f_{OSC}/N$: Floating

where f_{OSC} represents the VCO oscillation frequency and N the dividing ratio of programmable divider.

2.3 PROGRAMMABLE DIVIDER (P/D)

The programmable divider is a binary-down counter consisting of a swallow counter and a programmable counter. The swallow counter is a 5-bit presettable down counter, and the contents of NR0 (4-bits) and N_F register (1-bit) out of N registers can be preset at the period of reference frequency.

The programmable counter consists of 12 bits into which the contents of NR1 to NR3 out of the N registers are preset and counted down simultaneously with the swallow counter.

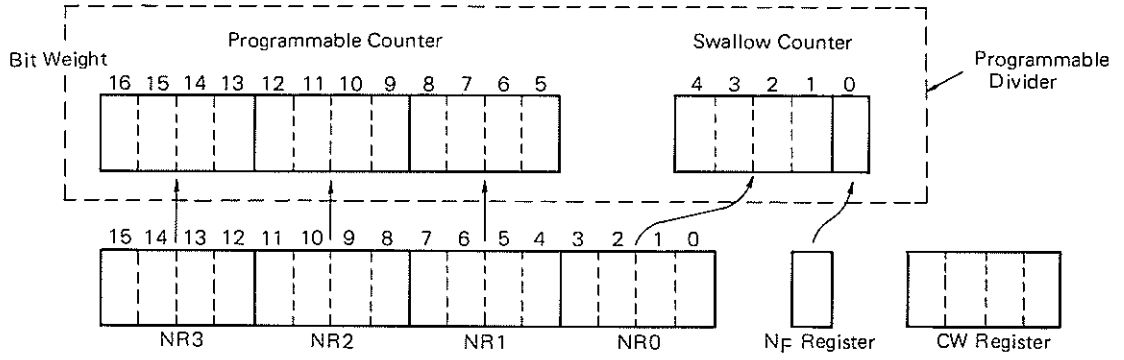


Fig. 2-1 Programmable Divider Configuration

2.4 PLL REGISTER

To control PLL of μPD17P23 needs the following three information:

(1) Dividing ratio (N)

(2) Reference frequency (f_r)

(3) Dividing method (direct and pulse swallowing methods)

The PLL register stores the above three information; and it is composed of N register (16 bits) and N_F register (1 bit), both of which set the dividing ratio, and control word register (4 bits) that sets the reference frequency and the dividing method. These registers correspond to N's words, N_F bit and control words (CW) of the data memory (RAM) respectively. The contents of above memory are **all transferred to PLL register at a time by the PLL instruction.**

The N's word is assigned to Addresses 00H, 10H, 20H, and 30H in the RAM, the N_F bit to the highest bit of a given general register, and CW to any RAM (BANK0 to BANK3) region excepting one word that contains the N's word and N_F bit.

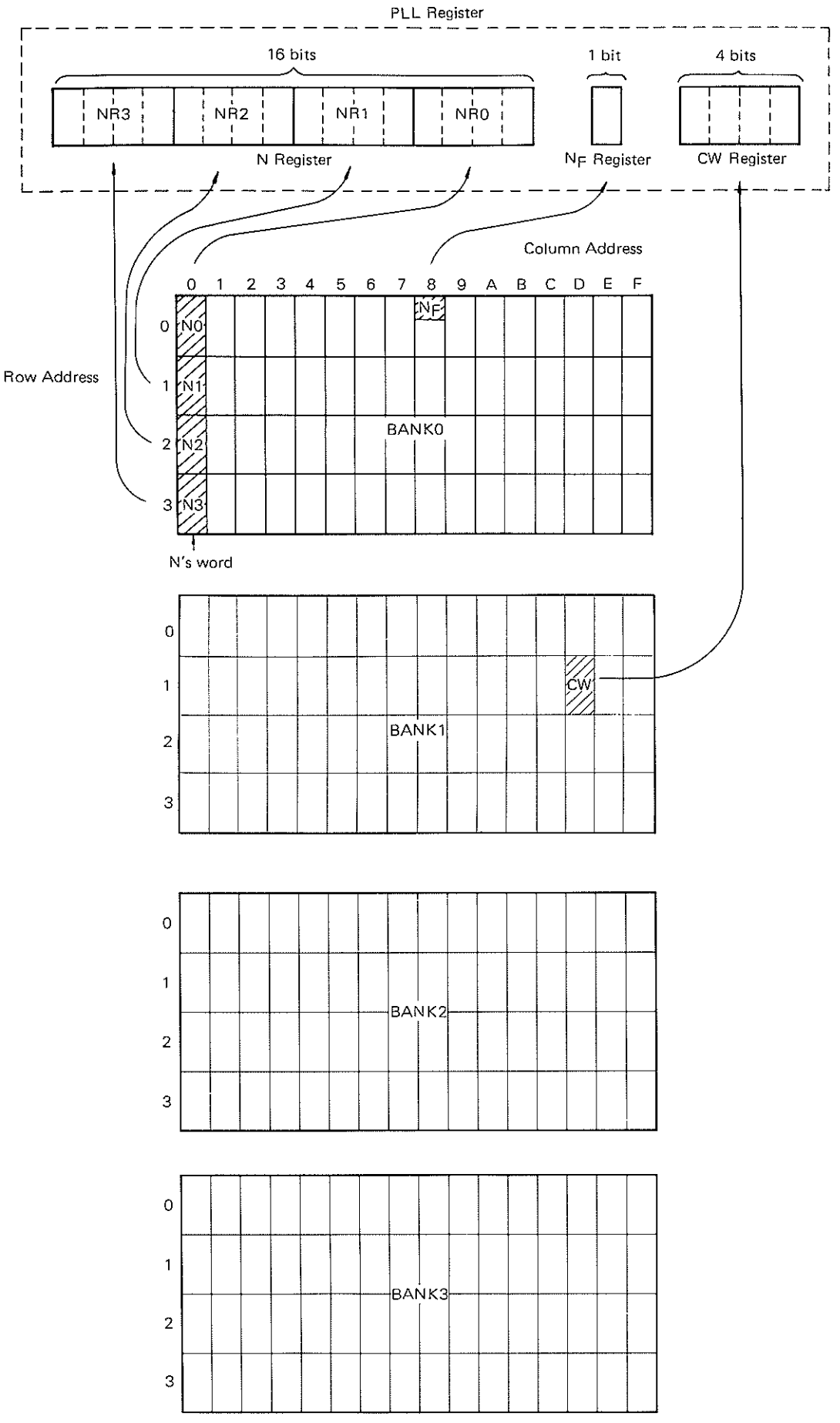


Fig. 2-2 Operation during PLL Instruction Execution

In the example shown Fig. 2-2, the control word (CW) is provided on BANK1, and the BANK1 instruction has to be executed before executing the PLL instruction. If the PLL instruction is executed while the control word is provided on BANK0, the content of Address 1DH on BANK0 is transferred to the CW register as the CW data, and the reference frequency, etc. cannot be set correctly.

Table 2-1 Control Word Code

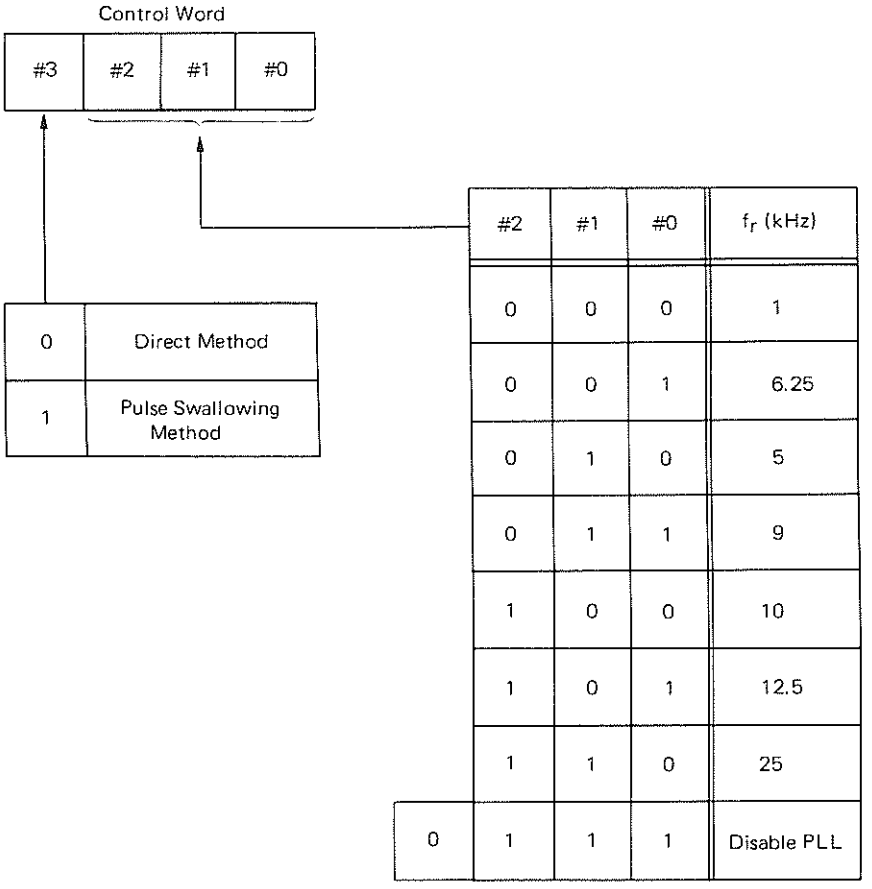


Table 2-1 shows control word data codes. Seven reference frequencies can be selected. The highest bit (#3) of the control word selects the frequency dividing method. The direct method is selected when the bit is "0", and the pulse swallowing method is selected when the bit is "1".

Pin VCOL is selected when the direct method is chosen, and frequencies 0.50 to 30 MHz (V_{in} = 0.3 V_{p-p}) can be input to Pin VCOL. At this time, the input frequency can be divided directly by the value of the programmable divider.

Pin VCOH will be selected and frequencies 10 to 150 MHz (V_{in} = 0.3 V_{p-p}) can be input to Pin VCOH when the pulse swallow system is chosen. At this time, the frequency input to Pin VCOH is introduced to the programmable counter through the 1/32 and 1/33 two-modulus prescaler.

The high and low limits of the frequency dividing ratios of the direct and pulse swallowing methods are shown in Table 2-2.

The PLL Disable mode can be set up by executing the PLL instruction after setting 07H in the control word (CW). This function is generally used to operate the clock function without executing the CKSTP instruction when Pin CE is set to the low level in case the specification has a clock function. Low current-consumption operation is possible by operating only the CPU while stopping the PLL operation.

Table 2-2 Status of pins VCOL and VCOH by Frequency Dividing Method

CW #3	Frequency Dividing Method	State of Pins VCOL and VCOH	Input Voltage	Input Frequency	Frequency Dividing Ratio
0	Direct Method	Pin VCOL = Active (Pin VCOH = Pull-down)	0.3 V_{p-p} MIN.	0.50 to 30 MHz	16 to ($2^{12}-1$) (1 step)
1	Pulse Swallowing Method	Pin VCOH = Active (Pin VCOL = Pull-down)	0.3 V_{p-p} MIN.	10 to 150 MHz	1024 to ($2^{17}-1$) (1 step)

2.5 PLL INFORMATION SETTING

The PLL information (frequency dividing ratio and dividing method and reference frequency) is set by a program. The dividing value of the programmable divider is set as follows:

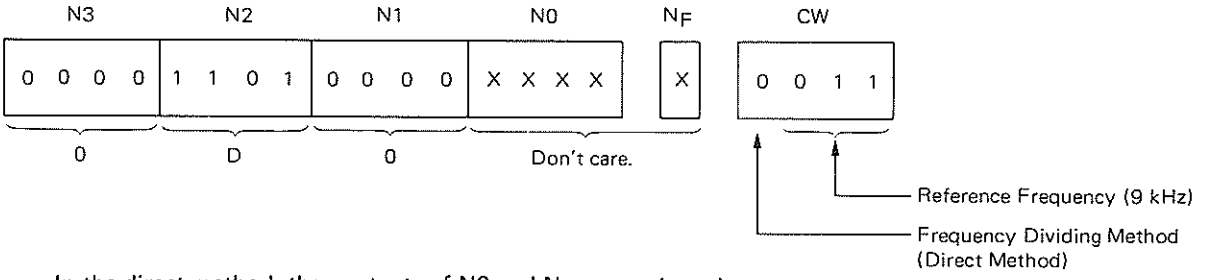
1. Direct Method

$$N = \frac{f_{VCOL}}{f_r}$$

f_{VCOL} : Input frequency to Pin VCOL
f_r : Reference frequency

Example: Reception of MW
(Reception frequency : 1422 kHz, reference frequency : 9 kHz, IF frequency : 450 kHz)

$$N = \frac{1422 + 450}{9} = 208$$
$$= 0D0H \text{ ("H" denotes a hexadecimal code.)}$$



In the direct method, the contents of N0 and N_F are neglected.

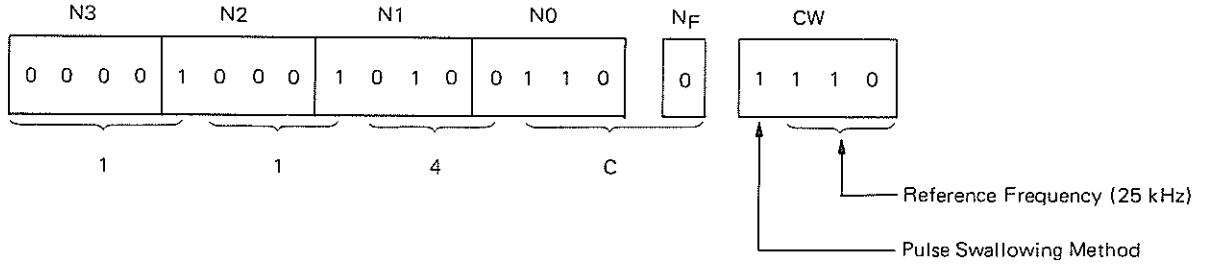
2. Pulse Swallowing Method (When VHF instruction is excuted)

$$N = \frac{f_{VCOH}}{f_r}$$

f_{VCOH} : Input frequency to Pin VCOH
f_r : Reference frequency

Example: Reception of FM (U.S.A. band)
(Reception frequency : 100.0 MHz, reference frequency : 25 kHz, IF frequency : 10.7 MHz)

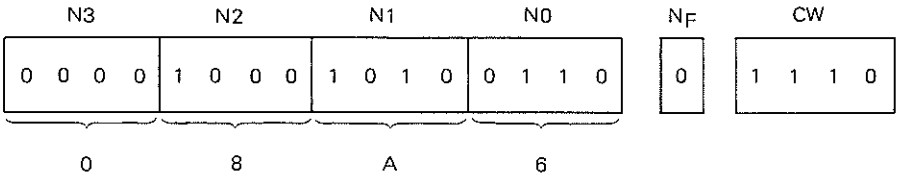
$$N = \frac{(100.0 + 10.7) \times 10^6}{25 \times 10^3} = 4428$$
$$= 114CH \text{ ("H" denotes a hexadecimal code.)}$$



In this example, the N_F bit is varied one by one as the lowest bit, and the VCO oscillation frequency varies 25 kHz for each variation this time. Vary one by one beginning with N0 when varying 50 kHz each. Vary two by two to change 100 kHz. Vary four by four to change 200 kHz.

In the foregoing example, the value of N is decided using the N_F bit as the lowest bit. Programming can be understood easier if setting is made by dividing the bits by four bits each starting with N0. Therefore, calculations shall be made considering the reference frequency to be 50 kHz.

$$N = \frac{(100.0 + 10.7) \times 10^6}{50 \times 10^3} = 2214$$
$$= 8A6H$$



As shown above, the same values as those when the reference frequency is 25 kHz are obtained.

As these two examples show, 17 bits from Bit N_F will be effective when the pulse swallowing method is selected and 12 bits from Word N1 will become effective when the direct method is selected.

3. PORTS

The μPD17P23 provides Ports A (PA_3 to PA_0) and C (PC_3 to PC_0) as input and output ports. It also provides Ports B (PB_3 to PB_0) and D (PD_3 to PD_1) as output-only ports. An output port Port L (PL_3 to PL_0), which is also used by the LCD segment, is additionally available.

The following ports are available as internal ports: Port E (PE_3 to PE_0), Port F (PF_3 to PF_0), Port G (PG_3 to PG_0), Port H (PH_3 to PH_0), Port J (PJ_3 to PJ_0), Port K (PK_1 and PK_0), Port N (PN_3 to PN_0), Port Q (PQ_3 to PQ_0), Port R (PR_3 to PR_0), and Port S (PS_3 to PS_0).

The internal ports can be handled by the same method as that for the external ports except that the internal ports have no external pins. The port operation instructions such as the IN, OUT, SPB, and RPB instructions can be used as they are.

These ports are addressed by direct addressing of two bits of the instruction operand part and by bank designation by BANK F/F0 and BANK F/F1. Table 3-1 shows the relationship between BANK F/F and BANK. Table 3-2 and 3-3 show port addresses and list of ports, respectively.

Table 3-1 Bank Designation

BANK	BANK F/F 0	BANK F/F 1
0	0	0
1	0	1
2	1	0
3	1	1

Table 3-2 Port Addresses

Direct Add.		BANK			
#1	#0	0	1	2	3
0	0	Port A	Port E	Port J	Port N
0	1	Port B	Port F	Port K	Port Q
1	0	Port C	Port G	Port L	Port R
1	1	Port D	Port H	—	Port S

Bank designation is common to port and RAM addressing. Therefore, be sure to set the bank to 0 when accessing the RAM of BANK0 after accessing a BANK1 to BANK3 port.

Example:

```
        MVI     0AH, 1111B  ;  Set "0FH" at Address 0AH of the register.
        MVI     0BH, 1101B  ;  Set "0DH" at Address 0BH of the register.
FLOOP:
        BANK1
        OUT     3, 0AH      ;  Output the content of Address 0AH to PH
        OUT     2, 0BH      ;  Output the content of Address 0BH to PG
        BANK0               ;  Return to BANK0 after accessing the internal port.
        CAL     WT1SEC      ;  Call the subroutine to wait for one second.
        SI      0BH, 0100B  ;
        SIS     0AH, 0      ;
        JMP     FLOOP
```

In this example, CGP is accessed and output is made in 64 stages for one second each from lower to higher frequencies after frequency-dividing the reference frequency of 18 kHz.

Table 3-3 List of Ports

BANK	Direct Add.	PORT	I/O	FUNCTION
0	0	PA_0/AMIF	I/O	General purpose input/output port or AMIF counter input pin
		PA_1/FMIF		General purpose input/output port or FMIF counter input pin
		PA_2/SI		General purpose input/output port or serial data input pin
		PA_3/$\overline{SCK}$		General purpose input/output port or serial clook pin
	1	PB_0/SO	O	General purpose output port or serial data output pin
		PB_1 to PB_3		General purpose output port
	2	PC_0 to PC_3	I/O	General purpose input/output port
	3	PD_1 to PD_3	O	General purpose output port
1	0	PE_0 to PE_3	O	A/D converter comparison data and serial I/O data latch
	1	PF_0 to PF_3		
	2	PG_0 to PG_3	O	CGP control data
	3	PH_0 to PH_3		
2	0	PJ_0 to PJ_3	O	Key source (Pins KS_0 to KS_{15}) control data
	1	PK_0 to PK_1		
	2	PL_0 to PL_3	O	General purpose output port (Jointly used with Pins LCD_{24} to LCD_{27} and the LCD control word has to be set when using it as a port.
3	0	PN_0 to PN_3	I	IF counter data (PR_2, PR_3, and PS_0 to PS_3 are used jointly with CGP control data)
	1	PQ_0 to PQ_3		
	2	PR_0 to PR_3		
	3	PS_0 to PS_3		

3.1 PORT A

Port A (PA_3, PA_2, PA_1, and PA_0) allows setting of input or output in one-bit units. Input and output can be set by the content of Address 1FH of BANK0 in the data memory (RAM) called **the PAIO word.** To set as an input port, set "0" in the bit of the PAIO word corresponding to the port. Set "1" when setting as an output port.

	#3	#2	#1	#0
PAIO Word (Address 1FH)	PA_3	PA_2	PA_1	PA_0

Example 1 : To set PA_3 to PA_0 as an output port

	#3	#2	#1	#0
PAIO Word (Address 1FH)	1	1	1	1

Example 2: To set PA_3 as an output port and PA_2, PA_1 and PA_0 as an input port

	#3	#2	#1	#0
PAIO Word (Address 1FH)	1	0	0	0

As shown above, Port A requires an input or output instruction is executed after setting input or output to the PAIO word. When once set, the input or output mode is maintained unless the content (data at Address 1FH) of the PAIO word changes.

Example :

```
BANK0
MVI     1FH, 1111B  ;  Set all bits of Port A in the output port.
  ⋮
MVI     08H, 1100B  ;  Set PA3, PA2 = high, PA1, PA0 = low
OUT     0, 08H      ;  Output PA3 and PA2 = high, PA1 and PA0 = low
  ⋮
```

But when the power is switched on (V_{DD} = Low to High), when the CKSTP instruction is executed, and when the CE pin is changed from low to high level.

Port A is automatically switched to input mode. Note that the PAIO word contents and port A input/output status may not be matched at this time, and that port A remains in input status until the PAIO word contents are set.

When output port mode is set by the PAIO word, the output latch circuit contents are passed directly to the port. Note that the output latch circuit contents are undefined when the power is switched on (V_{DD} changed from low to high), and are held when the clock is stopped. Therefore, where it is necessary to avoid an undefined output status when output port is first defined by PAIO word, specify output status by the OUT instruction before setting the port status.

If output status is specified by the SPB or RPB instruction, the contents of specified bit to "0" may be changed. Therefore, the output status must be specified by the OUT instruction.

PA_0 and PA_1 of Port A can also be used as input pins of the IF counter by setting PA_0 and PA_2 to the input mode and by selecting an input pin by the IFCW (IF counter control word). For the details, refer to the section describing the IF counter.

The ISB data of IFCW is set to "00" when the port A is used as I/O ports.

PA_3 and PA_2 of Port A can be used as serial I/O operate as the shift clock ($\overline{SCK}$) and serial input (SI) pins, respectively, by the SIO instruction. **Set Bits #3 and #2 of the PAIO word to "0", that is, PA_3 and PA_2 of Port A to the input mode, before executing the SIO instruction to use these pins as the $\overline{SCK}$ and SI pins.** (Refer to Chapter 5. SERIAL I/O).

Note: By leaving PA_0 and PA_1 designated with the IF counter when the level of Pin CE is low, the input amplifier is operated by noise, etc. to increase the current consumption. Therefore, PA_0 and PA_1 should be designated with a port when the level of Pin CE is low.

3.2 PORT B AND PORT D

Ports B (PB_3 to PB_0) and D (PD_3 to PD_1) are output-only ports of the CMOS type. Normally, these ports are accessed by an output instruction (OUT, SPB, or RPB instruction). The data currently being output is read in the designated register when an input instruction (IN instruction) is executed. The output data does not change by executing the IN instruction.

The high level (V_{DD} potential) is output when "1" is output during the execution of an output instruction. The low level (GND potential) is output when "0" is output.

The Port D consists of three bits, PD_3 to PD_1. "0" is always assigned to data corresponding to Bit #0 when the IN, TPT, or TPF instruction is executed to the Port D.

The output contents of Ports B and D are not fixed when power is turned on (V_{DD} = low to high). Therefore, these ports must be initialized by a program when turning on power. The output contents of Ports B and D do not change, and these ports continue to output the data that was output before when the clock is stopped, or when the level of CE pin change from high to low or low to high at the time when V_{DD} is high level.

Execute the program shown in the example to make the port state constant when the clock is stopping.

Example: Port resetting during clock stop

In the μPD17P23, Ports B and D maintain the state immediately before even if the clock stop instruction is executed. Reset the ports before the clock stop instruction as shown below to disable the current flowing from the ports (ports to low) in a clock stop state:

```
      :
①  TCET          ;  Does not skip if the level at Pin CE is lower than 100 μs before ① .
②  TCEF          ;  Skips if the level at Pin CE is lower than 133.3 μs before ② .
③  JMP NOTSTP    ;  To NOTSTP (does not CKSTP) if the level at Pin CE is decided to be high at ① or ② .
④  BANK0         ;  Set the bank to 0.
   RPB 1,1111B   ;  Reset Port B entirely.
   RPB 2,1111B   ;  Reset Port D entirely.
⑤  CKSTP         ;  Synchronize to 8 Hz and branch to Address 0 if the level of Pin CE is high after ③ .
                    (Resetting is applied.)
⑥  JMP $−1       ;  In this case, go round the loop between ⑤ and ⑥ until the 8 Hz signal rises. If the
                    level of Pin CE is still low at ⑤ , stop the clock.
```

Note: The methods described in ① and ② are taken to prevent maloperation as resetting is not possible even when a low level lower than 134 μs is input to Pin CE. A low level or 100 μs (three instruction cycles) or higher will be required to enable a decision as a low level in an instruction (TCET or TCEF). The CKSTP instruction stops the clock if the level of Pin CE is lower than 133.3 μs (four instruction cycles) before this instruction is given.

PB_0 of Port B is also used as a serial data output pin (SO) of serial I/O. As in the μPD1709, the state of PB_0 is "don't care" (can be 0 or 1) with the μPD17P23 when PB_0 is used as Pin SO. ("1" has to be output to PB_0 at this time with the μPD1707, μPD1711, and μPD1712.)

