NEC
ELECTRON DEVICE

MOS INTEGRATED CIRCUIT

μPD1708AG

SINGLE-CHIP MICROCOMPUTER WITH BUILT-IN RPESCALER, PLL FREQUENCY SYNTHESIZER AND LCD DRIVER

The μPD1708 is a 4-bit CMOS microcomputer for digital tuning, which incorporates a prescaler that can be operated up to 150 MHz, a PLL frequency synthesizer and an LCD driver (1/2 duty; 1/2 bias) into one chip.

Its CPU provides the functions of 4-bit parallel addition/subtraction (AD and SU instructions), logical operation (EXL instruction), multiple bit test (TMT instruction), carry F/F set/reset (STC instruction) and timer function.

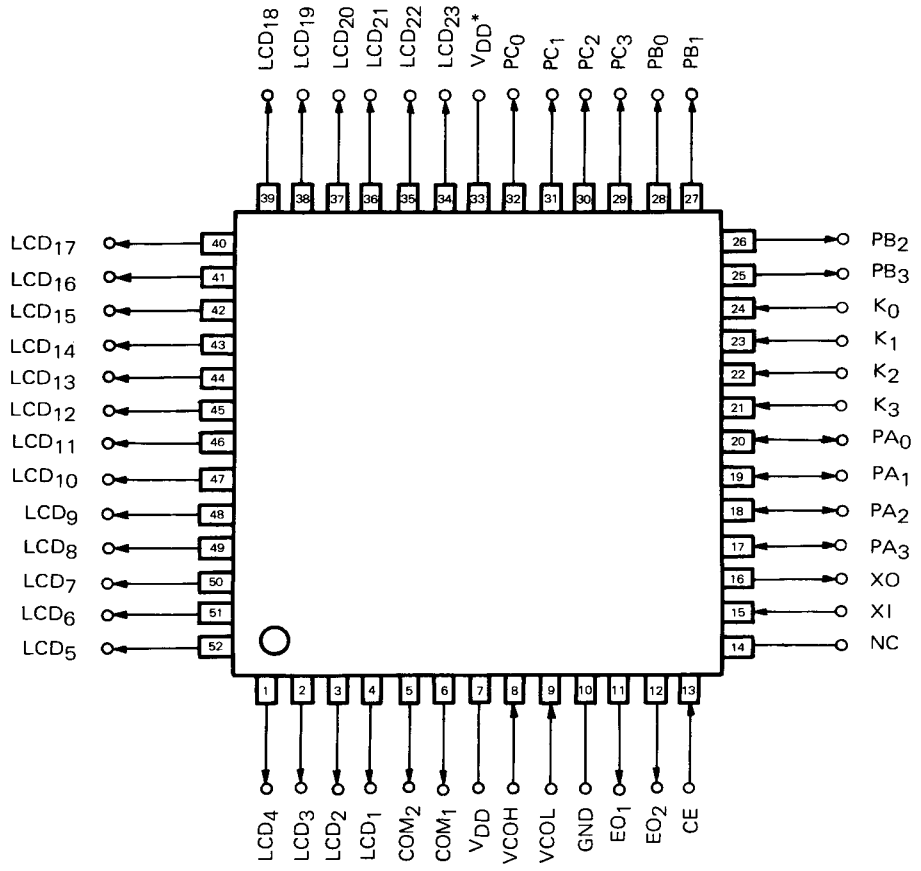
The μPD1708 is composed of a 52-pin flat package on which a number of input/output ports controllable by the powerful I/O instructions (IN and OUT) and input ports for the key switch are provided.

FEATURES

- A 4-bit microcomputer for digital tuning
- Built-in prescaler (150 MHz, MAX.), PLL frequency synthesizer and LCD driver
- Single power supply of 5 V±10 %
- Low power consumption CMOS
- Easy backup of data memory (RAM) (by CE pin)
- Program memory (ROM): 16 bits x 1,528 steps
- Data memory (RAM): 4 bits x 96 words
- A variety of powerful instruction sets (all of one-word instruction)
- Instruction execution time: 33.3 μs (with 4.5 MHz crystal)
- A rich set of addition & subtraction instructions (12 addition and 12 subtraction instructions)
- Power composite judgement instructions (TMT, TMF)
- Storage-to-storage data transfer at the same row address
- Indirect transfer between registers (MVRD and MVRS instructions)
- 16 powerful general registers (on RAM space)
- Single stack level
- Built-in LCD driver (1/2 duty, 1/2 bias driven, frame frequency; 100 Hz)
- Built-in PLA (Programmable Logic Array: User programmable) for display purpose (LCD pattern)
- Clock stop by instruction (CKSTP instruction; power current of 10 μA or less)
- 12 powerful I/O ports (PA_3 to PA_0: setting of input/output by one bit each, PB_3 to PB_0 and PC_3 to PC_0; for output only)
- Input ports for key input (K_3 to K_0)
- Superpower I/O instructions (IN, OUT)
- Test of status of input and output ports (TPT, TPF instructions)
- Built-in timer F/F (settable every 125 ms; easy timer setting)
- Built-in interval pulse output (internal output; test by pulse: TIP instructions every 5 ms (200 Hz, duty 60 %))
- Test of locked condition of PLL (TUL instruction)
- Transfer of data of rate and method of frequency division and reference frequency to PLL by single instruction (PLL instruction)

- Independent frequency input pin for AM and FM (maximum input frequency; 40 MHz (HF mode) for VCOL (AM) pin and 150 MHz (VHF mode) for VCOH (FM) pin)
- Two program-selectable method of frequency division, pulse-swallowing process and direct process (VHF, HF and MF mode).
- Two independent error out pins (EO_1 and EO_2 pins)
- Program selection of 7 different reference frequencies (1 kHz, 5 kHz, 6.25 kHz, 9 kHz, 10 kHz, 12.5 kHz and 25 kHz)

PIN CONFIGURATION (Top View)



NC: No Connection

* It is connected with 7 pin in the chip.

NOTE: In case of engineering sample (ceramic package), cap is connected with 7 pin, and 33 pin is NC.

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME	DESCRIPTION	OUTPUT TYPE
34 to 52 1 to 4	LCD_{23} to LCD_5 LCD_4 to LCD_1	LCD Segment Output	These are the output pins of segment signals to the LCD panel. A maximum of 46 dots can be displayed by the matrixes of COM_1 and COM_2, and the output to these pins can be made by the execution of LCDD instruction. Hence the content of data memory (RAM) at a given address as designated by the 1st operand of each LCDD instruction is output to the digit of LCD matrix as designated by the 2nd operand (see sect. 4, LCD driver). If an odd digit is designated, the content of data memory as designated by the 1st operand is loaded on the segment PLA (programmable logic array) and then output to these pins via PLA. The segment PLA is capable of generating 32 different kinds of patterns (see sect. 5, PLA). Note: The low level (display OFF mode) is automatically output when the power is turned on (V_{DD} = low to high) or when the CKSTP instruction is executed.	CMOS Push-pull
5 6	COM_2 COM_1	LCD Common Output	They are the output pins of common signals to the LCD panel. A maximum of 46 dots can be displayed by the matrixes of LCD_1 through LCD_{23}. Three different values of GND, $1/2V_{DD}$ and V_{DD} are output at the cycle of 50 Hz (at the intervals of 5 ms each). The segment where a potential difference of $\pm V_{DD}$ arises between any of these pins and any of LCD_1 through LCD_{23} is turned on (see sect. 4, LCD driver). Note: The low level (display OFF mode) is automatically output when the power is turned on (V_{DD} = low to high) or when the CKSTP instruction is executed.	CMOS Push-pull
7 33	V_{DD}	Power Supply	This is the device power pin, and it supplies the voltage of 5 V±10 % while the device is in operation. The voltage can be lowered to 2.5 V if to hold the internal data memory (RAM) (as the CKSTP instruction is being executed). When the voltage of 0 to 4.5 V is supplied to this pin, the device is reset and the program starts from the address 0 (see sect. 1.5, timer F/F). Note: No voltage needs be supplied to pins 7 nor 33 as both are connected inside the chip; both can be operated when either is voltage-supplied.	–
8	VCOH	VCO (High) Signal Input	This pin receives the local oscillation output (VCO output) from 10 to 150 MHz (0.5 V_{p-p} MIN.). Provided inside the chip are a 1/2 fixed frequency-divided prescaler and two-modulus prescalers of 1/32 and 1/33. To determine the frequency-dividing ratio of programmable divider, it must be based on the frequency bisecting the local oscillation output (see sect. 2.5, SETTING OF PLL INFORMATION). The VCOH pin is automatically pulled down (GND) when the direct frequency dividing mode is selected and the HF instruction is executed under the pulse swallowing mode, that is, when the VCOL pin is selected. The input to this pin must be first cut by the capacitor as the AC amplifier is provided inside (see sect. 2.4, PLL register).	Input

<table>
<tr><th>PIN NO.</th><th>SYMBOL</th><th>NAME</th><th>DESCRIPTION</th><th>OUTPUT TYPE</th></tr>
<tr><td>9</td><td>VCOL</td><td>VCO (Low) Signal Input</td><td>This pin receives the local oscillation outputs (VCO output) from 0.6 to 50 MHz (0.3 V_{p-p} minimum: see sect. 2.4, PLL register). This pin is chosen and turned active when the direct frequency dividing mode is selected and the HF instruction is executed under the pulse swallowing mode. Note that the suprenum of frequency, which can be input, differs between the above two frequency dividing modes.
<table>
<tr><th>FREQUENCY DIVIDING METHOD</th><th>INPUT VOLTAGE</th><th>INPUT FRE-QUENCY</th><th>DIVIDING RATIO</th></tr>
<tr><td>Direct method</td><td>0.1 V_{p-p} MIN.</td><td>0.59 to 20 MHz</td><td>16 to ($2^{12}-1$)</td></tr>
<tr><td rowspan="2">Pulse swallowing method (when HF instruction executed)</td><td>0.1 V_{p-p} MIN.</td><td>0.6 to 40 MHz</td><td rowspan="2">1024 to ($2^{17}-1$)</td></tr>
<tr><td>0.3 V_{p-p} MIN.</td><td>0.6 to 50 MHz</td></tr>
</table>
In case the VHF instruction is executed under the pulse swallowing mode, that is, when the VCOH pin is selected, the VCOL pin is automatically pulled down (GND). Note the input needs be first cut by the capacitor as the AC amplifier is provided inside.</td><td>Input</td></tr>
<tr><td>10</td><td>GND</td><td>Ground</td><td>The ground terminal of device</td><td>–</td></tr>
<tr><td>11
12</td><td>EO_1
EO_2</td><td>Error output</td><td>The error output pins of PLL; the high level is output from the pins when the divided frequency of local oscillation frequency (VCO output) is higher than the reference frequency or the low level if it is lower. The terminals turn to floating when both are equal. The outputs of pins are input into the external LPF (low-pass filter) and applied to the varactor diode via LPF. The selection of either pin is free to the user as both pins output the same waveform.</td><td>CMOS 3-state</td></tr>
<tr><td>13</td><td>CE</td><td>Chip enable</td><td>The input pin of device select signals. To operate the device normally, the high level needs be selected or the low level if it is not to be used. While this pin is held at the low level, PLL is inhibited or disabled. Any input less than 134 μs is not always acceptable. If the CKSTP instruction is executed while it is used by the program and the CE pin is at the low level (the instruction is valid only if CE = low or it works equal to the NOP instruction if CE = high), the operation of internal clock generator and CPU are stopped, thus holding the memory at a low current consumption (10 μA or less). The display outputs (LCD_1 to LCD_{23} and COM_1, COM_2) automatically turn to the display OFF mode (low level).
When CE pin is changed from the low to the high level, the device is reset and the program is started from address 0 (see sect. 1.5, Timer F/F). Also, the I/O port (Port A) is become input mode at this time.</td><td>Input</td></tr>
</table>

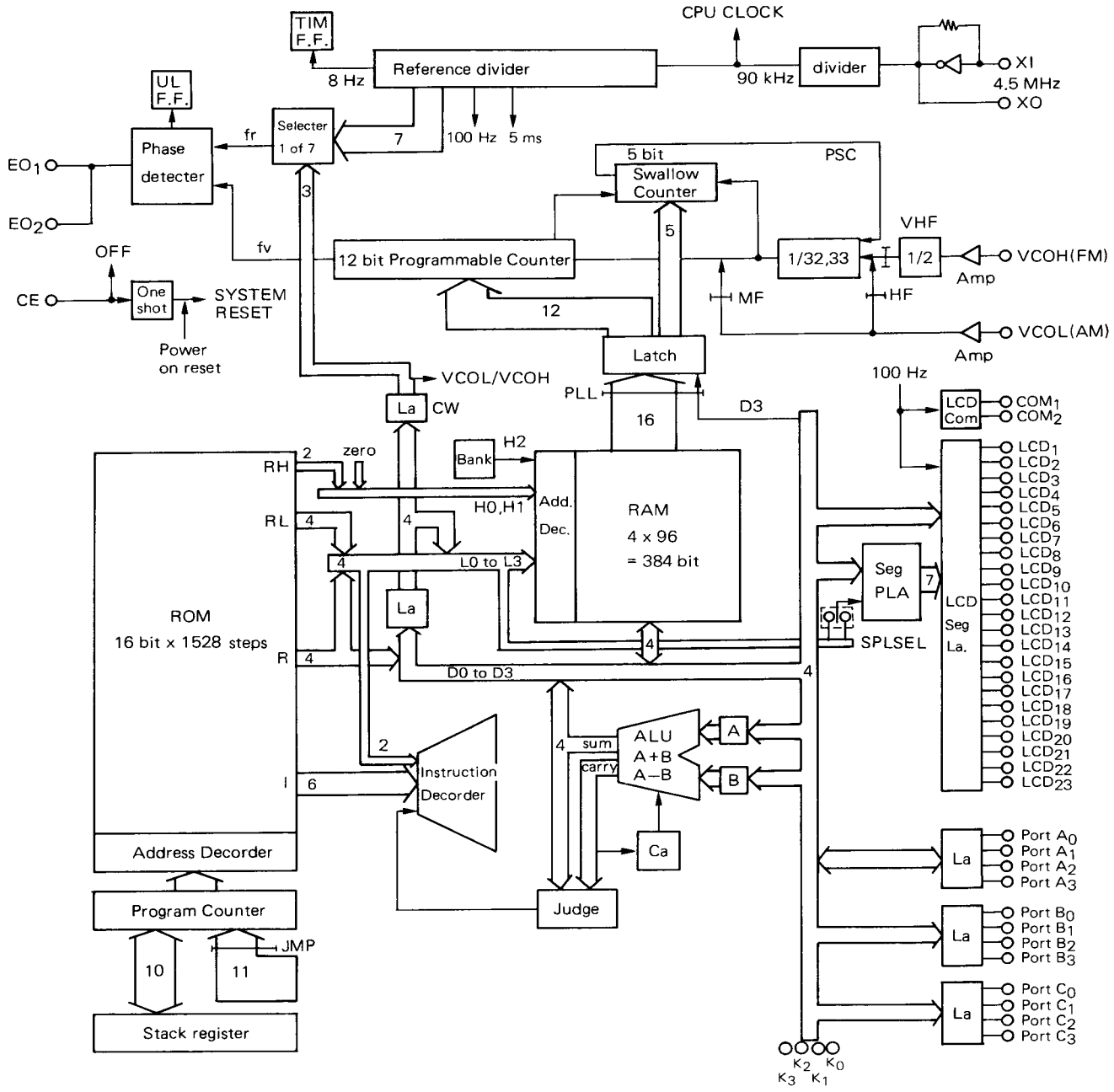
PIN NO.	SYMBOL	NAME	DESCRIPTION	OUTPUT TYPE
14	NC	Non-connection	This pin cannot be used as it is not connected to the inside chip. But it can be connected any of OPEN, GND or V_{DD}.	–
15 16	XI XO	X'tal	The connection pin of 4.5 MHz crystal oscillator. The oscillation frequency (4.5 MHz) must be adjusted while observing the XO terminal.	Input (XI) CMOS Push-pull(XO)
17 to 20	PA_3 to PA_0	Port A	These are the 4-bit I/O ports each of which allows to designate either the input or the output of 1 bit each, according to the content of data memory (RAM) at the address 1FH, known as the PAIO word. (see Notes 1 and 2 below)	CMOS Push-pull
21 to 24	K_3 to K_0	Key return signal inputs	These are the 4-bit input ports to be used for the input of key matrix. When either the KIN or KI instruction is executed, the status of these pins are read into the data memory (RAM) as designated by the instruction's operand. As the key return signal source, port B and the two higher-order bits (PC_3 and PC_2) of port C can be used.	Input
25 to 28	PB_3 to PB_0	Port B	These are the 4-bit output ports to be used as the key return signal source of key matrix as the absorbing currents are designed extremely low due to the device configuration. As any of these ports is used as the key return signal source, all the external diodes can be eliminated. Note however that the proper low output cannot be output, depending on the driving circuit, if they are used as ordinary output ports; then a pull-down resistor must be connected. (see Note 1 and 3 below).	CMOS Push-pull
29 to 32	PC_3 to PC_0	Port C	These are the 4-bit output ports out of which PC_3 and PC_2 can be used as the key return signal source of key matrix, similar to the port B, as their absorbing currents are designed extremely low. (Note 1 and 3)	CMOS Push-pull

Note 1: In execution of the port control instruction (IN, OUT, SPB or RPB instruction), PA_0 is set against the lowermost bit of register or operand data and PB_3 against the highermost bit. It goes to port B or port C as well.

Note 2: Port A (I/O port) turns to the input mode when the device is reset (V_{DD} changes from low to high and CE from low to high) or when the CKSTP instruction is executed.

Note 3: Unstable outputs are made to the output ports (ports B and C) when the power is turned on (V_{DD} = low to high) so that it must be initialized by the program. The content of data remains unchanged even if the CE pin changes from the low to the high level and the CKSTP instruction is executed. Then both ports must be initialized by the program when it seems necessary.

