8222K

SANYO

No.1095A

LC6599

C MOS LSI

EVALUATION CHIP FOR APPLICATION DEVELOPMENT

General Description

The LC6599 is an evaluation chip for CMOS single-chip 4-bit microcomputer LC6500 series. (SANYO Original)
The LC6599 is a CPU that contains all the functions (ROM: Connected externally) of the LC6500 series plus the functions, such as break, step functions and built-in RAM capacity control function, required for an evaluation chip. It is used for program debug, hardware simulation when performing the application development of the LC6500 series microcomputers.

Features of LC6599

(1) Capable of performing hardware simulation for all Type Nos. of the LC6500 series.
(2) Provides the instruction set covering all instructions used by the LC6500 series.
(3) Program memory of 4096 x 8 bits max. is connectable externally.
(4) Built-in data RAM capacity can be set as required for the Type No. to be evaluated.
(5) Abundant evaluation functions
- Break function to stop execution of program
- Single-step program execution function
- Capable of outputting the contents of internal registers (AC, PC, DP) to pins

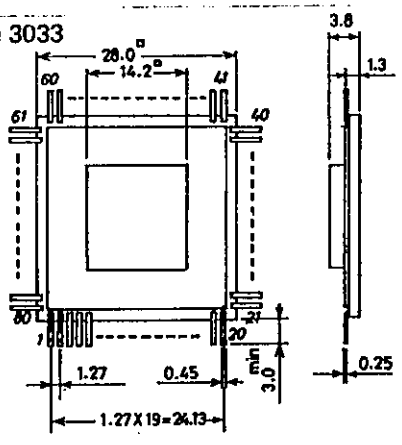
(6) Has built-in clock generator (C, R or ceramic resonator: Connected externally)
(7) +5 V single power supply
(8) 80-pin flat QIP

Note: The LC6599 is designed for application development tool of the LC6500 series microcomputers. It should be noted that it is not suited for use under high-temperature, high-humidity conditions.

LC6500 series and LC6599

Type No.	ROM	RAM	Package	Remarks
LC6502C	2048 bytes	128 x 4 bits	DIP-42 DIP-42S QIP-64	
LC6505C	1024 bytes	64 x 4 bits	DIP-42 DIP-42S QIP-64	
LC6599	External 4096 bytes max.	Built-in 192 x 4 bits max.	QIP-80	Evaluation chip

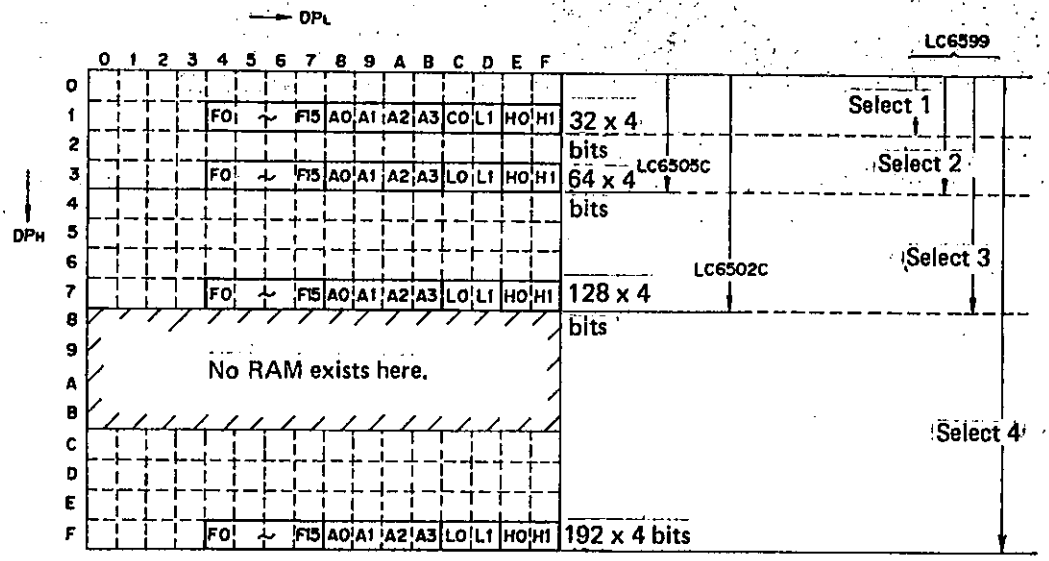
Case Outline 3033
(unit: mm)



These specifications are subject to change without notice.

TOKYO SANYO ELECTRIC CO., LTD. SEMICONDUCTOR DIVISION
15-13, 6-CHOME, SOTOKANDA, CHIYODA-KU, TOKYO 101 JAPAN

Built-in RAM capacity control function



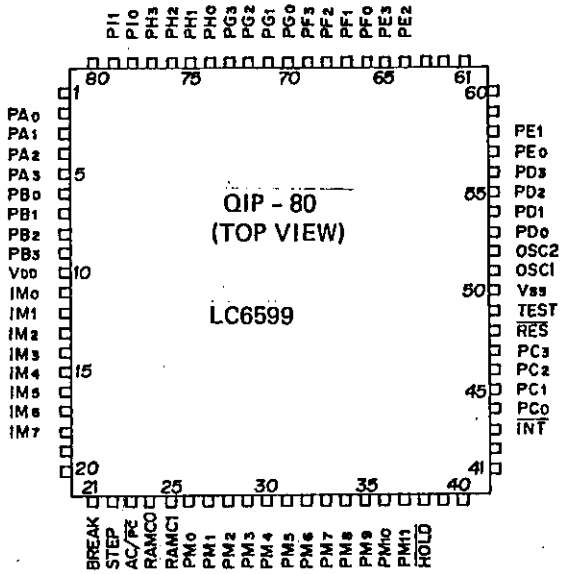
RAM capacity and locations of flags, working registers by Type Nos.