3.3 PORT C

Port C (PC_3 to PC_0) is a CMOS push-pull input/output port. Unlike Port A, designation of input and output per bit is not possible. The port can be used as 4-bit output or 4-bit input. Input and output can be changed freely during program execution. The port becomes an output port when an output instruction (OUT, SPB, or RPB instruction) is executed with Port C. The port becomes an input port when an input instruction (IN instruction) is executed. (Input and output do not change by a port test instruction (TPT or TPF instruction).

Note: Port C of μPD17P23 is different from input/output port of μPD1704 and μPD1710.
Input/output port of μPD1704 and μPD1710 can be set as an input port or an output port at one time only after to reset.

Port C becomes an input port when power is turned on (V_{DD}=low to high), when the clock is stopping, or when the level at Pin CE changes from low to high (when system resetting is applied).

If the SPB or RPB instructions are executed, for the port is switched from input port to output port, contents of the bit (port output) specified by "0" is influenced by input condition of the pin. (for example, "SPB 1010B" is executed and the port becomes output port from input port, outputs of PC_2 and PC_0 are indefinite just then. Outputs of PC_3 and PC_1 become high level naturally.) Therefore, if the port is set to output port from input port, OUT instruction is used.

Example: Port initializing when power is turned on.

```
  START :
000       MVI   1FH, 1111B ;  Set all the bits of Port A to output port.
001       MVI   0AH, 0     ;  Set port initialize data
002       OUT   0, 0AH     ;  Port A (PA3 to PA0) = All Low
003       OUT   1, 0AH     ;  Port B (PB3 to PB0) = All Low
004       OUT   2, 0AH     ;  Port C (PC3 to PC0) = All Low (Set to output port)
005       OUT   3, 0AH     ;  Port D (PD3 to PD1) = All Low
006       TTM              ;  If the timer F/F is set.
007       JMP   BACKUP     ;  The RAM is not initialized at this time.
008       MVI   00H, 0     ;  RAM initialize
009       MVI   01H, 0     ;  RAM initialize
            ⋮
  BACKUP :
```

3.4 PORT L

Port L is an output port of a 4-bit CMOS type jointly used with the LCD segment output. The μPD17P23 has 24 dedicated pins (LCD_{23} to LCD_0) as LCD segment outputs and four pins (LCD_{27}/PL_3 to LCD_{24}/PL_0) jointly used with Port L. A selection is possible whether to use two each of these four pins as a port or as LCD segment output.

Whether or not to use the pins as Port L or as LCD segment output is selected by the LCD control word. The LCD control word consists of F/Fs for four bits. The flip flops are written with the content of the data memory (RAM) designated by the first operand when the content of the second operand of the LCDD instruction is 0FH. Table 3-4 presents the match of each bit between the memory content at that time and LCD control word, as well as the functions of the bits.

All the LCD control words are reset to "0" when power is turned on (V_{DD} = low to high) or when the clock is stopped. Therefore, Pins LCD_{27}/PL_3 to LCD_{24}/PL_0 change to LCD segment output and the LCDE also changes to "0" simultaneously, the low level is output to thse pins.

Except that the state during resetting is different as mentioned above and that its output can be selected to LCD segment output, Port L operates exactly the same as the output-only ports of Ports B and D.

When Pins LCD_{27}/PL_3 to LCD_{24}/PL_0 are used as LCD segment output pins (when both or either of PLEL and PLEU is "0"), the bits of Port L corresponding to the pins designated as the segment, namely LCD_{25}/PL_1 and LCD_{24}/PL_0, when PLEL = 0, that is, #1 and #0, and #3 and #2 when PLEU = "0", can be used as data memory.

For example, Port L can be used as a 4-bit memory when PLEL = PLEU = "0", and writing and reading are possible by the OUT and IN instructions.

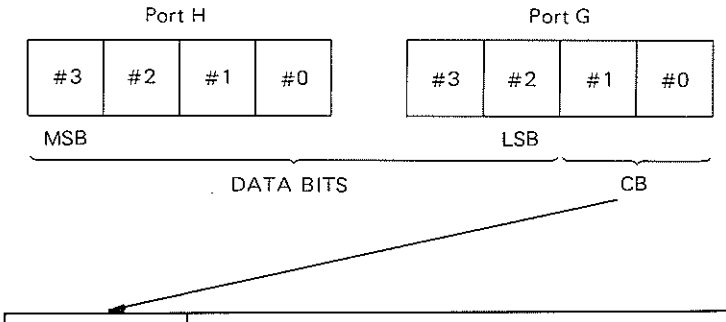
4. CGP

The CGP (clock generator port) is an output port of the CMOS type combining both VDP (variable duty pulse) generation and SG (signal generator) functions and is controlled by Ports G and H, which are internal ports.

Except that Ports G and H are internal ports, all the instructions regarding ports used with other ports can be executed with them. If the IN instruction is executed with Ports G and H, the data currently set in Ports G and H is read in the designated registers.

The CGP has four modes, and these modes are designated by Bits #1 and #0 of Port G called control bits (CBs).

Table 4-1 Control Bit (CB) Codes and Functions



CB (Control Bits)		Function
#1	#0	
0	0	PG #2 through-mode: to output the content of Port G Bit #2 to the CGP pin
1	0	VDP mode: to vary and output 64 steps of duty over the frequency 2.69 kHz.
0	1	SG mode 0: to output 64 divided frequencies of 18 kHz
1	1	SG Mode 1: to output 64 divided frequencies of 180 kHz

The six bits between Bit #3 of Port H and Bit #2 of Port G are called data bits to set the duty value when the VDP mode is designated or data of divided values when the SG mode is designated. The MSB of data bits corresponds to Bit #3 of Port H and the LSB, to Bit #2 of Port G.

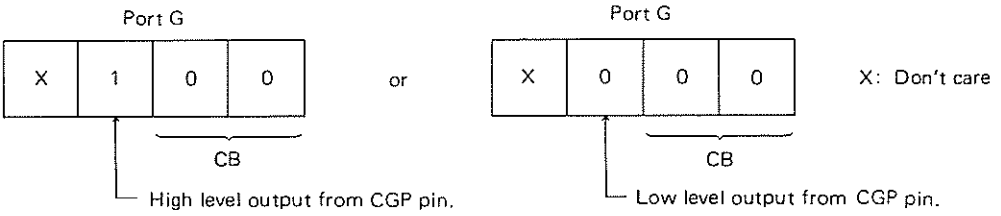
The level of Pin CGP becomes low when power is turned on (V_{DD} = low to high) or when the clock is stopped. The contents of the data latches of Ports G and H do not change at that time. (However, the contents of the data latches when power is turned on [V_{DD} = low to high] are not constant.)

Pin CGP becomes active when data is set in Port G and outputs a pulse (or a high/low level). The instruction (SPB 2, 0) that does not set any bit of Port G or one (RPB 2, 0) that does not reset any bit is used to make Pin CGP active leaving the contents of the data latches as they are. Pin CGP does not become active even if the instruction to operate Port H only is executed.

The level of Pin CGP does not become low if the level of Pin CE changes between low and high, and the previous state is maintained as it is. If the level of Pin CGP is desired to be changed to low at the time CE = high to low, the level of Pin CE is tested by the TCET or TCEF instruction. Based on the result of it, Pin CGP should be changed to the PG #2 through mode to execute the instruction to change to the low level. Output of X000B (Don't care what the bit is showed by X.) is required to Port G.

4.1 PG #2 THROUGH MODE

The data of Bit #2 of Port G is output to the CGP pin when CB=00B. When PG #2=1 then the high level is output or the low level if PG #2=0; and it can be used as one-bit output port.



Then the content of bit #3 of Port G is ignored and so is the content of Port H. It is therefore not required to output any data to port H under the PG #2 through-mode. The IF counter has to be used in this mode.

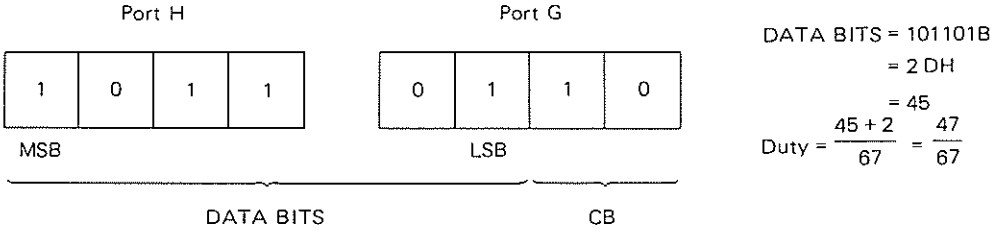
4.2 VDP MODE

The duty pulses as designated by the data, which are set to the data bits (PH_3 to PH_0, PG_3 and PG_2), are continuously output to the CGP pin when CB = 10B. The frequency of output pulse is 2.69 kHz. The following relation is established between the data set to the data bits and the duty of pulse to be output:

$$\text{Duty} = \frac{\text{Time of high level}}{\text{Period}} = \frac{\text{(data bits)} + 2}{67}$$

Hence the duty can be varied into 64 steps from 2/67 to 65/67. The duty can be varied fully into 66 steps, including the low or high output, when the above relation is used together with the afore-mentioned PG #2 through-mode.

Example:



As shown above, the bit #3 of Port H turns to MSB of the data bits and the bit #2 of Port G to LSB.

The pulses from the CGP pin are output at the time when 10B is set to CB, which is two lower-order bits of Port G. No pulses are, for instance, output with the data output to Port H when CB=00B, because it only sets the data to the data latch circuit of Port H.

Therefore, data has to be output first to Port H and then to Port G when setting up the VDP mode from a low level state (immediately after turning on power or immediately after releasing clock stop).

Otherwise the output from CGP pin is operated with the previous data to Port H until new data is output to it, and there would be certain period without any required duty level being output. It is however allowed to output data only to Port H or output data from Port H first if the device is operating under the VDP mode and no data to Port G are changed.

Program Example:

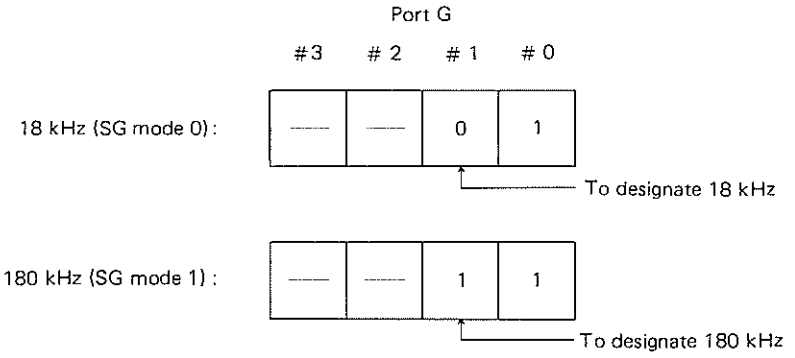
```
MVI     0AH, 1011B    ;  To set Port H data to the register 0AH
MVI     0BH, 0110B    ;  To set Port G data to the register 0BH
BANK1
OUT     3,0AH         ;  To output data to Port H
OUT     2,0BH         ;  To output data to Port G; pulse of duty 47/67 is output
BANK0
  ⋮
```

4.3 SG MODE

The CGP pin turns to the SG (signal generator) mode when changing the bit #0 (of Port G) of the control bits (CB) to "1". The SG mode is to output from the CGP pin the frequency pulse (duty=50 %) as designated by the data bits and the frequencies divided into 128 steps can be output under this mode. The following relation is established between the data, which is set to the data bits, and the pulse frequency (f_{OUT}) to be output:

$$f_{OUT} = \frac{f_B}{2\,(2 + (\text{DATA BITS}))}$$

where f_B (base frequency) is the reference frequency to be output to the CGP pin, which can be chosen from two kinds, 18 kHz or 180 kHz, depending on the value of CB bit #1 (bit #1 of Port G):



Similar to the VDP mode, the pulses from the CGP pin are output when data is output to Port G. Table 4-2 below shows the relation of frequencies to be output the CGP pin and the data bits:

Table 4-2 Outputs Frequencies in SG mode

DATA BITS			OUTPUT FREQUENCY		DATA BITS			OUTPUT FREQUENCY	
DEC	BINARY		mode 0 (Hz)	mode 1 (kHz)	DEC	BINARY		mode 0 (Hz)	mode 1 (kHz)
	PH	PG				PH	PG		
0	0 0 0 0	0 0	4500.000	45.0000	32	1 0 0 0	0 0	264.706	2.6471
1	0 0 0 0	0 1	3000.000	30.0000	33	1 0 0 0	0 1	257.143	2.5714
2	0 0 0 0	1 0	2250.000	22.5000	34	1 0 0 0	1 0	250.000	2.5000
3	0 0 0 0	1 1	1800.000	18.0000	35	1 0 0 0	1 1	243.243	2.4324
4	0 0 0 1	0 0	1500.000	15.0000	36	1 0 0 1	0 0	236.842	2.3684
5	0 0 0 1	0 1	1285.710	12.8571	37	1 0 0 1	0 1	230.769	2.3077
6	0 0 0 1	1 0	1125.000	11.2500	38	1 0 0 1	1 0	225.000	2.2500
7	0 0 0 1	1 1	1000.000	10.0000	39	1 0 0 1	1 1	219.512	2.1951
8	0 0 1 0	0 0	900.000	9.0000	40	1 0 1 0	0 0	214.286	2.1429
9	0 0 1 0	0 1	818.182	8.1818	41	1 0 1 0	0 1	209.302	2.0930
10	0 0 1 0	1 0	750.000	7.5000	42	1 0 1 0	1 0	204.545	2.0455
11	0 0 1 0	1 1	692.308	6.9231	43	1 0 1 0	1 1	200.000	2.0000
12	0 0 1 1	0 0	642.857	6.4286	44	1 0 1 1	0 0	195.652	1.9565
13	0 0 1 1	0 1	600.000	6.0000	45	1 0 1 1	0 1	191.489	1.9149
14	0 0 1 1	1 0	562.500	5.6250	46	1 0 1 1	1 0	187.500	1.8750
15	0 0 1 1	1 1	529.412	5.2941	47	1 0 1 1	1 1	183.673	1.8367
16	0 1 0 0	0 0	500.000	5.0000	48	1 1 0 0	0 0	180.000	1.8000
17	0 1 0 0	0 1	473.684	4.7368	49	1 1 0 0	0 1	176.471	1.7647
18	0 1 0 0	1 0	450.000	4.5000	50	1 1 0 0	1 0	173.077	1.7308
19	0 1 0 0	1 1	428.571	4.2857	51	1 1 0 0	1 1	169.811	1.6981
20	0 1 0 1	0 0	409.091	4.0909	52	1 1 0 1	0 0	166.667	1.6667
21	0 1 0 1	0 1	391.304	3.9130	53	1 1 0 1	0 1	163.636	1.6364
22	0 1 0 1	1 0	375.000	3.7500	54	1 1 0 1	1 0	160.714	1.6071
23	0 1 0 1	1 1	360.000	3.6000	55	1 1 0 1	1 1	157.895	1.5789
24	0 1 1 0	0 0	346.154	3.4615	56	1 1 1 0	0 0	155.172	1.5517
25	0 1 1 0	0 1	333.333	3.3333	57	1 1 1 0	0 1	152.542	1.5254
26	0 1 1 0	1 0	321.429	3.2143	58	1 1 1 0	1 0	150.000	1.5000
27	0 1 1 0	1 1	310.345	3.1034	59	1 1 1 0	1 1	147.541	1.4754
28	0 1 1 1	0 0	300.000	3.0000	60	1 1 1 1	0 0	145.161	1.4516
29	0 1 1 1	0 1	290.323	2.9032	61	1 1 1 1	0 1	142.857	1.4286
30	0 1 1 1	1 0	281.250	2.8125	62	1 1 1 1	1 0	140.625	1.4063
31	0 1 1 1	1 1	272.727	2.7273	63	1 1 1 1	1 1	138.462	1.3846

5. SERIAL I/O

The serial I/O is an 8-bit μCOM standard serial I/O that transfers and receives data synchronously with the internal or external clock. The following three pins are provided for the serial I/O:

SI	(concurrent with PA_2:	serial data input pin
SO	(concurrent with PB_0:	serial data output pin
$\overline{SCK}$	(concurrent with PA_3:	shift clock I/O pin (active-low)

All the three pins that are jointly used with other ports (PA_3, PA_2 and PB_0), cannot be used as the ports (PA_3, PA_2 and PB_0) at the timing as they are used as the serial I/O.

An 8-bit presettable shift register is used as the data buffer of this serial I/O; its 4 higher-order bits are assigned to Port F of the internal port and 4 lower-order bits to Port E. Hence data are written in or read out of the presettable shift register by the port operation instruction (OUT, SPB, RPB, IN instructions) that access to Port E and F. Four lower-order bits (Port E) of the presettable shift register are also used as the data latch under the A/D conversion so that it is not allowed to perform the serial I/O operation and the A/D conversion simultaneously.

Ports E and F of the presettable shift register are also used as data latches during A/D conversion. Therefore, serial I/O and A/D conversion operations cannot be performed simulaneously.

The serial I/O consists of SMR (shift mode register), PSR (presettable shift register), SCC (shift clock counter) and SCG (shift clock generator), the function of each being described below:

5.1 SMR (SHIFT MODE REGISTER)

SMR is a register consisting of three bits, SMR3, 1 and 0, that determines the mode of serial I/O. As the SIO instruction is executed, the immediate data of its operand is written into SMR, which however does not provide any bit corresponding to the bit #2 of the operand. The bit #2 of immediate data of the operand of the SIO instruction becomes therefore 'don't care'.

As the SIO instruction (SIO $b_3b_2b_1b_0$B) is executed, the following are set to SMR to start the respective mode operations:

Table 5-1 SMR Bit Functions

SIO b_3 b_2 b_1 b_0 B

Symbol	Name	Operation
SMR0	SO Enable bit	"0" : PB_0/SO pin function as PB_0 "1" : PB_0/SO pin function as SO
SMR1	Shift Enable bit	"0" : no shift takes place "1" : shift takes place
SMR3	Internal $\overline{SCK}$ Enable bit	"0" : to allow to input external clocks from $PA_3/\overline{SCK}$ pin "1" : to output internal clocks to $PA_3/\overline{SCK}$ pin

All bits of SMR are reset to "0" when the power is turned on (V_{DD}=low to high) or when the internal clock is stopped by the execution of CKSTP instruction.

(1) SMR 0 (SO Enable Bit)

Pin PB_0/SO will operate as an output pin of serial data, namely, as Pin SO, when "1" is set (SIO XXX1B instruction is executed) in SMR 0.

The data of the presettable shift register (PE, PF) is shifted in sequence beginning with the MSB and is output from Pin PB_0/SO when the SIO XX11B instruction is executed. The data of Pin PB_0/SO is output by synchronizing to the falling edge of the clock output from (or input to) $PA_3/\overline{SCK}$ at this time. The presettable shift register is shifted by the rising edge of the clock and reads data from Pin PA_2/SI simultaneously. The content output by PA_2 will be read if Pin PA_2/SI is operating as Pin PA_2 output at this time.

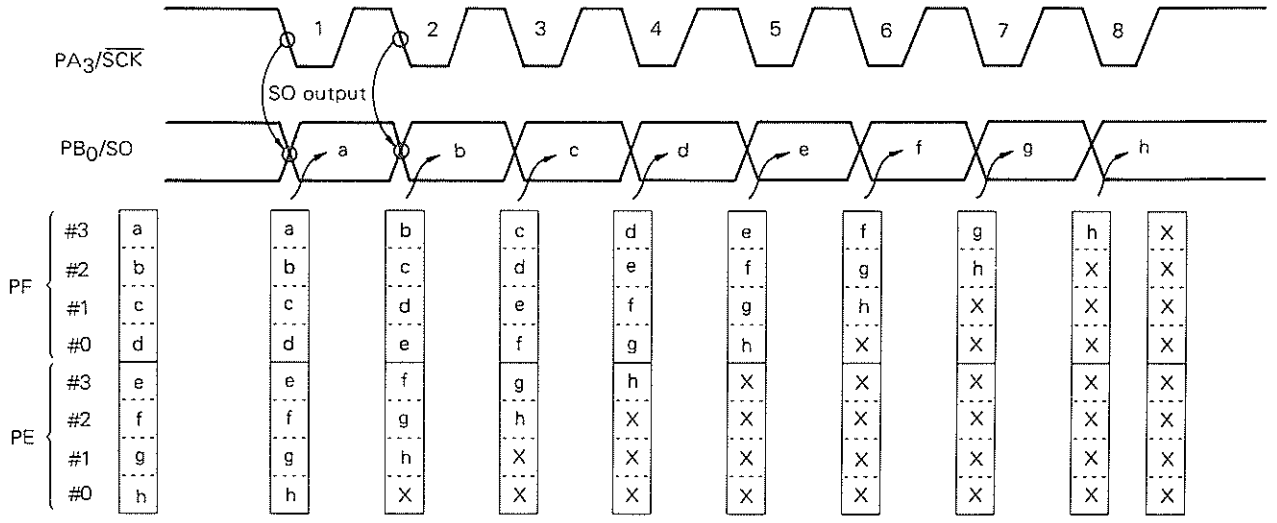


Fig. 5-1 SO Operation

By setting SMR 0 to "0" after SO operation, the output state of Pin PB_0/SO, that is, the content of the data latch of Port B that was output to the data latch of PB_0 before the SO mode was set will be output.

(2) SMR1 (Shift Enable Bit)

When "1" is set in SMR1, the contents of Pin PA_2/SI are shifted in sequence from the LSB (Bit #0 of Port E) of the presettable shift register to the MSB (Bit #3 of Port F) and is read by synchronzing to the clock output from (or input to) Pin $PA_3/\overline{SCK}$. The read timing is the rising edge of the clock.

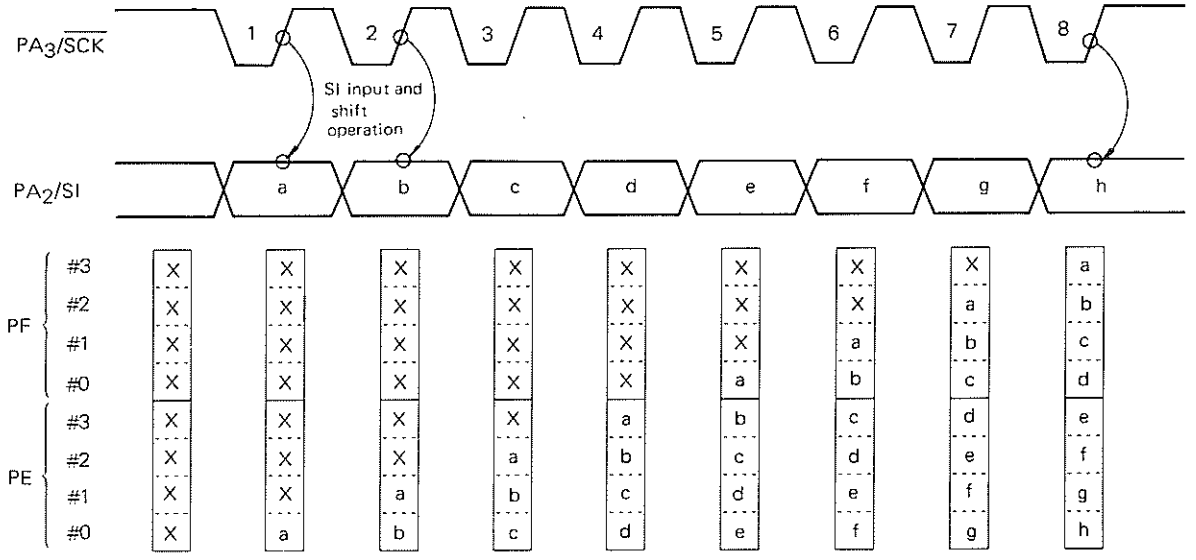


Fig. 5-2 SI Operation

PA_2 should be set to an input mode state (Bit #2 of the PAIO word set to "0") beforehand when inputting serial data from Pin PA_2/SI. The output contents of PA_2 are shifted and read in sequence in the presettable shift register (PE, PF) when PA_2 is set to the output mode. In other words, Pin PA_2/SI functions as PA_2 and all the contents of the presettable shift register will be written "1s" if "1" is output from PA_2 and with "0s" if "0" is output.

Be sure to set SMR1 to "0" during A/D conversion to disable the input of the presettable shift register shift clock.

Pins PA_2/SI and PB_0/SO operate simultaneously as Pins SI and SO, respectively, when "1" is set in SMR1 and "1" is set in SMR0 (by executing SIO XX11B).

The data output by Pin SO at this time is output by the falling edge of the clock output from (or input to) Pin $PA_3/\overline{SCK}$, and the data input to Pin SI is read by the rising edge of the clock. The serial I/O data is first output from Pin SO and is shifted and simultaneously read by Pin SI.

