OKI semiconductor

MSM6052

CMOS 4BIT SINGLE CHIP LOW POWER MICROCONTROLLER FOR TELEPHONE

GENERAL DESCRIPTION

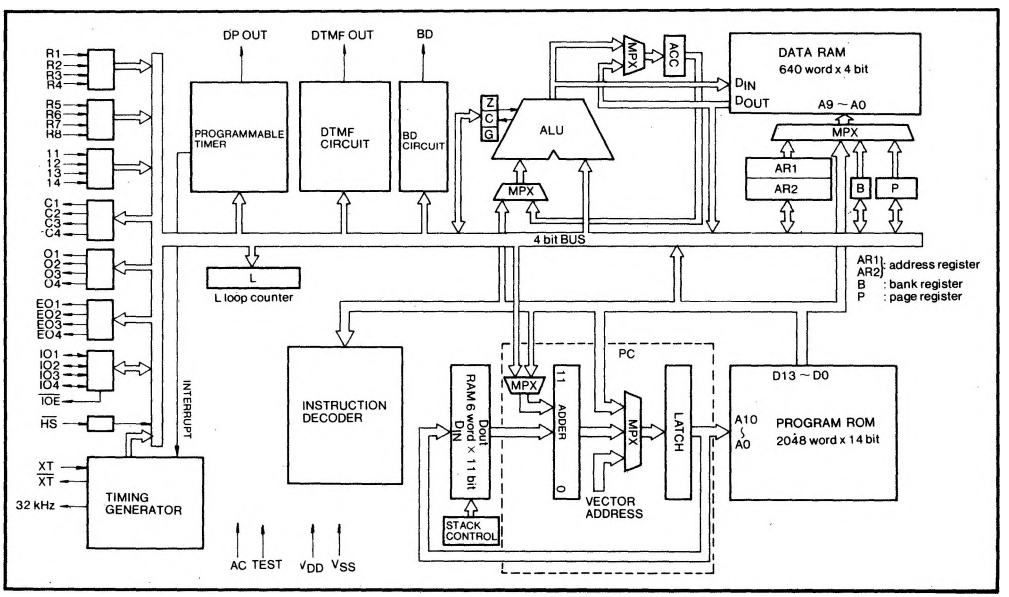
The OKI MSM6052 is low-power, high-performance single-chip 4-bit microcontroller employing complementary metal oxide semiconductor technology, especially designed for use in sophisticated telephone sets. Integrated onto a single chip are 4 bits of ALU, 28K bits of mask programmable ROM, 2560 bits of data RAM, programmable timer, oscillator, 12-bits of input port, 12-bits of output port and 4-bits of input/output port. In addition to these units, a DTMF generator is provided.

With the MSM6052, sophisticated telephone sets become feasible through a single chip instead of the conventional 3-chip configuration.

FEATURES

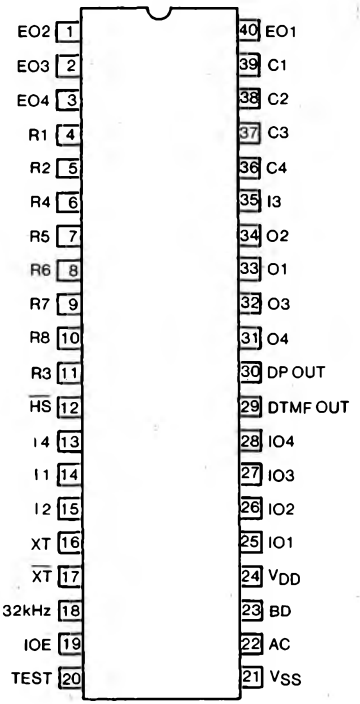
- Low Power Consumption 0.3mA Typical @3V (DTMF output off)
- 2048 × 14 Internal ROM
- 640 × 4 Internal RAM
- 3 × 4 Input Port
- 3 × 4 Output Port
- 1 × 4 Input/Output Port
- DTMF Generator
- Buzzer Sound Output
- 4-Bit Programmable Timer Applicable for Output of Dial Pulse
- Interrupt by Progammable Timer
- 5 Level Stack
- Power Down Mode
- 52 Instructions
- Instructions Useful for Data Management (Data Search and Block Data Transfer)
- 2.5 to 6.0V Operating Voltage
- 3.58 MHz Oscillator
- 17.9 μs Instruction Cycle
- −20 to 75°C Operating Temperature
- 28 Pin DIP or 40 Pin DIP