BLOCK DIAGRAM

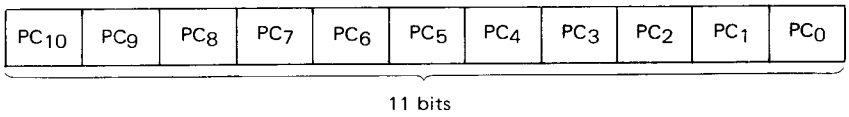


CONTENTS

1. CPU

1.1 PROGRAM COUNTER

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



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. When any skip instruction (e.g. ADS, TMT or RTS instruction) 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.

Note 1: The program counter of each of μPD1701, μPD1703, μPD1704, μPD1705, μPD1710 and μPD1711 (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 ten bits, there are two types of JMP instructions in μPD1708 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)

This register is composed of 1 x 11 bits and stores upon the execution of a subroutine call instruction the value adding one to the content of program counter or the return address (11 bits). The content of this stack register is loaded into the program counter by the execution of return instructions (RT, RTS) and returns into the main flow of program.

1.3 PROGRAM MEMORY (ROM)

ROM is composed of 1,528 steps of 16 bits each and stores the programs. The applicable range of ROM addressing is 1,528 steps starting with 000H and ending with 5F7H.

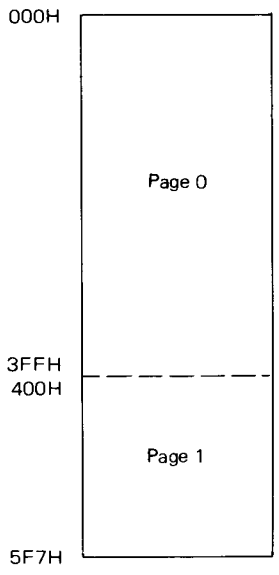


Fig. 1 ROM Configuration

This ROM of μPD1708 needs the concept of page, and its addresses from 000H to 3FFH are called the Page 0 and the remaining addresses from 400H to 5F7H the Page 1.

The head address of any subroutine must be set within the page 0 when creating a program. The subroutine contained in Page 1 at its head address however cannot be called neither from the Page 0 or the Page 1 (see Caution to Use of CAL Instruction).

The above concept of page is not applicable to the JMP instruction when it is described by the assembler, and it can be used between the addresses 000H and 5F7H under the same description (JMP ADDR).

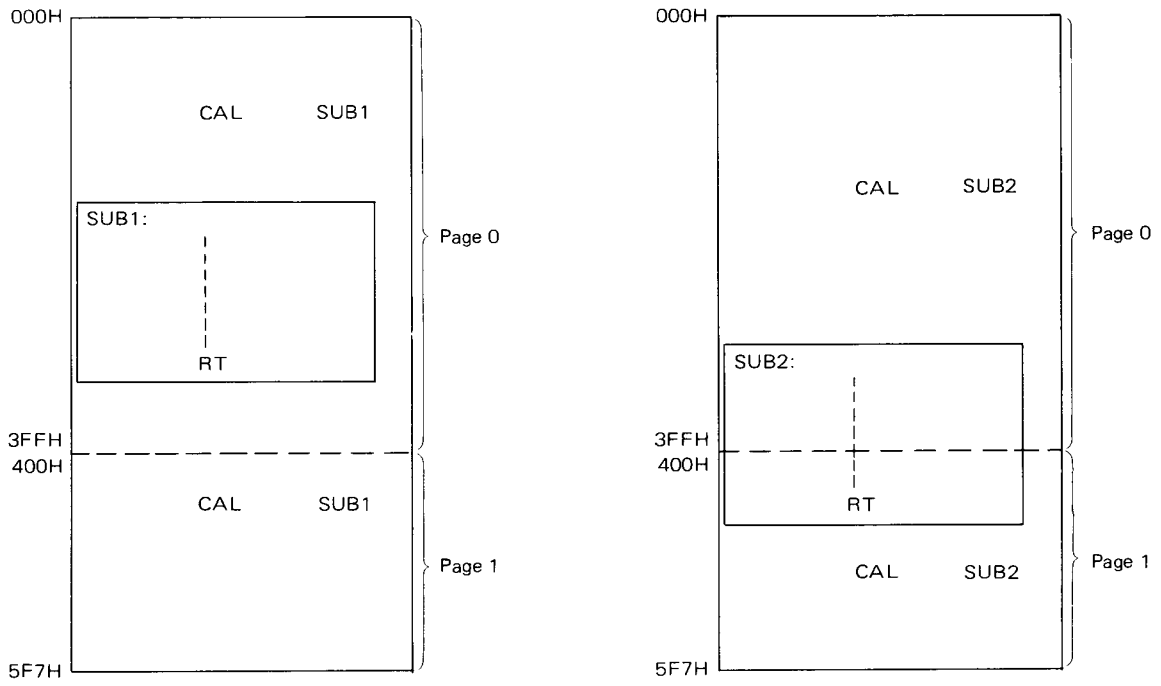
The operation codes of JMP instructions for Page 0 and Page 1 being different, care must be taken for the debug and patch correction (see Caution to Use of JMP Instruction).

The following points must be noted in the use of CAL and JMP instructions since ROM of μPD1708 applies the concept of page (discrimination between Page 0 and Page 1):

Caution to Use of CAL Instruction

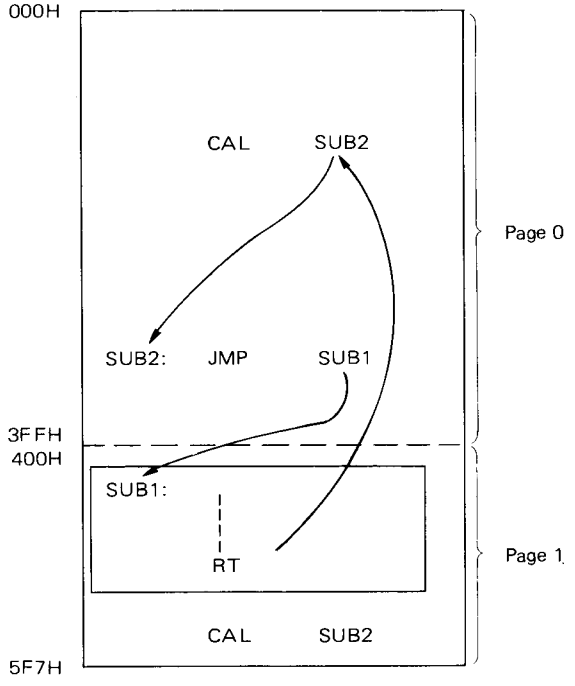
The calling address of CAL instruction or the head address of its subroutine must be set within the Page 0 (000H to 3FFH), but the subroutine at the head address of Page 1 (400H to 5F7H) cannot be called. The return address (RT and RTS instructions) can be set within the Page 1.

Example 1: Head address of subroutine within Page 0



If the head address of a given subroutine is set within the Page 0, as shown above, the return addresses (RT and RTS instructions) may be set within either the Page 0 or Page 1.

While the head addresses of subroutines are contained within the Page 0, the CAL instruction can be used without being conscious with the concept of page. The following technique is however effective if the head address of a given subroutine is not allowed to be within the Page 0 for the convenience of programming:



In other words, a JMP instruction is set in the Page 0 and a real subroutine (SUB1) is called via this JMP instruction.

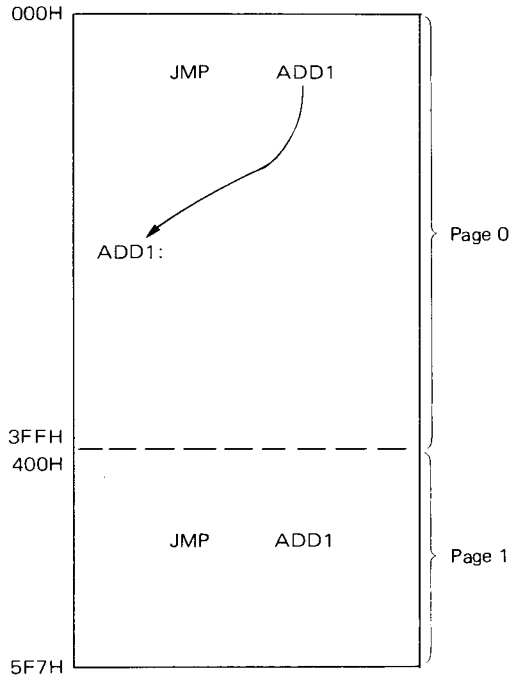
Caution to Use of JMP Instruction

As far as the JMP instructions are described by the assembler, the said instructions can be used without the concept of page between the ROM addresses 000H to 5F7H under the same description.

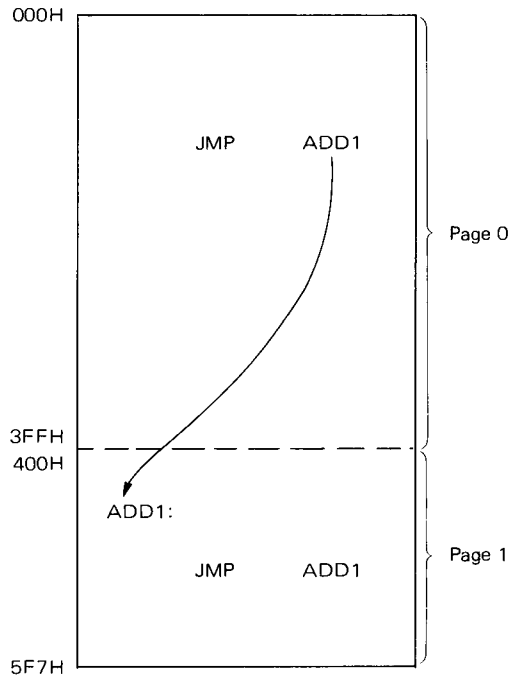
The operation codes of JMP instructions for Page 0 (000H to 3FFH) are different from those of other JMP instructions for Page 1 (400H to 5F7H). The operation code of JMP instruction for Page 0 is "06" while that of another JMP instruction for Page 1 is "02".

If the assembler of μPD1700 Series is used for assembling, the address to which the operation needs be jumped can be referred to and automatically converted by the assembler.

A Typical Case of Operation Code "06"
(if the jump address is within Page 0)

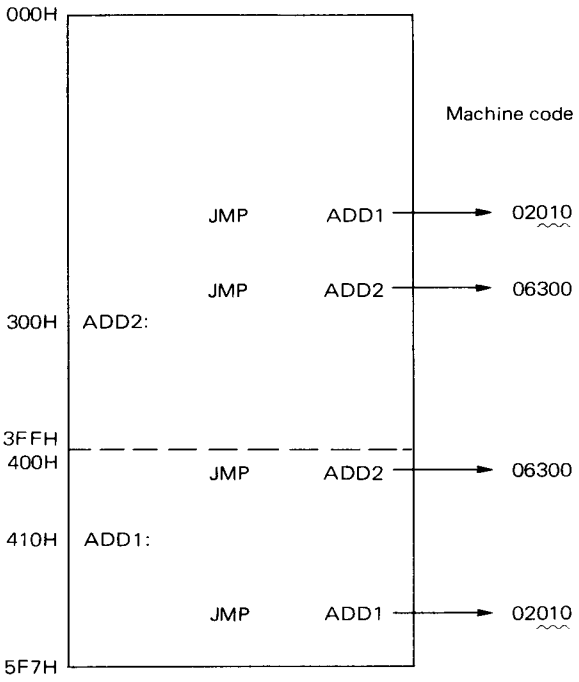


A Typical Case of Operation Code "02"
(if the jump address is within Page 1)



For the debug and patch correction the operation codes "06" and "02" must be converted by the programmer. And so must be the addresses if the JMP instruction designates any address beyond 400H (with the operation code of "02"). In this case the address 400H is turned to the address 000H and the remaining addresses are increased by one address each. Hence the address 5F7H is raised to the address 1F7H.

If JMP 400H is described by the assembler, for instance, and when you want to correct the patch, 02000 must be input so that JMP 000H turns to 06000.



1.4 DATA MEMORY (RAM)

RAM is composed of 4 bits x 96 words and can be used to store ordinary data. The area of 96 words of this RAM is divided into the BANK 0 (64 words) and the BANK 1 (32 words). Either BANK must be designated (by BANK0 or BANK1 instruction) before data are processed within the designated BANK.

The addresses from 00H to 0FH of BANK0 are called the general registers, which can be used for operations and transfers between the memory or which can be used as memories. No designations of BANKs are required to use the above addresses as the general registers, and they can be accessed under both status of BANK0 and BANK1. To use them as memories, these addresses should be designated in BANK0.

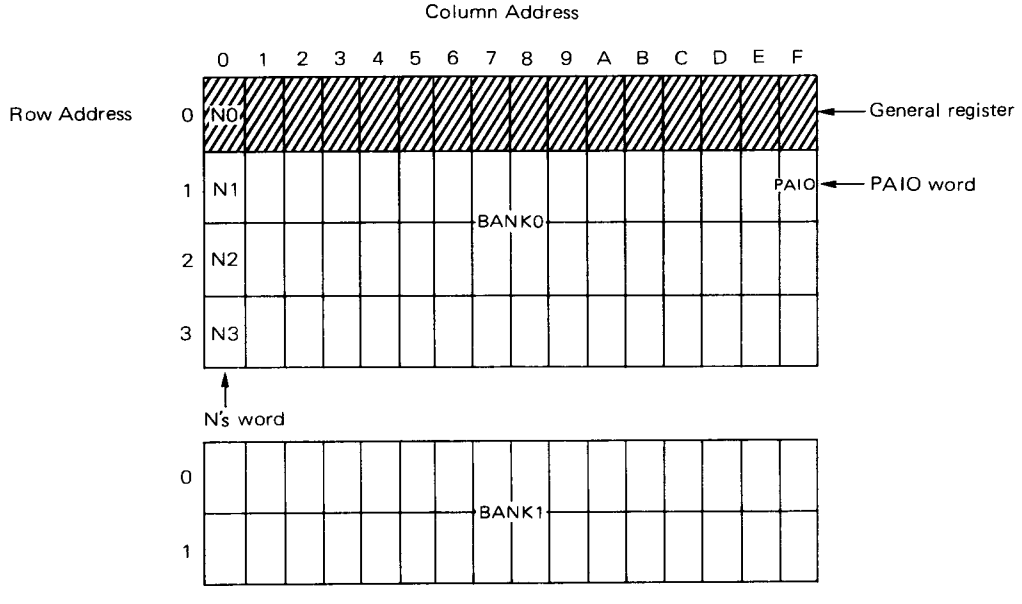


Fig. 2 RAM Configuration

All the pertinent information for PLL control (e.g. divided values, reference frequency or dividing method) can be set by RAM.

The bits to set a given divided value are assigned with a total of 17 bits composed of 4 bits x 4 words (N's words) at the address 00H, 10H, 20H and 30H and the highest-order bit (N_F bit) of a given general register; the bits to set both the reference frequency and the dividing method are assigned with one word (4 bits) of RAM, excepting N's words and one word including N_F bit. All these bits are to be transferred to the PLL register by the PLL 1 instruction.

The address 1FH of BANK0 is called the PAIO word to be used to designate the Port A for input or output.

Note: The most important point you should keep in mind in the general register operation in BANK1 is that **μPD1708 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 to count the clock (one second by 8 counts) or the mute time.

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.

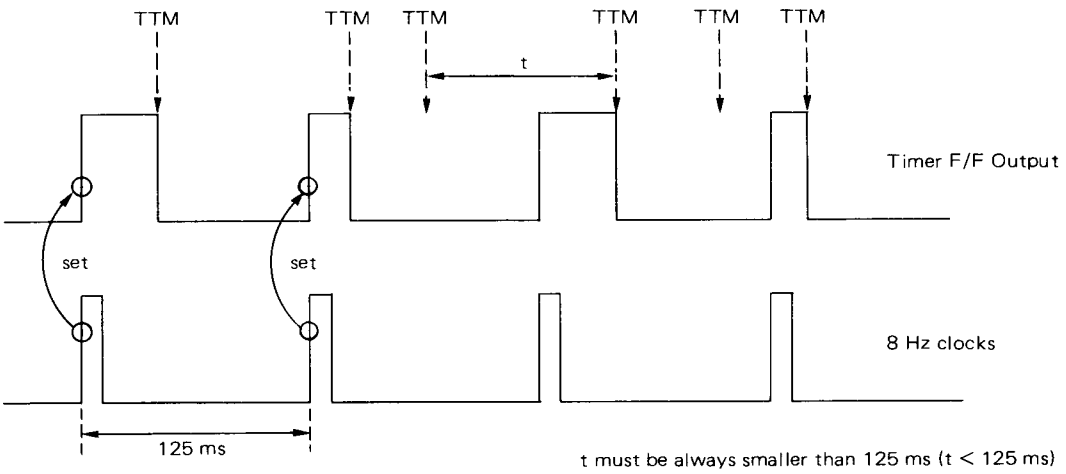


Fig. 3 Execution Timing of TTM Instruction

This timer F/F can be also used to judge the detection of power failure. It is reset when 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. Fig. 4 shows the status transition diagram illustrating the afore-mentioned relations.

