4282 Group

SINGLE-CHIP 4-BIT CMOS MICROCOMPUTER

DESCRIPTION

The 4282 Group enables fabrication of 8 × 7 key matrix and has the followin timers;

- an 8-bit timer which can be used to set each carrier wave and has two reload register
- an 8-bit timer which can be used to auto-control and has a reload register.

FEATURES

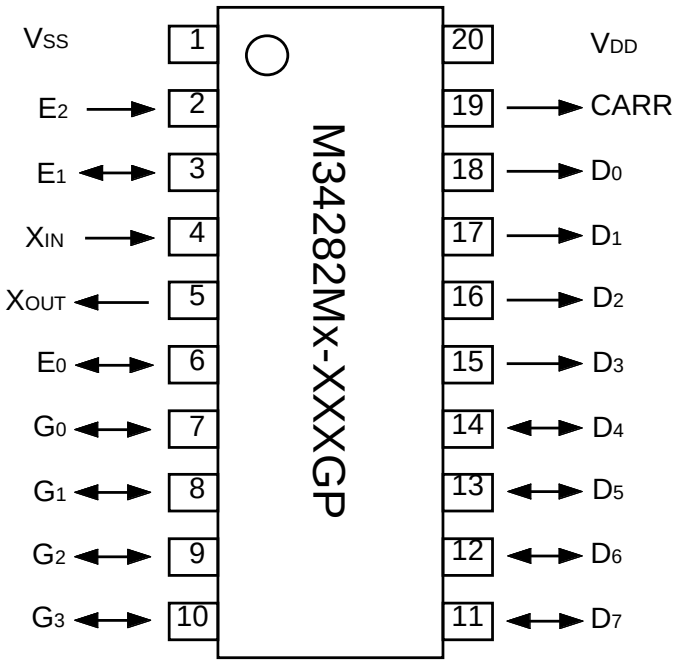
- Number of basic instructions 68
- Minimum instruction execution time ..... 8.0 μs (at f(X_{IN}) = 4.0 MHz, system clock = f(X_{IN})/8)
- Supply voltage ..... 1.8 V to 3.6 V
- Subroutine nesting ..... 4 levels
- Timer
  Timer 1 ..... 8-bit timer
  (This has a reload register and carrier wave output auto-control function)
  Timer 2 ..... 8-bit timer
  (This has two reload registers and carrier wave output function)
- Logic operation function (XOR, OR, AND)
- RAM back-up function
- Key-on wakeup function (ports D_4–D_7, E_0–E_2, G_0–G_3) 11
- I/O port (ports D, E, G, CARR) 16
- Oscillation circuit ..... Ceramic resonance
- Watchdog timer
- Power-on reset circuit
- Voltage drop detection circuit ..... Typical:1.50 V (system reset)

APPLICATION

Various remote control transmitters

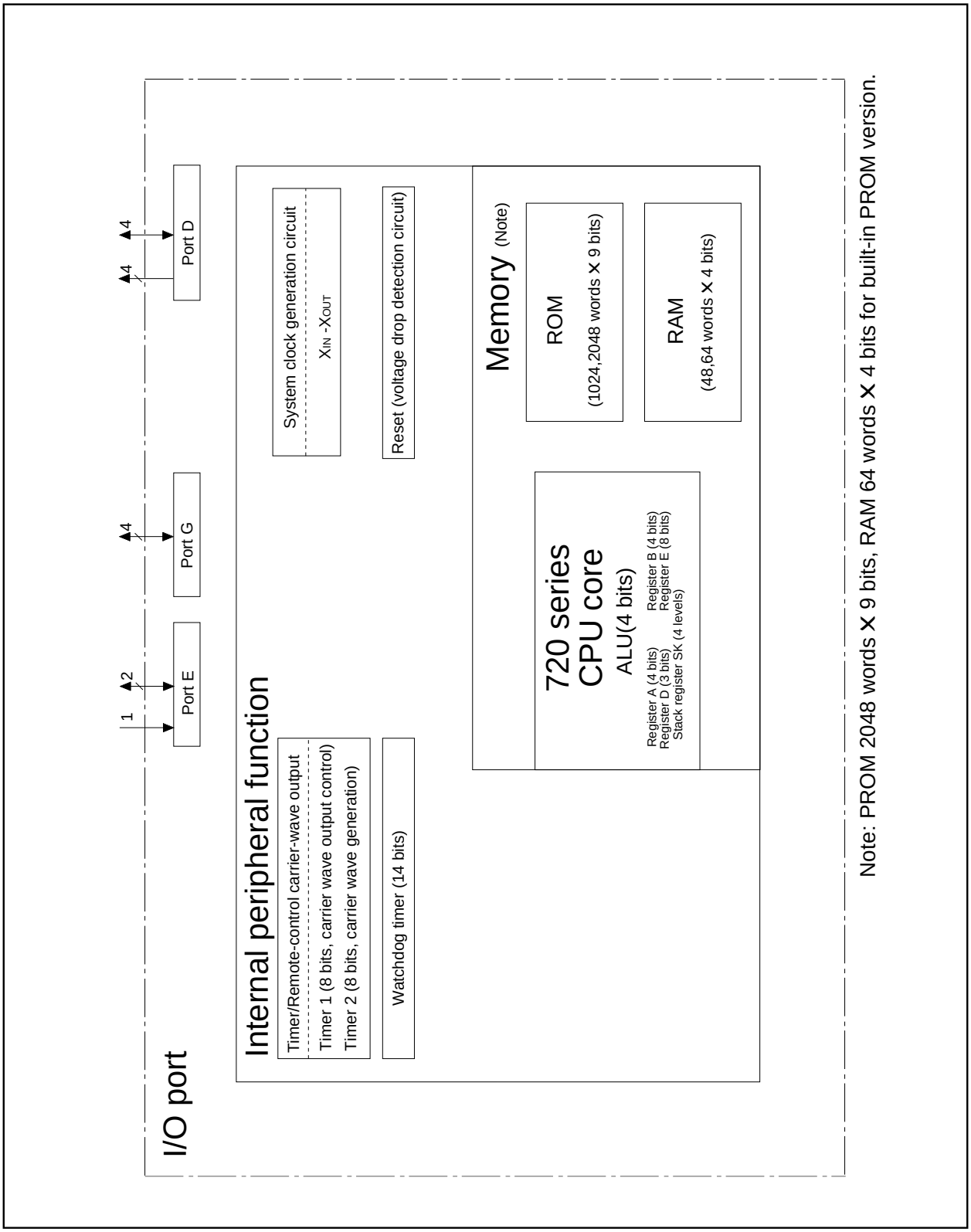
Product	ROM (PROM) size (× 9 bits)	RAM size (× 4 bits)	Package	ROM type
M34282M1-XXXGP	1024 words	48 words	20P2E/F-A	Mask ROM
M34282M2-XXXGP	2048 words	64 words	20P2E/F-A	Mask ROM
M34282E2GP	2048 words	64 words	20P2E/F-A	One Time PROM

PIN CONFIGURATION (TOP VIEW)



Outline 20P2E/F-A

BLOCK DIAGRAM



PERFORMANCE OVERVIEW

Parameter			Function
Number of basic instructions			68
Minimum instruction execution time			8.0 μs (f(X_{IN}) = 4.0 MHz, system clock = f(X_{IN})/8, V_{DD} = 3 V)
Memory sizes	ROM	M34282M2/E2	2048 words × 9 bits
		M34282M1	1024 words × 9 bits
	RAM	M34282M2/E2	64 words × 4 bits
		M34282M1	48 words × 4 bits
Input/Output ports	D_0–D_3	Output	Four independent output ports
	D_4–D_7	I/O	Four independent I/O ports with the pull-down function
	E_0–E_2	Input	3-bit input port with the pull-down function
	E_0, E_1	Output	2-bit output port (E_0, E_1)
	G_0–G_3	I/O	4-bit I/O port with the pull-down function
	CARR	Output	1-bit output port; CMOS output
Timer	Timer 1		8-bit timer with a reload register
	Timer 2		8-bit timer with two reload registers
Subroutine nesting			4 levels (However, only 3 levels can be used when the TABP p instruction is executed)
Device structure			CMOS silicon gate
Package			20-pin plastic molded SSOP (20P2E/F-A)
Operating temperature range			–20 °C to 85 °C
Supply voltage			1.8 V to 3.6 V
Power dissipation (typical value)	Active mode		400 µA (f(X_{IN}) = 4.0 MHz, system clock = f(X_{IN})/8, V_{DD} = 3 V)
	RAM back-up mode		0.1 μA (at room temperature, V_{DD} = 3 V)

PIN DESCRIPTION

Pin	Name	Input/Output	Function
V_{DD}	Power supply	—	Connected to a plus power supply.
V_{SS}	Ground	—	Connected to a 0 V power supply.
X_{IN}	System clock input	Input	I/O pins of the system clock generating circuit. Connect a ceramic resonator between pins X_{IN} and X_{OUT}. The feedback resistor is built-in between pins X_{IN} and X_{OUT}.
X_{OUT}	System clock output	Output	
D_0–D_3	Output port D	Output	Each pin of port D has an independent 1-bit wide output function. The output structure is P-channel open-drain.
D_4–D_7	I/O port D	I/O	1-bit I/O port. For input use, set the latch of the specified bit to “0.” When the built-in pull-down transistor is turned on, the key-on wakeup function using “H” level sense and the pull-down transistor become valid. The output structure is P-channel open-drain.
E_0–E_2	I/O port E	Output	2-bit (E_0, E_1) output port. The output structure is P-channel open-drain.
		Input	3-bit input port. For input use (E_0, E_1), set the latch of the specified bit to “0.” When the built-in pull-down transistor is turned on, the key-on wakeup function using “H” level sense and the pull-down transistor become valid. Port E_2 has an input-only port and has a key-on wakeup function using “H” level sense and pull-down transistor.
G_0–G_3	I/O port G	I/O	4-bit I/O port. For input use, set the latch of the specified bit to “0.” The output structure is P-channel open-drain. When the built-in pull-down transistor is turned on, the key-on wakeup function using “H” level sense and pull-down transistor become valid.
CARR	Carrier wave output for remote control	Output	Carrier wave output pin for remote control. The output structure is CMOS circuit.

CONNECTIONS OF UNUSED PINS

Pin	Connection
D_0–D_7	Open or connect to V_{DD} pin (Note 1).
E_0, E_1	Set the output latch to "1" and open, or connect to V_{DD} pin (Note 2).
E_2	Open or connect to V_{SS} pin.
G_0–G_3	Set the output latch to "1" and open, or connect to V_{DD} pin (Note 2).

Notes 1: Ports D_4–D_7: Set the bit 2 ($PU0_2$) of the pull-down control register PU1 to "0" by software and turn the pull-down transistor OFF.

2: Set the corresponding bits of the pull-down control register PU0 to "0" by software and turn the pull-down transistor OFF.

(Note in order to set the output latch to "1" to make pins open)

- After system is released from reset, a port is in a high-impedance state until the output latch of the port is set to "1" by software. Accordingly, the voltage level of pins is undefined and the excess of the supply current may occur.
- To set the output latch periodically is recommended because the value of output latch may change by noise or a program run away (caused by noise).

(Note when connecting to V_{SS} and V_{DD})

- Connect the unused pins to V_{SS} or V_{DD} at the shortest distance and use the thick wire against noise.

PORT FUNCTION

Port	Pin	Input/ Output	Output structure	Control bits	Control instructions	Control registers	Remark
Port D	D_0–D_3	Output (4)	P-channel open-drain	1 bit	SD RD CLD		
	D_4–D_7	I/O (4)			SD RD CLD SZD	PU1	Pull-down function and key-on wakeup function (programmable)
Port E	E_0 E_1	I/O (2)	P-channel open-drain	Output: 2 bits Input: 3 bits	OEA IAE	PU0	Pull-down function and key-on wakeup function (programmable)
	E_2	Input (1)			IAE		
Port G	G_0–G_3	I/O (4)	P-channel open-drain	4 bits	OGA IAG	PU0	Pull-down function and key-on wakeup function (programmable)
Port CARR	CARR	Output (1)	CMOS	1 bit	SCAR RCAR		

DEFINITION OF CLOCK AND CYCLE

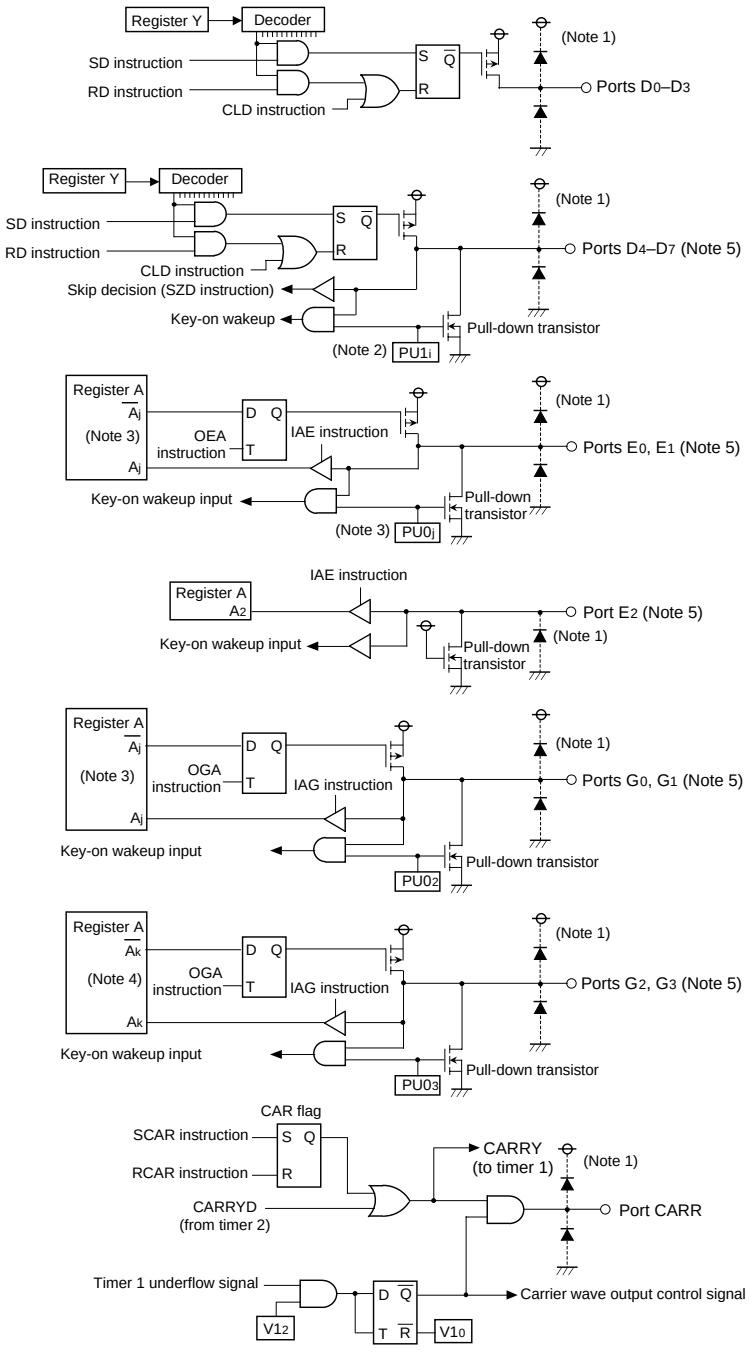
- System clock (STCK)
  The system clock is the source clock for controlling this product. It can be selected as shown below whether to use the CCK instruction.

CCK instruction	System clock	Instruction clock
When not using	$f(X_{IN})/8$	$f(X_{IN})/32$
When using	$f(X_{IN})$	$f(X_{IN})/4$

- Instruction clock (INSTCK)
  The instruction clock is a signal derived by dividing the system clock by 4, and is the basic clock for controlling CPU. The one instruction clock cycle is equivalent to one machine cycle.

- Machine cycle
  The machine cycle is the cycle required to execute the instruction.

PORT BLOCK DIAGRAMS



Notes 1: ⊣ This symbol represents a parasitic diode.
2: i represents bits 0 to 3.
3: j represents bits 0, 1.
4: k represents bits 2, 3.
5: Applied voltage must be less than VDD.

FUNCTION BLOCK OPERATIONS
CPU

(1) Arithmetic logic unit (ALU)

The arithmetic logic unit ALU performs 4-bit arithmetic such as 4-bit data addition, comparison, and bit manipulation.

(2) Register A and carry flag

Register A is a 4-bit register used for arithmetic, transfer, exchange, and I/O operation.

Carry flag CY is a 1-bit flag that is set to "1" when there is a carry with the AMC instruction (Figure 1).

It is unchanged with both A n instruction and AM instruction.

The value of A_0 is stored in carry flag CY with the RAR instruction (Figure 2).

Carry flag CY can be set to "1" with the SC instruction and cleared to "0" with the RC instruction.

(3) Registers B and E

Register B is a 4-bit register used for temporary storage of 4-bit data, and for 8-bit data transfer together with register A.

Register E is an 8-bit register. It can be used for 8-bit data transfer with register B used as the high-order 4 bits and register A as the low-order 4 bits (Figure 3).

(4) Register D

Register D is a 3-bit register.

It is used to store a 7-bit ROM address together with register A and is used as a pointer within the specified page when the TABP p, BLA p, or BMLA p instruction is executed (Figure 4).

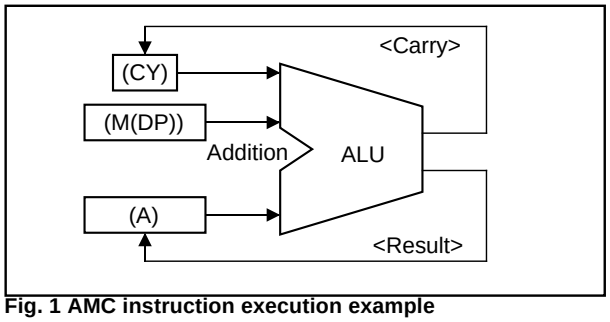


Fig. 1 AMC instruction execution example

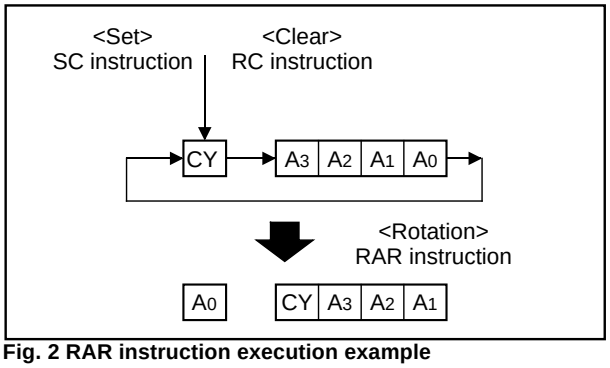


Fig. 2 RAR instruction execution example

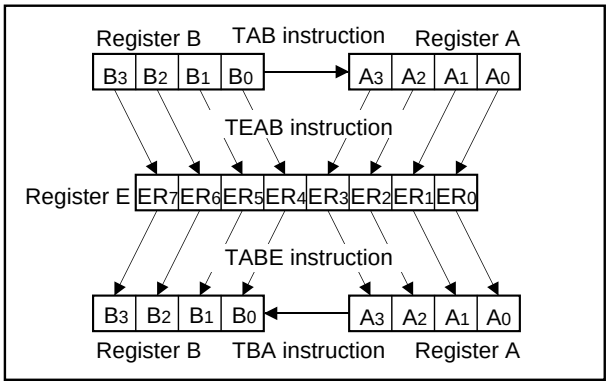


Fig. 3 Registers A, B and register E

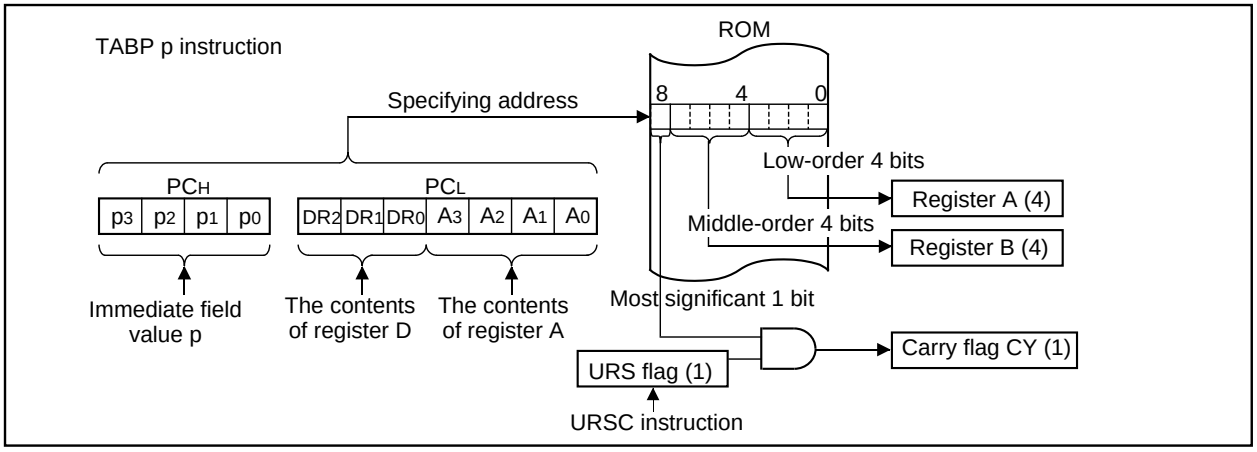


Fig. 4 TABP p instruction execution example

(5) Most significant ROM code reference enable flag (URS)

URS flag controls whether to refer to the contents of the most significant 1 bit (bit 8) of ROM code when executing the TABP p instruction. If URS flag is "0," the contents of the most significant 1 bit of ROM code is not referred even when executing the TABP p instruction. However, if URS flag is "1," the contents of the most significant 1 bit of ROM code is set to flag CY when executing the TABP p instruction (Figure 4).

URS flag is "0" after system is released from reset and returned from RAM back-up mode. It can be set to "1" with the URSC instruction, but cannot be cleared to "0."

(6) Stack registers (SKs) and stack pointer (SP)

Stack registers (SKs) are used to temporarily store the contents of program counter (PC) just before branching until returning to the original routine when;

- performing a subroutine call, or
- executing the table reference instruction (TABP p).