FUNCTIONAL BLOCK DIAGRAM

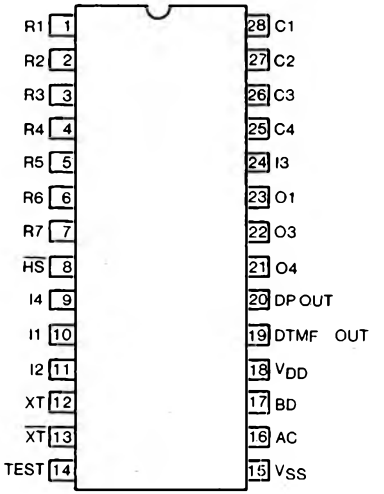


PIN CONFIGURATION

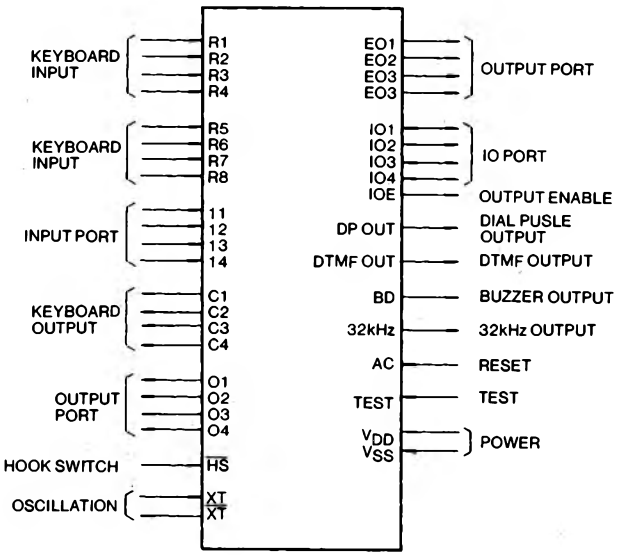
(Top View) 40 Lead Plastic DIP



(Top View) 28 Lead Plastic DIP



LOGIC SYMBOL



PIN DESCRIPTION

Designation	Function
V_{DD}	Pource source
V_{SS}	Circuit ground potential
AC	Terminal to clear internal logic, pulled down to V_{SS}. After power is turned on, the MSM6052 must be reset by this terminal.
TEST	Terminal to test internal logic, pulled down to V_{SS}. This terminal must be open in normal operation.
XT, $\overline{XT}$	Input and output terminals of oscillator inverter. 3.58 MHz ceramic resonator is connected to these terminals.
$\overline{HS}$	Input terminal connected to the hook switch, pulled up tp V_{DD}.
DP OUT	Output terminal of dial pulse. Dial pulse rate (10 pps or 20 pps) and Make Break ratio (40% or 33 %) can be selected by software.
DTMF OUT	Output terminal of DTMF signal
BD	Output terminal of buzzer sound
32 kHz	Output terminal of 32 kHz clock
$R_1 \sim R_4$ $R_5 \sim R_8$	Input port pulled down to V_{SS}.
$I_1 \sim I_4$	Input port having clocked pull-down resistor to V_{SS}. Only when this port is accessed, pull-down resistors are connected to this port.
$C_1 \sim C_4$ $O_1 \sim O_4$	Output port
$IO_1 \sim iO_4$	Tri-state bidirectional port
IOE	Output terminal When $IO_1 \sim IO_4$ is accessed, input completion signal (when read) or load signal (when written) is output from IOE terminal.

FUNCTIONAL DESCRIPTION

A block diagram of the MSM6052 is given on page 129. Each block of logic will be briefly discussed. For more information, please refer to the MSM6052 user's manual.

Program ROM

The MSM6052 will address up to 2 K words of internal mask programmable ROM. Each word consists of 14-bits and all instructions are one word. The instructions are routed to a programmed logic array which generates the signals necessary for control of logic.

Data RAM

Data is organized in 4-bit nibbles. Internal data RAM consists of 640 nibbles.