How to set built-in RAM capacity

RAM Select	RAM Control Pins		RAM capacity
	RAMC1	RAMC0	
Select 1	1	1	32 x 4 bits
Select 2	1	0	64 x 4 bits
Select 3	0	1	128 x 4 bits
Select 4	0	0	192 x 4 bits

Note) 0: "L" level input
1: "H" level input

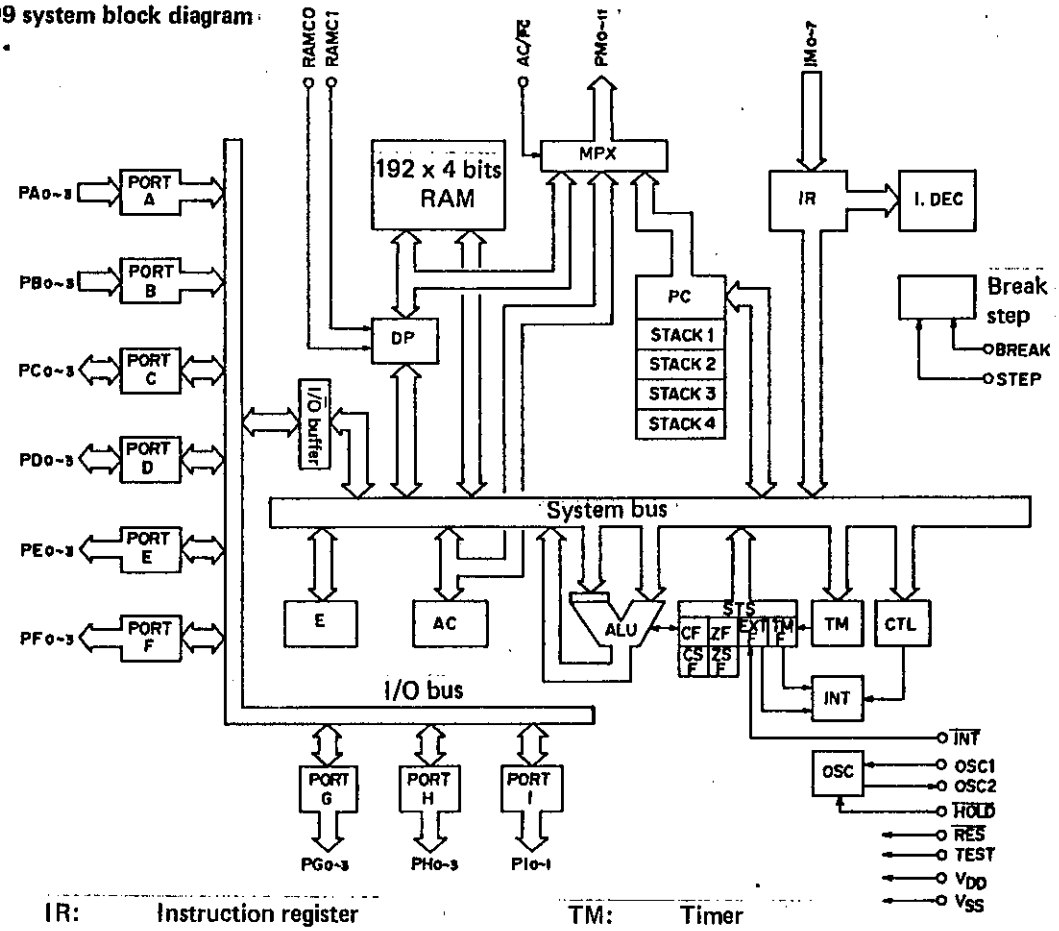
Pin assignment and functions



Pin Description

Pin name	Input/output	Functions
IM_{0-7}	Input	Instruction outputted from external program memory is inputted.
PM_{0-11}	Output	PC contents are outputted during program execution. In break mode, PC contents and AC, DP contents are outputted at AC/$\overline{PC}$ = "L" and "H" respectively.
AC/$\overline{PC}$	Input	Output contents at PM_{0-11} are changed-over in break mode.
BREAK	Input	Program is executed at "L" level. Break mode occurs at "H" level.
STEP	Input	Each time "H" level is applied in break mode, program is executed by one step.
RAMC0	Input	Built-in RAM capacity/organization are controlled.
RAMC1	Input	
$\overline{INT}$	Input	Interrupt request input
$\overline{HOLD}$	Input	Hold mode request input
$\overline{RES}$	Input	Reset input
PA_{0-3}	Input	Input ports $A_0 - 3$. Halt mode release input
PB_{0-3}	Input	Input ports $B_0 - 3$
PC_{0-3}	Input/output	Input/output common ports $C_0 - 3$
PD_{0-3}	Input/output	Input/output common ports $D_0 - 3$
PE_{0-3}	Output	Output ports $E_0 - 3$
PF_{0-3}	Output	Output ports $F_0 - 3$
PG_{0-3}	Output	Output ports $G_0 - 3$
PH_{0-3}	Output	Output ports $H_0 - 3$
PI_{0-1}	Output	Output ports $I_0 - 1$
OSC1	Input	External clock is inputted to this pin. / Pins for externally connecting ceramic resonator, CR for internal clock generation
OSC2	Output	
TEST	Input	LC6599 test pin. Normally, connected to V_{SS} (0 V)
V_{DD}	Input	+5 V / Power supply pin
V_{SS}	–	0 V

LC6599 system block diagram



IR:	Instruction register	TM:	Timer
I · DEC:	Instruction decoder	CTL:	Control register
PC:	Program counter	INT:	Interrupt control
MPX:	PM output multiplexer	OSC:	Oscillator
DP:	Data pointer	STS:	Status register
AC:	Accumulator	E:	E register
ALU:	Arithmetic and logic unit		

The LC6599 is designed for application development tool of the LC6500 series microcomputers. It should be noted that it is not suited for use under high-temperature, high-humidity conditions.
The specifications are subject to change without notice.

Absolute Maximum Ratings/T_a = 25°C, V_{SS} = 0 V			min	typ	max	unit	Remarks
Supply voltage	V_{DD}		−0.3		+7	V	
Input voltage	V_{IN}		−0.3		V_{DD}+0.3	V	Note 1
Output voltage	V_{OUT}		−0.3		V_{DD}+0.3	V	
Peak output current	I_O (1)		−2.0		+2.0	mA	Each pin of C-I ports
	I_O (2)		−26		+26	mA	All pins of C-I ports
	I_O (3)		−0.2		+0.2	mA	Each pin of PM0 ~ 11
	I_O (4)		−2.4		+2.4	mA	All pins of PM0 ~ 11
Power dissipation	P_d max	$T_a \leqq 50°C$			350	mW	
Operating temperature	T_{opg}		0		+50	°C	
Storage temperature	T_{stg}		−55		+125	°C	

Note1: For OSC1 pin, up to oscillation amplitude is allowable which appears when causing internal oscillation under the recommended conditions at the recommended circuit in Fig. 3.