Stack registers (SKs) are four identical registers, so that subroutines can be nested up to 4 levels. However, one of stack registers is used when executing a table reference instruction. Accordingly, be careful not to over the stack. The contents of registers SKs are destroyed when 4 levels are exceeded.

The register SK nesting level is pointed automatically by 2-bit stack pointer (SP).

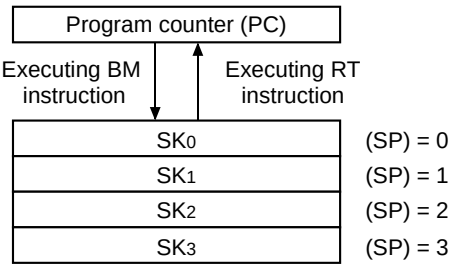
Figure 5 shows the stack registers (SKs) structure.

Figure 6 shows the example of operation at subroutine call.

(7) Skip flag

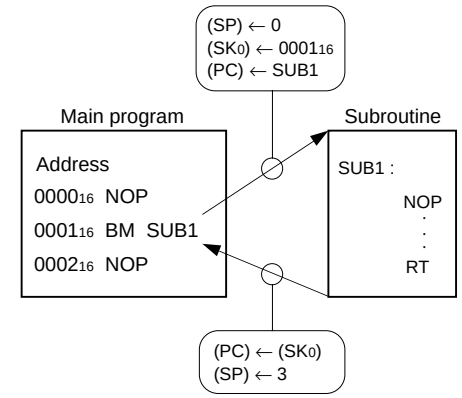
Skip flag controls skip decision for the conditional skip instructions and continuous described skip instructions.

Note : The 4282 Group just invalidates the next instruction when a skip is performed. The contents of program counter is not increased by 2. Accordingly, the number of cycles does not change even if skip is not performed. However, the cycle count becomes "1" if the TABP p, RT, or RTS instruction is skipped.



Stack pointer (SP) points "3" at reset or returning from RAM back-up mode. It points "0" by executing the first BM instruction, and the contents of program counter is stored in SK_0. When the BM instruction is executed after four stack registers are used ((SP) = 3), (SP) = 0 and the contents of SK_0 is destroyed.

Fig. 5 Stack registers (SKs) structure



Note: Returning to the BM instruction execution address with the RT instruction, and the BM instruction is equivalent to the NOP instruction.

Fig. 6 Example of operation at subroutine call

(8) Program counter (PC)

Program counter (PC) is used to specify a ROM address (page and address). It determines a sequence in which instructions stored in ROM are read. It is a binary counter that increments the number of instruction bytes each time an instruction is executed. However, the value changes to a specified address when branch instructions, subroutine call instructions, return instructions, or the table reference instruction (TABP p) is executed.

Program counter consists of PC_H (most significant bit to bit 7) which specifies to a ROM page and PC_L (bits 6 to 0) which specifies an address within a page. After it reaches the last address (address 127) of a page, it specifies address 0 of the next page (Figure 7).

Make sure that the PC_H does not exceed after the last page of the built-in ROM.

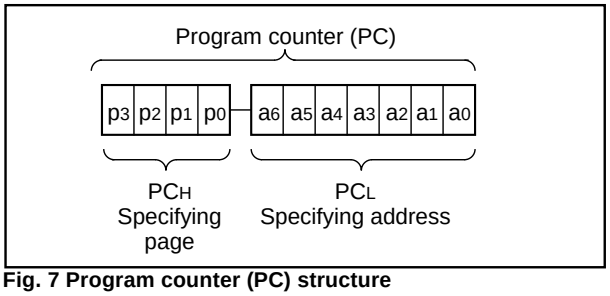


Fig. 7 Program counter (PC) structure

(9) Data pointer (DP)

Data pointer (DP) is used to specify a RAM address and consists of registers X and Y. Register X specifies a file and register Y specifies a RAM digit (Figure 8).

Register Y is also used to specify the port D bit position.

When using port D, set the port D bit position to register Y certainly and execute the SD, RD, or SZD instruction (Figure 9).

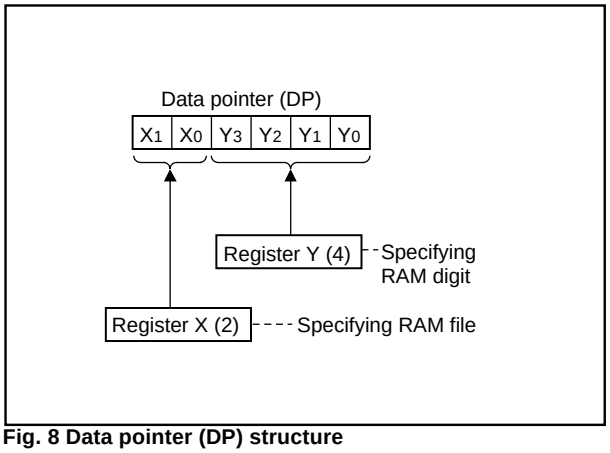


Fig. 8 Data pointer (DP) structure

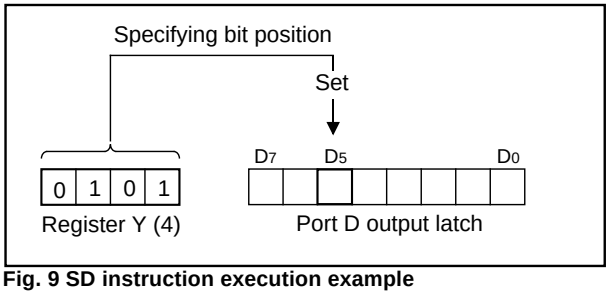


Fig. 9 SD instruction execution example

PROGRAM MEMORY (ROM)

The program memory is a mask ROM. 1 word of ROM is composed of 9 bits. ROM is separated every 128 words by the unit of page (addresses 0 to 127).

Table 1 ROM size and pages

Product	ROM size (× 9 bits)	Pages
M34282M2/E2	2048 words	16 (0 to 15)
M34282M1	1024 words	8 (0 to 7)

Page 2 (addresses 0100_{16} to $017F_{16}$) is the special page for subroutine calls. Subroutines written in this page can be called from any page with the 1-word instruction (BM). Subroutines extending from page 2 to another page can also be called with the BM instruction when it starts on page 2.
ROM pattern of all addresses can be used as data areas with the TABP p instruction.

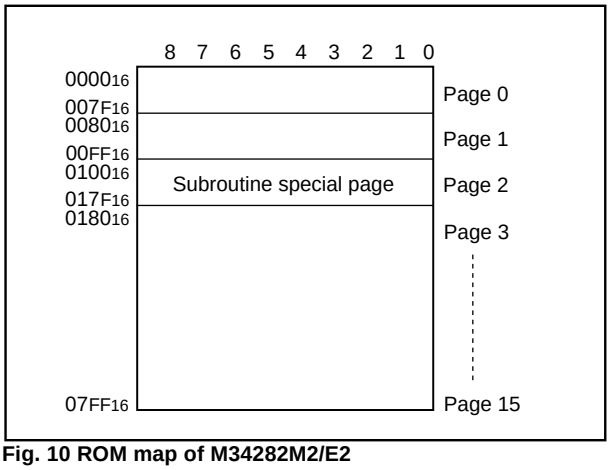


Fig. 10 ROM map of M34282M2/E2

DATA MEMORY (RAM)

1 word of RAM is composed of 4 bits, but 1-bit manipulation (with the SB j, RB j, and SZB j instructions) is enabled for the entire memory area. A RAM address is specified by a data pointer. The data pointer consists of registers X and Y. Set a value to the data pointer certainly when executing an instruction to access RAM.
Table 2 shows the RAM size. Figure 11 shows the RAM map.

Table 2 RAM size

Product	RAM size
M34282M2/E2	64 words × 4 bits (256 bits)
M34282M1	48 words × 4 bits (192 bits)

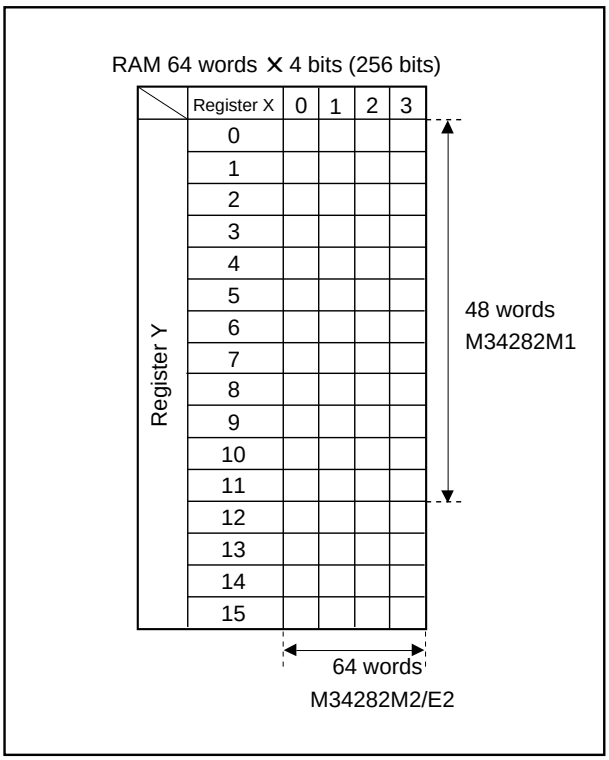


Fig. 11 RAM map

TIMERS

The 4282 Group has the programmable timer.

- Programmable timer
  The programmable timer has a reload register and enables the frequency dividing ratio to be set. It is decremented from a setting value n. When it underflows (count to n + 1), a timer 1 underflow flag is set to "1," new data is loaded from the reload register, and count continues (auto-reload function).

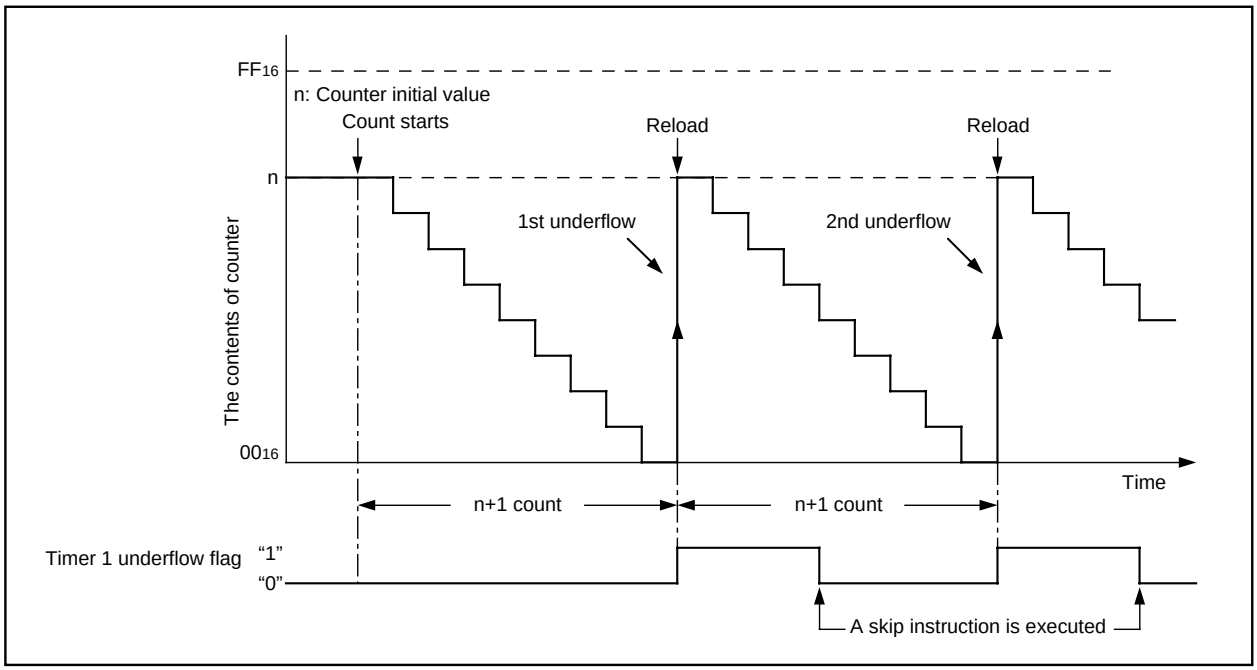


Fig. 12 Auto-reload function

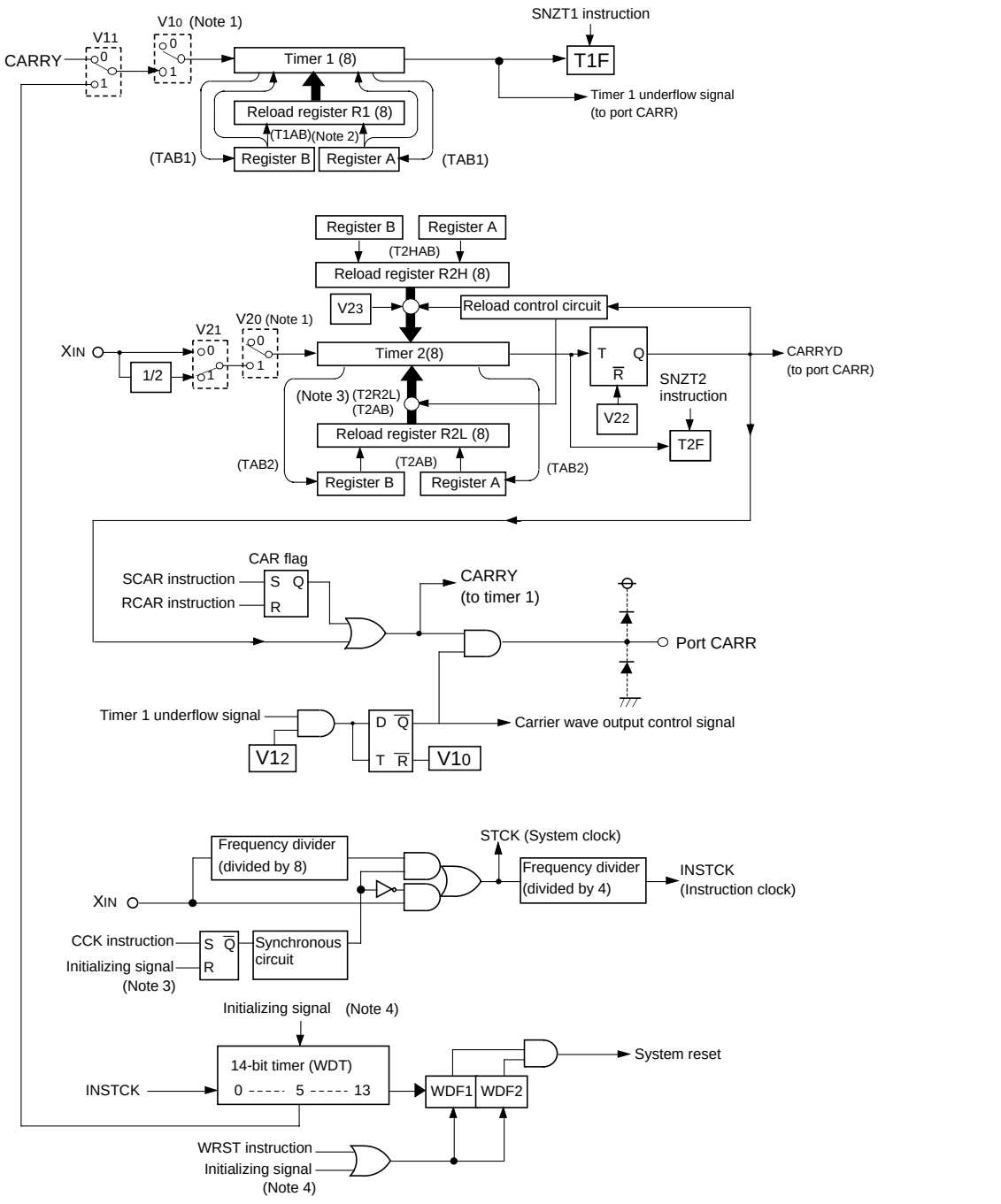
The 4282 Group timer consists of the following circuit.

- Timer 1 : 8-bit programmable timer
- Timer 2 : 8-bit programmable timer

These timers can be controlled with the timer control registers V1 and V2.
Each timer function is described below.

Table 3 Function related timer

Circuit	Structure	Count source	Frequency dividing ratio	Use of output signal	Control register
Timer 1	8-bit programmable binary down counter	• Carrier wave output (CARRY) • Bit 5 of watchdog timer	1 to 256	• Carrier wave output control	V1
Timer 2	8-bit programmable binary down counter	• f(XIN) • f(XIN)/2	1 to 256	• Carrier wave output	V2
14-bit timer	14-bit fixed frequency	• Instruction clock	16384	• Watchdog timer • Timer 1 count source	



Notes 1: Counting is stopped by clearing to "0."
2: When the T1AB instruction is executed after V10 is set to "1," writing is performed only to reload register R1.
3: The data of reload register R2L set with the T2AB instruction can be also written to timer 2 with the T2R2L instruction.
4: The initializing signal is output at reset or RAM back-up mode.

Fig. 13 Timers structure

Table 4 Control registers related to timer

Timer control register V1		at reset : 000_2	at RAM back-up : 000_2	W
$V1_2$	Carrier wave output auto-control bit	0	Auto-control output by timer 1 is invalid	
		1	Auto-control output by timer 1 is valid	
$V1_1$	Timer 1 count source selection bit	0	Carrier wave output (CARRY)	
		1	Bit 5 of watchdog timer (WDT)	
$V1_0$	Timer 1 control bit	0	Stop (Timer 1 state retained)	
		1	Operating	

Timer control register V1		at reset : 0000_2	at RAM back-up : 0000_2	W
$V1_3$	Carrier wave "H" interval expansion bit	0	To expand "H" interval is invalid	
		1	To expand "H" interval is valid (when $V2_2$=1 selected)	
$V1_2$	Carrier wave generation function control bit	0	Carrier wave generation function invalid	
		1	Carrier wave generation function valid	
$V1_1$	Timer 2 count source selection bit	0	$f(X_{IN})$	
		1	$f(X_{IN})/2$	
$V1_0$	Timer 2 control bit	0	Stop (Timer 2 state retained)	
		1	Operating	

Note: "W" represents write enabled.

(1) Control registers related to timer

- Timer control register V1
  Register V1 controls the timer 1 count source and auto-control function of carrier wave output from port CARR by timer 1. Set the contents of this register through register A with the TV1A instruction.
- Timer control register V2
  Register V2 controls the timer 2 count source and the carrier wave generation function by timer. Set the contents of this register through register A with the TV2A instruction.

(2) Precautions

Note the following for the use of timers.

- Count source
  Stop timer 1 or timer 2 counting to change its count source.
- Watchdog timer
  Be sure that the timing to execute the WRST instruction in order to operate WDT efficiently.
- Writing to reload register R1
  When writing data to reload register R1 while timer 1 is operating, avoid a timing when timer 1 underflows.
- Timer 1 count operation
  When the bit 5 of the watchdog timer (WDT) is selected as the timer 1 count source, the error of maximum ± 256 µs (at the minimum instruction execution time : 8 µs) is generated from timer 1 start until timer 1 underflow. When programming, be careful about this error.
- Stop of timer 2
  Avoid a timing when timer 2 underflows to stop timer 2.
- Writing to reload register R2H
  When writing data to reload register R2H while timer 2 is operating, avoid a timing when timer underflows.
- Timer 2 carrier wave output function
  When to expand "H" interval of carrier wave is valid, set "1" or more to reload register R2H.

(3) Timer 1

Timer 1 is an 8-bit binary down counter with the timer 1 reload register (R1).

When timer is stopped, data can be set simultaneously in timer 1 and the reload register (R1) with the T1AB instruction.

When timer is operating, data can be set to only reload register R1 with the T1AB instruction.

When setting the next count data to reload register R1 at operating, set data before timer 1 underflows.

Timer 1 starts counting after the following process;

① set data in timer 1,

② select the count source with the bit 1 of register V1, and

③ set the bit 0 of register V1 to "1."

Once count is started, when timer 1 underflows (the next count pulse is input after the contents of timer 1 becomes "0"), the timer 1 underflow flag (T1F) is set to "1," new data is loaded from reload register R1, and count continues (auto-reload function).

When a value set in reload register R1 is n, timer 1 divides the count source signal by n + 1 (n = 0 to 255).

When the bit 2 of register V1 is set to "1," the carrier wave output enable/disable interval of port CARR is alternately generated each timer 1 underflows (Figure 14).

Data can be read from timer 1 to registers A and B. When reading the data, stop the counter and then execute the TAB1 instruction.

(4) Timer 2

Timer 2 is an 8-bit binary down counter with the timer 2 reload registers (R2H and R2L).

Data can be set simultaneously in timer 2 and the reload register (R2L) with the T2AB instruction.

The contents of reload register (R2L) set with the T2AB instruction can be set again to timer 2 with the T2R2L instruction. Data can be set to reload register (R2H) with the T2HAB instruction.

Timer 2 starts counting after the following process;

① set data in timer 2,

② select the count source with the bit 1 of register V2, and

③ select the valid/invalid of the carrier wave generation function by bit 2 of register V1 (when this function is valid, select the valid/invalid of the carrier wave "H" interval expansion by bit 3), and

④ set the bit 0 of register V1 to "1."

When the carrier wave generation function is invalid ($V2_2$="0"), the following operation is performed;

Once count is started, when timer 2 underflows (the next count pulse is input after the contents of timer 2 becomes "0"), the timer 2 underflow flag (T2F) is set to "1," new data is loaded from reload register R2L, and count continues (auto-reload function).

When a value set in reload register R2L is n, timer 2 divides the count source signal by n + 1 (n = 0 to 255).

When the carrier wave generation function is valid ($V2_2$="1"), the carrier wave which has the "L" interval set to the reload register R2L and "H" interval set to the reload register R2H can be output (Figure 15).

After the count of the "L" interval of carrier wave is started, timer 2 underflows and the timer 2 underflow flag (T2F) is set to "1". Then, the "H" interval data of carrier wave is reloaded from the reload register R2H, and count continues.

When timer underflows again after auto-reload, the T2F flag is set to "1". And then, the "L" interval data of carrier wave is reloaded from the reload register R2L, and count continues. After that, each timer underflows, data is reloaded from reload register R2H and R2L alternately.