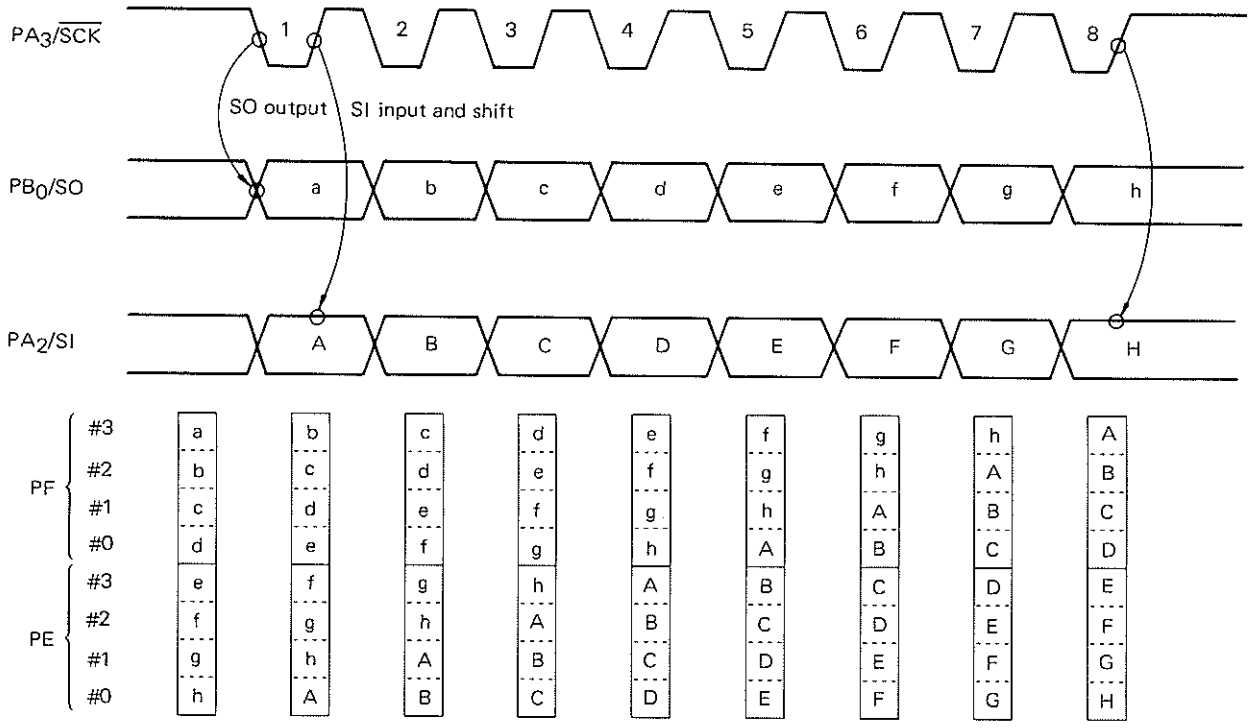


Fig. 5-3 SIO Operation

As shown in the above example, the presettable shift register (PF and PE) is sequentially shifted for data reading and writing.

Utilizing this example, simultaneous data exchanges will become possible by mutually connecting Pins SI and SO, which the μCOM standard has, and Pins SO and SI of μPD17P23.

As an ordinary application example, sending of serial data to external shift registers is possible.

The SIO 1X11B instruction is executed at this time. When this instruction is executed, data of the presettable shift register (PF and PE) is output by Pin PB_0/SO being sequentially shifted by synchronizing to the clock output by Pin $PA_3/\overline{SCK}$.

The presettable shift register is not shifted if the content of SMR1 is "0". Last data when the SIO instruction was executed prior to it is output by Pin SO. Therefore, SMR1 should always be set to "1" when serial I/O is used.

Example:

```
S OUT:
        ANI     PAIO, 0111B  ;  Set PAIO #3 to the input mode.
        SPB     PB, 0001B    ;  "1" is set in PB0.
        BANK1
        OUT     PE, 02H      ;  Set serial out data.
        OUT     PF, 03H      ;
        SIO     1011B        ;  Output data by the internal clock.
        TSET                 ;  Wait until eight clocks are output
        JMP     $-1
        SIO     0000B
        RT
```

(3) SMR3 (Internal $\overline{SCK}$ Enable Bit)

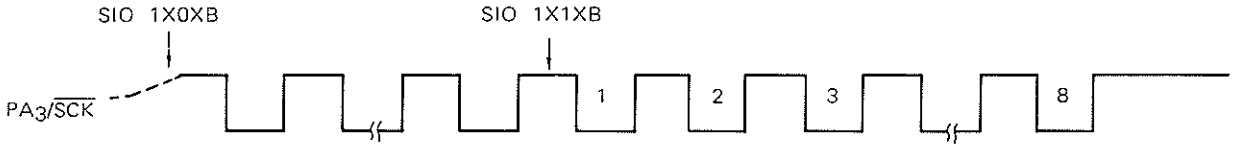
SMR3 is a bit to select whether to use an internal clock or an external clock as a shift clock. An internal clock (15 kHz, duty 50 %) is output by Pin $PA_3/\overline{SCK}$ by setting "1" in SMR3 and an external clock is input to Pin $PA_3/\overline{SCK}$ by setting "0".

When using Pin $PA_3/\overline{SCK}$ as Pin $\overline{SCK}$, that is, **when executing the SIO instruction, PA_3 should be set in the input mode (Bit #3 of the PAIO word is set to "0") beforehand.** Pin PA_3 should be pulled up at the same time. It should be noted that the device may be destroyed if Pin $\overline{SCK}$ is used by executing the SIO instruction while PA_3 is in the output mode (Bit #3 of the PAIO word is "1").

An internal clock is immediately generated and is output by Pin $PA_3/\overline{SCK}$ when "1" is set in SMR3 by the SIO instruction. When "1" is set in SMR1 before or simultaneously with setting of "1" in SMR3, the clock automatically stops after eight clocks are output.

The internal clock to be output synchronizes with the machine cycle. One clock corresponds to the time needed to execute two instructions (one instruction execution time is 33.3 μs). When the SIO 1X1XB instruction is executed, the clock stops after 16 instructions after the execution of this instruction.

In the execution of the SIO 1X0XB instruction, the internal clock is output continuously without being stopped.



By setting SMR3 to "0", an external clock can be input to Pin $PA_3/\overline{SCK}$. The frequency of the clock that can be input is between DC and 1 MHz. The presettable shift register shifts at this time synchronizing to the external clock to be input. However, in this mode, shift operation cannot be stopped automatically even when eight external clocks are input and shift operation is performed eight cycles. Shifting continues to next clock input. Therefore, the clock has to be stopped after inputting eight clocks each and reinput subsequent to data processing (read or write) of the presettable shift register.

Whether or not eight shift operations are performed after executing the SIO instruction regardless of the mode of the internal/external clock can be tested by the TSET (Test Shift End, then skip if True) or TSEF (Test Shift End, then skip if False) instruction.

During the external clock mode, the test result will be true only when shifting is performed 8 (2n + 1) cycles (n is a positive integer larger than 0). The test result will become false in all other cycle numbers.

The internal clock output is not stopped during output even if the level of Pin CE changes from high to low. The internal clock stops only in the following cases:

(1) When "0" is set in SMR3 (external clock mode). In this case, Pin $PA_3/\overline{SCK}$ has a high impedance (input mode) and caution should be exercised.
(2) If eight shift operations are performed, provided "1" is set in SMR1.
(3) The CKSTP instruction is executed. At this time, all the bits of SMR are automatically cleared to "0".

SMR SUMMARY

SMR0	PB State before SIO Instruction Execution	State of Pin PB_0/SO	
0	0	Low level output	Content of PB_0 is output as it is.
0	1	High level output	
1	0	Low level remains output.	
1	1	Operates as Pin SO	

SMR1	Content of $PAIO_2$ before SIO Instruction Execution	State of Pin PA_2/SI
0	0	PA_2 input port
0	1	PA_2 output port
1	0	Operates as Pin SI (Data to be input in Pin SI is input to presettable shift register.)
1	1	PA_2 output port (Data output in PA_2 is input to presettable shift register.)

SMR3	Content of $PAIO_3$ before SIO Instruction Execution	State of Pin $PA_3/\overline{SCK}$
0	0	PA_3 input port or external clock can be input (External clock can be monitored also by testing PA_3.)
0	1	PA_3 output port
1	0	Internal clock output (Internal clock can be monitored also by testing PA_3.)
1	1	Disable (Device may be destroyed.)

Principal Applications

Operation		SMR 3	SMR 1	SMR 0	Content of PB_0 before SIO Instruction Execution	Content of $PAIO_2$ before SIO Instruction Execution	Content of $PAIO_3$ before SIO Instruction Execution	Pin PB_0/SO	Pin PA_2/SI	Pin PA_3/$\overline{SCK}$
SI Operation	Internal clock output	1	1	0	X	0	0	PB_0	SI	$\overline{SCK}$ Output
	External clock input	0								$\overline{SCK}$ Input
SO Operation	Internal clock output	1	1	1	1	X	0	SO	PA_2	$\overline{SCK}$ Output
	External clock input	0								$\overline{SCK}$ Input
SIO Operation	Internal clock output	1	1	1	1	0	0	SO	SI	$\overline{SCK}$ Output
	External clock input	0								$\overline{SCK}$ Input
Internal clock only continuously output (no shift operation)		1	0	0	X	X	0	PB_0	PA_2	$\overline{SCK}$ Output
SIO function is not used. (Used as a port)		0	0	0	X	X	X	PB_0	PA_2	PA_3

X : Don't care

* : This state is set up when power is turned on (V_{DD} = low to high) and when reoperated after the CKSTP instruction is executed and the operation clock is stopped.

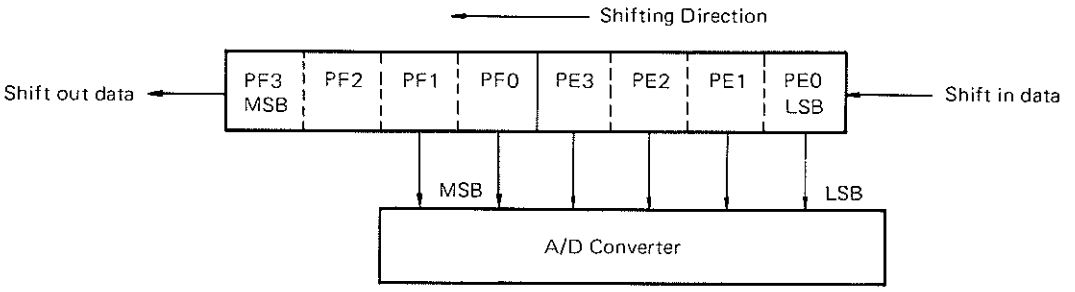
5.2 PSR (PRESETTABLE SHIFT REGISTER)

PSR consists of an 8-bit shift register; its four higher-order bits are assigned to Port F of the internal port and four lower-order bits to Port E. The data setting to or read from PSR is perfomred with the port operation instruction that accesses to Port E and F. Data are set by the OUT or SPB/RPB instruction, or read by the IN or TPT/TPF instruction.

As the shift clock that shifts PSR, either the internal or external clock can be chosen by SMR3; but the clock that is input to or output from the PA2/$\overline{SCK}$ pin can shift PSR only when "1" is set to SMR1. If it is set with "0", the input of shift clock into PSR is inhibited.

The PSR is also used as a data latch of the A/D converter. For this reason, **"0" should always be set in SMR1 to disable shift operation during A/D conversion.**

Shifting takes place from lower bits to higher bits, and shift-out data is output by the falling edge of the shift clock. PSR data is shifted by the rising edge of the shift clock. Shift-in data is read also by the rising edge of the shift clock.



Note: Data can be set in the PSR when SMR1 = 0 or SMR1 = 1 and the level of Pin PA3/$\overline{SCK}$ is high. In other cases, data cannot be set in the PSR correctly.

5.3 SCC (SHIFT CLOCK COUNTER)

The SCC is a binary counter consisting of four bits, and it counts the shift clocks to be input into PSR when "1" is set to SMR1 or when PSR can be shifted. Then it outputs "1" to the judge input of CPU when the counter counts up 8 (1000B) or it outputs "0" otherwise.

The judge output can be tested either by the TSET (Test Shift End, skip if True) instruction or the TSEF (Test Shift End, skip if False) instruction. The test turns true when the counter counts 8 or it otherwise turns false.

The content of SCC is cleared to "0" under any of the following conditions:

1) When the power is turned on (V_{DD}=low to high),
2) When the CKSTP instruction is executed and the internal clock stops,
3) When the SIO instruction is executed.

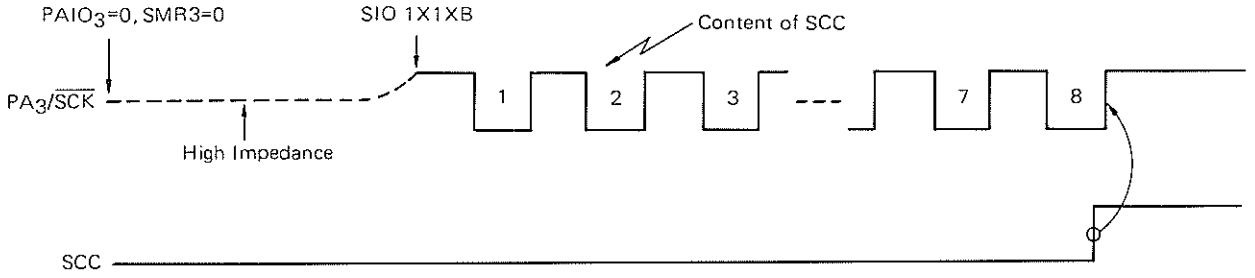
While the internal clocks are used as the shift clock, the internal clock is stopped by the afore-mentioned judge output so that only 8 bits of data can be transferred or received. Unless SCC is cleared, the execution of TSET/TSEF instruction results in the True condition. The shift clock input is not inhibited, if the external clocks are used, even if 8 clocks are input. **It is therefore necessary to stop the external clocks when 8 clocks are input.**

The judge output turns to be True only when 8 x (2n+1) external clocks were input (n = a positive integer other than 0); otherwise it turns to be False. Hence it is when the content of Bit #3 of SCC changes from "0" to "1" that SCC outputs "1" to the judge input.

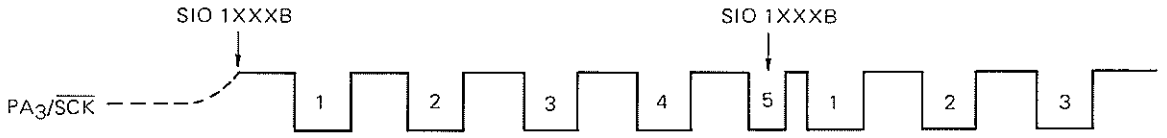
5.4 SCG (SHIFT CLOCK GENERATOR)

The SCG is a clock generator that produces the internal clocks of 15 kHz/50 % duty, which are output from the $PA_3/\overline{SCK}$ pin when "1" is set to SMR3. The clock is synchronized with the machine cycle and each clock is equivalent to the time consumed to execute two instructions (execute time: 33.3 μs).

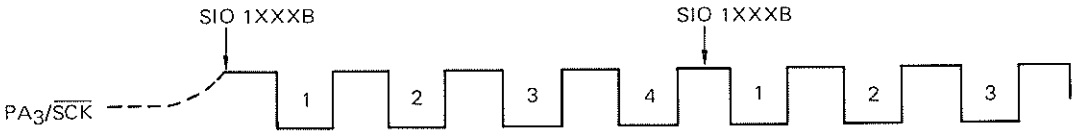
The SCG always output the clocks ranging from the high level to the active low level when "1" is set to SMR3, and it stops the clock generation with the high level output when the True judge output is made by SCC.



If "0" is set to SMR1 (to inhibit the shift) or when the SIO 1X0XB instruction is executed, the output of clocks continues by 8 clocks each without being automatically stopped. The clock output starts from the high level whenever the SIO instruction is executed. If the SIO 1XXXB instruction is executed again after the same instruction was once executed, the duty of clock deviates from 50 %, depending on the timing of execution.



To prevent such a deviation, the $PA_3/\overline{SCK}$ pin must be turned to PA_3 and tested (by executing the TPT/TPF instruction), and the SIO 1XXXB instruction must be executed by a instruction that falls an even numbered instruction after PA_3 is judged high. In other words it is enough to execute the SIO 1XXXB instruction once again when the clock becomes the high level. Simply speaking, the SIO instruction needs to be executed again at an even execution timing after the 1st execution of the SIO instruction.



The $PA_3/\overline{SCK}$ pin is held on a high impedance until after the SIO instruction is executed, for PA_3 is set under the input mode before executing the instruction. Therefore **it is required to connect the pull-up resistor to the $PA_3/\overline{SCK}$ pin when the serial I/O is used.**

5.5 EXTENDED APPLICATION OF SERIAL I/O

(1) Transfer of Data over 8 bits with Internal Clock

The continuous transfers of 8-bit or larger data with the internal clocks can be performed by setting the data of 4 bits each to PSR at a given timing during the shift and then executing the SIO instruction again. This action is detailed below with an example of 12-bit data transfer.

An 8-bit data is first set to PSR and the SIO 1X11B instruction is executed. The internal clock (PSR shift clock), which is output from the $PA_3/\overline{SCK}$ pin, is synchronous with the machine cycle, as mentioned before. The content of PSR is therefore shifted by four bits at the 7th instruction after the execution of SIO instruction. Four remaining bits are set at the 7th instruction to four lower-order bits of PSR, viz., Port E by the OUT instruction, and the SIO 1X11B instruction is executed at the 8th instruction. Thus another 8-bit data can be transferred without suspending the clock output, as the content of SCC is cleared by the execution of SIO instruction.

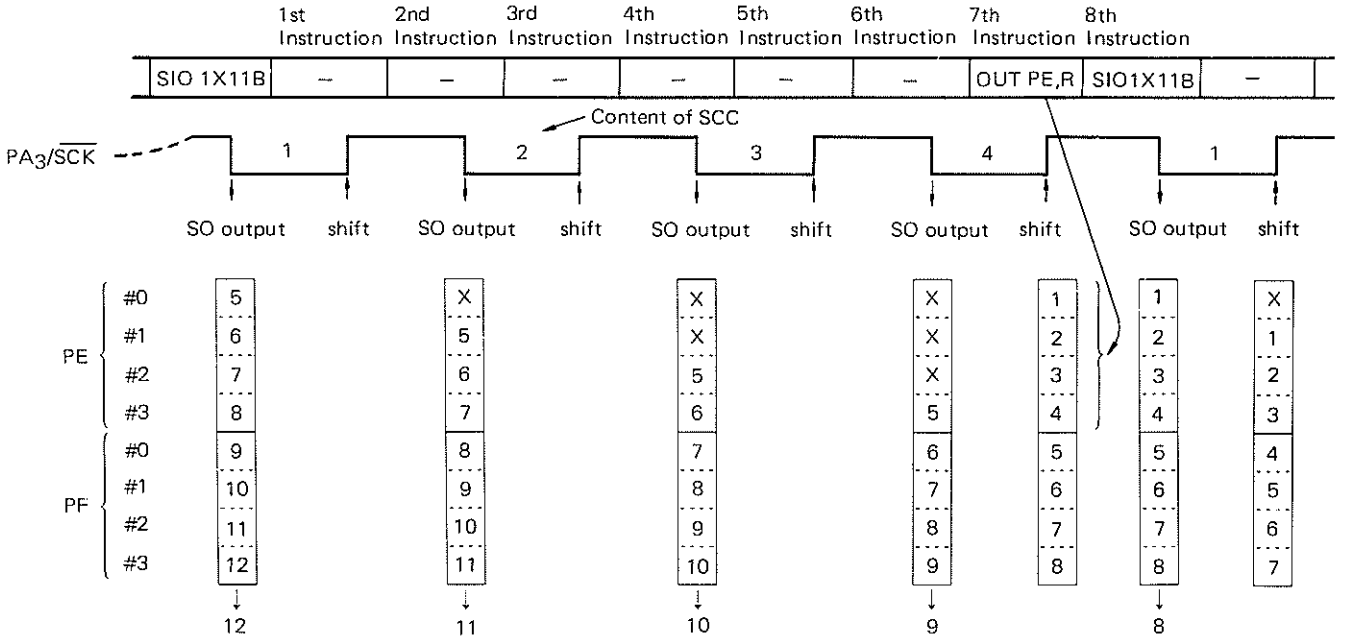


Fig. 5-4 Example of Sending 12-Bit Data

To similarly transfer the n-bits data ($8<n\leq12$), the OUT instruction must be executed at the (n−8)x2−1 instruction after the 1st SIO instruction execution, then the same instruction is to be executed again at the next instruction or (n−8)x2 instruction. The transfer of data of 12 or more bits can be likewise achieved by the repeated executions of both the OUT and SIO instructions.

(2) Reception of Data over 8 bits with Internal Clock

The afore-mentioned transfer process can be applied to the continuous reception of data more than 8 bits with the internal clock, provided that the execution timing of IN instruction differs from that of the OUT instruction. The following example is the reception of data more than 12 bits.

"0" is set in Bits #3 and #2 of the PAIO word before executing the first SIO 1X1XB instruction. Then the SIO instruction is again executed to clear the SCC content at the 8th instruction after the 1st execution or when a 4-bit serial data has been input to PSR. Then the IN instruction is executed against Port E (four-lower-order bits of PSR) at the next instruction or 9th instruction after the SIO instruction, thereby storing the 4-bit data into RAM. When the contents of both Port E and Port F are then stored when the shift is stopped, the reception of 12-bit serial data can be completed.

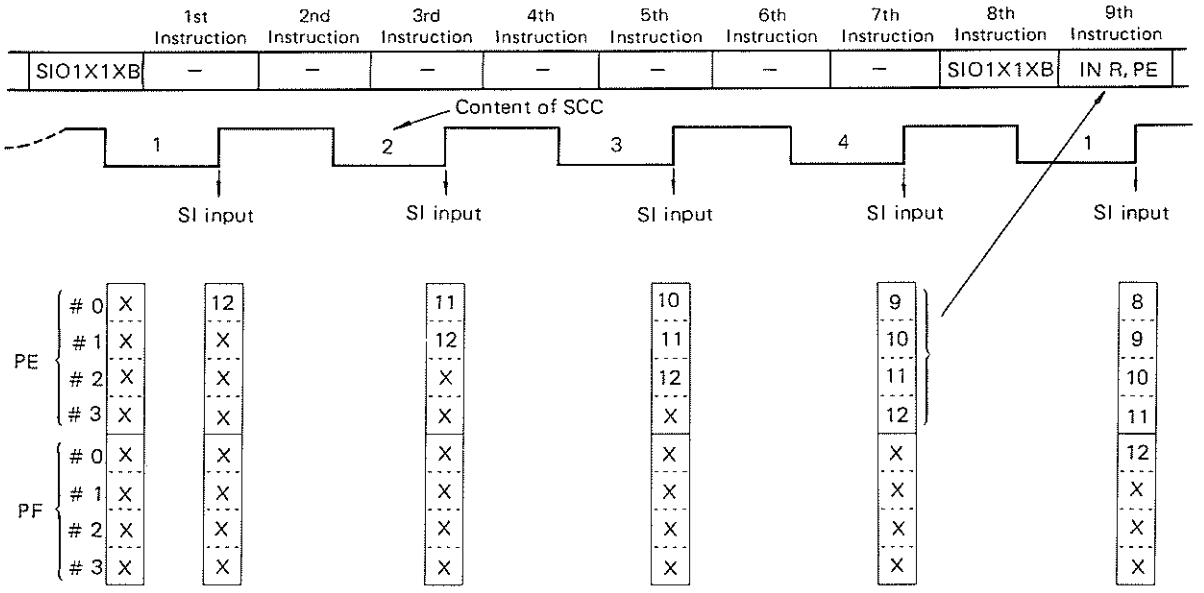


Fig. 5-5 Reception of 12-bit Data

To receive the n-bits data ($8<n\leq12$), the SIO instruction is generally executed at the (n–8)x2 instruction after the 1st SIO instruction execution, and the IN instruction is executed against Port E at the next instruction or (n–8)x2+1. The data more than 12 bits can be similarly received by the repeated executions of both the SIO and IN instructions.

Note: As apparent from the above two examples, it is not allowed to transfer and receive data more than 8 bits simultaneously. While the OUT instruction must be executed in the process of transfer at the (n–8)x2–1 instruction, the IN instruction must be executed in the process of data reception at the (n–8)x2+1 instruction. If data are simultaneously transferred and received, therefore, the 4-bit data initially stored is erased by the execution of OUT instruction. Then the data to be read out by the IN instruction is the data that is set by the OUT instruction rather than the shifted-in data.

6. A/D CONVERTER

The μPD17P23 has an on-chip 6-bit A/D converter of a sequential comparison system by programs.

When it is used for an FM/AM tuner, it can be used as an input pin of signal meter output by the IF filtering stage to measure the field intensity, etc. or to decide the stop level during automatic tuning. The AD pins can also be used as one-bit input ports that can freely set threshold levels by programs.

6.1 OPERATING PRINCIPLES

The μPD17P23 A/D converter consists of a 6-bit D/A converter of a resistance-strings system and a comparator.

Data is set in the D/A converter through the lower two bits (PF_1 and PF_0) of Port F and Port E (PE_3 to PE_0) which are internal ports. Comparison data is set by executing the output instruction (OUT, SPB, or RPB instruction) to Ports E and F, and comparison data is read by executing the input instruction (IN, TPT, or TPF instruction).

The contents of the higher two bits (PF_3 and PF_2) of Port F are neglected.