Recommended Operating Conditions/T_a = 25°C, V_{SS} = 0 V			min	typ	max	unit	Remarks
Recommended operating supply voltage	V_{DD} (1)		4.5	5.0	5.5	V	
Power-down supply voltage	V_{DD} (2)	$\overline{HOLD}$ = V_{IL} (3)	2.0		5.5	V	At HOLD mode
"H" level input voltage	V_{IH} (1)	V_{DD} = 5 V±10 %	0.7V_{DD}		V_{DD}	V	Input pins other than $\overline{INT}$, $\overline{RES}$, $\overline{HOLD}$, OSC1 (Note 2)
	V_{IH} (2)	V_{DD} = 5 V±10 %	0.75V_{DD}		V_{DD}	V	$\overline{INT}$, $\overline{RES}$, $\overline{HOLD}$, OSC1 pins
"L" level input voltage	V_{IL} (1)	V_{DD} = 5 V±10 %	V_{SS}		0.3 V_{DD}	V	Input pins other than $\overline{INT}$, $\overline{RES}$, , $\overline{HOLD}$, OSC1
	V_{IL} (2)	V_{DD} = 5 V±10 %	V_{SS}		0.25 V_{DD}	V	$\overline{INT}$, $\overline{RES}$, OSC1 pins
	V_{IL} (3)	V_{DD} = 2 ~ 5.5 V	V_{SS}		0.3V_{DD}−0.3	V	$\overline{HOLD}$ pin
	V_{IL} (4)	V_{DD} = 5 V±10 %	V_{SS}		0.3 V_{DD}	V	TEST pin
Operating clock frequency	f_{extosc}	For external clock V_{DD} = 5 V±10 %	20		420	kHz	Input at OSC1 pin Refer to Fig. 1
"H" level clock pulse width	$t_{w\phi H}$	do.	0.5			μs	do.
"L" level clock pulse width	$t_{w\phi L}$	do.	0.5			μs	do.
Clock input rise time	t_{oscR}	do.			0.2	μs	do.
Clock input fall time	t_{oscF}	do.			0.2	μs	do.
External capacitor value for CR oscillation	C_{ext}			33±5 %		pF	Refer to Fig. 2
External resistance value for CR oscillation	R_{ext}			100±1 %		kΩ	
External capacitance value for ceramic oscillation	C1	CSB400P		33±10 %		pF	Refer to Fig. 3.
		KBR400B		33±10 %		pF	
	C2	CSB400P		33±10 %		pF	
		KBR400B		33±10 %		pF	
External resistance value for ceramic oscillation	R1			4700±5 %		kΩ	
	R2			3.9±5 %		kΩ	
Power-down $\overline{HOLD}$ setup time	t_{HOLDS}			2		ms	Refer to Fig. 6.
Power-down $\overline{HOLD}$ hold time	t_{HOLDH}			2		ms	

Note 2: TEST pin is for LSI test only. It is not included in "input pin" unless otherwise specified.

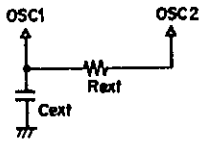


Fig. 2 Recommended oscillator circuit for CR oscillation

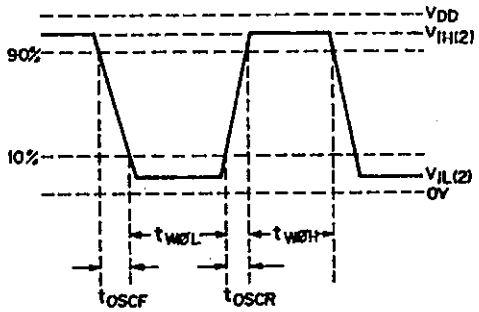
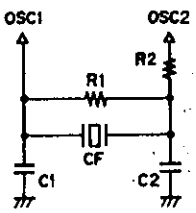


Fig. 1 Input waveform at OSC1 pin



CF : Ceramic resonator
CSB400P (Murata)
KBR400B (Kyocera)

Fig. 3 Recommended oscillator circuit for ceramic oscillation

Electrical Characteristics/T_a = 25°C, V_{DD} = 5 V±10 %, V_{SS} = 0 V			min	typ	max	unit	Remarks
"H" level input current	I_{IH}	$V_{IN} = V_{DD}$			1	μA	All input pins } each input pin
"L" level input current	I_{IL}	$V_{IN} = V_{SS}$	−1			μA	All input pins } each input pin
"H" level output voltage	V_{OH} (1)	I_{OH} = −1 mA	V_{DD}−2			V	Output pins other than OSC2, PM_0 to PM_{11}
	V_{OH} (2)	I_{OH} = −100 μA	V_{DD}−0.5			V	Output pins other than OSC2, PM_0 to PM_{11}
	V_{OH} (3)	I_{OH} = −100 μA	V_{DD}−2			V	PM_0 to PM_{11} pins
"L" level output voltage	V_{OL} (1)	I_{OL} = 1 mA			0.4	V	Output pins other than OSC2, PM_0 to PM_{11}
	V_{OL} (2)	I_{OL} = 100 μA			0.4	V	PM_0 to PM_{11} pins
Output off leak current	I_{OFF} (1)	$V_{OUT} = V_{DD}$			1	μA	Output pins other than OSC2, PM_0 to PM_{11}
	I_{OFF} (2)	$V_{OUT} = V_{SS}$	−1			μA	Output pins other than OSC2, PM_0 to PM_{11}
Clock oscillation frequency for ceramic oscillation	f_{xosc}	Recommended conditions for ceramic oscillation	384	400	417	kHz	Oscillator in Fig. 3
Clock oscillation frequency for CR oscillation	f_{CRosc}	C_{ext} = 33 pF±5 % R_{ext} = 100 kΩ±1 %	230	300	390	kHz	Oscillator in Fig. 2
Current dissipation	I_{DD} (1)	C_{ext} = 33 pF R_{ext} = 100 kΩ Output pin open Input pin $V_{IN} = V_{DD}$		0.4	1.0	mA	At CR oscillation
	I_{DD} (2)	Recommended condition for ceramic oscillation Output pin open Input pin $V_{IN} = V_{DD}$		0.5	1.0	mA	At ceramic oscillation
	I_{DD} (3)	V_{DD} = 5 V±10 % Test circuit in Fig. 4		0.5	10	μA	At halt mode Refer to Fig. 4.
	I_{DD} (4)	V_{DD} = 5 V±10 % Test circuit in Fig. 5		0.5	10	μA	At hold mode Refer to Fig. 5.
	I_{DD} (5)	V_{DD} = 2 V Test circuit in Fig. 5		0.1	2	μA	Hold mode Power-down Refer to Fig. 5.
Input capacitance	C_{IN}	f = 200 kHz		5		pF	Other than IM_0 to IM_7, PM_0 to PM_{11} pins
Output capacitance	C_{OUT}	f = 200 kHz Output high impedance		10		pF	Other than IM_0 to IM_7, PM_0 to PM_{11} pins
Input/output capacitance	C_{IO}	f = 200 kHz Output high impedance		10		pF	

