SINGLE-CHIP 4-BIT CMOS MICROCOMPUTER

DESCRIPTION

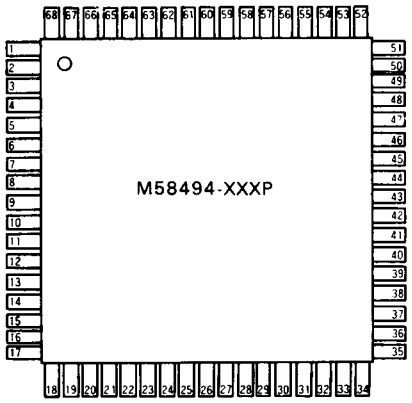
The M58494-XXXP is a single-chip 4-bit microcomputer fabricated using CMOS technology in a 68-pin plastic flat package. It has a 4096-word by 10-bit mask-programmable ROM and a 32-word by 4-bit RAM. RAM capacity can be expanded to as much as 4096 words by 4 bits by directly connecting generally available CMOS RAMs.

This device is designed for applications where the low power dissipation of CMOS is essential.

FEATURES

- Single 5V power supply
- Basic machine instructions: 92
- Basic instruction execution time (at 455 kHz clock frequency): 6.6μs
- Large memory capacity:
  - ROM: 4096-word x 10-bit
  - Internal RAM: 32-word x 4-bit
  - External RAM: 4096-word x 4-bit (max)
- Saving of last data pointer: 4-level
- Subroutine nesting: 12-level
- Internal timer:
  - Prescaled: 14-bit
  - Timer: 4-bit
- Internal event-counter: 4-bit
- I/O port for external RAMs (all three-state)
  - Address (port A): 12-bit
  - Control signals (R/W, 0D): 2-bit
  - Data I/O (port D): 4-bit
- General-purpose registers: 4 x 8-bit
- I/O port (port Q): 8-bit
- I/O port (port R): 2 x 4-bit

PIN CONFIGURATION (TOP VIEW)



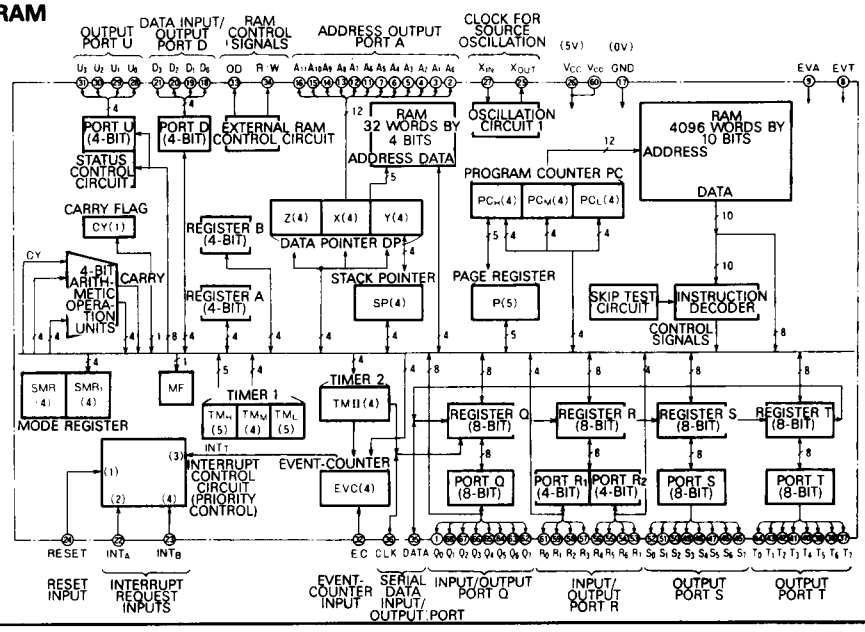
Outline 68P2

- I/O port (serial data port): 2-bit
- Output ports (port S, port T): 2 x 8-bit
- Output port (port U, three-state output): 4-bit
- Event-counter input (port EC): 1-bit
- Interrupt function (priority interrupt type): 4-factor, 1-level

APPLICATIONS

- Electronic cash registers, electronic calculators (with printer and/or programmable)
- Office machines, intelligent terminals, data terminals
- Sewing machines, knitting machines, etc.

BLOCK DIAGRAM



SINGLE-CHIP 4-BIT CMOS MICROCOMPUTER

FUNCTION

The M58494-XXXP consists of a mask ROM and RAM, a 4-bit arithmetic logical unit, a clock generator, input/output ports, interface for a multiprocessor system, timers, an event counter and interrupt circuit. RAM capacity can be expanded by connecting CMOS RAMs externally.

The ROM stores 32 pages by 128 words of program and its addressing is performed by a program counter. The program counter consists of a 7-bit binary sequential counter and a 5-bit page register. After the execution of the instruction at address 127 of each page is completed, the page designation counter is automatically incremented, and the 7-bit binary counter is reset to zero (goes to address 0 of the next page). The return addresses from a subroutine or interrupt are saved in the 12-bit by 12-level stack registers which are reserved in the fixed area of the external RAMs. When an interrupt occurs, the jump address is fixed as follows: in the case of a reset signal, the address is reset to page 0 address 0; for the INTA signal, to page 0 address 2. For the carry signal of either the timer or the event counter it is reset to page 0 address 8, and for the INT_B signal, to page 0 address 4.

The internal RAM is used to store data in the form of two files each consisting of 16 words by 4 bits. The external RAM can be expanded up to 4096 words by 4 bits. These addresses are designated by a 12-bit data pointer. The contents of the data pointer can be saved for up to 4 levels in the stack region (fixed region in the external RAMs) by execution of a special instruction. The external RAM can be easily expanded without any extra interface circuits by connecting a 12-bit address signal, the 2-bit RAM control signal and the 4-bit data input/output signal. These signals can address external RAMs for up to 4096 $\times$ 4-bit words, thus incrementing the basic external minimum RAM organization of 256 x 4-bit words.

The RAM addressing instructions, RAM-to-accumulator transfer instructions, arithmetic instructions, register-to-register transfer instructions, input and output instructions, input and output control instructions, and timer instructions are executed mainly with register A (accumulator).

RAM contains general registers of 32 bits for use by the arithmetic processing unit, which consists of the accumulator etc., and input/output ports. They are four 8-bit shift registers basically, and control the functional combinations of the serial input, the parallel input, the serial output and the parallel output by means of instructions. They execute the data transfer between output or input/output ports, loading the 8-bit value of the DATA field in the ROM, sending out internal serial data and receiving external serial data.

The input/output port Q consists of 8 bits. It has an 8-bit output latch and is connected to the 8-bit general-purpose register Q. Register Q is connected in parallel with registers A and B, and also with port Q for parallel data transfer, and is connected in serial with the external serial input for the serial data transfer. It can load the 8-bit data of the data field in the ROM. Thus, register Q stores the data transferred from registers A and B, the internal or external serial data, and data from the 8-bit data field in the ROM. The 8-bit input data to port Q can also be transferred to register A or B. The 8-bit data can be transferred between port Q and register Q at the same time by the input/output instructions.