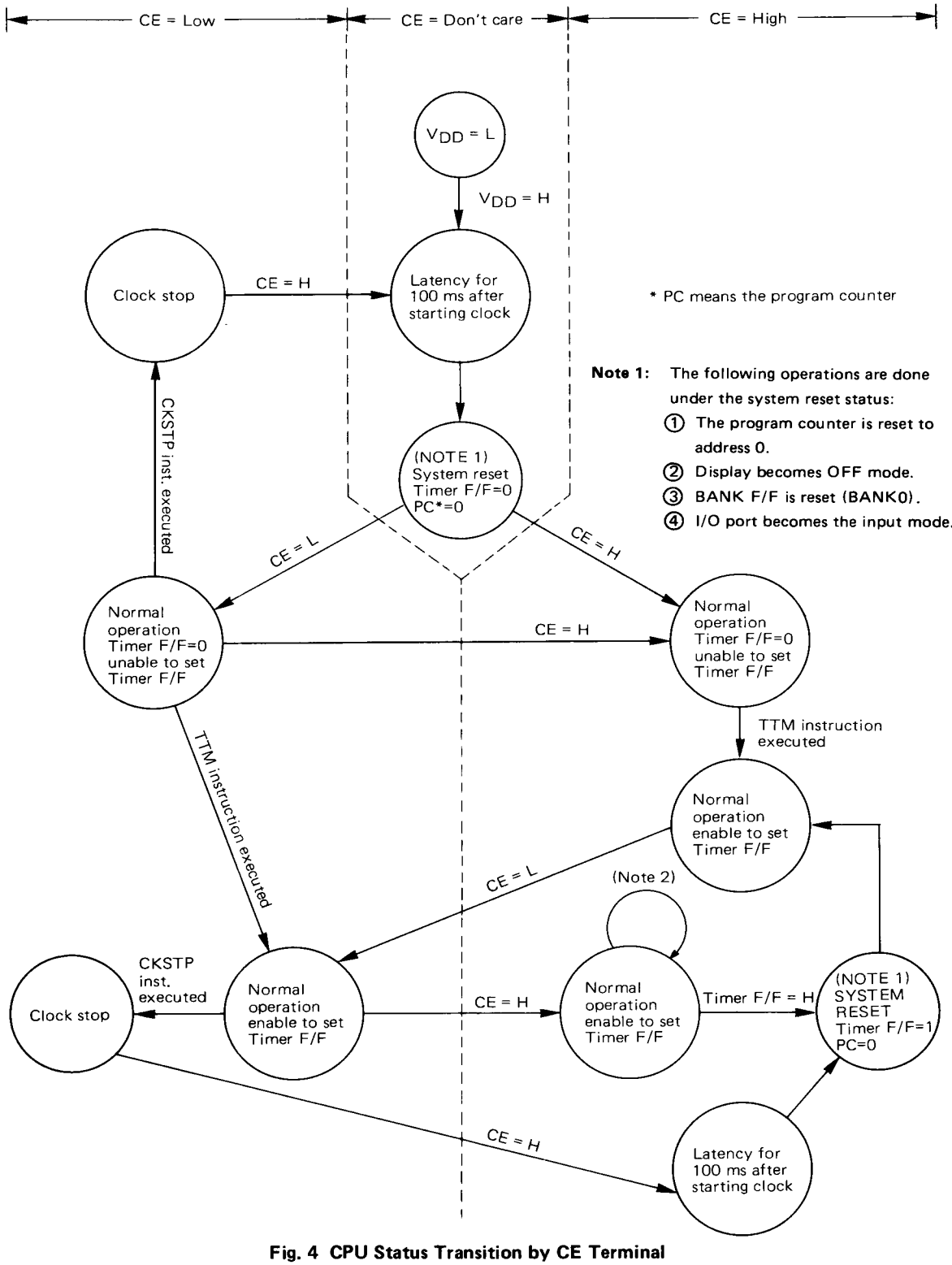


Fig. 4 CPU Status Transition by CE Terminal

Note 2: It is impossible to come out 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.

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 terminal changes from low to high, the program flow jumps to the address 0 immediately when the timer F/F is set. 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 program 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 program 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.

Note 1: 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 μPD1708. The timer F/F is reversely reset and the program starts from the address 0 under μPD1701, μPD1704 and μPD1710. It must be noted when executing the CKSTP instruction that the contents of timer F/F differ between μPD1708 and the group of μPD1701, μPD1704 and μPD1710.

Note 2: Even if the CE pin changes from low to high level, the program 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 pulses are 60 % duty pulses that are output at 5 ms each and can be tested by the TIP instruction. As no flipflops are provided, the pulse output cannot be reset even if the TIP instruction is executed.

A precise timer of the multiple of 5 ms can be created by the constant executions of TIP instruction to catch the edges of interval pulses.

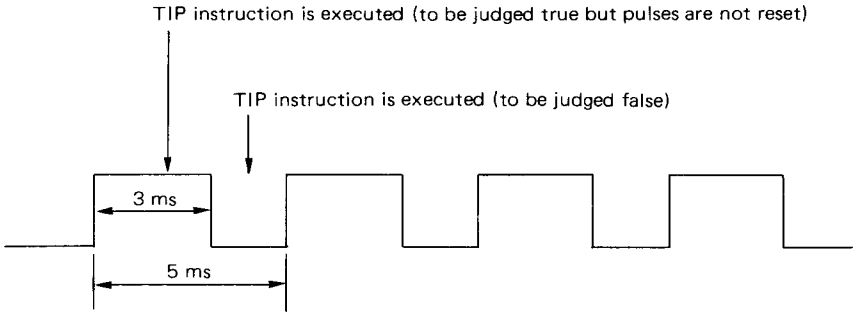


Fig. 5 Interval Pulse Timing

1.7 UNLOCK F/F (UL F/F)

The phase detector (ϕ-DET) outputs pulses at the cycle of reference frequency (f_r) if the PLL system is not locked or if the reference frequency (f_r) does coincides with the divided output frequency of VCO. The unlock F/F is set with this pulse and reset by the execution of TUL instruction. **The period of executing the TUL instruction should be always longer than that of f_r.** If it is shorter, the PLL system is considered to be locked, although it is not in fact, and it may lead to certain malfunction.

Similarly the TUL instruction, which is executed first after the PLL instruction, needs be executed after a period over the f_r period after executing the PLL instruction.

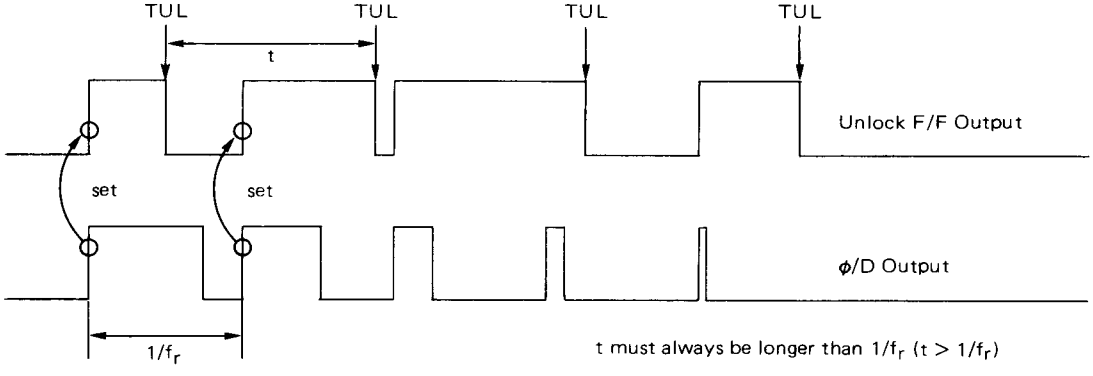


Fig. 6 Execution Timing of TUL Instruction

1.8 CARRY F/F (C F/F)

The carry F/F is set if a carry or a borrow is generated as the result of execution of a given operation instruction. Or it is reset if any of carry or borrow occured, and its content remains unchanged unless an operation instruction is executed.

The carry F/F can be also set or reset directly by the carry F/F set/reset instructions (STC and RSC instructions) as well as the status word operation instructions (SS and RS instructions).

1.9 BANK F/F (B F/F)

The BANK F/F is used to designate the BANK for the data memory (RAM and to address the port groups. RAM of 96 words is divided into the BANK0 (64 words) and BANK1 (32 words), and either BANK must be designated (execution of BANK0 or BANK1 instruction) before data are processed by either BANK. The data processing between two BANKs is carried out via the general registers (the address 00H to 0FH within BANK0). To use the above addresses of BANK0 as the general registers, and the data can be accessed from either the BANK0 or BANK1 without requiring the designation of BANKs. But the BANK0 must be designated in case the addresses are used as the memories.

The BANK F/F is also used for addressing of the port group. The addressing is then performed with two bits of the operands of a given instruction and the content of BANK F/F (see 3. Port).

The BANK F/F is reset and the BANK0 is automatically designated by the initial power input (V_{DD} = low to high) and CE changes from low to high or when the device is reset.

1.10 STATUS WORD

The status word is to split the inside status of a given device, which must be known for the execution of program or which must be designated unconditionally, into four bits each, and to thereby test, set or reset the status by the program.

Two kinds of the status words are provided, status word 1 and 2, to which any of the following terminals or F/F input or output is connected.

(1) Status Word 1 (write-only word)

Operation instruction: SS, RS, etc.

#3	#2	#1	#0
0	BANK F/F	Carry F/F	0

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

(2) Status Word 2 (read-only word)

Operation instruction: TST, TSF, etc.

#3	#2	#1	#0
0	BANK F/F	CE terminal	0

The contents 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., 5 kHz, 6.25 kHz, 9 kHz, 10 kHz, 12.5 kHz and 25 kHz, are derived by this generator, which divides the frequency of external crystal oscillator (4.5 MHz). The selection of required reference frequency can be made by the program (data of control word).

2.2 PHASE DETECTOR (φ-DET)

It is the circuit that detects the phase difference between the reference frequency (f_r) and those of the output of VCO, which are divided by the programmable divider.

The output is input into the internal charge pump, which in turn outputs the following pulses 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} means the oscillation frequency of VCO and N the dividing ratio of programmable divider.

2.3 PROGRAMMABLE DIVIDER (P/D)

The programmable divider is a binary down counter, composed of a swallow counter and a programmable counter. The swallow counter is a presettable down counter of 5 bits into which the contents of NR0 (4 bits) and N_F (1 bit) out of N registers are preset at the period of reference frequency.

The programmable counter is composed of 12 bits into which the contents of NR1 through NR3 of the N registers are preset and which is counted down simultaneously with the swallow counter.

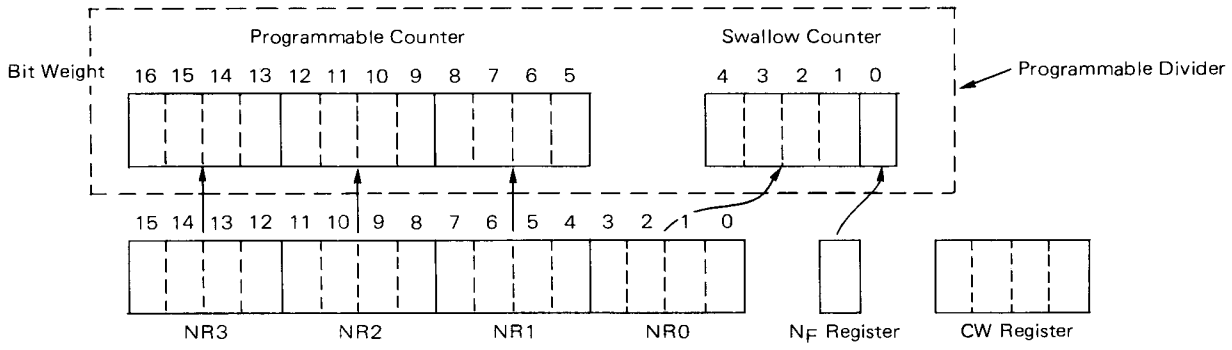


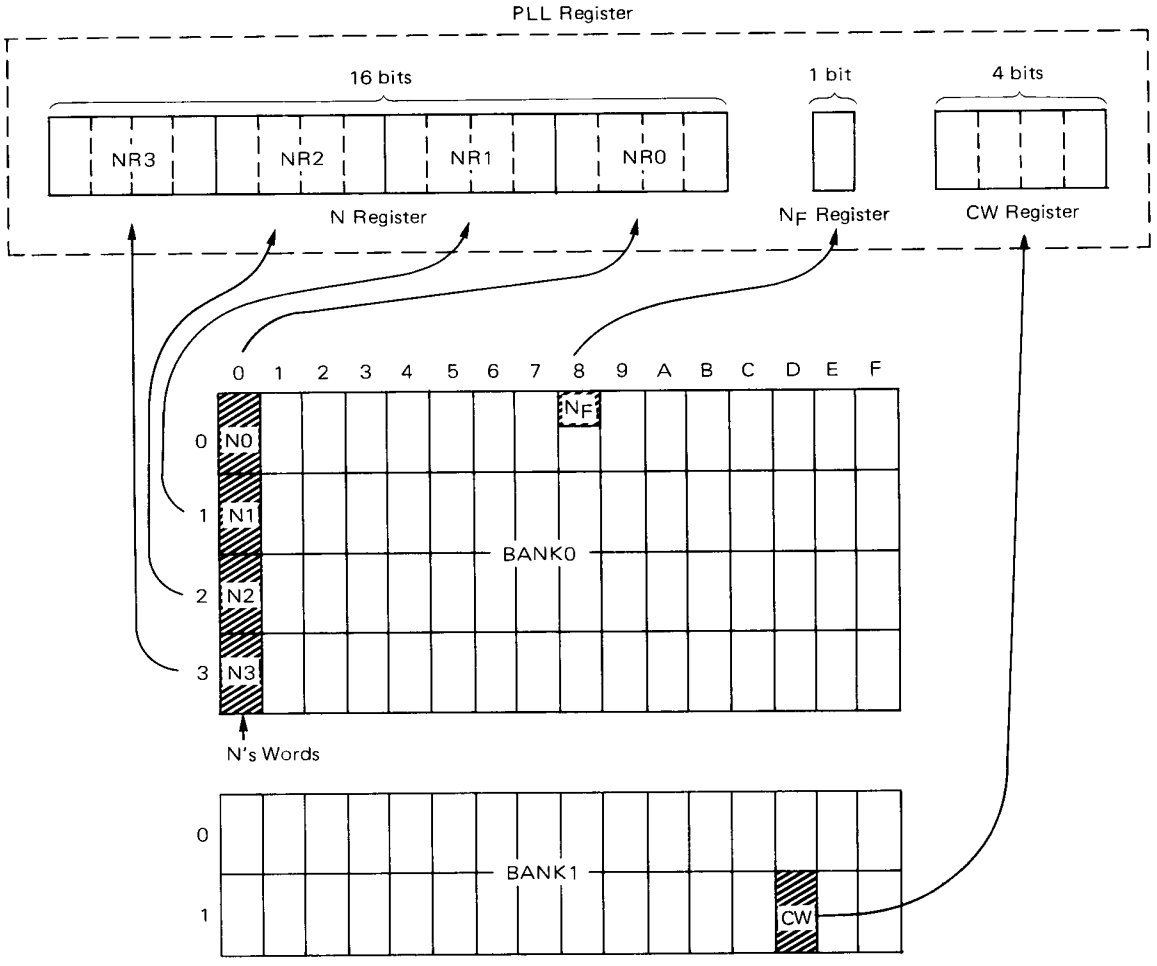
Fig. 7 Programmable Divider Configuration

2.4 PLL REGISTER

To control PLL of μPD1708 needs the following three information:

(1) Dividing ratio (N)
(2) Reference frequency (f_r)
(3) Dividing method (direct & 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.**



Note: Since CW is in BANK 1 in this example, the BANK must be set to "1" before the PLL instruction is executed.

Fig. 8 Operation by Execution of PLL Instruction

The N's words are assigned to the addresses 00H, 10H, 20H and 30H of RAM, the N_F bit to the highest-order bit of a general register and CW to any RAM area excluding the N's words and one word including the N_F bit.

The data code of control words, as shown in Table 1, enables you to select any of seven different reference frequencies. The highest-order bit (#3) of control word selects the direct dividing method if the bit is set to "0" or the pulse swallowing method if set to "1".

In case the direct method is chosen, the VCOL pin is also selected to which the frequencies ranging from 0.59 to 20 MHz (v_i=0.1 V_{p-p}) can be input. The then input frequency is directly divided according to the value of programmable divider.

Table 1 Control Word Codes

Control Word

#3	#2	#1	#0

0	Direct method
1	Pulse swallowing method

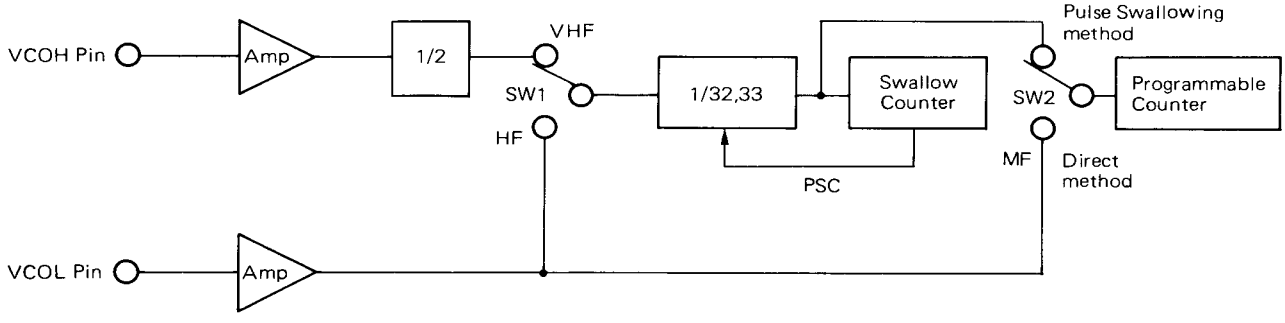
	#2	#1	#0	f_r (kHz)
	0	0	0	1
	0	0	1	6.25
	0	1	0	5
	0	1	1	9
	1	0	0	10
	1	0	1	12.5
	1	1	0	25
0	1	1	1	Disable PLL

If the pulse swallowing method is selected, the terminal that may be chosen by the execution of instruction varies with the VHF and HF instructions. **If then the pulse swallowing method was selected and the VHF instruction was executed,** the VCOH pin is chosen into which the frequencies from 10 to 130 MHz (v_i=0.3 V_{p-p}) can be input. These frequencies input into the VCOH pin are then divided by the 1/2 divider and sent to the programmable counter via two-modulus prescalers of 1/32 and 1/33.