When a value set in reload register R2H is n, "H" interval of carrier wave is as follows;

① When to expand "H" interval is invalid ($V2_3$ = "0"),
Count source × (n+1), n = 0 to 255

② When to expand "H" interval is valid ($V2_3$ = "1"),
Count source × (n+1.5), n = 1 to 255

When a value set in reload register R2L is m, "L" interval of carrier wave is as follows;

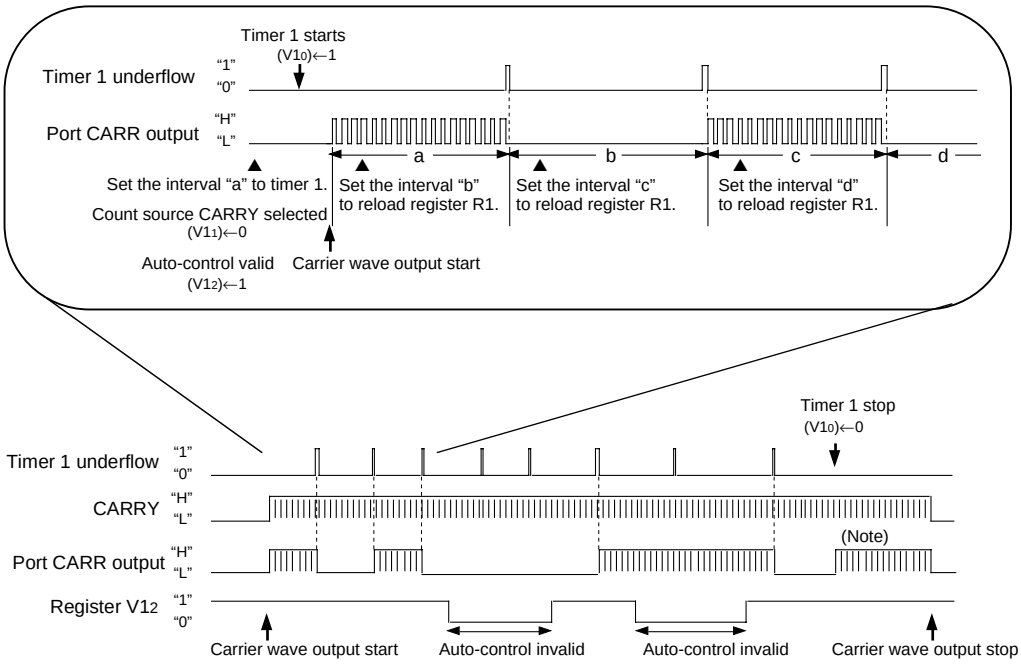
Count source × (m+1), m = 0 to 255

Data can be read from timer 2 to registers A and B. When reading the data, stop the counter and then execute the TAB2 instruction.

(5) Timer underflow flags (T1F, T2F)

Timer 1 underflow flag or timer 2 underflow flag is set to "1" when the timer 1 or timer 2 underflows. The state of flags T1F and T2F can be examined with the skip instruction (SNZT1, SNZT2).

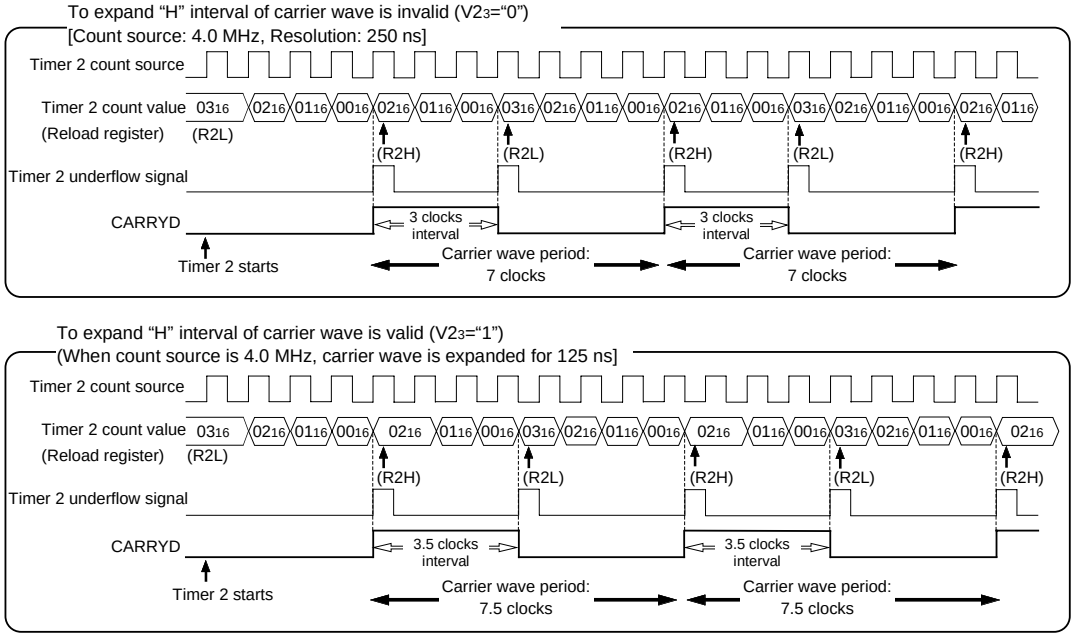
Flags T1F and T2F are cleared to "0" when the next instruction is skipped with a skip instruction.



Note: When timer 1 is stopped, the port CARR output auto-control is terminated regardless of bit 2 (V12) of register V1.

Fig. 14 Port CARR output control by timer 1

● In this case, the following is set;
- Timer 2 carrier wave generation function is valid (V22="1"),
- "L" interval (03_{16}) of carrier wave is set to reload register R2L
- "H" interval (02_{16}) of carrier wave is set to reload register R2H

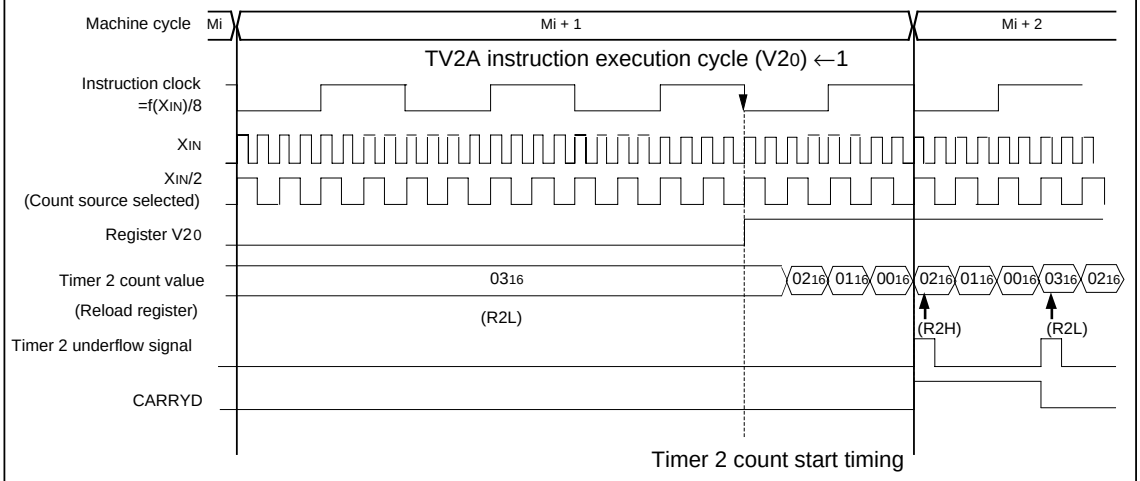


Note: When to expand "H" interval of the carrier wave is valid, set "01_{16}" or more to reload register R2H.

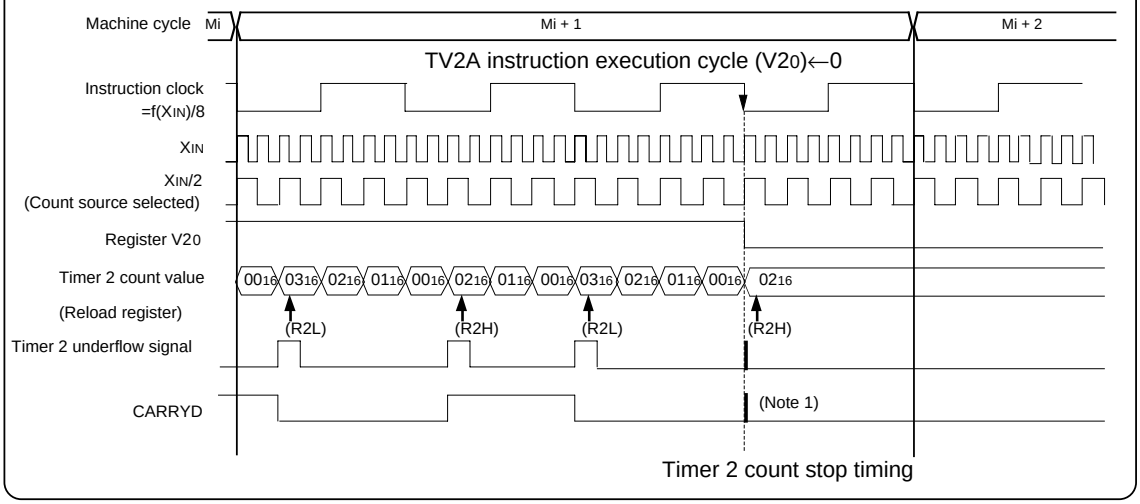
Fig. 15 Carrier wave generation example by timer 2

● In this case, the following is set;
- To expand "H" interval of carrier wave is invalid ($V2_3$ = "0"),
- Timer 2 carrier wave generation function is valid ($V2_2$="1"),
- Count source $X_{IN}/2$ selected ($V2_1$="1"),
- "L" interval (03_{16}) of carrier wave is set to reload register R2L
- "H" interval (02_{16}) of carrier wave is set to reload register R2H

Timer 2 count start timing



Timer 2 count stop timing



Notes 1: When the carrier wave generation function is vaild ($V2_2$="1"), avoid a timing when timer 2 underflows to stop timer 2. When the timer 2 count stop occurs at the same timing with the timer 2 underflows, hazard may occur in the carrier wave output waveform.

2: When the timer 2 is stopped during "H" output of carrier wave while the carrier wave generation function is valid, it is stopped after the "H" interval set by reload register R2H is output.

Fig. 16 Timer 2 count start/stop timing

WATCHDOG TIMER

Watchdog timer provides a method to reset and restart the system when a program runs wild. Watchdog timer consists of 14-bit timer (WDT) and watchdog timer flags (WDF1, WDF2).

Watchdog timer downcounts the instruction clock (INSTCK) as the count source immediately after system is released from reset. When the timer WDT count value becomes 0000_{16} and underflow occurs, the WDF1 flag is set to "1." Then, when the WRST instruction is not executed before the timer WDT counts 16383, WDF2 flag is set to "1" and internal reset signal is generated and system reset is performed.

Execute the WRST instruction at period of 16383 machine cycle or less to keep the microcomputer operation normal.

Timer WDT is also used for generation of oscillation stabilization time. When system is returned from reset and from RAM back-up mode by key-input, software starts after the stabilization oscillation time until timer WDT downcounts to $3E00_{16}$ elapses.

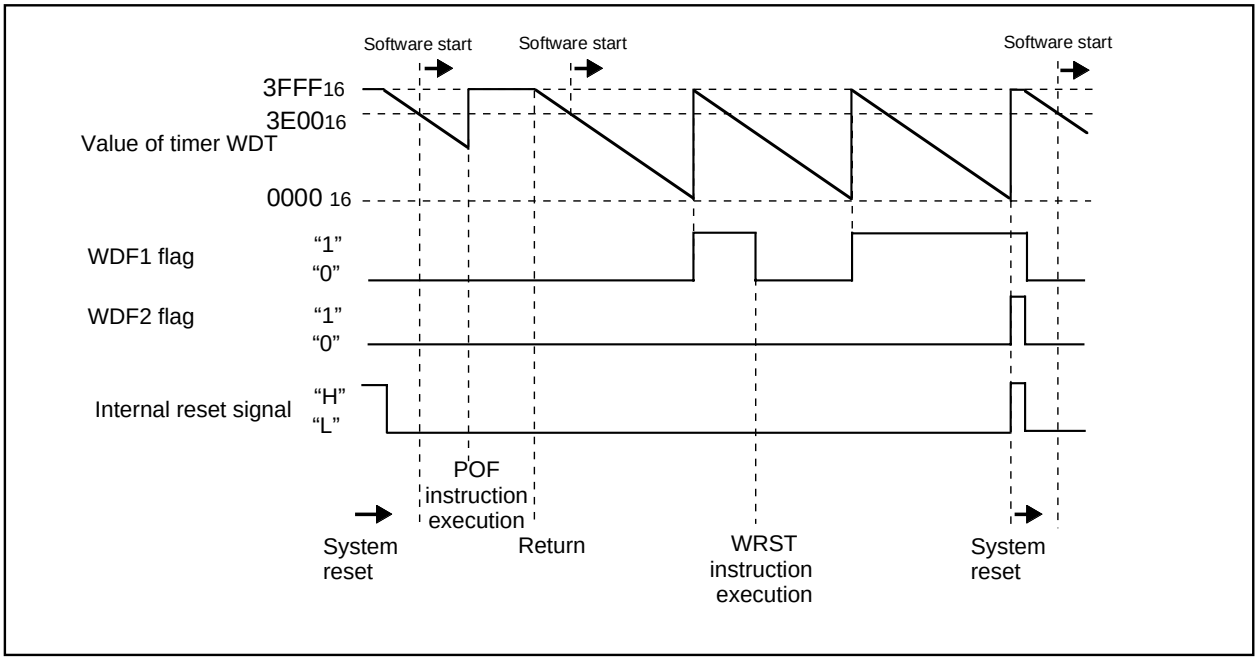


Fig. 17 Watchdog timer function

LOGIC OPERATION FUNCTION

The 4282 Group has the 4-bit logic operation function. The logic operation between the contents of register A and the low-order 4 bits of register E is performed and its result is stored in register A.

Each logic operation can be selected by setting logic operation selection register LO.

Set the contents of this register through register A with the TLOA instruction. The logic operation selected by register LO is executed with the LGOP instruction.

Table 5 shows the logic operation selection register LO.

Table 5 Logic operation selection register LO

Logic operation selection register LO		at reset : 00_2		at RAM back-up : 00_2	W
		LO_1	LO_0	Logic operation function	
LO_1	Logic operation selection bits	0	0	Exclusive logic OR operation (XOR)	
		0	1	OR operation (OR)	
LO_0		1	0	AND operation (AND)	
		1	1	Not available	

Note: "W" represents write enabled.

RESET FUNCTION

The 4282 Group has the power-on reset circuit, though it does not have $\overline{\text{RESET}}$ pin. System reset is performed automatically at power-on, and software starts program from address 0 in page 0.

In order to make the built-in power-on reset circuit operate efficiently, set the voltage rising time until V_{DD}= 0 to 2.2 V is obtained at power-on 1ms or less.

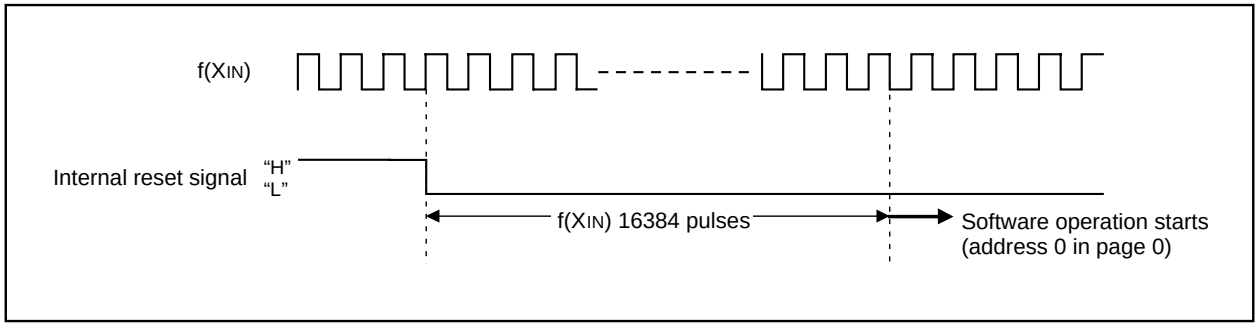


Fig. 18 Reset release timing

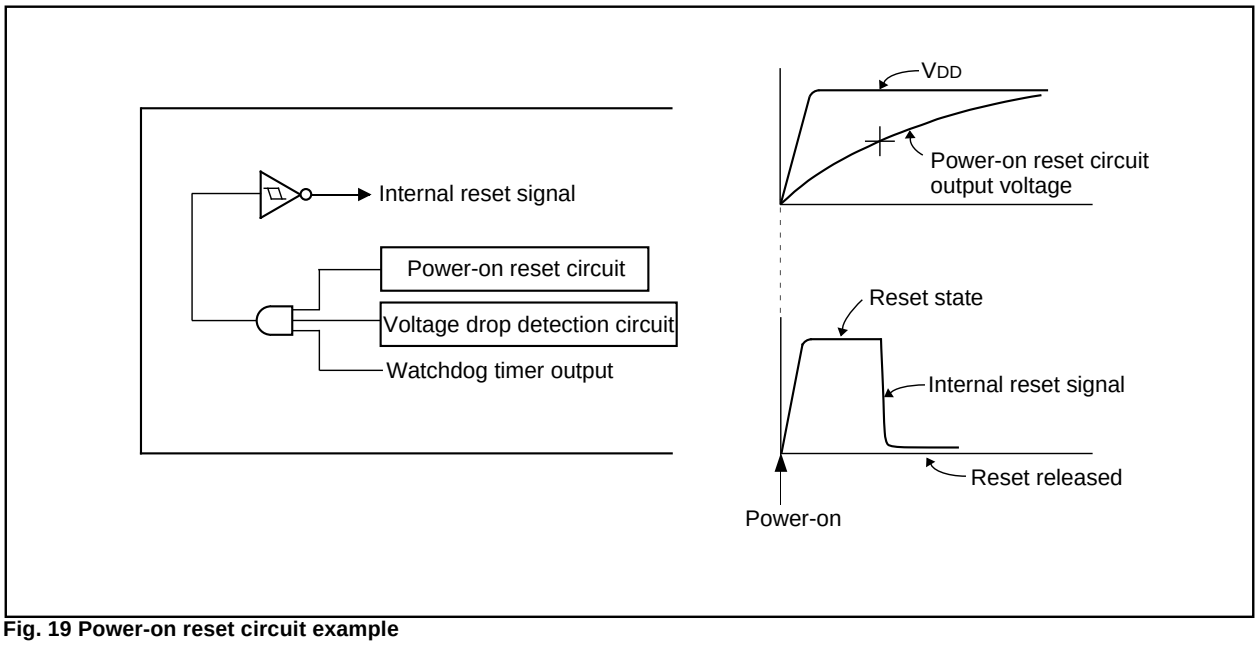


Fig. 19 Power-on reset circuit example

(1) Internal state at reset

Table 6 shows port state at reset, and Figure 20 shows internal state at reset (they are retained after system is released from reset).

The contents of timers, registers, flags and RAM except shown in Figure 20 are undefined, so set the initial value to them.

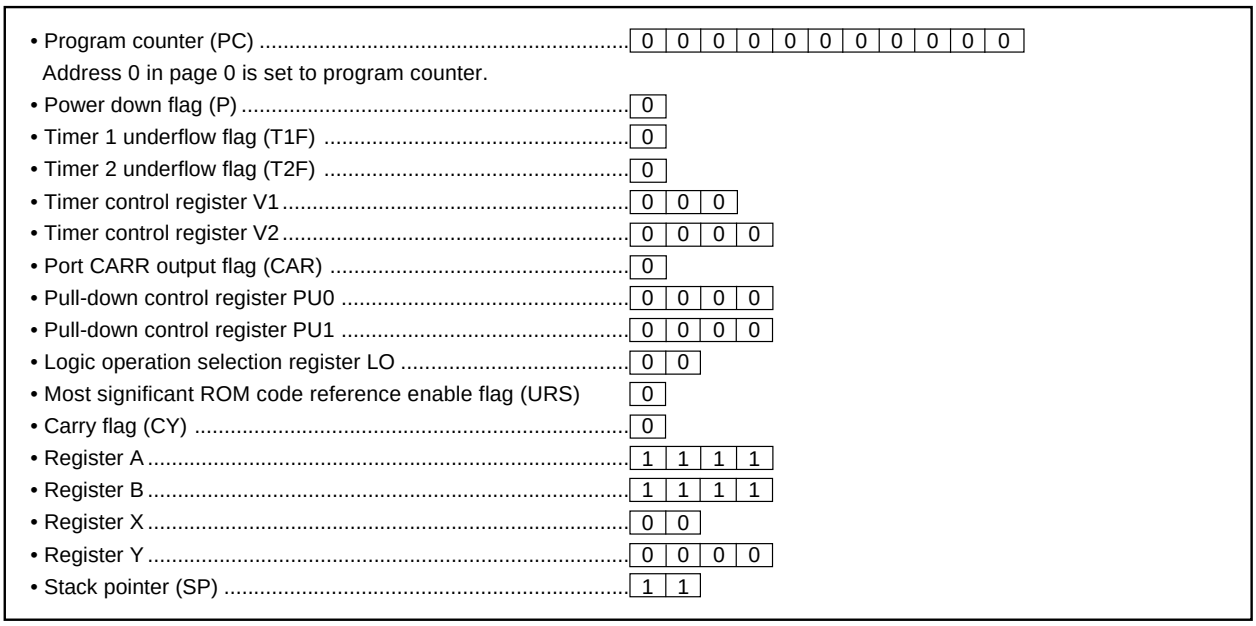


Fig. 20 Internal state at reset

Table 6 Port state at reset

Name	State at reset
D_0–D_3	High impedance state
D_4–D_7	High impedance state (Pull-down transistor OFF)
G_0–G_3	High impedance state (Pull-down transistor OFF)
E_0, E_1	High impedance state (Pull-down transistor OFF)
CARR	"L" output

Note: The contents of all output latch is initialized to "0."