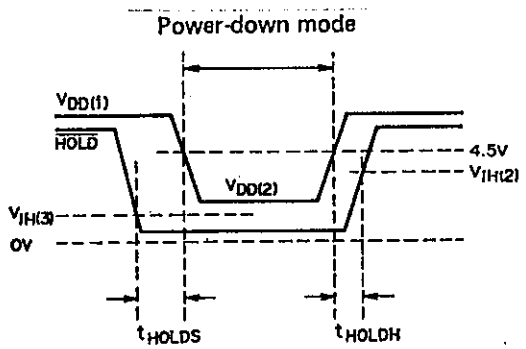


Fig. 6 Power-down mode timing

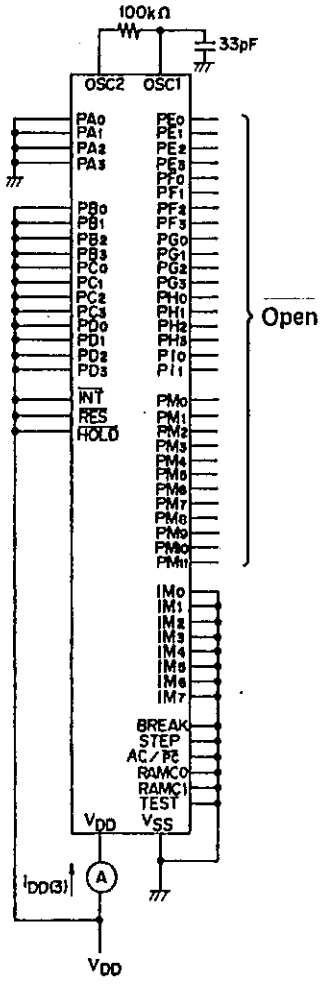


Fig. 4 I_{DD} (3) test circuit

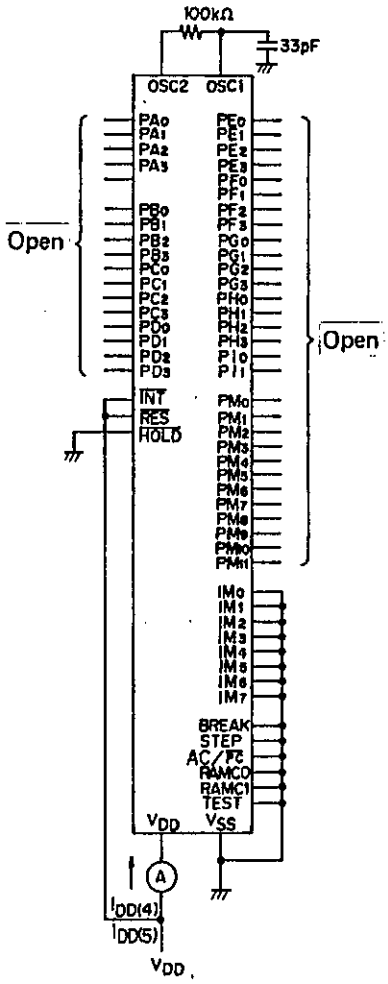
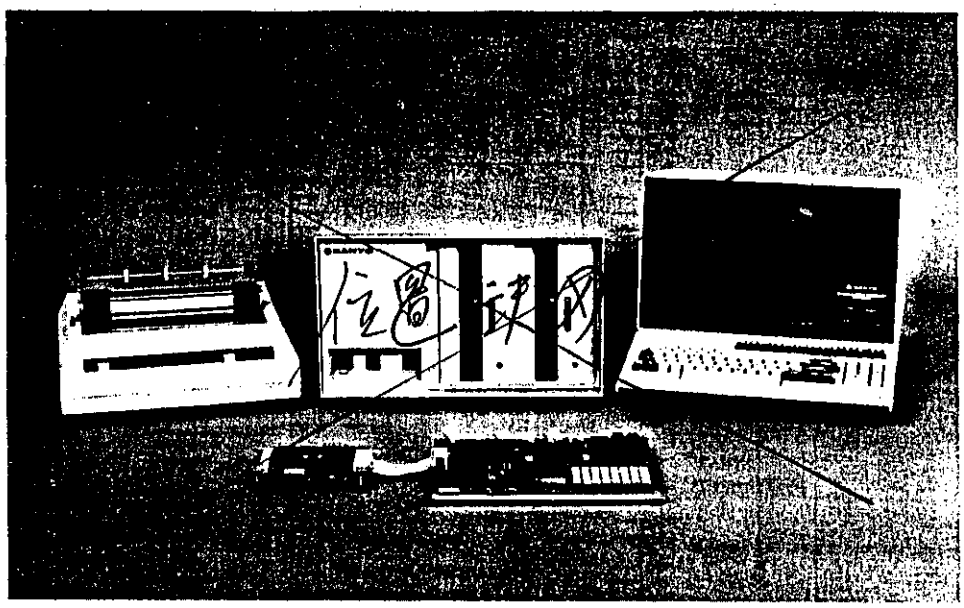


Fig. 5 I_{DD} (4), I_{DD} (5) test circuit

Input/output common ports C, D : Output inhibit
HALT instruction is executed and HALT mode is entered.

