

**SANYO**

No. 3933A

**LC66556A, LC66558A****4-bit Microcontrollers with Built-in ROM**

## OVERVIEW

The LC66556A and LC66558A are 4-bit microcontrollers with built-in 6 and 8 Kbyte ROMs, respectively. They incorporate RAM, input/output ports, a serial interface, a comparator and timers in a single chip.

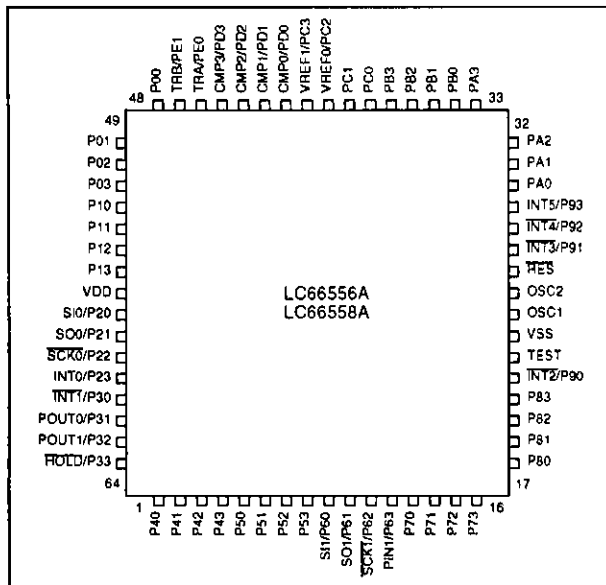
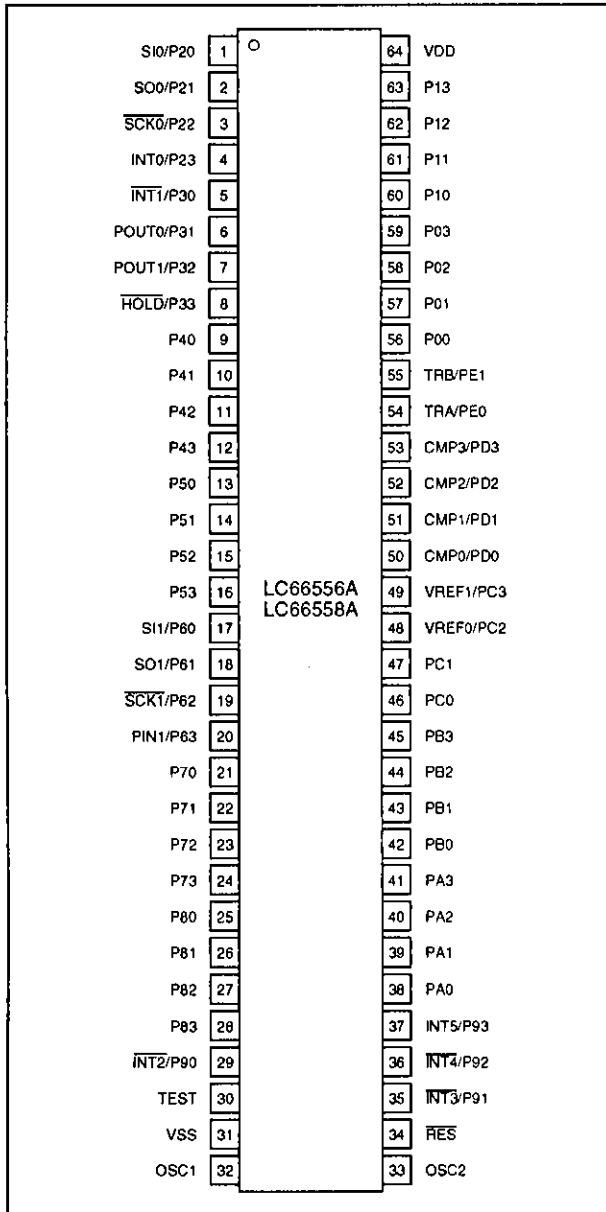
The LC66556A and LC66558A feature a large instruction set compatible with that of the LC66000 series devices. They are functionally identical to the LC66506B, LC66508B, LC66E516 and LC66P516, but have a different supply voltage range and hold-mode release time.

