MOS DIGITAL INTEGRATED CIRCUIT
μPD1707G

SINGLE-CHIP MICROCOMPUTER WITH A DIGITAL PHASE LOCKED LOOP FREQUENCY SYNTHESIZER

μPD1707 is a 4-bit digital tuning CMOS microcomputer incorporating a Phased Locked Loop (PLL) and controller in a chip. It can be connected directly to a fluorescent indicator panel.

The CPU has 4-bit parallel addition and subtraction instructions (AD and SU), logical operation instructions (such as EXL), bit test instructions (such as TMT), carry F/F set and reset instructions (such as STC), interrupt function, and timer function.

Each chip is made of a 52-pin plastic flat package provided with sufficient I/O ports controlled with effective input/output instructions (such as IN and OUT), 8-bit serial interface (serial I/O and shift CLOCK) of the μCOM standard, 6-bit A/D converter, and Clock Generator Port (CGP) with variable frequency and duty.

The A/D converter for FM/AM tuning can input the signal meter data output from the IF detection stage to determine the electric field intensity or the automatic tuning stop level.

The A/D converter for TV is applicable to various fields as a pin to input S-shaped curves for Automatic Fine Tuning (AFT).

The CGP can be used as a simple 64-step D/A converter by adding a Low Pass Filter (LPF) in output stage or used as a 128-step frequency generator pin.

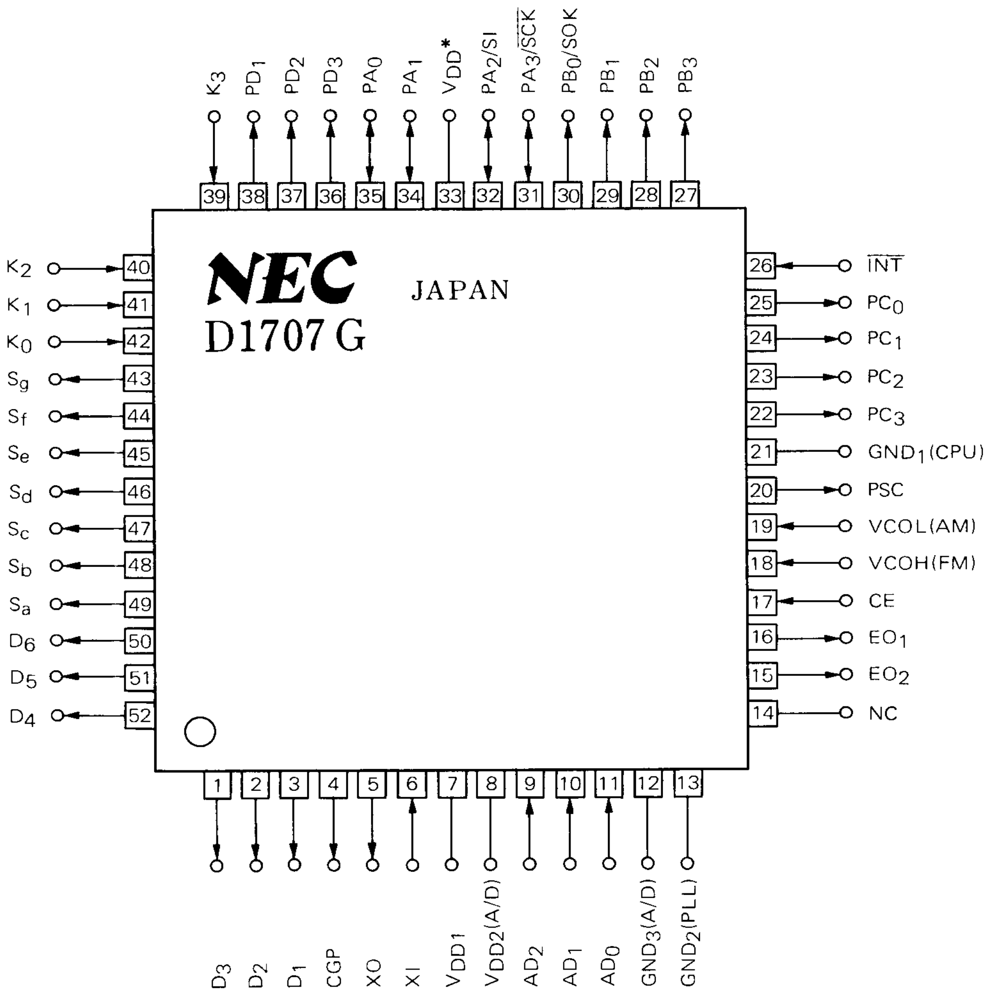
FEATURES

- 4-bit microcomputer for digital tuning
- Built-in PLL frequency synthesizer and controller
- Single power supply of 5 V ±10 %
- Low power consumption due to CMOS
- Easy data memory (RAM) backup (with CE pin)
- Programmable memory (ROM) : 16 bits x 2040 steps
- Data memory (RAM) : 4 bits x 128 (= 2 x 64) words
- Eighty-five 1-word instructions
- Instruction execution time : 33.3 μs (4.5 MHz crystal oscillator)
- Ample addition and subtraction instructions (12 addition instructions and 12 subtraction instructions)
- Compound decision instructions (such as TMT and TMF)
- Memory-to-memory data transfer enabled at same row address
- Instructions for indirect data transfer between registers (such as MVRD and MVRS)
- Sixteen general registers (allocated in RAM space)
- Stack level : 3 level
- Dedicated pin for display and key input (dynamic display in six 7-segment digits)
- Direct drive of Fluorescent Indicator Panel (FIP) (segments only)
- Segments (Sa to Sg) P-ch open drain output (maximum with stand voltage : 35 V)
- Clock stop (CKSTP) instruction (Power supply current reduced to less than 10 μA)
- Four I/O ports (PA_3 to PA_0) individual lines can be configured as input or output

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

- Eleven output lines (PB_3 to PB_0, PC_3 to PC_0, PD_3 to PD_1) are N-ch open drain outputs
- Built-in serial interface (PA_3 = shift clock, PA_2 = serial input, PB_0 = serial output) for data transfer in 8-bit units by SIO instruction
- Built-in 6-bit successive approximation A/D converter ($V_{ref}=V_{DD}$) executed by microcode using TADT and TADF instructions
- Built-in Clock Generator Port (CGP) (64-step variable duty at 2.69 kHz or variable frequency with 180 kHz or 18 kHz as the base frequency)
- Effective I/O instructions (such as IN and OUT)
- Input and output port test instructions (TPT and TPF)
- Edge triggered vectored interrupt function ($\overline{\text{INT}}$ pin)
- Built-in timer F/F (This F/F is set every 125 ms, the time of day clock function can be implemented easily)
- Built-in interval pulse output (internal output of 5 ms pulse (200 Hz with 60 % duty)) tested by TIP Instruction
- PLL lock state test (TUL instruction)
- Transferring frequency division ratio, frequency division method, and reference frequency data to PLL section with only one instruction (PLL instruction)
- Built-in Programmable Logic Array (PLA) for segments and digits (user mask programmable)
- Independent frequency input pins for VCOL(AM) and VCOH(FM) (maximum input frequency : 15 MHz for VCOH(FM) and VCOL(AM) pins)
- Direct connection to Two-Modules Prescaler: μPB553AC (maximum 130 MHz) and μPB562AC (maximum 1 GHz) signal connection is mode via VCOH(FM) and PSC pins
- Selection of pulse-swallow or direct frequency divisions by program
- Independent Error Out pin for AM and FM bands (EO_1 and EO_2 pins)
- Selection of seven reference frequencies by program: 1, 5, 6.25, 9, 10, 12.5, and 25 kHz

PIN CONFIGURATION (Top View)



* : This pin is internally connected to Pin 7.

NC : No Connection

PIN DESCRIPTION

PIN SYMBOL	PIN NAME	DESCRIPTION	OUTPUT TYPE
S_a–S_g	Segment Outputs	Display segment output pins. The SEG instruction unconditionally loads the contents of specified address in the data memory (RAM) to the programmable logic array (PLA). The selected data are output to these pins from PLA. (See the segment PLA section.) The data is used to address one of 32 locations in Segment PLA. These pins can also be used as a key return signal source for a key matrix. Since this port is P-channel open drain, an external pull-up resistor is necessary.	P-ch open drain
GND_3–GND_1	Ground	Device ground pins. GND_1 is for the CPU, GND_2 is for the PLL, and GND_3 is for the A/D converter.	–
PA_3–PA_0	Port A	4-bit input/output port. The contents at address 1FH in the data memory (RAM) bank 0 specify the input or output state of each bit in this port. In addition, this port can be used as a serial interface with an SIO instruction. In this case, the PA_3 pin operates as an $\overline{SCK}$ (Shift Clock) pin. In the same way, the PA_2 pin operates as an SI (Serial Input) pin and the PB0 pin operates as an SO (Serial Output) pin. This port enters the input mode when the device is reset (V_{DD} = low → high and CE = low → high) or when a CKSTP instruction is executed. To use the PA_3 pin as the $\overline{SCK}$ pin, a pull-up resistor must be connected to the PA_3 pin. Otherwise, no shift clocks are generated. (See Notes 1 and 2.)	CMOS push-pull
PB_3–PB_0	Port B	4-bit output port. Since this pin is of the N-ch open drain output type (with maximum voltage +8.5 V), an external pull-up resistor is necessary. The PB0 pin can be used as an SO (Serial Output) pin by executing an SIO instruction. (See Notes 1 and 3.)	N-ch open drain
PC_3–PC_0	Port C	4-bit output port. Since this pin is of the N-ch open drain output type (maximum voltage V_{DD}), an external pull-up resistor is necessary. (See Notes 1 and 3.)	N-ch open drain
PD_3–PD_1	Port D	3-bit output port. Since this pin is of the N-ch open drain output type (maximum voltage 8.5 V), an external pull-up resistor is necessary. (See Notes 1 and 3.)	N-ch open drain
D_6–D_1	Digit Outputs	6-bit output port. This port is used as digit signal output pins for display. Unlike other ports (PA, PB, PC and PD), this port is controlled by a DIG instruction. The register contents specified in the DIG instruction are loaded to the digit PLA, and these contents are output from the PLA to these pins. Sixteen patterns can be specified in the digit PLA. Whether or not to activate this port when CE pin = low can be specified at mask generation. When the CE pin goes to the high level from the low level, those pin return to the former condition that CE pin goes to the low level. When the power is turned on for the first time, those digit pins are in indefinite condition.	CMOS push-pull
K_3–K_0	Key Return Signal Inputs	4-bit input port. Unlike other ports, data can be input from these pins to a specified data memory (RAM) with a KI or KIN instruction. Normally, these pins are used as key return signal input pins from the externally connected key matrix.	Input

PIN SYMBOL	PIN NAME	DESCRIPTION	OUTPUT TYPE
CGP	Clock Generator Port	This terminal can be used as Clock Generator Port (CGP) or 1-bit output port (PG_2) by program control. Two types of signals can be generated by the CGP: variable duty signal or variable frequency signal. During the variable duty pulse (VDP) mode, 2.69 kHz signal is output continuously with its duty changeable in 64 steps. During the signal generator (SG) mode 50 % duty signal is output continuously with its frequency changeable in 64 steps with 180 Hz or 18 kHz as the reference frequency. Since this port is N-channel open drain, an external pull-up resistor is necessary. (See Notes 1 and 2.)	N-ch open drain
AD_2—AD_0	Analog Digital Input	Three analog-digital (A/D) converter input pins. The built-in A/D converter employs a 6-bit successive approximation method. The A/D converter reference voltage V_{DD2} (A/D) is V_{DD1} (5 V ±10 %).	Input
$\overline{INT}$	Interrupt	Interrupt request signal input pin. An interrupt request is issued at the trailing edge of the signal applied to this pin. When the interrupt request is accepted, control unconditionally jumps to address 1.	Input
VCOH(FM)	FM Local Oscillator Signal Input	Programmable counter input pin for pulse-swallow method. To this pin, input the local oscillator (VCO) output divided by 1/16 or 1/17 with the prescaler μPB553AC or by 1/128 or 1/136 with μPB562AC. Since this pin has an internal AC amplifier, cut the DC component with a capacitor before inputting.	Input
VCOL(AM)	AM Local Oscillator Signal Input	Programmable counter input pin for direct frequency division. Input the local oscillator (VCO) output to this pin. Since this pin has an internal AC amplifier, cut the DC component with a capacitor before inputting.	Input
PSC	Pulse Swallowing Control	Pin to output frequency division ratio selection signal to two-modulus prescaler μPB553AC or μPB562AC. Connect this pin directly to PSC pin of μPB553AC or μPB562AC. The frequency division ratio of μPB553AC is 1/16 or 1/17, and that of μPB562AC is 1/128 or 1/136.	CMOS push-pull
CE	Chip Enable	Device selection signal input pin. This pin must be high level to enable the device and low level to disable the device. When this pin is in the low level, the PLL, segment output, and digit outputs are disabled; however, an input of 134 μs or less will not be accepted as a valid low. When a CKSTP instruction in the program is executed while the CE pin is low level, the internal clock generator and CPU stop, and the memory can enter the hold state requiring low power consumption (10 μA or less). When the CE pin goes to the high level from the low level, the device is reset and the program starts from address 0. In this case, port A (PA_3 to PA_0) enters the input mode.	Input
XI, XO	X'tal	Crystal oscillator connector pin. Connect a 4.5 MHz crystal oscillator to this pin.	CMOS push-pull
V_{DD1}	Power Supply	Device power supply pin. This pin supplies 5 V ±10 % while the device is operating. This voltage can be dropped to 2.5 V to hold the internal data memory (RAM) with a CKSTP instruction. When the voltage applied to this pin changes from 0 → 4.5 V, the device is reset and the program starts from address 0. (within 500 ms)	—

PIN SYMBOL	PIN NAME	DESCRIPTION	OUTPUT TYPE
V_{DD2} (A/D)	Power Supply (A/D)	A/D converter reference voltage pin. The reference voltage must be V_{DD1}, power supply voltage. Therefore, V_{DD2} (A/D) must be connected to V_{DD1}.	–
EO_1, EO_2	Error Output	PLL error output pins. When the divided oscillation frequency is higher than the reference frequency, this pin outputs a high-level signal; when it is lower, this pin outputs a low-level signal; and when they are the same, this pin is set to high impedance. This output is applied to the tuner varactor diode via the low pass filter. Both AM and FM low pass filters can be connected to these pins because the same signal is output to the EO_1 and EO_2 pins simultaneously.	CMOS three states

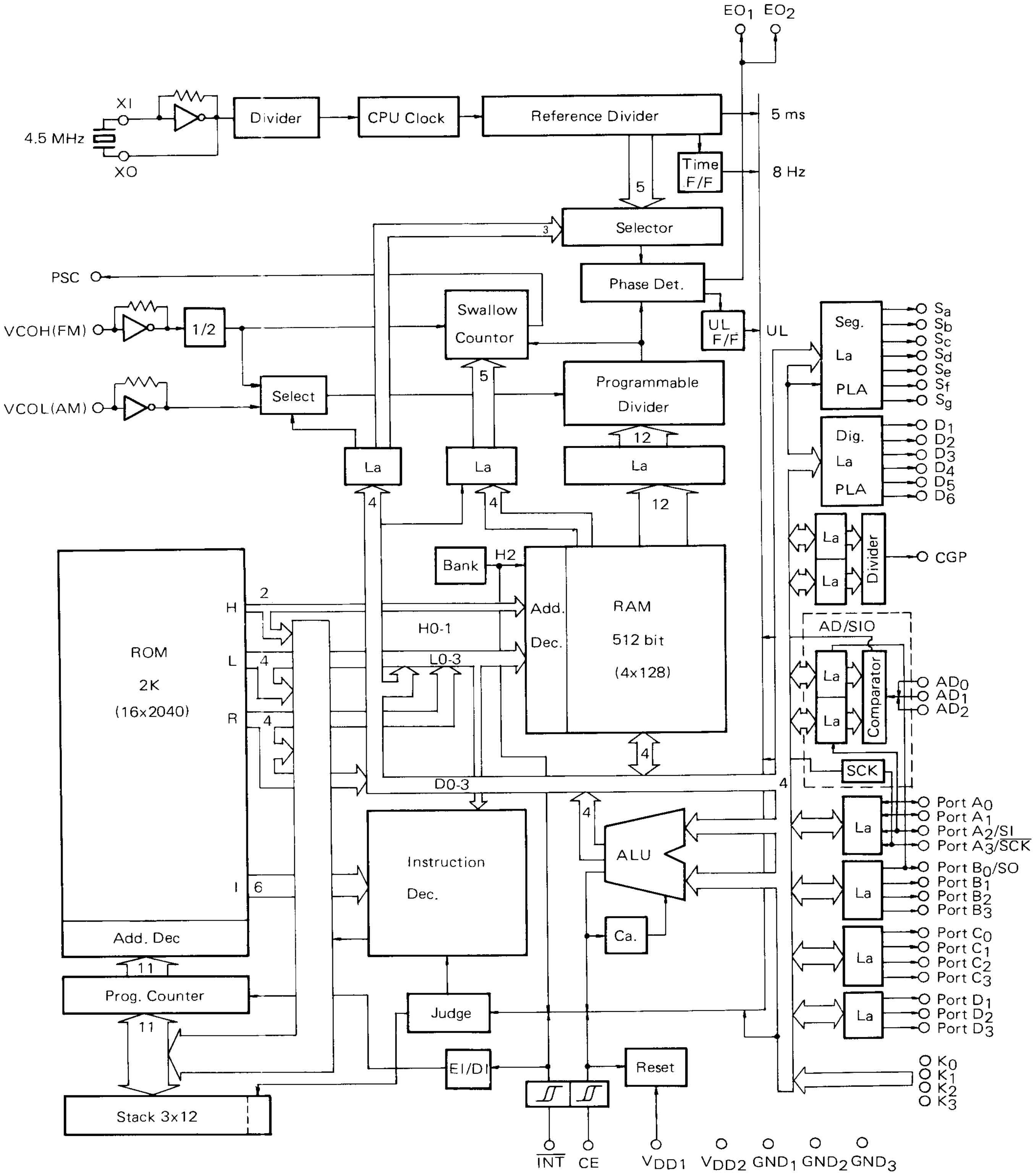
Notes: 1. PB_0 is associated with the least significant bit of the register or operand data, PB_3 is associated with the most significant bit of the register or operand data for port operation instructions (I/O, test, and set/reset instructions). This is the same as for PA, PC, PD and CGP (PG_2).

2. When the device is reset (V_{DD} = low → high and CE = low → high), or a CKSTP instruction is executed, port PA (PA_0 to PA_3) enters the input mode.

3. The output ports (PB, PC, PD and CGP) enter the high-impedance state when the device is reset (V_{DD} = low → high only) or when a CKSTP instruction is executed.

When the CE pin goes from low to high, the output ports remain in the same state and they do not enter the high-impedance state. When the V_{DD} pin goes from low to high or when the CE pin goes from low to high after executing a CKSTP instruction, the output ports are in the high-impedance state.

BLOCK DIAGRAM



CONTENTS

1. CPU

1.1 PROGRAM COUNTER

The program counter addresses the programmable memory (ROM) that stores the programs. μPD1707 has a 11-bit binary counter.

PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0

11 bits

Usually, the counter increments by one every time an instruction is executed. When a jump or subroutine call instruction is executed, the address specified in the operand field is loaded to the counter. When a skip instruction (such as ADS, TMT, RTS, etc.) is executed, the address succeeding this skip instruction address is loaded to the counter regardless of the skip conditions. If the skip conditions are satisfied, the instruction succeeding this skip instruction is assumed to be Not Operational (NOP). The NOP instruction is executed, then the next instruction address is loaded to the counter.

When an interrupt request is accepted, **address 1 is unconditionally** loaded to the counter.

Note: μPD1701, 1703, 1704, 1705, and 1710, (microcomputer series containing 1 K or fewer steps in the ROM) have a 10-bit program counter.

1.2 STACK REGISTER

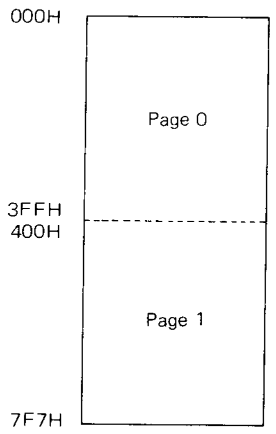
The stack register is a 3 x 12 bit register which stores the program counter value plus 1, that is the 11-bit return address, when a subroutine call instruction is executed or an interrupt request is accepted. In addition, if an instruction having a skip function is executed when accepting the interrupt request, the stack register also stores the 1-bit decision result. The stack register value is loaded to the program counter when a return instruction (RT or RTS) is executed and control returns to the main program flow.

The stack register can be used for both subroutine calls and interrupts. If one level is used for interrupts, the remaining two levels are used for subroutine calls.

1.3 PROGRAMMABLE MEMORY (ROM)

The ROM Capacity is 16 bits x 2040 steps. The accessible 2040 steps are at ROM addresses 000H to 7F7H.

ROM configuration



The μPD1707 ROM area is divided into two pages: **page 0 at addresses 000H to 3FFH and page 1 at addresses 400H to 7F7H.** A subroutine must start at page 0. A subroutine starting at page 1 cannot be called from page 0 nor page 1. (See CAL instruction operation precautions.)

In the assembler language, a JMP instruction can be executed (JMP ADDR) at addresses 000H to 7F7H without the consideration pages.