Application Development Tools

- SDS-410 System
  This consists of a CPU with two floppy disk units provided, a CRT, and a printer, and makes it possible to speedily and efficiently prepare the application development program of a microcomputer in assembly language.
- EVA-410
  This is an evaluation kit having the EPROM writer function and parallel/serial data communication function that transfers data to and from external equipment (SDS-410, etc.), and makes it possible to correct and debug the application development program on machine language level.
  By replacing the target board, the EVA-410 can be also used for the application development of the N channel microcomputer LM6400 series.
- EVA-402
  This consists of an EPROM and the LC6599 mounted on a board, and is used for checking the operation of a set or making a set by way of trial when the application development program is finished.
- LC65PG99
  This is a simulation chip having a 24-pin socket for EPROM on the package surface. The package is DIP-42 which is pin compatible with the LC6502C/LC6505C. By mounting this chip on the user's application equipment, mounting evaluation can be performed easily. (Under development)



APPENDIX I LC6500 SERIES INSTRUCTION SET (BY FUNCTIONS)

Notations

AC	: Accumulator	M(DP)	: Memory addressed by DP	( ), [ ]	: Contents
ACt	: Accumulator bit t	$P(DP_L)$	: Input/output port addressed by DP_L	←	: Transfer and direction
CF	: Carry flag	PC	: Program counter	+	: Addition
CTL	: Control register	STACK	: Stack register	−	: Subtraction
DP	: Data pointer	TM	: Timer	∧	: AND
E	: E register	TMF	: Timer (internal) interrupt request flag	∨	: Exclusive OR
EXTF	: External interrupt request flag	At, Ha, La	: Working register		
Fn	: Flag bit n	ZF	: Zero flag		
M	: Memory				

Instruction	Mnemonic		Instruction code $D_7D_6D_5D_4$	Instruction code $D_3D_2D_1D_0$	Bytes	Cycles	Function	Description	Status flag affected	Remarks
Accumulator manipulation instructions	CLA	Clear AC	1 1 0 0	0 0 0 0	1	1	AC ← 0	The AC contents are cleared.	ZF	* 1
	CLC	Clear CF	1 1 1 0	0 0 0 1	1	1	CF ← 0	The CF contents are cleared.	CF	
	STC	Set CF	1 1 1 1	0 0 0 1	1	1	CF ← 1	The CF is set.	CF	
	CMA	Complement AC	1 1 1 0	1 0 1 1	1	1	AC ← $\overline{(AC)}$	The AC contents are complemented.	ZF	
	INC	Increment AC	0 0 0 0	1 1 1 0	1	1	AC ← (AC) + 1	The AC contents are incremented +1.	ZF CF	
	DEC	Decrement AC	0 0 0 0	1 1 1 1	1	1	AC ← (AC) − 1	The AC contents are decremented −1.	ZF CF	
	RAL	Rotate AC left through CF	0 0 0 0	0 0 0 1	1	1	$AC_0 \leftarrow (CF), AC_{n+1} \leftarrow (AC_n), CF \leftarrow (AC_3)$	The AC contents are shifted left through the CF.	ZF CF	
	TAE	Transfer AC to E	0 0 0 0	0 0 1 1	1	1	E ← (AC)	The AC contents are transferred to the E.		
	XAE	Exchange AC with E	0 0 0 0	1 1 0 1	1	1	(AC) ⇆ (E)	The AC contents and the E conents are exchanged.		
Memory manipulation instructions	INM	Increment M	0 0 1 0	1 1 1 0	1	1	M(DP) ← (M(DP)) + 1	The M(DP) contents are incremented +1.	ZF CF	
	DEM	Decrement M	0 0 1 0	1 1 1 1	1	1	M(DP) ← (M(DP)) − 1	The M(DP) contents are decremented −1.	ZF CF	
	SMB bit	Set M data bit	0 0 0 0	1 0 B_1B_0	1	1	M(DP, B_1B_0) ← 1	A single bit of the M(DP) specified with B_1B_0 is set.		
	RMB bit	Reset M data bit	0 0 1 0	1 0 B_1B_0	1	1	M(DP, B_1B_0) ← 0	A single bit of the M(DP) specified with B_1B_0 is reset.	ZF	
Arithmetic operation/comparison instructions	AD	Add M to AC	0 1 1 0	0 0 0 0	1	1	AC ← (AC) + [M(DP)]	Binary addition of the AC contents and the M(DP) contents is performed and the result is stored in the AC.	ZF CF	
	ADC	Add M to AC with CF	0 0 1 0	0 0 0 0	1	1	AC ← (AC) + [M(DP)] + (CF)	Binary addition of the AC, CF contents and the M(DP) contents is performed and the result is stored in the AC.	ZF CF	
	DAA	Decimal adjust AC in addition	1 1 1 0	0 1 1 0	1	1	AC ← (AC) + 6	6 is added to the AC contents.	ZF	
	DAS	Decimal adjust AC in subtraction	1 1 1 0	1 0 1 0	1	1	AC ← (AC) + 10	10 is added to the AC contents.	ZF	
	EXL	Exclusive or M to AC	1 1 1 1	0 1 0 1	1	1	AC ← (AC) ∨ [M(DP)]	The AC contents and the M(DP) contents are exclusive-ORed and the result is stored in the AC.	ZF	
	AND	And M to AC	1 1 1 0	0 1 1 1	1	1	AC ← (AC) ∧ [M(DP)]	The AC contents and the M(DP) contents are ANDed and the result is stored in the AC.	ZF	
	OR	Or M to AC	1 1 1 0	0 1 0 1	1	1	AC ← (AC) ∨ [M(DP)]	The AC contents and the M(DP) contents are ORed and the result is stored in the AC.	ZF	
	CM	Compare AC with M	1 1 1 1	1 0 1 1	1	1	$\overline{(M(DP))} + (AC) + 1$	The AC contents and the M(DP) contents are compared and the CF and ZF are set/reset. (Comparison result: [M(DP)] > (AC): CF 0, ZF 0; [M(DP)] = (AC): CF 1, ZF 1; [M(DP)] < (AC): CF 1, ZF 0)	ZF CF	
	CI data	Compare AC with immediate data	0 0 1 0 / 0 1 0 0	1 1 0 0 / $I_3I_2I_1I_0$	2	2	$\overline{I_3I_2I_1I_0} + (AC) + 1$	The AC contents and the immediate data $I_3I_2I_1I_0$ are compared and the ZF and CF are set/reset. (Comparison result: $I_3I_2I_1I_0$ > (AC): CF 0, ZF 0; $I_3I_2I_1I_0$ = (AC): CF 1, ZF 1; $I_3I_2I_1I_0$ < (AC): CF 1, ZF 0)	ZF CF	
	CLI data	Compare DP_L with immediate data	0 0 1 0 / 0 1 0 1	1 1 0 0 / $I_3I_2I_1I_0$	2	2	$(DP_L) \vee I_3I_2I_1I_0$	The DP_L contents and the immediate data $I_3I_2I_1I_0$ are compared.	ZF	
Load/store instructions	LI data	Load AC with immediate data	1 1 0 0	$I_3I_2I_1I_0$	1	1	AC ← $I_3I_2I_1I_0$	The immediate data $I_3I_2I_1I_0$ is loaded in the AC.	ZF	* 1
	S	Store AC to M	0 0 0 0	0 0 1 0	1	1	M(DP) ← (AC)	The AC contents are stored in the M(DP).		
	L	Load AC from M	0 0 1 0	0 0 0 1	1	1	AC ← [M(DP)]	The M(DP) contents are loaded in the AC.	ZF	
	XM data	Exchange AC with M, then modify DP_H with immediate data	1 0 1 0	0 $M_2M_1M_0$	1	2	(AC) ⇆ (M(DP)), $DP_H \leftarrow (DP_H) \vee 0M_2M_1M_0$	The AC contents and the M(DP) contents are exchanged and then the DP_H contents are modified with the contents of $(DP_H) \vee 0M_2M_1M_0$.	ZF	The ZF is set/reset according to the result of $(DP_H) \vee 0M_2M_1M_0$.
	X	Exchange AC with M	1 0 1 0	0 0 0 0	1	2	(AC) ⇆ [M(DP)]	The AC contents and the M(DP) contents are exchanged.	ZF	The ZF is set/reset according to the DP_H contents at the time of instruction execution.
	XI	Exchange AC with M, then increment DP_L	1 1 1 1	1 1 1 0	1	2	(AC) ⇆ [M(DP)], $DP_L \leftarrow (DP_L) + 1$	The AC contents and the M(DP) contents are exchanged and then the DP_L contents are incremented +1.	ZF	The ZF is set/reset according to the result of $(DP_L + 1)$.
	XD	Exchange AC with M, then decrement DP_L	1 1 1 1	1 1 1 1	1	2	(AC) ⇆ [M(DP)], $DP_L \leftarrow (DP_L) - 1$	The AC contents and the M(DP) contents are exchanged and then the DP_L contents are decremented −1.	ZF	The ZF is set/reset according to the result of $(DP_L - 1)$.
	RTBL	Read table data from program ROM	0 1 1 0	0 0 1 1	1	2	AC, E ← ROM (PCh, E, AC)	The contents of ROM addressed by the PC whose low-order 8 bits are replaced with the E and AC contents are loaded in the AC and E.		