VOLTAGE DROP DETECTION CIRCUIT

The built-in voltage drop detection circuit is designed to detect a drop in voltage at operating and to reset the microcomputer if the supply voltage drops below the specified value (Typ. 1.50 V) or less.

The voltage drop detection circuit is stopped and power dissipation is reduced in the RAM back-up mode with the initialized CPU stopped.

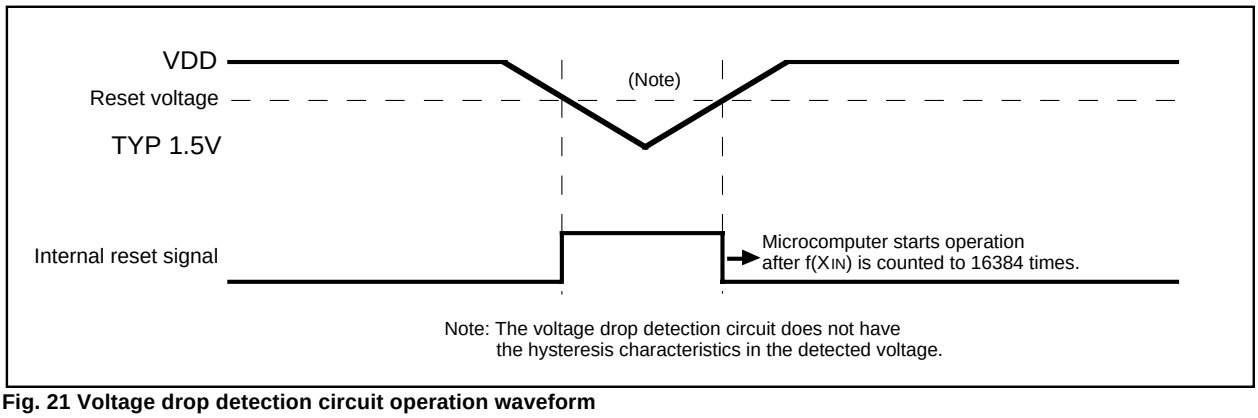


Fig. 21 Voltage drop detection circuit operation waveform

RAM BACK-UP MODE

The 4282 Group has the RAM back-up mode.
When the POF instruction is executed, system enters the RAM back-up state.
As oscillation stops retaining RAM, the functions and states of reset circuit at RAM back-up mode, power dissipation can be reduced without losing the contents of RAM. Table 7 shows the function and states retained at RAM back-up. Figure 22 shows the state transition.

(1) Warm start condition

When the external wakeup signal is input after the system enters the RAM back-up state by executing the POF instruction, the CPU starts executing the software from address 0 in page 0. In this case, the P flag is "1."

(2) Cold start condition

The CPU starts executing the software from address 0 in page 0 when any of the following conditions is satisfied .

- reset by power-on reset circuit is performed
- reset by watchdog timer is performed
- reset by voltage drop detection circuit is performed

In this case, the P flag is "0."

(3) Identification of the start condition

Warm start (return from the RAM back-up state) or cold start (return from the normal reset state) can be identified by examining the state of the power down flag (P) with the SNZP instruction.

Table 7 Functions and states retained at RAM back-up

Function	RAM back-up
Program counter (PC), registers A, B, carry flag (CY), stack pointer (SP) (Note 2)	×
Contents of RAM	O
Port CARR	×
Ports D_0–D_7	O
Ports E_0, E_1	O
Port G	O
Timer control registers V1, V2	×
Pull-down control registers PU0, PU1	O
Logic operation selection register LO	×
Timer 1 function, Timer 2 function	×
Timer underflow flags (T1F, T2F)	×
Watchdog timer (WDT)	×
Watchdog timer flags (WDF1, WDF2)	×
MostsignificantROMcodereferenceenableflag(URS)	×

Notes 1: "O" represents that the function can be retained, and "×" represents that the function is initialized.
Registers and flags other than the above are undefined at RAM back-up, and set an initial value after returning.
2:The stack pointer (SP) points the level of the stack register and is initialized to "11_2" at RAM back-up.

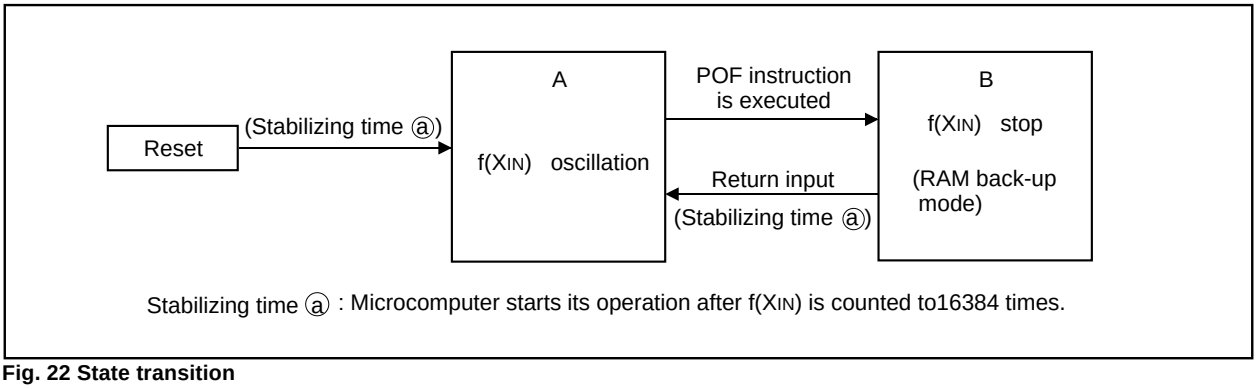


Fig. 22 State transition

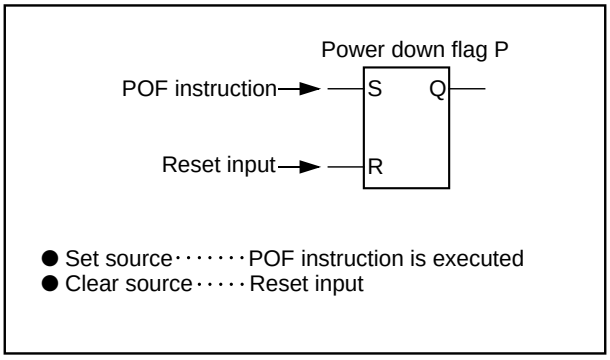


Fig. 23 Set source and clear source of the P flag

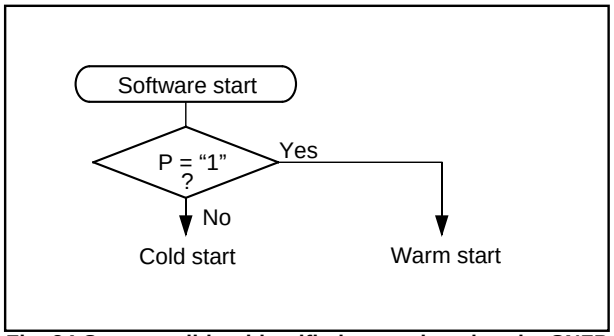


Fig. 24 Start condition identified example using the SNZP instruction

(4) Return signal

An external wakeup signal is used to return from the RAM back-up mode. Table 8 shows the return condition for each return source.

Table 8 Return source and return condition

Return source	Return condition	Remarks
Ports D_4–D_7	Return by an external "H" level input.	Only key-on wakeup function of the port whose pull-down transistor is turned ON by register PU1 is valid.
Ports E_0, E_1, G	Return by an external "H" level input.	Only key-on wakeup function of the port whose pull-down transistor is turned ON by register PU0 is valid.
Ports E_2	Return by an external "H" level input.	Key-on wakeup function is always valid.

(5) Pull-down control register

Registers PU0 and PU1 are 4-bit registers and control the ON/OFF of pull-down transistor and key-on wakeup function for ports E_0, E_1, G and ports D_4–D_7.

Set the contents of register PU0 or PU1 through register A with the TPU0A or TPU1A instruction, respectively.

Table 9 Pull-down control registers

Pull-down control register PU0		at reset : 0000_2		at RAM back-up : state retained	W
$PU0_3$	Ports G_2, G_3 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid		
		1	Pull-down transistor ON, key-on wakeup valid		
$PU0_2$	Ports G_0, G_1 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid		
		1	Pull-down transistor ON, key-on wakeup valid		
$PU0_1$	Port E_1 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid		
		1	Pull-down transistor ON, key-on wakeup valid		
$PU0_0$	Port E_0 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid		
		1	Pull-down transistor ON, key-on wakeup valid		

Pull-down control register PU1		at reset : 0000_2		at RAM back-up : state retained	W
$PU1_3$	Port D_7 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid		
		1	Pull-down transistor ON, key-on wakeup valid		
$PU1_2$	Port D_6 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid		
		1	Pull-down transistor ON, key-on wakeup valid		
$PU1_1$	Port D_5 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid		
		1	Pull-down transistor ON, key-on wakeup valid		
$PU1_0$	Port D_4 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid		
		1	Pull-down transistor ON, key-on wakeup valid		

Note: "W" represents write enabled.

CLOCK CONTROL

The clock control circuit consists of the following circuits.

- System clock generating circuit
- Control circuit to stop the clock oscillation
- Control circuit to return from the RAM back-up state

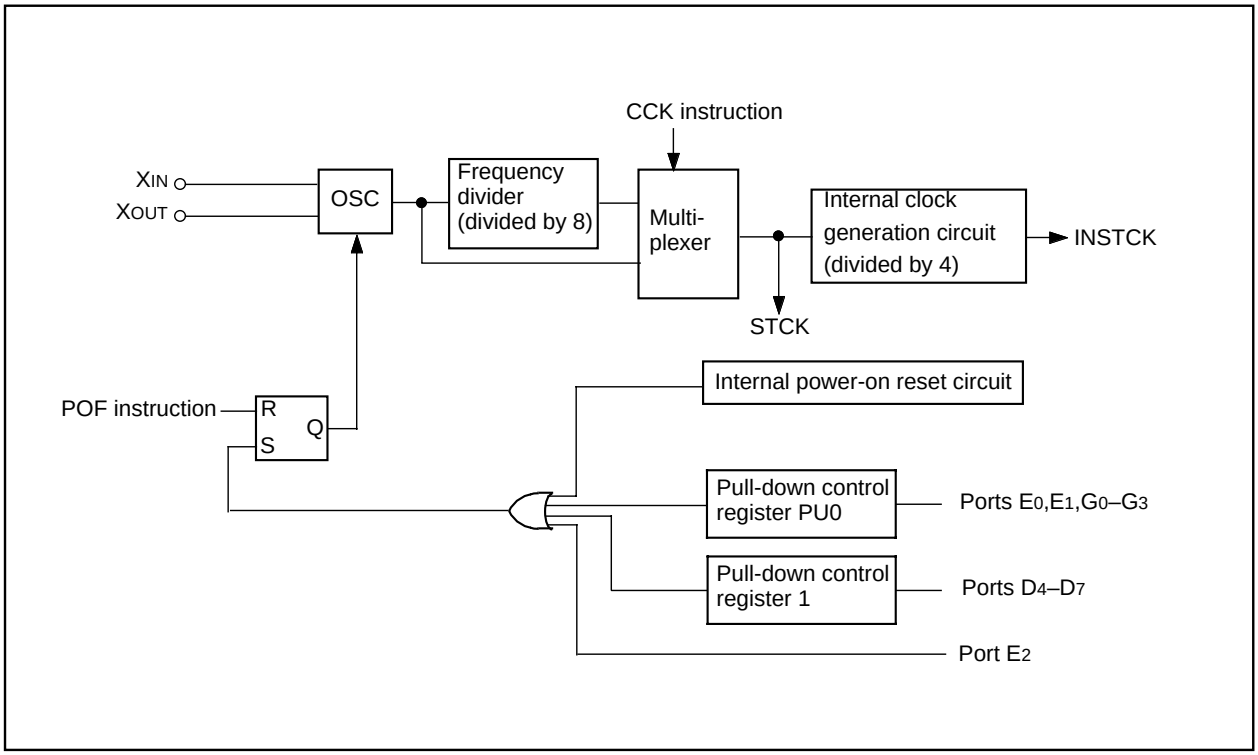


Fig. 25 Clock control circuit structure

System clock signal f(XIN) is obtained by externally connecting a ceramic resonator. Connect this external circuit to pins XIN and XOUT at the shortest distance as shown Figure 26.
A feedback resistor is built-in between XIN pin and XOUT pin.

ROM ORDERING METHOD

Please submit the information described below when ordering Mask ROM.

(1) Mask ROM Order Confirmation Form*
(2) Mark Specification Form
(3) Data to be written to ROM, in EPROM form (three identical copies) or one floppy disk.

* For the mask ROM confirmation, refer to the "Mitsubishi MCU Technical Information" Homepage (http://www.infomicom.maec.co.jp/indexe.htm).

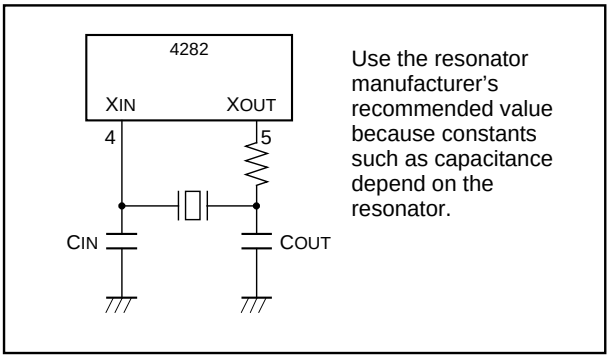


Fig. 26 Ceramic resonator external circuit

LIST OF PRECAUTIONS

① Noise and latch-up prevention

Connect a capacitor on the following condition to prevent noise and latch-up;

- connect a bypass capacitor (approx. 0.01 μF) between pins V_{DD} and V_{SS} at the shortest distance,
- equalize its wiring in width and length, and
- use the thickest wire.

In the One Time PROM version, port E_2 is also used as V_{PP} pin. Connect this pin to V_{SS} through the resistor about 5 kΩ which is assigned to E_2/V_{PP} pin as close as possible at the shortest distance.

② Notes on unused pins

(Note in order to set the output latch to "0" to make pins open)

- After system is released from reset, a port is in a high-impedance state until the output latch of the port is set to "0" by software.
  Accordingly, the voltage level of pins is undefined and the excess of the supply current may occur.
- To set the output latch periodically is recommended because the value of output latch may change by noise or a program run away (caused by noise).

(Note when connecting to V_{SS} and V_{DD})

- Connect the unused pins to V_{SS} and V_{DD} at the shortest distance and use the thick wire against noise.

③ Timer

- Count source
  Stop timer 1 or timer 2 counting to change its count source.
- Watchdog timer
  Be sure that the timing to execute the WRST instruction in order to operate WDT efficiently.
- Writing to reload register R1
  When writing data to reload register R1 while timer 1 is operating, avoid a timing when timer 1 underflows.
- Timer 1 count operation
  When the bit 5 of the watchdog timer (WDT) is selected as the timer 1 count source, the error of maximum ± 256 μs (at the minimum instruction execution time : 8 μs) is generated from timer 1 start until timer 1 underflow. When programming, be careful about this error.
- Stop of timer 2
  Avoid a timing when timer 2 underflows to stop timer 2.
- Writing to reload register R2H
  When writing data to reload register R2H while timer 2 is operating, avoid a timing when timer underflows.
- Timer 2 carrier wave output function
  When to expand "H" interval of carrier wave is valid, set "1" or more to reload register R2H.

④ Program counter

Make sure that the program counter does not specify after the last page of the built-in ROM.

INSTRUCTIONS

The 4282 Group has the 68 instructions. Each instruction is described as follows;
(1) List of instruction function
(2) Machine instructions (index by alphabet)
(3) Machine instructions (index by function)
(4) Instruction code table

SYMBOL

The symbols shown below are used in the following list of instruction function and the machine instructions.

Symbol	Contents	Symbol	Contents
A	Register A (4 bits)	D	Port D (8 bits)
B	Register B (4 bits)	E	Port E (3 bits)
DR	Register D (3 bits)	G	Port G (4 bits)
ER	Register E (8 bits)	CARR	Port CARR (1 bit)
V1	Timer control register V1 (3 bits)	CAR	CAR flag (1 bit)
V2	Timer control register V2 (4 bits)		
PU0	Pull-down control register PU0 (4 bits)	x	Hexadecimal variable
PU1	Pull-down control register PU1 (4 bits)	y	Hexadecimal variable
LO	Logic operation selection register LO (2 bits)	p	Hexadecimal variable
		n	Hexadecimal constant which represents the
X	Register X (2 bits)		immediate value
Y	Register Y (4 bits)	j	Hexadecimal constant which represents the
DP	Data pointer (6 bits)		immediate value
	(It consists of registers X and Y)	$A_3A_2A_1A_0$	Binary notation of hexadecimal variable A
PC	Program counter (11 bits)		(same for others)
PC_H	High-order 4 bits of program counter		
PC_L	Low-order 7 bits of program counter	←	Direction of data movement
SK	Stack register (11 bits × 4)	↔	Data exchange between a register and memory
SP	Stack pointer (2 bits)	?	Decision of state shown before "?"
CY	Carry flag	( )	Contents of registers and memories
R1	Timer 1 reload register	—	Negate, Flag unchanged after executing
T1	Timer 1		instruction
T1F	Timer 1 underflow flag	M(DP)	RAM address pointed by the data pointer
R2H	Timer 2 reload register	a	Label indicating address a_6 a_5 a_4 a_3 a_2 a_1 a_0
R2L	Timer 2 reload register	p, a	Label indicating address a_6 a_5 a_4 a_3 a_2 a_1 a_0
T2	Timer 2		in page p_3 p_2 p_1 p_0
T2F	Timer 2 underflow flag	C	Hex. number C + Hex. number x (also same for
WDT	Watchdog timer	+	others)
WDF1	Watchdog timer flag 1	x	
WDF2	Watchdog timer flag 2		
URS	Most significant ROM code reference enable flag		
P	Power down flag		
STCK	System clock		
INSTCK	Instruction clock		

Note : The 4282 Group just invalidates the next instruction when a skip is performed. The contents of program counter is not increased by 2. Accordingly, the number of cycles does not change even if skip is not performed. However, the cycle count becomes "1" if the TABP p, RT, or RTS instruction is skipped.

LIST OF INSTRUCTION FUNCTION

Grouping	Mnemonic	Function	Page
Register to register transfer	TAB	(A) ← (B)	38
	TBA	(B) ← (A)	40
	TAY	(A) ← (Y)	40
	TYA	(Y) ← (A)	42
	TEAB	(ER_7–ER_4) ← (B) (ER_3–ER_0) ← (A)	41
	TABE	(B) ← (ER_7–ER_4) (A) ← (ER_3–ER_0)	39
	TDA	(DR_2–DR_0) ← (A_2–A_0)	40
RAM addresses	LXY x, y	(X) ← x, x = 0 to 3 (Y) ← y, y = 0 to 15	31
	INY	(Y) ← (Y) + 1	31
	DEY	(Y) ← (Y) – 1	30
RAM to register transfer	TAM j	(A) ← (M(DP)) (X) ← (X) EXOR(j) j = 0 to 3	40
	XAM j	(A) ←→ (M(DP)) (X) ← (X) EXOR(j) j = 0 to 3	43
	XAMD j	(A) ←→ (M(DP)) (X) ← (X) EXOR(j) j = 0 to 3 (Y) ← (Y) – 1	43
	XAMI j	(A) ←→ (M(DP)) (X) ← (X) EXOR(j) j = 0 to 3 (Y) ← (Y) + 1	43
Arithmetic operation	LA n	(A) ← n n = 0 to 15	31
	TABP p	(SP) ← (SP) + 1 (SK(SP)) ← (PC) (PC_H) ← p p=0 to 15 (PC_L) ← (DR_2–DR_0, A_3–A_0) When URS=0 (B) ← (ROM(PC))$_{7\text{ to }4}$ (A) ← (ROM(PC))$_{3\text{ to }0}$ When URS=1 (CY) ← (ROM(PC))$_8$ (B) ← (ROM(PC))$_{7\text{ to }4}$ (A) ← (ROM(PC))$_{3\text{ to }0}$ (PC) ← (SK(SP)) (SP) ← (SP) – 1	39
	AM	(A) ← (A) + (M(DP))	27
	AMC	(A) ← (A) + (M(DP)) + (CY) (CY) ← Carry	27
	A n	(A) ← (A) + n n = 0 to 15	27
	SC	(CY) ← 1	35
	RC	(CY) ← 0	33
	SZC	(CY) = 0 ?	37
	CMA	(A) ← $\overline{(A)}$	30
	RAR	→CY→$A_3A_2A_1A_0$	33
	LGOP	Logic operation instruction XOR, OR, AND	31
Bit operation	SB j	(Mj(DP)) ← 1 j = 0 to 3	34
	RB j	(Mj(DP)) ← 0 j = 0 to 3	33
	SZB j	(Mj(DP)) = 0 ? j = 0 to 3	37

Grouping	Mnemonic	Function	Page
Comparison operation	SEAM	(A) = (M(DP)) ?	36
	SEA n	(A) = n ? n = 0 to 15	35
Branch operation	B a	$(PC_L) \leftarrow a_6–a_0$	27
	BL p, a	$(PC_H) \leftarrow p$ $(PC_L) \leftarrow a_6–a_0$	28
	BA a	$(PC_L) \leftarrow (a_6–a_4, A_3–A_0)$	28
	BLA p, a	$(PC_H) \leftarrow p$ $(PC_L) \leftarrow (a_6–a_4, A_3–A_0)$	28
Subroutine operation	BM a	(SP) ← (SP) + 1 (SK(SP)) ← (PC) $(PC_H) \leftarrow 2$ $(PC_L) \leftarrow a_6–a_0$	28
	BML p, a	(SP) ← (SP) + 1 (SK(SP)) ← (PC) $(PC_H) \leftarrow p$ p= 0 to 15 $(PC_L) \leftarrow a_6–a_0$	29
	BMLA p, a	(SP) ← (SP) + 1 (SK(SP)) ← (PC) $(PC_H) \leftarrow p$ p= 0 to 15 $(PC_L) \leftarrow (a_6–a_4, A_3–A_0)$	29
Return operation	RT	(PC) ← (SK(SP)) (SP) ← (SP) – 1	34
	RTS	(PC) ← (SK(SP)) (SP) ← (SP) – 1	34