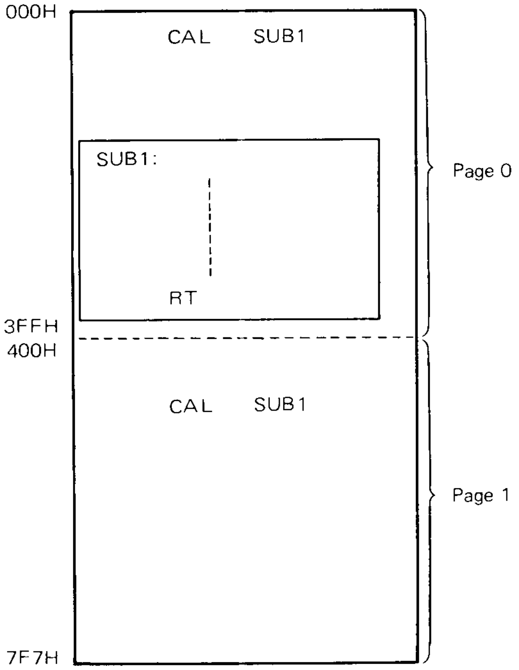
A JMP instruction for jumping into page 0 has a different operation code from that for jumping into page 1. Therefore, during program debugging, a careful attention must be made when jump code is generated. (See JMP instruction operation precautions.)

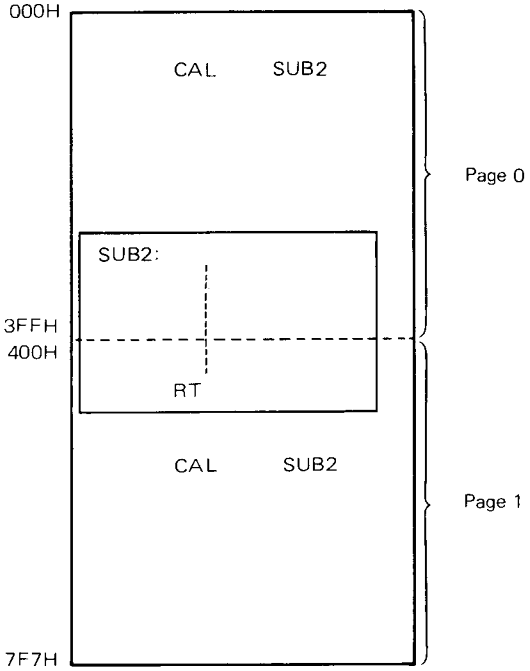
Since the μPD1707 ROM area is divided into pages 0 and 1, CAL and JMP instructions must be used considering the following precautions:

CAL Instruction Operating Precautions

When a CAL instruction is used, its calling address, that is the subroutine start address, must be in page 0 (addresses 000H to 3FFH). A subroutine, whose starting address is in page 1 (addresses 400H to 7F7H), cannot be called. The return address (for an RT or RTS instruction) may be in page 1.

Example 1. When a subroutine start address is in page 0

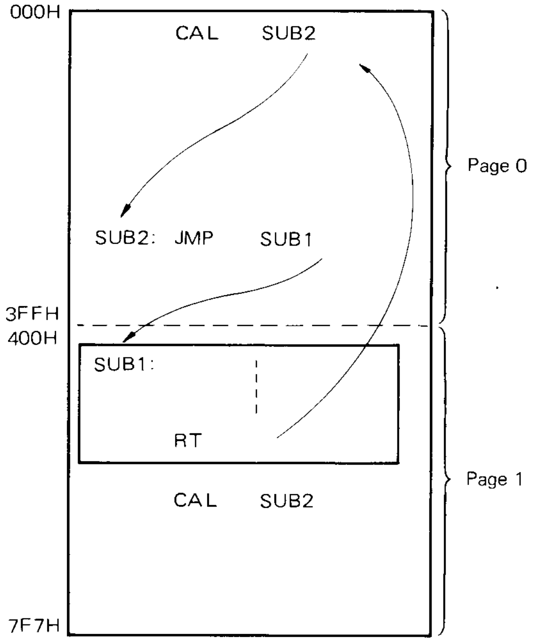




As shown above, the return address may be in either page 0 or 1 if the subroutine start address is in page 0.

As long as the subroutine start address is in page 0, a CAL instruction can be used without considering pages.

If the subroutine start address cannot be placed in page 0 for any reason, the following method should be used:



Specify a JMP instruction in page 0 and call the necessary Subroutine (SUB1) via this JMP instruction.

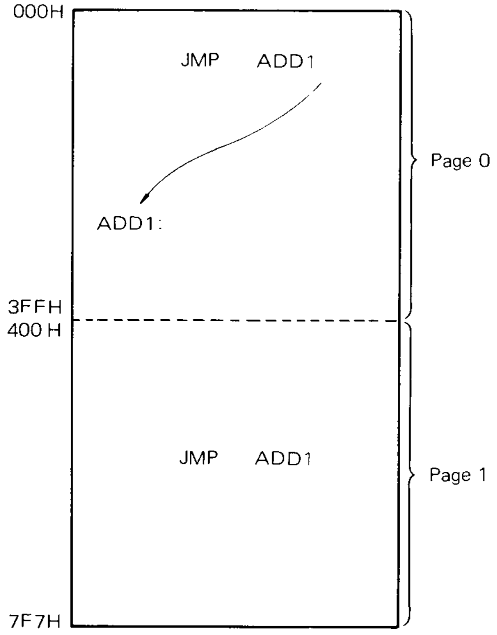
JMP Instruction Operating Precautions

In the assembler language, a JMP instruction can be used for the same description at addresses 000H to 7F7H without considering pages.

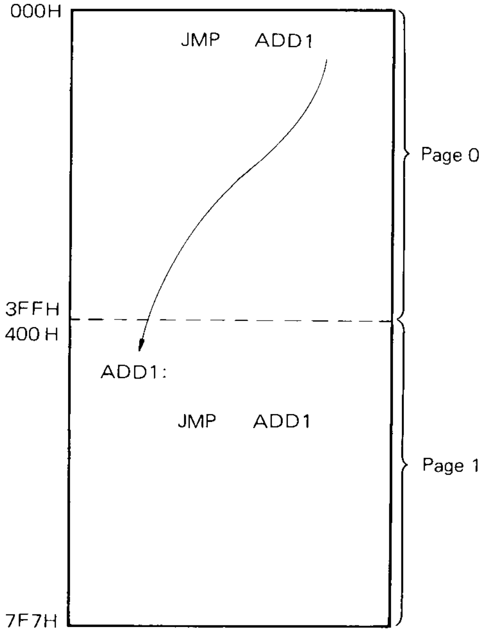
A JMP instruction to jump to page 0 (addresses 000H to 3FFH), has a different operation code from that for jumping to page 1 (addresses 400H to 7F7H). The operation code of the JMP instruction to jump to page 0 is 06 and to jump to page 1 is 02.

The μPD1700 Series assembler automatically converts these codes, referring to the jump destination.

When operation code is 06
(The jump destination address is in page 0.)



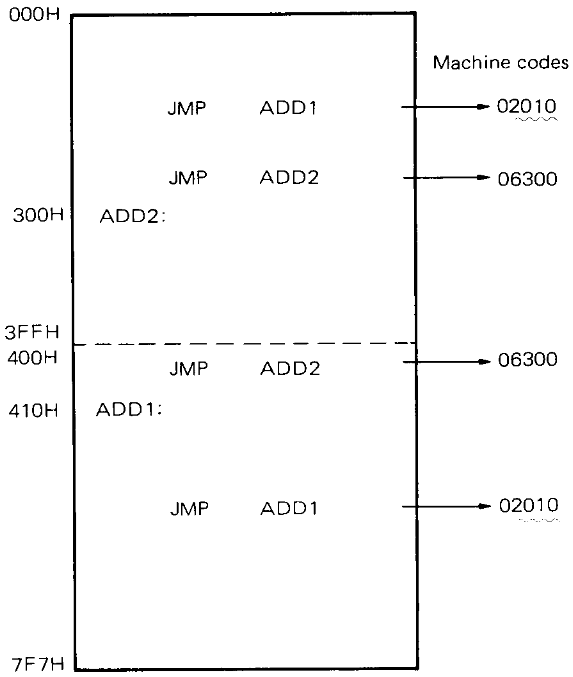
When operation code is 02
(The jump destination address is in page 1.)



For patch correction during program debugging, the programmer must select operation code 06 or 02.

If a JMP instruction causes a jump to a location at or after address 400H (with operation code 02), the addresses must also be offset by 400H. That is, address 400H must be considered as address 000H, from which the following addresses are incremented by 1 sequentially.

Therefore, address 7F7H is assumed as address 3F7H.



For example, JMP 400H written in the assembler must be input as 02000 for patch correction. JMP 000H must be input as 06000 in the same way.

1.4 DATA MEMORY (RAM)

The RAM consists of 4 bits x 128 words to store data. It is divided into banks 0 and 1, with 64 words each. To process data in either bank, the bank must be specified previously with a BANK0 or BANK1 instruction.

Addresses 00H to 0FH in bank 0 compose general registers used for arithmetic operations and data transfer to and from the memory. They can also be used as an ordinary memory. (When bank 0 is used as a register, bank specification is not necessary, but when it is used as a memory, BANK0 instruction must be used.)

The division ratio and reference frequency necessary for PLL control can also be set in the RAM.

The division ratio in PLL registers (16-bit N word) are loaded from designated RAM location at addresses 00H, 10H, 20H and 30H. The N_F bit is designated as MSB of specified general register (operand of PLL instruction). For the reference frequency, a specified word (4 bits) in the RAM (operand of PLL instruction), excluding locations 00, 10, 20, 30 and the word containing the N_F bit is assigned as the control word. The 17 bit N-data and CW are transferred to the PLL registers by the PLL instruction.

The area at address 1FH in bank 0, called a PAIO word, specifies the input and output mode of port A (PA_0 to PA_3).

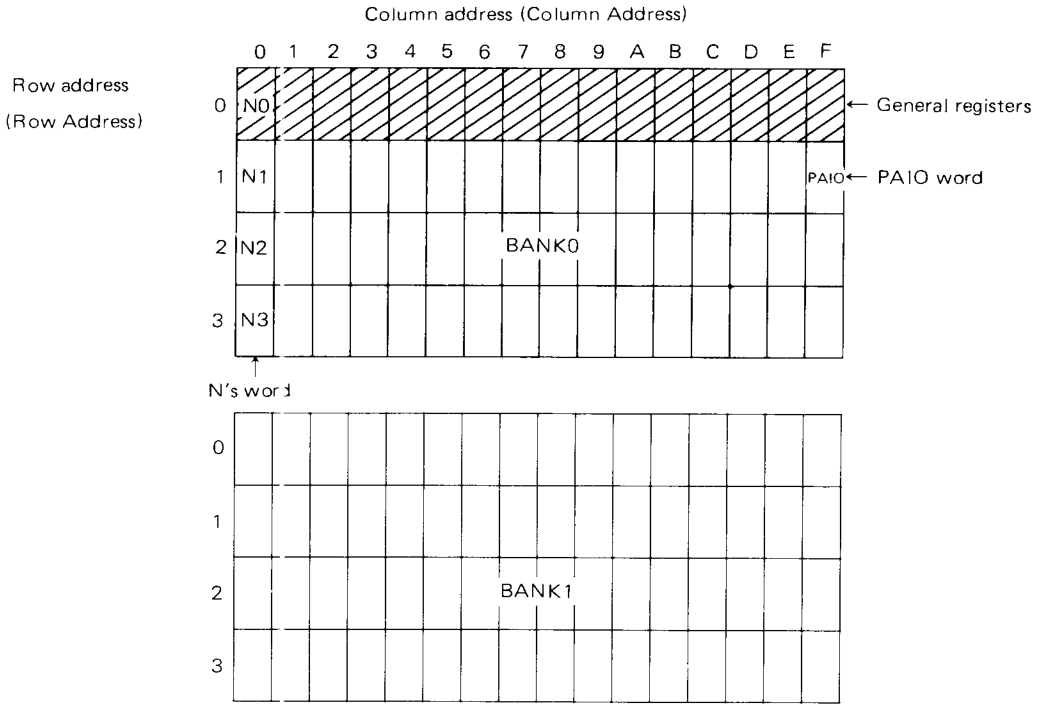


Fig. 1 RAM Configuration

Note: The most important point you should keep in mind in the general register operation in BANK1 is that μPD1707 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 set by a 8 Hz (125 ms) signal. It is reset by a Test Timer instruction (TTM).

Since the timer F/F is automatically set every 125 ms, it can be used for time of day clock (8 counts per second) and mute time counting.

The timer F/F is reset only by a TTM instruction, so **a TTM instruction must be executed within every 125 ms period**. Otherwise, the counting error will occur.

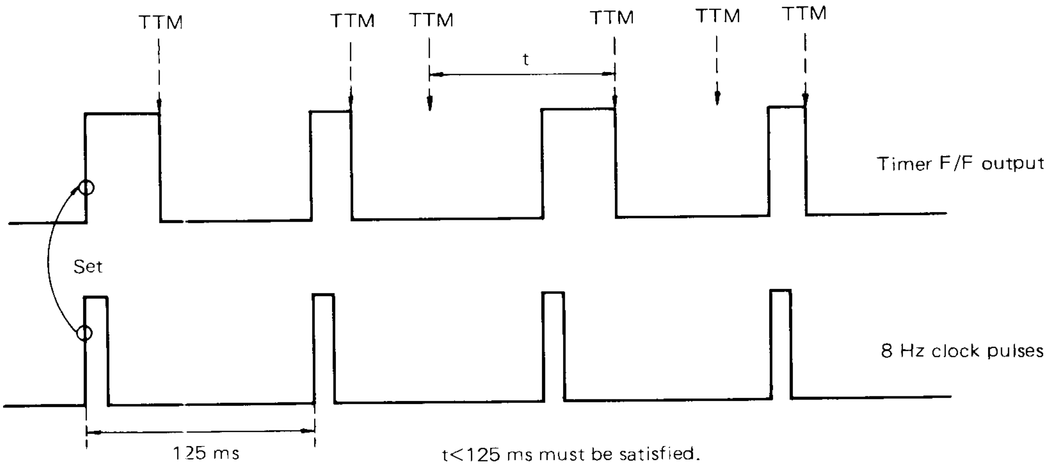


Fig. 2 TTM Instruction Execution Timings

The timer F/F can also detect a power failure. The timer F/F is reset when the power is turned on (the V_{DD} pin goes from low to high), and set when a CKSTP instruction is executed or the CE pin goes from low to high. (See Fig. 3 for the status transition diagram.)

As shown in Fig. 3, after the power is turned on (V_{DD} = low → high), the program starts from address 0 regardless of the CE pin state at this time, **the timer F/F is being reset.**

Subsequently, the timer F/F can not be set by the internal 8 Hz clock before a TTM instruction is executed (timer F/F set disable state). Once a TTM instruction is executed, the timer F/F enters the set enable state and it can be set at 125 ms cycle.

When CE pin goes from low to high, after being in the backup mode (V_{DD}=high), the control will not jump to address 0 until timer F/F is set. That is, the program starts from address 0 after **the timer F/F is set** by the internal 8 Hz clock.

As explained above, the timer F/F value at recovery from a power failure (V_{DD} = low → high) differs from that at recovery from a backup state (V_{DD} = high and CE = low → high). Therefore, recovery from a power failure or backup state can be recognized by checking the timer F/F value with a TTM instruction.

A TTM instruction must be executed **within 125 ms after the program starts from address 0**. If the result = 0 (false) indicates a power failure and the result = 1 (true) indicates backup state.

To continue the timer function (no CKSTP instruction is executed) when CE = low, the program must be coded carefully for recovery from a backup state (V_{DD} = high, CE = low → high). Since control jumps to address 0 immediately when the timer F/F is set, the timer must be updated after executing a TTM instruction for power failure detection (with true result). Otherwise, the timer lags 125 ms each time the CE pin goes from low to high.

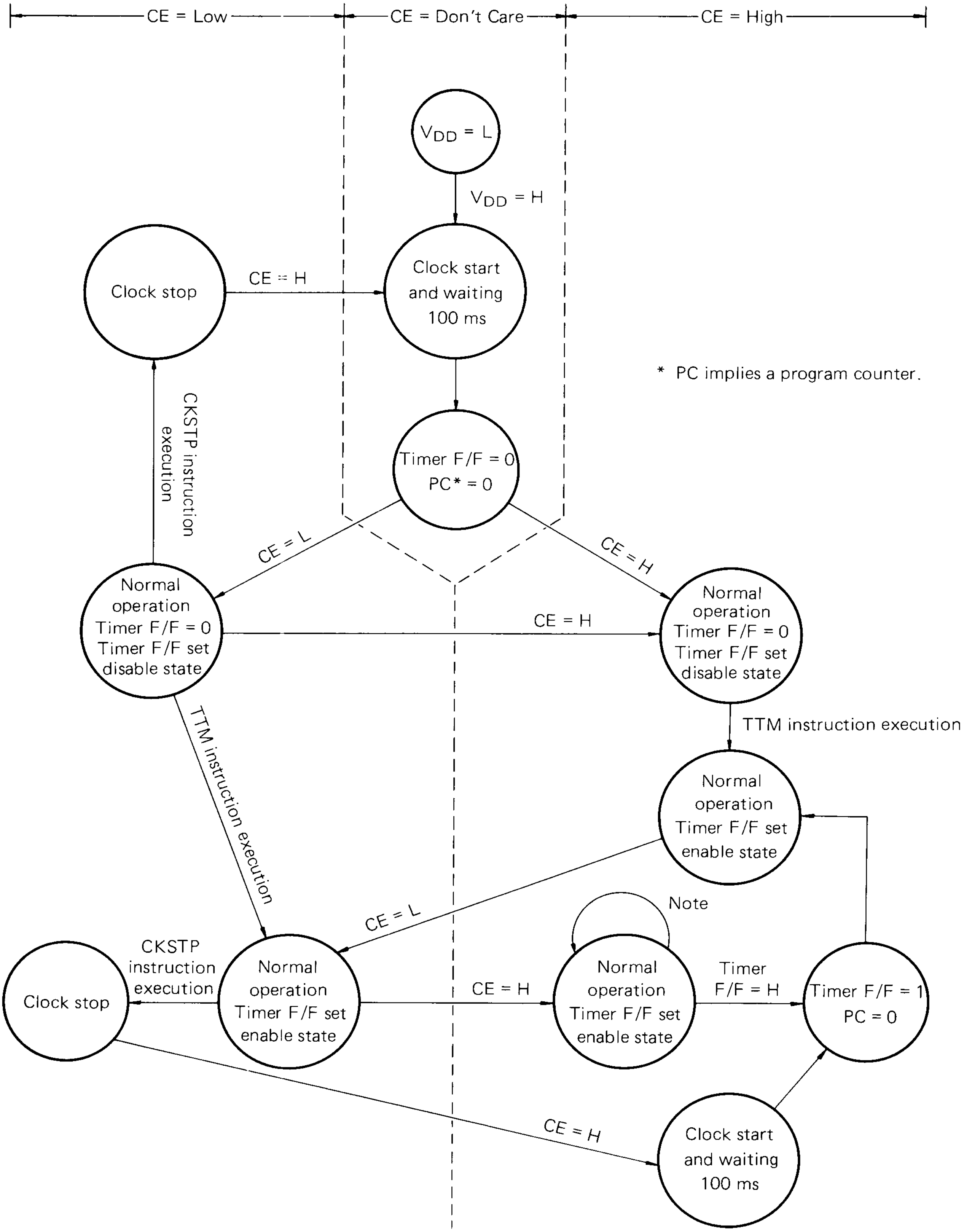

CE = Low
CE = Don't Care
CE = High
VDD = L
VDD = H
Clock stop
CE = H
Clock start and waiting 100 ms
* PC implies a program counter.
CKSTP instruction execution
Timer F/F = 0 PC* = 0
CE = L
CE = H
Normal operation Timer F/F = 0 Timer F/F set disable state
CE = H
Normal operation Timer F/F = 0 Timer F/F set disable state
TTM instruction execution
TTM instruction execution
Normal operation Timer F/F set enable state
CE = L
Note
Clock stop
CKSTP instruction execution
Normal operation Timer F/F set enable state
CE = H
Normal operation Timer F/F set enable state
Timer F/F = H
Timer F/F = 1 PC = 0
CE = H
Clock start and waiting 100 ms

Fig. 3 CPU Status Transition Diagram Associated with CE Pin

Note: 1. If TTM instructicn execution coincides with the setting of timer F/F, control cannot exit from this loop. The control can only exit w ıen the timer F/F is set, by jumping to address 0. Therefore, if execution of TTM instruction coincides with tiıner F/F setting (125 ms cycle), control cannot exit from this endless loop.

2. For μPD1707, the timer F/F is set and the program starts from address 0 when the CE pin goes from low to high after executing a CKSTP instruction. For μPD1701, μPD1703, μPD1704, and μPD1710, the timer F/F is reset and the program starts fıom address 0 in this case. Note that the timer F/F value after CKSTP instruction execution for μPD1707 differs ·rom that for μPD1701, μPD1703, μPD1704, and μPD1710.

1.6 INTERVAL PULSE

An interval pulse is a 60 % duty pulse signal at 5 ms cycle. It can be tested with a TIP instruction.

Since no Flip-Flop (F/F) is provided for this pulse, the pulse output will not be reset by executing a TIP instruction.

An accurate timer in 5 ms units can be generated by successively executing a TIP instruction to check the interval pulse edges.