If the pulse swallowing method was selected and the HF instruction was executed, the VCOL pin is chosen into which the frequencies from 0.6 to 40 MHz (v_i=0.1 V_{p-p}) can be input. The frequencies thus input into the VCOL pin are sent to the programmable counter via two-modulus prescalers of 1/32 and 1/33.

In any case where the pulse swallowing method is selected, either the VHF or HF instruction must be executed once again.

Fig. 9 shows the internal equivalent circuit, which explains the afore-mentioned operations.



Note: Both SW1 and SW2 in the above diagram remain unchanged when the power is turned on (V_{DD} changes from low to high).

Fig. 9 Internal Equivalent Circuit

In the above diagram SW1 is switched by the execution of VHF or HF instruction while SW2 is switched likewise by the execution of PLL instruction. Where the direct method was selected and either VHF or HF instruction was executed, the status of SW1 is switched by the instruction executed, but the PLL operation of direct method remains unaffected.

The direct method generally applies to the reception of MW and LW bands; the pulse swallowing method is then used for the reception of FM band (by VHF) or for the reception of SW band (by HF).

Table 2 Condition of VCOL and VCOH pin by dividing method

CW #3	FREQUENCY DIVIDING METHOD	CONDITION OF VCOL AND VCOH PIN	INPUT VOLTAGE	INPUT FREQUENCY	DIVIDING RATIO
0	Direct Method	VCOL pin = Active (VCOH pin = Pull-down)	0.1 V_{p-p} MIN.	0.59 to 20 MHz	16 to ($2^{12}-1$) (1 step)
1	Pulse Swallowing Method (When HF instruction executed)		0.1 V_{p-p} MIN.	0.6 to 40 MHz	1024 to ($2^{17}-1$) (1 step)
			0.3 V_{p-p} MIN.	0.6 to 50 MHz	
	Pulse Swallowing Method (When VHF instruction executed)	VCOH pin = Active (VCOL pin = Pull-down)	0.3 V_{p-p} MIN.	10 to 130 MHz	2048 to ($2^{17}-2$) (2 steps)
			0.5 V_{p-p} MIN.	10 to 150 MHz	

2.5 SETTING OF PLL INFORMATION

The PLL information (i.e. the dividing ratio, dividing method and reference frequencies) are set by the respective programs. Shown below is the setting of divided value of programmable divider.

(1) Direct Method

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

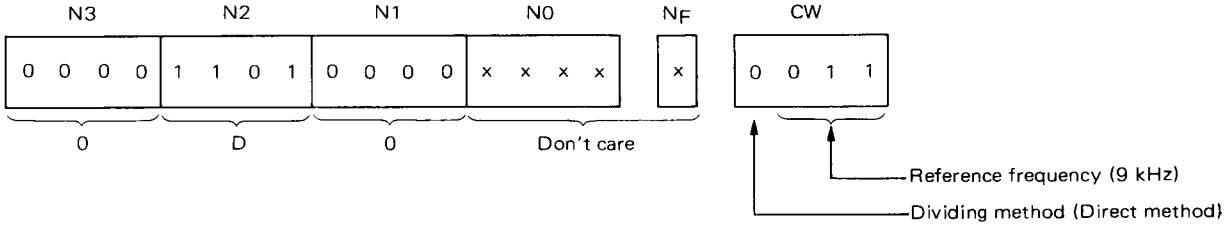
where f_{VCOL} = VCOL pin input frequency

f_r = Reference frequency

Example: To receive MW (the receiving frequency: 1,422 kHz, reference frequency: 9 kHz, IF frequency: 450 kHz)

$$N = \frac{1422 + 450}{9} = 208$$

= 0D0H (H means a hexadecimal code)



The contents of both N0 and N_F are ignored under the direct method.

(2) Pulse Swallowing Method (by execution of VHF instruction)

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

where f_{VCOH} = VCOH pin input frequency

f_r = Reference frequency

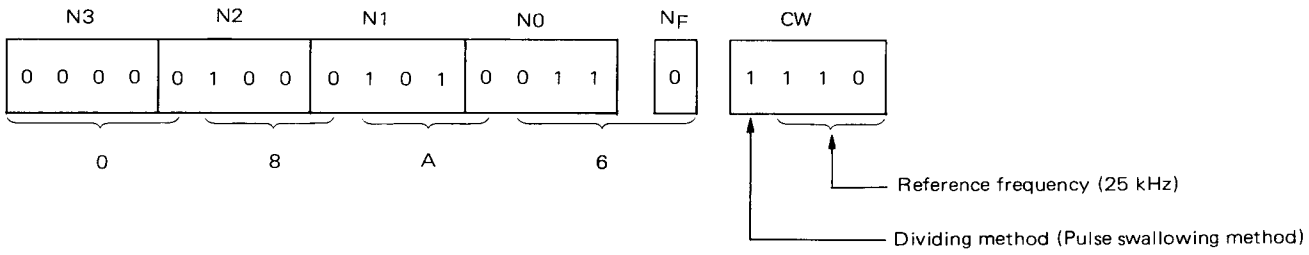
P = Divided value of 1/2 divider (=2)

Example: To receive FM (US band)

(the receiving frequency: 100.0 MHz, reference frequency: 25 kHz, IF frequency: 10.7 MHz)

$$N = \frac{(100.0 + 10.7) \times 10^6}{2 \times 25 \times 10^3} = 2214$$

= 8A6H (H means a hexadecimal code)

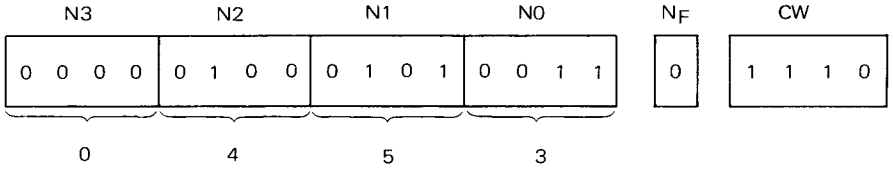


The above example is the case where the N_F bit is varied one by one as the lowest-order bit; then the VCO oscillation frequency changes by 50 kHz each. To change the VCO frequency by 100 kHz each, the bits must be varied by one each from N0 or by two each for 200 kHz change or by four each for 400 kHz change.

The value N is determined in the above example, assuming the N_F bit to be the lowest-order bit. It is however much easy to understand the programming when the bits are divided by four bits each from N0. The reference frequency is thus assumed to be 50 kHz in the computation.

$$N = \frac{(100.0 + 10.7) \times 10^6}{2 \times 50 \times 10^3} = 1107$$

= 453H



As shown above, the result of calculation becomes equal to the one where the reference frequency is assumed to be 25 kHz.

(3) Pulse Swallowing Method (by execution of HF instruction)

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

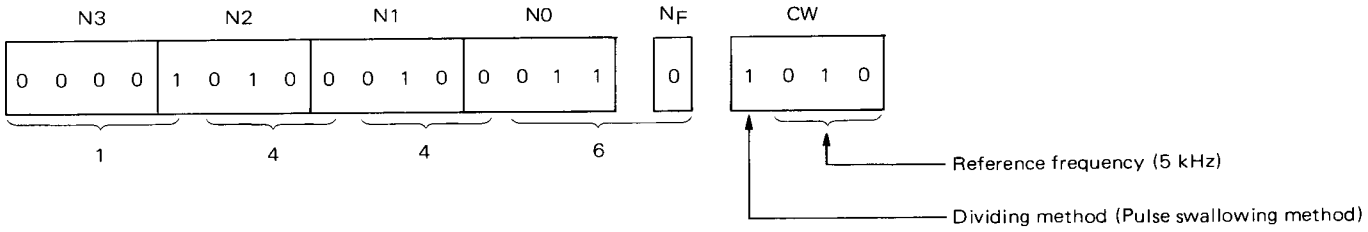
where f_{VCOL} = VCOL pin input frequency

f_r = Reference frequency

Example: To receive SW (the receiving frequency: 25.50 MHz, reference frequency: 5 kHz, IF frequency: 450 kHz)

$$N = \frac{25.5 \times 10^6 + 450 \times 10^3}{5 \times 10^3} = 5190$$

$$= 1446H$$



The above example is the case where the N_F bit is varied by one as the lowest-order bit; the VCO oscillation frequency then changes by 5 kHz each. To change it by 10 kHz each, the bits must be varied by one from N0 or by two for 20 kHz change or by four for 40 kHz change.

As apparent from the above three examples, the selection of pulse swallow method makes 17 bits valid from the N_F bit or the selection of direct method make 12 bits effective from the N1 word.

3. PORT

The μPD1708 has Port A (PA_3 to PA_0) as its I/O ports and Port B (PB_3 to PB_0) and Port C (PC_3 to PC_0) as its output ports.

The addressing of these ports is performed by the direct addressing of two bits contained in the operand of a given instruction and by the BANK F/F. The port addressings by these two elements are as shown in Table 3 below.

Table 3 Port Addresses

DIRECT ADD.		BANK F/F	
#1	#0	BANK0	BANK1
0	0	PA	–
0	1	PB	–
1	0	PC	–
1	1	–	–

Note: All the ports (Ports A, B and C) of μPD1708 are assigned within the BANK0. To access to a given port after accessing to RAM of BANK1, the BANK F/F must be set back to BANK0. No ports can be accessed while the BANK1 is chosen by the BANK F/F even if any port operation instruction (e.g. IN, OUT, SPB or RPB instruction) is executed.

3.1 PORT A

The Port A (PA_3 to PA_0) can be set either to the input or the output by the unit of 1 bit each. The input or output is settable according to the content at the address 1FH of BANK0 within the data memory (RAM) known as the PAIO word. To set the input port, "0" needs be set to the bit of PAIO word corresponding to the required port or "1" must be set to set any Port A to the output.

	#3	#2	#1	#0
PAIO Word (Address 1FH) BANK0	PA_3	PA_2	PA_1	PA_0

Example 1: To set PA_3 and PA_2 to the output and PA_1 and PA_0 to the input

	#3	#2	#1	#0
PAIO Word	1	1	0	0

Example 2: To set all Port A ports (PA_3 to PA_0) to the output

	#3	#2	#1	#0
PAIO Word	1	1	1	1

Either the input or the output must be set in the PAIO word for Port A; then the input or output instruction should be executed. The input mode or output mode once set up remains valid until after the content of PAIO word (data stored at the address 1FH of BANK 0) is changed. Port A however is changed automatically to the input mode when the power is turned on (V_{DD} changes to low to high) or after the execution of CKSTP instruction or if CE changes from low to high.

It must be noted that the contents of PAIO word do not coincide with the current mode of Port A, viz., the input or output mode. Port A works under the input mode continuously until the content of PAIO word is set.

Example:

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

3.2 PORT B & PORT C

Both the Port B (PB_3 to PB_0) and Port C (PC_3 to PC_0) of μPD1708 are the output ports of CMOS type, which can be generally accessed by the output instruction (e.g. OUT, SPB and RPB instructions). When an input instruction (IN) is executed, the content of data currently being output is read into the register as designated by the operand of the said instruction. The contents being output by the execution of IN instruction may remain unchanged. If "1" is output during the execution of an output instruction, the high level (V_{DD} potential) is output or the low level (GND potential) is output if "0" is output.

Note 1: When the power is turned on (V_{DD} changes from low to high), the contents of output to be made by Port B or Port C are unknown. It is therefore imperative to initialize both ports by the respective program after turning on the power.

Note 2: Where V_{DD} is set high, CE pin changes from high to low or it changes reversely, the contents of outputs of Port B and Port C do not change and the preceding status are retained. The output contents also remain unchanged even after the CKSTP instruction was executed.

Example 1: Port initialization before turning on the power

```
START:
        MVI   1FH, 1111B  ;  to set all bits of Port A to output port
        MVI   0AH, 0      ;  to set the port initialize data
        OUT   0, 0AH      ;  Port A (PA3 to PA0) = All Low
        OUT   1, 0AH      ;  Port B (PB3 to PB0) = All Low
        OUT   2, 0AH      ;  Port C (PC3 to PC0) = All Low
        TTM               ;  if the timer F/F is set,
        JMP   BACKUP      ;  then go to BACKUP
        MVI   00H, 0      ;  RAM initialize
        MVI   01H, 0      ;  RAM initialize
          ⋮
BACKUP:
```

Example 2: In the μPD1708, Ports B and C maintain the state immediately before even when 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 CE pin is low more than 100 μs before ① .
② TCEF              ; Skips if the level at CE pin is low more than 133.3 μs before ② .
③ JMP    NOTSTP     ; To NOTSTP (does not CKSTP)
                    ; if the level at CE pin 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 C entirely
⑤ CKSTP             ; Synchronize to 8 Hz and branch to Address 0 if the level of CE pin 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 rise.
                      If the level of CE pin 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 CE pin. 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 CE pin is low more than 133.3 μs (four instruction cycles) before this instruction is given.

4. LCD DRIVER

μPD1708 contains in it an LCD driver (frame frequency: 100 Hz) of 1/2 duty and 1/2 bias drive (voltage equalization system) type. Fig. 10 is a timing chart which illustrates the principle of the LCD driver. As can be seen from Fig. 10, two common 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 common signals output a potential of ±1/2 V_{DD} on both sides 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 common 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.

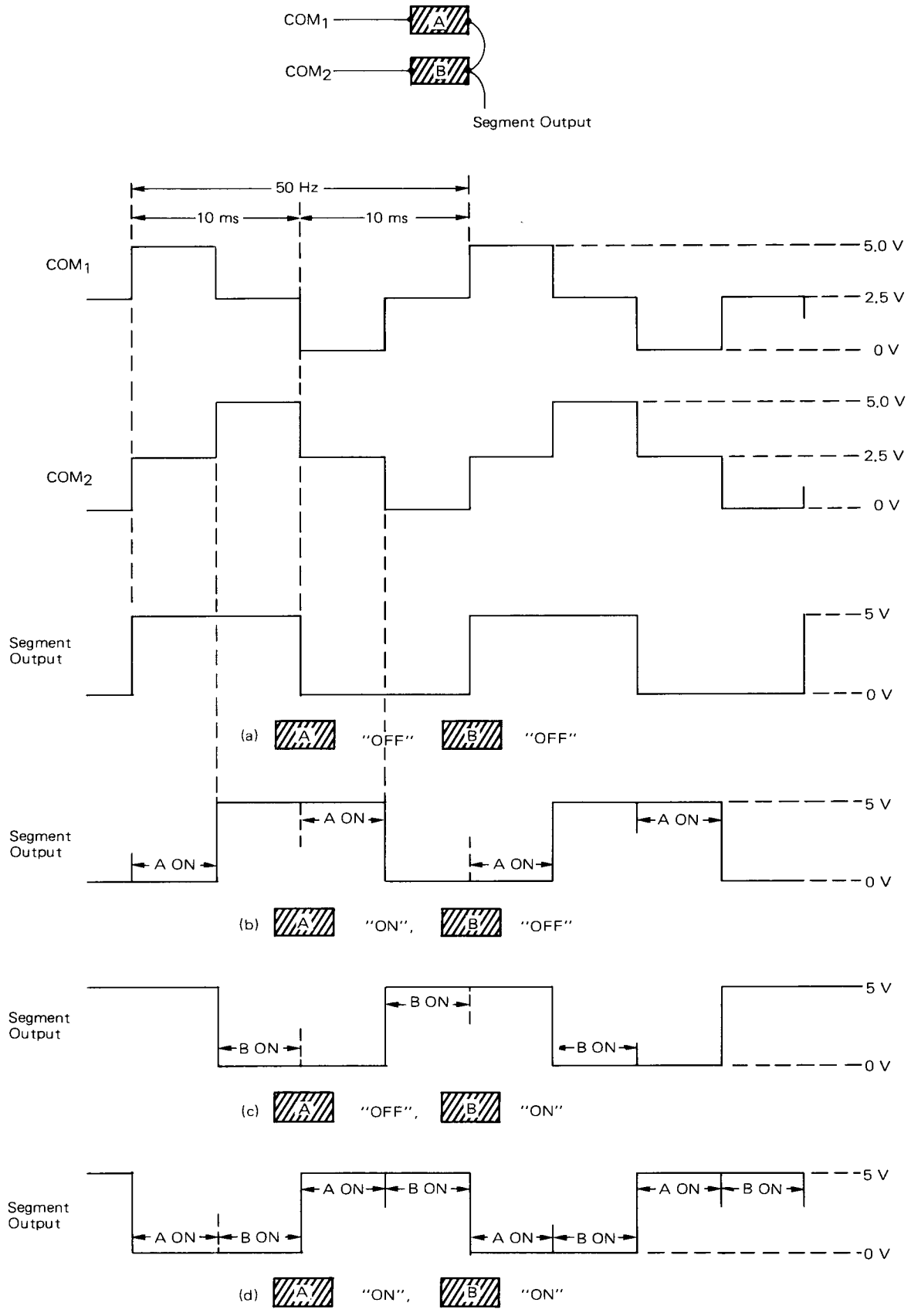


Fig. 10 Timing Chart of 1/2 Duty and 1/2 Bias (Voltage Equalization System) LCD Driver

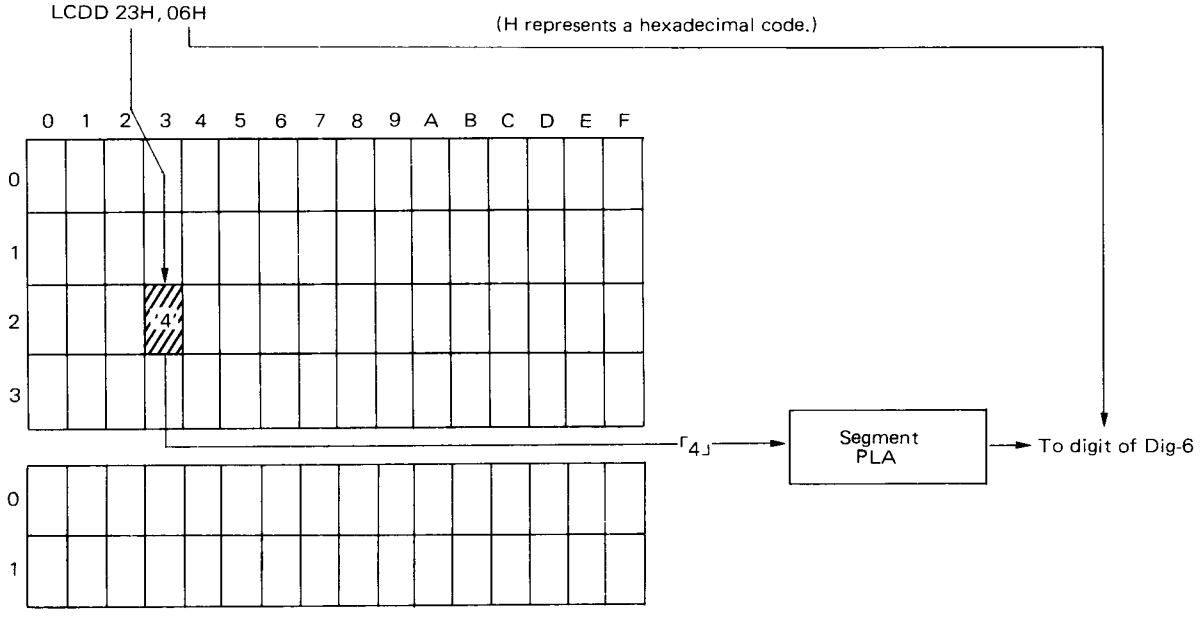
μPD1708 is able to make an outer LCD panel display data by executing an instruction LCDD M, D. "M" represents any address in a data memory (RAM), and "D" represents the number of digits of display. "D" takes a total of fifteen (15) values, i.e., fourteen (14) values 00H to 0DH of Dig-0 to Dig-D as shown in an LCD matrix of Fig. 11 and a special value 0FH which determines ON and OFF modes of display.