The input/output port R consists of 8 bits, has an 8-bit output latch and is connected to the 8-bit general-purpose register. Register R has the same function as the previously described register Q except that the serial data is read from the least significant bit of register Q. It stores the data transferred from registers A and B, serial data, and the 8-bit value of the immediate field in the ROM. An 8-bit signal applied to port R can be transferred to register B 4 bits at a time. The 8-bit data can also be transferred between port R and register R at the same time by input/output instructions.

The output port S consists of 8 bits, has an 8-bit output latch and is connected with the 8-bit general-purpose register S. Except that the serial data is read starting from the least significant bit of register R and that port S is not used for input, register S has the same configuration as that of register Q described above. It stores data transferred from registers A and B, and also serial data and the 8-bit value of the data field in the ROM. By use of an input/output instructions, port S and register S can transfer 8-bit data in parallel.

The output port T consists of 8 bits, has an 8-bit output latch and is connected with the 8-bit general-purpose register T. Except that the serial data is read from the least significant bit of register S for transfer of serial output from the port data, register T has the same configuration as that of register S described above. It stores the data transferred from registers A and B, serial data, and also the 8-bit value of the data field in the ROM. By suitable input/output instructions, port T and register T can transfer the 8-bit data in parallel.

By input/output instructions, the 8-bit data of port Q, port R, port S and port T can be transferred to each general-purpose register.

When port Q or port R is used for input or output, it is necessary to set the input or output mode by SMR instructions.

When the general-purpose register is used as a serial input or serial output shift register, the value of the data field stored in the register is shifted in the register, or '0'/'1' control data is entered as the input from the most significant

bit of the general-purpose register Q for shift control, and data is also transferred from the least significant bit of the general-purpose register T through the serial input/output port, DATA. External serial data are received at the serial input/output port DATA, and read into the most significant bit of the general-purpose register Q. Though the input/output port CLK is normally in floating status, it generates A shift clock pulse synchronized with transmitting data in the output mode, and reads the external shift clock pulse synchronized with receiving data.

Timer 1, the basic source oscillation frequency (1/3 of one machine cycle) or an external reference oscillation frequency, divided by 14, is used as a pre-scaler.

Timer 2 and the event counter consist of 4 bits each and are used as a discrete unit or in combination for multiple applications.

PIN DESCRIPTIONS

Pin	Name	Input or output	At reset	Function
X_{IN}	Source oscillation clock input	Input	—	Incorporates the clock oscillation circuit, for setting of the oscillation frequency. The oscillation reference device such as a ceramic filter for IF is connected between X_{IN} and X_{OUT}. When an external clock is used, connect the clock oscillation source to the X_{IN} pin and leave the X_{OUT} pin open.
X_{OUT}	Source oscillation clock output	Output	—	
RESET	Reset signal	Input	—	Resets the program counter PC and mode registers, and performs the reset initiation of the related input ports and output ports. For input/output ports, refer to the column for "At reset" of this table.
INTA	Interrupt request signal A	Input	Disable	Input signals for interrupt request. Request is accepted on the rising edge of the signal. Besides these external input signals, the interrupt requests T from timer 2/event counter are also received in the relative order $RESET > INT_A > INT_T > INT_B$. Since the interrupt requests are held at each latch, there will be none undetected.
INTB	Interrupt request signal B	Input	Disable	
EC	Event counter input	Input	—	The input signal for the event counter, which programs $2^0 \sim 2^4$ events of the event mode. This value is set as an initial value and countdown starts from this value to reach F_{16}, which then generates interrupt request signal INT_T.
$A_0 \sim A_{11}$	Address output port A	Output	Floating	The address signal for main memory (RAM) externally connected, in the form of a 3-state output. At MM mode where external memory is used, the data of the data pointer DP is read out directly. In SM mode where internal memory (RAM) is used, the data of the data pointer Y immediately before switching to MM mode is transferred to the auxiliary latch (4 bits) prior to read-out. However, the lower 8 bits of the address signal ($A_0 \sim A_7$) are not affected by this mode, since data pointers X and Z are not related to latch operation.
$D_0 \sim D_3$	Data input/output port D	Input/ output	Floating	A 3-state input/output port to execute data transfer in 4-bit units to/from an externally connected main memory (RAM). Switching of input-output is made automatically by instruction.
OD	External RAM read signal	Output	Floating	The output port is 3-state and the read signal is generated at the data input cycle of the externally connected main memory (RAM). During a read cycle, it becomes automatically set to low-level.
R/W	External RAM write signal	Output	Floating	The output port is 3-state and the write signal generated at the data write cycle is in the externally connected main memory (RAM). During a write cycle, it is automatically set to low-level.
$U_0 \sim U_3$	Output port U	Output	Floating	The output port enables 3-state setting per 1-bit unit. The 3-state condition is modified by the data content of register B, and the data of register A is output. The output setting of port U, however, is made either by instruction SU unconditionally or by the instruction TPRA, or TPRN, which transfers the data of the general-purpose register to ports Q, R, S and T.
$Q_0 \sim Q_7$	Input/output port Q	Input/ output	Input	The input/output port for 8-bit data transfer to/from register Q. Register Q enables data transfer between register A and register B. By instruction OPI, this port also functions to load the value (8-bit) of the immediate field of the ROM to register Q. Port Q data can be transferred to registers A and B as an input signal of 8 bits.
$R_0 \sim R_7$	Input/output port R	Input/ output	Input	The input/output port for 8-bit data transfer to/from register R. Register R enables data transfer between register A and register B. By instruction OPI, this port also functions to load the value (8-bit) of the data field of the ROM to register R. When port R is used as the input signal of a 4-bit unit, the data, 4 bits each, can be transferred to register B.
$S_0 \sim S_7$	Output port S	Output	Low-level	The output port that enables 8-bit data transfer to/from register S. Register S enables data transfer between register A and register B. By instruction OPI, this port also functions to load the value (8-bit) of the data field of the ROM to register S.
$T_0 \sim T_7$	Output port T	Output	Low-level	The output port for 8-bit data transfer to register S. Register T enables data transfer between register A and register B. By instruction OPI, this also functions to load the value (8-bit) of the immediate field of the ROM to register T.
DATA	Serial data port	Input/ output	Floating	The input/output port normally is floating to handle the serial data of the 32-bit general-purpose register. At output mode, data of the least significant bit of the general-purpose register (the least significant bit of register T) is read out, and at the input mode, the input is to the most significant bit of the general-purpose register (the most significant bit of register Q).
CLK	Serial data shift clock signal	Input/ output	Floating	The input/output port is normally floating to generate a shift clock pulse synchronized with the above serial data port. At output mode, a shift clock pulse synchronized with the data transmission is generated, and at the input mode, a shift pulse synchronized with the rate of data receiving is applied.