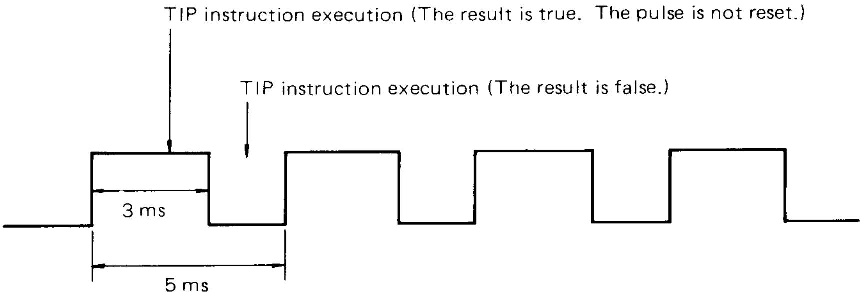

TIP instruction execution (The result is true. The pulse is not reset.)
TIP instruction execution (The result is false.)
3 ms
5 ms

Fig. 4 Interval Pulse Timings

1.7 UNLOCK F/F

While PLL is unlocked, that is, while the reference frequency (f_r) differs in phase with the divided frequency output from the VCO, the phase detector (ϕ-DET) outputs a pulse at the f_r cycle. The unlock F/F is set by this pulse and reset by executing a TUL instruction. Therefore, **a TUL instruction must be executed at an interval greater than 1/f_r.**

Otherwise, false PLL status can be detected. Furthermore, the execution of first TUL instruction must take place at least 1/f_r duration after execution of the PLL instruction.

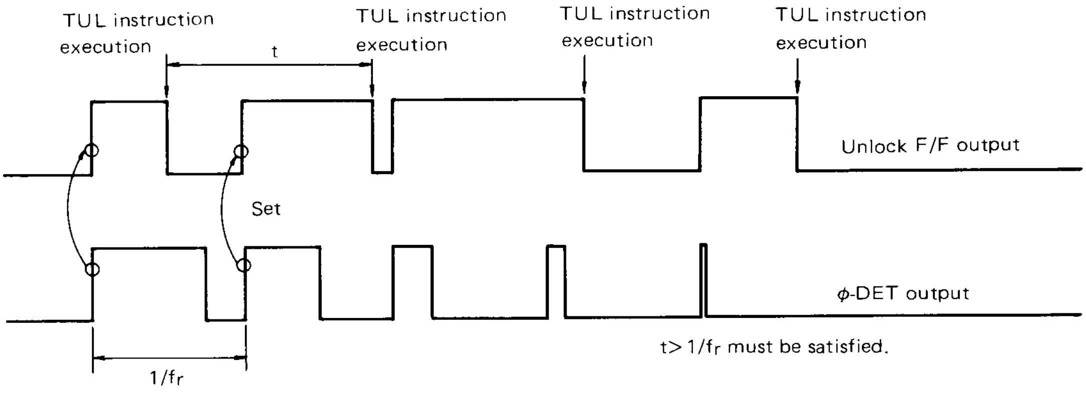

TUL instruction execution
t
TUL instruction execution
TUL instruction execution
TUL instruction execution
Unlock F/F output
Set
ϕ-DET output
t> 1/fr must be satisfied.
1/fr

Fig. 5 TUL Instruction Execution Timings

1.8 CARRY F/F

Usually, when a carry or borrow is generated by executing an arithmetic instruction, the carry F/F is set. This F/F is reset when no carry nor borrow is generated. The carry F/F value remains unchanged unless arithmetic instructions are executed. The carry F/F can be set or reset directly by carry F/F set/reset instructions (STC and RSC) or status word operation instructions (SS and RS).

Note: Even when an interrupt request is accepted, the carry F/F value is not automatically saved.

1.9 BANK F/F

The bank F/F is used for bank specification in the data memory (RAM) and port group addressing. RAM consists of 128 words, divided into banks 0 and 1 with 64 words each. In order to access data in either banks, the bank must be specified by executing a BANK0 or BANK1 instruction. The bank specification is not necessary when addresses 00H to 0FH in bank 0 are used as general registers. These registers can be accessed from either of these banks. To use these addresses as a memory, a BANK0 instruction must be specified.

The bank F/F is used for port group addressing. A port is addressed by the two bits in the instruction operand field and the contents of this bank F/F. (See explanation on ports.)

The bank F/F is automatically reset when the power is turned on for the first time (V_{DD} = low → high) or the CE pin goes from low to high, that is, when the device is reset. In this case, bank 0 is specified automatically.

Note: Even when an interrupt request is accepted, the contents of bank F/F are not automatically saved.

1.10 INT F/F AND INTE F/F

The INT F/F is set unconditionally by **the trailing edge** of a signal applied to the $\overline{\text{INT}}$ pin. If the internal INTE F/F is enabled, then interrupt request is accepted; if it is disabled, then interrupt request is not accepted. If the INT F/F is generated, while the INTE F/F is being enabled, the interrupt request will be accepted.

The INTE F/F can be enabled by executing an EI instruction in the program and disabled by executing a DI instruction.

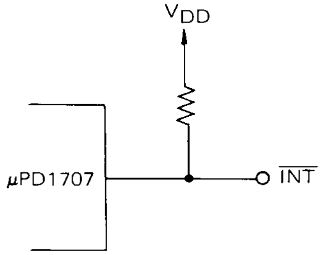
The INT F/F is reset when an instruction (DI or RS) which disables the INTE F/F is executed in the DI state.

When an interrupt request is accepted, the DI state is set automatically and control jumps to address 1 (interrupt processing routine). At this time, the address of the instruction succeeding that issuing the interrupt request is stored in the stack register. (When a JMP instruction is executed, the jump destination address is stored.) If the instruction issuing the interrupt request has a skip function, the skip condition decision result is also stored in the stack register. When an RT instruction is executed in the interrupt processing routine, the stack register contents are recovered and control returns to the source program. Since the carry F/F and bank F/F contents are not saved in the stack register, they must be saved by the routine.

Since the interrupt processing uses one stack level, stack level management is necessary.

The $\overline{\text{INT}}$ pin can also be used as an ordinary input port by a TITT or TITF instruction. **Note that $\overline{\text{INT}}$ pin value is tested true for a low level input and false for a high level input, that is, this pin is active only at the low level.**

Note: A high level signal must be input to the $\overline{\text{INT}}$ pin at least once before it can recognize a valid low. Otherwise, a low level input is internally recognized to be high until a high level signal is input. Once a high level signal is input, the interrupt will work correctly. Therefore, the $\overline{\text{INT}}$ pin must be pulled up to V_{DD} via a resister in the application circuit.



1.11 STATUS WORDS

Internal device states, which must be checked or specified for program execution, are stored in 4-bit status words. They can be set or reset by programs.

There are status words 1 and 2 to which some pins and F/F's are connected.

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

Operation instructions: SS, RS, etc.

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

Status word 1 is a write-only word to be set and reset by instructions such as SS, RS, and EI.

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

Operation instructions: TST, TSF, etc.

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

Status word 2 is a read-only word whose contents can be judged by instructions such as TST, TSF, and SBK0.

2. PLL

2.1 REFERENCE FREQUENCY GENERATOR

The external crystal oscillator (4.5 MHz) generates seven reference frequencies: 1 kHz, 5 kHz, 6.25 kHz, 9 kHz, 12.5 kHz, and 25 kHz. One of these frequencies can be selected by the program (control word data).

2.2 PHASE DETECTOR

The phase detector circuit detects phase difference between the reference frequency (f_r) and VCO output divided by the programmable divider. The output signal is input to the internal charge pump to output 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} is the VCO oscillation frequency and N is the division ratio of the programmable divider.

2.3 PROGRAMMABLE DIVIDER

The programmable divider consists of a swallow and programmable counters which are binary descremental counters.

The swallow counter is a 5-bit presettable descremental counter. The contents of the 4-bit NR0 register and 1-bit N_F register are loaded to this counter at every $1/f_r$ period.

The programmable counter is a 12-bit counter. The contents of the NR1 to NR3 registers are loaded to this counter which count down simultaneously with the swallow counter.

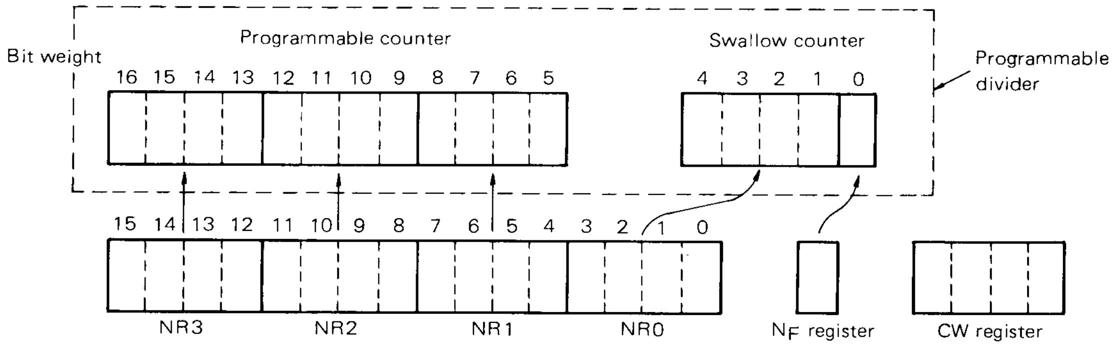


Fig. 6 Programmable Divider Configuration

The 5-bit swallow counter data is output from the PSC pin to the prescaler.

2.4 PLL REGISTER

To control the PLL of μPD1707, the following information must be stored in the PLL register:

(1) Division ratio (N)
(2) Frequency reference (f_r)
(3) Division method (direct or pulse swallow type)

The PLL register consists of the 16-bit N register and 1-bit N_F register to store the division ratio and the 4-bit Control Word (CW) register for storing the reference frequency. They are associated with the N's words, N_F bit, and Control Word (CW) in the data memory (RAM), respectively. **The contents designated RAM locations are transferred with a PLL instruction in a single execution cycle.** N's words are assigned to addresses 00H, 10H, 20H, and 30H in RAM bank 0, the N_F bit is assigned to the most significant bit of specified general register, and the CW is assigned to specified RAM area, excluding the N's words, and the word containing the N_F bit.

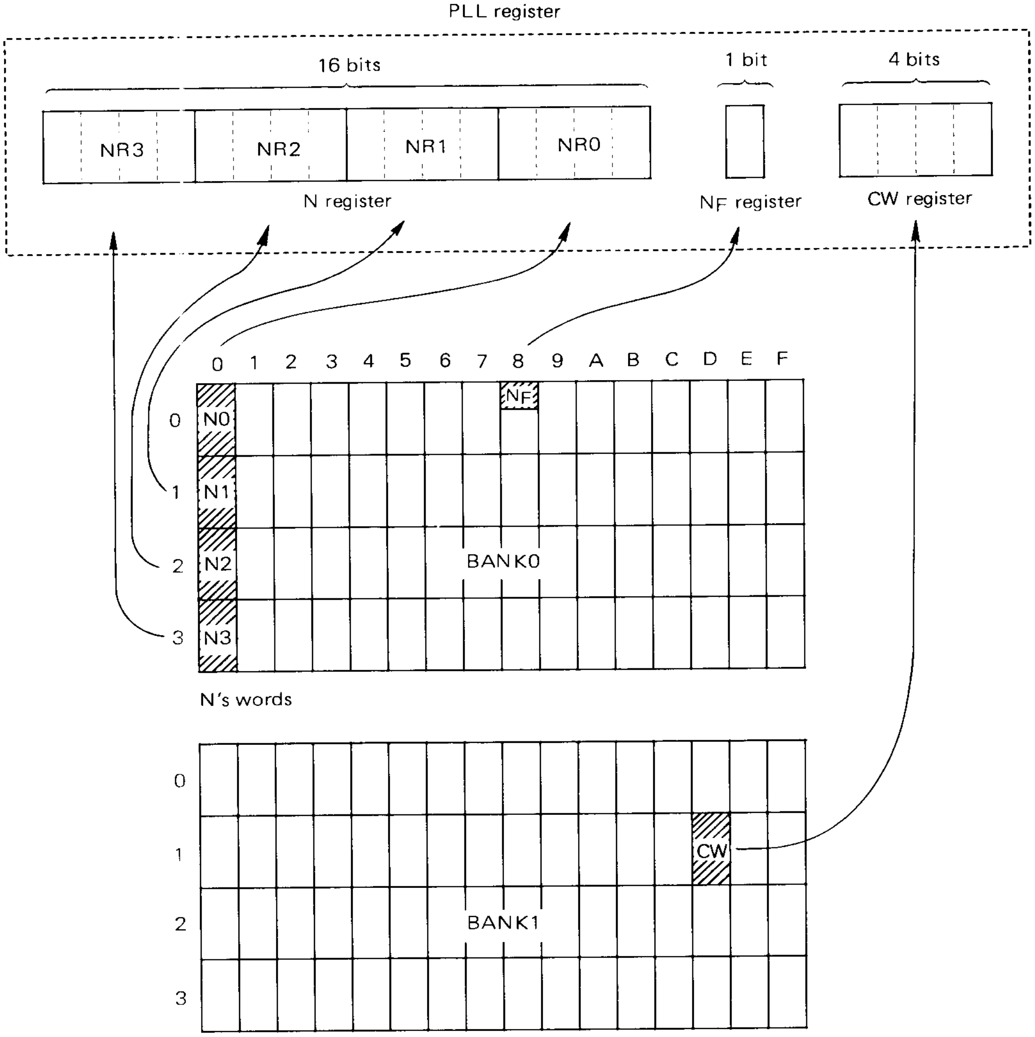
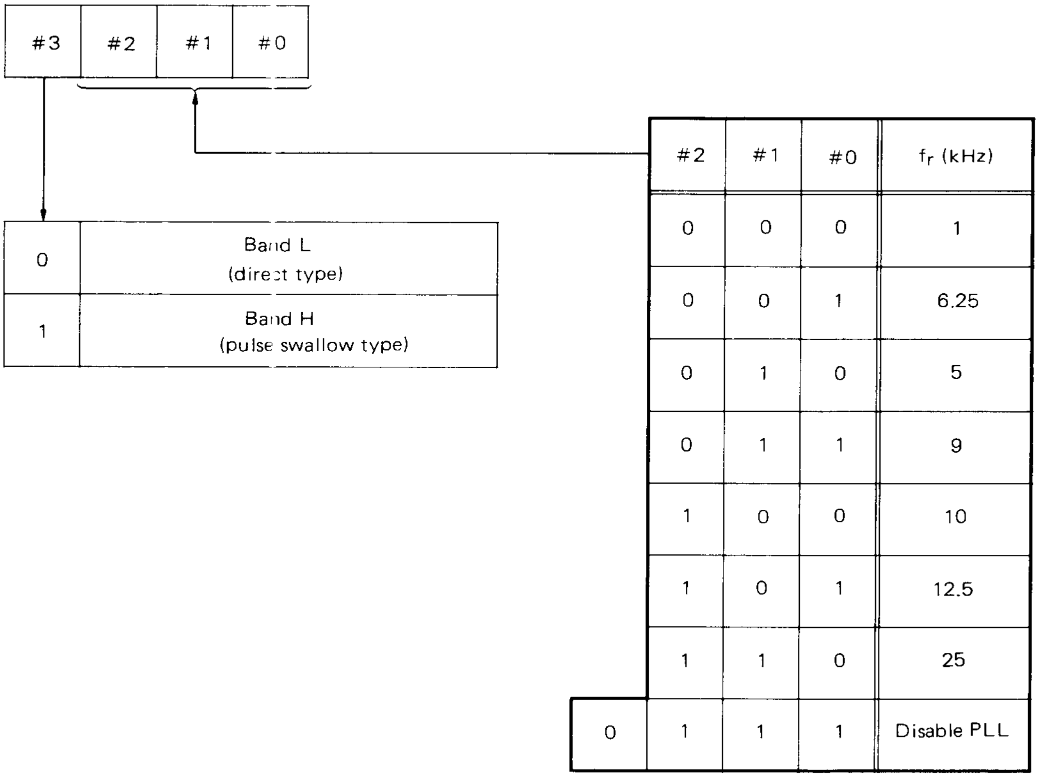


Fig. 7 PLL Instruction Functions

Since the Control Word (CW) is in bank 1 in the example above, a BANK1 instruction must be executed before the PLL instruction. Otherwise, the contents at address 1DH in bank 0 are transferred to the CW register as the CW data, disabling to set the correct reference frequency.

The control word data codes are as follows. The most significant bit (#3) specifies the division method. The three low-order bits (#2 to #0) specify one of the seven reference frequencies.

Table 1 Control Word Codes



#3	#2	#1	#0

#3	
0	Band L (direct type)
1	Band H (pulse swallow type)

	#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

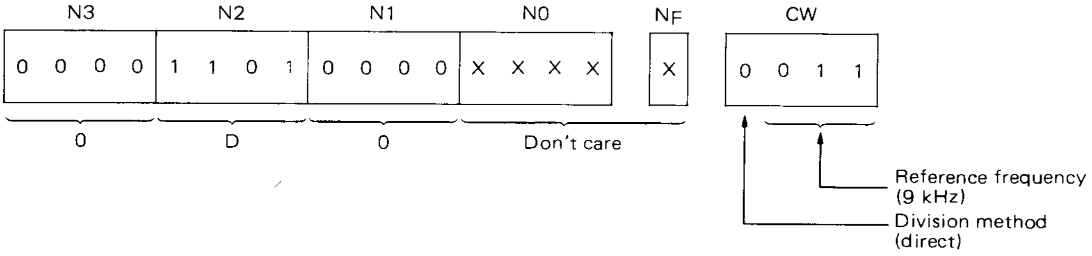
The PLL disable mode can be set by setting 07H in the Control Word (CW) and executing a PLL instruction. By disabling the PLL, the device can be put into a low power consumption mode.

2.5 PLL INFORMATION SETTING

The PLL information (division ratio, reference frequency, and division method) is set by the program. The programmable divider division ratio is set as follows:

Example 1. Direct method (AM)
(Receive frequency: 1422 kHz, reference frequency: 9 kHz, intermediate frequency (IF): 450 kHz)

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



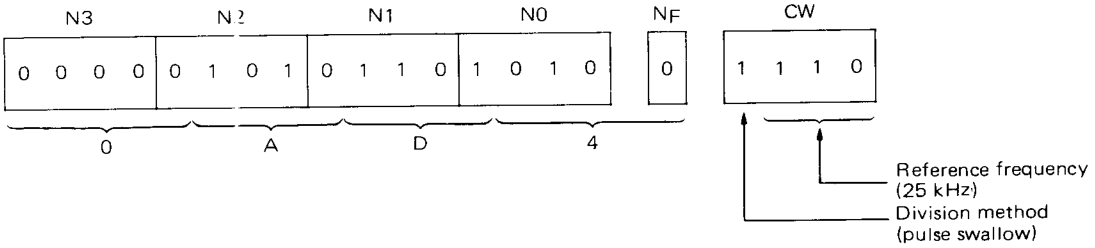
N3	N2	N1	N0	N_F	CW
0 0 0 0	1 1 0 1	0 0 0 0	X X X X	X	0 0 1 1
0	D	0	Don't care		

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

Example 2. Pulse swallcw method (FM)

(Receive frequency: 80 0 MHz, reference frequency: 25 kHz, IF: −10.7 MHz)

$$N = \frac{(80.C - 10.7) \times 10^3}{25} = 2772$$
$$= 0AD4H$$



N3	N2	N1	N0	NF	CW
0 0 0 0	0 1 0 1	0 1 1 0	1 0 1 0	0	1 1 1 0

In this example, the N_F bit is used to change the VCO oscillation frequency by 25 kHz steps. To change the VCO frequency in 50 kHz, 100 kHz, or 200 kHz steps, increment N0 by 1, 2, or 4.

Usually, the N_F bit is used for IF fine tuning. When the N_F bit is set to 1 in the example above, the division ratio N becomes 2773. Assuming that the receive frequency is the same 80.0 MHz, the IF (f_{IF}) is as follows:

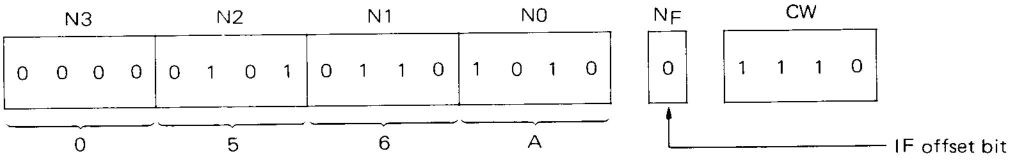
$$\frac{(80.0 - f_{IF}) \times 10^3}{25} = 2773$$
$$f_{IF} = (69.325 - 80.0)$$
$$= -10.675 \text{ MHz}$$

This is equivalent to the IF changed 25 kHz in the case above.

Thus, the N_F bit can be used for IF correction.

The N value is decided by assuming that the N_F bit is the least significant bit in the above example; however, the program would be understood more easily by grouping every four bits from N0. That is, assume that the reference frequency is 50 kHz.

$$N = \frac{(80.0 - 10.7) \times 10^3}{50} = 1386$$
$$= 056AH$$



N3	N2	N1	N0	NF	CW
0 0 0 0	0 1 0 1	0 1 1 0	1 0 1 0	0	1 1 1 0
0	5	6	A		

This is the same as when the reference frequency is 25 kHz.