When even digits (Dig-0, Dig-2, Dig-4, Dig-7, Dig-8, Dig-A and Dig-C) are entered in "D", data stored in the data memory (RAM) designated by "M" are unconditionally loaded on a PLA (Programmable Logic Array), and output (displayed) in a digit designated by "D" through the PLA.

When odd digits (Dig-1, Dig-3, Dig-5, Dig-7, Dig-9, Dig-B and Dig-D) are entered in "D", data stored in the data memory (RAM) is output (displayed) directly in a digit designated by "D", without passing through the PLA. Table 4 shows relations between bits of RAM data to be output in an odd digit at this time and segments arranged in odd digits.

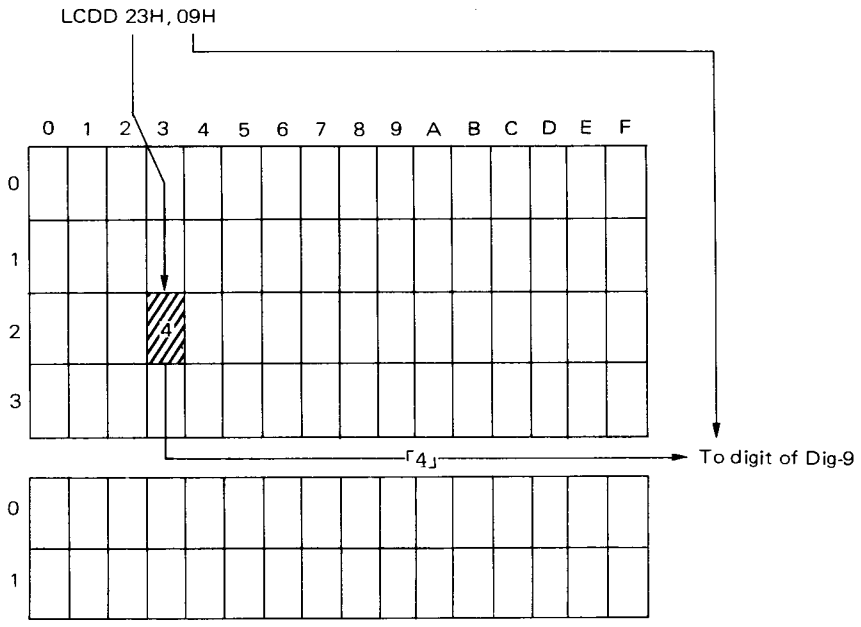
Examples of operations when executing an LCDD instructions in even and odd digits are described the following.

○ Example in Case of Even Digit



The data "4" stored in the address 23H of the data memory (RAM) is led to the PLA, and data stored in the PLA is output to the digit of Dig-6.

○ Example in Case of Odd Digit



The data "4" stored in the address 23H of the data memory is output directly to a digit of Dig-9. Four segments b_A, e_A, f_A and g_A are arranged in the digit of Dig-9, and when "4" = 0100B is output, only the segment f_A lights up. (See Table 4 "Bits Corresponding to Odd Digit (Which Does not Pass Through PLA)). When "5" = 0101B is output to the digit of Dig-9, segments f_A and e_A light up.

As can be seen from the LCD matrix of Fig. 11, the digit of Dig-8 (even digit) can be used also as a digit of Dig-D (odd digit). If, for instance, LCDD 10H, 08H is entered, data of the digit of Dig-8 is displayed through the PLA. If LCDD 10H, 0DH is entered, data is displayed as a digit of Dig-D without passing through the PLA.

Similarly, the digits of Dig-5 and Dig-7 (odd digits) can be used also as a digit of Dig-C (even digit), and those of Dig-9 and Dig-B (odd digits) can be used also as a digit of Dig-A (even digit). In other words, it can be selected whether segments of these digits light up through the PLA or not through the PLA.

When power is turned on (V_{DD} = Low → High) and an instruction CKSTP is executed, the levels at LCD segment terminals (LCD_1 to LCD_{23}) and LCD common terminals (COM_1 and COM_2) all automatically become low (**display OFF mode**). In the display OFF mode, therefore, all segments on the LCD panel are put out.

Even though an instruction LCDD M, D (provided that $0 \leqq D \leqq 0DH$) is executed in the display OFF mode, the display is not turned on. If an LCDD instruction is executed in this case, only such operation that the latch data of the LCD segment latch circuit is rewritten is made.

It is, therefore, necessary to set the LCD segment and common terminals once in the **display ON mode** for making the display on the LCD panel light up. When changing the operation mode from display OFF to display ON, the following instruction is executed.

```
LCDD  M, 0FH
```

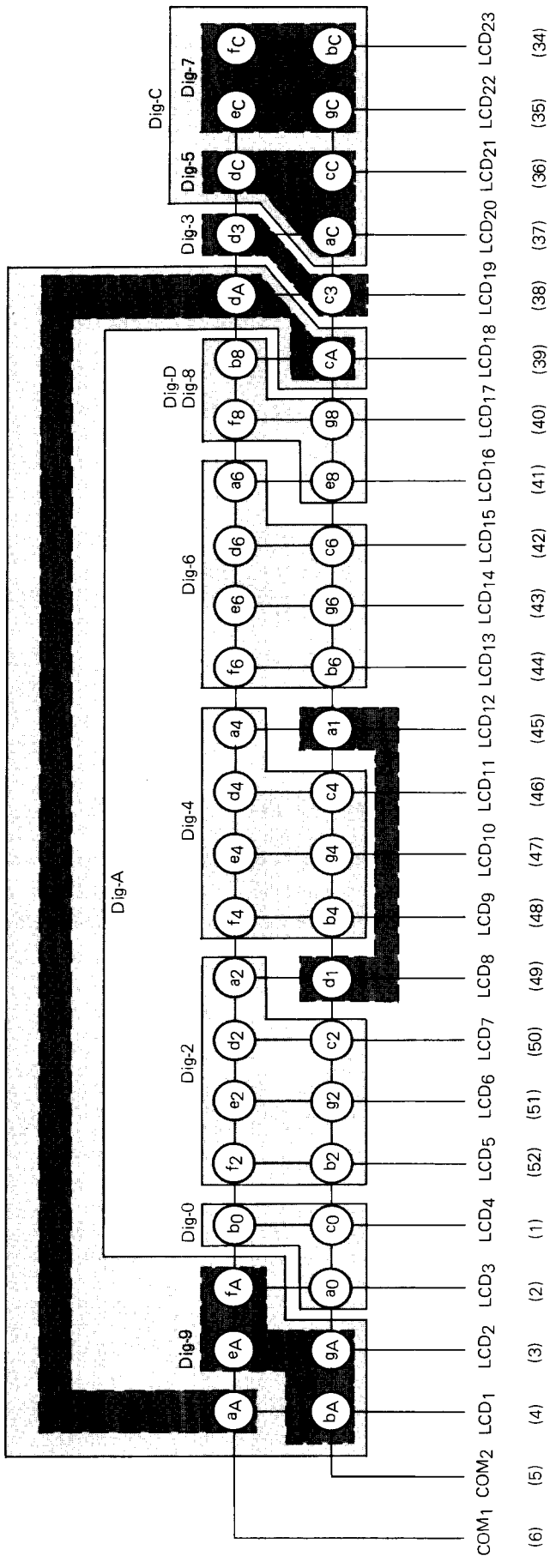
However, data to be stored in a data memory (RAM) addressed by "M" is larger than 8. That is, 0FH is entered in the second operand part (D) of the instruction LCDD, and a value larger than 8 ($8 \leqq (M) \leqq F$) is set for data to be stored in the data memory (RAM). In other words, a value larger than 8 should be output to the digit of Dig-F.

On the contrary, when changing a display ON mode to a display OFF one with a program, a value of smaller than 7 ($0 \leqq (M) \leqq 7$) is output to the digit of Dig-F. When a display mode is changed to display OFF, latch data of the LCD segment latch circuit does not change from the preceding data (data stored in the display ON mode), unless an instruction LCDD M, 0 (provided that $0 \leqq D \leqq 0DH$) is executed. When, therefore, preparing such a program as to make every segment of the LCD panel flash at fixed time intervals, the purpose is attained by alternately and repeatedly executing an instruction LCDD which sets a display mode to display ON or display OFF as mentioned above. Examples of programs to be prepared are given below.

```
        ⋮
        LCDD 12H, 0DH   ;  Output to digit of Dig-D.
        LCDD 11H, 0CH   ;  Output to digit of Dig-C.
        MVI  CONT, 8    ;  Sets 5-second timer.
        MVI  34H, 0     ;  Sets display OFF mode data.
FLASH:
        LCDD 34H, 0FH   ;  Display OFF mode (every segment OFF).
        CAL  WT500M     ;  Calls subroutine of waiting for 500 ms.
        AI   34H, 8     ;  Turn over the display.
        SIS  CONT, 1    ;  Skips after five times, i.e., five seconds.
        JMP  FLASH      ;  Returns to FLASH again if five seconds have not elapsed.
        ⋮
```

The above examples are programs of making every segment, i.e., whole of the LCD panel flash (OFF for 500 ms and ON for 500 ms) for five seconds at 500 ms intervals, after displaying data in all digits of the LCD panel.

Note: Contents of data to be applied to the LCD segment latch circuit when turning power on (V_{DD} = Low → High) is indefinite. If, therefore, a display mode is changed to display ON immediately after turning power on, an undesired display may be made. In such cases, it is necessary to beforehand output data, those are desired to be displayed before changing a display mode to display ON, i.e., in a display OFF mode, to all digits.



Figures given in parentheses represent terminal numbers of IC.

Fig. 11 LCD Matrix

Table 4 Bits Corresponding to Odd Digit (Which Does not Pass Through PLA)

	#3	#2	#1	#0
Segment which lights up.		c	d	a
	b	f	g	e

← Bit of data memory (RAM)

5. PLA (PROGRAMMABLE LOGIC ARRAY)

μPD1708 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 4 "LCD DRIVER").

5.1 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.

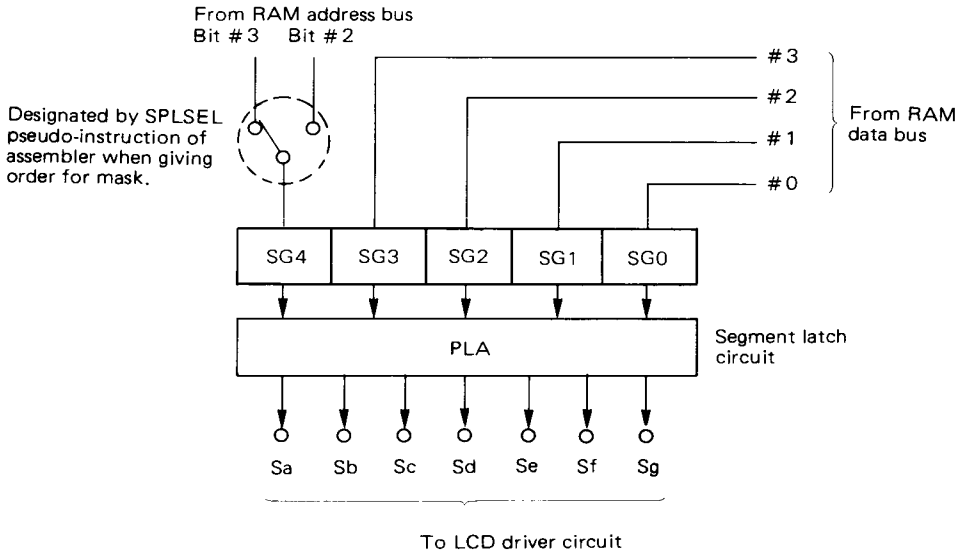
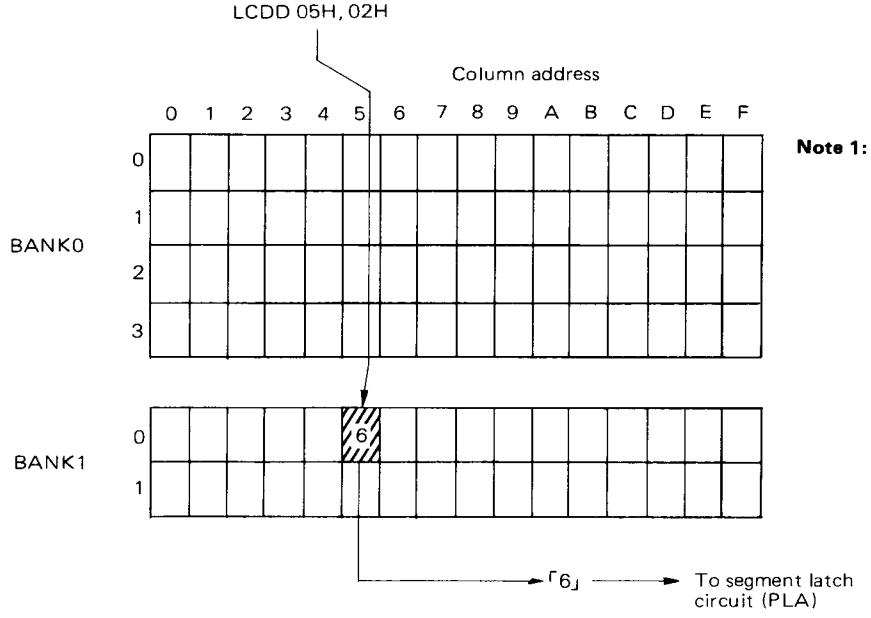


Fig. 12 Composition of Segment PLA

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), i.e., "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 the examples of programs of PLA.)

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.

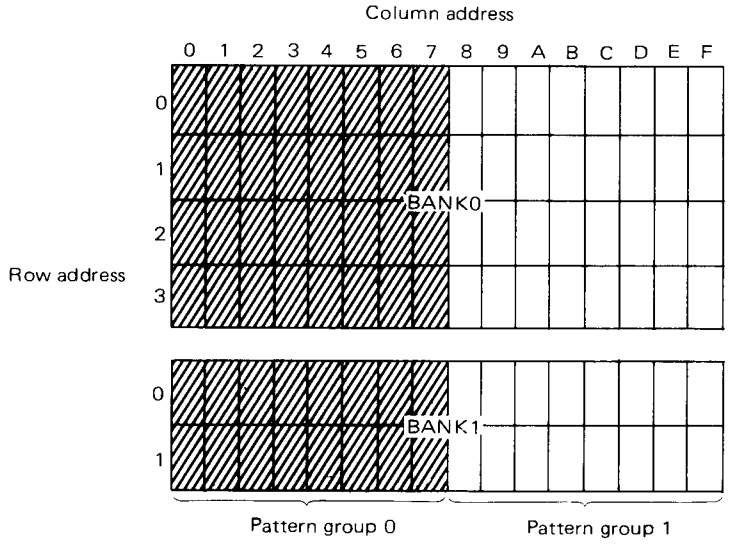


Fig. 13 Example of Dividing Patterns into Groups When SG4 is Set to Bit #3

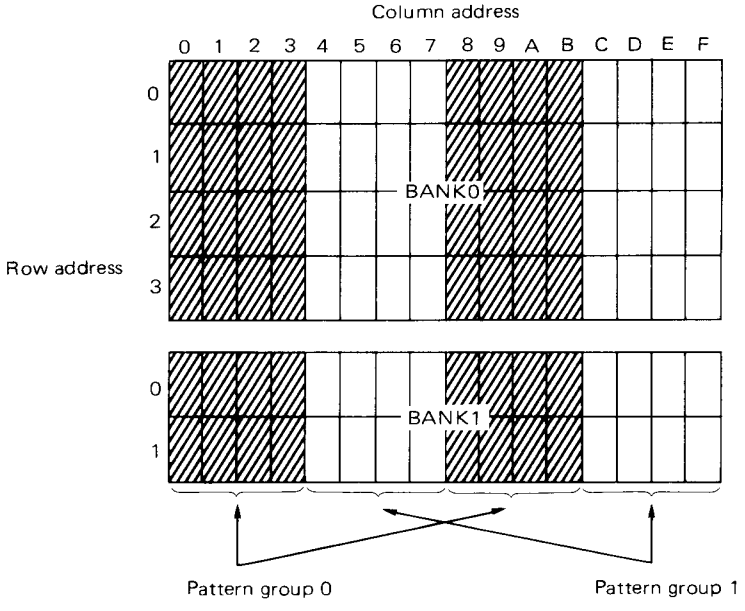


Fig. 14 Example of Dividing Patterns into Groups When SG4 is Set to Bit #2

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 the examples of PLA programs on page 41).