The LC66556A and LC66558A operate from a 5 V supply and are available in 64-pin DIPs and 64-pin QIPs.

## FEATURES

- 6 or 8 Kbyte ROM and 512-word, 4-bit RAM
- Instruction set compatible with the LC66000 series
- 8-bit serial interface which supports 16-bit cascade connection
- 1.96  $\mu$ s minimum cycle time at 3.0 to 5.5 V, and 3.92  $\mu$ s, at 2.2 to 5.5 V
- 12-bit timer for timeout function, event counter, pulse measurement and rectangular waveform generation
- 8-bit timer for timeout function, event counter, pulsewidth modulated output and rectangular waveform generation
- 12-bit pre-scaler for timebase function
- Six external interrupt inputs
- Five internal interrupt sources (two for the timers, two for the serial I/O and one for the pre-scaler)
- 20 mA driver outputs with 15 V withstand voltage
- Ternary-level and comparator inputs
- I/O pull-up resistor and open-drain options
- Runaway detection option
- Halt and hold modes for program-controlled power-down
- LC66599 evaluation chip, EVA850/800-TB665XX debugger, LC66E516 EPROM and LC66P516 PROM development tools available
- 5 V supply
- 64-pin DIP and 64-pin QIP

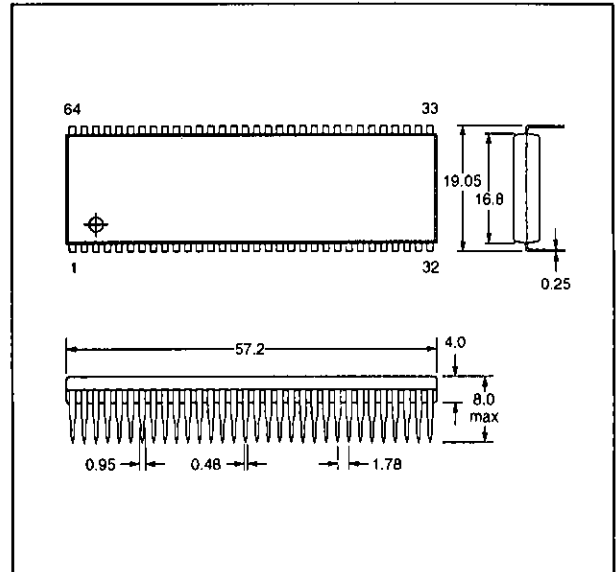
**PIN ASSIGNMENT**



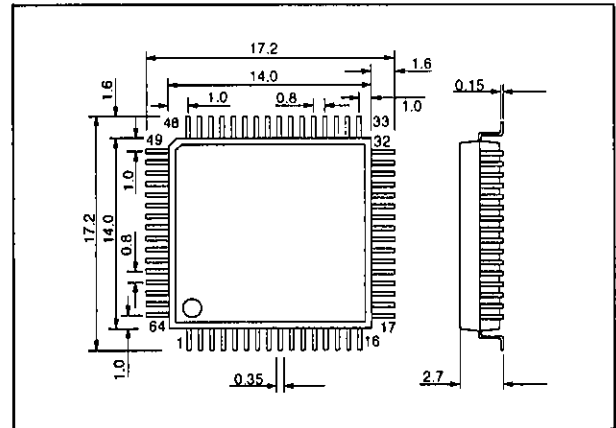
**PACKAGE DIMENSIONS**

Unit: mm

**3071-DIP64S**



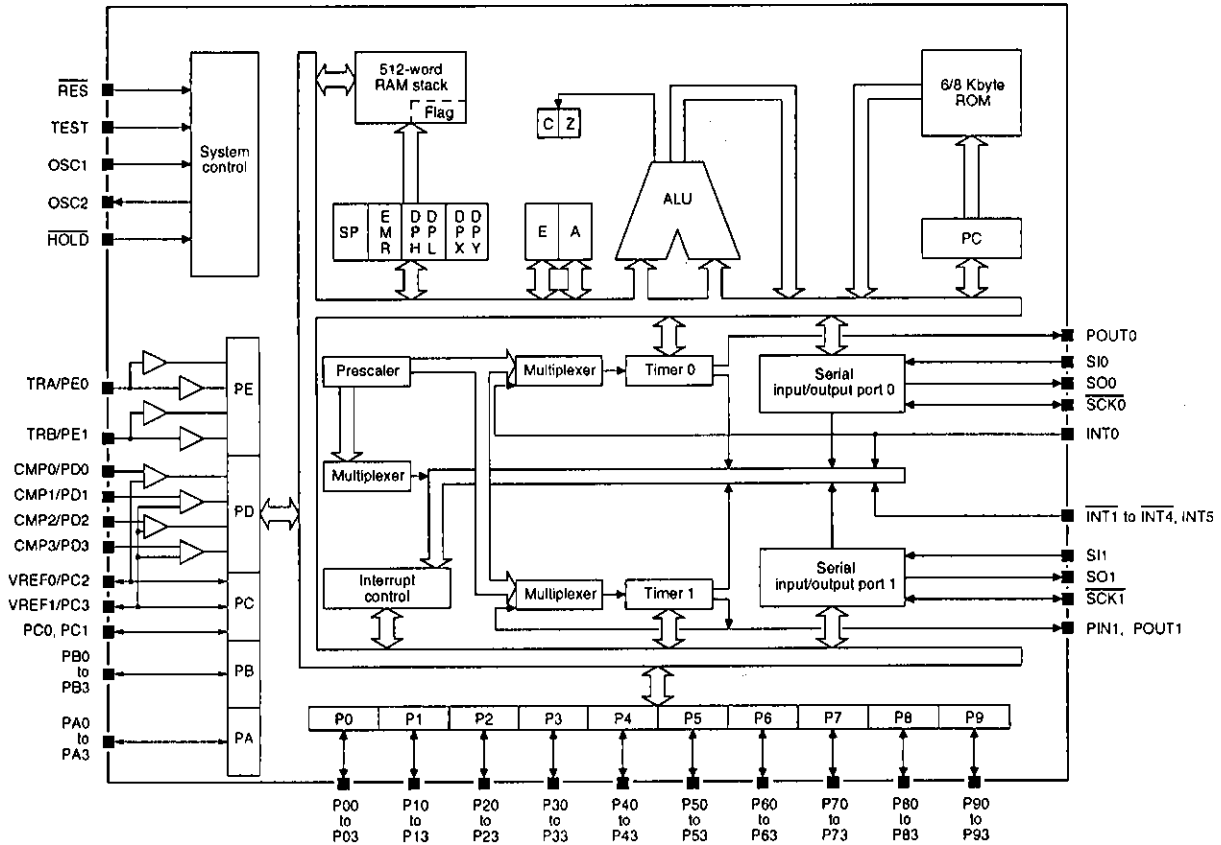
**3159-QIP64E**



**Note**

Reflow soldering is recommended for QIP packages. Please consult your local representative for further information.

**BLOCK DIAGRAM**



**PIN DESCRIPTION**

Number		Name	Description
DIP64S	QIP64E		
1	57	SI0/P20	Multiplexed 4-bit input/output port P2 (P20 to P23), serial input 0 (SI0), serial output 0 (SO0), serial clock 0 (SCK0) and interrupt request 0 (INT0)
2	58	SO0/P21	
3	59	SCK0/P22	
4	60	INT0/P23	
5	61	INT1/P30	Multiplexed 4-bit input/output port P3 (P30 to P33), interrupt request 1 (INT1), timer outputs (POUT0 and POUT1) and hold-mode control input (HOLD)
6	62	POUT0/P31	
7	63	POUT1/P32	
8	64	HOLD/P33	
9	1	P40	4-bit input/output port P4 (P40 to P43)
10	2	P41	
11	3	P42	
12	4	P43	
13	5	P50	4-bit input/output port P5 (P50 to P53)
14	6	P51	
15	7	P52	
16	8	P53	