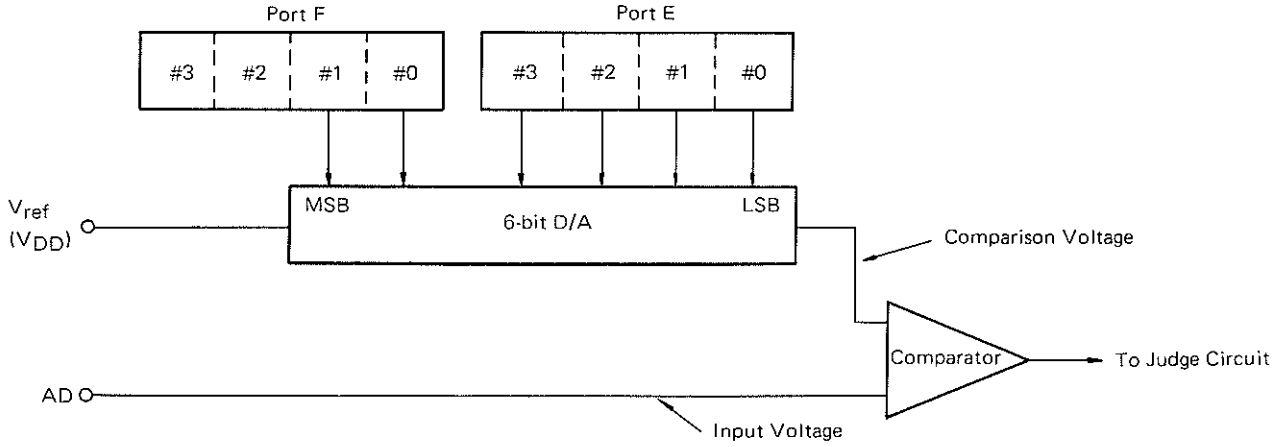


Fig. 6-1 A/D Converter Configuration

Ports E and F are also used as data latches during serial I/O, **and SMR1 should always be set to "0" (shift operation disable mode) during A/D conversion.**

The D/A converter generates 64 different voltages (voltage values obtained by dividing the reference voltage V_{ref} (V_{DD})) as comparison voltages based on data values set in Ports E and F. These comparison voltages are input to the comparator together with the analog voltage (input voltage) to be input to Pin A/D for comparison of both. The comparator comparison results can be obtained by executing the TADT (Test A/D comparator, then skip if True) instruction or the TADF (Test A/D comparator, then skip if False) instruction. The relationship between the input and comparison voltages at that time is as follows:

When input voltage > comparison voltage, True

When input voltage ≦ comparison voltage, False

6.2 CONSTRUCTION OF D/A CONVERTER

The D/A converter used in the μPD17P23 serially connects 64 stages of resistors between V_{ref} (V_{DD}) and GND and selects the voltage at each connecting point. It is a D/A converter of the so-called resistance-string system.

It should be noted that the values of serially-connected resistors differ from one to another and that the resistor closest to GND side has the resistance of 0.5/64 of the entire serial resistance value while the resistance of the resistor closest to the V_{ref} side is 1.5 /64. Fig. 6-2 shows the construction of the D/A converter.

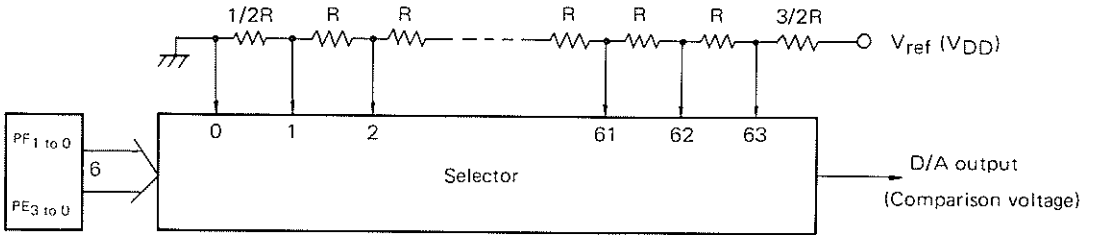


Fig. 6-2 Construction of D/A Converter

The D/A converter of this construction outputs the GND level when the value of 00H is set in the input data (Ports E and F) or outputs the voltage of $0.5/64 \times V_{ref}$ as a comparison voltage when the value of 01H is set. Similarly, the comparison voltage V_{out} when the value of n (decimal number) is set can be expressed by the following equation:

$$V_{out} = V_{ref} \times \frac{n - 0.5}{64} \quad (\text{provided : } 63 \geqq n \geqq 1)$$

If the following relationship establishes between the input voltage V_{in} and comparison voltage V_n to data n set to D/A in an A/D comparator of the sequential comparison system:

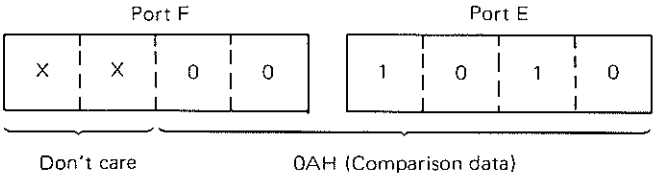
$$V_n < V_{in} \leqq V_{n+1}$$

n is generally output as A/D conversion data.

Table 6-1 Comparison Data-Comparison Voltage

COMPARISON DATA (Port F & E)		COMPARISON VOLTAGE		COMPARISON DATA (Port F & E)		COMPARISON VOLTAGE	
DEC.	HEX (Note)	X V_{ref}	V_{ref} = 5 V	DEC.	HEX (Note)	X V_{ref}	V_{ref} = 5 V
0	0 0 H	0	0 V	32	2 0 H	31.5/64	2.461 V
1	0 1 H	0.5/64	0.039	33	2 1 H	32.5/64	2.539
2	0 2 H	1.5/64	0.117	34	2 2 H	33.5/64	2.617
3	0 3 H	2.5/64	0.195	35	2 3 H	34.5/64	2.695
4	0 4 H	3.5/64	0.273	36	2 4 H	35.5/64	2.773
5	0 5 H	4.5/64	0.352	37	2 5 H	36.5/64	2.852
6	0 6 H	5.5/64	0.430	38	2 6 H	37.5/64	2.930
7	0 7 H	6.5/64	0.508	39	2 7 H	38.5/64	3.008
8	0 8 H	7.5/64	0.586	40	2 8 H	39.5/64	3.086
9	0 9 H	8.5/64	0.664	41	2 9 H	40.5/64	3.164
10	0 AH	9.5/64	0.742	42	2 AH	41.5/64	3.242
11	0 BH	10.5/64	0.820	43	2 BH	42.5/64	3.320
12	0 CH	11.5/64	0.898	44	2 CH	43.5/64	3.398
13	0 DH	12.5/64	0.977	45	2 DH	44.5/64	3.477
14	0 EH	13.5/64	1.055	46	2 EH	45.5/64	3.555
15	0 FH	14.5/64	1.133	47	2 FH	46.5/64	3.633
16	1 0 H	15.5/64	1.211	48	3 0 H	47.5/64	3.711
17	1 1 H	16.5/64	1.289	49	3 1 H	48.5/64	3.789
18	1 2 H	17.5/64	1.367	50	3 2 H	49.5/64	3.867
19	1 3 H	18.5/64	1.445	51	3 3 H	50.5/64	3.945
20	1 4 H	19.5/64	1.523	52	3 4 H	51.5/64	4.023
21	1 5 H	20.5/64	1.602	53	3 5 H	52.5/64	4.102
22	1 6 H	21.5/64	1.680	54	3 6 H	53.5/64	4.180
23	1 7 H	22.5/64	1.758	55	3 7 H	54.5/64	4.258
24	1 8 H	23.5/64	1.836	56	3 8 H	55.5/64	4.336
25	1 9 H	24.5/64	1.914	57	3 9 H	56.5/64	4.414
26	1 AH	25.5/64	1.992	58	3 AH	57.5/64	4.492
27	1 BH	26.5/64	2.070	59	3 BH	58.5/64	4.570
28	1 CH	27.5/64	2.148	60	3 CH	59.5/64	4.648
29	1 DH	28.5/64	2.227	61	3 DH	60.5/64	4.727
30	1 EH	29.5/64	2.305	62	3 EH	61.5/64	4.805
31	1 FH	30.5/64	2.383	63	3 FH	62.5/64	4.883

Note : The HEX codes shown above denote data of 6 bits in total — lower two bits of Port F and four bits of Port E. The contents of the higher two bits of Port F are neglected.



Assuming now V_{ref}=5.0 V and $V_{ref}/64$ or 0.0781 V is input into V_{in}; then the output voltages (comparison voltages) of D/A converter, V_0 , V_1 and V_2 when n=0, 1 and 2 are:

V_0 = 0 V
V_1 = 0.03906 V
V_2 = 0.11719 V

so that the comparison data of A/D converter should be n=1. If, therefore, the analog voltage V_{in} to be input is in the relation of 0.03906 V $< V_{in} \leqq$ 0.11719 V, the voltage of 0.0781 V is all input, or 0 V is considered to have been input if $V_0 < V_{in} \leqq V_1$. In this term the input voltage of $V_{in} = V_{ref}/64$ (1 LSB) is called the least decomposition voltage.

As apparent from the above, the input of 0.0781 V is considered to be made if the input voltage is held in the relation of 0.03906 V $< V_{in} \leqq$ 0.11719 V. There would be the least read error of ±0.03906 V against the actual input voltage. This kind of error is known the quantized error of A/D converter, which is expressed by ±1/2 LSB or by ±1/(64x2) = ± 0.781 % as it is a 4-bit A/D converter. The largest read error is ±1.5 LSB.

The measurable range of input voltage V_{in} of the A/D converter is:

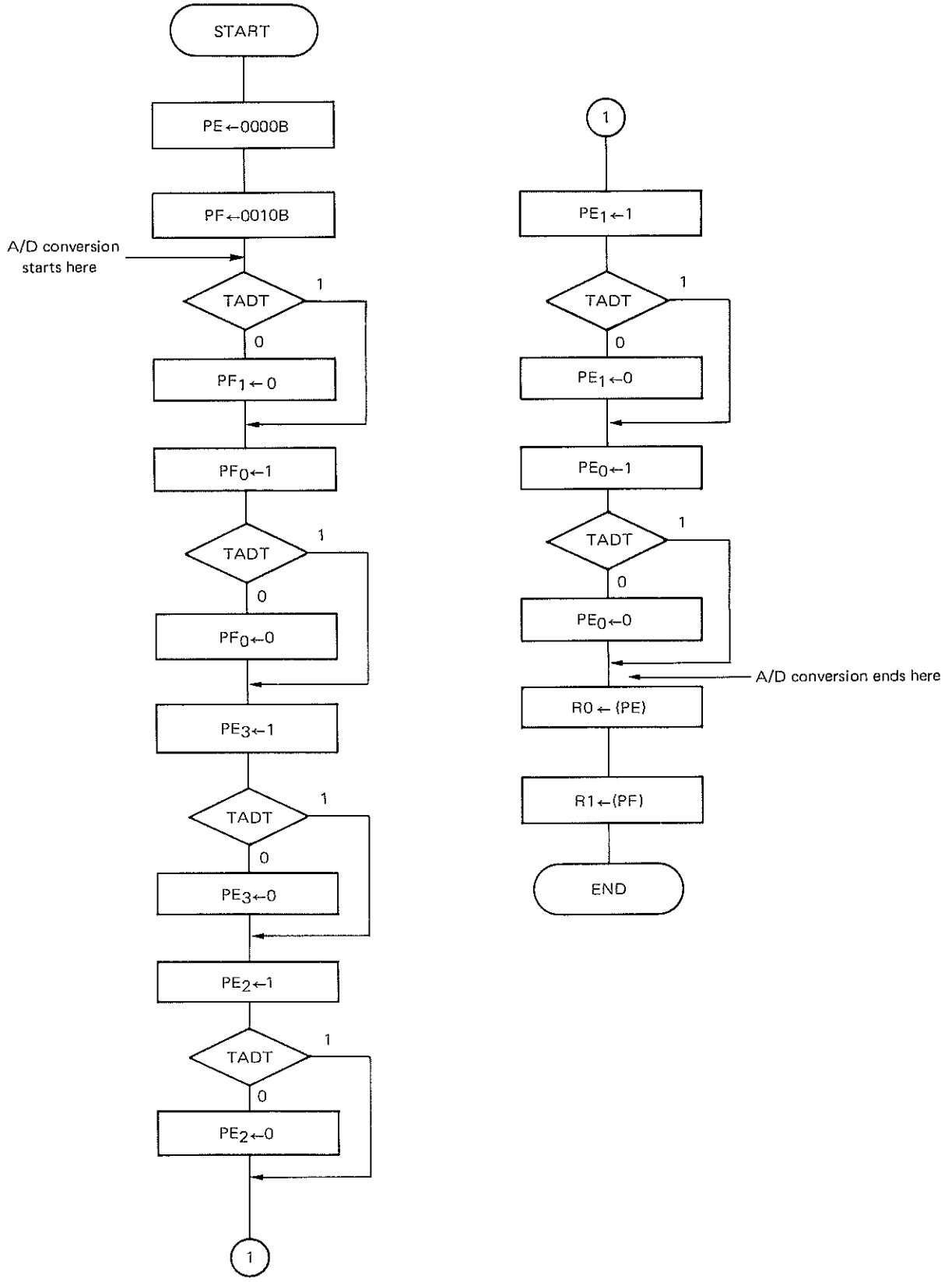
$$0\ V \leqq V_{in} \leqq V_{63} (V_{ref} \times 62.5/64)$$

If V_{in} is within the range of $V_{63} < V_{in} \leqq V_{ref}$ (V_{DD}), it is considered as an over-range.

The TADT or TADF instruction must be executed for A/D conversion, as mentioned above. When the comparison voltage of V_{13} is output against the input of analog voltage of $V_{in} = V_{14}$ (V_{ref} x 13.5/64), the input voltage is larger than the comparison voltage; hence the data True is output to the Judge circuit. Or the data False is output if V_{14} is output as the input voltage is equal to or smaller than the comparison voltage.

6.3 EXAMPLE OF A/D CONVERSION PROGRAM

Binary Search Method



Example of Coding

```
START:
        MVI     R0,0000B    ; PE data setting
        MVI     R1,0010B    ; PF data setting
        BANK1
        OUT     1,R1        ; PF←02H
        OUT     0,R0        ; PE←00H
        TADT
        RPB     1,0010B     ; D/A data MSB #5←1
        SPB     1,0001B     ; D/A data #4←1
        TADT
        RPB     1,0001B     ; D/D data #4←0
        SPB     0,1000B     ; D/A data #3←1
        TADT
        RPB     0,1000B     ; D/D data #3←0
        SPB     0,0100B     ; D/A data #2←1
        TADT
        RPB     0,0100B     ; D/A data #2←0
        SPB     0,0010B     ; D/A data #1←1
        TADT
        RPB     0,0010B     ; D/A data #1←0
        SPB     0,0001B     ; D/A data #0←1
        TADT
        RPB     0,0001B     ; D/A data #0←0
        IN      R0,0        ; R0 (Port E)
        IN      R1,1        ; R1 (Port F)
        BANK0
END:
```

A/D conversion time : 832.5 μs
Number of total steps : 25 steps

7. IF COUNTER

The μPD17P23 has an on-chip IF counter function to measure FM and AM IF frequencies. The IF counter consists of 16 bits and is mainly used to detect the stop signal during auto search tuning. If the desired IF frequency is counted by measuring frequencies input to Pins PA_1/FMIF and PA_0/AMIF during auto search tuning, a broadcast station can be considered to exist on the reception frequency at that time. Thus, by using the IF counter function to detect the stop signal, auto search tuning operation can be accomplished with smaller channel spacings such as 25kHz/step in the FM band and 1kHz/step in the AM band.

7.1 CONFIGURATION OF IF COUNTER

Fig. 7-1 shows the internal configuration of the IF counter.

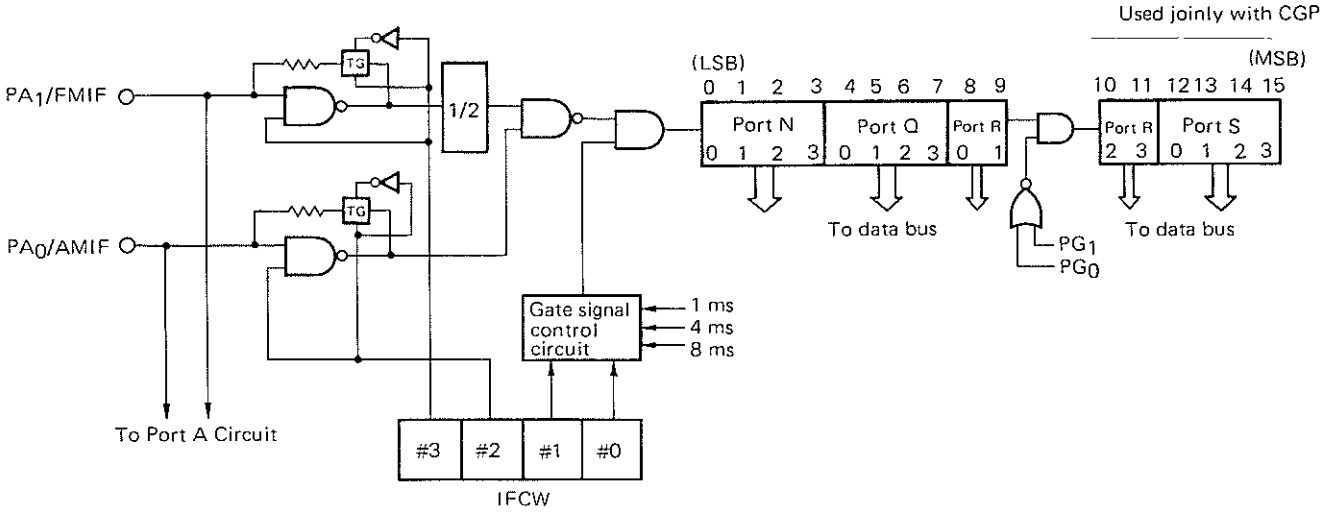


Fig. 7-1 Configuration of IF Counter

The IF counter of the μPD17P23 consists of binary counters for 16 bits, and the data contents of it can be read through Ports N, Q, R, and S of BANK3, which is an internal port. The LSB (Bit #0) of the IF counter corresponds to Bit #0 (PN 0) of Port N and the MSB (Bit # 15), to Bit # 3 (PS 3) of Port S, at this time.

Ports N, Q, R, and S are internal ports for reading only, and data cannot be set in the IF counter through these ports.

The higher 6 bits of the IF counter are also used by the counter that comprises the CGP. For this reason, the IF counter operates as a 10-bit counter in the modes other than the PG #2 through mode of CGP. Therefore, be sure to set up the CGP in the PG #2 through mode when operating the IF counter as a 16-bit counter. (The CGP control bits are not fixed during power ON resetting.) The contents of the ports for the higher six bits (PS_3 to PS_0, PR_3, and PR_2) are not fixed when the IF counter is used as a counter for ten bits. The values of these six bits are not affected at all even when they are counted at this time.

One of the four following count times (gate signals) of the IF counter can be selected by the IF counter control word (IFCW): 1 ms, 4 ms, 8 ms, and open (∞). The frequency input to Pin PA_1/FMIF or Pin PA_0/AMIF can be measured by deciding the number of pulses input to the IF counter within the above-mentioned times.

The IFCW can also select Pins PA_1/FMIF or PA_0/AMIF. If both Pins PA_1/FMIF and PA_0/AMIF are selected simultaneously at this time, the OR signal of the signal input to Pins PA_1/FMIF and PA_0/AMIF is input to the IF counter. If either pin only is selected, the unselected pin is internally pulled down automatically through a resistor.

The maximum frequency that can be input to Pin PA_1/FMIF is 20 MHz (V_{in} = 0.1 V_{p-p}) and that of Pin PA_0/AMIF is 5 MHz (V_{in} = 0.1 V_{p-p}). The signal input to Pin PA_0/AMIF is input directly to the IF counter. **The signal input to Pin PA_1/FMIF is input to the IF counter internally through the 1/2 frequency divider.** Therefore, the value of the IF counter will be 1/2 to the actual frequency to be input to Pin PA_1/FMIF if Pin PA_1/FMIF is selected.

The IF counter is reset during power ON resetting (V_{DD} = low to high) or when the clock is stopped. In this case also, the higher six bits (PS_3 to PS_0, PR_3, and PR_2) are reset only when the CGP control bit is in the PG #2 through mode. The CGP control bit during power ON resetting is not fixed.

When the IF counter enters a halt state, it maintains the state before the halt state. If the gate is open immediately before halting, the gate is closed after the designated time even during a halt. However, the gate is not opened during a halt if the gate is closed immediately before a halt.

7.2 IF COUNTER CONTROL WORD (IFCW)

The IF counter control word (IFCW) designates the input pin and input gate signal time of the IF counter. The IFCW consists of flip flops (F/Fs) for four bits and is set by the IFCW instruction. The match between IFCW instruction operand bits and IF counter control word and their functions are shown below:

All the bits of the IF counter control word are reset to "0" during power ON resetting (V_{DD} = low to high) or when the clock is stopped.

IFCW b3 b2 b1 b0 B

GTB (Gating Time Bits)		Gating Time
0	0	1 ms
0	1	4 ms
1	0	8 ms
1	1	OPEN (∞) (Gate is closed by program.)

ISB (Input Select Bits)		Input pin to be selected
0	0	Neither will be selected.
0	1	Pin PA_0/AMIF
1	0	Pin PA_1/FMIF
1	1	Signals of both PA_0/AMIF and PA_1/FMIF are ORed and are input.

7.3 GATE SIGNAL

The gate signal time of the IF counter is designated by the gating time bit (GTB) of the IF counter control word (IFCW). The basic clock of it is a 1-ms pulse signal which is not synchronous to the instruction. Gate signals of 1, 4, and 8 ms are generated based on this basic clock. For this reason, the IF counter does not start counting until the first basic clock after the execution of the IFC instruction is generated even if counting is commanded to be started by this instruction. However, even when counting is actually not performed, the gate can be judged to be open when the TGC instruction is executed immediately after counting is commanded to be started by the IFC instruction. When the TGC instruction is given, counting is actually performed immediately after counting is commanded to be started by the IFC instruction, and the gate is decided to be open until the gate actually closes. Fig. 7-2 shows an example when a gating time of 1 ms is designated.

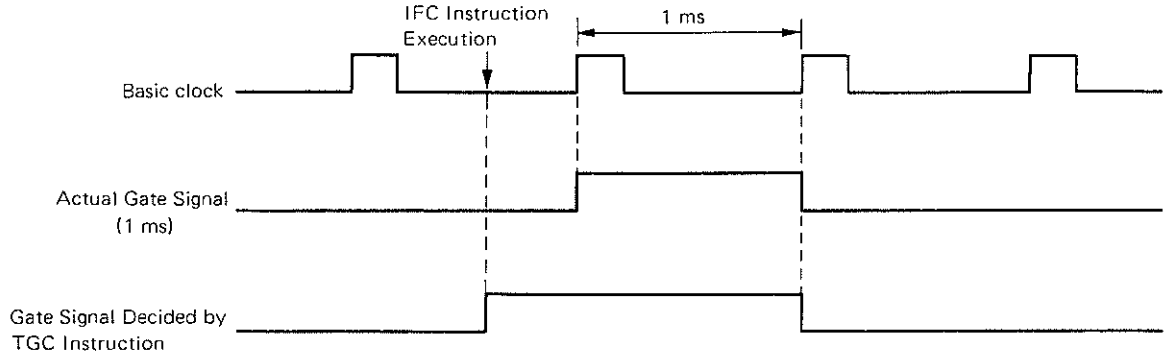


Fig. 7-2 Gating Time

As the timing chart in Fig. 7-2 shows, the time from the IFC instruction to the start of counting by the IF counter is maximum 1 ms depending on the IFC instruction execution timing when a gating time of 1, 4, or 8 ms is selected.

If "open" (GTB = 11B) is selected as a gating time, the IF counter starts counting even if it is not commanded to start counting by the IFC instruction. The IF counter starts counting immediately after the "IFCW XX11B" instruction (X is "0" or "1") is executed.

In this case, the gate can be closed by the following two methods. The one method is to set data other than "11B" in GTB. The other method is to set "00B" in ISB (input select bits) of the IFCW. Either of these two methods closes the gate simultaneously with resetting of the IFCW by the IFCW instruction. These two methods operate in exactly the same way when Pin PA_0/AMIF is selected as the input pin. However, they operate differently as follows if Pin PA_1/FMIF is selected as input.

The 1/2 frequency divider connected to Pin PA_1/FMIF is not reset when the GTB is set to other than 11B. However, this frequency divider is reset when the ISB is set to 00B.

When the gate is closed by setting the GTB to other than 11B, a counter start should not be commanded by the IFC instruction after GTB = 11B is designated and before changing the GTB to other value. If commanded, the gate is closed in an unfixed time after changing the GTB to a value other than 11B.

When the gate is closed by setting the ISB to 00B, the gate image in the TGC instruction remains open as long as the GTB remains 11B even if the gate closes. It is because merely input is not selected, instead of the gate actually closed in this method.

The IF counter is counted by the rising edge of the signal input to Pin PA_0/AMIF and by the falling edge of the signal input to Pin PA_1/FMIF.

7.4 ERROR

IF counter errors are gating time errors and counting errors. The gating time error depends on the oscillation frequency of the 4.5 MHz crystal resonator connected externally. This is because the basic pulse signal that decides the gating time is produced by dividing the 4.5 MHz frequency. Therefore, for example, if the oscillation frequency is shifted 20 ppm (90 Hz) to 4.5 MHz, the gating time also shifts 20 ppm (8×10^{-5} ns if 4 ms is selected).

Counting errors will be maximum $^{+1}_{-0}$. When Pin PA_1/FMIF is selected, Pin PA_1/FMIF is regarded equivalently as low if the gate closes with the input signal remaining in a "high" state (or if Pin PA_1/FMIF becomes non-selective), and is counted larger by 1. When Pin PA_0/AMIF is selected, the signal of low to high is equivalently input to the counter and is counted one more when the gate opens (or when Pin PA_0/AMIF is selected during counting) if the level of Pin PA_0/AMIF is already high.