Grouping	Mnemonic	Function	Page
Timer operation	TV1A	$(V1_2–V1_0) \leftarrow (A_2–A_0)$	42
	TAB1	$(B) \leftarrow (T1_7–T1_4)$ $(A) \leftarrow (T1_3–T1_0)$	39
	T1AB	at timer 1 stop ($V1_0$=0): $(R1_7–R1_4) \leftarrow (B)$ $(T1_7–T1_4) \leftarrow (B)$ $(R1_3–R1_0) \leftarrow (A)$ $(T1_3–T1_0) \leftarrow (A)$ at timer 1 operating ($V1_0$=1): $(R1_7–R1_4) \leftarrow (B)$ $(R1_3–R1_0) \leftarrow (A)$	37
	SNZT1	(T1F) = 1 ? After skipping the next instruction (T1F) ← 0	36
	TV2A	$(V2_3–V2_0) \leftarrow (A_3–A_0)$	42
	TAB2	$(B) \leftarrow (T2_7–T2_4)$ $(A) \leftarrow (T2_3–T2_0)$	39
	T2AB	$(R2L_7–R2L_4) \leftarrow (B)$ $(T2_7–T2_4) \leftarrow (B)$ $(R2L_3–R2L_0) \leftarrow (A)$ $(T2_3–T2_0) \leftarrow (A)$	38
	T2HAB	$(R2H_7–R2H_4) \leftarrow (B)$ $(R2H_3–R2H_0) \leftarrow (A)$	38
	T2R2L	$(T2_7–T2_4) \leftarrow (R2L_7–R2L_4)$ $(T2_7–T2_4) \leftarrow (R2L_3–R2L_0)$	38
	SNZT2	(T2F) = 1 ? After skipping the next instruction (T2F) ← 0	36

LIST OF INSTRUCTION FUNCTION (CONTINUED)

Grouping	Mnemonic	Function	Page
Input/Output operation	CLD	(D) ← 0	29
	RD	(D(Y)) ← 0 (Y) = 0 to 7	34
	SD	(D(Y)) ← 1 (Y) = 0 to 7	35
	SZD	(D(Y)) = 0 ? (Y) = 4 to 7	37
	OEA	(E_1, E_0) ← (A_1, A_0)	32
	IAE	(A_2–A_0) ← (E_2–E_0)	30
	OGA	(G) ← (A)	32
	IAG	(A) ← (G)	30
Carrier wave control operation	SCAR	(CAR) ← 1	35
	RCAR	(CAR) ← 0	33
Other operation	NOP	(PC) ← (PC) + 1	32
	POF	RAM back-up	32
	SNZP	(P) = 1 ?	36
	CCK	STCK changes to f(X_{IN})	29
	TLOA	(LO_1, LO_0) ← (A_1, A_0)	41
	URSC	(URS) ← 1	42
	TPU0A	($PU0_3$–$PU0_0$) ← (A_3–A_0)	41
	TPU1A	($PU1_3$–$PU1_0$) ← (A_3–A_0)	41
	WRST	(WDF1) ← 0	43

MACHINE INSTRUCTIONS (INDEX BY ALPHABET)

A n (Add n and accumulator)

Instruction code

D8								D0		
0	1	0	1	0	n_3	n_2	n_1	n_0	$_2$	0 A n $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	Overflow = 0

Operation: (A) ← (A) + n
n = 0 to 15

Grouping: Arithmetic operation

Description: Adds the value n in the immediate field to register A.
The contents of carry flag CY remains unchanged.
Skips the next instruction when there is no overflow as the result of operation.

AM (Add accumulator and Memory)

Instruction code

D8								D0		
0	0	0	0	0	1	0	1	0	$_2$	0 0 A $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← (A) + (M(DP))

Grouping: Arithmetic operation

Description: Adds the contents of M(DP) to register A. Stores the result in register A. The contents of carry flag CY remains unchanged.

AMC (Add accumulator, Memory and Carry)

Instruction code

D8								D0		
0	0	0	0	0	1	0	1	1	$_2$	0 0 B $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	0/1	–

Operation: (A) ← (A) + (M(DP)) + (CY)
(CY) ← Carry

Grouping: Arithmetic operation

Description: Adds the contents of M(DP) and carry flag CY to register A. Stores the result in register A and carry flag CY.

B a (Branch to address a)

Instruction code

D8								D0		
1	1	a_6	a_5	a_4	a_3	a_2	a_1	a_0	$_2$	1 8+a a $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (PC_L) ← a_6–a_0

Grouping: Branch operation

Description: Branch within a page : Branches to address a in the identical page.

BA a (Branch to address a + Accumulator)

Instruction code

D8								D0		
0	0	0	0	0	0	0	0	1	$_2$	0 0 1 $_{16}$
1	1	a_6	a_5	a_4	a_3	a_2	a_1	a_0	$_2$	1 8+a $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
2	2	–	–

Grouping: Branch operation

Operation: $(PC_L) \leftarrow a_6$–a_4, A_3–A_0

Description: Branch within a page : Branches to address (a_6 a_5 a_4 A_3 A_2 A_1 A_0) determined by replacing the low-order 4 bits of the address a in the identical page with register A.

BL p, a (Branch Long to address a in page p)

Instruction code

D8								D0		
0	0	0	1	1	p_3	p_2	p_1	p_0	$_2$	0 3 p $_{16}$
1	1	a_6	a_5	a_4	a_3	a_2	a_1	a_0	$_2$	1 8+a $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
2	2	–	–

Grouping: Branch operation

Operation:
$(PC_H) \leftarrow (P)$
$(PC_L) \leftarrow a_6$–a_0

Description: Branch out of a page : Branches to address a in page p.

Note: p is 0 to 7 for M34282M1,
p is 0 to 15 for M34282M2/E2.

BLA p, a (Branch Long to address a in page p)

Instruction code

D8								D0		
0	0	0	0	1	0	0	0	0	$_2$	0 1 0 $_{16}$
1	1	a_6	a_5	a_4	p_3	p_2	p_1	p_0	$_2$	1 8+a p $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
2	2	–	–

Grouping: Branch operation

Operation:
$(PC_H) \leftarrow (P)$
$(PC_L) \leftarrow (a_6$–a_4, A_3–$A_0)$

Description: Branch within a page : Branches to address (a_6 a_5 a_4 A_3 A_2 A_1 A_0) determined by replacing the low-order 4 bits of the address a in page p with register A.

Note: p is 0 to 7 for M34282M1,
p is 0 to 15 for M34282M2/E2.

BM a (Branch and Mark to address a in page 2)

Instruction code

D8								D0		
1	0	a_6	a_5	a_4	a_3	a_2	a_1	a_0	$_2$	1 a a $_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Grouping: Subroutine call operation

Operation:
$(SK(SP)) \leftarrow (PC)$
$(SP) \leftarrow (SP) + 1$
$(PC_H) \leftarrow 2$
$(PC_L) \leftarrow a_6$–a_0

Description: Call the subroutine in page 2 : Calls the subroutine at address a in page 2.

BML p, a (Branch and Mark Long to address a in page p)

Instruction code

	D8							D0	
	0	0	1	1	1	p3	p2	p1	p0
	1	0	a6	a5	a4	a3	a2	a1	a0

0	7	p	16
1	a	a	16

Number of words	Number of cycles	Flag CY	Skip condition
2	2	–	–

Operation:
(SK(SP)) ← (PC)
(SP) ← (SP) + 1
(PC_H) ← p
(PC_L) ← a_6–a_0

Grouping: Subroutine call operation

Description: Call the subroutine : Calls the subroutine at address a in page p.

Note: p is 0 to 7 for M34282M1,
p is 0 to 15 for M34282M2/E2.

BMLA p, a (Branch and Mark Long to address a in page p)

Instruction code

	D8							D0	
	0	0	1	0	1	0	0	0	0
	1	0	a6	a5	a4	p3	p2	p1	p0

0	5	0	16
1	a	p	16

Number of words	Number of cycles	Flag CY	Skip condition
2	2	–	–

Operation:
(SK(SP)) ← (PC)
(SP) ← (SP) + 1
(PC_H) ← p
(PC_L) ← (a_6–a_4, A_3–A_0)

Grouping: Subroutine call operation

Description: Call the subroutine : Calls the subroutine at address (a_6 a_5 a_4 A_3 A_2 A_1 A_0) determined by replacing the low-order 4 bits of address a in page p with register A.

Note: p is 0 to 7 for M34282M1,
p is 0 to 15 for M34282M2/E2.

CCK (Change system Clock to f(XIN))

Instruction code

	D8							D0	
	0	0	1	0	1	1	0	0	1

0	5	9	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: Change to STCK = f(XIN)

Grouping: Other operation

Description: Changes system clock (STCK) from f(XIN)/8 to f(XIN). Execute this instruction at address 0 in page 0.

CLD (CLear port D)

Instruction code

	D8							D0	
	0	0	0	0	1	0	0	0	1

0	1	1	16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (D) ← 1

Grouping: Input/Output operation

Description: Clears (0) to port D (high-impedance state).

CMA (CoMplement of Accumulator)

Instruction code

D8								D0			
0	0	0	0	1	1	1	0	0	$_2$	0 1 C	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← $\overline{(A)}$

Grouping: Arithmetic operation

Description: Stores the one's complement for register A's contents in register A.

DEY (DEcrement register Y)

Instruction code

D8								D0			
0	0	0	0	1	0	1	1	1	$_2$	0 1 7	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(Y) = 15

Operation: (Y) ← (Y) – 1

Grouping: RAM addresses

Description: Subtracts 1 from the contents of register Y. As a result of subtraction, when the contents of register Y is 15, the next instruction is skipped.

IAE (Input Accumulator from port E)

Instruction code

D8								D0			
0	0	1	0	1	0	1	1	0	$_2$	0 5 6	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A_2–A_0) ← (E_2–E_0)

Grouping: Input/Output operation

Description: Transfers the contents of port E to register A.

IAG (Input Accumulator from port G)

Instruction code

D8								D0			
0	0	0	1	0	1	0	0	0	$_2$	0 2 8	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← (G)

Grouping: Input/Output operation

Description: Transfers the contents of port G to register A.

INY (INcrement register Y)

Instruction code

D8								D0		
0	0	0	0	1	0	0	1	1	0	1	3
$_2$									$_{16}$	

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(Y) = 0

Operation: (Y) ← (Y) + 1

Grouping: RAM addresses

Description: Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped.

LA n (Load n in Accumulator)

Instruction code

D8								D0			
0	1	0	1	1	n_3	n_2	n_1	n_0	0	B	n
$_2$									$_{16}$		

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	Continuous description

Operation: (A) ← n
n = 0 to 15

Grouping: Arithmetic operation

Description: Loads the value n in the immediate field to register A.
When the LA instructions are continuously coded and executed, only the first LA instruction is executed and other LA instructions coded continuously are skipped.

LGOP (LoGic OPeration between accumulator and register E)

Instruction code

D8								D0			
0	0	1	0	0	0	0	0	1	0	4	1
$_2$									$_{16}$		

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: Logic operation XOR, OR, AND

Grouping: Arithmetic operation

Description: Executes the logic operation selected by logic operation selection register LO between the contents of register A and register E, and stores the result in register A.

LXY x, y (Load register X and Y with x and y)

Instruction code

D8								D0			
0	1	1	x_1	x_0	y_3	y_2	y_1	y_0	0	C +x	y
$_2$									$_{16}$		

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	Continuous description

Operation: (X) ← x, x = 0 to 3
(Y) ← y, y = 0 to 15

Grouping: RAM addresses

Description: Loads the value x in the immediate field to register X, and the value y in the immediate field to register Y. When the LXY instructions are continuously coded and executed, only the first LXY instruction is executed and other LXY instructions coded continuously are skipped.

NOP (No OPeration)

Instruction code	D8								D0			
	0	0	0	0	0	0	0	0	0	0	0	0
	0_2									0_{16}		

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (PC) ← (PC) + 1

Grouping: Other operation

Description: No operation

OEA (Output port E from Accumulator)

Instruction code	D8								D0			
	0	1	0	0	0	0	1	0	0	0	8	4
	0_2									0_{16}		

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (E_1, E_0) ← (A_1, A_0)

Grouping: Input/Output operation

Description: Outputs the contents of register A to port E.

OGA (Output port G from Accumulator)

Instruction code	D8								D0			
	0	1	0	0	0	0	0	0	0	0	8	0
	0_2									0_{16}		

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (G) ← (A)

Grouping: Input/Output operation

Description: Outputs the contents of register A to port G.

POF (Power OFf1)

Instruction code	D8								D0			
	0	0	0	0	0	1	1	0	1	0	0	D
	0_2									0_{16}		

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: RAM back-up

Grouping: Other operation

Description: Puts the system in RAM back-up state.

RAR (Rotate Accumulator Right)

Instruction code

D8								D0	
0	0	0	0	1	1	1	0	1	$_2$

0	1	D	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	0/1	–

Operation: →CY→A3A2A1A0→

Grouping: Arithmetic operation

Description: Rotates 1 bit of the contents of register A including the contents of carry flag CY to the right.

RB j (Reset Bit)

Instruction code

D8								D0	
0	0	1	0	0	1	1	j1	j0	$_2$

0	4	C+j	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (Mj(DP)) ← 0
j = 0 to 3

Grouping: Bit operation

Description: Clears (0) the contents of bit j (bit specified by the value j in the immediate field) of M(DP).

RC (Reset Carry flag)

Instruction code

D8								D0	
0	0	0	0	0	0	1	1	0	$_2$

0	0	6	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	0	–

Operation: (CY) ← 0

Grouping: Arithmetic operation

Description: Clears (0) to carry flag CY.

RCAR (Reset CAR flag)

Instruction code

D8								D0	
0	1	0	0	0	0	1	1	0	$_2$

0	8	6	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (CAR) ← 0

Grouping: Carrier wave control operation

Description: Clears (0) to port CARR output flag.

RD (Reset port D specified by register Y)

Instruction code

D8								D0		
0	0	0	0	1	0	1	0	0	2	0 1 4 16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (D(Y)) ← 0
However,
(Y) = 0 to 7

Grouping: Input/Output operation

Description: Clears (0) to a bit of port D specified by register Y (high-impedance state).

RT (ReTurn from subroutine)

Instruction code

D8								D0		
0	0	1	0	0	0	1	0	0	2	0 4 4 16

Number of words	Number of cycles	Flag CY	Skip condition
1	2	–	–

Operation: (SP) ← (SP) – 1
(PC) ← (SK(SP))

Grouping: Return operation

Description: Returns from subroutine to the routine called the subroutine.

RTS (ReTurn form subroutine and Skip)

Instruction code

D8								D0		
0	0	1	0	0	0	1	0	1	2	0 4 5 16

Number of words	Number of cycles	Flag CY	Skip condition
1	2	–	Skip at uncondition

Operation: (SP) ← (SP) – 1
(PC) ← (SK(SP))

Grouping: Return operation

Description: Returns from subroutine to the routine called the subroutine, and skips the next instruction at uncondition.

SB j (Set Bit)

Instruction code

D8								D0		
0	0	1	0	1	1	1	j_1	j_0	2	0 5 C+j 16

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (Mj(DP)) ← 0
j = 0 to 3

Grouping: Bit operation

Description: Sets (1) the contents of bit j (bit specified by the value j in the immediate field) of M(DP).

SC (Set Carry flag)

Instruction code

D8								D0			
0	0	0	0	0	0	1	1	1	$_2$	0 0 7	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	1	–

Operation: (CY) ← 1

Grouping: Arithmetic operation

Description: Sets (1) to carry flag CY.

SCAR (Set CAR flag)

Instruction code

D8								D0			
0	1	0	0	0	0	1	1	1	$_2$	0 8 7	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (CAR) ← 1

Grouping: Carrier wave control operation

Description: Sets (1) to port CARR output flag (CAR).

SD (Set port D specified by register Y)

Instruction code

D8								D0			
0	0	0	0	1	0	1	0	1	$_2$	0 1 5	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (D(Y)) ← 1
(Y) = 0 to 7

Grouping: Input/Output operation

Description: Sets (1) to a bit of port D specified by register Y.

SEA n (Skip Equal, Accumulator with immediate data n)

Instruction code

D8								D0			
0	0	0	1	0	0	1	0	1	$_2$	0 2 5	$_{16}$
0	1	0	1	1	n_3	n_2	n_1	n_0	$_2$	0 B n	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
2	2	–	(A) = n, n = 0 to 15

Operation: (A) = n ?
n = 0 to 15

Grouping: Comparison operation

Description: Skips the next instruction when the contents of register A is equal to the value n in the immediate field.

SEAM (Skip Equal, Accumulator with Memory)

Instruction code

D8								D0	
0	0	0	1	0	0	1	1	0	$_2$

0	2	6	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(A) = (M(DP))

Operation: (A) = (M(DP)) ?

Grouping: Comparison operation

Description: Skips the next instruction when the contents of register A is equal to the contents of M(DP).

SNZP (Skip if Non Zero condition of Power down flag)

Instruction code

D8								D0	
0	0	0	0	0	0	0	1	1	$_2$

0	0	3	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(P) = 1

Operation: (P) = 1 ?

Grouping: Other operation

Description: Skips the next instruction when P flag is “1”. After skipping, P flag remains unchanged.

SNZT1 (Skip if Non Zero condition of Timer 1 underflow flag)

Instruction code

D8								D0	
0	0	1	0	0	0	0	1	0	$_2$

0	4	2	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(T1F) = 1

Operation: (T1F) = 1 ?
After skipping, (T1F) ← 0

Grouping: Timer operation

Description: Skips the next instruction when the contents of T1F flag is “1.”
After skipping, clears (0) to T1F flag.

SNZT2 (Skip if Non Zero condition of Timer 2 inerrupt request flag)

Instruction code

D8								D0	
0	0	1	0	1	0	0	1	0	$_2$

0	5	2	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(T2F) = 1

Operation: (T2F) = 1 ?
After skipping, (T2F) ← 0

Grouping: Timer operation

Description: Skips the next instruction when the contents of T2F flag is “1.”
After skipping, clears (0) to T2F flag.

SZB j (Skip if Zero, Bit)

Instruction code

D8								D0	
0	0	0	1	0	0	0	j_1	j_0	$_2$

0	2	j	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(Mj(DP)) = 0 j = 0 to 3

Operation: (Mj(DP)) = 0 ?
j = 0 to 3

Grouping: Bit operation

Description: Skips the next instruction when the contents of bit j (bit specified by the value j in the immediate field) of M(DP) is "0."

SZC (Skip if Zero, Carry flag)

Instruction code

D8									D0	
0	0	0	1	0	1	1	1	1		$_2$

0	2	F	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(CY) = 0

Operation: (CY) = 0 ?

Grouping: Arithmetic operation

Description: Skips the next instruction when the contents of carry flag CY is "0."

SZD (Skip if Zero, port D specified by register Y)

Instruction code

D8								D0	
0	0	0	1	0	0	1	0	0	$_2$
0	0	0	1	0	1	0	1	1	$_2$

0	2	4	$_{16}$
0	2	B	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
2	2	–	(D(Y)) = 0 (Y) = 4 to 7

Operation: (D(Y)) = 0 ?
(Y) = 4 to 7

Grouping: Input/Output operation

Description: Skips the next instruction when a bit of port D specified by register Y is "0."

T1AB (Transfer data to timer 1 and register R1 from Accumulator and register B)

Instruction code

D8								D0	
0	0	1	0	0	0	1	1	1	$_2$

0	4	7	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: at timer 1 stop ($V1_0$=0)
($R1_7$–$R1_4$) ← (B), ($R1_3$–$R1_0$) ← (A)
($T1_7$–$T1_4$) ← (B), ($T1_3$–$T1_0$) ← (A)
at timer 1 operating ($V1_0$=1)
($R1_7$–$R1_4$) ← (B), ($R1_3$–$R1_0$) ← (A)

Grouping: Timer operation

Description: At timer 1 stop ($V1_0$ = 0), transfers the contents of register A and register B to timer 1 and reload register R1.
At timer 1 operating ($V1_0$ = 1), transfers the contents of register A and register B to reload register R1.

T2AB (Transfer data to timer 2 and register R2L from Accumulator and register B)

Instruction code

D_8								D_0			
0	1	0	0	0	1	0	0	0	$_2$	0 8 8	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation:
$(R2L_7–R2L_4) \leftarrow (B)$
$(R2L_3–R2L_0) \leftarrow (A)$
$(T2_7–T2_4) \leftarrow (B)$
$(T2_3–T2_0) \leftarrow (A)$

Grouping: Timer operation

Description: Transfers the contents of registers A and B to timer 2 and timer 2 reload register R2L.

T2HAB (Transfer data to register R2H Accumulator from register B)

Instruction code

D_8								D_0			
0	1	0	0	0	1	0	0	1	$_2$	0 8 9	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation:
$(R2H_7–R2H_4) \leftarrow (B)$
$(R2H_3–R2H_0) \leftarrow (A)$

Grouping: Timer operation

Description: Transfers the contents of register A and register B to reload register R2H.

T2R2L (Transfer data to timer 2 from register R2L)

Instruction code

D_8								D_0			
0	0	1	0	1	0	0	1	1	$_2$	0 5 3	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation:
$(T2_7–T2_4) \leftarrow (R2L_7–R2L_4)$
$(T2_3–T2_0) \leftarrow (R2L_3–R2L_0)$

Grouping: Timer operation

Description: Transfers the contents of reload register R2L to timer 2.

TAB (Transfer data to Accumulator from register B)

Instruction code

D_8								D_0			
0	0	0	0	1	1	1	1	0	$_2$	0 1 E	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: $(A) \leftarrow (B)$

Grouping: Register to register transfer

Description: Transfers the contents of register B to register A.

TAB1 (Transfer data to Accumulator and register B from timer 1)

Instruction code

D8								D0	
0	0	1	0	1	0	1	1	1	$_2$

0	5	7	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation:
(B) ← ($T1_7$–$T1_4$)
(A) ← ($T1_3$–$T1_0$)

Grouping: Timer operation

Description: Transfers the contents of timer 1 to registers A and B.