Instruction	Mnemonic		Instruction code $D_7 D_6 D_5 D_4$	Instruction code $D_3 D_2 D_1 D_0$	Bytes	Cycles	Function	Description	Status flag affected	Remarks
Data pointer manipulation instructions	LDZ data	Load DP_H with Zero and DP_L with immediate data respectively	1 0 0 0	$I_3 I_2 I_1 I_0$	1	1	$DP_H \leftarrow 0$ $DP_L \leftarrow I_3 I_2 I_1 I_0$	The DP_H and DP_L are loaded with 0 and the immediate data $I_3 I_2 I_1 I_0$ respectively.		
	LHI data	Load DP_H with immediate data	0 1 0 0	$I_3 I_2 I_1 I_0$	1	1	$DP_H \leftarrow I_3 I_2 I_1 I_0$	The DP_H is loaded with the immediate data $I_3 I_2 I_1 I_0$.		
	IND	Increment DP_L	1 1 1 0	1 1 1 0	1	1	$DP_L \leftarrow (DP_L)+1$	The DP_L contents are incremented +1.	ZF	
	DED	Decrement DP_L	1 1 1 0	1 1 1 1	1	1	$DP_L \leftarrow (DP_L)-1$	The DP_L contents are decremented −1.	ZF	
	TAL	Transfer AC to DP_L	1 1 1 1	0 1 1 1	1	1	$DP_L \leftarrow (AC)$	The AC contents are transferred to the DP_L.		
	TLA	Transfer DP_L to AC	1 1 1 0	1 0 0 1	1	1	$AC \leftarrow (DP_L)$	The DP_L contents are transferred to the AC.	ZF	
	XAH	Exchange AC with DP_H	0 0 1 0	0 0 1 1	1	1	$(AC) \rightleftarrows (DP_H)$	The AC contents and the DP_H contents are exchanged.		
Working register manipulation instructions	XAt XA0 XA1 XA2 XA3	Exchange AC with working register At	1 1 1 0	t: t_1 to 0 0 0 0	1	1	$(AC) \rightleftarrows (A0)$	The AC contents and the contents of working register At are exchanged. At is assigned one of A_0, A_1, A_2, A_3 according to $t_1 t_0$.		
			1 1 1 0	0 1 0 0	1	1	$(AC) \rightleftarrows (A1)$			
			1 1 1 0	1 0 0 0	1	1	$(AC) \rightleftarrows (A2)$			
			1 1 1 0	1 1 0 0	1	1	$(AC) \rightleftarrows (A3)$			
	XHa XH0 XH1	Exchange DP_H with working register Ha	1 1 1 1	a: 1 0 0 0	1	1	$(DP_H) \rightleftarrows (H0)$	The DP_H contents and the contents of working register Ha are exchanged. Ha is assigned either of H0 or H1 according to a.		
			1 1 1 1	1 1 0 0	1	1	$(DP_H) \rightleftarrows (H1)$			
	XLa XL0 XL1	Exchange DP_L with working register La	1 1 1 1	a: 0 0 0 0	1	1	$(DP_L) \rightleftarrows (L0)$	The DP_L contents and the contents of working register La are exchanged. La is assigned either of L0 or L1 according to a.		
			1 1 1 1	0 1 0 0	1	1	$(DP_L) \rightleftarrows (L1)$			
Flag manipulation instructions	SFB flag	Set flag bit	0 1 0 1	$B_3 B_2 B_1 B_0$	1	1	$Fn \leftarrow 1$	The flag specified with $B_3 B_2 B_1 B_0$ is set.		
	RFB flag	Reset flag bit	0 0 0 1	$B_3 B_2 B_1 B_0$	1	1	$Fn \leftarrow 0$	The flag specified with $B_3 B_2 B_1 B_0$ is reset.	ZF	The flags are divided into 4 groups of F_0 to F_3, F_4 to F_7, F_8 to F_{11}, F_{12} to F_{15}. The ZF is set/reset according to the 4 bits including a single bit specified with the immediate data $B_3 B_2 B_1 B_0$.
Jump/subroutine instructions	JMP addr	Jump in the current bank	0 1 1 0 $P_7 P_6 P_5 P_4$	1 $P_{10} P_9 P_8$ $P_3 P_2 P_1 P_0$	2	2	$PC \leftarrow PC_{11}$ (又は $\overline{PC_{11}}$), $P_{10} P_9 P_8 P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0$	A jump to the address specified with the PC_{11} (or $\overline{PC_{11}}$) and immediate data $P_{10} P_9 P_8 P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0$ occurs.		If the BANK and JMP instructions are executed consecutively, $\overline{PC_{11}} \longrightarrow PC_{11}$.
	JPEA	Jump in the current page modified by E and AC	1 1 1 1	1 0 1 0	1	1	$PC_{7 \sim 0} \leftarrow (E, AC)$	A jump to the address specified with the contents of the PC whose low-order 8 bits are replaced by the E and AC contents occurs.		
	CZP addr	Call subroutine in the zero page	1 0 1 1	$P_3 P_2 P_1 P_0$	1	1	$STACK \leftarrow (PC)+1$ $PC_{11 \sim 6}, PC_{1 \sim 0} \leftarrow 0$ $PC_{5 \sim 2} \leftarrow P_3 P_2 P_1 P_0$	A subroutine in page 0 of bank 0 is called.		