All locations are addressed by 10-bit address registers (AR_1, AR_2), 2-bit bank register (B), 4-bit page register (P) or a part of the instruction's operand.

ALU

The ALU performs a 4-bit parallel operation on RAM and ACC contents, or on RAM contents and an immediate digit. It sets or resets the three flags (Z, C, G) depending on the condition.

Program Counter (PC)

The program counter is an 11-bit wide counter that specifies the address of program ROM.

The PC is incremented by one at every execution of the instruction, and then specifies the next instruction to be executed. However, the contents of the PC are rewritten by the execution of a Jump, Call or Branch instruction.

As there is no boundary in the ROM, and a Jump, Call or Branch instruction can be put anywhere in the ROM.

Stack

The MSM6052 has a 5 level stack apart from the data RAM. The contents of the PC are loaded into stack when a Call instruction is executed or an interrupt is generated. Nesting of subroutines within subroutines can continue up to 4 times, including the interrupt.

- **Input Port**

Port (R1 ~ R4)

4-bit input port. Each pin of the port is pulled down to V_{SS} by an internal resistor, and status of the port is fetched by an input instruction.

Port (R5 ~ R8)

4-bit input port. Each pin of the port is pulled down to V_{SS} by an internal resistor, and the status of the port is fetched by an input instruction.

Port (I1 ~ I4)

4-bit input port. Each pin of the port is pulled down to V_{SS} by an internal resistor and transistor. Only when it is desired to fetch status of the port, input current flows through these pins. Status of the port is fetched by an input instruction.

- **Output Port**

Port (C1 ~ C4)

4-bit output port. These ports consist of data latches and buffers, and the contents of the data latches are rewritten by an output instruction.

Port (O1 ~ O4)

4-bit output port. This port consists of data latches and buffers, and the contents of the data latches are rewritten by an output instruction.

Electrical characteristics of O3 and O4 are different from those of O1 and O2. O3 and O4 of the ports are used as $\overline{\text{XMIT}}$ $\overline{\text{MUTE}}$ and $\overline{\text{MUTE}}$ normally.

Port (EO1 ~ EO4)

4-bit output port. This port consists of data latches and buffers, and the contents of data latches are rewritten by an output instruction.

- **Input/Output Port**

Port (IO1 ~ IO4)

4-bit bidirectional port. This port consists of data latches, output buffers and input buffers. The contents of the data latches are rewritten by an output instruction, and status of the port is fetched by an input instruction.

Address Registers (AR1, AR2)

The address registers are used to specify the 10-bit address of data RAM, when a data search instruction (RDAR) or block data transfer instruction (MVAR) is executed.

This register is an up/down counter, and is incremented or decremented by 1 with execution of the instruction.

Timing Generator

By connecting a 3.58 MHz ceramic resonator to the XT and $\overline{\text{XT}}$ terminal, the timing generator generates a basic timing signal to control the MSM6052.

The MSM6052 can operate in 2 modes, normal operating mode and power down mode. STOP instruction is used to place the MSM6052 in the power down mode. The oscillation stops and all functions are stopped. However, the contents of RAM and all registers are maintained.

Programmable Timer

The programmable timer consists of a 4-bit down counter and a 1/100 prescaler.

Any of a 7990.1 Hz clock, 1997.5 Hz clock and 998.8 Hz clock is input to the 1/100 prescaler. Output of the 1/100 prescaler decrements the 4-bit down counter by 1.

When the contents of the 4-bit down counter is decremented to 0, the programmable timer generates an interrupt.

This programmable timer can be used as a dial pulse generator. The dial pulse rate (10 pps, 20 pps) and Make/Break ratio (40%, 33%) of the dial pulse which the programmable timer generates are selectable.

DTMF Circuit

DTMF circuit is used to generate a DTMF signal. 12 kinds of DTMF signal (0 to 9, #, *) can be output by an output instruction.

BD Circuit

The BD circuit generates the square wave which can be used as the confirmation sound, warning sound and so on. 15 kinds of sound (4.66 to 0.82 kHz) are output by an output instruction specifying the frequency.