Example 3. For USA TV band 02 ch

$$N = \frac{f_P + f_{IF}}{P \times f_r} \quad (\text{where, } f_{osc} = f_P + f_{IF})$$

f_p = Picture carrier frequency

f_{IF} = Intermediate frequency

P = Prescaler division ratio

f_r = Reference frequency

N = Programmable divider division ratio

f_{osc} = Local oscillation frequency

When the prescaler is μPB562AC, use the prescaler division ratio p = 8.

$$(f_P = 55.25 \text{ MHz}, \ f_{IF} = 45.75 \text{ MHz}, P = 8, f_r = 5 \text{ kHz})$$

$$N = \frac{(55.25 + 45.75) \times 10^3}{8 \times 5} = 2525$$

$= 09DDH$ (H implies a hexadecimal number.)

N3	N2	N1	N0	N_F	CW
0 0 0 0	0 1 0 0	1 1 1 0	1 1 1 0	1	1 0 1 0
0	9	D	D		Reference frequency (5 kHz)

In this example, the N_F bit is used to change the VCO oscillation frequency $f_{osc} = f_p + f_{IF}$ by 40 kHz ($p \times f_r$) steps. That is, the resolution is the product of the prescaler division ratio and reference frequency.

In the example above, when bit 0 of N0 is changed (N0 is changed by 1), f_{osc} is changed by 80 kHz; when bit 1 is changed (N0 is changed by 2), f_{osc} is changed by 160 kHz; and when bit 2 is changed (N0 is changed by 4), f_{osc} is changed by 320 kHz.

The N value is decided assuming that the N_F bit is the least significant bit in the above example; however, the program would be understood more easily by grouping every four bits from N0. That is, assume that the N_F bit is fixed and change N0. In this case, N can be calculated as follows:

$$N = \frac{(f_P + f_{IF}) - (p \times f_r) \ N_F}{2 \times (p \times f_r)}$$

When the N_F bit is "1", $(p \times f_r) \ N_F$ becomes $(p \times f_r)$, when this bit is 0, this item becomes 0. $2 \times (p \times f_r)$ implies changing 80 kHz since the bits are changed from N0.

The value in Example 1 can be applied to the above expression as follows:

$$N = \frac{(55.25 + 45.75) \times 10^3 - 40}{2 \times 8 \times 5} = 1262$$

$=04EEH$

N3	N2	N1	N0	N_F	CW
0 0 0 0	0 1 0 0	1 1 1 0	1 1 1 0	1	1 0 1 0
0	4	E	E	Set to 1.	

Thus, the programmable divider contents are the same as those in Example 1.

For the USA TV band, the above expression assumes that the N_F bit is always 1 if f_{IF} = 45.75 MHz.

3. PORTS

μPD1707 have I/O port A (PA_3 to PA_0) and output ports B (PB_3 to PB_0) and C (PC_3 to PC_0). Since the output ports are of the N-ch open drain type, they require pull-up resistors.

In addition, there are internal ports E (PE_3 to PE_0), F (PF_3 to PF_0), G (PG_3 to PG_0), and H (PH_3 to PH_0).

Ports E and F (eight bits in total) are used as a data buffer for accessing A/D converter. Ports G and H (eight bits in total) are used as a data buffer for the Clock Generator Port (CGP): during VDP mode 6-bit duty cycle data is stored, during SG mode 6-bit division ratio is stored.

These ports are addressed according to the direct accessing method specified in 2 bits of the instruction operand and the BANK F/F as follows:

Table 2 Port Addressing

Direct Add.		BANK F/F	
#1	#0	BANK 0	BANK 1
0	0	PA	PE
0	1	PB	PF
1	0	PC	PG
1	1	–	PH

To access internal ports E, F, G, and H, set the BANK F/F to specify BANK1. But, RAM bank 0 cannot be accessed while BANK1 is being set. That is, to access RAM BANKO after accessing the port group in BANK1, the BANK F/F setting must be changed to specify BANKO.

Example

```
        MVI     0AH, 1111B  ;  Sets 0FH at address 0AH in the register.
        MVI     0BH, 1101B  ;  Sets 0DH at address 0BH in the register.
FLOOP:
        BANK1
        OUT     3, 0AH      ;  Outputs the contents of address 0AH to PH.
        OUT     2, 0BH      ;  Outputs the contents of address 0BH to PG.
        BANKO               ;  Returns to BANKO after accessing internal ports.
        CAL     WT1SEC      ;  Calls a subroutine to wait 1 second.
        SI      0BH, 0100B  ;
        SIBS    0AH, 0      ;
        JMP     FLOOP       ;
```

In the example above, the CGP is accessed and 64 frequencies divided from reference frequency 18 kHz are individually output 1 second, beginning with the lowest frequency.

3.1 PORT A

Port A (PA_3 to PA_0) can be set to input or output mode by each bit. The input or output mode is determined by the stored in location at address 1FH of data memory (RAM) BANK0, called PAIO word. To set the input mode, write a 0, for output 1, to the corresponding bit of PAIO word.

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

Example 1. Setting PA_3 to PA_0 as an output port.

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

Example 2. Setting PA_3 as an output port and PA_2 to PA_0 as input ports.

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

For Port A, as explained above, an I/O instruction can be executed after setting the I/O mode to the PAIO word. The I/O mode once set remains unchanged unless the PAIO word contents (data at address 1FH) are changed.

Port A automatically enters the input mode when the power is turned on (V_{DD} = low → high), a CKSTP instruction is executed, or the CE pin goes from low to high.

Note that the PAIO word contents may not match port A I/O mode. Subsequently, port A is in the input mode until the PAIO word contents are set.

The PA_3 and PA_2 pins of port A can also be used as serial I/O pins. They operate as shift clock ($\overline{SCK}$) and Serial Input (SI) pins, respectively, during execution of an SIO instruction. **To use these pins as $\overline{SCK}$ and SI pins, it is necessary to set bits 3 and 2 of the PAIO word to 0s, that is, place PA_3 and PA_2 of port A in the input mode.** (See 5. "Serial I/O.")

3.2 PORTS B, C AND D

Port B (PB_3 to PB_0), port C (PC_3 to PC_0) and port D (PD_3 to PD_1) are output ports of the N-ch open drain type. They can be controlled with output instructions (OUT, SPB, and RPB). When an Input Instruction (IN) is executed, the contents of the output latch are read into the specified register. The IN instruction does not change the output latch data.

When "1" is output during output instruction execution, these ports go high (pull-up voltage); when 0 is output, these ports go low (ground voltage).

When the power is turned on (V_{DD} = low → high) or a CKSTP instruction is executed, "1" is output (high-impedance state), that is, the pull-up voltage is output.

At this time the internal output data latch contents do not change. (When the power is turned on (V_{DD} = low → high), the output data latch contents are undefined.) The output data latch contents can be output immediately by executing the following instruction. This is an example for port C.

```
SPB 2,0000B   or   RPB 2,0000B
```

These instructions do not set or reset port C bits.

When the CE pin goes from low to high, ports B and C remain in the same state, not going to the high-impedance state.

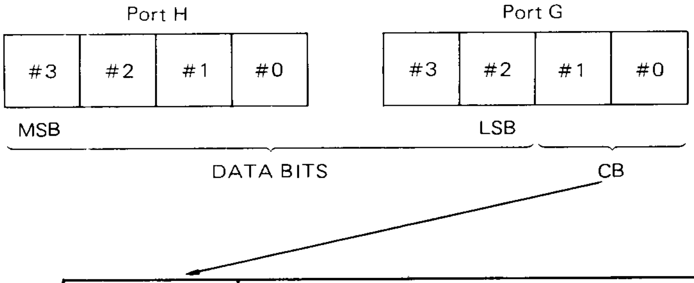
4. CGP

The Clock Generator Port (CGP), controlled by internal ports G and H, has the Variable Duty Pulse (VDP) generation function and Signal Generator (SG) function.

Ports G and H act like other ports, except that they are internal ports, controlled by any port operation instructions. When an IN instruction is executed for ports G and H, the data being set in these ports are read into the specified register.

The CGP can be set to one of four modes specified by bits 0 and 1 of port G, called the Control Bits (CB).

Table 3 Control Bit (CB) Codes and Functions



CB (Control Bits)		Function
#1	#0	
0	0	PG#2 through mode outputs the value of port G bit 2 to the CGP pin.
1	0	VDP mode Variable duty (64 steps) output at frequency 2.69 kHz.
0	1	SG mode 0 Outputs 64 frequencies divided from 18 kHz.
1	1	SG mode 1 Outputs 64 frequencies divided from 180 kHz.

Port H bits 3–0 and port G bits 3–2 are called data bits; they specify the duty value in the VDP mode or the division value in the SG mode. The most significant data bit is port H bit 3 and the least significant data bit is port G bit 2.

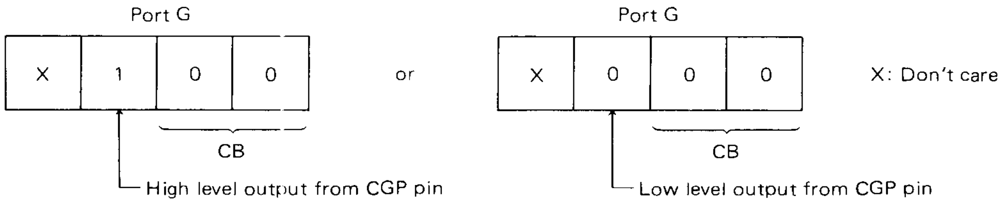
Since the CGP pin is of the N-ch open drain output type, it goes to the high impedance state unconditionally when the power is turned on (V_{DD} = low → high) or a CKSTP instruction is executed. The data latch contents of ports G and H remain unchanged when CKSTP instruction is executed. (The data latch contents are undefined when the power is turned on (V_{DD} = low → high).)

The CGP pin becomes active from the high-impedance state when data in port G are accessed by I/O instructions. The CGP pin can be activated, holding the data latch contents, by using an instruction (SPB 2,0) which sets no bits in port G or (RPB 2,0) which resets no bits. Executing an instruction which operates only port H will not activate the CGP pin from the high-impedance state.

The status of CGP pin remains in the same level when the CE pin level changes. To set the CGP pin to the low level when the CE pin goes from high to low, it is necessary to test the CE pin level with a TCET or TCEF instruction. Then, execute the instruction which places the CGP in the PG #2 through mode, and set CGP pin in the low level according to the test result of CE pin. That is, X000B should be output to port G. (X: don't care)

4.1 PG #2 THROUGH MODE

When CB = 00B, the port G bit 2 data is directly output to the CGP pin, that is, the CGP pin goes high when PG #2 = 1 and low when PG #2 = 0. Thus, the CGP pin can be used as a 1-bit output port.



In this case, the port G bit 3 value is ignored. In the same way, the port H value is ignored, therefore, it is not necessary to output data to the port H in the PG #2 through mode.

4.2 VDP MODE

When CB = 10B, a signal having the duty specified by the data in Ports G and H (PH_3 to PH_0, PG_3 and PG_2) is output continuously to the CGP pin. The output signal frequency is 2.69 kHz.

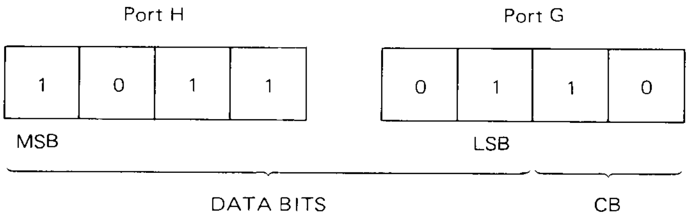
The relationships between the output signal duty and the data set in Ports G and H are as follows:

$$\textbf{Duty} = \frac{\textbf{High level duration}}{\textbf{Cycle}} = \frac{\textbf{(data bits) + 2}}{\textbf{67}}$$

The duty can take 64 values ranging from 2/67 to 65/67.

When the VDP and PG #2 through modes are used in combination, the duty can take 66 values including high and low outputs.

Example



DATA BITS = 101101B
= 2DH
= 45

$$\text{Duty} = \frac{45+2}{67} = \frac{47}{67}$$

As this example shows, port H bit 3 becomes the most significant bit of the data bits and port G bit 2 becomes the least significant bit. The CGP pin starts outputting pulses when 10B is set in the CB, the two low-order bits of port G. For example, if CB = 00B, VDP will be disabled and 2.69 kHz signal will not be output.

Therefore, to set the VDP mode from the high-impedance state, it is necessary first to output data to port H, then to port G. If data is first output to port G, the CGP pin operates according to the current port H data before new data is set to port H. Therefore, the desired duty value cannot be obtained.

If the VDP mode has been set and port G data remains unchanged, data can be output to port H alone or to port H first.

Programming example

```
MVI     0AH, 1011B  ;  Sets the port H data in the register.
MVI     0BH, 0110B  ;  Sets the port G data in the register (VDP mode).
BANK1
OUT     3, 0AH      ;  Outputs data to port H.
OUT     2, 0BH      ;  Outputs data (pulses with duty 64/67) to port G.
BANK0
 .
 .
 .
```

4.3 SG MODE

When the Control Bit (CB) 0 (port G bit 0) is set to 1, the CGP pin enters the Signal Generator (SG) mode. The SG mode outputs signals (duty 50 %) at the frequency specified in the data bits to the CGP pin. Frequencies varying in 128 steps can be output during this mode.

The relationship between the data set in the data bits and the output signal frequency (f_{OUT}) are as follows:

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

where, f_B (base frequency) is the reference frequency to output to the CGP pin. The frequency can be selected between 18 kHz and 180 kHz by the CB 1 (port G bit 1) value.

Port G

		#3	#2	#1	#0
18 kHz (SG mode 0)	:	—	—	0	1

Specifies 18 kHz.

180 kHz (SG mode 1)	:	—	—	1	1

Specifies 180 kHz.

Like the VDP mode, the CGP pin outputs a signal when bit 0 of Port B is set.

Table 4 gives the relation between CGP pin output frequencies and data bits.

Table 4 Output 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	0000	00	4500.000	45.0000	32	1000	00	264.706	2.6471
1	0000	01	3000.000	30.0000	33	1000	01	257.143	2.5714
2	0000	10	2250.000	22.5000	34	1000	10	250.000	2.5000
3	0000	11	1800.000	18.0000	35	1000	11	243.243	2.4324
4	0001	00	1500.000	15.0000	36	1001	00	236.842	2.3684
5	0001	01	1285.710	12.8571	37	1001	01	230.769	2.3077
6	0001	10	1125.000	11.2500	38	1001	10	225.000	2.2500
7	0001	11	1000.000	10.0000	39	1001	11	219.512	2.1951
8	0010	00	900.000	9.0000	40	1010	00	214.286	2.1429
9	0010	01	818.182	8.1818	41	1010	01	209.302	2.0930
10	0010	10	750.000	7.5000	42	1010	10	204.545	2.0455
11	0010	11	692.308	6.9231	43	1010	11	200.000	2.0000
12	0011	00	642.857	6.4286	44	1011	00	195.652	1.9565
13	0011	01	600.000	6.0000	45	1011	01	191.489	1.9149
14	0011	10	562.500	5.6250	46	1011	10	187.500	1.8750
15	0011	11	529.412	5.2941	47	1011	11	183.673	1.8367
16	0100	00	500.000	5.0000	48	1100	00	180.000	1.8000
17	0100	01	473.684	4.7368	49	1100	01	176.471	1.7647
18	0100	10	450.000	4.5000	50	1100	10	173.077	1.7308
19	0100	11	428.571	4.2857	51	1100	11	169.811	1.6981
20	0101	00	409.091	4.0909	52	1101	00	166.667	1.6667
21	0101	01	391.304	3.9130	53	1101	01	163.636	1.6364
22	0101	10	375.000	3.7500	54	1101	10	160.714	1.6071
23	0101	11	360.000	3.6000	55	1101	11	157.895	1.5789
24	0110	00	346.154	3.4615	56	1110	00	155.172	1.5517
25	0110	01	333.333	3.3333	57	1110	01	152.542	1.5254
26	0110	10	321.429	3.2143	58	1110	10	150.000	1.5000
27	0110	11	310.345	3.1034	59	1110	11	147.541	1.4754
28	0111	00	300.000	3.0000	60	1111	00	145.161	1.4516
29	0111	01	290.323	2.9032	61	1111	01	142.857	1.4286
30	0111	10	281.250	2.8125	62	1111	10	140.625	1.4063
31	0111	11	272.727	2.7273	63	1111	11	138.462	1.3846

5. SERIAL I/O

The serial I/O is an 8-bit serial I/O of the COM standard to transfer data in synchronization with the internal or external clock. The serial I/O is associated with three pins:

SI (shared by PA_2): Serial data input pin
SO (shared by PB_0): Serial data output pin
$\overline{SCK}$ (shared by PA_3): Shift clock input/output pin (active low)

Since these pins are shared by other ports (PA_3, PA_2, and PB_0), these pins cannot be used as ports (PA_3, PA_2, and PB_0) when they are used as serial I/O pins.

The 8-bit presettable shift register for the serial I/O data buffer is composed of 2 internal ports. The four high-order bits are allocated to internal port F and four low-order bits are allocated to internal port E. Therefore, data can be written to or read from the presettable shift register with a port operation instruction (such as OUT, SPB, RPB, IN, etc.) which accesses ports E and F.

Ports E and F, the presettable shift register are also used as data latch during A-D conversion. Therefore, serial I/O operation cannot be performed simultaneously with A-D conversion operation.

During A-D conversion operation, inputting the shift clocks to the presettable shift register must be inhibited by resetting SMR1 (bit 1) of the shift mode register.

The serial I/O consists of the Shift Mode Register (SMR), Presettable-Shift Register (PSR), Shift Clock Counter (SCC), and Shift Clock Generator (SCG). They are explained in the following sections.

5.1 SHIFT MODE REGISTER (SMR)

The SMR, consisting of three bits (SMR3, SMR1, and SMR0), determines the serial I/O mode. When an SIO instruction is executed, the immediate data in its operand is written in the SMR. The SMR has no bits associated with operand bit 2. That is, immediate data bit 2 in the operand is ignored during SIO instruction execution.

Executing an SIO instruction (SIO b3 b2 b1 b0 B) sets the following data in the SMR and starts the associated mode operation:

Table 5 SMR Bit Functions

SIO b3 b2 b1 b0 B

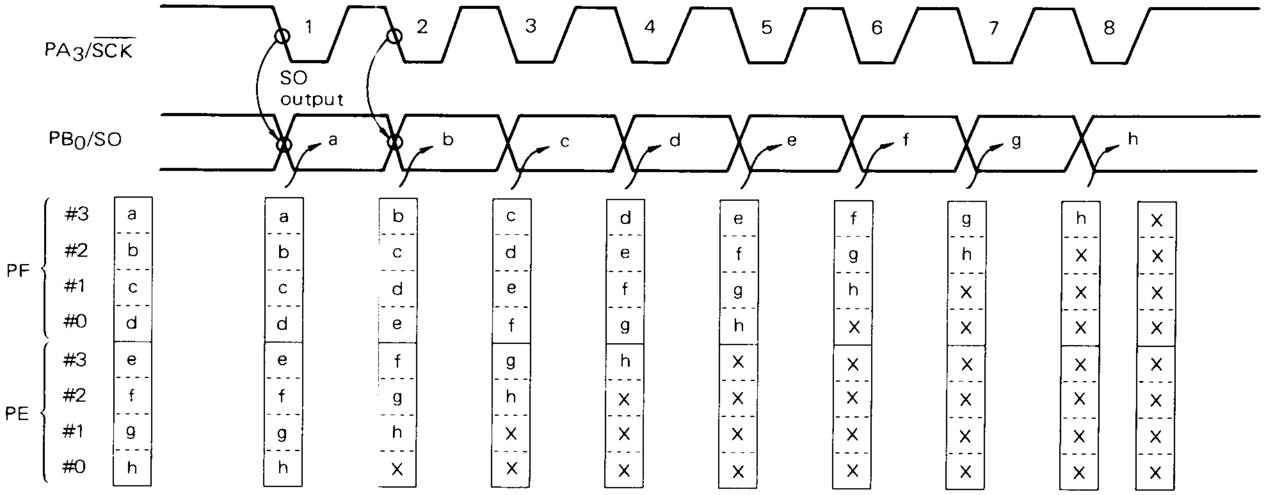
Symbol	Name	Function
SMR0	SO Enable bit	0: Uses PB_0/SO pin as PB_0. 1: Uses PB_0/SO pin as SO.
SMR1	Shift Enable bit	0: Enables shift operation. 1: Disables shift operation.
SMR3	Internal $\overline{SCK}$ Enable bit	0: Enables inputting external clock from the $PA_3/\overline{SCK}$ pin. 1: Outputs internal clock to the $PA_3/\overline{SCK}$ pin.

When the power is turned on (V_{DD} = low → high) or the internal clock is stopped with a CKSTP instruction, the SMR bits are reset to all 0s.

(1) SMR0 (SO enable bit)

After 1 is set in SMR0 (by executing an SIO XXX1B instruction), the PB_0/SO pin operates as a serial data output pin, that is, an SO pin. **Before executing an SIO XXX1B instruction to use the PB_0/SO pin as SO, 1 must be set in PB_0.** Otherwise, the SO pin keeps outputting low-level data.

Executing an SIO XX11B instruction sequentially shifts the presettable shift register (PE and PF bits) data from the most significant bit and outputs the shifted data from the PB_0/SO pin. The PB_0/SO pin outputs data in synchronization with the falling edge of the clock output (or input) from $PA_3/\overline{SCK}$. The presettable shift register contents are also shifted at the rising edge of the clock, and at the same time, this register reads data from the PA_2/SI pin. If the PA_2/SI pin is operating as a PA_2 output pin, the read data is the contents output from the PA_2 pin.