	*CAL addr	Call subroutine in the zero bank	1 0 1 0 $P_7 P_6 P_5 P_4$	1 $P_{10} P_9 P_8$ $P_3 P_2 P_1 P_0$	2	2	$STACK \leftarrow (PC)+2$ $PC_{11 \sim 0} \leftarrow 0 P_{10} P_9 P_8 P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0$	A subroutine in bank 0 is called.		
	RT	Return from subroutine	0 1 1 0	0 0 1 0	1	1	$PC \leftarrow (STACK)$	A return from a subroutine occurs.		
	RTI	Return from interrupt routine	0 0 1 0	0 0 1 0	1	1	$PC \leftarrow (STACK)$ $CF\ ZF \leftarrow CSF, ZSF$	A return from an interrupt service routine occurs.	ZF CF	
	BANK	Change bank	1 1 1 1	1 1 0 1	1	1	$PC_{11} \leftarrow \overline{(PC_{11})}$	The bank is changed.		Effective only when used immediately before the JMP instruction.
Branch instructions	BAt addr	Branch on AC bit	0 1 1 1 $P_7 P_6 P_5 P_4$	0 0 $t_1 t_0$ $P_3 P_2 P_1 P_0$	2	2	$PC_{7 \sim 0} \leftarrow P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0$ if $AC_t = 1$	If a single bit of the AC specified with the immediate data $t_1 t_0$ is 1, a branch to the address specified with the immediate data $P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0$ within the same page occurs.		Mnemonic is BA0 to BA3 according to the value of t.
	BNAt addr	Branch on no AC bit	0 0 1 1 $P_7 P_6 P_5 P_4$	0 0 $t_1 t_0$ $P_3 P_2 P_1 P_0$	2	2	$PC_{7 \sim 0} \leftarrow P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0$ if $AC_t = 0$	If a single bit of the AC specified with the immediate data $t_1 t_0$ is 0, a branch to the address specified with the immediate data $P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0$ within the same page occurs.		Mnemonic is BNA0 to BNA3 according to the value of t.
	BMt addr	Branch on M bit	0 1 1 1 $P_7 P_6 P_5 P_4$	0 1 $t_1 t_0$ $P_3 P_2 P_1 P_0$	2	2	$PC_{7 \sim 0} \leftarrow P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0$ if $(M(DP, t_1 t_0)) = 1$	If a single bit of the M(DP) specified with the immediate data $t_1 t_0$ is 1, a branch to the address specified with the immediate data $P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0$ within the same page occurs.		Mnemonic is BM0 to BM3 according to the value of t.
	BNMt addr	Branch on no M bit	0 0 1 1 $P_7 P_6 P_5 P_4$	0 1 $t_1 t_0$ $P_3 P_2 P_1 P_0$	2	2	$PC_{7 \sim 0} \leftarrow P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0$ if $(M(DP, t_1 t_0)) = 0$	If a single bit of the M(DP) specified with the immediate data $t_1 t_0$ is 0, a branch to the address specified with the immediate data $P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0$ within the same page occurs.		Mnemonic is BNM0 to BNM3 according to the value of t.
	BPt addr	Branch on Port bit	0 1 1 1 $P_7 P_6 P_5 P_4$	1 0 $t_1 t_0$ $P_3 P_2 P_1 P_0$	2	2	$PC_{7 \sim 0} \leftarrow P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0$ if $(P(DP_L, t_1 t_0)) = 1$	If a single bit of port $P(DP_L)$ specified with the immediate data $t_1 t_0$ is 1, a branch to the address specified with the immediate data $P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0$ within the same page occurs.		Mnemonic is BP0 to BP3 according to the value of t.
	BNPt addr	Branch on no Port bit	0 0 1 1 $P_7 P_6 P_5 P_4$	1 0 $t_1 t_0$ $P_3 P_2 P_1 P_0$	2	2	$PC_{7 \sim 0} \leftarrow P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0$ if $(P(DP_L, t_1 t_0)) = 0$	If a single bit of port $P(DP_L)$ specified with the immediate data $t_1 t_0$ is 0, a branch to the address specified with the immediate data $P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0$ within the same page occurs.		Mnemonic is BNP0 to BNP3 according to the value of t.
	BTM addr	Branch on timer	0 1 1 1 $P_7 P_6 P_5 P_4$	1 1 0 0 $P_3 P_2 P_1 P_0$	2	2	$PC_{7 \sim 0} \leftarrow P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0$ if TMF = 1 then TMF ← 0	If the TMF is 1, a branch to the address specified with the immediate data $P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0$ within the same page occurs. The TMF is reset.	TMF	