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Conditions	Limits	Unit
Supply Voltage	V_{DD}	Ta = 25°C	−0.3 to 7.0	V
Input Voltage	V_I	Ta = 25°C	−0.3 to V_{DD}+0.3	V
Output Voltage	V_O	Ta = 25°C	−0.3 to V_{DD}+0.3	V
Storage Temperature	Tstg	–	−55 to 125	°C
Power Dissipation	P_D	Ta = 25°C	200	mW

OPERATING CONDITIONS

Parameter	Symbol	Limits	Unit
Operating Voltage	V_{DD}	2.5 to 6.0	V
Memory Retension Voltage	V_{DDM}	1.2 to 6.0	V
Operating Temperature	Topr	−20 to 75	°C

DC CHARACTERISTICS

(V_{DD} = 3V, Ta = −20 to 75°C)

Parameter	Symbol	Conditions		Limits			Unit
				Min.	Typ.	Max.	
"H" Input Voltage	V_{IH}	V_{DD}=3V		2.2	–	–	V
		V_{DD}=6V		4.4	–	–	V
"L" Input Voltage	V_{IL}	V_{DD}=3V		–	–	0.8	V
		V_{DD}=6V		–	–	1.6	V
"H" Output Current (1)	I_{OH_1}	O_3, O_4 DP OUT	V_{OH}=2.6V	−200	–	–	μA
"L" Output Current (1)	I_{OL_1}		V_{OL}=0.4V	500	–	–	μA
"H" Output Current (2)	I_{OH_2}	C_1 ~ C_4	V_{OH}=2.6V	−1	–	–	mA
"L" Output Current (2)	I_{OL_2}		V_{OL}=0.4V	10	–	–	μA
"H" Output Current (3)	I_{OH_3}	O_1, O_2, BD	V_{OH}=2.6V	−20	–	–	μA
"L" Output Current (3)	I_{OL_3}		V_{OL}=0.4V	10	–	–	μA

DC CHARACTERISTICS (CONT.)

Parameter	Symbol	Conditions		Limits			Unit
				Min.	Typ.	Max.	
"H" Output Current (4)	I_{OH4}	$IO_1 \sim IO_4$ IOE $EO_1 \sim EO_4$	V_{OH}= 2.6V	−150	–	–	μA
"L" Output Current (4)	I_{OL4}		V_{OL}= 0.4V	300	–	–	μA
"H" Output Current (5)	I_{OH5}	32 kHz	V_{OH}=2.6V	−40	–	–	μA
"L" Output Current (5)	I_{OL5}		V_{OL}=0.4V	25	–	–	μA
Pull-up Resistance	R_{UP}	$\overline{HS}$		17	–	150	kΩ
Pull down Resistance (1)	Rdwon 1	$R_1 \sim R_8$		33	–	300	kΩ
Pull down Resistance (2)	Rdwon 2	$I_1 \sim I_4$, AC, TEST		10		100	kΩ
Input Leak Current	I_{IL}	$IO_1 \sim IO_4$	$0 \leq V_{IN} \leq V_{DD}$ V_{DD}= 2.5 to 6.0V	–	–	±2	μA
Current Consumption (1)	I_{DDP}	DTMF output off	V_{DD}= 3V	–	0.3	0.6	mA
			V_{DD}= 6V	–	1.2	2.4	mA
Current Consumption (2)	I_{DDT}	DTMF output on	V_{DD}= 3V	–	1.2	2.4	mA
			V_{DD}=6V	–	3.5	7.0	mA
Memory retention Current	I_{DDM}	ON HOOK V_{DD}=2.5V	Ta=25°C	–	0.01	0.2	μA
			Ta=−20to75°C	–	–	2	μA

AC CHARACTERISTICS

(VDD = 3V, Ta = −20 to 75°C)