PF #3	a	a	b	c	d	e	f	g	h	X
PF #2	b	b	c	d	e	f	g	h	X	X
PF #1	c	c	d	e	f	g	h	X	X	X
PF #0	d	d	e	f	g	h	X	X	X	X
PE #3	e	e	f	g	h	X	X	X	X	X
PE #2	f	f	g	h	X	X	X	X	X	X
PE #1	g	g	h	X	X	X	X	X	X	X
PE #0	h	h	X	X	X	X	X	X	X	X

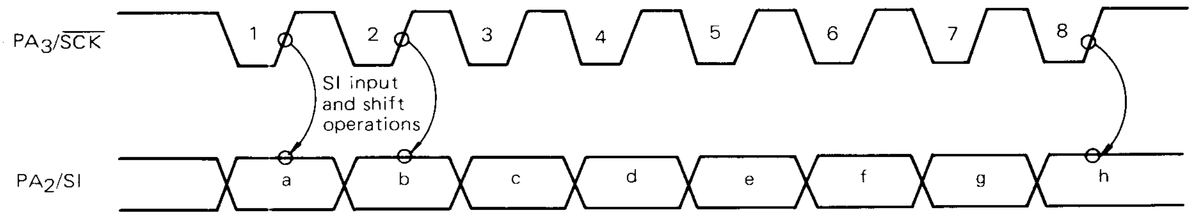
Fig. 8 SO Operation

To use the PB_0/SO pin as PB_0, 0 must be set in SMR0 (by executing an SIO XXX0B instruction). When an instruction (such as OUT, SPB, RPB, etc.) for outputting data to PB_0 is executed while SMR 0 is 1, correct data will not be output.

When SMR0 is reset to 0 after a serial output SO operation, the PB_0/SO pin outputs the contents of the PB_0 data latch before the SO execution, that is, the port B data latch contents.

(2) SMR1 (Shift enable bit)

When 1 is set in SMR1, the PA_2/SI pin contents are read, while being shifted toward the presettable shift register most significant bit (port F bit 3) from the least significant bit (port E bit 0), in synchronization with the clock output (or input) from the $PA_3/\overline{SCK}$ pin. The contents are read at the clock is rising edges.



PF	#3	X	X	X	X	X	X	X	X	a
	#2	X	X	X	X	X	X	X	a	b
	#1	X	X	X	X	X	X	a	b	c
	#0	X	X	X	X	X	a	b	c	d
PE	#3	X	X	X	X	a	b	c	d	e
	#2	X	X	X	a	b	c	d	e	f
	#1	X	X	a	b	c	d	e	f	g
	#0	X	a	b	c	d	e	f	g	h

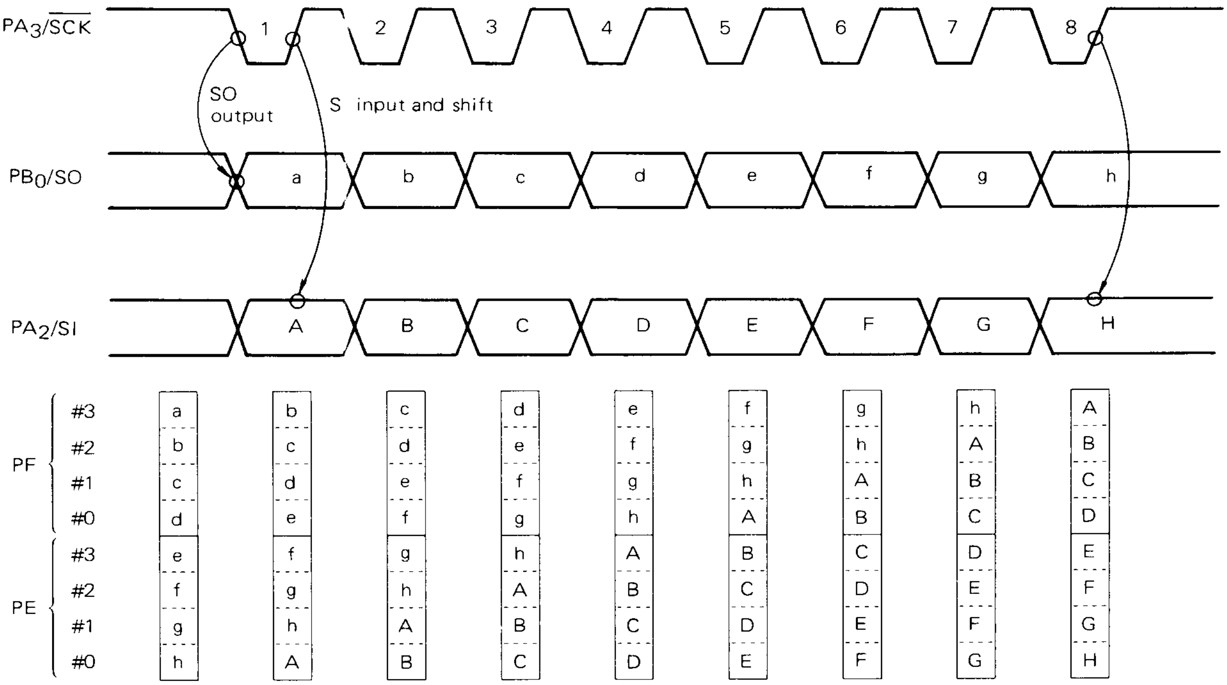
Fig. 9 SI Operation

Before inputting serial data from the PA_2/SI pin, PA_2 must be placed in the input mode, that is, PAIO word bit 2 must be set to 0. If PA_2 is in the output mode, the PA_2 output contents are sequentially read into the presettable shift register (bits PE and PF) while being shifted. The presettable shift register is set to all 1s when PA_2 outputs 1, and to all 0s when PA_2 outputs 0.

During A-D conversion operation, inputting the shift clock to the presettalbe shift register must be inhibited by resetting SMR1 to 0.

When 1 is set in SMR0 (by executing an SIO XX11B instruction) while SMR1 is 1, the PA_2/SI and PB_0/SO pins operate as the SI and SO pins.

At this time, the data from the SO pin is output at the falling edge of the clock output (or input) from the $PA_3/\overline{SCK}$ pin and data is shifted in, via SI pin, at the rising edge of the same clock. That is, serial I/O data is first output from the SO pin from the MSB of Port F, then read from the SI pin to the LSB of Port E synchronized with the shift clock.



PF	#3	a	b	c	d	e	f	g	h	A
	#2	b	c	d	e	f	g	h	A	B
	#1	c	d	e	f	g	h	A	B	C
	#0	d	e	f	g	h	A	B	C	D
PE	#3	e	f	g	h	A	B	C	D	E
	#2	f	g	h	A	B	C	D	E	F
	#1	g	h	A	B	C	D	E	F	G
	#0	h	A	B	C	D	E	F	G	H

Fig. 10 SIO Operation

As shown in the example above, the presettable shift register contents (bits PF and PE) are sequentially shifted, read, and written. When the μCOM standard SI and SO pins are connected to the μPD1711 (or μPD1712) SO and SI pins, the data can be transferred between these devices.

As a general application, serial data can be sent to an external shift register by executing an SIO 1X11B instruction. Executing this instruction sequentially shifts and outputs the presettable shift register contents (bits PF and PE) from the PB0/SO pin, in synchronization with the clock output from the PA3/$\overline{\text{SCK}}$ pin.

The presettable register contents are not shifted if SMR1 is 0. The SO pin outputs the last data obtained by executing the previous SIO instruction. Therefore, SMR1 must be set to 1 before using the serial I/O.

Example

```
SOUT: ANI    PAIO, 0111B ;  Places PAIO#3 in the input mode.
      SPB    PB, 0001B   ;  Sets 1 in PB0.
      BANK1
      OUT    PE, 02H     ;  Sets serial output data.
      OUT    PE, 03H     ;
      SIO    1011B       ;  Outputs data by the internal clock.
      TSET               ;  Waits until eight clocks are output.
      JMP    $-1
      SIO    0000B
      RT
```

(3) SMR3 (internal $\overline{\text{SCK}}$ enable bit)

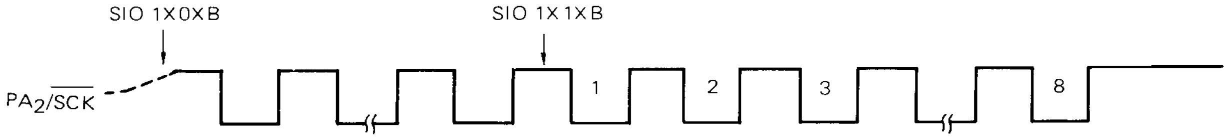
SMR3 selects whether or not the internal or external clock is to be used as the shift clock. Setting 1 in SMR3 causes output of internal c ock (15 kHz, duty 15 %) from the PA$_3$/$\overline{\text{SCK}}$ pin and setting 0 in SMR3 enables external clock input to the PA$_3$/$\overline{\text{SCK}}$ pin.

Before using the PA$_3$/$\overline{\text{SCK}}$ pin as the $\overline{\text{SCK}}$ pin, **that is, executing an SIO instruction, PA$_3$ must be placed in the input mode (PAIO word bit 3 must be set to 0).** At the same time, a pull-up resistor must be connected to the PA$_3$ pin. Note that executing an SIO instruction using the $\overline{\text{SCK}}$ pin while PA$_3$ is in the output mode (PAIO word bit 3 is 1) may damage the device.

When 1 is set in SMR3 with an SIO instruction, the internal clock is immediately generated and output from the PA$_3$/$\overline{\text{SCK}}$ pin. If SMR1 is set to 1 on or before SMR3 is set to 1, clock output stops automatically after outputting eight clock pulses.

The internal shift clock is synchronized with the machine cycle and one clock pulse is equivalent to the duration of executing two instructions (executing an instruction requires 33.3 μs). That is, when an SIO 1X1XB instruction is executed, the clock stops after executing 16 instructions.

Executing an SIO 1X0XB instruction outputs the internal shift clock contiguously, without stopping.



Setting 0 in SMR3 inputs the external clock with frequency 0 (DC) to 200 kHz to the PA$_3$/$\overline{\text{SCK}}$ pin. At this time, the presettable shift register contents are shifted in synchronization with the input external clock. In this mode, the shift operation does not stop automatically after inputting eight external clock pulses. That is, every clock pulse causes a shift operation. Therefore, it is necessary to stop external clock after inputting eight clocks, until the 8-bit data is processed.

After executing an SIO instruction in either the internal or external clock mode, a TSET (Test Shift End, then skip if True) or TSEF (Test Shift End, then skip if False) instruction can be executed to see whether all eight shift operations have been performed.

In the external clock mode, the result of the instruction above is true only when 8 (2n + 1) clock edges have been input, otherwise false. (n is 0 or a positive integer).

The internal clock output does not stop even when the CE pin goes from high to low. It stops in the following cases:

(1) When 0 is set in SMR3 (external clock mode). Note that the PA$_3$/$\overline{\text{SCK}}$ pin enters the high-impedance state (input mode).

(2) When eight shift operations are performed while SMR1 is 1.

(3) When a CKSTP instruction is executed. In this case, SMR is automatically cleared to all 0s.

SMR SUMMARY

SMR0	PB state before SIO instruction execution	PA_0/SO pin state	
0	0	Low-level output	Outputs PB_0 contents
0	1	High-level output	
1	0	Remains on the low level.	
1	1	Operates as an SO pin.	

SMR1	PAIO2 contents before SIO instruction execution	PA_2/SI pin state
0	0	PA_2 input port
0	1	PA_2 output port
1	0	Operates as an SI pin. (The data input to the SI pin is input to the presettable shift register.)
1	1	PA_2 output port. (The data output to PA_2 is input to the presettable shift register.)

SMR3	PAIO3 contents before SIO instruction execution	$PA_3/\overline{SCK}$ pin state
0	0	Operates as a PA_3 input port or enables external clock input. (The external clock can be monitored by testing PA_3.)
0	1	PA_3 output port
1	0	Internal clock output (The internal clock can be monitored by testing PA_3.)
1	1	Inhibited. (This may damage the device.)

Operation		SMR			PB_0 contents before SIO instruction execution	PAIO2 contents before SIO instruction execution	PAIO3 contents before SIO instruction execution	PB_0/SO pin	PA_2/SI pin	$PA_3/\overline{SCK}$ pin
		3	1	0						
SI operation	Outputs internal clock.	1	1	0	X	0	0	PB_0	SI	$\overline{SCK}$ output
	Inputs external clock.	0								$\overline{SCK}$ input
SO operation	Outputs internal clock.	1	1	1	1	X	0	SO	PA_2	$\overline{SCK}$ output
	Input external clock.	0								$\overline{SCK}$ input
SIO operation	Outputs internal clock.	1	1	1	1	0	0	SO	SI	$\overline{SCK}$ output
	Input external clock.	0								$\overline{SCK}$ input
Outputs only internal clock contiguously (no shift operation).		1	0	0	X	X	0	PB_0	PA_2	$\overline{SCK}$ output
SIO function unused (used as a port).		0	0	0	X	X	X	PB_0	PA_2	PA_3

X : Don't care.

* : This state is set when the power is turned on (V_{DD} = low → high) or when the clock restarts after it was stopped by a CKSTP instruction.

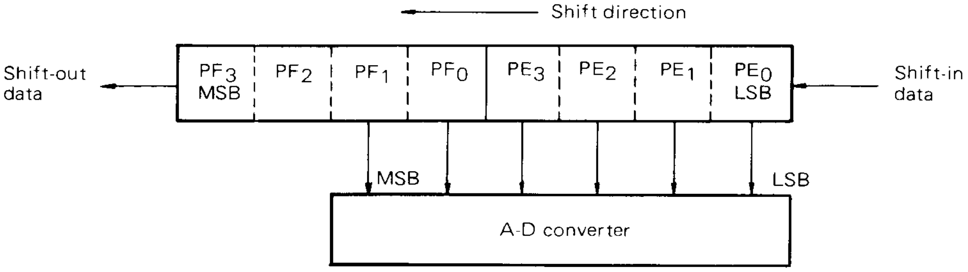
5.2 PRESETTABLE SHIFT REGISTER (PSR)

The PSR is an 8-bit shift register. Its four high-order bits are allocated to internal port F and the four low-order bits are allocated to internal port E. A port operation instruction which accesses ports E and F can write data in or read data from the PSR. An OUT or SPB/RPB instruction is necessary to write data and an IN or TPT/TPF instruction is necessary to read data.

The internal or external clock can be selected with SMR3 as the shift clock for the PSR shift operation. The shift operation with the clock input to or output from the $PA_3/\overline{SCK}$ pin is enabled only while SMR1 is 1. While SMR1 is 0, shift clock input to the PSR is inhibited and no shift operation is performed.

The PSR is used also for A-D converter data latch. **Therefore, the shift operation must be inhibited by setting SMR1 to 0 during A-D conversion operation.**

The shift operation is performed from the most significant bit to the least significant bit. The shift-out data is output at the falling edge of the shift clock; at the rising edge, the PSR data is shifted in.



Note: Data can be set in the PSR only while SMR1 = 0 or while SMR = 1 and the $PA_3/\overline{SCK}$ pin is on the high level. Otherwise, data cannot be set correctly in the PSR.

5.3 SHIFT CLOCK COUNTER (SCC)

The SCC is a 4-bit binary counter which counts the shift clock input to the PSR while SMR1 is 1, that is, the PSR shift operation is enabled. When the counter value is 8 (1000B), 1 is output to the CPU Judge, otherwise, 0 is output.

The Judge output can be tested with a TSET (Test Shift End, skip if True) instruction or a TSEF (Test Shift End, skip if False) instruction. The result of this instruction is true when the counter value is 8 and false when not.

The SCC contents are cleared to 0s only in the following cases:

(1) When the power is turned on (V_{DD} = low → high)
(2) When the internal clock is stopped with a CKSTP instruction
(3) When an SIO instruction is executed

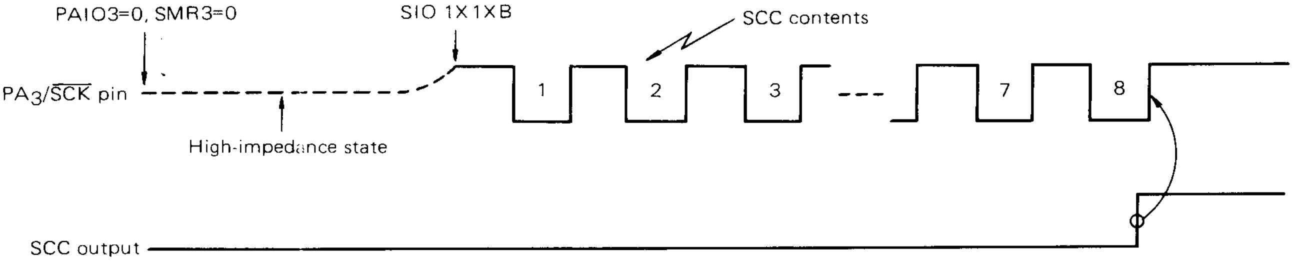
When the internal clock is used as the shift clock, the above Judge output stops the internal clock, allowing only eight data bits to be sent. Unless the SCC is cleared, the subsequent result of TSET or TSEF instruction is always true. When the external clock is being used, shift clock input is not inhibited after inputting eight clock pulses. **Therefore, external clock input must stop after inputting eight clock pulses.**

The Judge output is tested true input only when 8 (2n + 1) clock pulses have been input, otherwise false (n is 0 or a positive integer). That is, the SCC outputs 1 to the Judge input only when the SCC bit 3 is changed from 0 to 1.

5.4 SHIFT CLOCK GENERATOR (SCG)

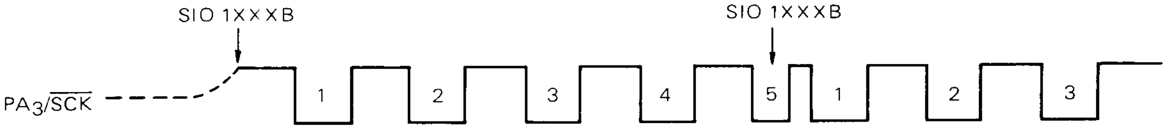
The SCG is an internal clock generator which generates a clock (15 kHz, duty 50 %) from the PA3/$\overline{SCK}$ pin when 1 is set in SMR3. The output clock is synchronized with the machine cycle. One clock pulse is equivalent to the duration of executing two instructions (executing an instruction requires 33.3 μs).

When 1 is set in SMR3, the SCG outputs a clock which changes from high to low level. The SCG will remain at high level if SCC generates a true judge output. (8 Clocks)

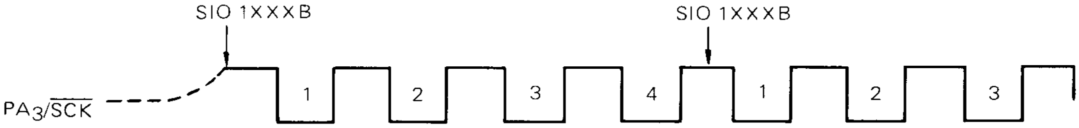


When 0 is set in SMR1 (shift operation inhibited) with an SIO 1X0XB instruction, the clock output does not stop automatically after outputting eight clock pulses.

Each time an SIO instruction is executed, the SCG starts clock output from the high level. Therefore, if an SIO 1XXXB instruction is executed again after an SIO 1XXXB instruction, the clock duty may be changed from 50 % according to the execution timing.



To prevent this, test the PA3/$\overline{SCK}$ pin as PA3 by executing a TPT or TPF instruction, then execute an SIO 1XXXB instruction after n instruction from the time a high level is judged, n being an even number, that is, execute this instruction while the clock level is high.



The PA3/$\overline{SCK}$ pin, must be set to input mode before executing an SIO instruction and it remains in the high-impedance state until an SIO instruction is executed. Therefore, a pull-up resistor must be connected to the PA3/$\overline{SCK}$ pin before using the serial I/O.

5.5 EXPANDED SERIAL I/O APPLICATIONS

(1) Example of sending data of eight or more bits by internal clock

Data of eight or more bits can be sent successively by the internal clock by setting 4-bit data sequentially in the PSR and executing SIO instructions.

An example of sending 12-bit data is given below.

First set 8-bit data in the PSR, then execute an SIO 1X11B instruction. Since the internal clock (PSR shift clock) output from the $PA_3/\overline{SCK}$ pin is synchronized with the instruction cycle as explained before, the first fourth bits of the PSR contents are shifted by the seventh instruction counted from the one succeeding the SIO instruction. Set the remaining 4-bit data in the four low-order bits of the PSR, that is port E, by using an OUT instruction as the seventh instruction. Execute an SIO instruction again as the eighth instruction to clear the SCC contents. Thus, the next 8-bit data can be sent without stopping the clock.