TAB2 (Transfer data to Accumulator and register B from timer 2)

Instruction code

D8								D0	
0	0	1	0	0	0	0	0	0	$_2$

0	4	0	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation:
(B) ← ($T2_7$–$T2_4$)
(A) ← ($T2_3$–$T2_0$)

Grouping: Timer operation

Description: Transfers the contents of timer 2 to registers A and B.

TABE (Transfer data to Accumulator and register B from register E)

Instruction code

D8								D0	
0	0	0	1	0	1	0	1	0	$_2$

0	2	A	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation:
(B) ← (ER_7–ER_4)
(A) ← (ER_3–ER_0)

Grouping: Register to register transfer

Description: Transfers the contents of register E to registers A and B.

TABP p (Transfer data to Accumulator and register B from Program memory in page p)

Instruction code

D8								D0	
0	1	0	0	1	p_3	p_2	p_1	p_0	$_2$

0	9	p	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	3	– 0/1	–

Operation:
SK(SP)) ← (PC) , (SP) ← (SP) + 1
(PC_H) ← p, p = 0 to 7, (PC_L) ← (DR_2–DR_0, A_3–A_0)
When URS = 0,
(B) ← (ROM(PC))$_{7\ to\ 4}$, (A) ← (ROM(PC))$_{3\ to\ 0}$
When URS = 1,
(CY) ← (ROM(PC))$_8$
(B) ← (ROM(PC))$_{7\ to\ 4}$, (A) ← (ROM(PC))$_{3\ to\ 0}$
(SP) ← (SP) – 1, (PC) ← (SK(SP))

Note:
p is 0 to 7 for M34282M1,
p is 0 to 15 for M34282M2/E2.

Grouping: Arithmetic operation

Description:
Transfers bits 7 to 4 to register B and bits 3 to 0 to register A when URS flag is cleared to “0.” These bits 7 to 0 are the ROM pattern in address (DR_2 DR_1 DR_0 A_3 A_2 A_1 A_0) specified by registers A and D in page p.
Transfers bit 8 of ROM pattern is transferred to flag CY when URS flag is set to “1” (after the URSC instruction is executed).
(One of stack is used when the TABP p instruction is executed.)

TAM j (Transfer data to Accumulator from Memory)

Instruction code

D_8								D_0	
0	0	1	1	0	0	1	j_1	j_0	$_2$

0	6	4+j	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← (M(DP))
(X) ← (X)EXOR(j)
j = 0 to 3

Grouping: RAM to register transfer

Description: After transferring the contents of M(DP) to register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.

TAY (Transfer data to Accumulator from register Y)

Instruction code

D_8								D_0	
0	0	0	0	1	1	1	1	1	$_2$

0	1	F	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (A) ← (Y)

Grouping: Register to register transfer

Description: Transfers the contents of register Y to register A.

TBA (Transfer data to register B from Accumulator)

Instruction code

D_8								D_0	
0	0	0	0	0	1	1	1	0	$_2$

0	0	E	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (B) ← (A)

Grouping: Register to register transfer

Description: Transfers the contents of register A to register B.

TDA (Transfer data to register D from Accumulator)

Instruction code

D_8								D_0	
0	0	0	1	0	1	0	0	1	$_2$

0	2	9	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (DR_2–DR_0) ← (A_2–A_0)

Grouping: Register to register transfer

Description: Transfers the contents of register A to register D.

TEAB (Transfer data to register E from Accumulator and register B)

Instruction code

D8								D0	
0	0	0	0	1	1	0	1	0	$_2$

0	1	A	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (ER_7–ER_4) ← (B)
(ER_3–ER_0) ← (A)

Grouping: Register to register transfer

Description: Transfers the contents of register A and register B to register E.

TLOA (Transfer data to register LO from Accumulator)

Instruction code

D8								D0	
0	0	1	0	1	1	0	0	0	$_2$

0	5	8	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (LO_1, LO_0) ← (A_1, A_0)

Grouping: Other operation

Description: Transfers the contents of register A to logic operation selection register LO.

TPU0A (Transfer data to register PU0 from Accumulator)

Instruction code

D8								D0	
0	1	0	0	0	1	1	1	1	$_2$

0	8	F	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: ($PU0_3$–$PU0_0$) ← (A_3–A_0)

Grouping: Other operation

Description: Transfers the contents of register A to pull-up control register PU0.

TPU1A (Transfer data to register PU1 from Accumulator)

Instruction code

D8								D0	
0	1	0	0	0	1	1	1	0	$_2$

0	8	E	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: ($PU1_3$–$PU1_0$) ← (A_3–A_0)

Grouping: Other operation

Description: Transfers the contents of register A to pull-up control register PU1.

TV1A (Transfer data to register V1 from Accumulator)

Instruction code

D_8								D_0	
0	0	1	0	1	0	1	1	1	$_2$

0	5	B	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: $(V1_2–V1_0) \leftarrow (A_2–A_0)$

Grouping: Timer operation

Description: Transfers the contents of register A to register V1.

TV2A (Transfer data to register V2 from Accumulator)

Instruction code

D_8								D_0	
0	0	1	0	1	1	0	1	0	$_2$

0	5	A	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: $(V2_3–V2_0) \leftarrow (A_3–A_0)$

Grouping: Timer operation

Description: Transfers the contents of register A to register V2.

TYA (Transfer data to regiser Y from Accumulator)

Instruction code

D_8								D_0	
0	0	0	0	0	1	1	0	0	$_2$

0	0	C	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: $(Y) \leftarrow (A)$

Grouping: Register to register transfer

Description: Transfers the contents of register A to register Y.

URSC (Sets Upper ROM Code reference enable flag)

Instruction code

D_8								D_0	
0	1	0	0	0	0	0	1	0	$_2$

0	8	2	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: $(URS) \leftarrow 1$

Grouping: Other operation

Description: Sets the most significant ROM code reference enable flag (URS) to “1.”

WRST (Watchdog timer ReSeT)

Instruction code

D8								D0	
0	0	0	0	0	1	1	1	1	$_2$

0	0	F	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation: (WDF1) ← 0

Grouping: Other operation

Description: Initializes the watchdog timer flag (WDF1).

XAM j (eXchange Accumulator and Memory data)

Instruction code

D8								D0	
0	0	1	1	0	0	0	j_1	j_0	$_2$

0	6	j	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	–

Operation:
(A) ←→ (M(DP))
(X) ← (X)EXOR(j)
j = 0 to 3

Grouping: RAM to register transfer

Description: After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.

XAMD j (eXchange Accumulator and Memory data and Decrement register Y and skip)

Instruction code

D8								D0	
0	0	1	1	0	1	1	j_1	j_0	$_2$

0	6	C +j	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(Y) = 15

Operation:
(A) ←→ (M(DP))
(X) ← (X)EXOR(j)
j = 0 to 3
(Y) ← (Y) – 1

Grouping: RAM to register transfer

Description: After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.
Subtracts 1 from the contents of register Y. As a result of subtraction, when the contents of register Y is 15, the next instruction is skipped.

XAMI j (eXchange Accumulator and Memory data and Increment register Y and skip)

Instruction code

D8								D0	
0	0	1	1	0	1	0	j_1	j_0	$_2$

0	6	8 +j	$_{16}$

Number of words	Number of cycles	Flag CY	Skip condition
1	1	–	(Y) = 0

Operation:
(A) ←→ (M(DP))
(X) ← (X)EXOR(j)
j = 0 to 3
(Y) ← (Y) + 1

Grouping: RAM to register transfer

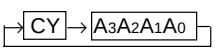
Description: After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.
Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped.

MACHINE INSTRUCTIONS (INDEX BY FUNCTION)

Type of instructions	Mnemonic	Instruction code D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal notation	Number of words	Number of cycles	Function
Register to register transfer	TAB	0	0	0	0	1	1	1	1	0	0 1 E	1	1	(A) ← (B)
	TBA	0	0	0	0	0	1	1	1	0	0 0 E	1	1	(B) ← (A)
	TAY	0	0	0	0	1	1	1	1	1	0 1 F	1	1	(A) ← (Y)
	TYA	0	0	0	0	0	1	1	0	0	0 0 C	1	1	(Y) ← (A)
	TEAB	0	0	0	0	1	1	0	1	0	0 1 A	1	1	(ER_7–ER_4) ← (B) (ER_3–ER_0) ← (A)
	TABE	0	0	0	1	0	1	0	1	0	0 2 A	1	1	(B) ← (ER_7–ER_4) (A) ← (ER_3–ER_0)
	TDA	0	0	0	1	0	1	0	0	1	0 2 9	1	1	(DR_2–DR_0) ← (A_2–A_0)
RAM addresses	LXY x, y	0	1	1	x_1	x_0	y_3	y_2	y_1	y_0	0 C +x y	1	1	(X) ← x, x = 0 to 3 (Y) ← y, y = 0 to 15
	INY	0	0	0	0	1	0	0	1	1	0 1 3	1	1	(Y) ← (Y) + 1
	DEY	0	0	0	0	1	0	1	1	1	0 1 7	1	1	(Y) ← (Y) – 1
RAM to register transfer	TAM j	0	0	1	1	0	0	1	j_1	j_0	0 6 4 +j	1	1	(A) ← (M(DP)) (X) ← (X) EXOR(j) j = 0 to 3
	XAM j	0	0	1	1	0	0	0	j_1	j_0	0 6 j	1	1	(A) ←→ (M(DP)) (X) ← (X) EXOR(j) j = 0 to 3
	XAMD j	0	0	1	1	0	1	1	j_1	j_0	0 6 C +j	1	1	(A) ←→ (M(DP)) (X) ← (X) EXOR(j) j = 0 to 3 (Y) ← (Y) – 1
	XAMI j	0	0	1	1	0	1	0	j_1	j_0	0 6 8 +j	1	1	(A) ←→ (M(DP)) (X) ← (X) EXOR(j) j = 0 to 3 (Y) ← (Y) + 1

Skip condition	Carry flag CY	Detailed description
–	–	Transfers the contents of register B to register A.
–	–	Transfers the contents of register A to register B.
–	–	Transfers the contents of register Y to register A.
–	–	Transfers the contents of register A to register Y.
–	–	Transfers the contents of registers A and B to register E.
–	–	Transfers the contents of register E to registers A and B.
–	–	Transfers the contents of register A to register D.
Continuous description	–	Loads the value x in the immediate field to register X, and the value y in the immediate field to register Y. When the LXY instructions are continuously coded and executed, only the first LXY instruction is executed and other LXY instructions coded continuously are skipped.
(Y) = 0	–	Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped.
(Y) = 15	–	Subtracts 1 from the contents of register Y. As a result of subtraction, when the contents of register Y is 15, the next instruction is skipped.
–	–	After transferring the contents of M(DP) to register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.
–	–	After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X.
(Y) = 15	–	After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. Subtracts 1 from the contents of register Y. As a result of subtraction, when the contents of register Y is 15, the next instruction is skipped.
(Y) = 0	–	After exchanging the contents of M(DP) with the contents of register A, an exclusive OR operation is performed between register X and the value j in the immediate field, and stores the result in register X. Adds 1 to the contents of register Y. As a result of addition, when the contents of register Y is 0, the next instruction is skipped.

MACHINE INSTRUCTIONS (CONTINUED)

Parameter / Type of instructions	Mnemonic	Instruction code												Number of words	Number of cycles	Function
		D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal notation					
Arithmetic operation	LA n	0	1	0	1	1	n_3	n_2	n_1	n_0	0	B	n	1	1	(A) ← n n = 0 to 15
	TABP p	0	1	0	0	1	p_3	p_2	p_1	p_0	0	9	p	1	3	(SK(SP)) ← (PC) (SP) ← (SP) + 1 (PC_H) ← p, p=0 to 7 (Note) (PC_L) ← (DR_2–DR_0, A_3–A_0) When URS=0, (B) ← (ROM(PC))$_{7\ to\ 4}$ (A) ← (ROM(PC))$_{3\ to\ 0}$ When URS=1, (CY) ← (ROM(PC))$_8$ (B) ← (ROM(PC))$_{7\ to\ 4}$ (A) ← (ROM(PC))$_{3\ to\ 0}$ (SP) ← (SP) – 1 (PC) ← (SK(SP))
	AM	0	0	0	0	0	1	0	1	0	0	0	A	1	1	(A) ← (A) + (M(DP))
	AMC	0	0	0	0	0	1	0	1	1	0	0	B	1	1	(A) ← (A) + (M(DP))+ (CY) (CY) ← Carry
	A n	0	1	0	1	0	n_3	n_2	n_1	n_0	0	A	n	1	1	(A) ← (A) + n n = 0 to 15
	SC	0	0	0	0	0	0	1	1	1	0	0	7	1	1	(CY) ← 1
	RC	0	0	0	0	0	0	1	1	0	0	0	6	1	1	(CY) ← 0
	SZC	0	0	0	1	0	1	1	1	1	0	2	F	1	1	(CY) = 0 ?
	CMA	0	0	0	0	1	1	1	0	0	0	1	C	1	1	(A) ← ($\overline{A}$)
	RAR	0	0	0	0	1	1	1	0	1	0	1	D	1	1	
	LGOP	0	0	1	0	0	0	0	0	1	0	4	1	1	1	Logic operation instruction XOR, OR, AND

Note: p is 0 to 7 for M34282M1, p is 0 to 15 for M34282M2/E2.

Skip condition	Carry flag CY	Detailed description
Continuous description	–	Loads the value n in the immediate field to register A. When the LA instructions are continuously coded and executed, only the first LA instruction is executed and other LA instructions coded continuously are skipped.
–	–	Transfers bits 7 to 4 to register B and bits 3 to 0 to register A when URS flag is cleared to "0." These bits 7 to 0 are the ROM pattern in address (DR_2 DR_1 DR_0 A_3 A_2 A_1 A_0) specified by registers A and D in page p.
	0/1	Transfers bit 8 of ROM pattern is transferred to flag CY when URS flag is set to "1" (after the URSC instruction is executed). (One of stack is used when the TABP p instruction is executed.)
–	–	Adds the contents of M(DP) to register A. Stores the result in register A. The contents of carry flag CY remains unchanged.
–	0/1	Adds the contents of M(DP) and carry flag CY to register A. Stores the result in register A and carry flag CY.
Overflow = 0	–	Adds the value n in the immediate field to register A. The contents of carry flag CY remains unchanged. Skips the next instruction when there is no overflow as the result of operation.
–	1	Sets (1) to carry flag CY.
–	0	Clears (0) to carry flag CY.
(CY) = 0	–	Skips the next instruction when the contents of carry flag CY is "0."
–	–	Stores the one's complement for register A's contents in register A.
–	0/1	Rotates 1 bit of the contents of register A including the contents of carry flag CY to the right.
–	–	Executes the logic operation selected by logic operation selection register LO between the contents of register A and register E, and stores the result in register A.

MACHINE INSTRUCTIONS (CONTINUED)

Parameter / Type of instructions	Mnemonic	Instruction code D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal notation	Number of words	Number of cycles	Function
Bit operation	SB j	0	0	1	0	1	1	1	j_1	j_0	0 5 C+j	1	1	(Mj(DP)) ← 1 j = 0 to 3
	RB j	0	0	1	0	0	1	1	j_1	j_0	0 4 C+j	1	1	(Mj(DP)) ← 0 j = 0 to 3
	SZB j	0	0	0	1	0	0	0	j_1	j_0	0 2 j	1	1	(Mj(DP)) = 0 ? j = 0 to 3
Comparison operation	SEAM	0	0	0	1	0	0	1	1	0	0 2 6	1	1	(A) = (M(DP)) ?
	SEA n	0	0	0	1	0	0	1	0	1	0 2 5	2	2	(A) = n ? n = 0 to 15
		0	1	0	1	1	n_3	n_2	n_1	n_0	0 B n			
Branch operation	B a	1	1	a_6	a_5	a_4	a_3	a_2	a_1	a_0	1 8+a a	1	1	(PC_L) ← a_6–a_0
	BL p, a	0	0	0	1	1	p_3	p_2	p_1	p_0	0 3 p	2	2	(PC_H) ← p (PC_L) ← a_6–a_0 (Note)
		1	1	a_6	a_5	a_4	a_3	a_2	a_1	a_0	1 8+a a			
	BA a	0	0	0	0	0	0	0	0	1	0 0 1	2	2	(PC_L) ← (a_6–a_4, A_3–A_0)
		1	1	a_6	a_5	a_4	a_3	a_2	a_1	a_0	1 8+a a			
	BLA p, a	0	0	0	0	1	0	0	0	0	0 1 0	2	2	(PC_H) ← p (PC_L) ← (a_6–a_4, A_3–A_0)
		1	1	a_6	a_5	a_4	p_3	p_2	p_1	p_0	1 8+a p			(Note)

Note: p is 0 to 7 for M34282M1, p is 0 to 15 for M34282M2/E2.

Skip condition	Carry flag CY	Detailed description
–	–	Sets (1) the contents of bit j (bit specified by the value j in the immediate field) of M(DP).
–	–	Clears (0) the contents of bit j (bit specified by the value j in the immediate field) of M(DP).
(Mj(DP)) = 0 j = 0 to 3	–	Skips the next instruction when the contents of bit j (bit specified by the value j in the immediate field) of M(DP) is "0."
(A) = (M(DP))	–	Skips the next instruction when the contents of register A is equal to the contents of M(DP).
(A) = n n = 0 to 15	–	Skips the next instruction when the contents of register A is equal to the value n in the immediate field.
–	–	Branch within a page : Branches to address a in the identical page.
–	–	Branch out of a page : Branches to address a in page p.
–	–	Branch within a page : Branches to address (a_6 a_5 a_4 A_3 A_2 A_1 A_0) determined by replacing the low-order 4 bits of the address a in the identical page with register A.
–	–	Branch out of a page : Branches to address (a_6 a_5 a_4 A_3 A_2 A_1 A_0) determined by replacing the low-order 4 bits of the address a in page p with register A.

MACHINE INSTRUCTIONS (CONTINUED)

Type of instructions	Mnemonic	Instruction code D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal notation	Number of words	Number of cycles	Function
Subroutine operation	BM a	1	0	a_6	a_5	a_4	a_3	a_2	a_1	a_0	1 a a	1	1	(SK(SP)) ← (PC) (SP) ← (SP) + 1 (PC_H) ← 2 (PC_L) ← a_6–a_0
	BML p, a	0	0	1	1	1	p_3	p_2	p_1	p_0	0 7 p	2	2	(SK(SP)) ← (PC) (SP) ← (SP) + 1 (PC_H) ← p
		1	0	a_6	a_5	a_4	a_3	a_2	a_1	a_0	1 a a			(PC_L) ← a_6–a_0 (Note)
	BMLA p, a	0	0	1	0	1	0	0	0	0	0 5 0	2	2	(SK(SP)) ← (PC) (SP) ← (SP) + 1
		1	0	a_6	a_5	a_4	p_3	p_2	p_1	p_0	1 a p			(PC_H) ← p (PC_L) ← (a_6–a_4, A_3–A_0) (Note)
Return operation	RT	0	0	1	0	0	0	1	0	0	0 4 4	1	2	(SP) ← (SP) – 1 (PC) ← (SK(SP))
	RTS	0	0	1	0	0	0	1	0	1	0 4 5	1	2	(SP) ← (SP) – 1 (PC) ← (SK(SP))
Timer operation	T1AB	0	0	1	0	0	0	1	1	1	0 4 7	1	1	at timer 1 stop ($V1_0$=0) ($R1_7$–$R1_4$) ← (B), ($R1_3$–$R1_0$) ← (A) ($T1_7$–$T1_4$) ← (B), ($T1_3$–$T1_0$) ← (A) at timer 1 operating ($V1_0$=1) ($R1_7$–$R1_4$) ← (B), ($R1_3$–$R1_0$) ← (A)
	TAB1	0	0	1	0	1	0	1	1	1	0 5 7	1	1	(B) ← ($T1_7$–$T1_4$) (A) ← ($T1_3$–$T1_0$)
	TV1A	0	0	1	0	1	1	0	1	1	0 5 B	1	1	($V1_2$–$V1_0$) ← (A_2–A_0)
	SNZT1	0	0	1	0	0	0	0	1	0	0 4 2	1	1	(T1F) = 1 ? After skipping the next instruction (T1F) ← 0
	T2AB	0	1	0	0	0	1	0	0	0	0 8 8	1	1	($R2L_7$–$R2L_4$) ← (B) ($R2L_3$–$R2L_0$) ← (A) ($T2_7$–$T2_4$) ← (B), ($T2_3$–$T2_0$) ← (A)

Note : p is 0 to 7 for M34282M1, and p is 0 to 15 for M34282M2/E2.

Skip condition	Carry flag CY	Detailed description
–	–	Call the subroutine in page 2 : Calls the subroutine at address a in page 2.
–	–	Call the subroutine : Calls the subroutine at address a in page p.
–	–	Call the subroutine : Calls the subroutine at address (a_6 a_5 a_4 A_3 A_2 A_1 A_0) determined by replacing the low-order 4 bits of address a in page p with register A.
–	–	Returns from subroutine to the routine called the subroutine.
Skip at uncondition	–	Returns from subroutine to the routine called the subroutine, and skips the next instruction at uncondition.
–	–	At timer 1 stop ($V1_0$ = 0), transfers the contents of register A and register B to timer 1 and reload register R1. At timer 1 operating ($V1_0$ = 1), transfers the contents of register A and register B to reload register R1.
–	–	Transfers the contents of timer 1 to registers A and B.
–	–	Transfers the contents of register A to registers V1.
(T1F) = 1	–	Skips the next instruction when the contents of T1F flag is “1.” After skipping, clears (0) to T1F flag.
–	–	Transfers the contents of register A and register B to timer 2 and reload register R2L.