Parameter	Symbol	Conditions		Limits			Unit
				Min.	Typ.	Max.	
Key Input Time	T_{KIN}	V_{DD}=2.5 to 6.0V		33	–	–	ms
Tone Output Voltage	V_{OUT}	Row only R_L=1 kΩ	V_{DD}=2.5V	150	250	350	mV rms
			V_{DD}=4.0V	200	350	570	
			V_{DD}=6.0V	300	480	850	
High/Low Level Ratio	dB_{CR}	V_{DD}=2.5 to 6.0V		1	2	3	dB
Distortion Ratio	%DIS	R_L=1 kΩ		–	1	5	%
Rise/Fall Time (1)	t_{TLH1}	O_3, O_4, DP OUT C_L=50 pF		–	–	0.5	μS
	t_{THL1}			–	–	0.5	
Rise/Fall Time (2)	t_{TLH2}	$C_1 \sim C_4$ C_L=50 pF		–	–	0.5	μS
	t_{THL2}			–	–	10	
Rise/Fall Time (3)	t_{TLH3}	O_1, O_2, BD, 32 kHz C_L=50 pF		–	–	5	μS
	t_{THL3}			–	–	10	
Rise/Fall Time (4)	t_{TLH4}	$IO_1 \sim IO_4$, IOE, $EO_1 \sim EO_4$ C_L=50 pF		–	–	1	μS
	t_{THL4}			–	–	1	

DESCRIPTION OF INSTRUCTIONS

	Mnemonic	Instruction Code																Operation
		13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Arithmetic and logic	ADD ACC, AP	0	0	0	0	0	P	0	1	0	0	A						AP ← (AP) + ACC
	ADD #D, AP	0	1	1	0	0	P	D				A						AP ← (AP) + D
	ADC AP	0	0	0	0	0	P	0	1	0	1	A						AP ← (AP) + ACC + C
	SUB ACC, AP	0	0	0	0	1	P	0	1	0	0	A						AP ← (AP) − ACC
	SUB #D, AP	0	1	1	0	1	P	D				A						AP ← (AP) − D
	SBC AP	0	0	0	0	1	P	0	1	0	1	A						AP ← (AP) − ACC − C
	CMP ACC, AP	0	0	0	0	1	P	1	1	1	0	A						(AP) − ACC
	CMP #D, AP	0	1	0	1	1	P	D				A						(AP) − D
	XOR ACC, AP	0	0	0	0	0	P	0	1	1	1	A						AP ← (AP) ∀ ACC
	XOR #D, AP	0	1	1	1	1	P	D				A						AP ← (AP) ∀ D
Bit operation	BIT ACC, AP	0	0	0	0	0	P	1	1	1	0	A						(AP) V $\overline{ACC}$
	BIT #D, AP	0	1	0	1	0	P	D				A						(AP) V $\overline{D}$
	BIS ACC, AP	0	0	0	0	0	P	0	1	1	0	A						AP ← (AP) V ACC
	BIS #D, AP	0	1	0	0	0	P	D				A						AP ← (AP) V D
	BIC ACC, AP	0	0	0	0	1	P	0	1	1	0	A						AP ← (AP) Λ $\overline{ACC}$
	BIC #D, AP	0	1	0	0	1	P	D				A						AP ← (AP) Λ $\overline{D}$
Rotate	ROR AP	0	0	0	0	0	P	0	0	1	0	A						(AP) → C
	ROL AP	0	0	0	0	1	P	0	0	1	0	A						(AP) ← C
	ASR AP	0	0	0	0	0	P	0	0	1	1	A						0 → (AP) → C
	ASL AP	0	0	0	0	1	P	0	0	1	1	A						C ← (AP) ← 0
Flag operation	SEZ	0	0	0	0	1	0	1	0	1	0	0	0	0	0			Z ← 1
	CLZ	0	0	0	0	0	0	1	0	1	0	0	0	0	0			Z ← 0
	SEC	0	0	0	0	1	0	1	0	0	1	0	0	0	0			C ← 1
	CLC	0	0	0	0	0	0	1	0	0	1	0	0	0	0			C ← 0
	SEG	0	0	0	0	1	0	1	0	0	0	0	0	0	0			G ← 1
	CLG	0	0	0	0	0	0	1	0	0	0	0	0	0	0			G ← 0
	SEA	0	0	0	0	1	0	1	0	1	1	0	0	0	0			Z ← 1, C ← 1, G ← 1
	CLA	0	0	0	0	0	0	1	0	1	1	0	0	0	0			Z ← 0, C ← 0, G ← 0
Data transfer	MOV ACC, AP	1	1	1	1	0	1	0	0	0	0	A						AP ← ACC
	MOV ACC, AX	1	1	1	1	0	0	X				A						AX ← ACC
	MOV #D, AP	0	1	1	1	0	P	D				A						AP ← D
	MOV AP, ACC	1	1	1	1	1	1	0	0	0	0	A						ACC ← (AP)
	MOV AX, ACC	1	1	1	1	1	0	X				A						ACC ← (AX)
	CHG AP	1	1	1	0	0	1	0	0	0	0	A						(AP) ⟷ ACC

DESCRIPTION OF INSTRUCTIONS (CONT.)