7.5 EXAMPLE OF IF COUNTER DATA CALCULATIONS

(1) When Pin FMIF is selected as IF counter input pin

The frequency input to Pin PA_1/FMIF is input to the IF counter through the 1/2 frequency divider. For this reason, the data value of the IF counter will be 1/2 of the frequency input to Pin PA_1/FMIF.

Example: FM IF frequency (f_{FMIF}) : 10.7 MHz

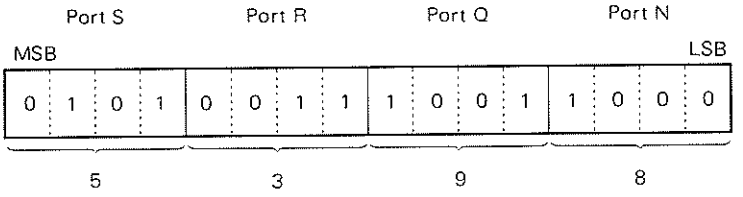
Gate signal (T_G): 4 ms

IF counter value (N)

$$N = \frac{f_{FMIF}}{2} \times T_G = \frac{10.7 \times 10^6}{2} \times 4 \times 10^{-3}$$

$$= 21400$$

$$= 5398H \text{ ("H" denotes a hexadecimal numeral)}$$



(2) When Pin AMIF is selected as IF counter input pin

The frequency input to Pin PA_0/AMIF is directly input to the IF counter.

Example: AM IF frequency (f_{AMIF}) : 450 kHz

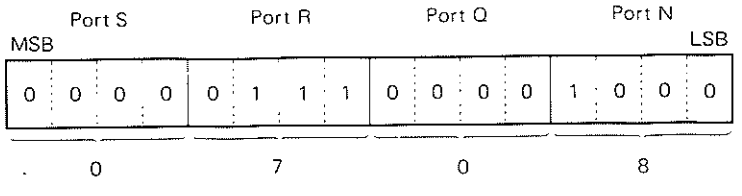
Gate signal (T_G): 4ms

IF counter value (N)

$$N = f_{AMIF} \times T_G = 450 \times 10^3 \times 4 \times 10^{-3}$$

$$= 1800$$

$$= 708H \text{ ("H" denotes hexadecimal numeral)}$$



8. LCD DRIVER

μPD17P23 contains in it an LCD driver (frame frequency: 100 Hz) of 1/2 duty and 1/2 bias drive (voltage equalization system type). Fig. 8-1 is a timing chart which illustrates the principle of the LCD driver. As can be seen from Fig. 8-1, two COM signals deviating from each other in phase by 1/4 output three potentials: 0 V (GND), 5 V (V_{DD}) and 2.5 V (1/2 V_{DD}) intermediate between them. In other words, the COM signals output a potential of ±1/2 V_{DD} on both side of 1/2 V_{DD}. The above display system is, therefore, referred to as 1/2 bias drive system.

In this system, two segments (A and B) are driven by an output of one segment, and a segment whose potential (V_{DD}) is most different from that of the com signal lights up. Four clock timings (a) to (d) are outputted as a segment output according to combinations of ON and OFF of two connected segments A and B. A segment to be lighted at this time repeats ON and OFF at 5 ms intervals. In other words, the segment is kept lighted on at a frequency of 100 Hz and duty factor of 1/2.

LCD_0/KS_0 to LCD_{15}/KS_{15} pins of μPD17P23 can be used as key source of key matrix also.

These pins are output the segment signal or the key source signal by controlling time.

The following shows normal drive waveform in Fig. 8-1, and waveform when using as key source signal in Fig. 8-2.

Also, LCD_{24}/PL_0 to LCD_{27}/PL_3 pins can be used as standard output port (Port L) when not used as segment pin. (See 3.4 PORT L) And, LCD display possible in executing HALT instruction.

8.1 DIGIT

LCD matrix is divided into fifteen groups called digit.

56-segment for display is divided into 15-digit (Dig-0 to Dig - E), it is controlled by LCDD instruction. (See Fig. 8-3)

When even digits (Dig-0, Dig-2, Dig-4, Dig-6, Dig-8, Dig-A, Dig-C and Dig-E), data stored by LCDD instruction are unconditionally input to the segment PLA, and the output is displayed. (Segment PLA is user programmable.)

When odd digits (Dig-1, Dig-3, Dig-5, Dig-7, Dig-9, Dig-B and Dig-D), data stored in the data memory (RAM) is displayed directly.

Table 8-1 shows relations between bits of RAM data to be output in an odd digit at this time and segments arranged in odd digits.

And, there is a dot which belongs to two kinds of digits. For instance, the dot of intersecting point of COM_2 and LCD_{19} is belonged to the digit of Dig-1 and Dig-E. The content of display can be changed by controlling either digit.

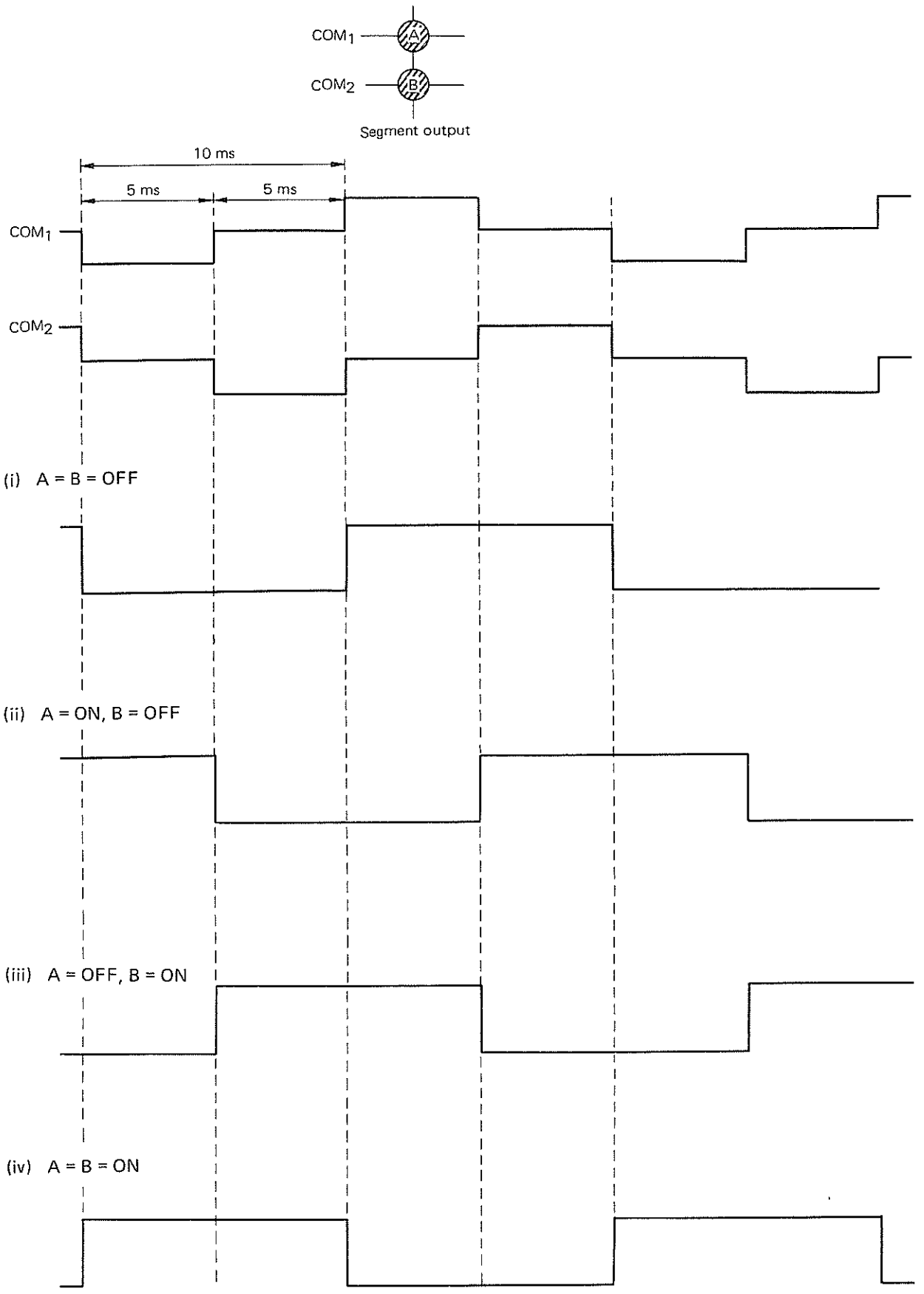


Fig. 8-1 LCD Drive Waveform ..... when the Key source is not jointly used. (KLE = 0)

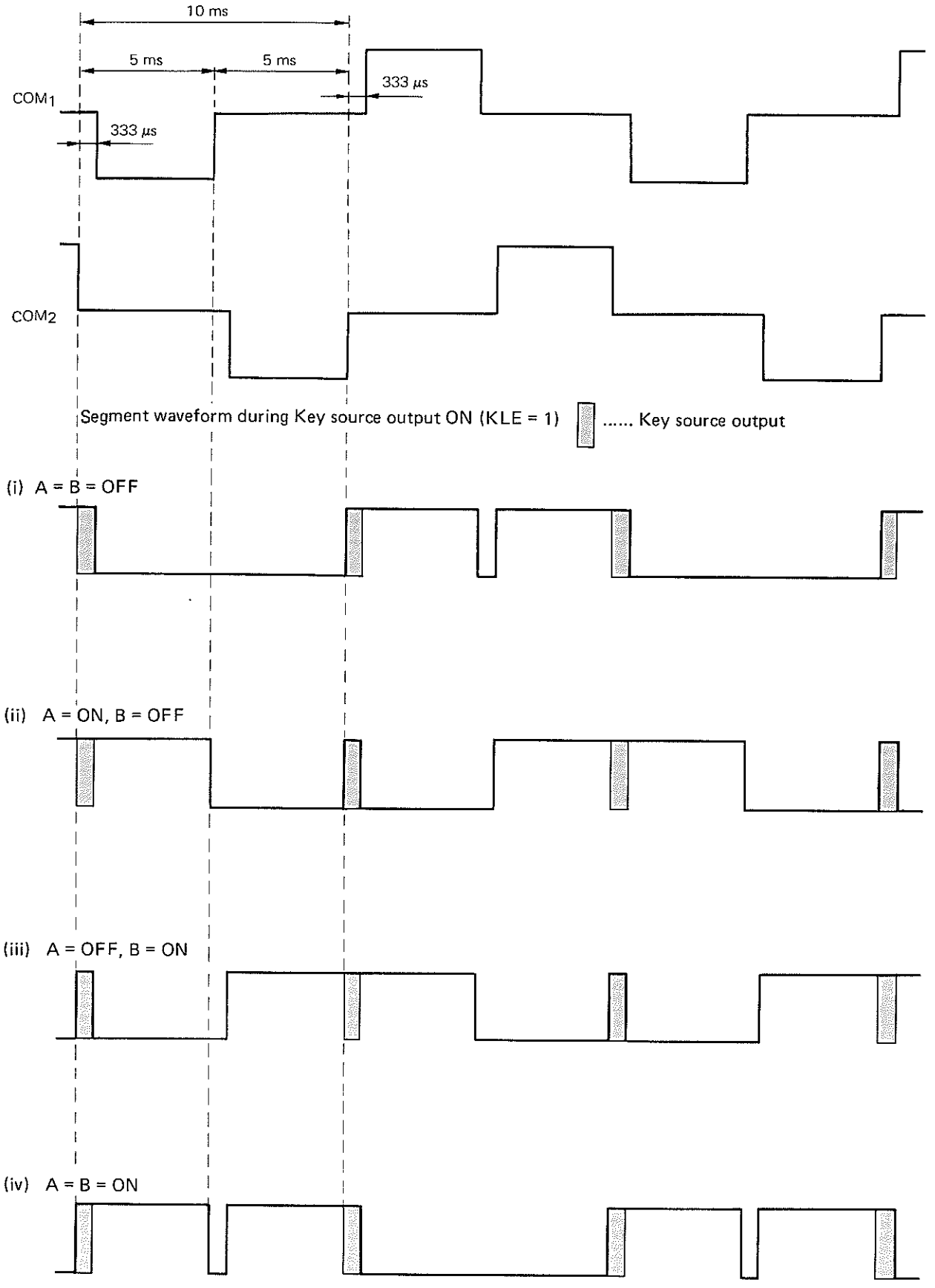


Fig. 8-2 LCD Drive Waveform .... when the Key source is jointly used (KLE = 1)

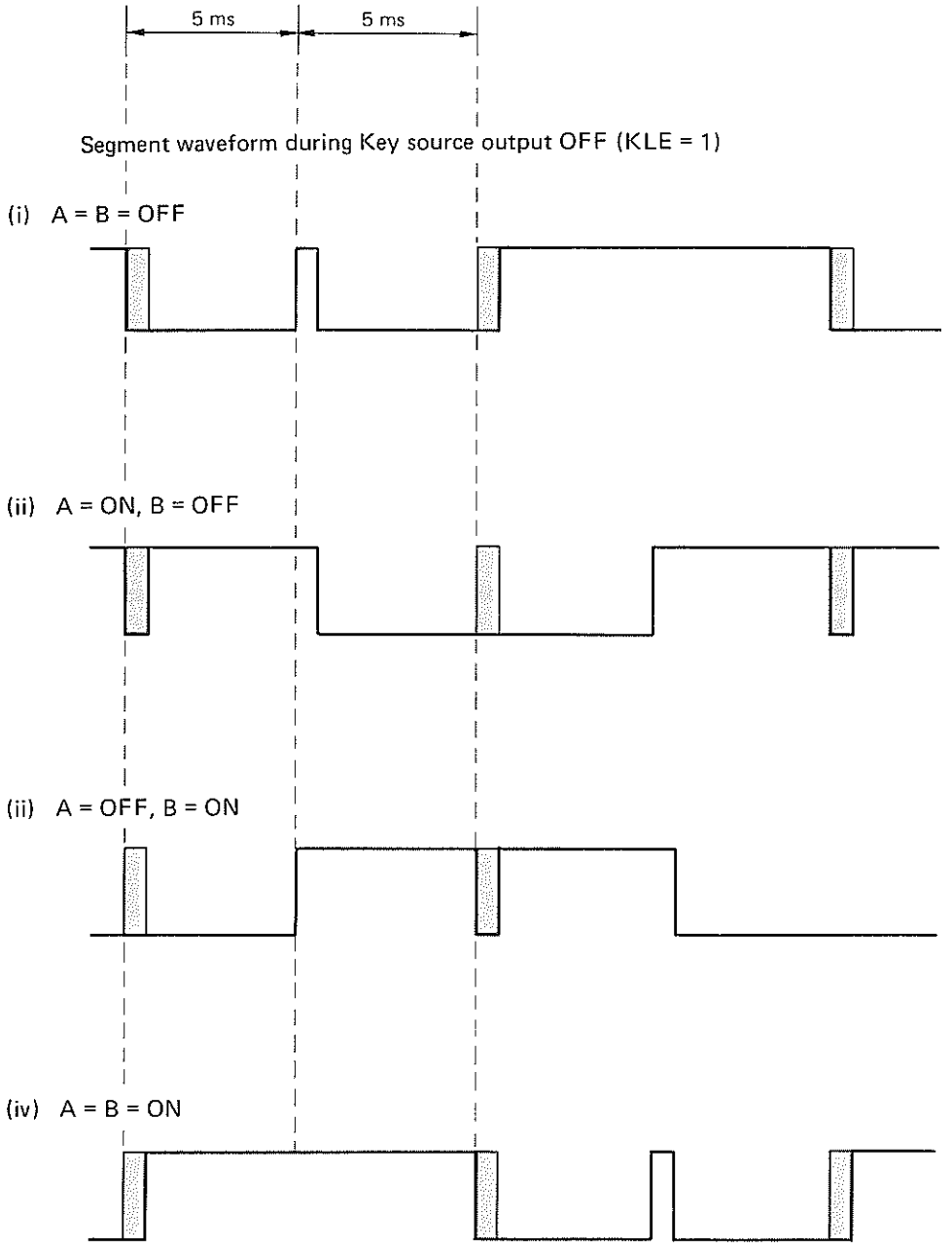
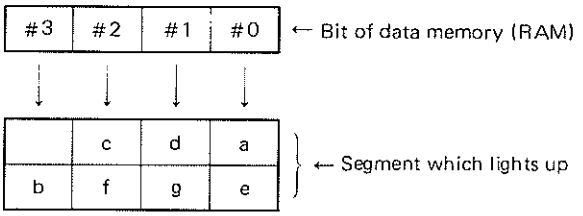
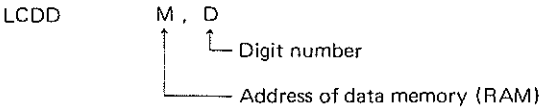

5 ms
5 ms
Segment waveform during Key source output OFF (KLE = 1)
(i) A = B = OFF
(ii) A = ON, B = OFF
(ii) A = OFF, B = ON
(iv) A = B = ON

Table 8-1 Bits Corresponding to Odd Digit (Which Does not Pass Through PLA)



μPD17P23 is able to make the LCD panel display data by executing an LCDD instruction. This instruction is described as the follows.

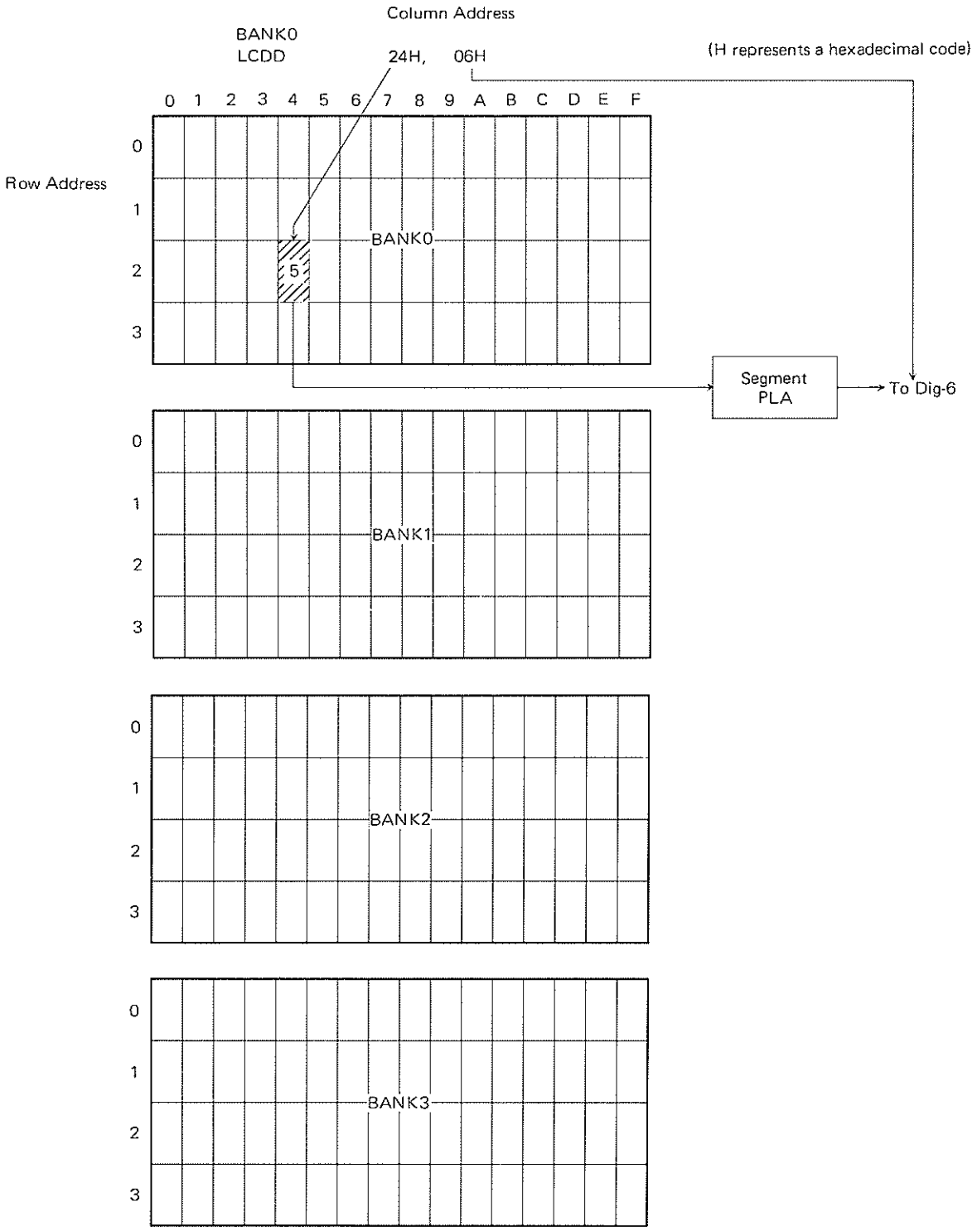


"M" represents any address in a data memory (RAM), and "D" represents the number of digits of display.

"D" takes fifteen (15) values 00H to 0EH of Digit-0 to Digit-E which are mentioned before and a special value 0FH called LCD control word.

Examples of operations when executing an LCDD instruction in even and odd digits are described in the following.

(1) In case of even digit (Dig-0, Dig-2, Dig-4, Dig-6, Dig-8, Dig-A, Dig-C and Dig-E)



The data "5" stored in the address 24H of the data memory (RAM) is led to the PLA, and data stored in the PLA is output to the digit of Dig-6.

The PLA pattern of 24H of RAM is selectable the pattern 0 or the pattern 1.

Therefore, in case "SPLSEL 3" is described in assembler, the pattern corresponding input data "5" of the pattern 0 is output. (The output data is 1101101B in examples of table 10-1) In case "SPLSEL 2" is selected, the pattern corresponding input data "5" of the pattern 1 is output. (The output data is 1100000B in examples of table 10-2.)

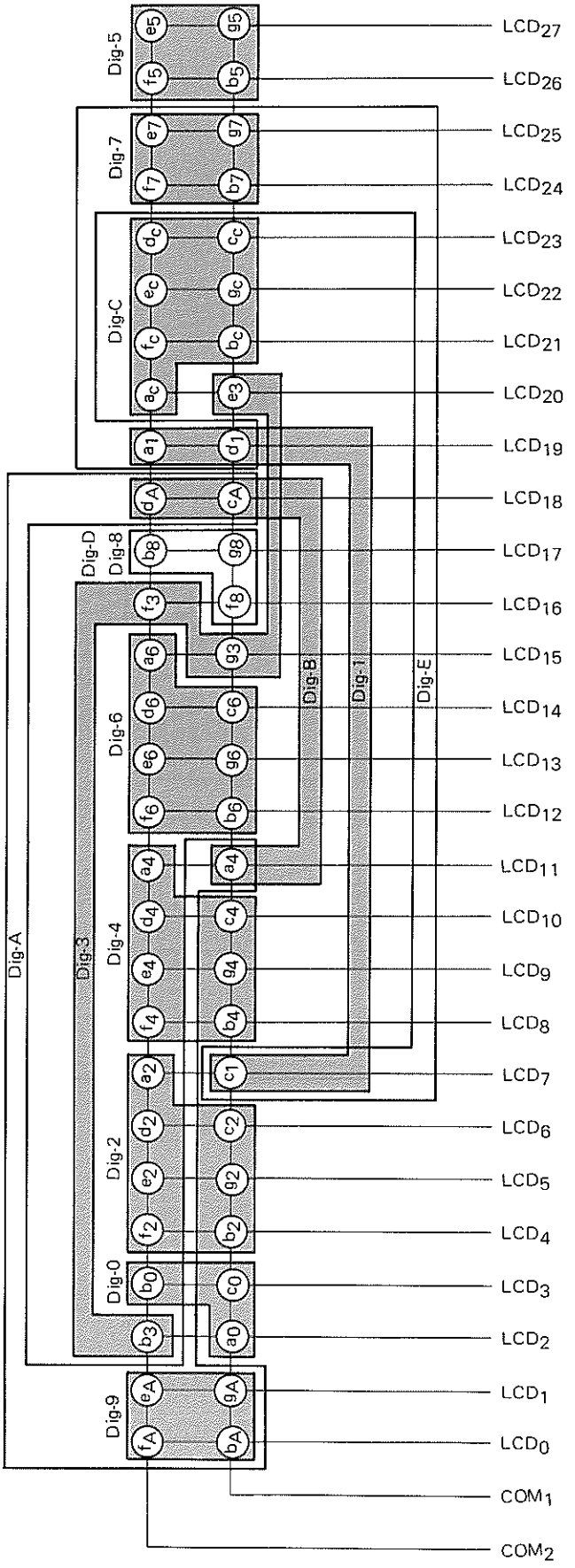
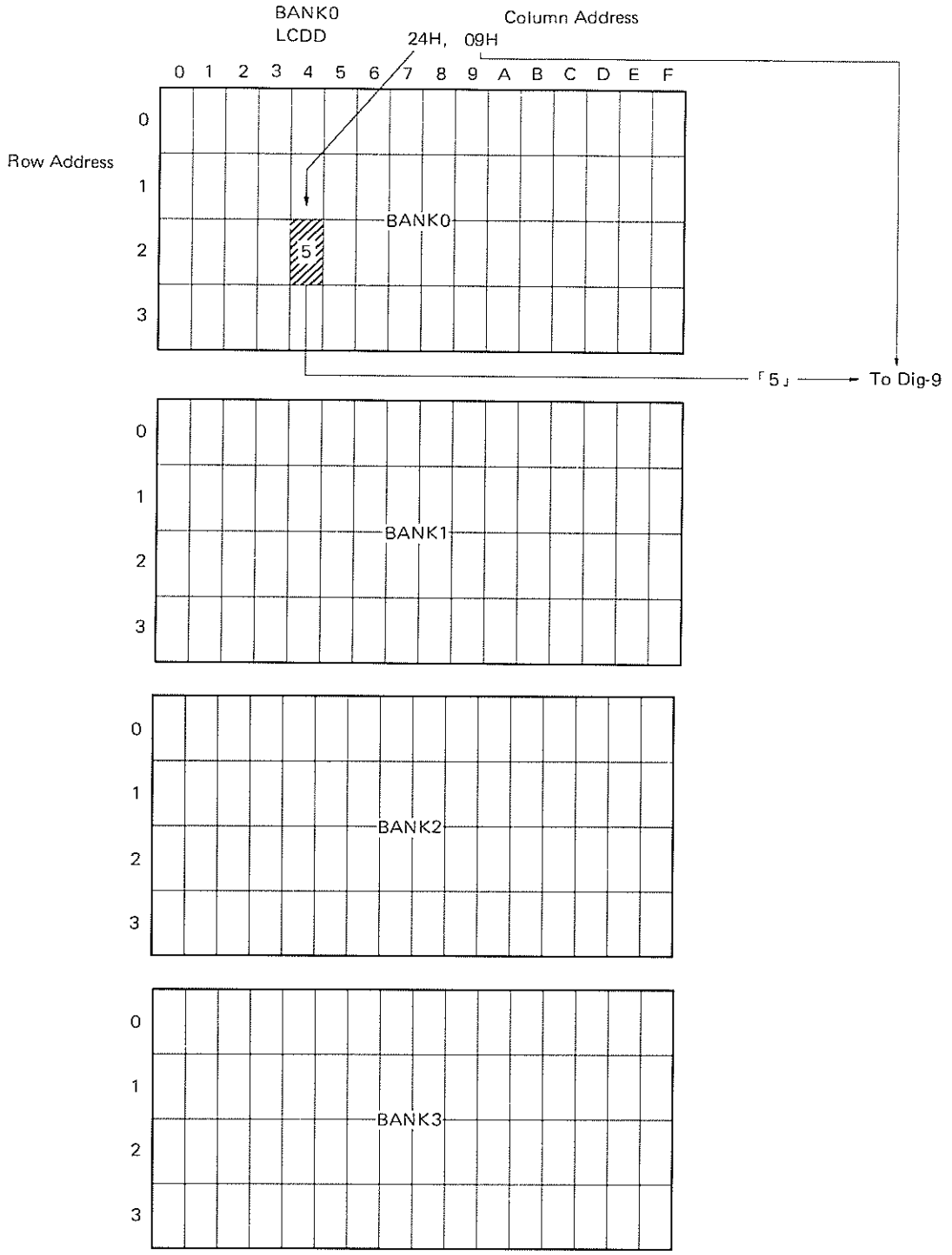


Fig. 8-3 LCD Matrix

(2) In case of odd digit (Dig-1, Dig-3, Dig-5, Dig-7, Dig-9, Dig-B and Dig-D)



The data "5" stored in the address 24H of the data memory is output directly to a digit of Dig-9.