7

SINGLE-CHIP 4-BIT CMOS MICROCOMPUTER

BASIC FUNCTIONAL BLOCKS AND THEIR OPERATIONS

Program Counter PC

The program counter consists of 12 bits, the upper 5 bits of which compose the page register, and the lower 7 bits the binary counter. It designates the address of the 4096 words of 10 bits mask-programmable ROM, and controls the read-out sequence of instructions stored in the ROM. The ROM is composed of 32 pages of 128 words, and when program execution completes instruction at address 127, the binary counter is set to 0 and the next page is automatically incremented in the page-designation register. In the page register of the program counter, the contents of register P are loaded by instructions BL, BA, BM, and BMA. The binary counter is incremented as every single instruction is executed, but in the execution of instructions B, BL and BM, the value of the data field of the ROM is loaded, becoming the value of the specified address. In the execution of instructions BA and BMA, the contents of register A are loaded in the upper 4 bits of the binary counter, and the value of the data field is loaded in the lower 3 bits. Thereby a multi-branch instruction modified by the contents of the A register can be carried out.

There are three instructions relating to register P: instruction LP loads the value (5 bits) of the data field of the ROM, and instructions TPAC and TACP transfer data between register A and carry flag CY.

The 12-bit contents of the program counter PC can be saved for up to 12 levels in the fixed stack area of the external main memory (RAM). In the execution of instructions BM and BMA, control can be returned to a former routine by storing the contents of the program counter before branching, in the execution of instructions RT, RTS, and RTI. The fixed addresses to be jumped to and the priority order of four factors in the interrupt request are defined as follows:

(1) In case of by reset signal RESET page 0, address 0
(2) In case of interrupt signal INT_A page 0, address 2
(3) In case of interrupt signal INT_T page 0, address 8
(4) In case of interrupt signal INT_B page 0, address 4

INT_T is the interrupt request signal from timer II and the event counter.

Instruction BMAB is provided for easy handling of data conversion or using ROM as data tables, and usually by application of this instruction in combination with the instruction OPI. In two machine cycles it can load the 8-bit value of the data field of the ROM addressed by the contents of registers A and B into an arbitrary general-purpose register (Q, R, S and T).

Instruction BMAB branches unconditionally to the address derived by using the contents of register A for the low-order 4 bits of the 12-bit PC, those of register B for the middle 4 bits, and those of the upper 4 bits of the 5-bit register P for the upper 4 bits, and then executes the instruction OPI of the branch, and simultaneously returns automatically.

Instruction OPI loads one of the four general-purpose registers selected by the input/output address r with the value (8 bits) of the data field. The input/output address r is latched with the contents of the lower 2 bits of the data field in the execution of the instructions BMAB, TNAB, TABN, TPRN and TRPN and determines the register which loads data in the execution of the instruction OPI. To enable independent use of instruction OPI, the input/output address r, along with the carry flag CY and the mode flag MF, gives and takes data to and from the register by instructions TACM and TCMA. Thereby data can be saved and returned at interrupt time.

Table 1 Relationship between input/output address r and general-purpose registers

Input/output address	Immediate data r in execution of the instructions BMAB, TNAB, TPRN and TRPN		General-purpose register to be selected
N	I_1	I_0	
0	0	0	Register Q
1	0	1	Register R
2	1	0	Register S
3	1	1	Register T

Stack Pointer SP

A stack of 12 levels is provided for saving of the program counter PC in the fixed address area within the external main memory (RAM), and the contents of the stack pointer are used during addressing. The contents of the stack pointer are incremented by an interruption or in the execution of instructions BM and BMA, and are decremented in the execution of instructions RT, RTS and RTI.

Data Pointer DP

This is a register of 12 bits addressing memory, being composed of registers X, Y, and Z, having 4 bits each. Register X addresses 16 files, each of which comprises 16 words. By making an exclusive OR with 2 bits of the data field in the execution of instructions TAM, XAM, XAMD, and XAMI, the lower 2 bits can modify the next file designation. Register Y addresses data of 16 files (a file comprises 16 words), being incremented and decremented by the arithmetic unit in the execution of the instructions INY, DEY, XAMI, XAMD, TSMI, and TMSI. When the contents become an O or F which is the boundary of a file, control skips execution of the next instruction. Register Z permits address specification such that data memory may be extended up to maximum of 16 sets of 4096 words by 4 bits, where one unit comprises 16 files (256 words by 4 bits).

Since the address of the external main memory (4096 words by 4 bits maximum) and the internal scratch-pad

memory (32 words by 4 bits) are designated identically, the external main memory is selected by instruction MM, and the internal scratch-pad memory by instruction SM.

The contents of DP can be saved for up to 4 levels in the fixed stack region of the external main memory. This pointer is saved during the execution of instruction SDP, and is restored by instruction LDP. Address signals $A_0 \sim A_{11}$ for the external main memory are output from the contents of DP in the MM mode, except in case of interruption or the execution of instructions BM, BMA, RT, RTS, RTI, SDP, and LDP. In the SM mode the address signals, except for $A_0 \sim A_7$, are output from DP after latching the contents of the data pointer to the auxiliary output latch, but prior to changing the mode. During an interrupt or execution of instructions BM, BMA, RT, RTS, RTI, SDP or LDP, the partial address is secured independent of the mode. This is designated by Z=0 (external basic main memory), X=D, E or F where Y=the contents of the stack pointer SP of the program counter PC, or the value of the data field (indicated save level of the data pointer), as shown in the following figure. When the data pointer stack is not used, all of the stack region can be used as program counter stack for a total of 16 levels.

Fig. 1 External basic main memory (Z=0) and RAM map

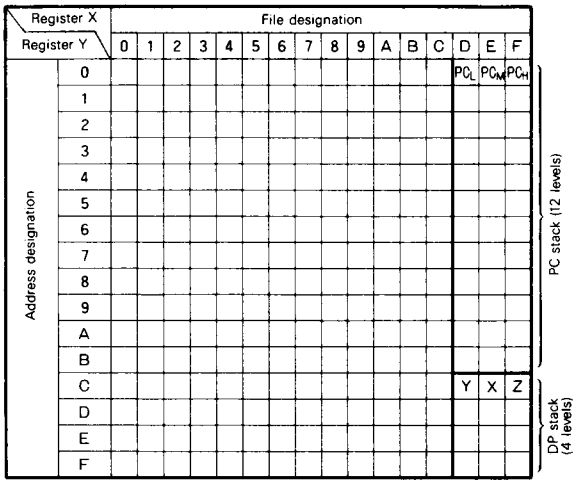


Table 2 Address designation of data pointer stack

Value of data field during execution of instructions SDP and LDP		Stack DP (file designated by register Y)
I_1	I_0	
0	0	C
0	1	D
1	0	E
1	1	F

When the internal scratch-pad memory is addressed in the SM mode, only five bits (four bits of register Y plus the least significant bit of register X) are employed.

Accumulator (Register A), Carry Flag CY

Register A is an accumulator forming the central unit of a 4-bit-wide microcomputer. Data processing operations such as arithmetic, data transfer, data exchange, data conversion, input/output, etc. are executed principally with this register.