	Mnemonic	Instruction Code														Operation
		13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Data transfer	CHG AX	1	1	1	0	0	0	X				A				(AX) ⟷ ACC
	RDAR	1	1	0	0	0	0	0	0	0	0	0	0	0	0	ACC ← (AR_1)
	RDAR + (−)	1	1	0	0	0	0	0	0	1	D/I	0	0	0	0	ACC ← (AR_1), AR_1 ← $AR_1 \pm 1$
	RDAR + (−), Z	1	1	0	0	0	0	0	1	0	D/I	0	0	0	0	ACC ← (AR_1) if (AR_1)=0 then PC ← PC + 1 else AR_1 ← $AR_1 \pm 1$, repeat
	RDAR + (−), N	1	1	0	0	0	0	1	0	0	D/I	0	0	0	0	ACC ← (AR_1) if (AR_1)≠0 then PC ← PC + 1 else AR_1 ← $AR_1 \pm 1$, repeat
	RDAR + (−), Z, L	1	1	0	0	1	0	0	1	0	D/I	0	0	0	0	ACC ← (AR_1), L ← L − 1 if (AR_1)=0 or L=0 then PC← PC+1 else AR_1 ← $AR_1 \pm 1$, repeat
	RDAR + (−), N, L	1	1	0	0	1	0	1	0	0	D/I	0	0	0	0	ACC ← (AR_1), L ← L − 1 if (AR_1)≠0 or L=0 then PC ← PC + 1 else AR_1 ← $AR_1 \pm 1$, repeat
	MVAR	1	1	0	1	0	0	0	0	0	0	0	0	0	0	AR_2 ← (AR_1)
	MVAR + (−)	1	1	0	1	0	0	0	0	1	D/I	0	0	0	0	AR_2 ← (AR_1), AR_1 ← $AR_1 \pm 1$, AR_2 ← $AR_2 \pm 1$
	MVAR + (−), Z	1	1	0	1	0	0	0	1	0	D/I	0	0	0	0	AR_2 ← (AR_1), if (AR_1)=0 then PC ← PC +1 else AR_1 ← $AR_1 \pm 1$, AR_2 ← $AR_2 \pm 1$, repeat
	MVAR + (−), N	1	1	0	1	0	0	1	0	0	D/I	0	0	0	0	AR_2 ← (AR_1) if (AR_1)≠0 then PC ← PC +1 else AR_1 ← $AR_1 \pm 1$, AR_2 ← $AR_2 \pm 1$, repeat
	MVAR + (−), L	1	1	0	1	1	0	0	0	0	D/I	0	0	0	0	AR_2 ← (AR_1), L ← L − 1 if L=0 then PC ← PC +1 else AR_1 ← $AR_1 \pm 1$, AR_2 ← $AR_2 \pm 1$, repeat
	MVAR + (−), Z, L	1	1	0	1	1	0	0	1	0	D/I	0	0	0	0	AR_2 ← (AR_1), L ← L −1 if (AR_1)=0 or L=0 then PC ← PC +1 else AR_1 ← $AR_1 \pm 1$, AR_2 ← $AR_2 \pm 1$, repeat
	MVAR + (−), N, L	1	1	0	1	1	0	1	0	0	D/I	0	0	0	0	AR_2 ← (AR_1), L ← L −1 if (AR_1)≠0 or L=0 then PC ← PC +1 else AR_1 ← $AR_1 \pm 1$, AR_2 ← $AR_2 \pm 1$, repeat
Sub routine	CALL adrs	1	0	1	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	STACK ← (PC), PC ← adrs
	RET	0	0	0	0	0	0	1	1	0	0	0	0	0	0	PC ← (STACK) + 1
	RTI	0	0	0	0	1	0	1	1	0	0	0	0	0	0	PC ← (STACK) or PC ← (STACK) +1

DESCRIPTION OF INSTRUCTIONS (CONT.)