Four segments b_A, f_A, g_A and e_A are arranged in the digit of Dig-9, and when "5" = 0101B is output, the segments b_A and g_A is put off, and the segments f_A and e_A lights up. (See Table 8-1 "Bits Corresponding to Odd Digit (Which Does not Pass Through PLA)).

When "4" = 0100B is output to the digit of Dig-9, only the segment f_A lights up. Also, four segments b_A, f_A, g_A, e_A are belonged to the digit of Dig-A, data of the digit of Dig-A is displayed through the PLA.

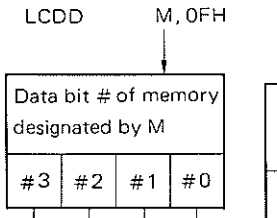
However, when the LCDD instruction is executed for Dig-A, not only four segments (b_A, f_A, g_A, e_A) are changed but also the segments d_A and c_A are changed.

8.2 LCD CONTROL WORD

The LCD control word is a 4-bit register which resides in digit 0FH of the LCD Controller/Driver. The contents of this register can be manipulated by LCDD M, 0FH instruction. M is an address that points to a RAM location containing the new data to be written. The contents of LCD control word will remain unchanged until the device is reset during power on clear or when clock stop instruction, CKSTP, is executed.

Upon the occurrence of either of above two events, the LCD control word will be cleared to 0.

Table 8-2 Configuration of LCD Control Word



NAME	FUNCTION	CONTENT DURING RESETTING
PLEL (Port L Enable Lower F/F)	To two pins (LCD_{25}/PL_1 and LCD_{24}/PL_0): 0: LCD segment signal is output 1: Lower two bits of Port L are output	0
PLEU (Port L Enable Upper F/F)	To two pins (LCD_{27}/PL_3 and LCD_{26}/PL_2): 0: LCD segment signal is output. 1: Higher two bits of Port L are output.	0
KLE (Key Latch Enable F/F)	Data input from Pins K_3 to K_0 is. 0: Directly read without being fed to the key latch. 1: Read through the key latch.	0
LCDE (LCD Enable F/F)	Setting of LCD display mode 0: LCD OFF mode (COM_1 pin, COM_2 pin and all the pins designed by LCD segment becomes low level) 1: LCD ON mode (normal LCD operation)	0

Note: "During resetting" mentioned above means when the system is reset or the clock stop instruction, CKSTP is executed.

(1) **PLEL** (Enable Port L lower nibble), **PLEU** (Enable Port L upper nibble)

A selection is possible whether to use two each of four pins (LCD_{27}/PL_3 to LCD_{24}/PL_0 pins) as port or as LCD segment output.

Bit 0 and Bit 1 of the LCD control word determines the function of LCD_{24}/PL_0 to LCD_{27}/PL_3. If the corresponding bit is set to 0 then the port will operate as LCD segment drivers. If set to 1 then it will operate as Port L. (See 3.4 PORT L)

(2) **KLE** (Enable key latch)

The source signal for key matrix can be supplied by the LCD segment signals, LCD_{15}/KS_{15} to LCD_0/KS_0. The 16 key source signals are multiplexed with the 16 LCD driver segment signals.

Since the key source signals are not synchronized with program execution, the valid key matrix data can not be read directly. In order to store the key input data while the key source signals are active a four bit key input latch is integrated.

Maximum of 16 x 4 key matrix can be realized by using the multiplexed key source signals and key input latch while driving the LCD segments. The input data to K_3 to K_0 are latched, in synchronous to the key source signal. The data in the latch can be read by executing KIN or KI.

The KLE bit, when set to 1 will enable the key input data to be latched and when set to 0 the key input data is not latched. Therefore if LCD_{15}/KS_{15} to LCD_0/KS_0 are used for key source signals, KLE must be set to 1 in order to read valid key data. If regular I/O port is used for key source signals, then KLE must be set to 0. (See Fig. 8-2 in 8. LCD DRIVER about output wave form. And see 9. THE WAY OF CONSISTING KEY MATRIX.)

(3) **LCDE** (Enable LCD)

The LCDE controls the operation of LCD driver. When LCDE is set to 0 the output latch is disabled and display is blanked (**display OFF mode**). If the LCDE is set to 1 the output latch is enabled (**display ON mode**).

See Fig. 8-4.

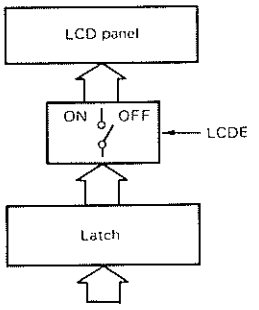


Fig. 8-4 Conception of LCD Driver

During the display OFF mode (LCDE = 0), all segment pins **specifying for LCD display** (LCD pins) and common outputs (COM_1, COM_2 pins) are set to low level (0 V). Pins specifying for LCD display means that pins are specified to use as LCD segment pin by PLEL and PLEU. If PL_3/LCD_{27} to PL_0/LCD_{24} operate as Port L, then LCDE will not affect the output status of PL_3 to PL_0.

During the display ON mode (LCDE=1), the data in segment latch is output in biplex form. The contents of LCD control word are cleared to 0 when the device is reset by power on clear or clock is stopped by CKSTP instruction. Therefore the display will be turned off if either of above 2 events take place. This prevents invalid data to be displayed after power on clear or when clock is stopped. Prior to setting the LCDE to 1, the segment output latch can be updated with valid data.

In addition, if LCDE=0, the key source signal will be disabled. Therefore if key source signal is necessary while the display is blanked, the segment output latch must all be cleared by LCDD instruction while keeping LCDE = 1.

9. KEY MATRIX CONFIGURATION

Two methods can be used to implement a key matrix with μPD17P23. The first method is to use regular I/O ports for key source signals and K_3 to K_0 as key return inputs. The second method is to use LCD_{15}/KS_{15} to LCD_0/KS_0 of the key source signals and K_3 to K_0 as key return inputs. The first method requires KLE = 0 and second method requires KLE and LCDE to be set to 1.

9.1 GENERAL PURPOSE I/O FOR KEY SOURCE SIGNAL

This method requires the setting of KLE = 0. The key input data can be read by executing KIN or KI instruction. The advantage of using the method over the second method is the short time required to read the key input data.

Since the KLE is set to 0 upon power on clear or execution of CKSTP, this will be the default mode.

9.2 MULTIPLEXED KEY SOURCE SIGNAL AND LCD SEGMENT OUTPUT

The use of LCD_{15}/KS_{15} to LCD_0/KS_0 as key source signal requires setting of KLE to 1. In addition the LCDE must be set to 1 so that LCD_{15}/KS_{15} to LCD_0/KS_0 will output multiplexed key source signal and LCD segment output. The input key data is latched when the key source signal is active. The key source signal is output every 10 msec for the duration of 333 μsec. The key source signal that is selected by the Port J, address 0 of BANK2, and Port K, address 1 of BANK2. The data in Port J and Port K are decoded by internal decoder and output to LCD_{15}/KS_{15} to LCD_0/KS_0. The set up time for key source is 333 μsec. Fig. 9-1 illustrates the internal key source hardware.

In order to insure proper LCD voltages at all time a diode must be placed in case multiple keys from different key source signals are pressed. By same reasoning, pull down resister must not be used.

The pull down resisters are internally provided for KLE = 0 and if KLE = 1 the pull down resisters are active only when key data are latched.

If transister switch is used, care must be taken so that I_{OH} does not degrade the V_{OH} to cause degration of LCD display.

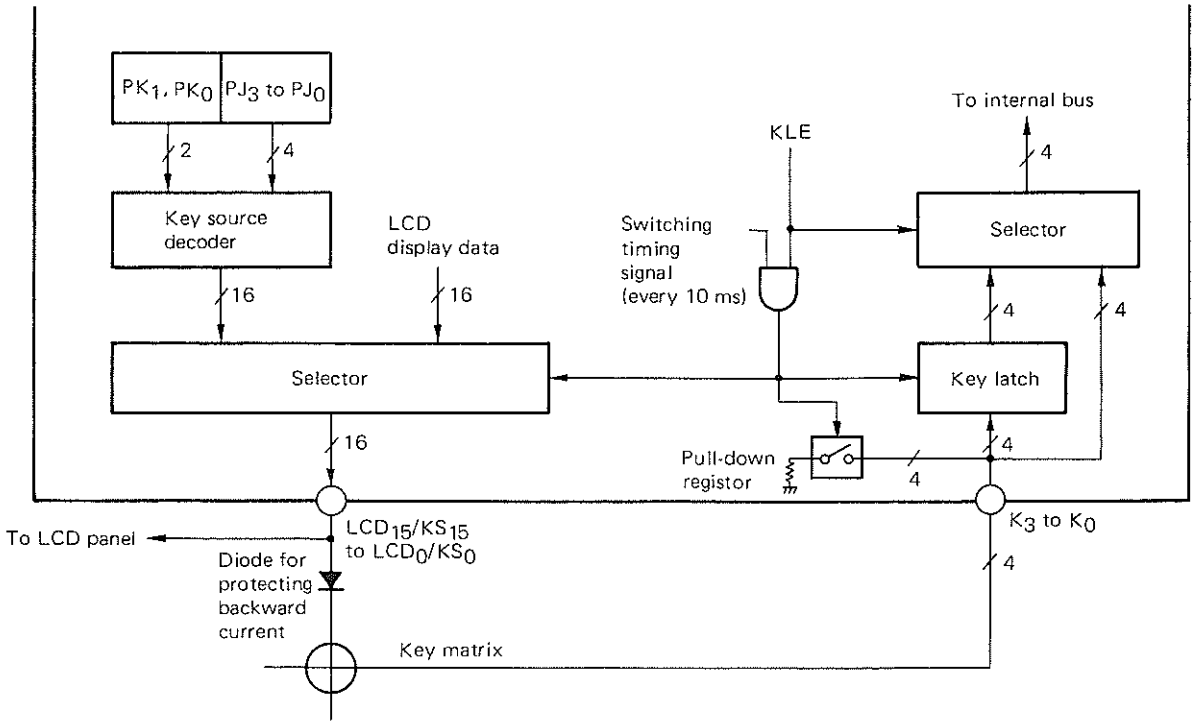


Fig. 9-1 Concept of key input operations using LCD pins for key source

9.3 KEY SOURCE DECODER

If LCD_{15}/KS_{15} to LCD_0/KS_0 are used for key source signals, KLE = 1 and LCDE = 1, then data is output every 10 msec for 333 μsec. The time that it will take to determine one of n source signal from the key matrix is 10 x n msec. Therefore if there was 8 key source signals it will take 80 msec.

In order to minimize this time a binary search using the key source decoder can be used.

The key source decoder is 6 bits wide for input and 16 bits wide for output, where upper 2 bits reside in lower 2 bits of Port K (PK_1 and PK_0) and lower 4 bits reside in Port J (PJ_3 to PJ_0). The key source decoder is illustrated in table 9-1.

Port K and Port J are internal ports and they differ from other ports only by the fact that there is no external outputs. The contents of Port K and Port J remain valid after the clock is stopped, executing CKSTP instruction. The data in Port K and Port J can be input and tested by IN, TPT, and TPF instructions. Since only the lower 2 bits of Port K are used, upper 2 bits, when read, will be 0.

The data specified by Port K determines the "PHASE" or the level of binary search and Port J determines the "KEY SOURCE BIT" or which source bits are active.

Table 9-1 Key Source Decode

DECODE INPUT		DECODE OUTPUT (LCD_{15}/KS_{15} to LCD_0/KS_0)															
PK	PJ	SEGMENT LINE (KEY SIGNAL SOURCE LINE)															
1 0	3 2 1 0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1 1	0 0 0 0									■	■	■	■	■	■	■	■
	1 0 0 0	■	■	■	■	■	■	■	■								
	| | | |																
1 0	0 0 0 0													■	■	■	■
	0 1 0 0									■	■	■	■				
	1 0 0 0					■	■	■	■								
	1 1 0 0	■	■	■	■												
0 1	0 0 0 0															■	■
	0 0 1 0													■	■		
	0 1 0 0											■	■				
	0 1 1 0									■	■						
	1 0 0 0							■	■								
	1 0 1 0					■	■										
	1 1 0 0			■	■												
	1 1 1 0	■	■														
0 0	0 0 0 0																■
	0 0 0 1															■	
	0 0 1 0														■		
	0 0 1 1													■			
	0 1 0 0												■				
	0 1 0 1											■					
	0 1 1 0										■						
	0 1 1 1									■							
	1 0 0 0								■								
	1 0 0 1							■									
	1 0 1 0						■										
	1 0 1 1					■											
	1 1 0 0				■												
	1 1 0 1			■													
	1 1 1 0		■														
	1 1 1 1	■															

Note 1: The shading indicates that the key source signal is output to the pertinent pin.

Note 2: If other than the data listed in this table is input to Port K or Port L, all pins can't be selected (key source signals are not output.)

9.4 KEY LATCH F/F

The key source signals output by multiplexed LCD_{15}/KS_{15} to LCD_0/KS_0 are asynchronous to the execution of instructions. In order to determine if the key source signals have been output and valid key data are latched, key latch F/F is provided.

When data is written in Port K or Port J or if TKLT or TKLF instructions are executed, the key latch F/F is reset. The latch F/F is set when key source signal is output for 333 μsec and key data are latched.

Key latch F/F status can be determined by executing either TKLT or TKLF instructions.

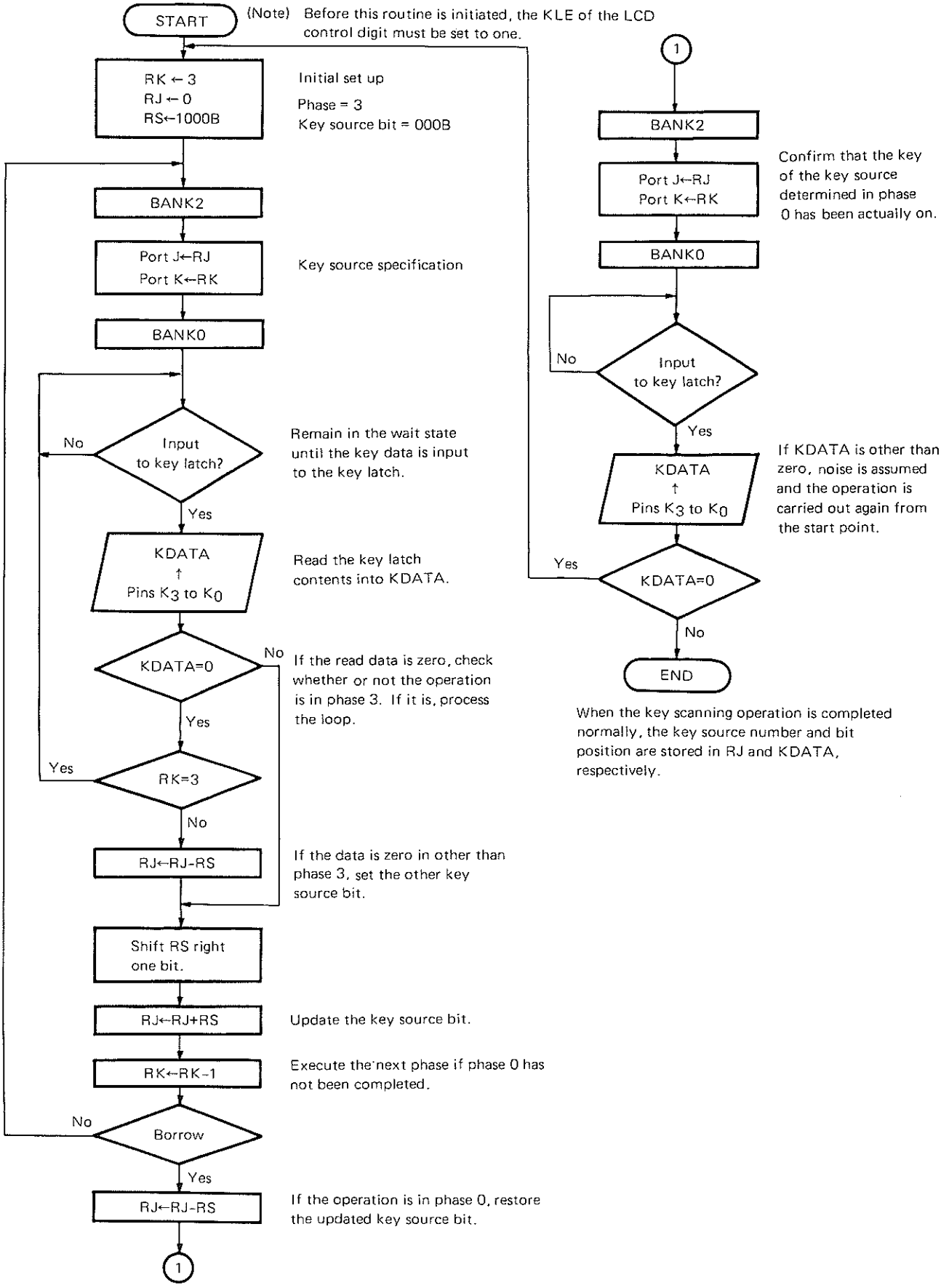
The key latch F/F is active only if KLE is set to 1 and it is inactive if KLE is set to 0. The key latch F/F is set to 0 during power on clear and when CKSTP is executed.

9.5 SAMPLE PROGRAM OF USING KEY SOURCE SIGNAL

A program which demonstrates the use of binary search when LCD_{15}/KS_{15} to LCD_8/KS_8 are used for key source signals.

This routine takes approximately 50 msec to determine which one of 32 keys were pressed. Since the key source signals are active every 10 msec, key chattering routine is not necessary if key chattering time of 10 msec or less is acceptable.

Key Search Program Example (Binary Search Method)



CODING EXAMPLE

```
KEYSCAN:
        MVI     RK, 3       ;  RK = PHASE data
        MVI     RJ, 0
        MVI     RS, 0000B   ;  RS = Key source bit
KSOUT1:
        BANK2
        OUT     PJ, RJ      ;  Key source specification
        OUT     PK, RK
        BANK0
WAIT:
        TKLT                ;  Remain in the wait state until the key data is input to the key latch.
        JMP     WAIT
        KIN     KDATA       ;  KDATA ← Key Latch Data
        JMP     RSHIFT      ;  If key input, to RSHIFT
        SNEI    RK, 3       ;  If PHASE is "3", continuing the loop until key input
        JMP     WAIT
        SU      RJ, RS      ;  Reset the key source bit
RSHIFT:
        MVI     RC, 3       ;  Shift RS left 4 bits
SFTLOOP:
        ADN     RS, RS
        ORI     RS, 1
        SIS     RC, 1
        JMP     SFTLOOP
        AD      RJ, RS      ;  OR bits shifted to key source bit
        SIS     RK, 1       ;  If phase 0 has not been completed, update phase and repeat from KSOUT1.
        JMP     KSOUT1
        SU      RJ, RS
        BANK2               ;  Confirm that the key of the key source determined in phase 0 has been
        RPB     PK, 0FH        actually on.
        OUT     PJ, RJ
        BANK0
WAIT1 :
        TKLT                ;  Remain in the wait state until the key data is input to the key latch
        JMP     WAIT1
        KIN     KDATA
        JMP     SCNEND      ;  If KDATA is not 0. Over key searching.
                               Segment number in which there is pushed key is input to KDATA.
        JMP     KEYSCAN     ;  If KDATA is ZERO, noise is assumed and the operation is carried out
                               again from the start point.
```

10. PLA (PROGRAMMABLE LOGIC ARRAY)

μPD17P23 contains in it a segment PLA based on a user's program. Display patterns of the LCD panel are usually programmed in the segment PLA, and a total of 32 types (16 types x 2) of patterns can be generated. The segment PLA is selected only when an even digit is designated by an instruction LCDD as mentioned in the above section (refer to Section 8 "LCD DRIVER").

10.1 COMPOSITION OF SEGMENT PLA

The segment PLA is composed of a 5-bit segment latch circuit and a PLA to which an output of the segment latch circuit is input, and which outputs seven bits corresponding to display patterns of seven segments.