The carry flag CY stores the carry or borrow from the most significant bit of the arithmetic unit in the execution of specific arithmetic instructions, and is available for multipurpose uses as a one-bit flag.

Auxiliary Register (Register B)

Register B is composed of four bits. It is employed for bit operating functions, temporary memory of four-bit data and transfer of eight-bit data when coupled with register A, etc.

Four-Bit Arithmetic Logic Unit (ALU)

This unit carries out four-bit arithmetic and logical functions, and is composed of a four-bit adder and a logic circuit associated with it. It carries out addition, complement conversion, logic arithmetic comparison, arithmetic comparison, bit processing, etc.

7

Timers and Event Counter

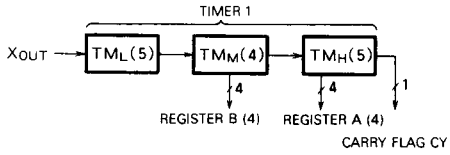
This block is composed of a 14-bit timer 1, a 4-bit timer 2 and a 4-bit event counter.

Timer 1 is a standard timer that continuously counts the frequency X_{IN}, divided by fourteen. The timer performs accurate counting and the period is given by the following formula:

(Fundamental output frequency X_{IN}) x
2^5 (TM_L) x 2^4 (TM_M) x 2^5 (TM_H) = cycle time of timer 1

By the continuous use of instructions TATM and TBTM, the contents of TM_M are stored in register B, the contents of the lower 4 bits of TM_H in register A, and the high-order bit of TM_H in carry flag CY, respectively. The contents of timer 1 can be accessed. Instruction RTM clears the contents of timer 1 and resets it to 0.

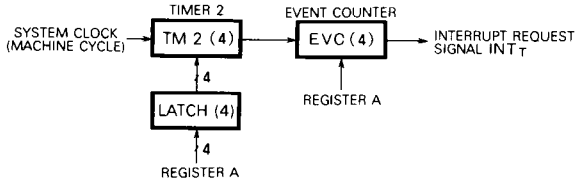
Fig. 2 Outline of timer 1 configuration



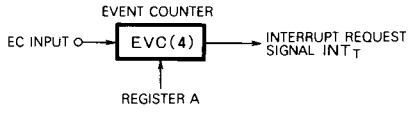
Timer 2 is composed of a 4-bit counter and a 4-bit latch. The contents of register A are stored as the starting value in the latch and the counter by an STM instruction, whereupon counting down starts in synchronization with each machine cycle. When the contents of the counter become F during countdown, the pre-programmed starting value is restored in the counter from the latch.

Fig. 3 Outline configuration of timer 2 and event counter

(1) Timer mode: When TMM=1 is set by the instruction SMR1



(2) Event mode: When TMM=0 is set by the instruction SMR1



The cycle period of timer 2 is given by the following formula:

Machine cycle x [1 + (2^0~2^4)]

Where the timer mode is set by SMR1 instruction, timer 2 is connected to the event counter. Every time the contents of timer 2 become F the event counter counts down once. For the event counter, the contents of register A can be stored in the counter and used as a starting value by using instruction SEC.

When the event mode is set using instruction SMR1, the event counter is counted down by sensing the rising edge of external event counter input EC.

In both timer mode and event mode, the event counter is counted down from a starting value, and an interrupt request signal is generated when the contents become F.

The time necessary for INT_T generation from the starting value is given by the following formulas:

Timer mode

Machine cycle x [1 + (2^0~2^4) x (2^0~2^4)]

Event mode

EC input period x (2^0~2^4)

The recurrence period of the shift clock pulse CLK which is generated in synchronization with data transmission for series data transmission is given by the period programmed by timer 2. The CLK output is set by the mode SDM=1, RVM=0 of instruction SMR1. It is generated with a period determined by the contents of the latch of timer 2 and by the execution of the instructions SST and RST.

General-Purpose Registers Q, R, S, and T

These general-purpose registers comprise a set of four 8-bit shift registers. When using combinations of functions such as serial input, serial output, parallel input and parallel output, by properly selected instructions, they are employed for data transfer between register A and register B, data transfer between output ports or input/output ports, data storage of the data field of the ROM value (8 bits), transmission of internal serial data, receiving of external serial data, etc.

When the general-purpose registers are used as a single 32-bit shift register, four kinds of modes as shown in Table 3 can be set by instruction SMR1. The shift instructions comprise instruction SST, which shifts 32 bits of data by setting the shift register input to '1'; instruction RST, which shifts 32 bits of data and resets the input to '0'; and instruction IST, which shifts the data by reading the data of the serial data input data. Instruction IST, except in the mode where SDM=0 and RVM=1, reads the serial data of the DATA terminal (which is used as the input or the output terminal). In the mode where SDM=0 and RVM=1, it reads the data at the rising edge of the clock input CLK and shifts the data. Note that it cannot respond to data input that has a transfer rate faster than the machine cycle rate, since detection of the rising edge of CLK is carried out by using the internal clock pulse. In the mode where SDM=1 and RVM=0, data is output synchronized with the clock pulse output CLK generated by the period programmed in timer 2 (from the least significant bits of the general-purpose register). In the mode where SDM=1 and RVM=1, data is output synchronized with the shift instruction and is transmitted from the least significant bit of the general-purpose register.

Instruction TNAB stores 8-bit data from registers A and B in one of the general-purpose registers designated by the input/output address r, which is defined by 2 bits of the data field.

Instruction TABN transfers 8-bit data from one of the general-purpose registers designated by the input/output address r, which is defined by 2 bits of the data field into A and B registers.

Instruction OPI stores the 8-bit value of the data field in one of the general-purpose registers designated by the input/output address r by using it in combination with instruction BMAB or by using instruction OPI alone, as described in the explanation of the program counter function.

Instruction TPRN stores the contents of the register designated by the input/output address r in the latch of the output port corresponding to the input/output address r, which is then usable.

Instruction TRPA stores the contents of the 32-bit register in corresponding latches of all the output ports. These are then valid at the outputs.

Instruction TRPN can restore from the contents of the output latch port designated by the input/output address r into the register corresponding to the input/output address r.

Interrupt Function

This microcomputer has a hardware interrupt function for four conditions by one-level. The interrupt requests comprise: the RESET signal; the interrupt request signals INT_A and INT_B as external signals; and the interrupt request signal INT_T by the internal event counter. The order of priority is determined as follows:

RESET > INT_A > INT_T > INT_B

The interrupt enable instructions comprise EIA, EIB, EIAB and EIT and the interrupt disable instructions comprise DIA, DIB, DIAB and DIT. A RESET signal restores the hardware to the initial state, independent of any current instruction.

In an interrupt enable state, the interrupt is accepted at the rising edge of interrupt request signals INT_A and INT_B. When an interruption is requested in an interrupt disable state, the interrupt is not executed. If the interrupt disable state is removed thereafter and a corresponding interrupt enable instruction is executed, the interrupt routine will be executed immediately because the interrupt request has been held in the latch. The current interrupt request, held in a latch during the interrupt disable state, is reset by the interrupt disable instruction.