5.2 EXAMPLE OF PATTERNS OF SEGMENT PLA

Examples of patterns of the pattern group 0 and 1 are given in the following Tables 5 and 6, respectively.

Table 5 Example of Patterns of Pattern Group 0

SEGMENT LATCH					SEGMENT OUTPUT							OUTPUT PATTERN
SG4	SG3	SG2	SG1	SG0	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	E
	1	1	0	0	0	1	1	1	0	0	1	F
	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 6 Example of Patterns of Pattern Group 1

SEGMENT LATCH					SEGMENT OUTPUT							OUTPUT PATTERN
SG4	SG3	SG2	SG1	SG0	g	f	e	d	c	b	a	
1	0	0	0	0	1	0	1	0	0	1	0	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	FM, 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	5
	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

5.3 EXAMPLE OF DISPLAY PROGRAM

An example of programs of displaying an information on the LCD panel shown in Fig. 15 using PLA patterns given in Tables 5 and 6 is shown below. This example is based on the assumption that bit #2 (SPLSEL2) of a column address is designated for an input of SG4 of the segment latch circuit.

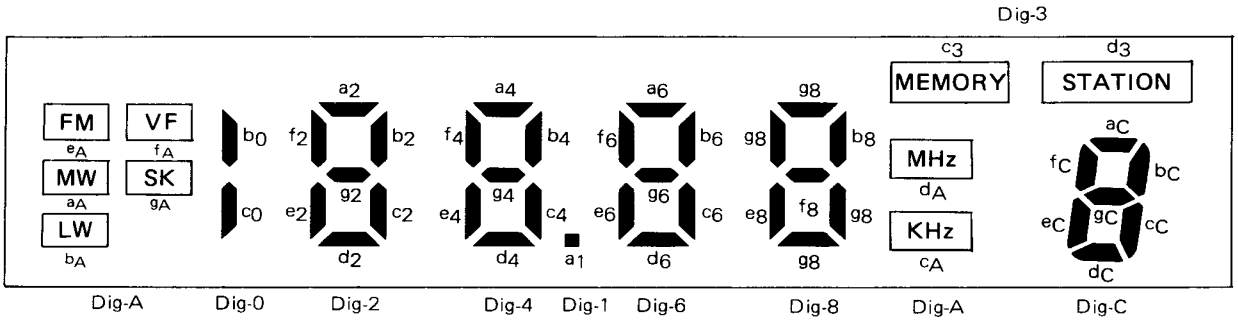
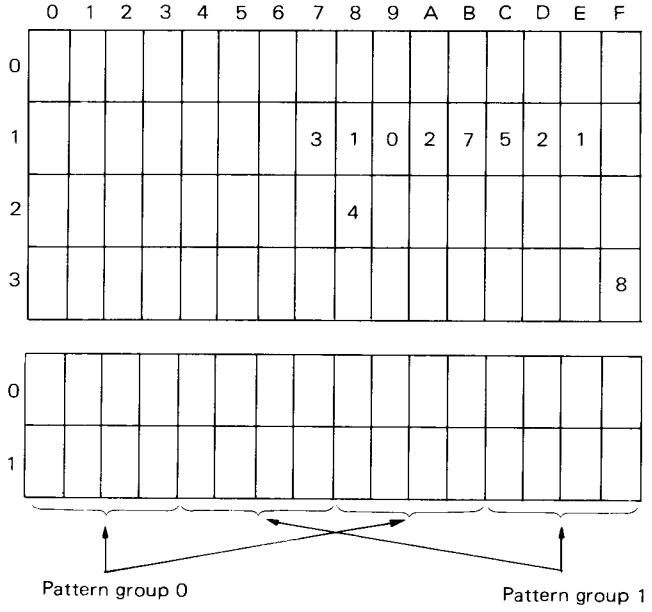
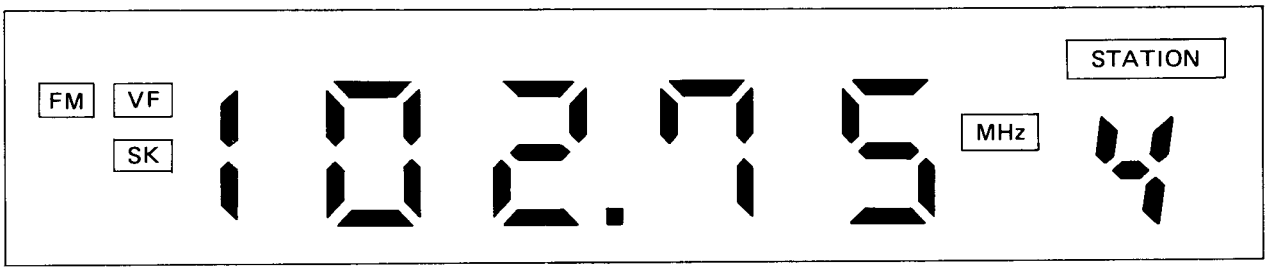


Fig. 15 Example of LCD Panel

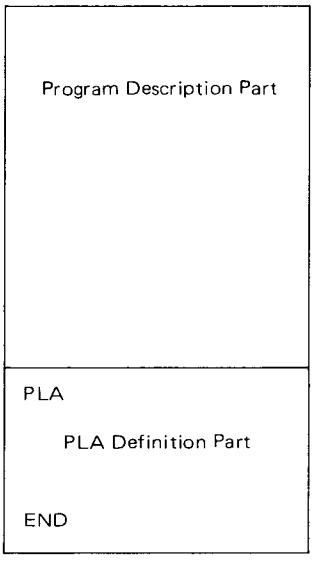



```
LCDD   17H, 0AH    ;  "FM", "MHz", "VF", "SK" ON
LCDD   18H, 00H    ;  "1" FREQUENCY
LCDD   19H, 02H    ;  "0" FREQUENCY
LCDD   1AH, 04H    ;  "2" FREQUENCY
LCDD   1BH, 06H    ;  "7" FREQUENCY
LCDD   1CH, 08H    ;  "5" FREQUENCY
LCDD   1DH, 03H    ;  "STATION"
LCDD   1EH, 01H    ;  DECIMAL POINT
LCDD   28H, 0CH    ;  "4" PRESET STATION NO.
LCDD   3FH, 0FH    ;  ALL DIGITS ON
```



5.4 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    1111001B
       ↑      ↑
       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 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.

Example of Programs of PLA

```
;* * * * PLA DEFINITION * * * *
PLA
;
SPLSEL      2
;
;* * * SEGMENT PATTERN 0 * * *
;
;           g f e d c b a
DSP         0 1 1 1 1 1 1 B     ;    0
DSP         0 0 0 0 1 1 0 B     ;    1
DSP         1 0 1 1 0 1 1 B     ;    2
DSP         1 0 0 1 1 1 1 B     ;    3
DSP         1 1 0 0 1 1 0 B     ;    4
DSP         1 1 0 1 1 0 1 B     ;    5
DSP         1 1 1 1 1 0 1 B     ;    6
DSP         0 1 0 0 1 1 1 B     ;    7
DSP         1 1 1 1 1 1 1 B     ;    8
DSP         1 1 0 1 1 1 1 B     ;    9
DSP         0 0 0 0 0 0 0 B     ;    BLANK
DSP         1 1 1 1 0 0 1 B     ;    E
DSP         0 1 1 1 0 0 1 B     ;    □
DSP         1 1 1 0 1 1 0 B     ;    H
DSP         1 1 1 0 0 1 1 B     ;    P
DSP         1 0 0 0 0 0 0 B     ;    —
;
;* * * SEGMENT PATTERN 1 * * *
;
DSP         1 0 1 0 0 1 0 B     ;    0 (FM 50 kHz)
DSP         0 0 1 1 0 0 0 B     ;    FM, MHz
DSP         0 1 1 1 0 0 0 B     ;    FM, MHz, VF
DSP         1 1 1 1 0 0 0 B     ;    FM, MHz, VF, SK
DSP         0 0 0 0 0 0 0 B     ;    NO USE
DSP         1 1 0 0 0 0 0 B     ;    5 (FM 50 kHz)
DSP         0 0 0 0 1 0 1 B     ;    MW, kHz
DSP         0 0 0 0 1 1 0 B     ;    LW, kHz
DSP         0 0 0 0 0 0 0 B     ;    NO USE
DSP         0 0 0 0 0 0 0 B     ;    NO USE
DSP         0 0 0 0 0 0 0 B     ;    BLANK
DSP         0 0 0 0 0 0 0 B     ;    NO USE
DSP         0 0 0 0 0 0 0 B     ;    NO USE
DSP         0 0 0 0 0 0 0 B     ;    NO USE
DSP         0 0 0 0 0 0 0 B     ;    NO USE
DSP         0 0 0 0 0 0 0 B     ;    NO USE
;
;
END
```

6. INSTRUCTION

6.1 INSTRUCTION SET

b_{13} b_{12} b_{11} b_{10} \ b_{15} b_{14}		00	01	10	11
		0	1	2	3
0 0 0 0	0	NOP VHF HF	KIN M KI M	———	ST M, r
0 0 0 1	1	SPB P, N SS N_1 BANK1 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 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	SIB M, I	LCDD M, D	SB r, M
1 0 0 0	8	SBK0 TPF P, N TSF N_2 TCEF	AIN M, I	———	ADN r, M
1 0 0 1	9	SBK1 TPT P, N TST N_2 TCET	SIN M, I	TTM TIP	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

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 to 0F of BANK0
- M : Data memory address
  One of data memories (96 memories of addresses 00 to 3FH of BANK0 and addresses 00 to 1FH of BANK1)
- P : Port, $0 \leq P \leq 2$
- N_1 : Bit position of status word 1 (#3=BANK F/F 2, #2=BANK F/F 1, #1=Carry F/F, #0=0)
- N_2 : Bit position of status word 2 (BANK/CE), $0 \leq N_2 \leq 3$
- ( ) : Content of register or memory
- c : Carry
- b : Borrow
- ( )n : Content of bit N of register or memory
- h : Halt release conditions $0 \leq h \leq 7$

	MNEMONIC	OPERAND		FUNCTION	OPERATION	MACHINE CODE			
		1ST	2ND			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 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

	MNEMONIC	OPERAND		FUNCTION	OPERATION	MACHINE CODE			
		1ST	2ND			OPERATION CODE			
ADDITION	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 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 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 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

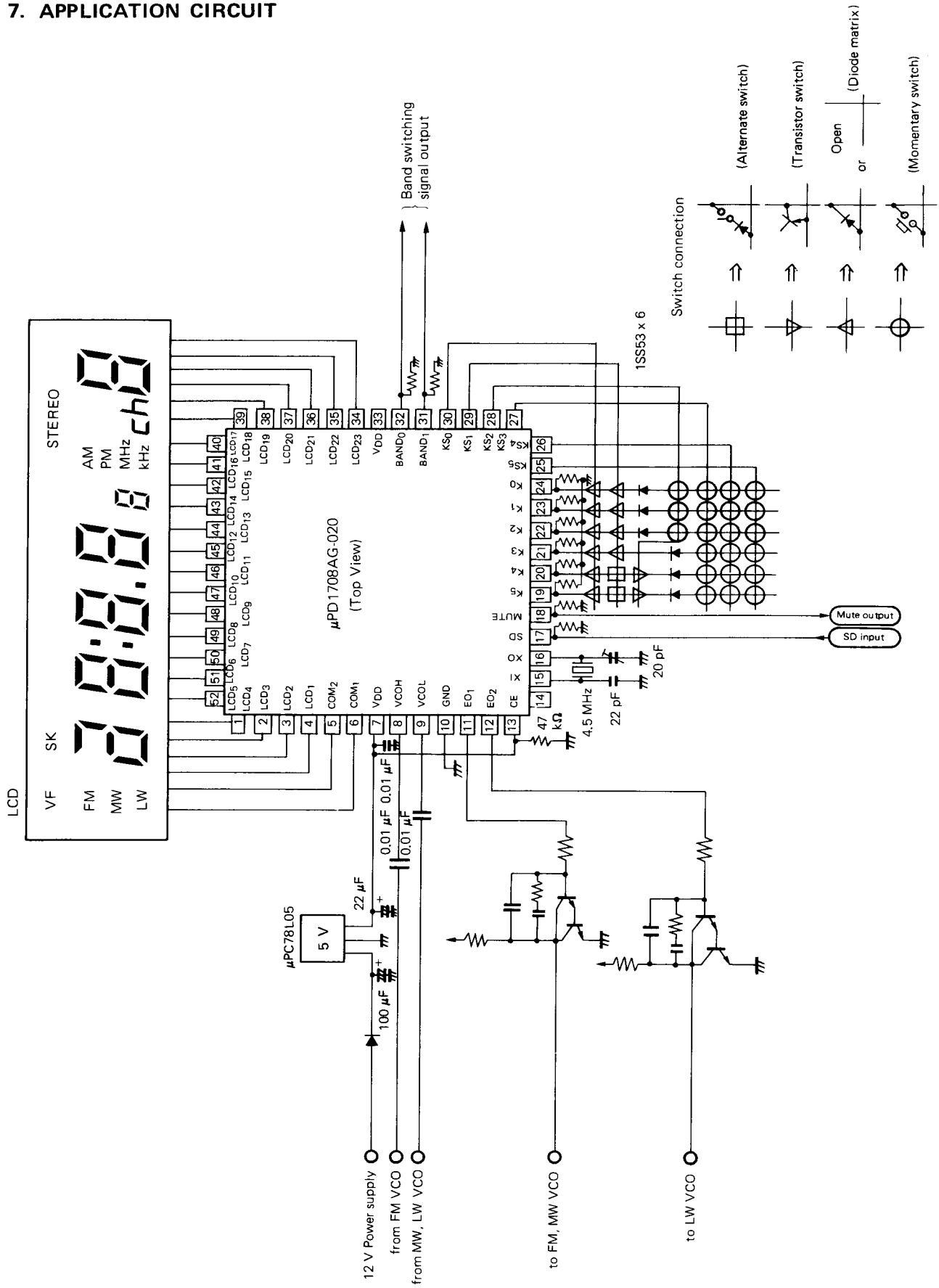
	MNEMONIC	OPERAND		FUNCTION	OPERATION	MACHINE CODE			
		1ST	2ND			OPERATION CODE			
SUBTRACTION	SIBS	M	I	Subtract immediate data with borrow, then skip if borrow	M ← (M) − I − b skip if borrow	0 1 1 1 1 1	D_H	D_L	I
	SIBN	M	I	Subtract immediate with borrow, then skip if not borrow	M − (M) − I − b skip if not borrow	0 1 1 0 1 1	D_H	D_L	I
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) $\geq$ (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 equal 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 register than or equal to immediate data	M − I skip if not borrow (M) $\geq$ 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	$\bar{I}$
	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

	MNEMONIC	OPERAND 1ST	OPERAND 2ND	FUNCTION	OPERATION	MACHINE CODE: OPERATION CODE			
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	$[D_H, Rn] \leftarrow (M)$	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	$M \leftarrow [D_H, Rn]$	1 1 0 0 0 1	D_H	D_L	Rn
	MVSR	M_1	M_2	Move memory to memory in the same row	$[D_H, D_{L1}] \leftarrow [D_H, D_{L2}]$	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 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 in page 0	0 0 0 1 1 0	ADDR (10 bits)		
				Jump to the address specified in page 1	PC ← ADDR in page 1	0 0 0 0 1 0			
SUBROUTINE	CAL	ADDR		Call subroutine in page 0	Stack ← (PC) + 1	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 unconditionary	PC ← (stack), and skip	0 0 0 1 0 1	—	—	—

	MNEMONIC	OPERAND 1ST	OPERAND 2ND	FUNCTION	OPERATION	MACHINE CODE OPERATION CODE			
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	——	——	——
F/F TEST	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	——	——	——
TIMER TEST	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
STATUS/WORD AND TERMINAL TEST	SS	N_1		Set status word 1	(STATUS WORD 1) $_N \leftarrow 1$	0 0 0 0 0 1	——	N_1	——
STATUS/WORD AND TERMINAL TEST	RS	N_1		Reset status word 1	(STATUS WORD 1) $_N \leftarrow 0$	0 0 0 0 1 1	——	N_1	——
STATUS/WORD AND TERMINAL TEST	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	——
STATUS/WORD AND TERMINAL TEST	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	——
STATUS/WORD AND TERMINAL TEST	STC			Set carry F/F	Carry F/F ← 1	0 0 0 0 0 1	——	0 0 1 0	——
STATUS/WORD AND TERMINAL TEST	RSC			Reset carry F/F	Carry F/F ← 0	0 0 0 0 1 1	——	0 0 1 0	——
STATUS/WORD AND TERMINAL TEST	BANK0			Select BANK0	BANK F/F ← 0	0 0 0 0 1 1	——	1 1 0 0	——
STATUS/WORD AND TERMINAL TEST	BANK1			Select BANK1	BANK F/F ← 1	0 0 0 0 0 1	——	0 1 0 0	——
STATUS/WORD AND TERMINAL TEST	TCET			Test CE, skip if true	if CE=1, then skip	0 0 1 0 0 1	——	0 0 1 0	——
STATUS/WORD AND TERMINAL TEST	TCEF			Test CE, skip if false	if CE=0, then skip	0 0 1 0 0 0	——	0 0 1 0	——
STATUS/WORD AND TERMINAL TEST	SBK0			Skip if BANK0	if BANK F/F=0, then skip	0 0 1 0 0 0	——	1 1 0 0	——
STATUS/WORD AND TERMINAL TEST	SBK1			Skip if BANK1	if BANK F/F=1, then skip	0 0 1 0 0 1	——	0 1 0 0	——

	MNEMONIC	OPERAND		FUNCTION	OPERATION	MACHINE CODE			
		1ST	2ND			OPERATION CODE			
INPUT/OUTPUT	LCDD	M	D	Output segment pattern to LCD DIG "D" based on the memory, output memory to LCD DIG "D" directly	LCD(D) ← SEG PLA ← M, or LCD(D) ← M	1 0 0 1 1 1	D_H	D	D_L
	KIN	M		Input key data to memory, then skip if data are zero	M ← K_0~K_3, skip if (M)=0	0 1 0 0 0 0	D_H	D_L	——
	KI	M		Input key data to memory	M ← K_0~K_3	0 1 0 0 0 0	D_H	D_L	0 0 0 0
	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
OTHERS	VHF			Connect FM terminal to prescaler via 1/2 divider	FM term. → 1/2 → Prescaler	0 0 0 0 0 0	0 0	0 0 0 0	0 0 0 1
	HF			Connect AM terminal to prescaler directly	AM term. → Prescaler	0 0 0 0 0 0	0 0	0 0 0 0	0 0 0 0
	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 1
	NOP			No operation		0 0 0 0 0 0	——	——	——

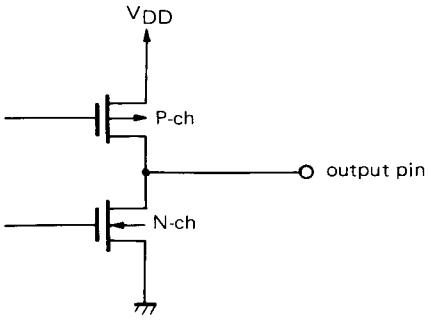
7. APPLICATION CIRCUIT



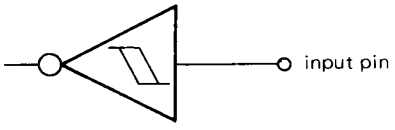
8. INPUT/OUTPUT CIRCUITS

The followings are input/output circuits of each terminal of μPD1708. (There are some omissions in it.)