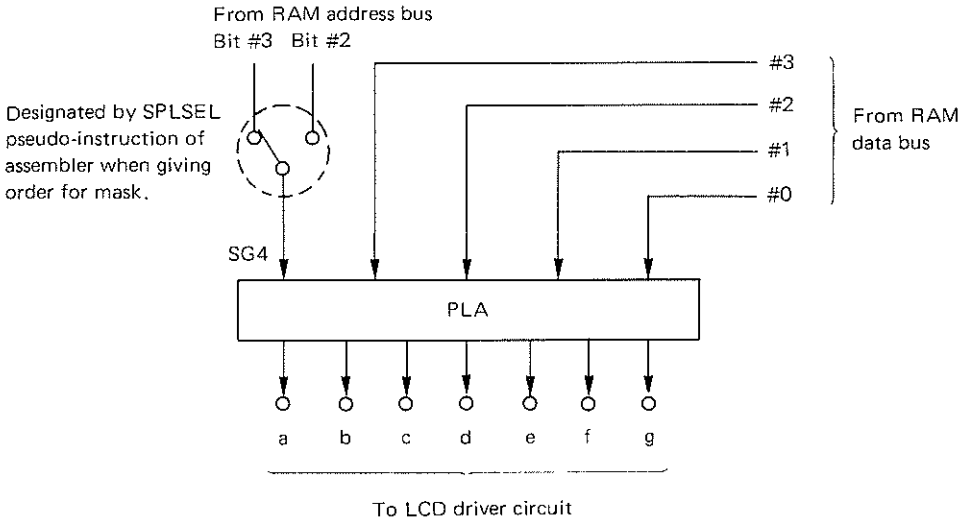
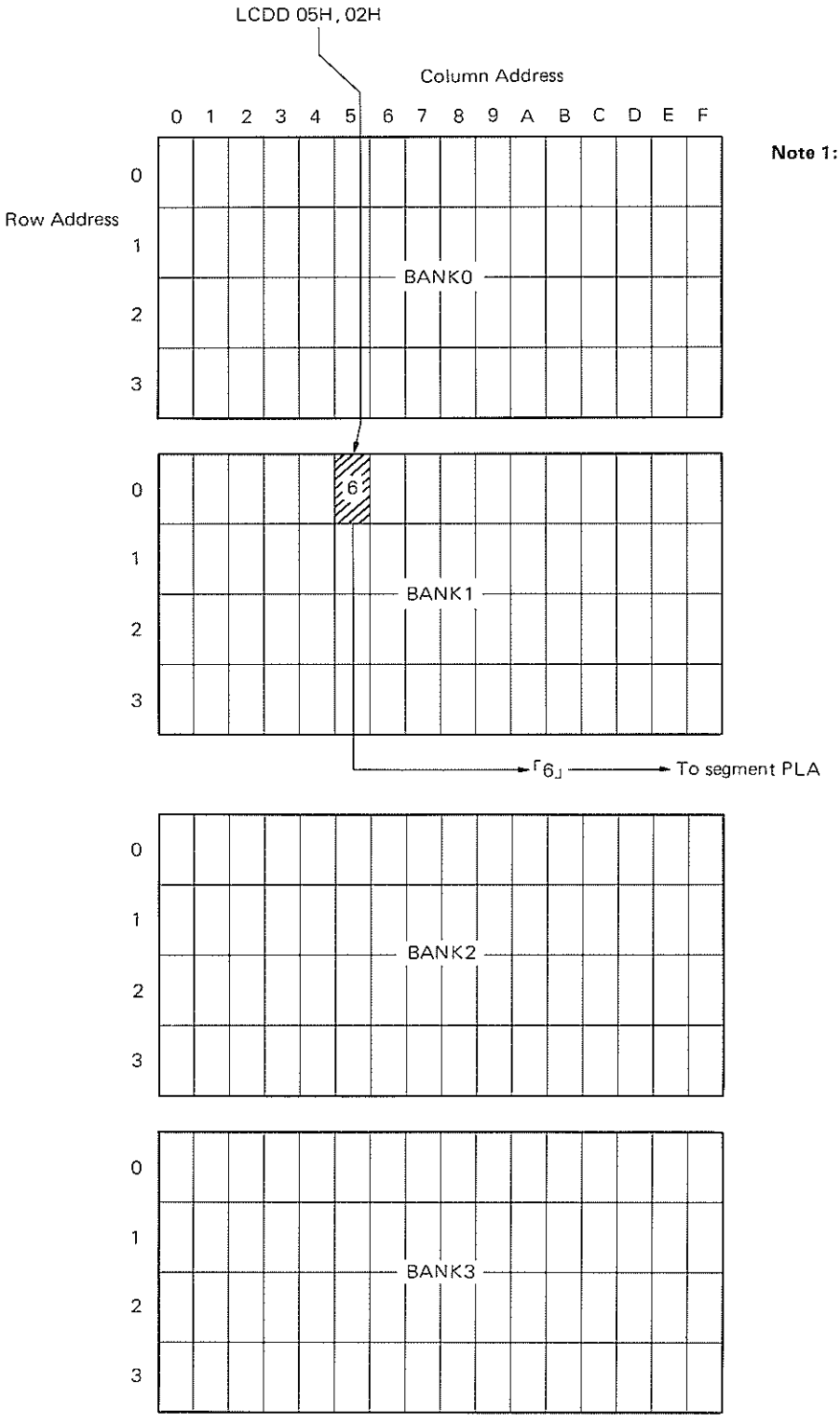


Fig. 10-1 Composition of Segment PLA

Data stored in a data memory (RAM) addressed by the first operand part of an instruction LCDD are latched in the lower four bits (SG0 to SG3) of the segment latch circuit. When, for instance, an instruction LCDD is executed with contents of RAM shown below, data stored in the address 05H of the BANK1 of the data memory (RAM), that is, "6" is latched.



Note 1: Since designated RAM is RAM in BANK1, it is necessary to execute instruction of BANK1 before instruction LCDD is executed.

Contents of the bit #3 or #2 of a column address of RAM designated by an instruction LCDD are latched in the most significant bit SG4 of the segment latch circuit. In such cases, it is necessary to designate that data of which one of the bits #3 and #2 should be latched, when giving an order for a mask. (See 10.3 the EXAMPLES OF PLA PROGRAM.)

When the bit #3 is designated, "0" and "1" are latched in SG4 respectively when RAM of the column addresses 00H to 07H and that of the column addresses 08H to 0FH are designated by an instruction LCDD. When the bit #2 is designated, "0" and "1" are latched in SG4 respectively when RAM of the column addresses 00H to 03H and 08H to 0BH and that of the column addresses 04H to 07H and 0CH to 0FH are designated.

32 types of patterns of the segment PLA may be divided into two patterns groups each comprising 16 types of patterns according to data to be latched in SG4. Therefore, even though data to be stored in RAM are the same, two different types of display patterns can be generated, if column addresses designated by an instruction LCDD are different.

16 types of patterns to be generated when data latched in SG4 is "0" are referred to as "pattern group 0", and those to be generated when data latched in SG4 is "1" are referred to "pattern group 1", respectively.

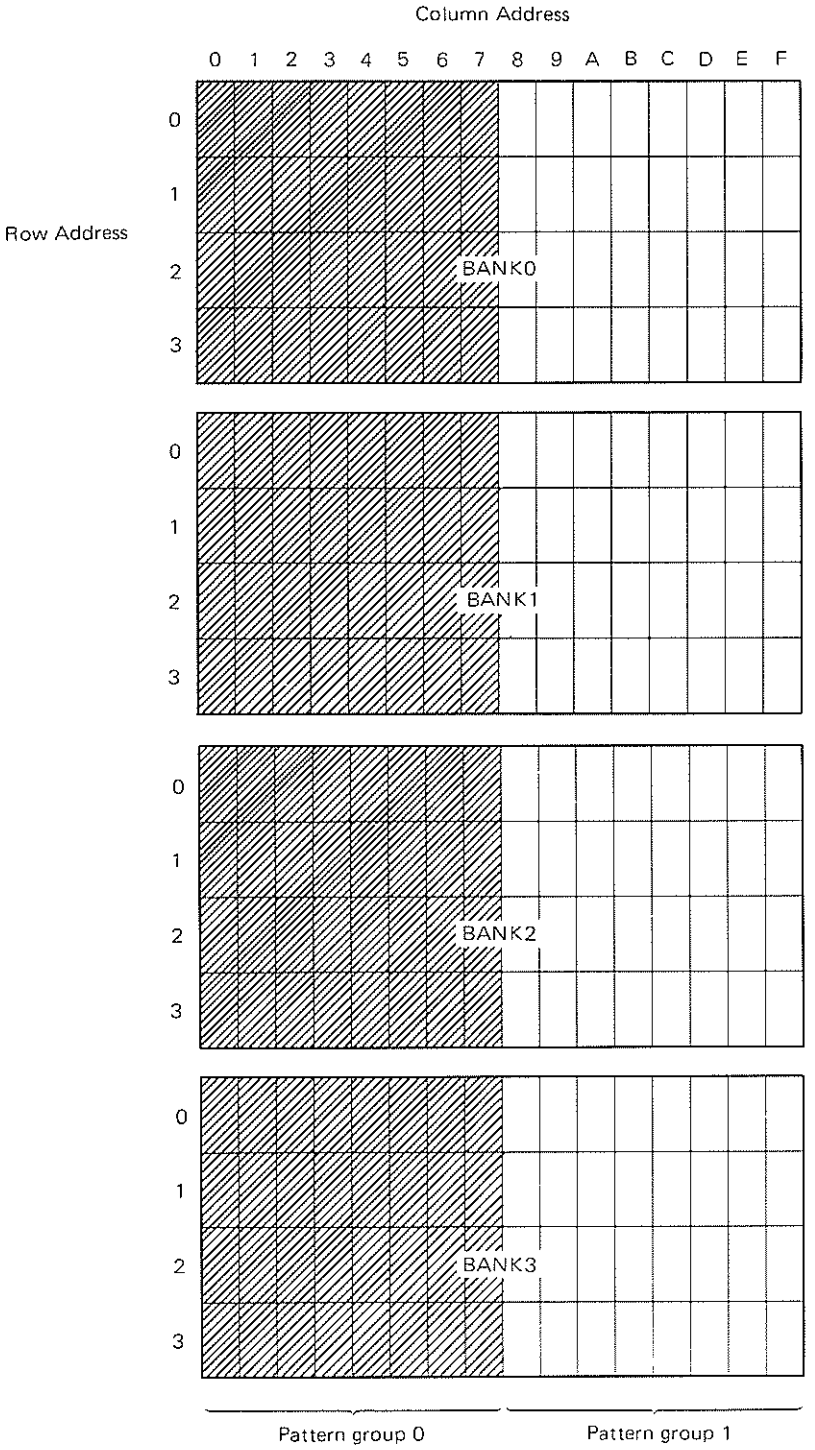


Fig. 10-2 Example of Dividing Patterns into Groups when SG4 is set to bit #3

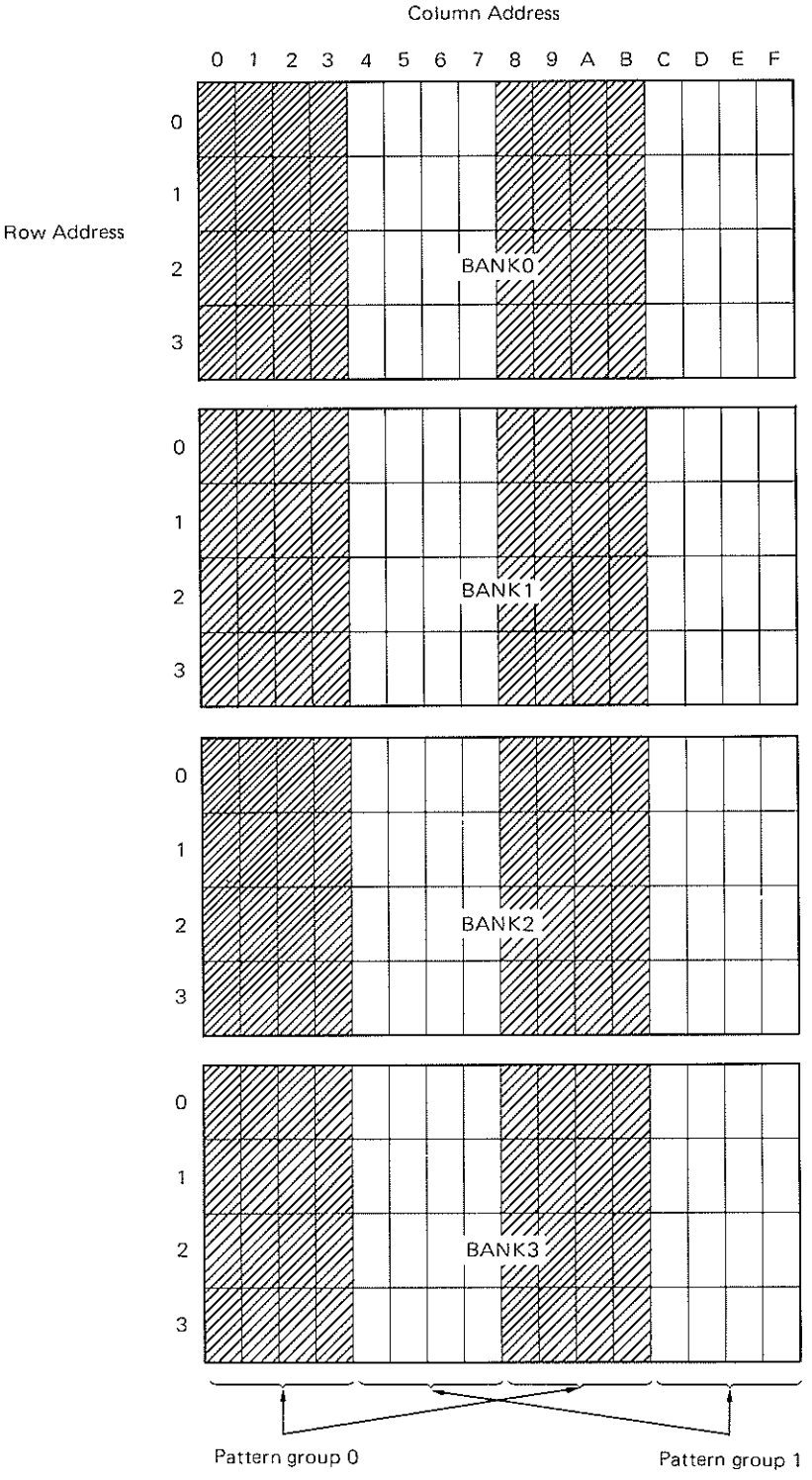


Fig. 10-3 Example of Dividing Patterns into Groups when SG4 is Set to bit #2

How to divide these pattern groups is determined with due regard to the efficiency of the RAM or program in preparing a program. It is achieved by an SPLSEL pseudo-instruction to designate an input of SG4 to the bit #3 or #2, and "SPLSEL 3" or "SPLSEL 2" is entered (see 10.3 the EXAMPLE OF PLA PROGRAM).

10.2 PATTERN EXAMPLE OF SEGMENT PLA

Examples of pattern of the pattern group 0 and 1 are given in the following Tables 10-1 and 10-2, respectively.

Table 10-1 Example of Patterns of Pattern Group 0

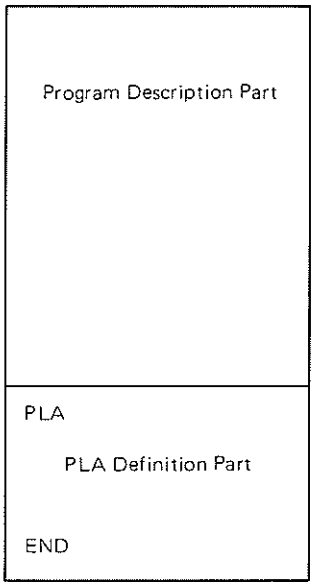
SEGMENT PLA INPUT					SEGMENT PLA OUTPUT							OUTPUT PATTERN
Add. of RAM	#3	#2	#1	#0	g	f	e	d	c	b	a	
0	0	0	0	0	0	1	1	1	1	1	1	0
	0	0	0	1	0	0	0	0	1	1	0	1
	0	0	1	0	1	0	1	1	0	1	1	2
	0	0	1	1	1	0	0	1	1	1	1	3
	0	1	0	0	1	1	0	0	1	1	0	4
	0	1	0	1	1	1	0	1	1	0	1	5
	0	1	1	0	1	1	1	1	1	0	1	6
	0	1	1	1	0	1	0	0	1	1	1	7
	1	0	0	0	1	1	1	1	1	1	1	8
	1	0	0	1	1	1	0	1	1	1	1	9
	1	0	1	0	0	0	0	0	0	0	0	BLANK (display OFF)
	1	0	1	1	1	1	1	1	0	0	1	A
	1	1	0	0	0	1	1	1	0	0	1	L
	1	1	0	1	1	1	1	0	1	1	0	H
	1	1	1	0	1	1	1	0	0	1	1	P
	1	1	1	1	1	0	0	0	0	0	0	-

Table 10-2 Example of Patterns of Pattern Group 1

SEGMENT PLA INPUT					SEGMENT PLA OUTPUT							OUTPUT PATTERN
Add. of RAM	#3	#2	#1	#0	g	f	e	d	c	b	a	
1	0	0	0	0	1	0	1	0	0	1	0	
	0	0	0	1	0	0	1	1	0	0	0	FM, MHz
	0	0	1	0	0	1	1	1	0	0	0	MF, MHz, VF
	0	0	1	1	1	1	1	1	0	0	0	FM, MHz, VF, SK
	0	1	0	0	0	0	0	0	0	0	0	NO USE
	0	1	0	1	1	1	0	0	0	0	0	
	0	1	1	0	0	0	0	0	1	0	1	MW, KHz
	0	1	1	1	0	0	0	0	1	1	0	LW, KHz
	1	0	0	0	0	0	0	0	0	0	0	NO USE
	1	0	0	1	0	0	0	0	0	0	0	NO USE
	1	0	1	0	0	0	0	0	0	0	0	BLANK (display OFF)
	1	0	1	1	0	0	0	0	0	0	0	NO USE
	1	1	0	0	0	0	0	0	0	0	0	NO USE
	1	1	0	1	0	0	0	0	0	0	0	NO USE
	1	1	1	0	0	0	0	0	0	0	0	NO USE
	1	1	1	1	0	0	0	0	0	0	0	NO USE

10.3 EXAMPLE OF PLA PROGRAM

Definition of PLA is always necessary for every type of μPD1700 series. When giving an order for a tape, a tape whose PLA part is not defined is unacceptable. Definition of PLA is described at the end of a source program of assembler, and consists of the items shown below. These items must be all described, and even one item should not be omitted.



(1) PLA Pseudo-instruction

A PLA pseudo-instruction is a description which represents the end of a program description part and, at the same time, the beginning of a PLA definition part.

(2) SPLSEL (Segment PLA Select) Pseudo-instruction

An SPLSEL pseudo-instruction is a description which selects the division of RAM addresses where segment patterns groups 0 and 1 are generated. SPLSEL pseudo-instructions may be classified into the following two types:

SPLSEL 3, SPLSEL 2

(3) DSP (Define Segment PLA) Pseudo-instruction

This pseudo-instruction defines 32 types of patterns of the segment PLA. In this case, it is necessary to define patterns in order beginning with 16 types of the pattern group 0. An example of description is shown below. The first bit corresponds to the segment g, and the following bits correspond to f, e, d, c, b and a, respectively.

```
DSP    1 1 1 1 0 0 1 B
       ↑         ↑
       g         a
```

(4) END

This description represents the end of a PLA definition part and, at the same time, the end of a source program. No assembly is made when this description is not made.

Note 1: With respect to types with a DIGIT PLA (μPD1701, μPD1703, μPD1704, μPD1705, μPD1707, μPD1710, μPD1711 and μPD1712) among the μPD1700 series, definition of the DIGIT PLA is also necessary. Definition of the DIGIT PLA is made by a DDP (Define Digit PLA) pseudo-instruction.
In the same way, with respect to type with a Digit pin (μPD1701, μPD1703, μPD1704, μPD1705, μPD1707, μPD1709, μPD1710, μPD1711, μPD1712), definition of the MTDIG is also necessary.

Note 2: When defining a PLA, it is necessary to enter "PLA" at first and "END" lastly, but "SPLSEL", "DSP", "DDP" and "MTDIG" between "PLA" and "END" may be freely entered regardless of order.

11. μPD17P23 INSTRUCTION

11.1 INSTRUCTION SET

b_{13} b_{12} b_{11} b_{10} \ b_{15} b_{14}		0 0	0 1	1 0	1 1
		0	1	2	3
0 0 0 0	0	SIO s IFCW w IFC t NOP	KIN M KI M	———	ST M, r
0 0 0 1	1	SPB P, N SS N_1 BANK1 BANK2 BANK3 EI STC	ORI M, I	———	MVRS M, r
0 0 1 0	2	JMP ADDR (page 1)	MVI M, I	OUT P, r	IN r, P
0 0 1 1	3	RPB P, N RS N_1 BANK0 DI RSC	ANI M, I	CKSTP HALT h	MVRD r, M
0 1 0 0	4	RT	AI M, I	MVSR M_1, M_2	AD r, M
0 1 0 1	5	RTS	SI M, I	EXL r, M	SU r, M
0 1 1 0	6	JMP ADDR (page 0)	AIC M, I	LD r, M	AC r, M
0 1 1 1	7	CAL ADDR (page 0)	SIB M, I	LCDD M, D	SB r, M
1 0 0 0	8	SBK0 TPF P, N TSF N_2 TCEF TITF TB0F TB1F	AIN M, I	TKLT TKLF TSET TSEF TADT TADF	ADN r, M
1 0 0 1	9	TPT P, N TST N_2 TCET TITT TB0T TB1T	SIN M, I	TTM TIP TGC	SUN r, M
1 0 1 0	A	TMF M, N	AICN M, I	TUL	ACN r, M
1 0 1 1	B	TMT M, N	SIBN M, I	PLL M, r	SBN r, M
1 1 0 0	C	SLTI M, I	AIS M, I	SLT r, M	ADS r, M
1 1 0 1	D	SGEI M, I	SIS M, I	SGE r, M	SUS r, M
1 1 1 0	E	SEQI M, I	AICS M, I	SEQ r, M	ACS r, M
1 1 1 1	F	SNEI M, I	SIBS M, I	SNE r, M	SBS r, M

11.2 INSTRUCTIONS

NOTE: D_H: Data memory address high (row address) (2 bits)
D_L: Data memory address low (column address) (4 bits)
Rn: Register number (4 bits)
I : Immediate data (4 bits)
N : Bit position (4 bits)
ADDR: Program memory address (10 bits)
—: All "1"
r : General register
One of addresses 00−0FH of BANK0
M : Data memory address
One of 00−3FH of BANK0 and 00−3FH of BANK1
P : Port $0 \leqq P \leqq 3$

N_1 : Bit position of status word 1 $0 \leqq N_1 \leqq 7$
N_2 : Bit position of status word 2 $0 \leqq N_2 \leqq 7$
( ) : Contents of register or memory
c : Carry
b : Borrow
s : Data to S.M.R. $0 \leqq s \leqq 0FH$
w : Data to IF Control Word $0 \leqq w \leqq 0FH$
t : Trigger conditions $0 \leqq t \leqq 3$
( )n : Contents of bit N in memory or register
h : Halt release conditions $0 \leqq h \leqq 7$

	Mnemonic	Operand 1ST	Operand 2ND	Function	Operation	Machine code: Operation code			
Addition	AD	r	M	Add memory to register	r←(r)+(M)	1 1 0 1 0 0	D_H	D_L	Rn
	ADS	r	M	Add memory to register, then skip if carry	r←(r)+(M) skip if carry	1 1 1 1 0 0	D_H	D_L	Rn
	ADN	r	M	Add memory to register, then skip if not carry	r←(r)+(M) skip if not carry	1 1 1 0 0 0	D_H	D_L	Rn
	AC	r	M	Add memory to register with carry	r←(r)+(M)+c	1 1 0 1 1 0	D_H	D_L	Rn
	ACS	r	M	Add memory to register with carry, then skip if carry	r←(r)+(M)+c skip if carry	1 1 1 1 1 0	D_H	D_L	Rn
	ACN	r	M	Add memory to register with carry, then skip if not carry	r←(r)+(M)+c skip if not carry	1 1 1 0 1 0	D_H	D_L	Rn
	AI	M	I	Add immediate data to memory	M←(M)+I	0 1 0 1 0 0	D_H	D_L	I
	AIS	M	I	Add immediate data to memory, then skip if carry	M←(M)+I skip if carry	0 1 1 1 0 0	D_H	D_L	I
	AIN	M	I	Add immediate data to memory, then skip if not carry	M←(M)+I skip if not carry	0 1 1 0 0 0	D_H	D_L	I
	AIC	M	I	Add immediate data to memory with carry	M←(M)+I+c	0 1 0 1 1 0	D_H	D_L	I
	AICS	M	I	Add immediate data to memory with carry, then skip if carry	M←(M)+I+c skip if carry	0 1 1 1 1 0	D_H	D_L	I
	AICN	M	I	Add immediate data to memory with carry, then skip if not carry	M←(M)+I+c skip if not carry	0 1 1 0 1 0	D_H	D_L	I
Subtraction	SU	r	M	Subtract memory from register	r←(r)−(M)	1 1 0 1 0 1	D_H	D_L	Rn
	SUS	r	M	Subtract memory from register, then skip if borrow	r←(r)−(M) skip if borrow	1 1 1 1 0 1	D_H	D_L	Rn
	SUN	r	M	Subtract memory from register, then skip if not borrow	r←(r)−(M) skip if not borrow	1 1 1 0 0 1	D_H	D_L	Rn
	SB	r	M	Subtract memory from register with borrow	r←(r)−(M)−b	1 1 0 1 1 1	D_H	D_L	Rn
	SBS	r	M	Subtract memory from register with borrow, then skip if borrow	r←(r)−(M)−b skip if borrow	1 1 1 1 1 1	D_H	D_L	Rn
	SBN	r	M	Subtract memory from register with borrow, then skip if not borrow	r←(r)−(M)−b skip if not borrow	1 1 1 0 1 1	D_H	D_L	Rn
	SI	M	I	Subtract immediate data from memory	M←(M)−I	0 1 0 1 0 1	D_H	D_L	I
	SIS	M	I	Subtract immediate data from memory, then skip if borrow	M←(M)−I skip if borrow	0 1 1 1 0 1	D_H	D_L	I
	SIN	M	I	Subtract immediate data from memory, then skip if not borrow	M←(M)−I skip if not borrow	0 1 1 0 0 1	D_H	D_L	I
	SIB	M	I	Subtract immediate data from memory with borrow	M←(M)−I−b	0 1 0 1 1 1	D_H	D_L	I
	SIBS	M	I	Subtract immediate data from memory with borrow, then skip if borrow	M←(M)−I−b skip if borrow	0 1 1 1 1 1	D_H	D_L	I
	SIBN	M	I	Subtract immediate data from memory with borrow, then skip if not borrow	M←(M)−I−b skip if not borrow	0 1 1 0 1 1	D_H	D_L	I