When two and more interrupt requests of four factors occur simultaneously, the interrupt processing is by order of the highest priority routine. The interrupt request of lower priority order is held in the corresponding latch in an interrupt disable state. When the interrupt disable state is removed by the interrupt enable instruction (after completion of the interrupt process of upper priority order), the interrupt request of next lower priority is initiated.

Table 3 Mode setting by instruction SMR 1; when the general-purpose registers are employed as a 32-bit shift register

Mode flag					
	SDM	0	0	1	1
	RVM	0	1	0	1
DATA terminal		Input	Input	Output	Output
CLK terminal		Floating	Input (rising edge trigger)	Output (generated by timer 2)	Output (generated by shift instruction)
Shift data input	SST, RST	Immediate field data	DATA terminal input	Immediate field data	Immediate field data
	IST	DATA terminal outpt		DATA terminal output	DATA terminal output
Shift clock pulse		Instructions SST, RST, IST	CLK input	Instructions SST, RST, IST	Instructions SST, RST, IST
Transmission, receiving		Receiving (only in instruction IST)	Receiving	Transmission	Transmission

Mode Register

The mode register is composed of 10 bits, and can select operation modes and functions, etc. of the associated input port or output port by setting or resetting the mode flag corresponding to a bit in register A.

The mode setting by the instruction SMR is shown in Table 5.

The mode flag IMQ will select the use of the input/output port Q for input or output.

The mode flags IMR_1 and IMR_2 select the use of the input/output port R for input or for output in 4-bit unit.

The mode flag LCD will select the instruction that activates output port U. In case of status '0', only instruction SU is valid. In case of status '1', port U can be set by instruction SU and by instructions TPRN and TPRA, which move the contents of the general-purpose register to the output port. Every bit of port U, as shown in Table 4, can be programmed for three states as determined by the contents of registers A and B. For example, by using instruction TPRA in LCD=1, we can drive a 1/2-duty liquid crystal display panel by the 1/2 voltage equalization method, where port U is used for the common output and the ports Q, R, S and T for the segment outputs.

The mode setting by instruction SMR1 is shown in Table 6.

The mode flag TMM determines whether the event counter is in the independent event mode or in the timer mode by connecting with timer 2.

The mode flag determines whether all the three-state signals for the external main memory (RAM), $A_0 \sim D_{11}$, $D_0 \sim D_3$, OD and R/W, are in floating or activated state.

The mode flags RVM and SDM select the functions of the terminals DATA and CLK, which are the transmission/receiving and input/output ports, when the 32-bit general-purpose register is used as a shift register. For further details, refer to the explanation of the general-purpose register.

Table 4 Three-state-condition setting of output port U

Contents of register B	Contents of register A	State of port U (when executing SU, TPRN or TRRA)
0	0	Floating
0	1	Floating
1	0	0
1	1	1

Note 1 : Registers A, B, and port U correspond to one another in regard to ther bit order.

Table 5 SMR mode setting

Bits of register A	Mode flag (contents of register A are stored)	Status	Function	Mode flag at reset
A_0	IMQ	0	Port Q is used as an 8-bit input port.	0
		1	Port Q is used as an 8-bit output port.	
A_1	LCD	0	Port Q is used as an 8-bit output port.	0
		1	For output port U, instructions TPRN and TPRA for port Q, R, S and T can also set port U.	
A_2	IMR1	0	Port R_1 is used as a 4-bit input port.	0
		1	Port R_1 is used as a 4-bit output port.	
A_3	IMR2	0	Port R_2 is used as a 4-bit input port.	0
		1	Port R_2 is used as a 4-bit output port.	

Table 6 SMR 1 mode setting

Bits of register A	Mode flag (contents of register A are stored)	Status	Function	Mode flag at reset
A_0	TMM	0	Event mode: event counter is used with EC input.	0
		1	Timer mode: event counter is used in combination with timer 2.	
A_1	BF	0	All signals ($A_{11} \sim A_0$, $D_3 \sim D_0$, OD and R/W) for external main memory (RAM) are put in floating.	0
		1	All signals ($A_{11} \sim A_0$, $D_3 \sim D_0$, OD and R/W) for external main memory (RAM) are activated.	
A_2	RVM	0	When the general-purpose registers are used as a 32-bit shift register, functions of transmission/receiving, terminals DATA and CLK are employed properly by RVM, SDM flags. For further details, refer to explanation of the general-purpose register.	0
		1		
A_3	SDM	0		0
		1		

Outline Specifications of M58494-XXXP

Item			Performance
Number of basic instructions			92
Execution time of basic instructions			6.6μs (at V_{CC}=5V, f=455kHz)
Clock frequency			200~455kHz
Memory capacity	ROM		4096 words by 10 bits
	RAM (built-in)		32 words by 4 bits
	RAM (external)		4096 words by 4 bits (max.)
Input/output port for external RAM	Address (port A)		12 bits × 1 (3 states)
	Control signal (port OD and R/W)		2 bits (3 states)
	Data bus (port D)		4 bits × 1 (3 states)
Input/output port	Q	Input	8 bits × 1
		Output	8 bits × 1
	R	Input	4 bits × 2
		Output	8 bits × 1
	S	Output	8 bits × 1
	T	Output	8 bits × 1
	DATA	Serial data	1 bit (input/output port)
	CLK	Synchronizing pulse	1 bit (input/output port)
	U	Output	4 bit × 1 (3-state)
	EC	Input	1 bit
Subroutine nesting			12 levels
Interrupt request			4 factors 1 level
Saving of data pointer			4 levels
Clock generation circuit			Built-in (oscillation reference element is outside)
Ports input/output characteristics	Absolute maximum rating voltage		V_{CC}
	Input/output characteristics		Interchangeable with CMOS logic series
Power supply voltage	V_{CC}		5V (nominal)
	V_{SS}		0V
Element structure			CMOS
Package			68-pin plastic molded flat package
Power dissipation			5 mW (at V_{CC}=5V, f=455kHz)

Note 2 : Ports to be used will be determined in accordance with user's specifications.