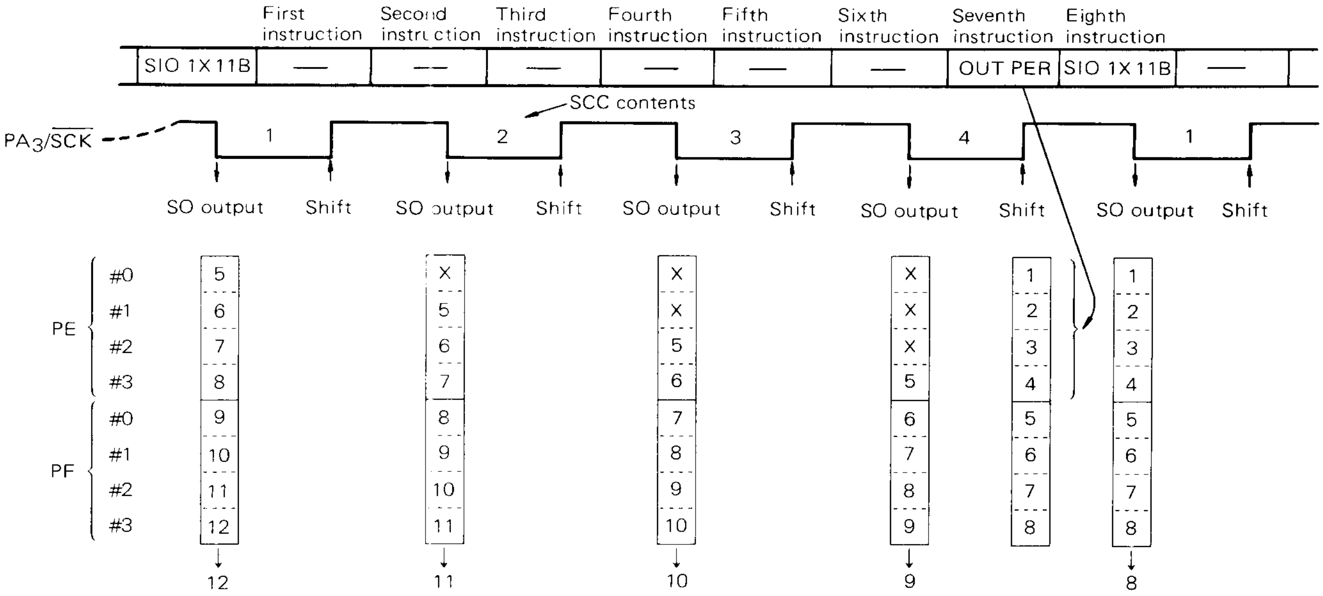


Fig. 11 **Example of Sending 12-Bit Data**

In the same way, to send n-bit data ($8 < n \leq 12$), execute an OUT instruction port E as the (n–8)x2–1st instruction counted from the one succeeding the first SIO instruction, then execute an SIO instruction again as the (n–8)x2nd instruction.

Data of more than 12 bits can also be sent in the same way, by executing OUT and SIO instructions alternately.

(2) Example of receiving data of eight or more bits by internal clock

Data of eight or more bits can be received successively by the internal clock in the same way as sending explained in (1) above. The IN instruction execution timing differs from that of an OUT instruction for sending.

An example of receiving 12-bit data is given below.

Before executing the first SIO 1X1XB instruction, set 0s in PAIO word bits 2 and 3. After 4-bit serial data is input to the PSR, execute an SIO instruction again as the eighth instruction counted from the one succeeding the first SIO instruction to clear the SCC contents. Execute an IN instruction from Port E as the ninth instruction (four low-order bits of the PSR) to store the first 4-bit data in the RAM. When the second shift operation stops, read the contents of ports E and F to complete 12-bit serial data reception.

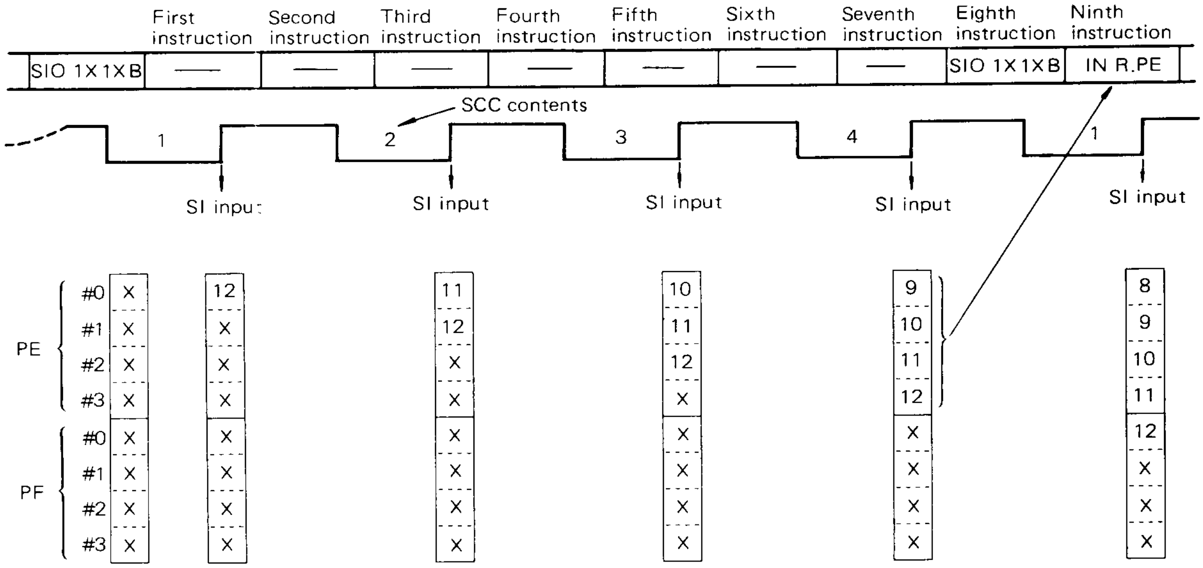


Fig. 12 Example of Receiving 12-Bit Data

To receive n-bit data (8 < n ≦ 12), execute an SIO instruction again as the (n–8)x2nd instruction counted from the one succeeding the first SIO instruction. Then execute an IN instruction as the (n–8)x2+1st instruction for port E.

Data of more than 12 bits can also be sent in the same way, by executing SIO and IN instructions alternately.

Note: As shown in the two examples above, data of eight or more bits cannot be sent and received at the same time. An OUT instruction must be executed as the (n–8)x2–1st instruction for sending and an IN instruction must be executed as the (n–8)x2+1st instruction for receiving. Therefore, when sending and receiving are performed simultaneously, the first 4-bit data read is deleted with an OUT instruction. As the result, executing an IN instruction reads the data written by an OUT instruction, not the shift-in data.

6. A-D CONVERTER

μPD1707 have a built-in 6-bit A-D converter that employs a successive approximation method.

The A-D converter for FM/AM tuner can be used digitized signal meter data, output from the IF detection stage. It can measure the electric field ntensity or determine the stop level during automatic tuning. The A-D converter for TV application can input an S-shaped curve signal from the Video Intermediate Frequency (VIF) section. It can measure the signal to determine the optimum receiving frequency during Automatic Fine Tuning (AFT). The AD_0 to AD_2 pins can be selected arbitrarily by the program as 1-bit ports to set different threshold level.

6.1 OPERATION

The A-D converter consists of a 6-bit D-A converter and comparator.

The 6-bit data is set in the D-A converter via the two low-order bits (PF_1 and PF_2) of internal port F and bits PE_3 to PE_0 of internal port E. That is, the comparison data is set by executing an output instruction (OUT, SPB, or RPB) for ports E and F and it is read by executing an input instruction (IN, TPT, or TPF).

The two high-order bits (PF_3 and PF_2) of port F are used to select any one of the three A-D input channels.

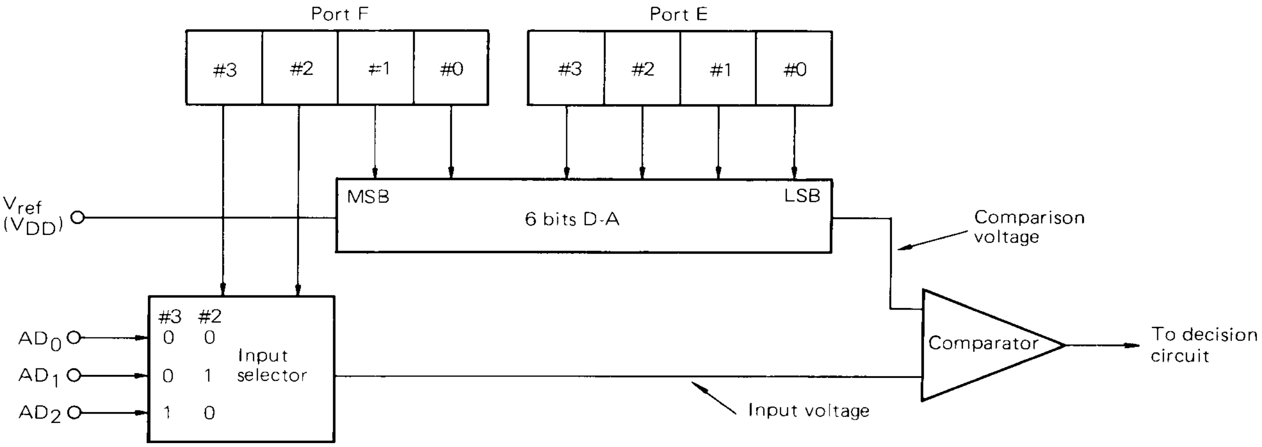


Fig. 13 A/D Converter Configuration

Since ports E and F are also used for during serial I/O operation, **SMR1 must remain 0 (shift operation inhibit mode) during A-D conversion operation.**

The D-A converter generates 64 voltage levels (divided from reference voltage V_{ref} (V_{DD})) according to the data set in ports E and F as the A-D comparison voltage. This comparison voltage is input to the comparator together with the analog voltage (input voltage) from one of the A-D pin. Thus, both voltage levels are compared. The comparison result can be tested by executing a TADT (Test A-D comparator, then skip if True) or TADF (Test A-D comparator, then skip if False) instruction. The results of the input and comparison voltages of the comparator are as follows:

Input voltage $>$ comparison voltage . . . . True

Input voltage $\leq$ comparison voltage . . . . False

6.2 D-A CONVERTER CONFIGURATION

The D-A converter is of the resistor string type which selects one of the connecting points of the 64 resistors serially connected between V_{ref} and GND.

Note that the serially connected resistors have different resistances. The resistor closest to the GND has resistance 0.5/64 and that closest to V_{ref} has the resistance 1.5/64 to the total series resistance. Figure 14 shows the D-A converter structure.

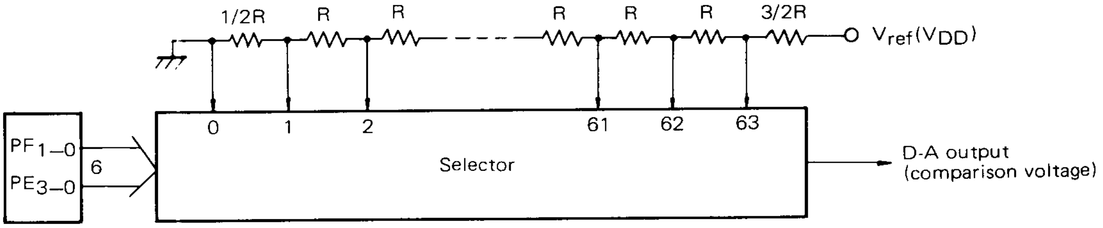


Fig. 14 D-A Converter Configuration

When 00H is set as data in ports E and F, this D-A converter outputs the GND level; when 01H is set, it outputs voltage 0.5/64 × V_{ref} as the comparison voltage. Likewise, when decimal n is set, comparison voltage V_{out} is as follows:

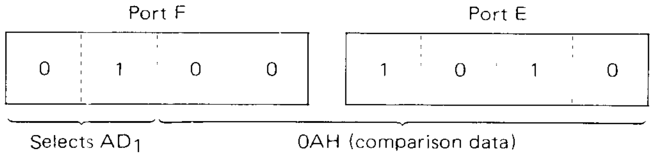
$$V_{out} = V_{ref} \times \frac{n-0.5}{64} \quad (63 \geqq n \geqq 1)$$

If the input voltage V_{in} and the comparison voltage V_n for data n set in the D-A converter are $V_n < V_{in} \leqq V_{n+1}$, the A-D converter employing successive approximation outputs n as the conversion data.

Table 6 Relation of Comparison Data and Comparison Voltage

Comparison data (set in ports E and F)		Comparison voltage		Comparison data (set in ports E and F)		Comparison voltage	
DEC.	HEX (Note)	× V_{ref}	V_{ref} = 5 V	DEC.	HEX (Note)	× V_{ref}	V_{ref} = 5 V
0	0 0 H	0	0 V	32	2 0 H	31.5/64	2.461
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 A H	9.5/64	0.742	42	2 A H	41.5/64	3.242
11	0 B H	10.5/64	0.820	43	2 B H	42.5/64	3.320
12	0 C H	11.5/64	0.898	44	2 C H	43.5/64	3.398
13	0 D H	12.5/64	0.977	45	2 D H	44.5/64	3.477
14	0 E H	13.5/64	1.055	46	2 E H	45.5/64	3.555
15	0 F H	14.5/64	1.133	47	2 F H	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 A H	25.5/64	1.992	58	3 A H	57.5/64	4.492
27	1 B H	26.5/64	2.070	59	3 B H	58.5/64	4.570
28	1 C H	27.5/64	2.148	60	3 C H	59.5/64	4.648
29	1 D H	28.5/64	2.227	61	3 D H	60.5/64	4.727
30	1 E H	29.5/64	2.305	62	3 E H	61.5/64	4.805
31	1 F H	30.5/64	2.383	63	3 F H	62.5/64	4.883

Note: The hexadecimal codes listed in this table consist of the two low-order bits of port F and the four bits of port E. The two high-order bits of port F are ignored.



Assume that V_{ref} = 5.0 V and 0.0781 V (= $V_{ref}/64$) is input to V_{in}. The D-A converter output voltages (comparison voltages) V_0, V_1, and V_2 when n = 0, 1, and 2 are:

$$V_0 = 0 \text{ V}$$
$$V_1 = 0.03906 \text{ V}$$
$$V_2 = 0.11719 \text{ V}$$

Therefore, the A-D converter comparison data is 1. Since the input analog voltage V_{in} 0.0781 V is in the range 0.03906 V $< V_{in} \leqq$ 0.11719 V. When input voltage V_{in} is in the range of $V_0 < V_{in} \leqq V_1$, 0 V is assumed to be input that is converted data is 0. Input voltage $V_{in} = V_{ref}/64$ (1LSB) is called resolution voltage.

Since input voltage V_{in} in the range of 0.03906 V $< V_{in} \leqq$ 0.11719 V is assumed to be 0.0781 V, as explained above, it causes a read error of at least ±0.03906 V for the actual input voltage. Generally, such error in the A-D converter is called a quantization error, and is expressed as ±1/2 LSB. It is also represented as ±1/ (64x2) = ±0.781% for a 6-bit A-D converter. The maximum converter error is ±1 LSB.

This A-D converter accepts input voltage V_{in} in the following range:

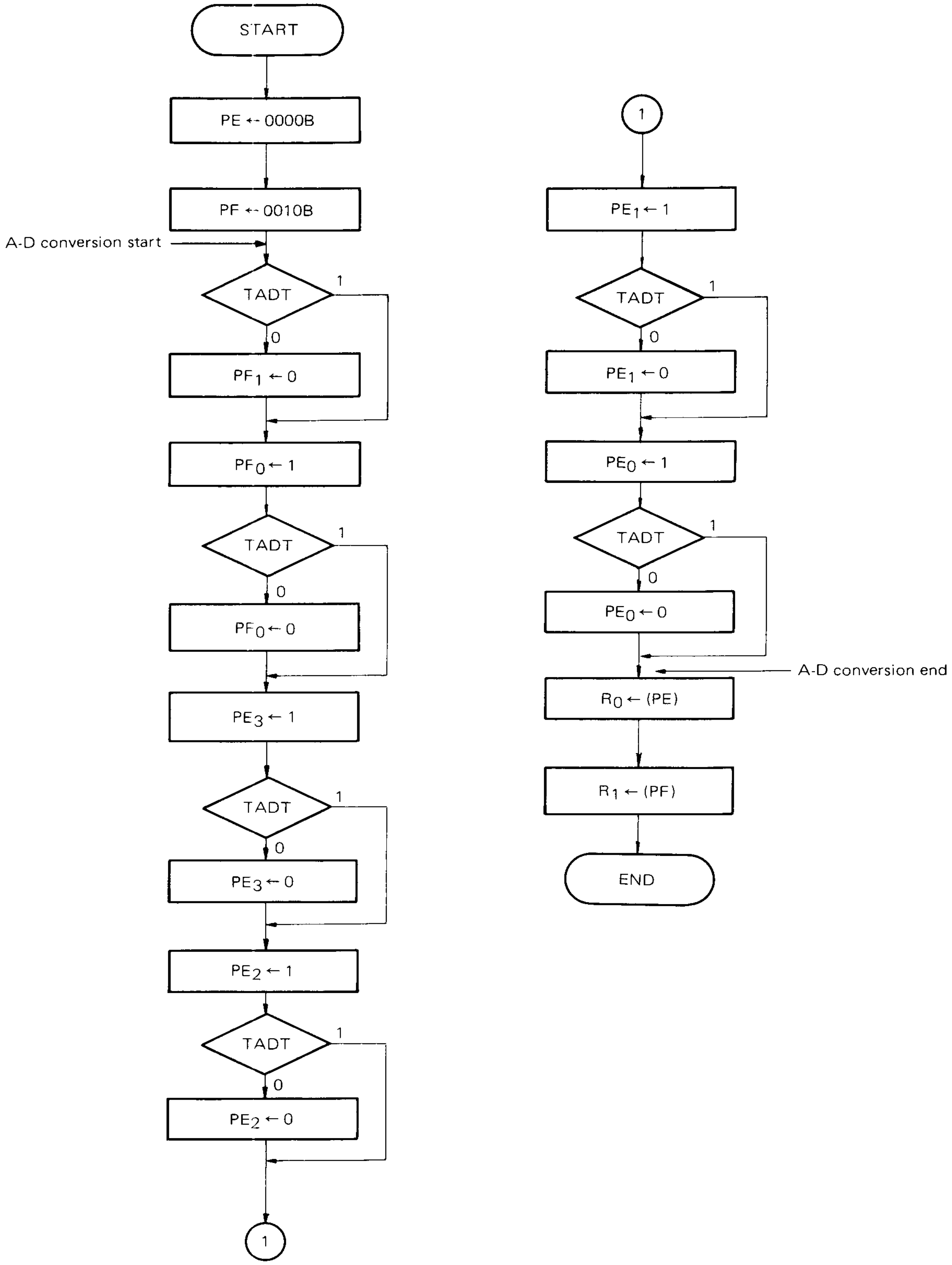
$$0 \text{ V} \leqq V_{in} \leqq V_{63} \ (V_{ref} \times 62.5/64)$$

If V_{in} is outside the range of $V_{63} < V_{in} \leqq V_{ref}$ (V_{DD}), an OVER RANGE error occurs.

A-D conversion requires executing a TADT or TADF instruction as explained above. For example, when analog voltage $V_{in} = V_{14}$ ($V_{ref} \times$ 13.5/64) is input and V_{13} is output as the comparison voltage, a true result is output to the decision circuit because the input voltage > comparison voltage. When V_{14} is input as the comparison voltage, a false result is output because the input voltage $\leqq$ comparison voltage.

6.3 A-D CONVERSION PROGRAM EXAMPLE

Binary search



Coding Example

```
START:
        MVI     R0, 0000B ;  Sets PE data.
        MVI     R1, 0010B ;  Sets PF data.
        BANK1
        OUT     1, R1     ;  PF ← 02H
        OUT     0, R0     ;  PF ← 01H
        TADT
        RPB     1, 0010B  ;  D-A data   MSB #5 ← 0
        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)
END:
```

A-D conversion time: 832.5 μs

Total number of steps: 25

7. PROGRAMMABLE LOGIC ARRAY (PLA)

μPD1707 have segment PLA and digit PLA for user programs. Usually, dynamic display patterns and key return signal source patterns are programmed in these PLAs. The segment PLA can generate 32 different patterns and the digit PLA can generate 16 different patterns.

7.1 SEGMENT PLA CONFIGURATION

The segment PLA consists of a 5-bit segment latch circuit for addressing and a PLA with 7-bit outputs to which the segment pins (S_a to S_g).