(1) LCD_1 to LCD_{23}, PB_0 to PB_3, PC_0 to PC_3, EO_1, EO_2

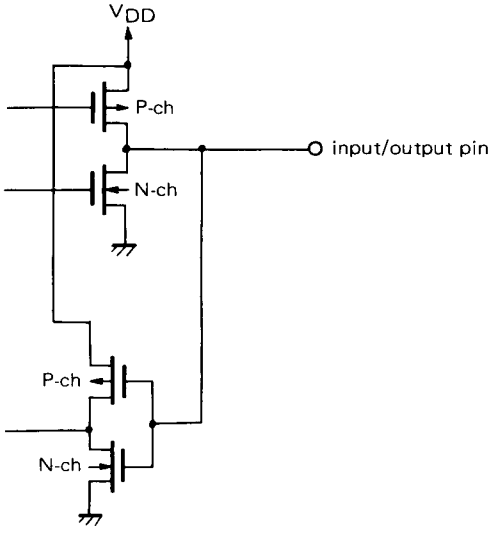


(2) CE

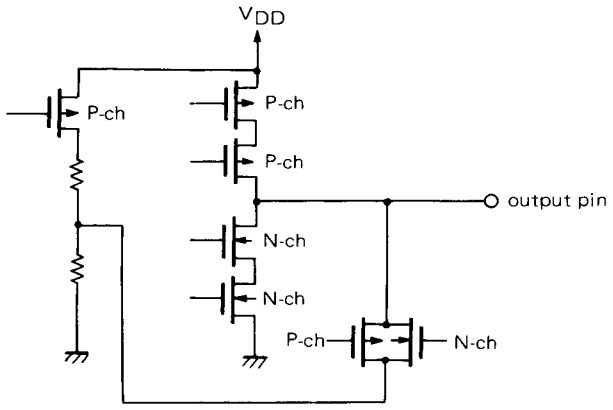


Schmitt-Trigger

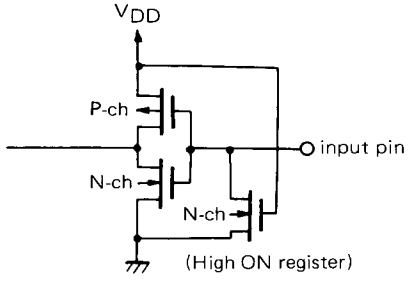
(3) PA_0 to PA_3



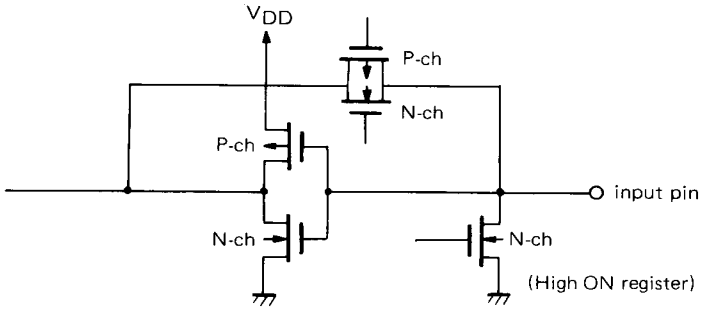
(4) COM_1, COM_2



(5) K_0 to K_3

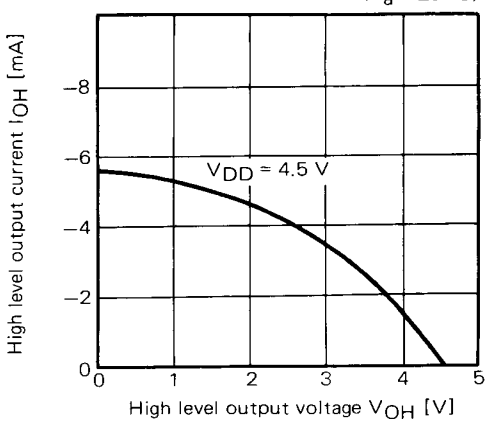


(6) VCOH, VCOL

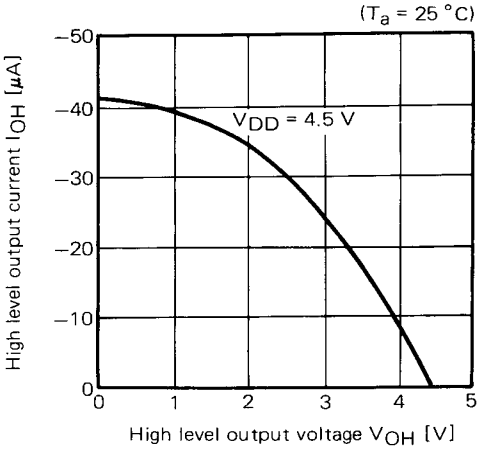


CHARACTERISTIC CURVES

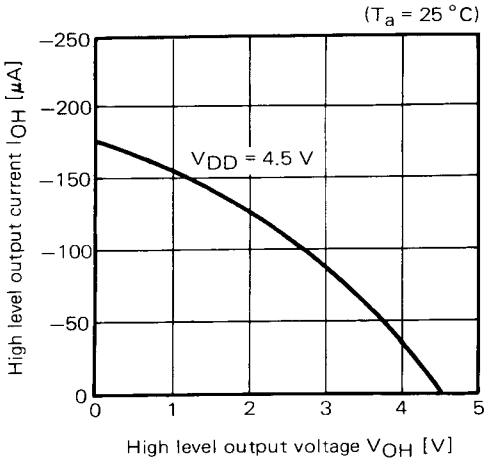
I_{OH} vs. V_{OH} TYPICAL
(Port A, B, C, EO_1, EO_2)
(T_a = 25 °C)



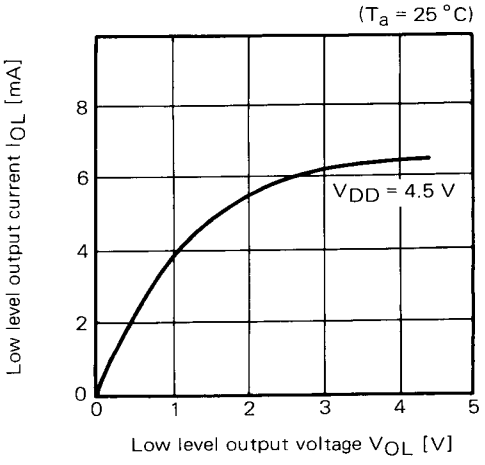
I_{OH} vs. V_{OH} TYPICAL
(LCD_1 to LCD_{23})
(T_a = 25 °C)



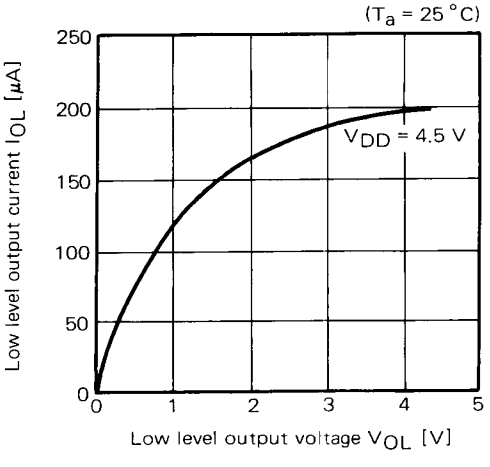
I_{OH} vs. V_{OH} TYPICAL
(COM_1, COM_2)
(T_a = 25 °C)



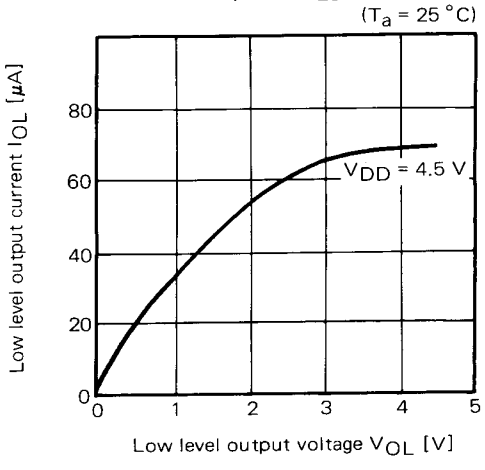
I_{OL} vs. V_{OL} TYPICAL
(Port A, C_0, C_1, EO_1, EO_2)
(T_a = 25 °C)



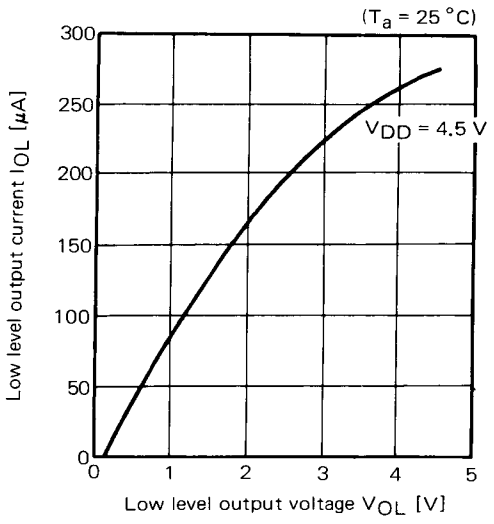
I_{OL} vs. V_{OL} TYPICAL
(Port B, C_2, C_3)
(T_a = 25 °C)



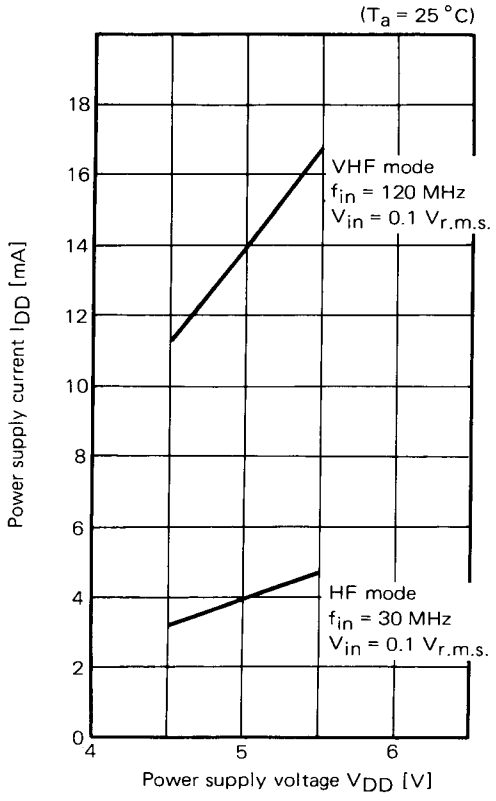
I_{OL} vs. V_{OL} TYPICAL
(LCD_1 to LCD_{23})
(T_a = 25 °C)



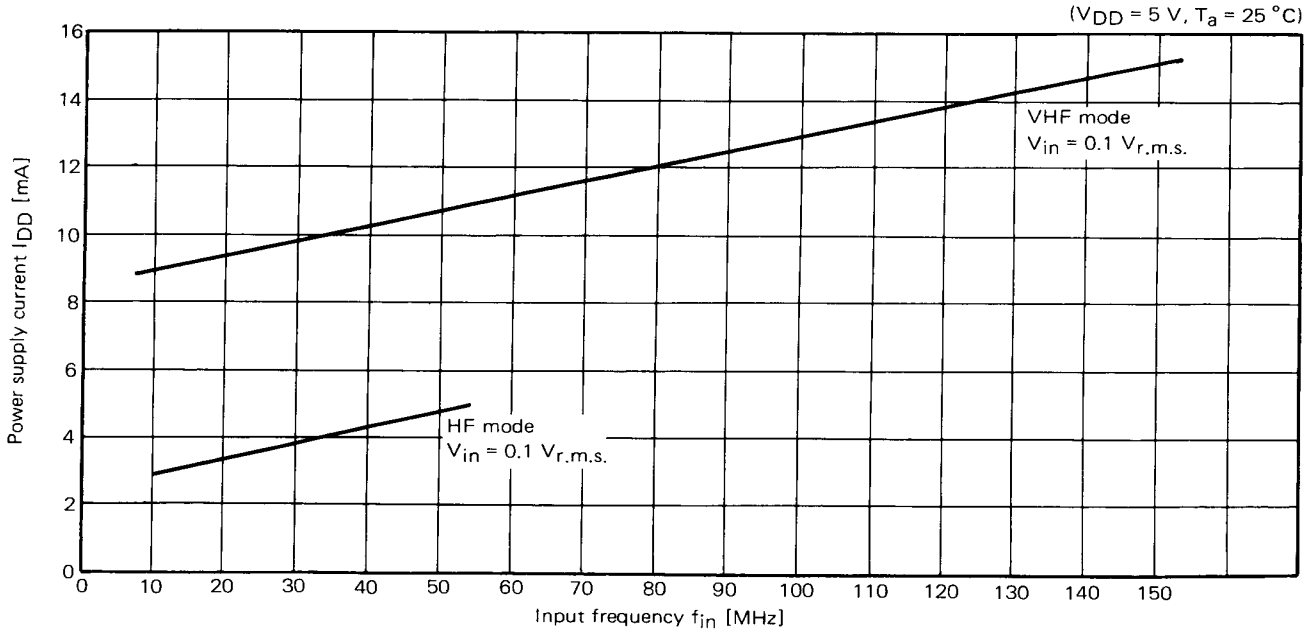
I_{OH} vs. V_{OH} TYPICAL
(COM_1, COM_2)



I_{DD} vs. V_{DD} TYPICAL

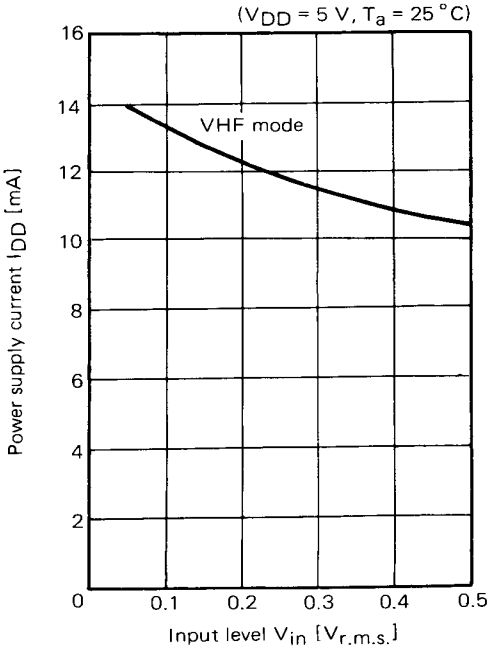


I_{DD} vs. f_{in} TYPICAL



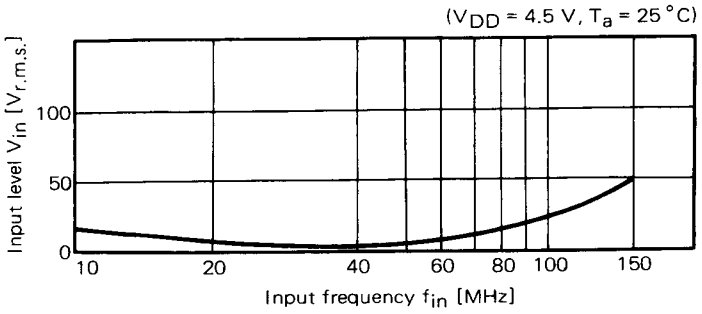
I_{DD} vs. V_{in} TYPICAL

(V_{DD} = 5 V, T_a = 25 °C)