	Mnemonic	Operand 1ST	Operand 2ND	Function	Operation	Machine code: Operation code			
Comparison	SEQ	r	M	Skip if register equals memory	r−M skip if zero	1 0 1 1 1 0	D_H	D_L	Rn
	SNE	r	M	Skip if register not equals memory	r−M skip if not zero	1 0 1 1 1 1	D_H	D_L	Rn
	SGE	r	M	Skip if register is greater than or equal to memory	r−M skip if not borrow (r) ≧ (M)	1 0 1 1 0 1	D_H	D_L	Rn
	SLT	r	M	Skip if register is less than memory	r−M skip if borrow (r) < (M)	1 0 1 1 0 0	D_H	D_L	Rn
	SEQI	M	I	Skip if memory equals immediate data	M−I skip if zero	0 0 1 1 1 0	D_H	D_L	I
	SNEI	M	I	Skip if memory not equals immediate data	M−I skip if not zero	0 0 1 1 1 1	D_H	D_L	I
	SGEI	M	I	Skip if memory is greater than or equal to immediate data	M−I skip if not borrow (M) ≧ I	0 0 1 1 0 1	D_H	D_L	I
	SLTI	M	I	Skip if memory is less than immediate data	M−I skip if borrow (M) < I	0 0 1 1 0 0	D_H	D_L	I
Logical operation	ANI	M	I	Logic AND of memory and immediate data	M←(M) ∧ I	0 1 0 0 1 1	D_H	D_L	$\overline{I}$
	ORI	M	I	Logic OR of memory and immediate data	M←(M) ∨ I	0 1 0 0 0 1	D_H	D_L	I
	EXL	r	M	Exclusive OR Logic of memory and register	r←(r) ⊕ (M)	1 0 0 1 0 1	D_H	D_L	Rn
Transfer	LD	r	M	Load memory to register	r←(M)	1 0 0 1 1 0	D_H	D_L	Rn
	ST	M	r	Store register to memory	M←(r)	1 1 0 0 0 0	D_H	D_L	Rn
	MVRD	r	M	Move memory to destination memory referring to register in the same row	(D_H, Rn)←(M)	1 1 0 0 1 1	D_H	D_L	Rn
	MVRS	M	r	Move source memory referring to register to memory in the same row	M←(D_H, Rn)	1 1 0 0 0 1	D_H	D_L	Rn
	MVSR	M_1	M_2	Move memory to memory in the same row	(D_H, D_{L1})←(D_H, D_{L2})	1 0 0 1 0 0	D_H	D_{L1}	D_{L2}
	MVI	M	I	Move immediate data to memory	M← I	0 1 0 0 1 0	D_H	D_L	I
	PLL	M	r	Load N0~N3, N_F & memory to PLL registers	PLLR←(N0~N3), N_F & (M)	1 0 1 0 1 1	D_H	D_L	Rn
Bit test	TMT	M	N	Test memory bits, then skip if all bits specified are true	if M(N) = all "1", then skip	0 0 1 0 1 1	D_H	D_L	N
	TMF	M	N	Test memory bits, then skip if all bits specified are false	if M(N) = all "0", then skip	0 0 1 0 1 0	D_H	D_L	N
Jump	JMP	ADDR		Jump to the address specified in page 0	PC←ADDR, PAGE←0	0 0 0 1 1 0	ADDR(10 bits)		
				Jump to the address specified in page 1	PC←ADDR, PAGE←1	0 0 0 0 1 0			
Subroutine	CAL	ADDR		Call subroutine in page 0	Stack←((PC)+1, PAGE), PC←ADDR, PAGE←0	0 0 0 1 1 1	ADDR(10 bits)		
	RT			Return to main routine	PC←(stack)	0 0 0 1 0 0	—	—	—
	RTS			Return to main routine, then skip unconditionally	PC←(stack), and skip	0 0 0 1 0 1	—	—	—
Interrupt	EI			Enable interrupt	INTE F/F←1	0 0 0 0 0 1	—	0 0 0 1	—
	DI			Disable interrupt	INTE F/F←0	0 0 0 0 1 1	—	0 0 0 1	—
F/F test	TTM			Test and reset timer F/F, then skip if it has not been set	if Timer F/F=1, then Timer F/F←0 if Timer F/F=0, then skip	1 0 1 0 0 1	—	—	—
	TUL			Test and reset unlock F/F, then skip if it has not been set	if UL F/F=1, then UL F/F←0 if UL F/F=0, then skip	1 0 1 0 1 0	—	—	—
	TKLT			Test then reset Key Latch F/F, then skip if true	if KL F/F=1, then skip and KL F/F←0	1 0 1 0 0 0	—	0 0 0 1	0 0 0 0
	TKLF			Test then reset Key Latch F/F, then skip if false	if KL F/F=1, then KL F/F←0 if KL F/F=0, then skip	1 0 1 0 0 0	0 1	0 0 0 1	0 0 0 0
Test timer	TIP			Test interval pulse, then skip if low	if IPG=0, then skip	1 0 1 0 0 1	—	0 0 0 0	0 0 0 0
IF counter	IFCW	w		Set immediate data to IFCW	IFCW←w	0 0 0 0 0 0	1 0	0 0 0 0	w
	IFC	t		Trigger and/or reset IF counter	Trigger←t_1, Reset←t_0	0 0 0 0 0 0	0 1	0 0 0 0	0 0 t
	TGC			Test IF counter gate, skip if close	if IFC gate=close, then skip	1 0 1 0 0 1	0 0	—	—

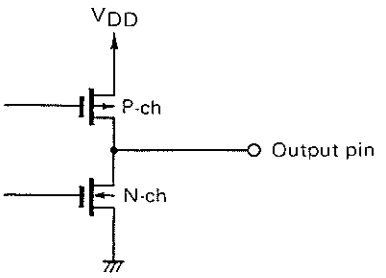
	Mnemonic	Operand 1ST	Operand 2ND	Function	Operation	Machine code: Operation code			
Status word and terminal test	SS	N_1		Set status word 1	(STATUS WORD 1)$_N$←1	0 0 0 0 0 1	—	N_1	—
	RS	N_1		Reset status word 1	(STATUS WORD 1)$_N$←0	0 0 0 0 1 1	—	N_1	—
	TST	N_2		Test status word 2 true	if (STATUS WORD 2)$_N$=all 1, then skip	0 0 1 0 0 1	—	N_2	—
	TSF	N_2		Test status word 2 false	if (STATUS WORD 2)$_N$=all 0, then skip	0 0 1 0 0 0	—	N_2	—
	STC			Set carry F/F	carry F/F←1	0 0 0 0 0 1	—	0 0 1 0	—
	RSC			Reset carry F/F	carry F/F←0	0 0 0 0 1 1	—	0 0 1 0	—
	BANK0			Select BANK0	BANK F/F0←0, BANK F F1←0	0 0 0 0 1 1	—	1 1 0 0	—
	BANK1			Select BANK1	BANK F/F0←1, BANK F F1←0	0 0 0 0 0 1	—	0 1 0 0	—
	BANK2			Select BANK2	BANK F/F0←0, BANK F F1←1	0 0 0 0 0 1	—	1 0 0 0	—
	BANK3			Select BANK3	BANK F/F0←1, BANK F F1←1	0 0 0 0 0 1	—	1 1 0 0	—
	TITT			Test INT, skip if true	if $\overline{INT}$=0, then skip	0 0 1 0 0 1		0 0 0 1	—
	TITF			Test INT, skip if false	if $\overline{INT}$=1, then skip	0 0 1 0 0 0		0 0 0 1	—
	TCET			Test CE, skip if true	if CE=1, then skip	0 0 1 0 0 1		0 0 1 0	—
	TCEF			Test CE, skip if false	if CE=0, then skip	0 0 1 0 0 0	—	0 0 1 0	—
	SBK0			Skip if BANK0	if BANK F/F0=BANK F F1=0, then skip	0 0 1 0 0 0		1 1 0 0	—
	TB0T			Test BANK F/F0, skip if true	if BANK F/F0=1, then skip	0 0 1 0 0 1	—	0 1 0 0	—
	TB0F			Test BANK F/F0, skip if false	if BANK F/F0=0, then skip	0 0 1 0 0 0	—	0 1 0 0	—
	TB1T			Test BANK F/F1, skip if true	if BANK F/F1=1, then skip	0 0 1 0 0 1	—	1 0 0 0	—
	TB1F			Test BANK F/F1, skip if false	if BANK F/F1=0, then skip	0 0 1 0 0 0		1 0 0 0	—
Input / output	LCDD	M	D	Output segment pattern to LCD digit "D" based on memory, or output to LCD digit directly	LCD(D)←SEG PLA←(M), or LCD(D)←(M)	1 0 0 1 1 1	D_H	D	D_L
	KI	M		Input key data to memory	M←$K_{0\sim3}$	0 1 0 0 0 0	D_H	D_L	0 0 0 0
	KIN	M		Input key data to memory, then skip if data are zero	M←$K_{0\sim3}$, skip if(M)=0	0 1 0 0 0 0	D_H	D_L	—
	IN	r	P	Input data on port to register	r←(Port (P))	1 1 0 0 1 0	P	—	Rn
	OUT	P	r	Output contents of register to port	(Port (P))←(r)	1 0 0 0 1 0	P	—	Rn
	SPB	P	N	Set port bits	(Port (P))$_N$←1	0 0 0 0 0 1	P	0 0 0 0	N
	RPB	P	N	Reset port bits	(Port (P))$_N$←0	0 0 0 0 1 1	P	0 0 0 0	N
	TPT	P	N	Test port bits, then skip if all bits specified are true	if (Port (P))$_N$=all 1s, then skip	0 0 1 0 0 1	P	0 0 0 0	N
	TPF	P	N	Test port bits, then skip if all bits specified are false	if (Port (P))$_N$=all 0s, then skip	0 0 1 0 0 0	P	0 0 0 0	N
Serial I/O	SIO	s		Serial input/output	SMR (3,1,0)←s (3,1,0)	0 0 0 0 0 0	0 0	0 0 0 1	s
	TSET			Test shift end, then skip if true	if SCC=8/(2n+1), then skip (n≧0)	1 0 1 0 0 0	1 0	0 0 0 1	—
	TSEF			Test shift end, then skip if false	if SCC≠8/(2n+1), then skip (n≧0)	1 0 1 0 0 0	0 0	0 0 0 1	—
Test A/D	TADT			Test A−D comparator, then skip if true	if Vin > Vcomp, then skip	1 0 1 0 0 0	0 0	0 0 0 0	—
	TADF			Test A−D comparator, then skip if false	if Vin ≦ Vcomp, then skip	1 0 1 0 0 0	1 0	0 0 0 0	—
Others	CKSTP			Clock stop by CE	stop clock if CE=0	1 0 0 0 1 1	—	1 1 1 0	1 1 1 0
	HALT	h		Halt the CPU, Restart by condition h	Halt	1 0 0 0 1 1	0 0	—	h l
	NOP			No operation		0 0 0 0 0 0	—	—	—

Note : Signals marked by an asterisk are reformed.

12. INPUT/OUTPUT CIRCUITS

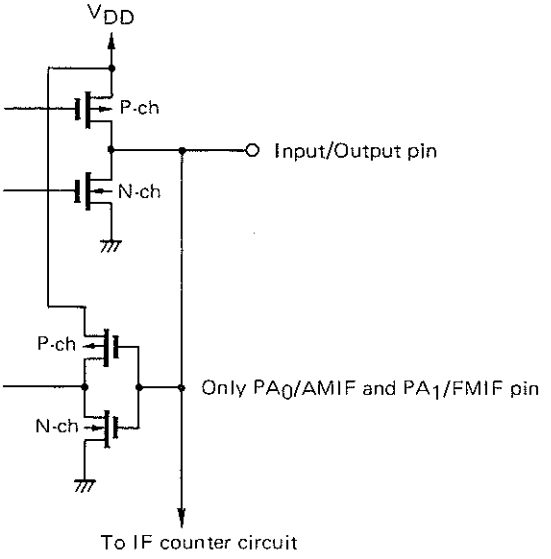
The followings are input/output circuits of each pin of μPD17P23 (There are some omissions in it).

(1) LCD_0/KS_0 to LCD_{27}/PL_3, CGP, PB_0/SO to PB_3, PD_1 to PD_3, EO_1 and EO_2

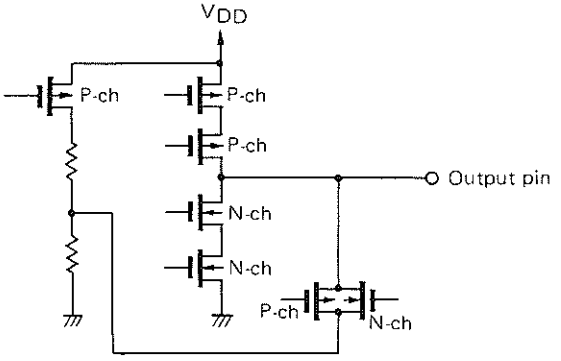


(2) PA_0/AMIF, PA_1/FMIF, PA_2/SI, $PA_3/\overline{SCK}$ and PC_0 to PC_3

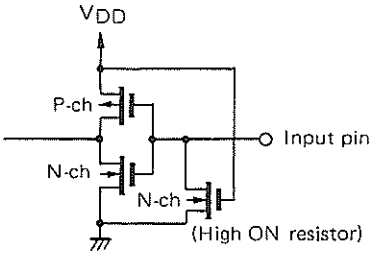
(See Fig. 7-1 about PA_0/AMIF and PA_1/FMIF)



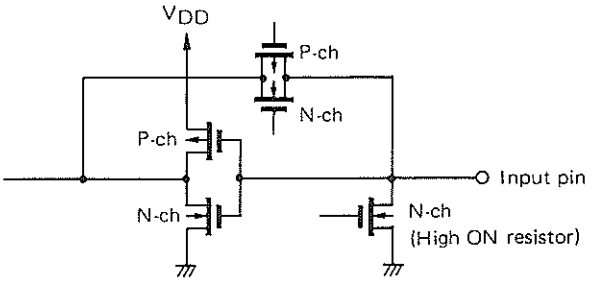
(3) COM_1 and COM_2



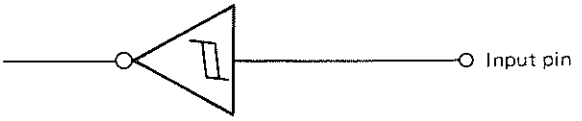
(4) K_0 to K_3



(5) VCOH and VCOL



(6) CE and $\overline{INT}$



Schmitt-Trigger

13. ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

Supply Voltage	V_{DD}	-0.3 to +6.0	V
Input Voltage	V_I	-0.3 to $+V_{DD}$+0.3	V
Output Voltage	V_O	-0.3 to $+V_{DD}$+0.3	V
Output Absorption Current	I_O	10	mA
Storage Tmeperature	T_{stg}	-55 to +125	°C
Operating Temperature	T_a	-40 to +85	°C

RECOMMENDED OPERATING CONDITIONS

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT	CONDITION
Supply Voltage	V_{DD1}	4.5	5	5.5	V	CPU and PLL operation
Supply Voltage	V_{DD2}	3.5	5	5.5	V	CPU operation
Data Retention Voltage	V_{DR}	2.5		5.5	V	Crystal oscillation stopped
Supply Voltage Rising Time	T_{rise}			500	ms	V_{DD} = Low to High
Input Oscillation Voltage	V_{in1}	0.3		V_{DD}	V_{p-p}	VCOL, VCOH
Input Oscillation Voltage	V_{in2}	0.1		V_{DD}	V_{p-p}	AMIF, FMIF
Operating Temperature	T_a	-40		85	°C	

DC CHARACTERISTICS (T_a = -40 to +85 °C, V_{DD} = 4.5 to 5.5 V)

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT	CONDITION
High Level Input Voltage	V_{IH1}	0.7 V_{DD}			V	PORTS A, C
High Level Input Voltage	V_{IH2}	0.8 V_{DD}			V	CE, $\overline{INT}$
High Level Input Voltage	V_{IH3}	0.6 V_{DD}			V	K_3 to K_0
Low Level Input Voltage	V_{IL1}			0.2 V_{DD}	V	PORT A, PORT C, CE, $\overline{INT}$
Low Level Input Voltage	V_{IL2}			0.15 V_{DD}	V	K_3 to K_0
High Level Output Current	I_{OH1}	−0.4			mA	PORTS A, B, C, D $V_{OH}=V_{DD}-0.4$ V
High Level Output Current	I_{OH2}	−0.5			mA	EO_1, EO_2, CGP, LCD_{27}/PL_3 to LCD_{24}/PL_0 $V_{OH}=V_{DD}-1$ V
High Level Output Current	I_{OH3}	−200	−300		μA	LCD_0 to LCD_{23} $V_{OH}=V_{DD}-1$ V
Low Level Output Current	I_{OL1}	0.6			mA	PORTS A, B, C, D, CGP, LCD_{27}/PL_3 to LCD_{24}/PL_0 V_{OL}=0.4 V
Low Level Output Current	I_{OL2}	0.5			mA	EO_1, EO_2 V_{OL}=1 V
Low Level Output Current	I_{OL3}	200	300		μA	LCD_0 to LCD_{23} V_{OL}=1 V
High Level Input Current	I_{IH1}	15	90	150	μA	K_3 to K_0 $V_I=V_{DD}$=4.5 V
High Level Input Current	I_{IH2}	100			μA	VCOH, VCOL, XI $V_I=V_{DD}$=4.5 V
Output Voltage	V_{COM1}	4.8	5.0		V	COM_1, COM_2 V_{DD}=5 V, Output Open
Output Voltage	V_{COM2}	2.3	2.5	2.7	V	COM_1, COM_2 V_{DD}=5 V, Output Open
Output Voltage	V_{COM3}	0	0.2		V	COM_1, COM_2 V_{DD}=5 V, Output Open
Output OFF Leak Current	I_L		10^{-3}	1	μA	EO_1, EO_2 $V_O=V_{DD}$, T_a=25° C
A/D Converter Resolution				6	bit	
A/D Converter Absolute Accuracy			1	1.5	LSB	T_{opt}=−10 to +50°C
A/D Input Resister	R_I	1			MΩ	
PLL Operation Supply Current	I_{DD1}		20		mA	CPU and PLL operation (fin=150 MHz) input to VCOH V_{DD}=5 V, T_a=25 °C
CPU Operation Supply Current	I_{DD2}		1.5		mA	CPU operation V_{DD}=5 V, T_a=25 °C
Data Retention Current	I_{DR}		3	10	μA	Crystal oscillation stopped, T_a=25 °C

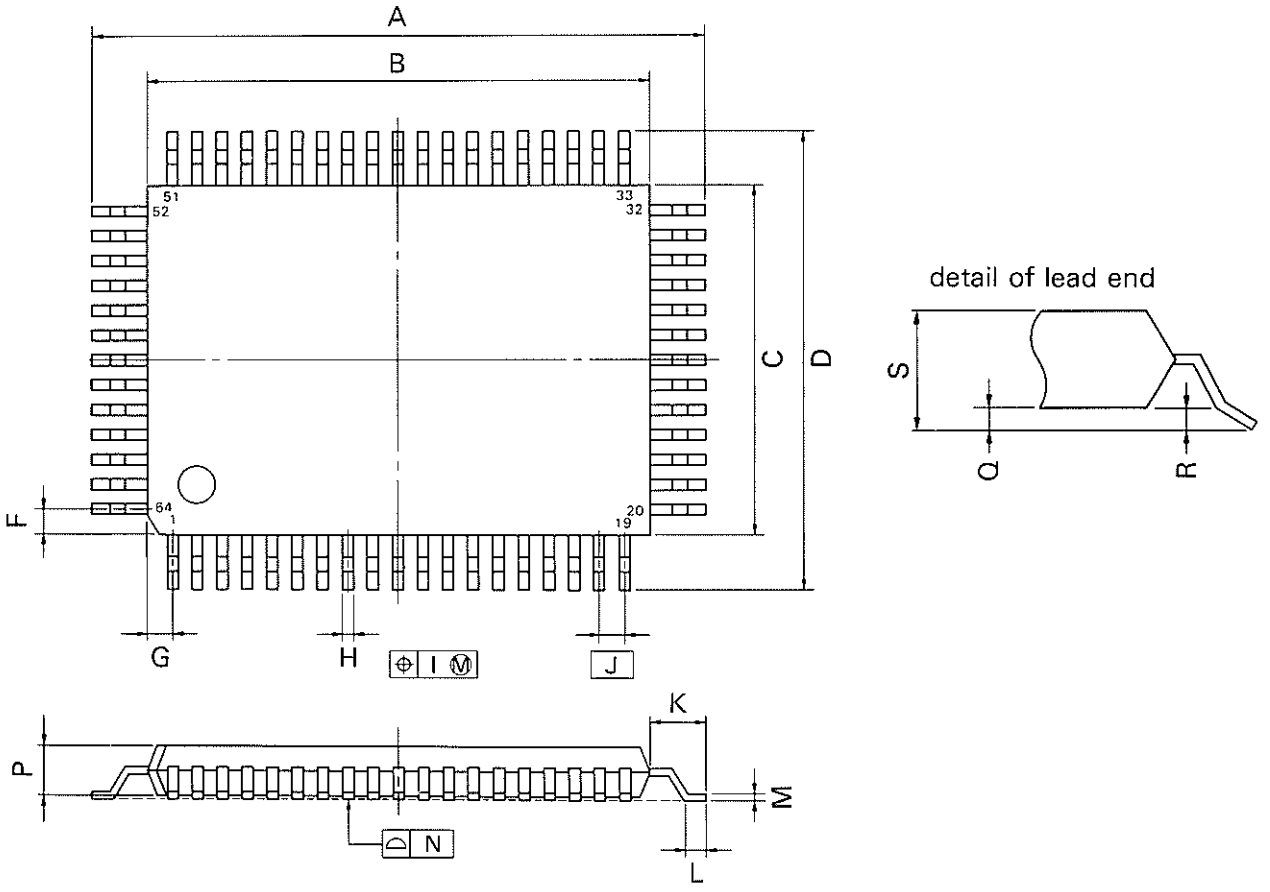
AC CHARACTERISTICS (T_a = -40 to +85 °C, V_{DD} = 4.5 to 5.5 V)

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT	CONDITION
Operating Frequency	f_{in1}	15		150*	MHz	VCOH (sine wave) V_{in}=0.3 V_{p-p}
Operating Frequency	f_{in2}	0.5		30	MHz	VCOL (sine wave) V_{in}=0.3 V_{p-p}
Operating Frequency	f_{in3}	3		20	MHz	PA_1/FMIF (sine wave) V_{in}=0.1 V_{p-p}
Operating Frequency	f_{in4}	0.3		5	MHz	PA_0/AMIF (sine wave) V_{in}=0.1 V_{p-p}

Note *: Operating frequency (f_{in1}) is 200 MHz MAX. for μPD1723.

14. PACKAGE DIMENSIONS

64 PIN PLASTIC QFP (14×20)



S64GF-100-3B8,3BE

NOTE

Each lead centerline is located within 0.20 mm (0.008 inch) of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS	INCHES
A	$23.2^{\pm 0.4}$	$0.913^{+0.017}_{-0.016}$
B	$20^{+0.2}$	$0.787^{+0.009}_{-0.008}$
C	$14^{\pm 0.2}$	$0.551^{+0.009}_{-0.008}$
D	$17.2^{+0.4}$	$0.677^{+0.016}$
F	1.0	0.039
G	1.0	0.039
H	$0.40^{+0.10}$	$0.016^{+0.004}_{-0.005}$
I	0.20	0.008
J	1.0 (T.P.)	0.039 (T.P.)
K	$1.6^{+0.2}$	$0.063^{\pm 0.008}$
L	$0.8^{+0.2}$	$0.031^{+0.009}_{0.008}$
M	$0.15^{+0.10}_{0.05}$	$0.006^{+0.004}_{0.003}$
N	0.15	0.006
P	2.7	0.106
Q	$0.1^{+0.1}$	$0.004^{+0.004}$
R	$0.1^{+0.1}$	$0.004^{\pm 0.004}$
S	3.0 MAX.	0.119 MAX.

15. SUPPORT TOOLS

The following support tools are available for developing systems using the μPD17P23

Hard ware	EVAKIT-1700		This is an evaluation board usable in common in the μPD1700 series. In case of μPD17P23, EVAKIT-1700 must be used with option I/O board (EV-1714) to develop systems. It is operated on PROM base and possible that immediately adding and revising the program through the console.		
	SE-1700		This is a simulation board in which the program developed by EVAKIT-1700 is loaded and provided in a system instead of the μPD17P23 to evaluate the system. In case of μPD17P23, SE-1700 must be used with option I/O board (EV-1714) to evaluate systems.		
	EV-1714		This is an option I/O board connecting to EVAKIT-1700 or SE-1700 for developing and evaluating the program of μPD17P23. EV-1714 can be used for developing the program of μPD1714, μPD1719, μPD1722 or μPD1723 also.		
Soft ware	μPD1700 series Assembler	Absolute Assembler	Host machine	OS	Order name
			PDA-880	CP/M-80™	μS281AS1700
			PDA-800 + PDA-800FDD		
			MD-080 series		
			MD-086 series	MP/M-86™	μS171AS1700
EPROM programmer	AF-9703 (Ando electric Co., Ltd. make)		This is EPROM programmer. In case of using for μPD17P23, AF-9703 must be used with the adaptor AF-9776. (Please use EPROM programmer which its version number is 5 or over)		
	AF-9776 (Ando electric Co., Ltd. make)		This adaptor is used with AF-9703. This adaptor can be used when writing to μPD17P08, μPD17P13 or μPD17P19 too.		

Notes to use of the support tool

"μPD17P19" must be specified as the product type when assembling the program of μPD17P23, or when using EVAKIT-1700.

Remarks:

1. Please inquire of the Ando electric Co., Ltd. about the EPROM programmer.
2. The software is supplied on 8-inch double-sided double-density floppy disk.
3. Assembler operation is guaranteed when operating in an NEC host machine (PDA or MD Series) by an OS also supplied by NEC (see above table).
4. The PDA-80, PDA-800, and PDA-880 models have been discontinued, and apart from PDA-880 (using CP/M as OS) the machine-based software has become maintenance products. For this reason, these machine-based software versions are not being upgraded.
5. To convert the PDA-880, and PDA-800 + PDA-800FDD OS to CP/M-80, contact NEC or an NEC agency.
6. EVAKIT is a trademark of the NEC corporation.
7. CP/M-80™ and MP/M-086™ are the trademark of the Digital Research Corp., Ltd.

IC—2429
April 1989 P
Printed in Japan