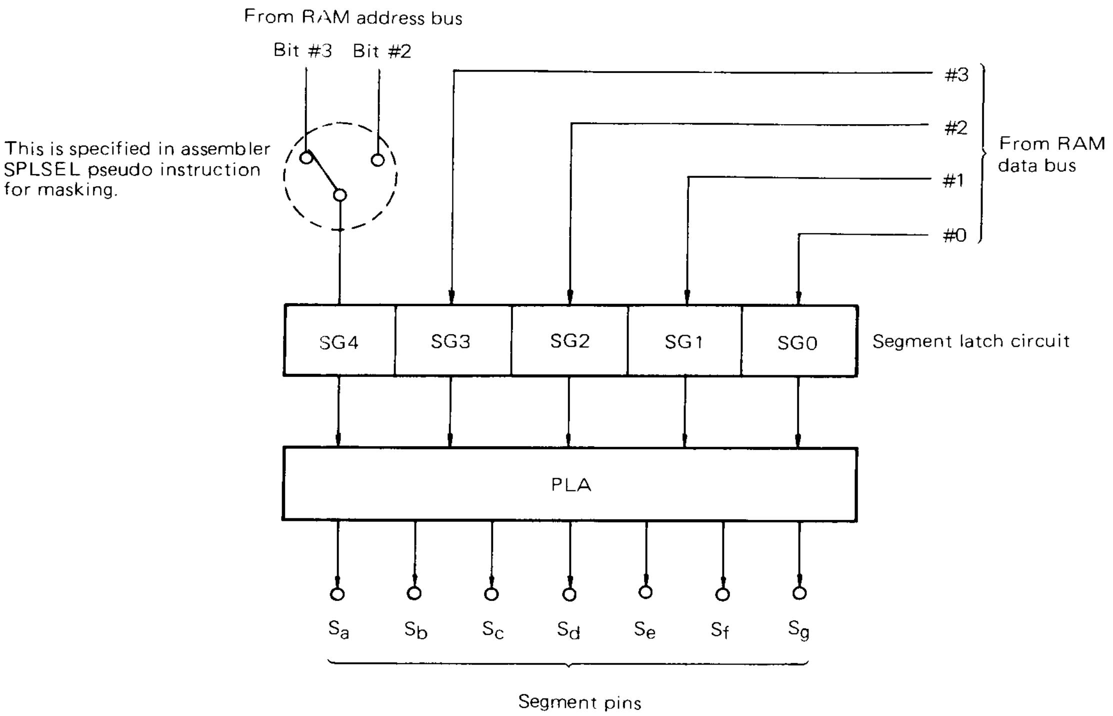
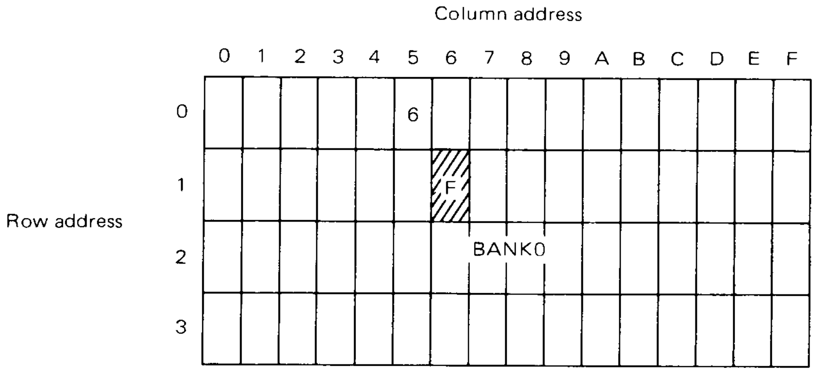


Fig. 15 Segment PLA Configuration

The contents of the data memory (RAM) specified in the an SEG instruction operand are latched in the four low-order bits (SG0 to SG3) of the segment latch circuit. For example, executing an SEG 1, 05H instruction latches the RAM contents addressed with 1 (specifying the RAM row address) and register 05H contents (specifying the RAM column address), that is, data "F" at address 16H.



The value of bit 3 or 2 at the RAM column address specified in the SEG instruction operand is latched in the most significant bit of the segment latch circuit SG4. When ordering the mask, bit 3 or 2 must be specified. (See the PLA program example in Section 7.4) When bit 3 has been specified, 0 is latched in SG4 if RAM addresses 00H to 07H are specified in the SEG instruction and 1 is latched if RAM addresses 08H to 0FH are specified. When bit 2 has been specified, 0 is latched if RAM addresses 00H to 03H or 08H to 0BH are specified and 1 is latched if RAM addresses 04H to 07H or 0CH to 0FH are specified.

The 32 segment patterns are divided into two groups of 16-patterns groups according to the data latched in SG4. Therefore, even when the RAM data is the same, different segment patterns can be output from the segment pins if different column addresses are specified in the SEG instructions.

The 16-pattern group generated when SG4 is 0 is called pattern group 0, and that generated when SG4 is 1 is called pattern group 1.

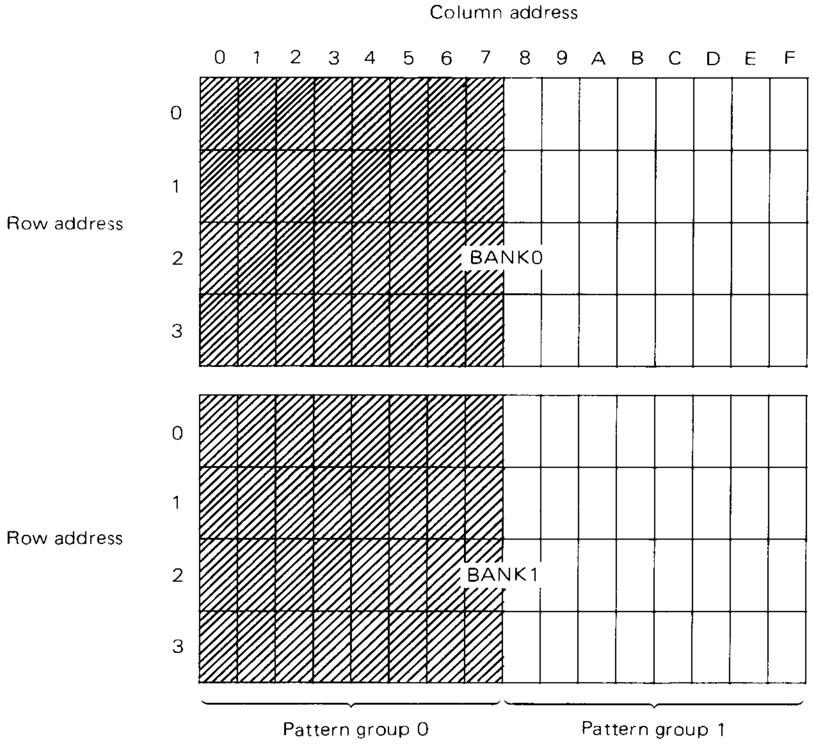


Fig. 16 Pattern Grouping When Bit 3 is Specified for SG4 Input

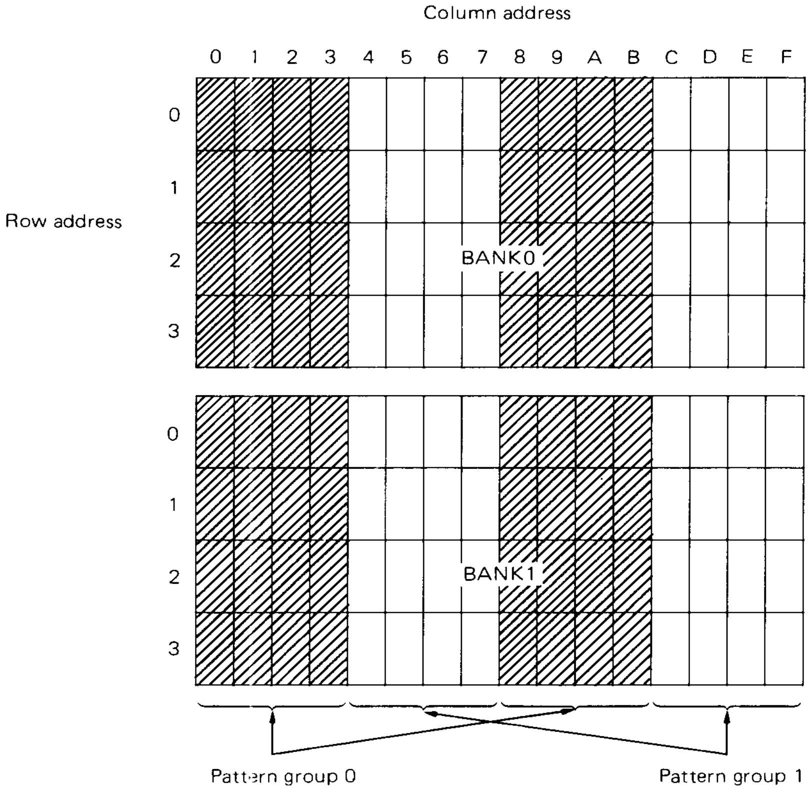


Fig. 17 Pattern Grouping When Bit 2 is Specified for SG4 Input

The pattern grouping is decided considering the RAM or program (ROM) efficiency during programming. SPLSEL must be used to select bit 3 or 2 for SG4 input in the following format. (See the PLA program example in Section 7.4.)

SPLSEL 3   or   SPLSEL 2

7.2 SEGMENT PLA PATTERN EXAMPLES

Tables 7 and 8 list the pattern examples in pattern groups 0 and 1, respectively.

Table 7 Pattern Examples in Pattern Group 0

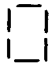
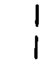
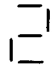
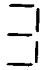
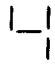
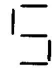
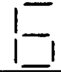
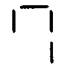
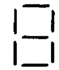
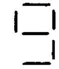
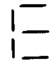
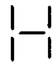
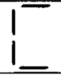
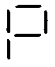
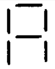
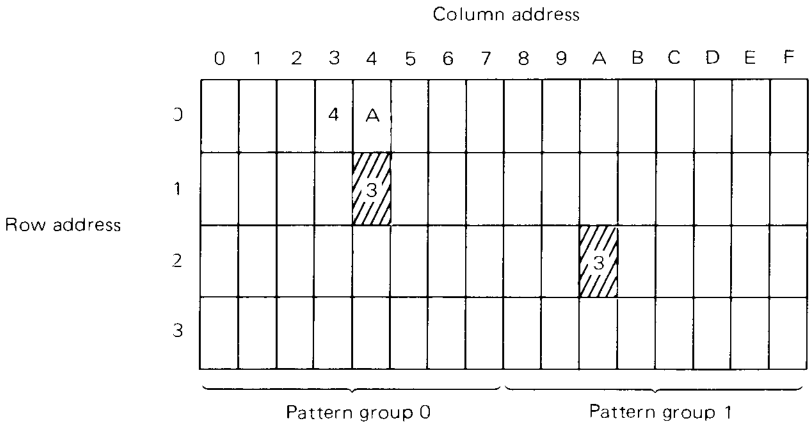
Segment latch					Segment output							Output pattern
SG4	SG3	SG2	SG1	SG0	g	f	e	d	c	b	a	
0	0	0	0	0	0	1	1	1	1	1	1	
	0	0	0	1	0	0	0	0	1	1	0	
	0	0	1	0	1	0	1	1	0	1	1	
	0	0	1	1	1	0	0	1	1	1	1	
	0	1	0	0	1	1	0	0	1	1	0	
	0	1	0	1	1	1	0	1	1	0	1	
	0	1	1	0	1	1	1	1	1	0	1	
	0	1	1	1	0	1	0	0	1	1	1	
	1	0	0	0	1	1	1	1	1	1	1	
	1	0	0	1	1	1	0	1	1	1	1	
	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	
	1	1	0	0	1	1	1	0	1	1	0	
	1	1	0	1	0	1	1	1	0	0	1	
	1	1	1	0	1	1	1	0	0	1	1	
	1	1	1	1	1	1	1	0	1	1	1	

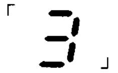
Table 8 Pattern Examples in Pattern Group 1

Segment latch					Segment output							Output pattern
SG4	SG3	SG2	SG1	SG0	g	f	e	d	c	b	a	
1	0	0	0	0	0	0	0	0	0	0	0	Key all OFF
	0	0	0	1	0	0	0	0	0	0	1	Seg a (Key)
	0	0	1	0	0	0	0	0	0	1	0	Seg b (Key)
	0	0	1	1	0	0	0	0	1	0	0	Seg c (Key)
	0	1	0	0	0	0	0	1	0	0	0	Seg d (Key)
	0	1	0	1	0	0	1	0	0	0	0	Seg e (Key)
	0	1	1	0	0	1	0	0	0	0	0	Seg f (Key)
	0	1	1	1	1	0	0	0	0	0	0	Seg g (Key)
	1	0	0	0	1	1	1	1	1	1	1	Key all ON
	1	0	0	1	1	0	0	0	0	1	0	FM, MHz
	1	0	1	0	0	1	0	0	0	0	1	MW, kHz
	1	0	1	1	0	0	1	0	0	0	1	LW, kHz
	1	1	0	0	1	0	0	1	0	1	0	FM, ME, MHz
	1	1	0	1	0	1	0	1	0	0	1	MW, ME, kHz
	1	1	1	0	0	0	1	1	0	0	1	LW, ME, kHz
	1	1	1	1	0	0	0	0	0	0	0	No use

When column address bit 3 has been specified as the SG4 input to the segment latch circuit, executing an SEG instruction with the above PLA patterns will output the following Data:



(1) SEG 1,03H: Latches the contents (0011B) at RAM address 14H.

Output pattern: 

(2) SEG 2,04H: Latches the contents (0011B) at RAM address 2AH.
Output pattern: ⌜SEG C⌟

Even if the same data is set in the four low-order bits of the segment latch circuit, different patterns can be output by changing the SG4 value.

7.3 DIGIT PLA CONFIGURATION

The digit PLA consists of a 4-bit digit latch circuit and a PLA with six digit pins (D_1 to D_6) to which the digit latch circuit data is output.

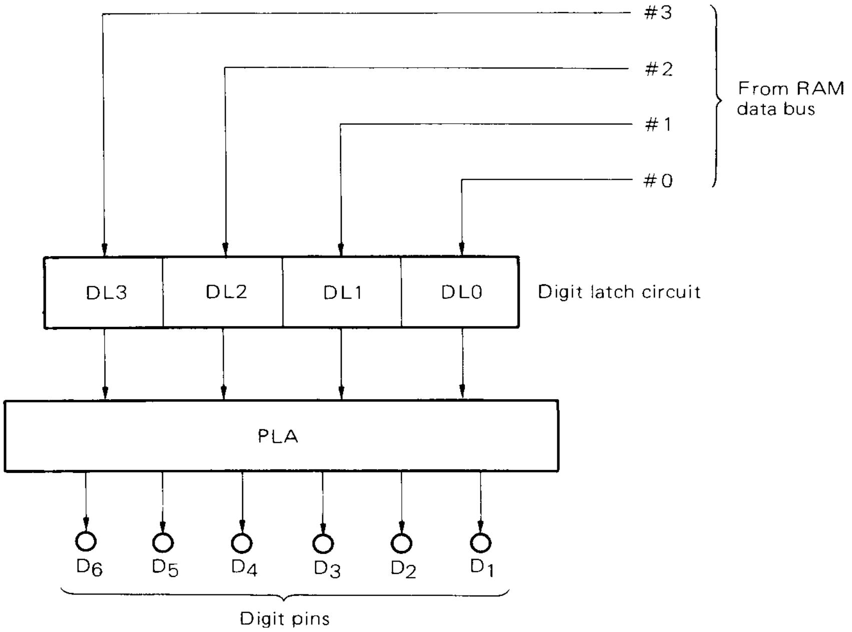


Fig. 18 Digit PLA Configuration

The contents of the register specified in the DIG instruction operand are latched in DL0 to DL3 of the digit circuit. For example, executing ε DIG 05H instruction for the following RAM latches "0AH" at register address 05H:

Column address

Row address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0						A										
1																
2								BANK0								
3																

When the digit PLA patterns listed in Table 9 have been defined, executing a DIG 05H instruction sets only the D_2 pin to ON (high level output).

Table 9 Digit PLA Pattern Examples

Digit latch				Digit output						Output pattern
DL3	DL2	DL1	DL0	D_6	D_5	D_4	D_3	D_2	D_1	
0	0	0	0	0	0	0	0	0	0	All OFF
0	0	0	1	0	0	0	0	0	1	D_1 ON
0	0	1	0	0	0	0	0	1	0	D_2 ON
0	0	1	1	0	0	0	1	0	0	D_3 ON
0	1	0	0	0	0	1	0	0	0	D_4 ON
0	1	0	1	0	1	0	0	0	0	D_5 ON
0	1	1	0	1	0	0	0	0	0	D_6 ON
0	1	1	1	0	0	0	0	0	0	No Use
1	0	0	0	0	0	0	0	0	0	All OFF
1	0	0	1	0	0	0	0	0	1	D_1 ON
1	0	1	0	0	0	0	0	1	0	D_2 ON
1	0	1	1	0	0	0	1	0	0	D_3 ON
1	1	0	0	0	0	1	0	0	0	D_4 ON
1	1	0	1	0	1	0	0	0	0	D_5 ON
1	1	1	0	1	0	0	0	0	0	D_6 ON
1	1	1	1	0	0	0	0	0	0	No Use

Assembler MTDIG pseudo-instruction specifies the output state of the digit pins when the CE pin is low level. An example is given below.

MTDIG 0 1 0 1 0 1 B (B means binary.)

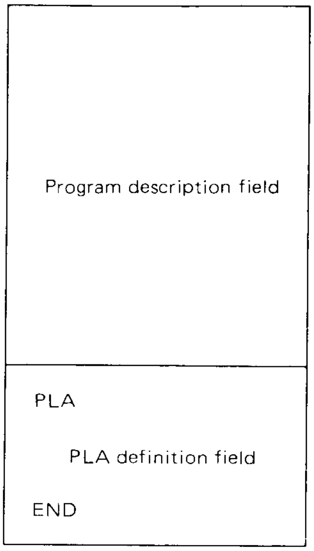
Bit associated with the D_1 pin

Bit associated with the D_6 pin

When the CE pin goes low, the D_6, D_4, and D_2 pins go low and the D_5, D_3 and D_1 pins go high automatically.

7.4 PLA PROGRAM EXAMPLE

The PLA must be defined for all the μPD1700 Series devices. **Tape with undefined PLA cannot be accepted.** PLA definition is at the end of the assembler source program and it contains the following items. None of these items can be omitted.



1. **PLA pseudo-instruction**

Specifies the end of the program description field and, at the same time, the beginning of the PLA definition area.

2. **SPLSEL (Segment PLA Select) pseudo-instruction**

Selects RAM address division to generate segment pattern groups 0 and 1. This must be specified in the following formats:

SPLSEL 3 or SPLSEL 2

3. **DSP (Define Segment PLA) pseudo-instruction**

Defines 32 segment PLA patterns. They must be defined sequentially from the 16 patterns in pattern group 0. An example is given below.

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

The bits are associated with g, f, e, d, c, b, and a sequentially from the beginning.

4. **DDP (Define Digit PLA) pseudo-instruction**

Defines 16 digit PLA patterns. An example is given below.

```
DDP    1 0 0 0 0 0 B
       ↑         ↑
       D6        D1
```

The bits are associated with the D_6, D_5, . . . , D_1 pins sequentially from the beginning.

5. MTDIG

Defines whether to activate the digit pins (D_1 to D_6) when the CE pin goes low. An example is given below.

$$\text{MTDIG}\qquad \underset{D_6}{\underset{\uparrow}{1}}\;0\;0\;0\;0\;\underset{D_1}{\underset{\uparrow}{0}}\;\text{B}$$

The bits are associated with the D_6, D_5, . . . , D_1 pins sequentially from the beginning. In the above example, the D6 pin goes high and other digit pins go low when the CE pin goes low.

6. END

Specifies the end of the source program together with the PLA definition end. If this is omitted, assembly does not occur.

Note: The PLA definition must begin with "PLA" and end with "END"; however, "SPLSEL", "DSP", "DDP", and "MTDIG" can be defined in arbitrary order between "PLA" and "END".