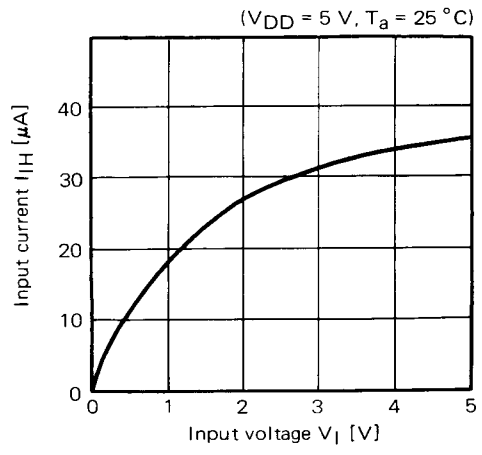
V_{in} vs. f_{in} TYPICAL

(V_{DD} = 4.5 V, T_a = 25 °C)



I_{IH} vs. V_I TYPICAL
(K_0 to K_3)

(V_{DD} = 5 V, T_a = 25 °C)



9. ELECTRICAL CHARACTERISTICS

9.1 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 Sink Current	I_O	10	mA
Operating Temperature	T_a	−40 to +85	°C
Storage Temperature	T_{stg}	−55 to +125	°C

9.2 RECOMMENDED OPERATING CONDITIONS

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT	CONDITION
Supply Voltage	V_{DD}	4.5	5.0	5.5	V	
Operating Temperature	T_a	−40		+85	°C	
Input Oscillation Voltage	V_{in1}	0.3		4.5	V_{p-p}	VCOL pin
Input Oscillation Voltage	V_{in2}	0.5		4.5	V_{p-p}	VCOH pin

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

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT	CONDITION
Supply Voltage	V_{DD1}	4.5	5.0	5.5	V	CPU and PLL operation
Supply Voltage	V_{DD2}	3.5		5.5	V	CPU operation, only
Supply Current	I_{DD1}		15		mA	120 MHz input to VCOH pin
Supply Current	I_{DD2}		400		μA	CPU operation, only
Supply Current	I_{DD3}		6	10	mA	HF mode, input 50 MHz
Data Retention Voltage	V_{DR}	2.5		V_{DD}	V	Crystal oscillation stoped (V_{DD}=5.0 V)
Data Retention Current	I_{DR}		1	10	μA	Crystal oscillation stoped (V_{DD}=5.0 V)
High Level Output Current	I_{OH1}	−1.0	−2.5		mA	Port A, B, C, EO_1, EO_2 pins (V_{OH}=V_{DD}−1.0 V)
High Level Output Current	I_{OH2}	−10	−18		μA	LCD_1 to LCD_{23} pins (V_{OH}=V_{DD}−1 V)
High Level Output Current	I_{OH3}	−20	−60		μA	COM_1 and COM_2 pins (V_{OH}=V_{DD}−1.0 V)
Low Level Output Current	I_{OL1}	1.0	3.8		mA	Port A, PC_1, PC_0, EO_1, EO_2 pins (V_{OL}=1.0 V)
Low Level Output Current	I_{OL2}	25	100		μA	Port B, PC_3, PC_2 pins (V_{OL}=1.0 V)
Low Level Output Current	I_{OL3}	10	30		μA	LCD_1 to LCD_{23} pins (V_{OL}=1 V)
Low Level Output Current	I_{OL4}	20	80		μA	COM_1 and COM_2 pins (V_{OL}=1 V)
High Level Input Current	I_{IH1}	10	35	60	μA	K_3 to K_0 pins (V_{DD}=V_{IH}=5.0 V)
High Level Input Current	I_{IH2}	100	300		μA	VCOH, VCOL, XI pins (V_{DD}=V_{IH}=5.0 V)
Output Leakage Current	I_L			±1.0	μA	EO_1, EO_2 pins
High Level Input Voltage	V_{IH1}	0.7 V_{DD}			V	PA_3 to PA_0 pins
High Level Input Voltage	V_{IH2}	0.6 V_{DD}			V	K_3 to K_0 pins
High Level Input Voltage	V_{IH3}	0.8 V_{DD}			V	CE pin
Low Level Input Voltage	I_{IL1}			0.3 V_{DD}	V	CE, Port A pins
Low Level Input Voltage	I_{IL2}			0.2 V_{DD}	V	K_3 to K_0 pins
Low Level Input Voltage	V_{IL3}			0.2 V_{DD}	V	CE pin
Output Level	V_O	2.3		2.8	V	COM_1, COM_2 pins 1/2 bias voltage (V_{DD}=5 V)

9.4 CAPACITANCE (V_{DD}=5 V, T_a=25 °C)

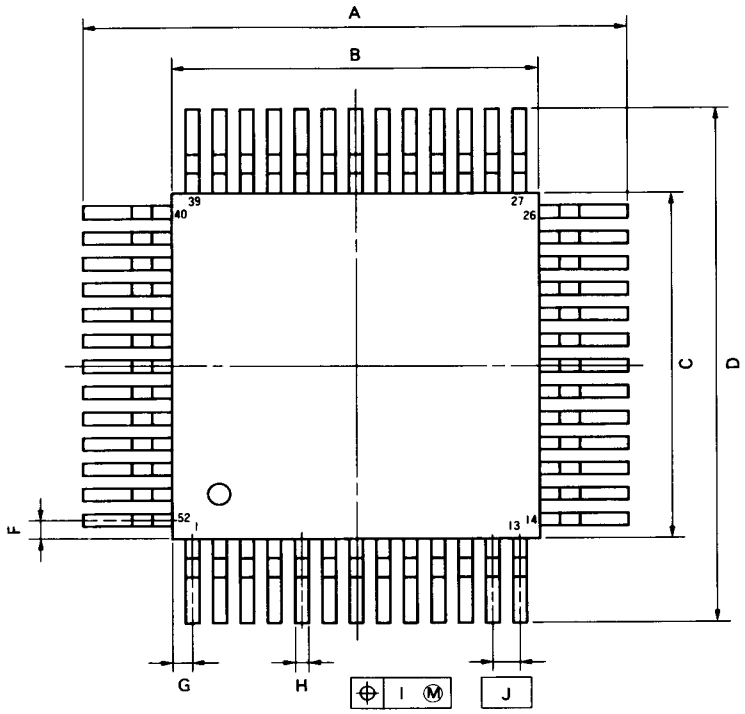
CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT	CONDITION
Input Capacitance	C_{in1}		6.5		pF	VCOH pin f_{in}=100 MHz
Input Capacitance	C_{in2}		3.5		pF	VCOL pin f_{in}=1 MHz

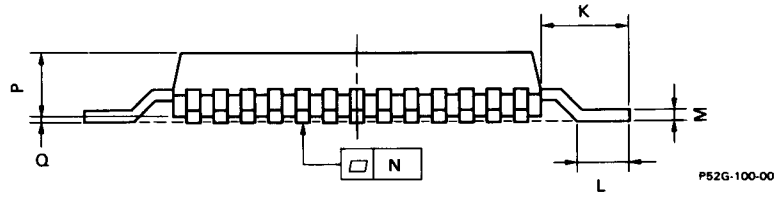
9.5 AC CHARACTERISTICS (V_{DD}=4.5 to 5.5 V, T_a=−40 to +85 °C)

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT	CONDITION
Operating Frequency	f_{in1}	0.59		20	MHz	VCOL pin (MF Mode), v_i=0.1 V_{p-p}, V_{DD}=4.5 V
Operating Frequency	f_{in2}	0.6		40	MHz	VCOL pin (HF mode) v_i=0.1 V_{p-p}, V_{DD}=4.5 V
Operating Frequency	f_{in3}	0.6		50	MHz	VCOL pin (HF mode) v_i=0.3 V_{p-p}, V_{DD}=4.5 V
Operating Frequency	f_{in4}	10		130	MHz	VCOH pin (VHF mode) v_i=0.3 V_{p-p}, V_{DD}=4.5 V
Operating Frequency	f_{in5}	10		150	MHz	VCOH pin (VHF mode) v_i=0.5 V_{p-p}, V_{DD}=4.5 V

10. PACKAGE DIMENSIONS (Unit: mm)

Lead bended type





P52G-100-00

NOTE

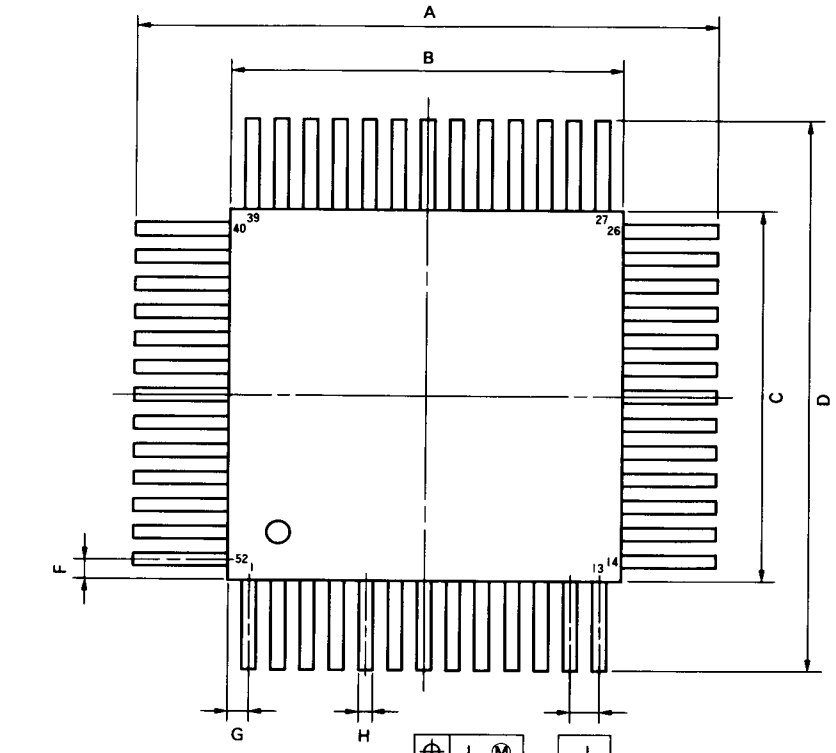
Each lead centerline is located within 0.20 mm (0.008 inch) of its true position (T.P.) at maximum material condition.

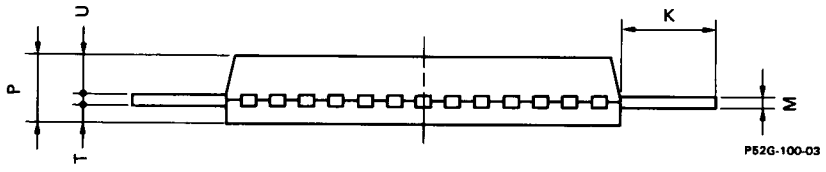
ITEM	MILLIMETERS	INCHES
A	$21.0^{+0.4}$	$0.827^{+0.016}$
B	$14^{+0.2}$	$0.551^{+0.008}_{-0.009}$
C	$14^{+0.2}$	$0.551^{+0.008}_{-0.009}$
D	$21.0^{+0.4}$	$0.827^{+0.016}$
F	1.0	0.039
G	1.0	0.039
H	$0.40^{+0.10}$	$0.016^{+0.004}_{-0.005}$
I	0.20	0.008
J	1.0 (T.P.)	0.039 (T.P.)
K	$3.5^{+0.2}$	$0.138^{+0.008}_{-0.009}$
L	$2.2^{+0.2}$	$0.087^{+0.008}_{-0.009}$
M	$0.15^{+0.10}_{-0.05}$	$0.006^{+0.004}_{-0.003}$
N	0.15	0.006
P	$2.6^{+0.2}_{-0.1}$	$0.102^{+0.009}_{-0.004}$
Q	$0.1^{+0.1}$	$0.004^{+0.004}$

When you will order, please designate about the product of this package type as follows.

μPD1708AG-XXX-00

Straight lead type





P52G-100-03

NOTE

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

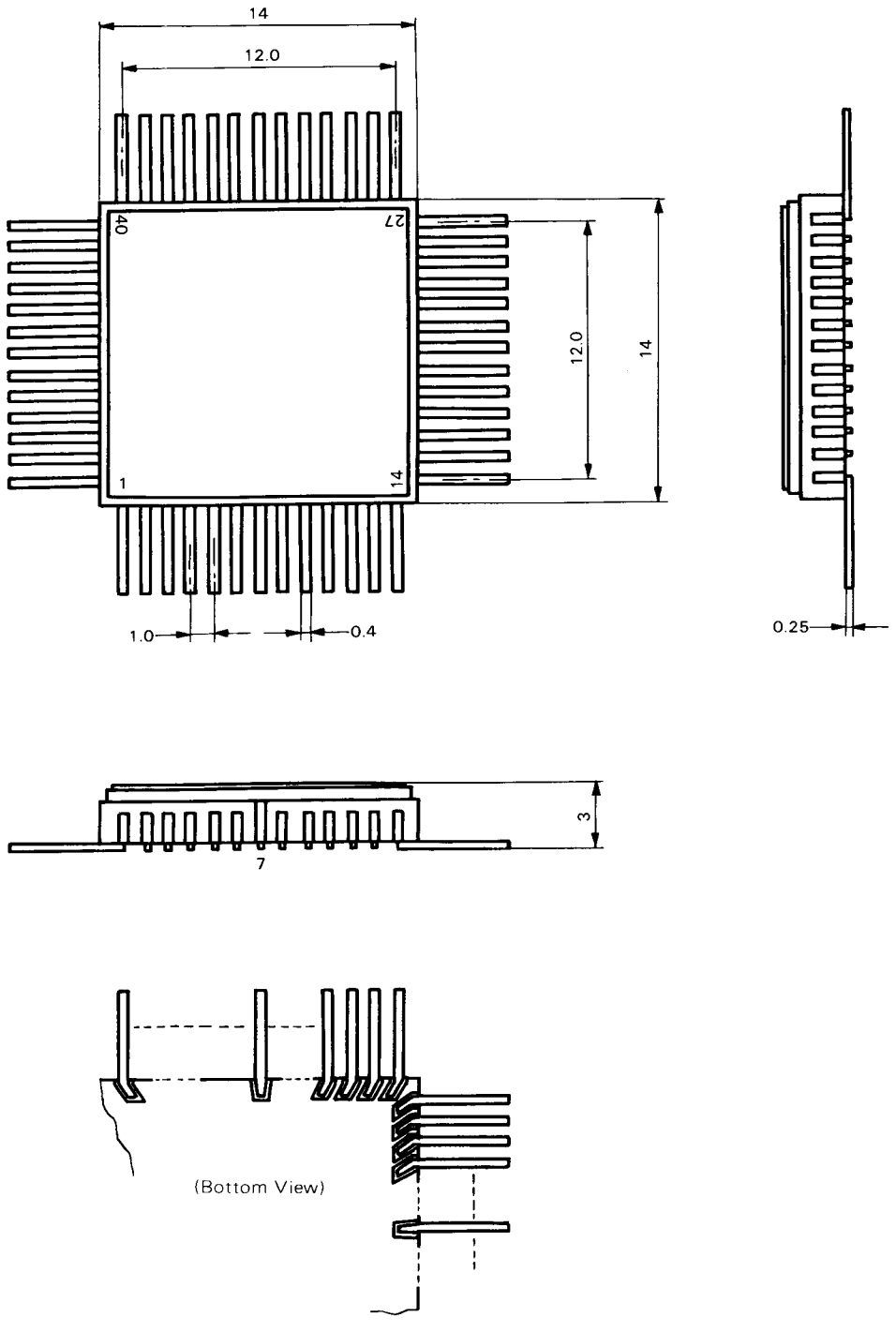
ITEM	MILLIMETERS	INCHES
A	$19.8^{+0.4}$	$0.780^{+0.016}_{-0.017}$
B	$14^{+0.2}$	$0.551^{+0.008}_{-0.009}$
C	$14^{+0.2}$	$0.551^{+0.008}_{-0.009}$
D	$19.8^{+0.4}$	$0.780^{+0.016}_{-0.017}$
F	1.0	0.039
G	1.0	0.039
H	$0.40^{+0.10}$	$0.016^{+0.004}_{-0.005}$
I	0.20	0.008
J	1.0 (T.P.)	0.039 (T.P.)
K	$2.9^{+0.2}$	$0.114^{+0.008}_{-0.009}$
M	$0.15^{+0.10}_{-0.05}$	$0.006^{+0.004}_{-0.003}$
P	$2.6^{+0.2}_{-0.1}$	$0.102^{+0.008}_{-0.004}$
T	1.0	0.039
U	1.45	0.057

When you will order, please designate about the product of this package type as follows.

μPD1708AG-XXX-03

PACKAGE DIMENSION FOR ENGINEERING SAMPLE (Unit: mm)

52-pin Ceramic Flat Package



Please attention to the following.

- ○ Metal cap is connected to pin 7, and it is positive power supply level.
- ○ Pin 33 is NC (No Connection), and pin 7 can be used only for providing positive power supply.
- ○ Leads of the bottom are formed on the slope.

11. SUPPORT TOOLS

The following support tools are available for developing systems using the μPD1708.

Hard ware	EVAKIT-1700		This is an evaluation board usable in common in the μPD1700 series. In case of μPD1708, EVAKIT-1700 must be used with option I/O board (EV-1708) 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 μPD1708 to evaluate the system. In case of μPD1708, SE-1700 must be used with option I/O board (EV-1708) to evaluate systems.		
	EV-1708		This is an option I/O board connecting to EVAKIT-1700 or SE-1700 for developing and evaluating the program of μPD1708.		
Soft ware	μPD1700 series Assembler	Absolute Assembler	Host	OS	Order name
			PDA-880	CP/M™	μS281AS1700
			PDA-800 + PDA-800FDD		
			MD-080 series		
			MD-086 series	MP/M-86™	μS171AS1700

Remarks
1. Supply format of software is 8-inch FD double-side double density.
2. Regarding CP/M version of OS of PDA-800 and PDA-880 systems, please contact our authorized agent.
3. CP/M™ and MP/M-86™ are the trademark of Digital Research Co., Ltd.

IN-1763
September 1986P
Printed in Japan