	Mnemonic	Instruction Code																Operation
		13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Jump	JMP adrs	1	0	0	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0			PC ← adrs
	JMP @AP	0	0	0	0	0	P	1	1	0	1	A						PC ← (PC) + (AP) + 1
	JMPIO @AP	0	0	0	0	1	P	1	1	0	1	A						PC ← (PC) + {(AP) ∧ 7H} +1
Branch	BEQ n (BZE n)	1	1	1	0	1	P	0	1	0	n_4	n_3	n_2	n_1	n_0			if Z=1 then PC←PC − n or PC ← PC + n +1 else ← PC ← PC + 1
	BNE n (BNZ n)	1	1	1	0	1	P	1	1	0	n_4	n_3	n_2	n_1	n_0			if Z=0 then PC←PC − n or PC ← PC + n + 1 else PC ← PC +1
	BCS n	1	1	1	0	1	P	0	0	0	n_4	n_3	n_2	n_1	n_0			if C=1 then PC ← PC −n or PC ← PC + n + 1 else PC ← PC +1
	BCC n	1	1	1	0	1	P	1	0	0	n_4	n_3	n_2	n_1	n_0			if C=0 then PC ← PC − n or PC ← PC+n+1 else PC ← PC +1
	BGT n	1	1	1	0	1	P	0	0	1	n_4	n_3	n_2	n_1	n_0			if G=1 then PC ← PC − n or PC ← PC + n + 1 else PC ← PC +1
	BLE n	1	1	1	0	1	P	1	0	1	n_4	n_3	n_2	n_1	n_0			if G=0 then PC ← PC − n or PC ← PC +n+1 else PC ← PC +1
	BGE n	1	1	1	0	1	P	0	1	1	n_4	n_3	n_2	n_1	n_0			if G=1 or Z=1 then PC ← PC − n or PC ← PC+n+1, else PC ← PC +1
	BLT n	1	1	1	0	1	P	1	1	1	n_4	n_3	n_2	n_1	n_0			if G=0 and Z=0 then PC ← PC −n or PC ← PC+n+1 else PC ← PC+1
Input/ Output	IN PORT, AP	0	0	0	1	0	P	P_L				A						AP ← (PORT)
	OUT AP, PORT	0	0	1	0	P_H	P	P_L				A						PORT ← (AP)
	OUT #D, PORT	0	0	1	1	P_H	0	P_L				D						PORT ← D
CPU control and others	STOP	0	0	1	1	1	0	0	0	0	0	0	0	0	0			Stop system clock
	HALT	0	0	1	1	1	0	0	0	0	1	0	0	0	0			Halt CPU
	ACT	0	0	1	1	1	0	0	0	1	0	0	0	0	0			Activate CPU
	EI	0	0	1	1	1	0	0	1	1	0	1	0	0	0			Enable timer interrupt
	DI	0	0	1	1	1	0	0	1	1	0	0	1	0	0			Disable timer interrupt
	ET	0	0	1	1	1	0	0	1	1	0	0	0	1	0			Enable timer activate
	DT	0	0	1	1	1	0	0	1	1	0	0	0	0	1			Disable timer activate
	EC	0	0	1	1	1	0	0	1	1	1	1	0	0	0			Enable output port (C_1~C_4)
	DC	0	0	1	1	1	0	0	1	1	1	0	1	0	0			Disable output port (C_1~C_4)

DESCRIPTION OF INSTRUCTIONS (CONT.)

	Mnemonic	Instruction Code														Operation
		13	12	11	10	9	8	7	6	5	4	3	2	1	0	
CPU control and others	OM	0	0	1	1	1	0	0	1	1	1	0	0	1	0	Set I/O port (IO_1~IO_4) to output mode
	IM	0	0	1	1	1	0	0	1	1	1	0	0	0	1	Set I/O port (IO_1~IO_4) to input mode
	RST	0	0	1	1	1	0	1	0	0	1	0	0	0	0	Reset divider
	NOP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	No operation