Instruction	Mnemonic		Instruction code		Bytes	Cycles	Function	Description	Status flag affected	Remarks
			$D_7\ D_6\ D_5\ D_4$	$D_3\ D_2\ D_1\ D_0$						
Branch instructions	BNTM addr	Branch on no timer	0 0 1 1 $P_7\ P_6\ P_5\ P_4$	1 1 0 0 $P_3\ P_2\ P_1\ P_0$	2	2	$PC_{7\sim0} \leftarrow P_7P_6P_5P_4P_3P_2P_1P_0$ if TMF = 0 then TMF ← 0	If the TMF is 0, a branch to the address specified with the immediate data $P_7P_6P_5P_4P_3P_2P_1P_0$ within the same page occurs. The TMF is reset.	TMF	
	BI addr	Branch on interrupt	0 1 1 1 $P_7\ P_6\ P_5\ P_4$	1 1 0 1 $P_3\ P_2\ P_1\ P_0$	2	2	$PC_{7\sim0} \leftarrow P_7P_6P_5P_4P_3P_2P_1P_0$ if EXTF = 1 then EXTF ← 0	If the EXTF is 1, a branch to the address specified with the immediate data $P_7P_6P_5P_4P_3P_2P_1P_0$ within the same page occurs. The EXTF is reset.	EXTF	
	BNI addr	Branch on no interrupt	0 0 1 1 $P_7\ P_6\ P_5\ P_4$	1 1 0 1 $P_3\ P_2\ P_1\ P_0$	2	2	$PC_{7\sim0} \leftarrow P_7P_6P_5P_4P_3P_2P_1P_0$ if EXTF = 0 then EXTF ← 0	If the EXTF is 0, a branch to the address specified with the immediate data $P_7P_6P_5P_4P_3P_2P_1P_0$ within the same page occurs. The EXTF is reset.	EXTF	
	BC addr	Branch on CF	0 1 1 1 $P_7\ P_6\ P_5\ P_4$	1 1 1 1 $P_3\ P_2\ P_1\ P_0$	2	2	$PC_{7\sim0} \leftarrow P_7P_6P_5P_4P_3P_2P_1P_0$ if CF = 1	If the CF is 1, a branch to the address specified with the immediate data $P_7P_6P_5P_4P_3P_2P_1P_0$ within the same page occurs.		
	BNC addr	Branch on no CF	0 0 1 1 $P_7\ P_6\ P_5\ P_4$	1 1 1 1 $P_3\ P_2\ P_1\ P_0$	2	2	$PC_{7\sim0} \leftarrow P_7P_6P_5P_4P_3P_2P_1P_0$ if CF = 0	If the CF is 0, a branch to the address specified with the immediate data $P_7P_6P_5P_4P_3P_2P_1P_0$ within the same page occurs.		
	BZ addr	Branch on ZF	0 1 1 1 $P_7\ P_6\ P_5\ P_4$	1 1 1 0 $P_3\ P_2\ P_1\ P_0$	2	2	$PC_{7\sim0} \sim P_7P_6P_5P_4P_3P_2P_1P_0$ if ZF = 1	If the ZF is 1, a branch to the address specified with the immediate data $P_7P_6P_5P_4P_3P_2P_1P_0$ within the same page occurs.		
	BNZ addr	Branch on no ZF	0 0 1 1 $P_7\ P_6\ P_5\ P_4$	1 1 1 0 $P_3\ P_2\ P_1\ P_0$	2	2	$PC_{7\sim0} \leftarrow P_7P_6P_5P_4P_3P_2P_1P_0$ if ZF = 0	If the ZF is 0, a branch to the address specified with the immediate data $P_7P_6P_5P_4P_3P_2P_1P_0$ within the same page occurs.		
	BFn addr	Branch on flag bit	1 1 0 1 $P_7\ P_6\ P_5\ P_4$	$n_3n_2n_1n_0$ $P_3\ P_2\ P_1\ P_0$	2	2	$PC_{7\sim0} \leftarrow P_7P_6P_5P_4P_3P_2P_1P_0$ if Fn = 1	If the flag bit of the 16 flags specified with the immediate data $n_3n_2n_1n_0$ is 1, a branch to the address specified with the immediate data $P_7P_6P_5P_4P_3P_2P_1P_0$ within the same page occurs.		Mnemonic is BF0 to BF15 according to the value of n.
	BNFn addr	Branch on no flag bit	1 0 0 1 $P_7\ P_6\ P_5\ P_4$	$n_3n_2n_1n_0$ $P_3\ P_2\ P_1\ P_0$	2	2	$PC_{7\sim0} \leftarrow P_7P_6P_5P_4P_3P_2P_1P_0$ if Fn = 0	If the flag bit of the 16 flags specified with the immediate data $n_3n_2n_1n_0$ is 0, a branch to the address specified with the immediate data $P_7P_6P_5P_4P_3P_2P_1P_0$ within the same page occurs.		Mnemonic is BNF0 to BNF15 according to the value of n.
Input/Output instructions	IP	Input port to AC	0 0 0 0	1 1 0 0	1	1	AC ← (P(DPL))	Port P(DPL) contents are loaded in the AC.	ZF	
	OP	Output AC to port	0 1 1 0	0 0 0 1	1	1	P(DPL) ← (AC)	The AC contents are outputted to port P(DPL).		
	SPB bit	Set port bit	0 0 0 0	0 1 B_1 B_0	1	2	P(DPL, B_1B_0) ← 1	A single bit in port P(DPL) specified with the immediate data B_1B_0 is set.		When this instruction is executed, the E contents are destroyed.
	RPB bit	Reset port bit	0 0 1 0	0 1 B_1 B_0	1	2	P(DPL, B_1B_0) ← 0	A single bit in port P(DPL) specified with the immediate data B_1B_0 is reset.	ZF	When this instruction is executed, the E contents are destroyed.
Other instructions	SCTL bit	Set control register bit(S)	0 0 1 0 1 0 0 0	1 1 0 0 $B_3\ B_2\ B_1\ B_0$	2	2	CTL ← (CTL) V $B_3B_2B_1B_0$	The bits of the control register specified with the immediate data $B_3B_2B_1B_0$ are set.		
	RCTL bit	Reset control register bit(S)	0 0 1 0 1 0 0 1	1 1 0 0 $B_3\ B_2\ B_1\ B_0$	2	2	CTL ← (CTL) Λ $\overline{B_3B_2B_1B_0}$	The bits of the control register specified with the immediate data $B_3B_2B_1B_0$ are reset.	ZF	
	WTTM	Write timer	1 1 1 1	1 0 0 1	1	1	TM ← (E), (AC) TMF ← 0	The E and AC contents are loaded in the timer. The TMF is reset.	TMF	
	HALT	Halt	1 1 1 1	0 1 1 0	1	1	Halt	All operations stop.		Only when all pins of port PA are set at L, stop.
	NOP	No operation	0 0 0 0	0 0 0 0	1	1	No operation	No operation is performed, but 1 machine cycle is consumed.		

*1 If the CLA instruction is used continuously in such a manner as CLA, CLA, ———, the first CLA instruction only is effective and the following CLA instructions are changed to the NOP instructions. This is also true of the LI instruction.