7

SINGLE-CHIP 4-BIT CMOS MICROCOMPUTER

MACHINE INSTRUCTIONS

Item / Classification	Symbol	Code I9 I8 I7 I6 I5 I4 I3 I2 I1 I0	16mal notation	No. of words	No. of cycle	Function	Skip conditions	Flag CY
RAM address	**MM**	00 1000 0010	**082**	1	1	(MF)←1, Selects external main memory	—	—
	SM	00 1000 0000	**080**	1	1	(MF)←0, Selects internal scratch-pad memory	—	—
	LY y	01 1000 yyyy	**18y**	1	1	(Y)←y, where y = 0 ~ 15	Consecutively described	—
	LX x	01 1011 xxxx	**1Bx**	1	1	(X)←x, where x = 0 ~ 15	Consecutively described	—
	LZ z	01 1010 zzzz	**1Az**	1	1	(Z)←z, where z = 0 ~ 15	Consecutively described	—
	INY	00 0111 1100	**07C**	1	1	(Y)←(Y) + 1	(Y) = 0	—
	DEY	00 0111 1000	**078**	1	1	(Y)←(Y) − 1	(Y) = 15	—
	TAY	00 0010 0000	**020**	1	1	(A)←(Y)	—	—
	TAX	00 0010 0010	**022**	1	1	(A)←(X)	—	—
	TAZ	00 0010 0011	**023**	1	1	(A)←(Z)	—	—
	TYA	00 0100 0000	**040**	1	1	(Y)←(A)	—	—
	TXA	00 0100 0010	**042**	1	1	(X)←(A)	—	—
	TZA	00 0100 0011	**043**	1	1	(Z)←(A)	—	—
	SDP j	00 0111 01jj	**074** + j	1	3	(Mj)←(DP), where j = 0 ~ 3	—	—
	LDP j	00 1111 01jj	**0F4** + j	1	3	(DP)←(Mj), where j = 0 ~ 3	—	—
Register-to-register transfer	**TSM**	00 1011 1100	**0BC**	1	1	(SM(DP)) ← (MM(DP))	—	—
	TSMI	00 1111 1100	**0FC**	1	1	(SM(DP)) ← (MM(DP)), (Y)←(Y) + 1	(Y) = 0	—
	TMS	00 1011 1110	**0BE**	1	1	(MM(DP)) ← (SM(DP))	—	—
	TMSI	00 1111 1110	**0FE**	1	1	(MM(DP)) ← (SM(DP)), (Y)←(Y) + 1	(Y) = 0	—
	TAB	00 1010 0000	**0A0**	1	1	(A)←(B)	—	—
	TBA	00 1100 0000	**0C0**	1	1	(B)←(A)	—	—
	TASP	00 1010 0010	**0A2**	1	1	(A)←(SP)	—	—
	TSPA	00 1100 0010	**0C2**	1	1	(SP)←(A)	—	—
	TACM	00 1000 0100	**084**	1	1	(A)←(N, MF, CY), where $A_{3\sim2}$=N, A_1=MF, A_0=CY	—	—
	TCMA	00 1100 1100	**0CC**	1	1	(N, MF, CY)←(A), where $A_{3\sim2}$=N, A_1=MF, A_0=CY	—	—
Transfer between RAM and accumulator	**TAM j**	00 0010 01jj	**024** + j	1	1	(A)←(M(DP)), (X)←(X)∀j, where j = 0 ~ 3	—	—
	XAM j	00 0110 01jj	**064** + j	1	1	(A)←→(M(DP)) (X)←(X)∀j, where j = 0 ~ 3	—	—
	XAMD j	00 0110 10jj	**068** + j	1	1	(A)←→(M(DP)), (Y)←(Y) − 1 (X)←(X)∀j, where j = 0 ~ 3	(Y) = 15	—
	XAMI j	00 0110 11jj	**06C** + j	1	1	(A)←→(M(DP)), (Y)←(Y) + 1 (X)←(X)∀j, where = 0 ~ 3	(Y) = 0	—
	XAMD1 j	00 1110 10jj	**0E8** + j	1	1	(A)←→(M(DP)), (Y)←(Y) − 1 (X)←(X)∀j, where j = 0 ~ 3	(Y) = 3, 7, 11, 15	—
	XAMI1 j	00 1110 11jj	**0EC** + j	1	1	(A)←→(M(DP)), (Y)←(Y) + 1 (X)←(X)∀j, where j = 0 ~ 3	(Y) = 4, 8, 12, 0	—
	TMA	00 0100 0100	**044**	1	1	(M(DP))←(A)	—	—
Arithmetic	**LA n**	01 1001 nnnn	**19n**	1	1	(A)←n, where , n = 0 ~ 15	Consecutively described	—
	AM	00 0110 0000	**060**	1	1	(A)←(A) + (M(DP))	—	—
	AMC	00 0110 0010	**062**	1	1	(A)←(A) + (M(DP)) + (CY), (CY)←Carry	—	0/1
	AMCS	00 0110 0011	**063**	1	1	(A)←(A) + (M(DP)) + (CY), (CY)←Carry	Carry = 1	0/1
	A n	00 0101 nnnn	**05n**	1	1	(A)←(A) + n, where , n = 0 ~ 15	Carry = 0	—
	SC	00 1000 1010	**08A**	1	1	(CY)←1	—	1
	RC	00 1000 1000	**088**	1	1	(CY)←0	—	0
	SZC	00 1011 1000	**0B8**	1	1		(CY) = 0	—
	CMA	00 1011 1010	**0BA**	1	1	(A)←($\overline{A}$)	—	—

MITSUBISHI ELECTRIC

Item / Classification	Symbol	Code		No. of words	No. of cycle	Function	Skip conditions	Flag CY
		I9 I8 I7 I6 I5 I4 I3 I2 I1 I0	16mal notation					
Bit manipulation	SB j	00 1000 11jj	08C + j	1	1	(B(j))←1, where, j= 0 ~ 3	—	—
	RB j	00 1010 11jj	0AC + j	1	1	(B(j))←0, where, j= 0 ~ 3	—	—
	SZB j	00 0011 10jj	038 + j	1	1		(B(j))= 0 where, j=0~3	—
	SZM j	00 0000 01jj	004 + j	1	1		(Mj(DP))= 0 where, j=0~3	—
Compare	SEAM	00 1110 0000	0E0	1	1		(A)=(M(DP))	—
	SEY n	00 0001 nnnn	01n	1	1		(Y)=n where, n=0~15	—
	SEI n	00 1001 nnnn	09n	1	1		(A)=n where, n=0~15	—
Branch	B xy	01 0xxx yyyy	1xy	1	1	(PCL)←y, (PCM)←x where 16x+y=0~127	—	—
	BL xy	11 0xxx yyyy	3xy	1	1	(PCL)←y, (PCM)←(P0, x) (PCH)←(P4, P3, P2, P1) where 16x+y= 0 ~127	—	—
	BA i	00 1101 0iii	0Di	1	1	(PCL)←(A0, i) where, i= 0 ~ 7 (PCM)←(P0, A3, A2, A1) (PCH)←(P4, P3, P2, P1)	—	—
	BMAB r	00 1100 10rr	0C8 + r	1	1	(PCL)←(A) (PCM)←(B) (PCH)←(P4, P3, P2, P1) but returns unconditionally after one machine cycle. Input/output address r = 0~3 designates general-purpose register.	—	—
	LP p	01 110p pppp	1C0 ++ pp	1	1	(P)←p where memory page number (p = 0, 1, 2, 3, 4, 5.....30, 31) corresponds to decimal notation 0, 16, 1, 17, 3, 18.....15, 31	Consecutively described	—
	TPAC	00 1100 0100	0C4	1	1	(P)←(CY, A)	—	—
	TACP	00 1010 0100	0A4	1	1	(CY, A)←(P)	—	—
Subroutine call	BM xy	11 1xxx yyyy	38y + x	1	3	(PCL)←y (PCM)←(P0, x), where 16x + y =0~127 (PCH)←(P4, P3, P2, P1) (M(SP))←(PC) (SP)←(SP)+ 1	—	—
	BMA i	00 1101 1iii	0D8 + j	1	3	(PCL)←(A0, i), where, j=0~7 (PCM)←(P0, A3, A2, A1) (PCH)←(P4, P3, P2, P1) (M(SP))←(PC) (SP)←(SP)+ 1	—	—
Return	RT	00 1111 1000	0F8	1	3	(PC)←(M(SP)) (SP)←(SP)− 1	—	—
	RTS	00 1111 1010	0FA	1	4	(PC)←(M(SP)) (SP)←(SP)− 1 (PC)←(PC)+ 1	Unconditionally	—
	RTI	00 1111 1001	0F9	1	3	(PC)←(M(SP)) (SP)←(SP)− 1 Resets the interrupt control.	—	—