PLA programming example

```
PLA
; ********************* PLA DEFINISION *********************
;
;
SPLSEL 3                  ; RAM ADDR BIT #3
;
;********  SEGMENT PATTERN 0  ********
;
;       GFEDCBA
DSP     0111111B          ; "0"
DSP     0000110B          ; "1"
DSP     1011011B          ; "2"
DSP     1001111B          ; "3"
DSP     1100110B          ; "4"
DSP     1101101B          ; "5"
DSP     1111101B          ; "6"
DSP     0100111B          ; "7"
DSP     1111111B          ; "8"
DSP     1101111B          ; "9"
DSP     0000000B          ; BLANK (DISPLAY OFF)
DSP     1111001B          ; "E"
DSP     1110110B          ; "H"
DSP     0111001B          ; "C"
DSP     1110011B          ; "P"
DSP     1110111B          ; "A"
;
;********  SEGMENT PATTERN 1  ********
;
DSP     0000000B          ; KEY ALL OFF
DSP     0000001B          ; SEG A (KEY) ON
DSP     0000010B          ; SEG B (KEY) ON
DSP     0000100B          ; SEG C (KEY) ON
DSP     0001000B          ; SEG D (KEY) ON
DSP     0010000B          ; SEG E (KEY) ON
DSP     0100000B          ; SEG F (KEY) ON
DSP     1000000B          ; SEG G (KEY) ON
DSP     1111111B          ; KEY ALL ON
DSP     1000010B          ; "FM" & "MHz"
DSP     0100001B          ; "MW" & "kHz"
DSP     0010001B          ; "LW" & "kHz"
DSP     1001010B          ; "FM", "ME" & "MHz"
DSP     0101001B          ; "MW", "ME" & "kHz"
DSP     0011001B          ; "LW", "ME" & "kHz"
DSP     0000000B          ; No use
;
;
```

```
;
;********  DIGIT PATTERN  *********
;
;DIGIT 654321
DDP     000000B        ; ALL OFF
DDP     000001B        ; D1 ON
DDP     000010B        ; D2 ON
DDP     000100B        ; D3 ON
DDP     001000B        ; D4 ON
DDP     010000B        ; D5 ON
DDP     100000B        ; D6 ON
DDP     000000B        ; No use
DDP     000000B        ; ALL OFF
DDP     000001B        ; D1 ON
DDP     000010B        ; D2 ON
DDP     000100B        ; D3 ON
DDP     001000B        ; D4 ON
DDP     010000B        ; D5 ON
DDP     100000B        ; D6 ON
DDP     000000B        ; No use
;
;
MTDIG   000000B        ; ALL DIGIT OFF WHEN CE=LOW
;
END
```

μPD1707 INSTRUCTION SETS

Table 10 μPD1707 Instruction Sets

				$b_{15}b_{14}$	0 0	0 1	1 0	1 1
b_{13}	b_{12}	b_{11}	b_{10}		0	1	2	3
0	0	0	0	0	NOP SIO N	KIN M KI M	DIG r	ST M, r
0	0	0	1	1	SPB P, N SS N BANK1 EI STC	ORI M, I	SEG D_H, r	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 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	SIB M, I	———	SB r, M
1	0	0	0	8	SBK0 TPF P, N TSF N_2 TCEF TITF	AIN M, I	TSET TSEF TADT TADF	ADN r, M
1	0	0	1	9	SBK1 TPT P, N TST N_2 TCET TITT	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

μPD1707 INSTRUCTIONS

Explanatory note

D_H	:	Data memory address, high (row address) (2 bits)
D_L	:	Data memory address, low (column address) (2 bits)
R_n	:	Register number (4 bits)
I	:	Immediate data (4 bits)
N	:	Bit position (4 bits)
ADDR	:	Program memory address (10 bits)
———	:	All Is
r	:	General register one of addresses 00H to 0FH of bank 0
h	:	Halt release conditions $0 \leq h \leq 7$
M	:	Data memory address one of 64 words at data memory addresses 00H to 3FH of banks 0 and 1
P	:	Port $0 \leq P \leq 3$
N_1	:	Bit position of status word 1 $0 \leq N_1 \leq 7$
N_2	:	Bit position of status word 2 $0 \leq N_2 \leq 7$
( )	:	Register or memory contents
c	:	Carry
b	:	Borrow
$( )_n$	:	Contents of register or memory bit N
	:	Value decoded from register or memory contents

Instruction group	Mnemonic	Operand		Function	Operation	Machine code			
		1ST	2ND			Operation code			
Addition instructions	AD	r	M	Add memory to register	$r \leftarrow (r) + (M)$	1 1 0 1 0 0	D_H	D_L	R_n
	ADS	r	M	Add memory to register, then skip if carry	$r \leftarrow (r) + (M)$ skip if carry	1 1 1 1 0 0	D_H	D_L	R_n
	ADN	r	M	Add memory to register, then skip if not carry	$r \leftarrow (r) + (M)$ skip if not carry	1 1 1 0 0 0	D_H	D_L	R_n
	AC	r	M	Add memory to register with carry	$r \leftarrow (r) + (M) + c$	1 1 0 1 1 0	D_H	D_L	R_n
	ACS	r	M	Add memory to register with carry, then skip if carry	$r \leftarrow (r) + (M) + c$ skip if carry	1 1 1 1 1 0	D_H	D_L	R_n
	ACN	r	M	Add memory to register with carry, then skip if not carry	$r \leftarrow (r) + (M) + c$ skip if not carry	1 1 1 0 1 0	D_H	D_L	R_n
	AI	M	I	Add immediate data to memory	$M \leftarrow (M) + I$	0 1 0 1 0 0	D_H	D_L	I
	AIS	M	I	Add immediate data to memory, then skip if carry	$M \leftarrow (M) + I$ 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	$M \leftarrow (M) + I$ 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	$M \leftarrow (M) + I + c$	0 1 0 1 1 0	D_H	D_L	I

Instruction group	Mnemonic	Operand		Function	Operation	Machine code			
		1ST	2ND			Operation code			
Addition instructions	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 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 instructions	SU	r	M	Subtract memory from register	r←(r)−(M)	1 1 0 1 0 1	D_H	D_L	R_n
	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	R_n
	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	R_n
	SB	r	M	Subtract memory from register with borrow	r←(r)−(M)−b	1 1 0 1 1 1	D_H	D_L	R_n
	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	R_n
	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	R_n
	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 me mory, with borrow	M←(M)−I−b	0 1 0 1 1 1	D_H	D_L	I
	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 data with borrow, then skip if not borrow	M←(M)−I−b skip not borrow	0 1 1 0 1 1	D_H	D_L	I
Comparison instructions	SEQ	r	M	Skip if register equal memory	r−M skip if zero	1 0 1 1 1 0	D_H	D_L	R_n

Instruction group	Mnemonic	Operand		Function	Operation	Machine code			
		1ST	2ND			Operation code			
Comparison instructions	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	R_n
	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	R_n
	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	R_n
	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)≧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 instructions	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	R_n
Move instructions	LD	r	M	Load memory to register	r←(M)	1 0 0 1 1 0	D_H	D_L	R_n
	ST	M	r	Store register to memory	M←(r)	1 1 0 0 0 0	D_H	D_L	R_n
	MVRD	r	M	Move memory to destination memory referring to register in the same row	$[D_H, R_n]$←(M)	1 1 0 0 1 1	D_H	D_L	R_n
	MVRS	M	r	Move source memory referring to register to memory in the same row	M←$[D_H, R_n]$	1 1 0 0 0 1	D_H	D_L	R_n
	MVSR	M_1	M_2	Move memory to memory in the same row	$[D_H, DL_1]$←$[D_H, DL_2]$	1 0 0 1 0 0	D_H	D_{L1}	D_{L2}

Instruction group	Mnemonic	Operand		Function	Operation	Machine code			
		1ST	2ND			Operation code			
Move instructions	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~N), N_F & (M)	1 0 1 0 1 1	D_H	D_L	R_n
Bit test instructions	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 instruction	JMP	ADDR		Jump to the address specified in page 0	PC←ADDR, PAGE←0	0 0 0 1 1 0	ADDR (10bits)		
				Jump to the address specified in page 1	PC←ADDR, PAGE←1	0 0 0 0 1 0			
Subroutine instructions	CAL	ADDR		Call subroutine in page 0	Stack←[(PC)+1, Page], PC←ADDR PAGE←0	0 0 0 1 1 1	ADDR (10bits)		
	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	—	—	—
Interrupt control instructions	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 instructions	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	—	—	—

Timer test instruction	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

Instruction group	Mnemonic	Operand		Function	Operation	Machine code			
		1ST	2ND			Operation code			
Status word and pin test instructions	SS	N_1		Set status word 1	$(\text{STATUS WORD 1})_N \leftarrow 1$	0 0 0 0 0 1	–	0 $\overset{N_1}{\longleftrightarrow}$	–
	RS	N_1		Reset status word 1	$(\text{STATUS WORD 1})_N \leftarrow 0$	0 0 0 0 1 1	—	0 $\overset{N_1}{\longleftrightarrow}$	—
	TST	N_2		Test status word 2 true	if $(\text{STATUS WORD 2})_N$=all "1", then skip	0 0 1 0 0 1	–	0 $\overset{N_2}{\longleftrightarrow}$	—
	TSF	N_2		Test status word 2 false	if $(\text{STATUS WORD 2})_N$=all "0", then skip	0 0 1 0 0 0	·	0 $\overset{N_2}{\longleftrightarrow}$	—
	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/F← 0	0 0 0 0 1 1	—	1 1 0 0	—
	BANK1			Select BANK1	BANK F/F← 1	0 0 0 0 0 1	—	0 1 0 0	—
	TITT			Test INT, skip if true	if INT=0, then skip	0 0 1 0 0 1	—	0 0 0 1	—
	TITF			Test INT, skip if false	if $\overline{\text{INT}}$=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, if 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/F=0, then skip	0 0 1 0 0 0	—	1 1 0 0	—
	SBK1			Skip if BANK1	if BANK F/F=1, then skip	0 0 1 0 0 1	—	0 1 0 0	—

Instruction group	Mnemonic	Operand		Function	Operation	Machine code			
		1ST	2ND			Operation code			
Input/output instructions	SEG	D_H	r	Output segment pattern based on the memory specified indirectly	$SG_{0\sim4} \leftarrow [D_H, (R_n)]$, $S_{a\sim g} \leftarrow \{SG_{0\sim4}\}$	1 0 0 0 0 1	D_H	—	R_n
	DIG	r		Output digit pattern based on the register	$DL_{0\sim3} \leftarrow \{(Rn)_{0\sim3}\}$ $\{D_{1\sim6}\} \leftarrow DL_{0\sim3}$	1 0 0 0 0 0	—	—	r
	KIN	M		Input key data to memory, then skip if data are zero	$M \leftarrow K_0 \sim K_3$, skip if$(M)=0$	0 1 0 0 0 0	D_H	D_L	—
	KI	M		Input key data to memory	$M \leftarrow K_0 \sim K_3$	0 1 0 0 0 0	D_H	D_L	0 0 0 0
	IN	r	P	Input data on port to register	$r \leftarrow (Port(P))$	1 1 0 0 1 0	P	—	R_n
	OUT	P	r	Output contents of register to port	$(Port(P)) \leftarrow (r)$	1 0 0 0 1 0	P	—	R_n
	SPB	P	N	Set port bits	$(Port(P))_N \leftarrow 1$	0 0 0 0 0 1	P	0 0 0 0	N
	RPB	P	N	Reset port bits	$(Port(P))_N \leftarrow 0$	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 instructions	SIO	N		Serial input/output	$SMR(3, 1, 0) \leftarrow N(3, 1, 0)$	0 0 0 0 0 0	0 0	0 0 0 1	N
	TSET			Test shift end, then skip if true	if $SCC=8\times(2n+1)$, then skip $n\geqq0$	1 0 1 0 0 0	1 0	0 0 0 1	—
	TSEF			Test shift end, then skip if false	if $SCC\neq8\times(2n+1)$, then skip $n\geqq0$	1 0 1 0 0 0	0 0	0 0 0 1	—

Instruction group	Mnemonic	Operand		Function	Operation	Machine code			
		1ST	2ND			Operation code			
A-D converter test instructions	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	
Other instructions	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			

8. ELECTRICAL CHARACTERISTICS

8.1 ABSOLUTE MAXIMUM RATINGS

Supply voltage	V_{DD}	−3.0 to +6.0	V
Input voltage	V_I	−0.3 to +V_{DD}	V
Output voltage	V_O	−0.3 to +V_{DD}	V
Output sink current	I_{O1}	80 (total of PC_0 to PC_3)	mA
Output sink current	I_{O2}	35 (total of sink current other than PC_0 to PC_3)	mA
Output source current	$\|-I_{O3}\|$	40 (total of source current containing segments)	mA
Output source current	I_{O4}	7 (source current per segment)	mA
Output voltage	V_{BDS}	−40 (voltage of Sa to Sg source drains)	V
Output voltage	V_{BDS}	+10.5 (PORT B, PORT D, CGP)	V
Storage temperature	T_{stg}	−55 to +125	°C
Operating temperature	T_{opt}	−40 to +85 (V_{DD} = +4.5 to +5.5 V, $GND_1 = GND_2 = GND_3$, $f_{x'tal}$ = 4.5 MHz)	°C

8.2 RECOMMENDED OPERATING CONDITIONS

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT	CONDITION
PLL operation supply voltage	V_{DD}	4.5	5	5.5	V	V_{DD1}
CPU operation supply voltage	V_{DDC}	3.8	5	5.5	V	PLL stop, V_{DD1} (A/D)
Data holding voltage	V_{DDR}	2.5		5.5	V	Crystal oscillator stop
Power supply voltage—rise time	t_{rise}			500	ms	V_{DD} = 0 to 4.5 V

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

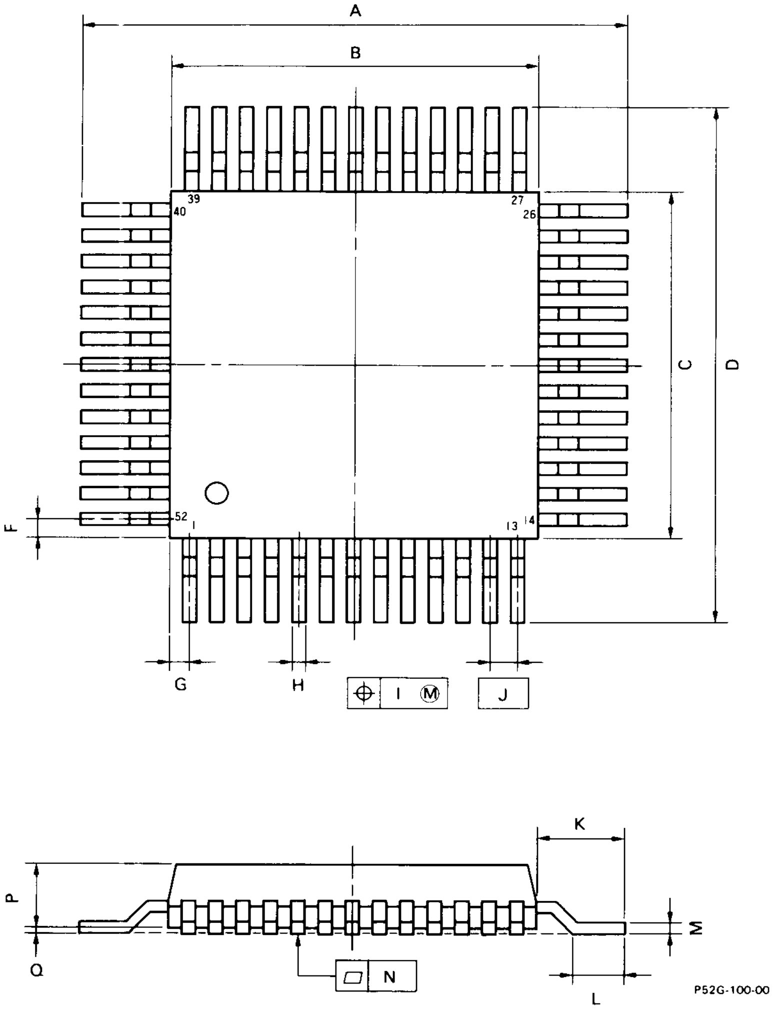
CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT	CONDITION
Input voltage high	V_{IH1}	0.7 V_{DD}			V	PORT A
Input voltage high	V_{IH2}	0.8 V_{DD}			V	CE, $\overline{INT}$
Input voltage high	V_{IH3}	0.6 V_{DD}			V	K_3 to K_0
Input voltage low	V_{IL1}			0.3 V_{DD}	V	PORT A
Input voltage low	V_{IL2}			0.2 V_{DD}	V	K_3 to K_0, CE, $\overline{INT}$
Output current high	$-I_{OH1}$	0.4			mA	PORT A $V_{OH}=V_{DD}-0.4$ V
Output current high	$-I_{OH2}$	0.5			mA	D_6 to D_1 $V_{OH}=V_{DD}-1$ V
Output current high	$-I_{OH3}$	0.5			mA	PSC, EO $V_{OH}=V_{DD}-1.5$ V
Output current high	$-I_{OH4}$	1.0			mA	S_a to S_g $V_{OH}=V_{DD}-1.5$ V
Output current low	I_{OL1}	0.6			mA	PORT A, B, D, CGP $V_{OL}=0.4$ V
Output current low	I_{OL2}	0.5			mA	D_6 to D_1, EO, PSC $V_{OL}=0.5$ V
Output current low	I_{OL3}	12		40	mA	PORT C $V_{OL}=1$ V
Input current high	I_{IH1}	5		50	µA	K_3 to K_0 $V_I=V_{DD}=4.5$ V
Input current high	I_{IH2}	100			µA	VCOH, VCOL, XI $V_I=V_{DD}=4.5$ V
Output leakage current high	I_{OH4}			5	µA	PORT B, D, CGP $V_O=8.5$ V
Output leakage current high	I_{OH5}			5	µA	PORT C $V_O=V_{DD}$
Output leakage current low	$-I_{OL4}$			5	µA	S_a to S_g $\lvert V_{SD}\rvert=35$ V
Output off-leakage 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	
Absolute A/D conversion accuracy		±1/2LSB		±1LSB		T_{opt} = −10 to +50 °C $V_{DD2}=V_{DD1}$
A/D conversion reference voltage	V_{DD2}	V_{DD}		V_{DD}	V	GND_3(A/D)=GND_1(CPU)
PLL operating supply current	I_{DDP}		10		mA	CPU and PLL operation (f_{in} = 15 MHz)
CPU operating supply current	I_{DDC}		0.5		mA	PLL stop and CPU operation
Data retention current	I_{DDR}		3	10	µA	Crystal oscillator stop, T_a = 25 °C

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

CHARACTERISTIC	SYMBOL	MIN.	TYP.	MAX.	UNIT	CONDITION
PLL input frequency	f_{in}			15	MHz	VCOL, VCOH (V_{in}=0.8 V_{p-p} sine wave)

9. PACKAGE DIMENSIONS (Unit: mm)

52 PIN PLASTIC FLAT PACKAGE (LEAD BENDED TYPE)

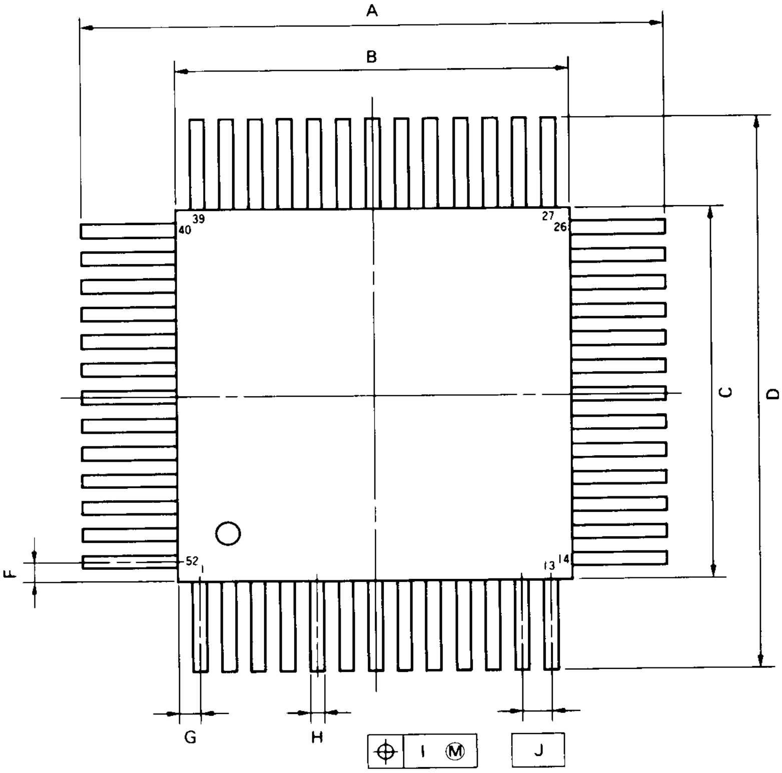


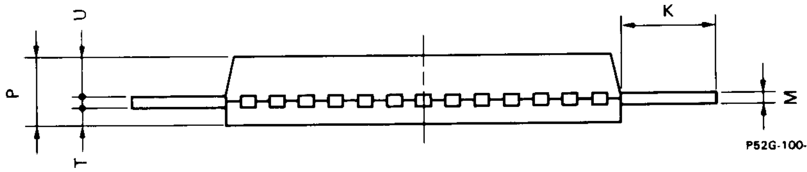
NOTE

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	$21.0^{\pm0.4}$	$0.827^{\pm0.016}$
B	$14^{\pm0.2}$	$0.551^{+0.009}_{-0.008}$
C	$14^{\pm0.2}$	$0.551^{+0.009}_{-0.008}$
D	$21.0^{\pm0.4}$	$0.827^{\pm0.016}$
F	1.0	0.039
G	1.0	0.039
H	$0.40^{\pm0.10}$	$0.016^{+0.004}_{-0.005}$
I	0.20	0.008
J	1.0 (T.P.)	0.039 (T.P.)
K	$3.5^{\pm0.2}$	$0.138^{+0.008}_{-0.009}$
L	$2.2^{\pm0.2}$	$0.087^{+0.008}_{-0.009}$
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	$0.1^{\pm0.1}$	$0.004^{\pm0.004}$

52 PIN PLASTIC FLAT PACKAGE (STRIGHT 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	$19.8^{\pm 0.4}$	$0.780^{+0.016}_{-0.017}$
B	$14^{+0.2}$	$0.551^{+0.009}_{-0.008}$
C	$14^{+0.2}$	$0.551^{+0.009}_{-0.008}$
D	$19.8^{\pm 0.4}$	$0.780^{+0.016}_{-0.017}$
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	$2.9^{+0.2}$	$0.114^{+0.009}_{-0.008}$
M	$0.15^{+0.10}_{-0.05}$	$0.006^{+0.004}_{-0.003}$
P	$2.6^{+0.2}_{-0.1}$	$0.102^{+0.009}_{-0.004}$
T	1.0	0.039
U	1.45	0.057

10. SUPPORT TOOLS

The following support tools are available for developing systems using the μPD1707.

Hardware	EVAKIT-' 700		This is an evaluation board usable in common in the μPD1700 series. It is consisted of CPU board, PLA & TRACER board and SUPERVISOR board (SB-8201). It must be used with option I/O board (EV-1707). It is possible that immediately adding and revising the program through the console.		
	EV-1707		EV-1707 is an option I/O board for EVAKIT-1700 and SE-1700. It is for developping the program of μPD1707, μPD1711 and μPD1712.		
	SE-1700		SE-1700 is a system evaluation board using evaluation chip of μPD1700 series, and in which the program developed by the EVAKIT-1700 is loaded and provided in a system instead of the μPD1707 to evaluate the system. Also, SE-1700 must be used with EV-1707.		
Software	μPD1700 series Assembler	Absolute Assembler	Host		Order name
				OS	
			PDA-880	CP/M^{TM}	μS281AS1700
			PDA-800 + PDA-800FDD		
			MD-080 series		
			MD-086 series	MP/M-86^{TM}	μS171AS1700

Remarks 1. Supply format of software is 8 ° FD double-side double density.

2. Regarding CP/M version of OS of PDA-800 and PDA-880

3. CP/M^{TM} and MP/M-86 ^{TM} are the trademark of Digital Research Co., Ltd.

IC—1741
May 1986P
Printed in Japan

72

013686