MACHINE INSTRUCTIONS (CONTINUED)

Type of instructions	Mnemonic	Instruction code D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal notation	Number of words	Number of cycles	Function
Timer operation	TAB2	0	0	1	0	0	0	0	0	0	040	1	1	(B) ← ($T2_7$–$T2_4$), (A) ← ($T2_3$–$T2_0$)
	TV2A	0	0	1	0	1	1	0	1	0	05A	1	1	($V2_3$–$V2_0$) ← (A_3–A_0)
	SNZT2	0	0	1	0	1	0	0	1	0	052	1	1	(T2F) = 1 ? After skipping the next instruction (T2F) ← 0
	T2HAB	0	1	0	0	0	1	0	0	1	089	1	1	($R2H_7$–$R2H_4$) ← (B) ($R2H_3$–$R2H_0$) ← (A)
	T2R2L	0	0	1	0	1	0	0	1	1	053	1	1	($T2_7$–$T2_4$) ← ($R2L_7$–$R2L_4$) ($T2_3$–$T2_0$) ← ($R2L_3$–$R2L_0$)
Carrier wave control operation	SCAR	0	1	0	0	0	0	1	1	1	087	1	1	(CAR) ← 1
	RCAR	0	1	0	0	0	0	1	1	0	086	1	1	(CAR) ← 0
Input/Output operation	CLD	0	0	0	0	1	0	0	0	1	011	1	1	(D) ← 0
	RD	0	0	0	0	1	0	1	0	0	014	1	1	(D(Y)) ← 0 (Y) = 0 to 7
	SD	0	0	0	0	1	0	1	0	1	015	1	1	(D(Y)) ← 1 (Y) = 0 to 7
	SZD	0	0	0	1	0	0	1	0	0	024	2	2	(D(Y)) = 0 ? (Y) = 4 to 7
		0	0	0	1	0	1	0	1	1	02B			
	OEA	0	1	0	0	0	0	1	0	0	084	1	1	(E_1, E_0) ← (A_1, A_0)
	IAE	0	0	1	0	1	0	1	1	0	056	1	1	(A_2–A_0) ← (E_2–E_0)
	OGA	0	1	0	0	0	0	0	0	0	080	1	1	(G) ← (A)
	IAG	0	0	0	1	0	1	0	0	0	028	1	1	(A) ← (G)

Skip condition	Carry flag CY	Detailed description
–	–	Transfers the contents of timer 2 to registers A and B.
–	–	Transfers the contents of register A to registers V2.
(T2F) = 1	–	Skips the next instruction when the contents of T2F flag is “1.” After skipping, clears (0) to T2F flag.
–	–	Transfers the contents of register A and register B to reload register R2H.
–	–	Transfers the contents of reload register R2L to timer 2.
–	–	Sets (1) to port CARR output flag (CAR).
–	–	Clears (0) to port CARR output flag (CAR).
–	–	Clears (0) to port D (high-impedance state).
–	–	Clears (0) to a bit of port D specified by register Y (high-impedance state).
–	–	Sets (1) to a bit of port D specified by register Y.
(D(Y)) = 0 (Y) = 4 to 7	–	Skips the next instruction when a bit of port D specified by register Y is “0.”
–	–	Outputs the contents of register A to port E.
–	–	Transfers the contents of port E to register A.
–	–	Outputs the contents of register A to port G.
–	–	Transfers the contents of port G to register A.

Parameter / Type of instructions	Mnemonic	Instruction code										Number of words	Number of cycles	Function
		D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Hexadecimal notation			
Other operation	NOP	0	0	0	0	0	0	0	0	0	0 0 0	1	1	(PC) ← (PC) + 1
	POF	0	0	0	0	0	1	1	0	1	0 0 D	1	1	RAM back-up
	SNZP	0	0	0	0	0	0	0	1	1	0 0 3	1	1	(P) = 1 ?
	CCK	0	0	1	0	1	1	0	0	1	0 5 9	1	1	STCK changes to f(X_{IN})
	TLOA	0	0	1	0	1	1	0	0	0	0 5 8	1	1	(LO_1, LO_0) ← (A_1, A_0)
	URSC	0	1	0	0	0	0	0	1	0	0 8 2	1	1	(URS) ← 1
	TPU0A	0	1	0	0	0	1	1	1	1	0 8 F	1	1	($PU0_3$–$PU0_0$) ← (A_3–A_0)
	TPU1A	0	1	0	0	0	1	1	1	0	0 8 E	1	1	($PU1_3$–$PU1_0$) ← (A_3–A_0)
	WRST	0	0	0	0	0	1	1	1	1	0 0 F	1	1	(WDF1) ← 0

Skip condition	Carry flag CY	Detailed description
–	–	No operation
–	–	Puts the system in RAM back-up state.
(P) = 1	–	Skips the next instruction when P flag is “1.” After skipping, P flag remains unchanged.
–	–	System clock (STCK) changes to f(X_{IN}) from f(X_{IN})/8. Execute this CCK instruction at address 0 in page 0.
–	–	Transfers the contents of register A to the logic operation selection register LO.
–	–	Sets the most significant ROM code reference enable flag (URS) to “1.”
–	–	Transfers the contents of register A to register PU0.
–	–	Transfers the contents of register A to register PU1.
–	–	Initializes the watchdog timer flag (WDF1).

INSTRUCTION CODE TABLE

D8–D4 / D3–D0 Hex. notation	00000	00001	00010	00011	00100	00101	00110	00111	01000	01001	01010	01011	01100	01101	01110	01111	10000 10111	11000 11111
0000 0	NOP	BLA	SZB 0	BL	TAB2	BMLA	XAM 0	BML	OGA	TABP 0	A 0	LA 0	LXY 0,0	LXY 1,0	LXY 2,0	LXY 3,0	BM	B
0001 1	BA	CLD	SZB 1	BL	LGOP	—	XAM 1	BML	—	TABP 1	A 1	LA 1	LXY 0,1	LXY 1,1	LXY 2,1	LXY 3,1	BM	B
0010 2	—	—	SZB 2	BL	SNZT1	SNZT2	XAM 2	BML	URSC	TABP 2	A 2	LA 2	LXY 0,2	LXY 1,2	LXY 2,2	LXY 3,2	BM	B
0011 3	SNZP	INY	SZB 3	BL	—	T2R2L	XAM 3	BML	—	TABP 3	A 3	LA 3	LXY 0,3	LXY 1,3	LXY 2,3	LXY 3,3	BM	B
0100 4	—	RD	SZD	BL	RT	—	TAM 0	BML	OEA	TABP 4	A 4	LA 4	LXY 0,4	LXY 1,4	LXY 2,4	LXY 3,4	BM	B
0101 5	—	SD	SEAn	BL	RTS	—	TAM 1	BML	—	TABP 5	A 5	LA 5	LXY 0,5	LXY 1,5	LXY 2,5	LXY 3,5	BM	B
0110 6	RC	—	SEAM	BL	—	IAE	TAM 2	BML	RCAR	TABP 6	A 6	LA 6	LXY 0,6	LXY 1,6	LXY 2,6	LXY 3,6	BM	B
0111 7	SC	DEY	—	BL	T1AB	TAB1	TAM 3	BML	SCAR	TABP 7	A 7	LA 7	LXY 0,7	LXY 1,7	LXY 2,7	LXY 3,7	BM	B
1000 8	—	—	IAG	BL*	—	TLOA	XAMI 0	BML*	T2AB	TABP 8*	A 8	LA 8	LXY 0,8	LXY 1,8	LXY 2,8	LXY 3,8	BM	B
1001 9	—	—	TDA	BL*	—	CCK	XAMI 1	BML*	T2HAB	TABP 9*	A 9	LA 9	LXY 0,9	LXY 1,9	LXY 2,9	LXY 3,9	BM	B
1010 A	AM	TEAB	TABE	BL*	—	TV2A	XAMI 2	BML*	—	TABP 10*	A 10	LA 10	LXY 0,10	LXY 1,10	LXY 2,10	LXY 3,10	BM	B
1011 B	AMC	—	—	BL*	—	TV1A	XAMI 3	BML*	—	TABP 11*	A 11	LA 11	LXY 011	LXY 1,11	LXY 2,11	LXY 3,11	BM	B
1100 C	TYA	CMA	—	BL*	RB 0	SB 0	XAMD 0	BML*	—	TABP 12*	A 12	LA 12	LXY 0,12	LXY 1,12	LXY 2,12	LXY 3,12	BM	B
1101 D	POF	RAR	—	BL*	RB 1	SB 1	XAMD 1	BML*	—	TABP 13*	A 13	LA 13	LXY 0,13	LXY 1,13	LXY 2,13	LXY 3,13	BM	B
1110 E	TBA	TAB	—	BL*	RB 2	SB 2	XAMD 2	BML*	TPU1A	TABP 14*	A 14	LA 14	LXY 0,14	LXY 1,14	LXY 2,14	LXY 3,14	BM	B
1111 F	WRST	TAY	SZC	BL*	RB 3	SB 3	XAMD 3	BML*	TPU0A	TABP 15*	A 15	LA 15	LXY 0,15	LXY 1,15	LXY 2,15	LXY 3,15	BM	B

The above table shows the relationship between machine language codes and machine language instructions. D3–D0 show the low-order 4 bits of the machine language code, and D8–D4 show the high-order 5 bits of the machine language code. The hexadecimal representation of the code is also provided. There are one-word instructions and two-word instructions, but only the first word of each instruction is shown. Do not use the code marked "–."

The codes for the second word of a two-word instruction are described below.

	The second word
BL	1 1aaa aaaa
BML	1 0aaa aaaa
BA	1 1aaa aaaa
BLA	1 1aaa pppp
BMLA	1 0aaa pppp
SEA	0 1011 nnnn
SZD	0 0010 1011

* cannot be used in the M34282M1.

REGISTER STRUCTURE

Timer control register V1		at reset : 000_2	at RAM back-up : 000_2	W
$V1_2$	Carrier wave output auto-control bit	0	Auto-control output by timer 1 is invalid	
		1	Auto-control output by timer 1 is valid	
$V1_1$	Timer 1 count source selection bit	0	Carrier wave output (CARRY)	
		1	Bit 5 of watchdog timer (WDT)	
$V1_0$	Timer 1 control bit	0	Stop (Timer 1 state retained)	
		1	Operating	

Timer control register V1		at reset : 0000_2	at RAM back-up : 0000_2	W
$V1_3$	Carrier wave "H" interval expansion bit	0	To expand "H" interval is invalid	
		1	To expand "H" interval is valid (when $V2_2$=1 selected)	
$V1_2$	Carrier wave generation function control bit	0	Carrier wave generation function invalid	
		1	Carrier wave generation function valid	
$V1_1$	Timer 2 count source selection bit	0	$f(X_{IN})$	
		1	$f(X_{IN})/2$	
$V1_0$	Timer 2 control bit	0	Stop (Timer 2 state retained)	
		1	Operating	

Logic operation selection register LO		at reset : 00_2	at RAM back-up : 00_2	W
LO_1	Logic operation selection bits	LO_1	LO_0	Logic operation function
		0	0	Exclusive logic OR operation (XOR)
LO_0		0	1	OR operation (OR)
		1	0	AND operation (AND)
		1	1	Not available

Pull-down control register PU0		at reset : 0000_2	at RAM back-up : state retained	W
$PU0_3$	Ports G_2, G_3 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid	
		1	Pull-down transistor ON, key-on wakeup valid	
$PU0_2$	Ports G_0, G_1 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid	
		1	Pull-down transistor ON, key-on wakeup valid	
$PU0_1$	Port E_1 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid	
		1	Pull-down transistor ON, key-on wakeup valid	
$PU0_0$	Port E_0 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid	
		1	Pull-down transistor ON, key-on wakeup valid	

Pull-down control register PU1		at reset : 0000_2	at RAM back-up : state retained	W
$PU1_3$	Port D_7 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid	
		1	Pull-down transistor ON, key-on wakeup valid	
$PU1_2$	Port D_6 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid	
		1	Pull-down transistor ON, key-on wakeup valid	
$PU1_1$	Port D_5 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid	
		1	Pull-down transistor ON, key-on wakeup valid	
$PU1_0$	Port D_4 pull-down transistor control bit	0	Pull-down transistor OFF, key-on wakeup invalid	
		1	Pull-down transistor ON, key-on wakeup valid	

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Ratings	Unit
V_{DD}	Supply voltage		–0.3 to 5	V
V_I	Input voltage		–0.3 to V_{DD}+0.3	V
V_O	Output voltage		–0.3 to V_{DD}+0.3	V
P_d	Power dissipation	Ta = 25 °C	300	mW
T_{opr}	Operating temperature range		–20 to 85	°C
T_{stg}	Storage temperature range		–40 to 125	°C

RECOMMENDED OPERATING CONDITIONS

(Ta = –20 °C to 85 °C, V_{DD} = 1.8 V to 3.6 V, unless otherwise noted)

Symbol	Parameter		Conditions	Limits			Unit
				Min.	Typ.	Max.	
V_{DD}	Supply voltage			1.8		3.6	V
V_{RAM}	RAM back-up voltage (at RAM back-up mode)			1.1		3.6	V
V_{SS}	Supply voltage				0		V
V_{IH}	"H" level input voltage Ports D_4–D_7, E, G		V_{DD} = 3.0 V	0.7V_{DD}		V_{DD}	V
V_{IH}	"H" level input voltage X_{IN}		V_{DD} = 3.0 V	0.8V_{DD}		V_{DD}	V
V_{IL}	"L" level input voltage Ports D_4–D_7, E, G		V_{DD} = 3.0 V	0		0.2V_{DD}	V
V_{IL}	"L" level input voltage X_{IN}		V_{DD} = 3.0 V	0		0.2V_{DD}	V
I_{OH}(peak)	"H" level peak output current Ports D, E_1, G		V_{DD} = 3.0 V			–4	mA
I_{OH}(peak)	"H" level peak output current Port E_0		V_{DD} = 3.0 V			–24	mA
I_{OH}(peak)	"H" level peak output current CARR		V_{DD} = 3.0 V			–20	mA
I_{OL}(peak)	"L" level peak output current CARR		V_{DD} = 3.0 V			4	mA
I_{OH}(avg)	"H" level average output current Ports D, E_1, G		V_{DD} = 3.0 V			–2	mA
I_{OH}(avg)	"H" level average output current Port E_0		V_{DD} = 3.0 V			–12	mA
I_{OH}(avg)	"H" level average output current CARR		V_{DD} = 3.0 V			–10	mA
I_{OL}(avg)	"L" level average output current CARR		V_{DD} = 3.0 V			2	mA
f(X_{IN})	System clock frequency	when STCK = f(X_{IN})/8 selected	Ceramic resonance			4	MHz
		when STCK = f(X_{IN}) selected	Ceramic resonance			500	kHz
V_{DET}	Voltage drop detection circuit detection voltage			1.10		1.80	V
			Ta=25 °C	1.40	1.50	1.56	
T_{DET}	Voltage drop detection circuit low voltage determination time		When supply voltage passes the detected voltage at ±50V/s.		0.2	1.2	ms
T_{PON}	Power-on reset circuit valid power source rising time		V_{DD} = 0 to 2.2 V			1	ms

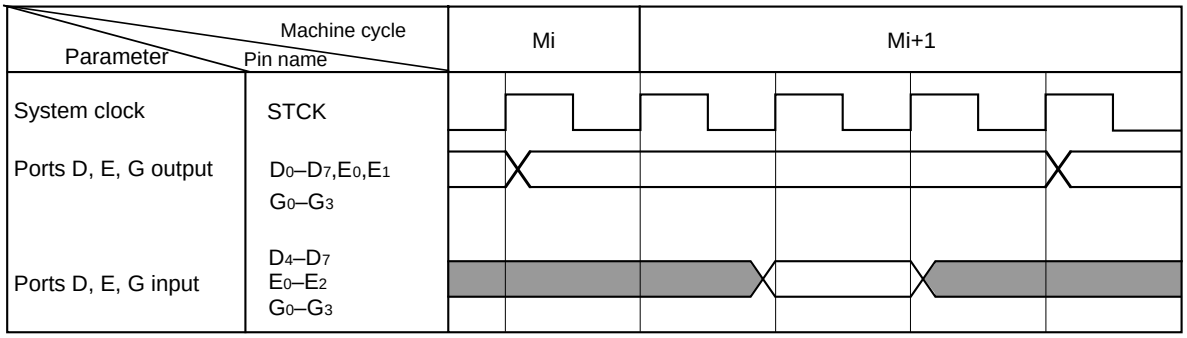
Note: The average output current ratings are the average current value during 100 ms.

ELECTRICAL CHARACTERISTICS

(Ta = –20 °C to 85 °C, V_{DD} = 3 V, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OL}	"L" level output voltage Port CARR	I_{OL} = 2 mA			0.9	V
V_{OL}	"L" level output voltage X_{OUT}	I_{OL} = 0.2 mA			0.9	V
V_{OH}	"H" level output voltage Ports D, E_1, G	I_{OH} = –2 mA	2.1			V
V_{OH}	"H" level output voltage Port E_0	I_{OH} = –12 mA	1.5			V
V_{OH}	"H" level output voltage CARR	I_{OH} = –10 mA	1.0			V
V_{OH}	"H" level output voltage X_{OUT}	I_{OH} = –0.2 mA	2.1			V
I_{IL}	"L" level input current Ports D_4–D_7, E, G	V_I = V_{SS}			–1	μA
I_{IH}	"H" level input current Ports E_0, E_1	V_I = V_{DD} Pull-down transistor in off-state			1	μA
I_{OZ}	Output current at off-state Ports D, E_0, E_1, G	V_O = V_{SS}			–1	μA
I_{DD}	Supply current (when operating)	f(X_{IN}) = 4.0 MHz		400	800	μA
		f(X_{IN}) = 500 kHz		250	500	μA
	Supply current (at RAM back-up)			1	3	μA
		Ta = 25 °C		0.1	0.5	μA
R_{PH}	Pull-down resistor value Ports D_4–D_7, E, G	V_{DD} = 3 V, V_I = 3 V	75	150	300	kΩ
R_{OSC}	Feedback resistor value between X_{IN}–X_{OUT}		700		3200	kΩ

BASIC TIMING DIAGRAM



BUILT-IN PROM VERSION

In addition to the mask ROM versions, the 4282 Group has the One Time PROM versions whose PROMs can only be written to and not be erased.

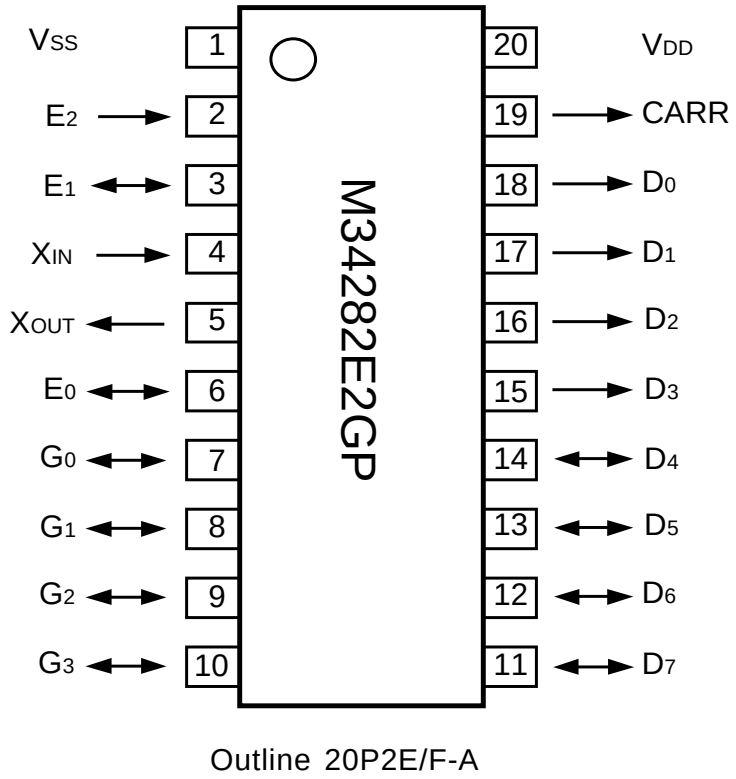
The built-in PROM version has functions similar to those of the mask ROM versions, but it has PROM mode that enables writing to built-in PROM.

Table 10 shows the product of built-in PROM version. Figure 27 and 28 show the pin configurations of built-in PROM versions. The One Time PROM version has pin-compatibility with the mask ROM version.

Table 10 Product of built-in PROM version

Product	PROM size (× 9 bits)	RAM size (× 4 bits)	Package	ROM type
M34282E2GP	2048 words	64 words	20P2E/F-A	One Time PROM [shipped in blank]

PIN CONFIGURATION (TOP VIEW)



Outline 20P2E/F-A

Fig. 27 Pin configuration of built-in PROM version

(1) PROM mode (serial input/output)

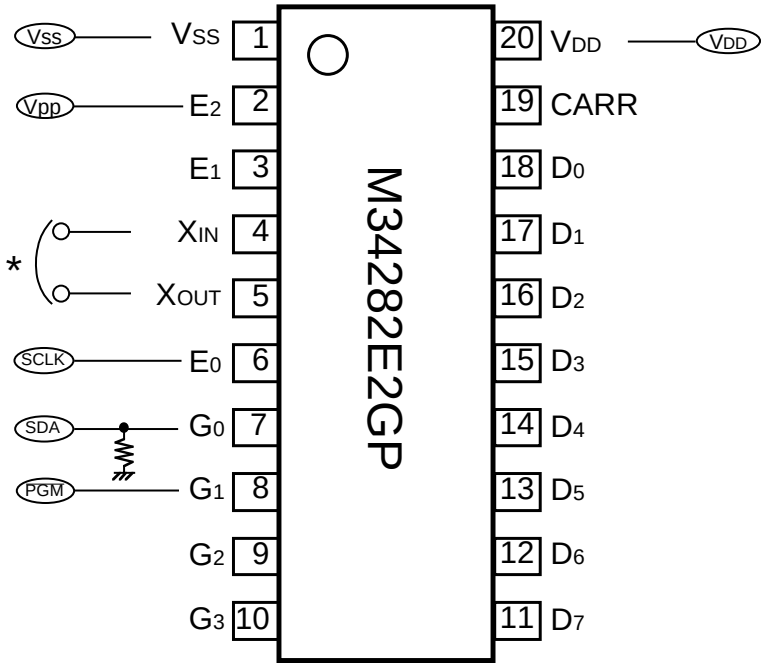
The M34282E2GP has a PROM mode in addition to a normal operation mode. It has a function to serially input/output the command codes, addresses, and data required for operation (e.g., read and program) on the built-in PROM using only a few pins. This mode can be selected by setting pins SDA (serial data input/output), SCLK (serial clock input), $\overline{PGM}$ and V_{PP} to "H" after connecting wires as shown in Figure 28 and powering on the V_{DD} pin, and then applying 12.5V to the V_{PP} pin.