7

SINGLE-CHIP 4-BIT CMOS MICROCOMPUTER

Item / Classification	Symbol	Code $I_9 I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0$	16mal notation	No. of words	No. of cycle	Function	Skip conditions	Flag CY
Interrupt flip-flop control	EIA	00 0000 1001	009	1	1	Enables interruption of INT_A signal.	—	—
	EIB	00 0000 1010	00A	1	1	Enables interruption of INT_B signal.	—	—
	EIAB	00 0000 1011	00B	1	1	Enables interruption of INT_A and INT_B signals.	—	—
	EIT	00 0000 1000	008	1	1	Enables interruption of INT_T signal.	—	—
	DIA	00 0000 1101	00D	1	1	Disables interruption of INT_A signal.	—	—
	DIB	00 0000 1110	00E	1	1	Disables interruption of INT_B signal.	—	—
	DIAB	00 0000 1111	00F	1	1	Disables interruption of INT_A and INT_B signals.	—	—
	DIT	00 0000 1100	00C	1	1	Disables interruption of INT_T signal.	—	—
Timer	TBTM	00 0010 1111	02F	1	1	$(B) \leftarrow (TM_M)$	—	—
	TATM	00 1010 0111	0A7	1	1	$(A) \leftarrow (TM_{H3}, TM_{H2}, TM_{H1}, TM_{H0})$ $(CY) \leftarrow (TM_{H4})$		
	RTM	00 1011 0100	0B4	1	1	$(TM_L) \leftarrow 0$, $(TM_M) \leftarrow 0$, $(TM_H) \leftarrow 0$	—	—
	STM	00 1100 0111	0C7	1	1	$(TM\,II) \leftarrow (A)$	—	—
	SEC	00 1100 0110	0C6	1	1	$(EVC) \leftarrow (A)$	—	—
Input/output	ID	00 0010 1110	02E	1	1	$(B) \leftarrow (D)$, $(OD) \leftarrow$ low level	—	—
	OD	00 0100 1100	04C	1	1	$(D) \leftarrow (B)$, $(R/W) \leftarrow$ low level	—	—
	OPI s	10 ssss ssss	2ss	1	1	$(R(r)) \leftarrow s$ Where the general-purpose register is designated with r = 0~3	—	—
	TNAB r	00 0100 10rr	048 + r	1	1	$(R(r)) \leftarrow (A, B)$ Where the general-purpose register is designated with r = 0~3	—	—
	TABN r	00 0010 10rr	028 + r	1	1	$(A, B) \leftarrow (R(r))$ Where the general-purpose register is designated with r = 0~3	—	—
	IQ	00 1010 1000	0A8	1	1	$(A, B) \leftarrow (P(Q))$	—	—
	IR1	00 0010 1100	02C	1	1	$(B) \leftarrow (P(R_1))$	—	—
	IR2	00 0010 1101	02D	1	1	$(B) \leftarrow (P(R_2))$	—	—
Input/output control	SMR	00 0011 0100	034	1	1	$(MR) \leftarrow (A)$	—	—
	SMR1	00 0011 0110	036	1	1	$(MR1) \leftarrow (A)$	—	—
	SST	00 0011 1100	03C	1	1	$(R(Q_0)) \leftarrow 1$, $R(All) \leftarrow$ 1-bit shift R (All)	—	—
	RST	00 0011 1101	03D	1	1	$(R(Q_0)) \leftarrow 0$, $R(All) \leftarrow$ 1-bit shift R (All)	—	—
	IST	00 0011 1110	03E	1	1	$(R(Q_0)) \leftarrow (DATA)$, $R(All) \leftarrow$ 1-bit shift R (All)	—	—
	SU	00 0100 1110	04E	1	1	$(U) \leftarrow (A)$, when $(B)_I = 1$ $(U) \leftarrow$ floating, when $(B)_I = 0$	—	—
	CLP	00 0000 0001	001	1	1	$(P(All)) \leftarrow 0$	—	—
	TPRA	00 1011 0000	0B0	1	1	$(P(All)) \leftarrow (R(All))$	—	—
	TPRN r	00 1111 00rr	0Fr	1	1	$(P(r)) \leftarrow (R(r))$	—	—
	TRPN r	00 0111 00rr	07r	1	1	$(R(r)) \leftarrow (P(r))$	—	—
Others	NOP	00 0000 0000	000	1	1	No operation	—	—

MITSUBISHI ELECTRIC

Symbol	Details	Symbol	Details
A	4-bit register (accumulator)	P(Q)	8-bit port Q
A_i	Indicates the bits of register A. Where i = 0~3.	$P(R_1)$	4-bit port R_1
B	4-bit auxiliary register	$P(R_2)$	4-bit port R_2
B(j)	The bit of register B. addressed when j = 0~3.	R(All)	Indicates all the 8-bit registers, Q, R, S, T (32-bit)
CY	1-bit carry flag		
D	4-bit input/output port (3-state)	$R(Q_0)$	1st bit of register Q
DATA	1-bit input/output port for serial data	R(r)	The register selected by r (r corresponds with registers Q, R, S, and T where r = 0~3)
DP	12-bit data pointer composed of registers X, Y and Z		
EVC	4-bit event counter	R/W	1-bit output port which is used for the write signal of the external main memory
M(DP)	4-bit data memory addressed by the data pointer DP		
		SM(DP)	The 4-bit internal scratch-pad memory addressed by the data pointer DP
M_i	12-bit data from the scratch-pad memory addressed by i = 0~3 (data pointer number in the fixed area)		
		SP	4-bit stack pointer
M_i(DP)	4-bit data from external memory addressed by the contents data pointer DP, where i = 0~3.	TM1	14-bit counter composed of TM_L, TM_M and TM_H counters
		TM2	4-bit counter
MF	1-bit flag for selection of internal scratch-pad memory (MF ← 0 at instruction SM) or external main memory (MF ← 1 at instruction MM)	TM_H	5-bit counter
		TM_{Hi}	Indicates the bit of TM_H counter, where i = 0~4
		TML	5-bit counter
MM(DP)	4-bit external main memory data addressed by the data pointer DP	TM_M	4-bit counter
		U	4-bit output port (3-state)
M(SP)	12-bit data from external memory addressed by the stack pointer SP (return address stored in the fixed area)	X	4-bit register where X = 0~15, addressing the field of 16 words by 4 bits per file
MR	4-bit mode flag (IMQ, LCD, IMRI, IMR2)		
MR1	4-bit mode flag (TMM, BF, RVM, SDM)	Y	4-bit register where Y = 0~15, which addresses the word unit of 16 words by 4 bits.
OD	1-bit output port used for the read signal for external main memory.	Z	4-bit register where Z = 0~15, which addresses 16 files × 16 words × 4 bits
P	5-bit page register		
P_i	Indicates the bits of register P, where i = 0~4.	iii	3-bit binary variable
PC	12-bit program counter composed of counters PC_L, PC_M and PC_H	jj	2-bit binary constant
		nnnn	4-bit binary constant
PC_H	4-bit counter	ppppp	5-bit binary constant
PC_L	4-bit counter	r	Input/output address to select one of the general-purpose registers Q, R, S and T. (r = 0~3)
PC_M	4-bit counter		
P(All)	Indicates all the 8-bit ports, Q, R, S, T (32-bit)	rr	2-bit binary constant
		ssss ssss	8-bit binary constant
P(r)	The port selected by r (corresponds with ports Q, R, S, and T at r = 0~3)	xxxx	4-bit binary variable
		yyyy	4-bit binary variable
		zzzz	4-bit binary variable

7

INSTRUCTION CODE LIST

I_9~I_4 / 16mal notation / I_3~I_0		00 0000	00 0001	00 0010	00 0011	00 0100	00 0101	00 0110	00 0111	00 1000	00 1001	00 1010	00 1011	00 1100	00 1101	00 1110	00 1111	01 0000 ↓ 01 0111	01 1000	01 1001	01 1010	01 1011	01 1100 ↓ 01 1101	01 1110 ↓ 01 1111	10 0000 ↓ 10 1111	11 0000 ↓ 11 0111	11 1000 ↓ 11 1111
		00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10~17	18	19	1A	1B	1C~1D	1E~1F	20~2F	30~37	38~3F
0000	0	NOP	SEY 0	TAY	–	TYA	A 0	AM	TRPN 0	SM	SEI 0	TAB	TPRA	TBA	BA 0	SEAM	TPRN 0	B	LY 0	LA 0	LZ 0	LX 0	LP	–	OPI	BL	BM
0001	1	CLP	SEY 1	*	–	*	A 1	*	TRPN 1	*	SEI 1	*	*	*	BA 1	*	TPRN 1	B	LY 1	LA 1	LZ 1	LX 1	LP	–	OPI	BL	BM
0010	2	–	SEY 2	TAX	–	TXA	A 2	AMC	TRPN 2	MM	SEI 2	TASP	*	TSPA	BA 2	*	TPRN 2	B	LY 2	LA 2	LZ 2	LX 2	LP	–	OPI	BL	BM
0011	3	–	SEY 3	TAZ	–	TZA	A 3	AMCS	TRPN 3	*	SEI 3	*	*	*	BA 3	*	TPRN 3	B	LY 3	LA 3	LZ 3	LX 3	LP	–	OPI	BL	BM
0100	4	SZM 0	SEY 4	TAM 0	SMR	TMA	A 4	XAM 0	SDP 0	TACM	SEI 4	TACP	RTM	TPAC	BA 4	–	LDP 0	B	LY 4	LA 4	LZ 4	LX 4	LP	–	OPI	BL	BM
0101	5	SZM 1	SEY 5	TAM 1	*	*	A 5	XAM 1	SDP 1	*	SEI 5	*	*	*	BA 5	–	LDP 1	B	LY 5	LA 5	LZ 5	LX 5	LP	–	OPI	BL	BM
0110	6	SZM 2	SEY 6	TAM 2	SMR1	*	A 6	XAM 2	SDP 2	*	SEI 6	–	–	SEC	BA 6	–	LDP 2	B	LY 6	LA 6	LZ 6	LX 6	LP	–	OPI	BL	BM
0111	7	SZM 3	SEY 7	TAM 3	*	*	A 7	XAM 3	SDP 3	*	SEI 7	TATM	–	STM	BA 7	–	LDP 3	B	LY 7	LA 7	LZ 7	LX 7	LP	–	OPI	BL	BM
1000	8	EIT	SEY 8	TABN 0	SZB 0	TNAB 0	A 8	XAMD 0	DEY	RC	SEI 8	IQ	SZC	BMAB 0	BMA 0	XAMD1 0	RT	B	LY 8	LA 8	LZ 8	LX 8	LP	–	OPI	BL	BM
1001	9	EIA	SEY 9	TABN 1	SZB 1	TNAB 1	A 9	XAMD 1	*	*	SEI 9	*	*	BMAB 1	BMA 1	XAMD1 1	RTI	B	LY 9	LA 9	LZ 9	LX 9	LP	–	OPI	BL	BM
1010	A	EIB	SEY 10	TABN 2	SZB 2	TNAB 2	A 10	XAMD 2	*	SC	SEI 10	*	CMA	BMAB 2	BMA 2	XAMD1 2	RTS	B	LY 10	LA 10	LZ 10	LX 10	LP	–	OPI	BL	BM
1011	B	EIAB	SEY 11	TABN 3	SZB 3	TNAB 3	A 11	XAMD 3	*	*	SEI 11	*	*	BMAB 3	BMA 3	XAMD1 3	*	B	LY 11	LA 11	LZ 11	LX 11	LP	–	OPI	BL	BM
1100	C	DIT	SEY 12	IR1	SST	OD	A 12	XAMI 0	INY	SB 0	SEI 12	RB 0	TSM	TCMA	BMA 4	XAMI1 0	TSMI	B	LY 12	LA 12	LZ 12	LX 12	LP	–	OPI	BL	BM
1101	D	DIA	SEY 13	IR2	RST	*	A 13	XAMI 1	*	SB 1	SEI 13	RB 1	*	*	BMA 5	XAMI1 1	*	B	LY 13	LA 13	LZ 13	LX 13	LP	–	OPI	BL	BM
1110	E	DIB	SEY 14	ID	IST	SU	A 14	XAMI 2	*	SB 2	SEI 14	RB 2	TMS	*	BMA 6	XAMI1 2	TMSI	B	LY 14	LA 14	LZ 14	LX 14	LP	–	OPI	BL	BM
1111	F	DIAB	SEY 15	TBTM	*	*	A 15	XAMI 3	*	SB 3	SEI 15	RB 3	*	*	BMA 7	XAMI1 3	*	B	LY 15	LA 15	LZ 15	LX 15	LP	–	OPI	BL	BM

Note 3: I_3~I_0 indicate the low-order 4 bits of the machine code and I_9~I_4 show the high-order 6 bits.
Hexadecimal expressions of the codes are also given. All instructions are one word.
*, –: Do not use these codes