In the PROM mode, three types of software commands (read, program, and program verify) can be used. Clock-synchronous serial I/O is used, beginning from the LSB (LSB first).

Refer to the "Mitsubishi Microcomputer Development Support Tools" Hompage (http://www.tool-spt.maec.co.jp/index_e.htm).

about the serial programmer for the Mitsubishi single-chip microcomputers.

PIN CONFIGURATION (TOP VIEW)



Outline 20P2E/F-A

* : connected to the ceramic resonance circuit.

Note: The state of disconnected pins are the same as that at reset.

Fig. 28 Pin configuration of built-in PROM version (continued)

(2) Functional outline

In the PROM mode, data is transferred with the clock-synchronous serial input/output. The input data is read through the SDA pin into the internal circuit synchronously with the rising edge of the serial clock pulse. The output data is output from the SDA pin synchronously with the falling edge of the serial clock pulse. Data is transferred in units of 8 bits.

In the first transfer, the command code is input. Then, address input or data input/output is performed according to the contents of the command code. Table 11 shows the software command used in the PROM mode. The following explains each software command.

Table 11 Software command

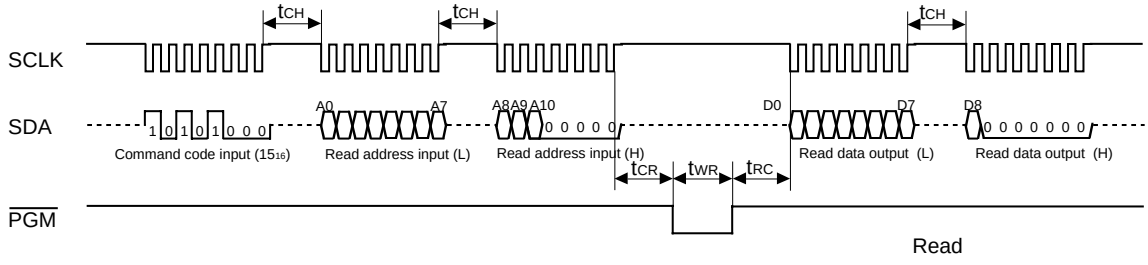
Number of transfer / Command	First command code input	Second	Third	Fourth
Read	15_{16}	Read address L (input)	Read address H (input)	Read data L (output)
Program	25_{16}	Program address L (input)	Program address H (input)	Program data L (input)
Program verify	35_{16}	Program address L (input)	Program address H (input)	Program data L (input)

Number of transfer / Command	Fifth	Sixth	Seventh
Read	Read data H (output)	————	————
Program	Program data H (input)	————	————
Program verify	Program data H (input)	Verify data L (output)	Verify data H (output)

(3) Read

Input the command code 15_{16} in the first transfer. Proceed and input the low-order 8 bits and the high-order 8 bits of the address and pull the $\overline{PGM}$ pin to "L." When this is done, the contents of input address is read and stored into the internal data latch.

When the $\overline{PGM}$ pin is released back to "H" and serial clock is input to the SCLK pin, the low-order 8 bits and high-order 8 bits of read data which have been stored into the data latch, are serially output from the SDA pin.



Note: When outputting the read data, the SDA pin is switched for output at the first falling of the serial clock. The SDA pin is placed in the high-impedance state during the $t_{h(C-E)}$ period after the last rising edge of the serial clock (at the 16th bit).

Fig. 29 Timing at reading

(4) Program

Input command code 25_{16} in the first transfer. Proceed and input the low-order 8 bits and high-order 8 bits of the address and the low-order 8 bits and high-order 8 bits of program data, and pull the $\overline{PGM}$ pin to "L." When this is done, the program data is programmed to the specified address.

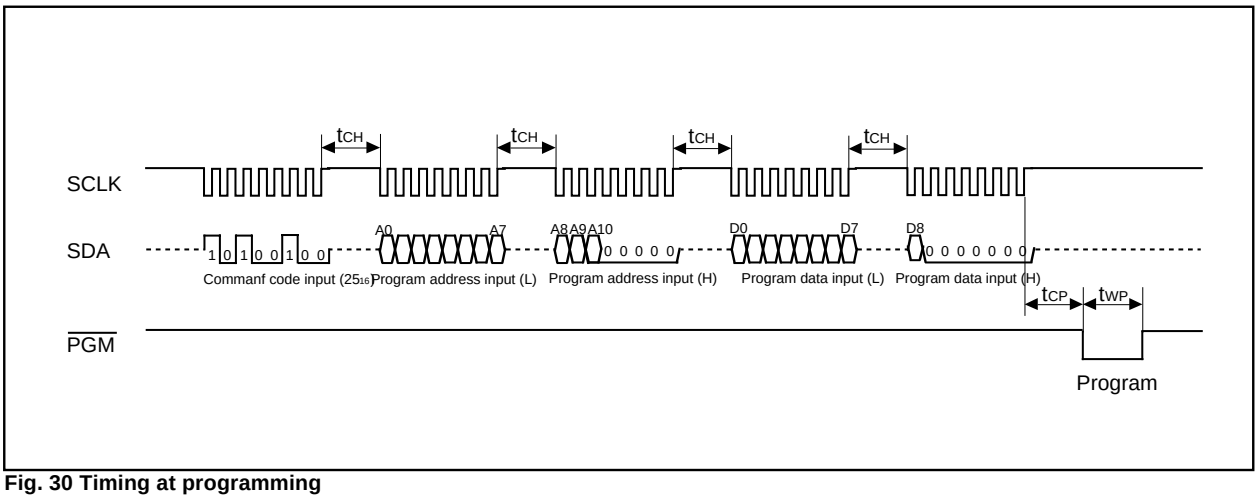
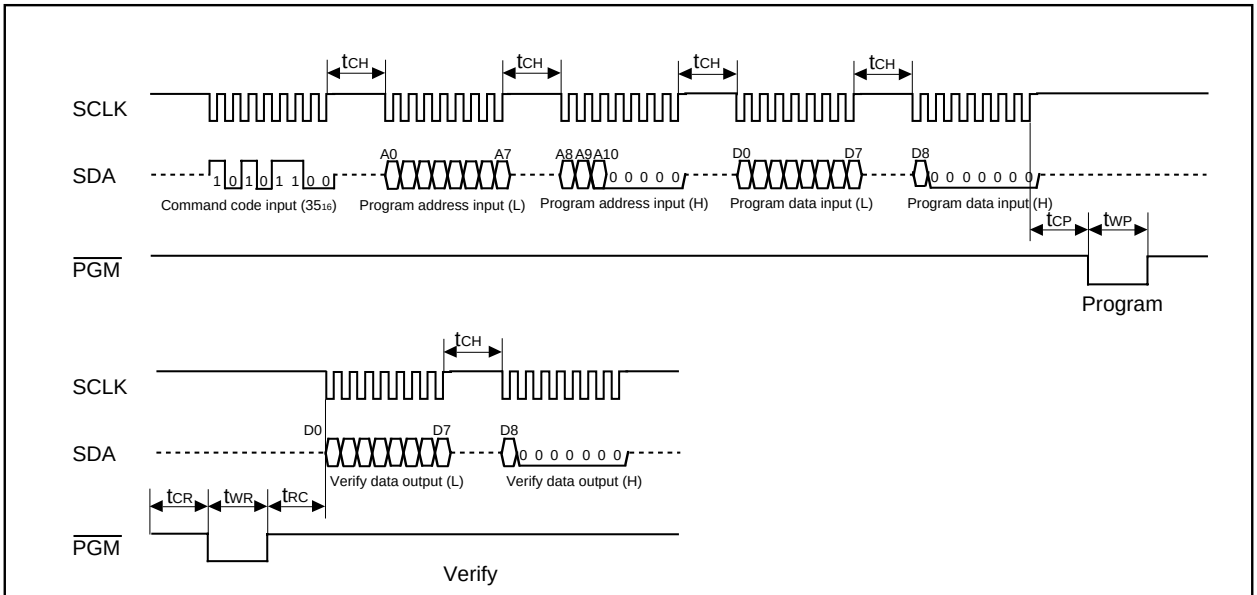


Fig. 30 Timing at programming

(5) Program verify

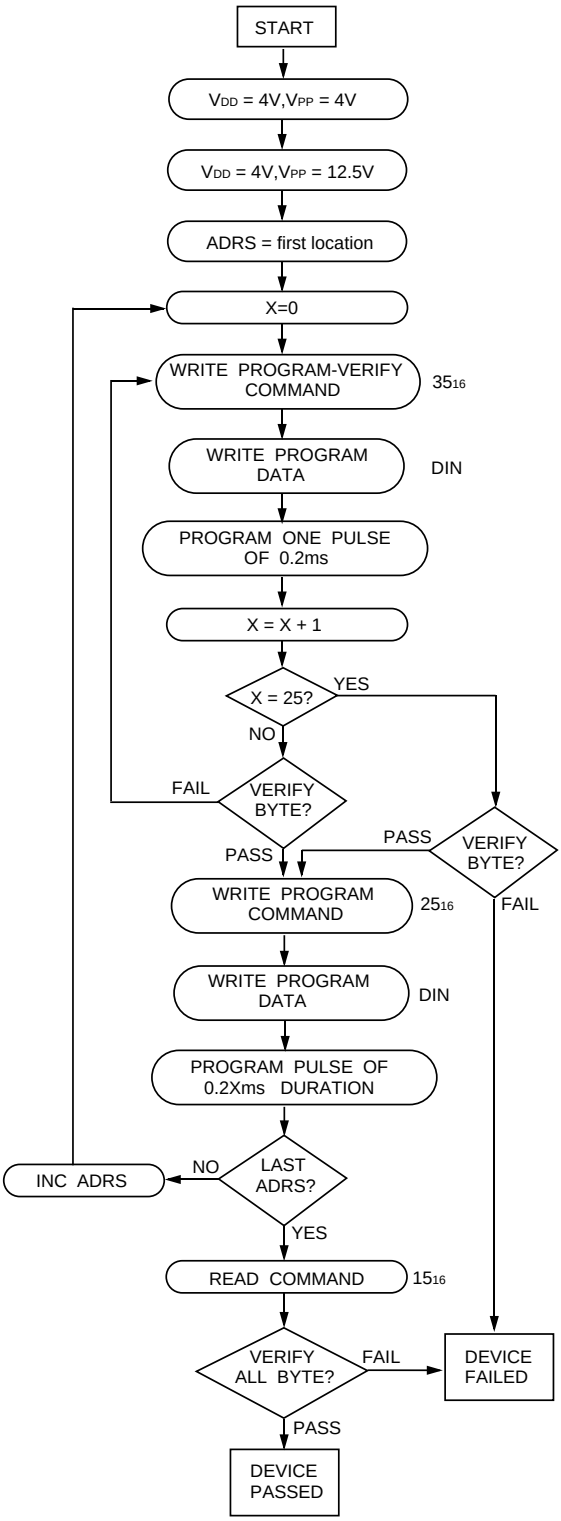
Input command code 35_{16} in the first transfer. Proceed and input the low-order 8 bits and high-order 8 bits of the address and the low-order 8 bits and high-order 8 bits of program data, and pull the $\overline{PGM}$ pin to "L." When this is done, the program data is programmed to the specified address. Then, when the $\overline{PGM}$ pin is pulled to "L" again after it is released back to "H," the address programmed with the program command is read and verified and stored into the internal data latch. When the $\overline{PGM}$ pin is released back to "H" and serial clock is input to the SCLK pin, the verify data that has been stored into the data latch is serially output from the SDA pin.



Note: When outputting the verify data, the SDA pin is switched for output at the first falling of the serial clock. The SDA pin is placed in the high-impedance state during the $t_{h(C-E)}$ period after the last rising edge of the serial clock (at the 16th bit).

Fig. 31 Timing at program verifying

PROGRAM ALGORITHM FLOW CHART

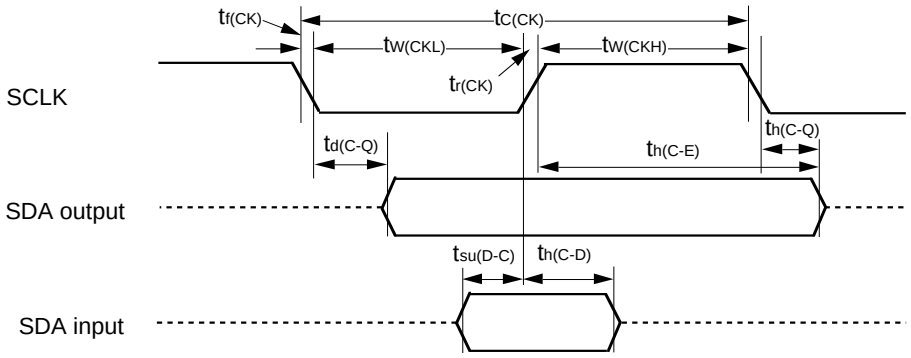


TIMING REQUIREMENT CONDITION AND SWITCHING CHARACTERISTICS

(Ta = 25 °C, V_{DD} = 4.0 V, V_{PP} = 12.5 V)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
t_{CH}	Serial transfer width time	2.0		μs
t_{CR}	Read wait time after transfer	2.0		μs
t_{WR}	Read pulse width	500		ns
t_{RC}	Transfer wait time after read	2.0		μs
t_{CP}	Program wait time after transfer	2.0		μs
t_{WP}	Program pulse width	0.19	0.21	ms
t_{OWP}	Added program pulse width	0.19	5.25	ms
$t_{C(CK)}$	SCLK input cycle time	1.0		μs
$t_{W(CKH)}$	SCLK “H” pulse width	450		ns
$t_{W(CKL)}$	SCLK “L” pulse width	450		ns
$t_{r(CK)}$	SCLK rising time	40		ns
$t_{f(CK)}$	SCLK falling time	40		ns
$t_{d(C-Q)}$	SDA output delay time	0	180	ns
$t_{h(C-Q)}$	SDA output hold time	0		ns
$t_{h(C-E)}$	SDA output hold time (only for 16th bit)	100		ns
$t_{su(D-C)}$	SDA input set-up time	60		ns
$t_{h(C-D)}$	SDA input hold time	180		ns

TIMING DIAGRAM



Measurement condition
Output timing voltage: V_{OL} = 0.8 V, V_{OH} = 2.0 V
Input timing voltage: V_{IL} = 0.2 V_{DD}, V_{IH} = 0.8 V_{DD}

(6) Notes on handling

① A high-voltage is used for writing. Take care that overvoltage is not applied. Take care especially at turning on the power.

② For the One Time PROM version, Mitsubishi Electric corp. does not perform PROM writing test and screening in the assembly process and following processes. In order to improve reliability after writing, performing writing and test according to the flow shown in Figure 32 before using is recommended.

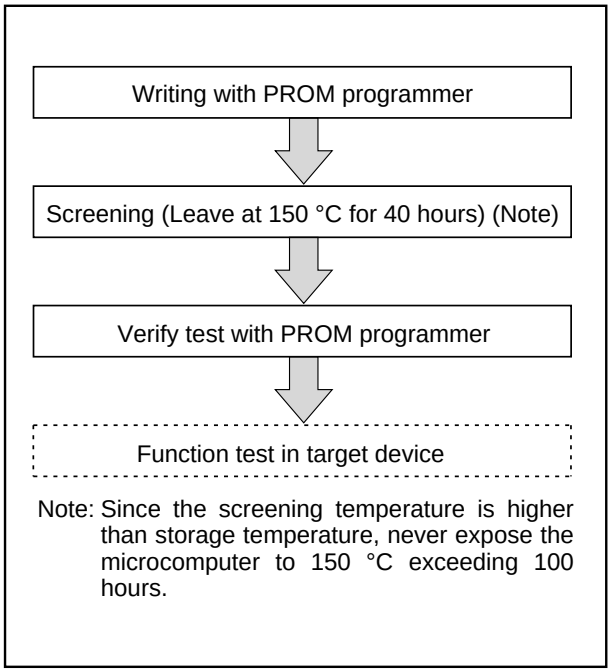


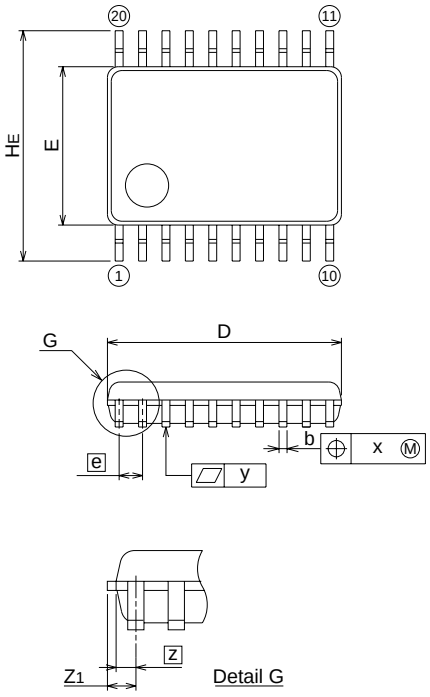
Fig. 32 Flow of writing and test of the product shipped in blank

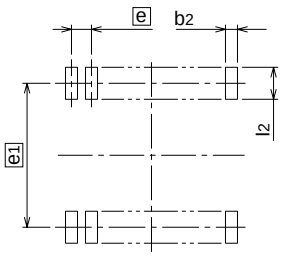
PACKAGE OUTLINE

20P2E/F-A

Plastic 20pin 225mil SSOP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
SSOP20-P-225-0.65	–	0.08	Alloy 42/Cu Alloy





Recommended Mount Pad

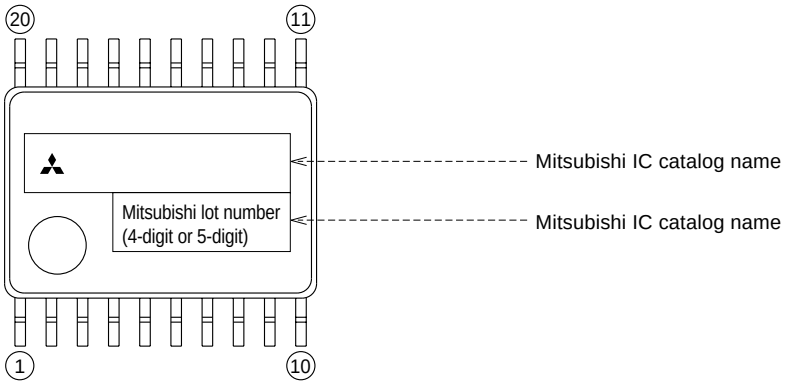
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	1.45
A_1	0	0.1	0.2
A_2	–	1.15	–
b	0.17	0.22	0.32
c	0.13	0.15	0.2
D	6.4	6.5	6.6
E	4.3	4.4	4.5
e	–	0.65	–
H_E	6.2	6.4	6.6
L	0.3	0.5	0.7
L_1	–	1.0	–
z	–	0.325	–
Z_1	–	–	0.475
x	–	–	0.13
y	–	–	0.1
θ	0°	–	10°
b_2	–	0.35	–
e1	–	5.8	–
l_2	1.0	–	–

20P2E/F-A (20-PIN SSOP) MARK SPECIFICATION FORM

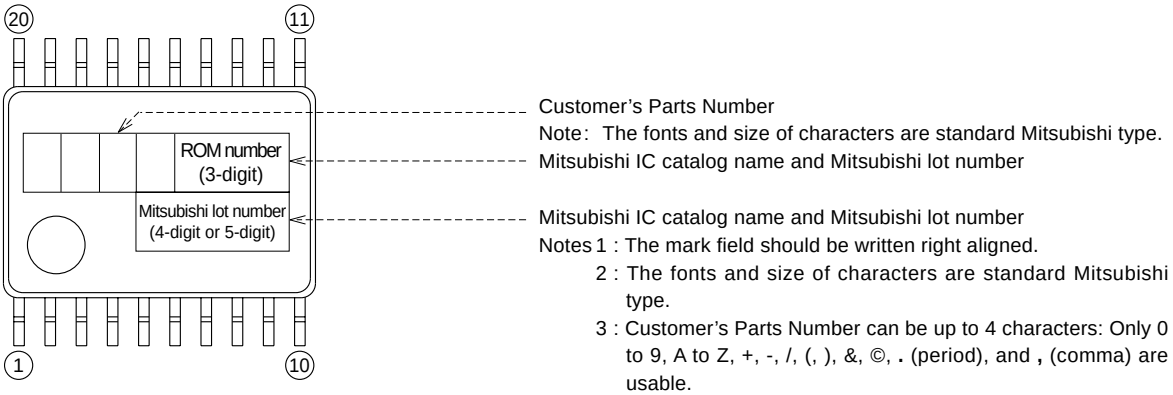
Mitsubishi IC catalog name

Please choose one of the marking types below (A, B), and enter the Mitsubishi IC catalog name and the special mark (if needed).

A. Standard Mitsubishi Mark



B. Customer's Parts Number + Mitsubishi IC Catalog Name



Notes 1 : The mark field should be written right aligned.
2 : The fonts and size of characters are standard Mitsubishi type.
3 : Customer's Parts Number can be up to 4 characters: Only 0 to 9, A to Z, +, -, /, (,), &, ©, . (period), and , (comma) are usable.

4282 Group

SINGLE-CHIP 4-BIT CMOS MICROCOMPUTER

REVISION DESCRIPTION LIST	4282 GROUP DATA SHEET

Rev. No.	Revision Description	Rev. date
1.0	First Edition	000619
1.1	Page 12 (2) Precautions revised. Page 13 (3) Timer 1, (4) Timer 2 revised. Page 22 ③ Timer revised.	000725
1.2	Pages 7, 8, 14, 18, 21: Character fonts errors revised.	000823
1.3	All pages: "PRELIMINARY Notice: This is not a final specification. Some parametric limits are subject to change." eliminated. Page 1: Product name table; "Under development" eliminated. Page 9: 48 words × 4 bits (128 bits) → 48 words × 4 bits (192 bits) Page 21: ROM ORDERING METHOD revised. Page 61: "Mitsubishi Microcomputer Development Support Tools" Hompage (http://www.tool-spt.mesc.co.jp/index_e.htm) → (http://www.tool-spt.maec.co.jp/index_e.htm)	010703