**LC66556A, LC66558A**

Number		Name	Description
DIP64S	QIP64E		
17	9	SI1/P60	Multiplexed 4-bit input/output port P6 (P60 to P63), serial input 1 (SI1), serial output 1 (SO1), serial clock 1 (SCK1) and event counter input (PIN1)
18	10	SO1/P61	
19	11	SCK1/P62	
20	12	PIN1/P63	
21	13	P70	4-bit output port P7 (P70 to P73)
22	14	P71	
23	15	P72	
24	16	P73	
25	17	P80	4-bit output port P8 (P80 to P83)
26	18	P81	
27	19	P82	
28	20	P83	
29	21	INT2/P90	Multiplexed 4-bit input/output port P9 (P90 to P93) and interrupt requests (INT2 to INT4 and INT5)
35	27	INT3/P91	
36	28	INT4/P92	
37	29	INT5/P93	
30	22	TEST	CPU test input
31	23	VSS	Ground
32	24	OSC1	External oscillator connections
33	25	OSC2	
34	26	RES	Reset input
38	30	PA0	4-bit output port PA (PA0 to PA3)
39	31	PA1	
40	32	PA2	
41	33	PA3	
42	34	PB0	4-bit output port PB (PB0 to PB3)
43	35	PB1	
44	36	PB2	
45	37	PB3	
46	38	PC0	Multiplexed 4-bit input/output port PC (PC0 to PC3), comparator 0 reference voltage input (VREF0) and comparators 1, 2 and 3 reference voltage input (VREF1)
47	39	PC1	
48	40	VREF0/PC2	
49	41	VREF1/PC3	Multiplexed 4-bit input port PD (PD0 to PD3) and comparator inputs (CMP0 to CMP3)
50	42	CMP0/PD0	
51	43	CMP1/PD1	
52	44	CMP2/PD2	
53	45	CMP3/PD3	

## LC66556A, LC66558A

Number		Name	Description
DIP64S	QIP64E		
54	46	TRA/PE0	Multiplexed 2-bit input port PE (PE0 to PE1) and ternary inputs (TRA and TRB)
55	47	TRB/PE1	
56	48	P00	4-bit input/output port P0 (P00 to P03)
57	49	P01	
58	50	P02	
59	51	P03	
60	52	P10	4-bit input/output port P1 (P10 to P13)
61	53	P11	
62	54	P12	
63	55	P13	
64	56	VDD	5 V supply

## SPECIFICATIONS

### Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit
Supply voltage range	$V_{DD}$	-0.3 to 7.0	V
Ports P2, P3 (excluding P33) and P6 input voltage range. See note 1.	$V_{I1}$	-0.3 to 15.0	V
Input voltage range for all inputs. See note 2.	$V_{I2}$	-0.3 to $V_{DD} + 0.3$	V
Ports P2, P3 (excluding P33), P6, P7 and PA output voltage range. See note 1.	$V_{O1}$	-0.3 to 15.0	V
Output voltage range for all outputs. See note 2.	$V_{O2}$	-0.3 to $V_{DD} + 0.3$	V
Ports P0, P1, P4, P5, P7, PA and PB output source current	$-I_{OP1}$	2	mA
Ports P2, P3 (excluding P33), P6, P8, P9 and PC output source current	$-I_{OP2}$	4	mA
Ports P0 to P6 (excluding P33), P8, P9 and PC output sink current	$I_{ON1}$	4	mA
Ports P7, PA and PB output sink current	$I_{ON2}$	20	mA
Ports P2 to P8 (excluding P33) total sink current	$\Sigma I_{ON1}$	75	mA
Ports P0, P1 and P9 to PC total sink current	$\Sigma I_{ON2}$	75	mA
Ports P2 to P8 (excluding P33) total source current	$-\Sigma I_{OP1}$	25	mA
Ports P0, P1 and P9 to PC total source current	$-\Sigma I_{OP2}$	25	mA
Power dissipation (DIP64S)	$P_{D1}$	600	mW
Power dissipation (QIP64E)	$P_{D2}$	430	mW
Operating temperature range	$T_{opr}$	-30 to 70	°C
Storage temperature range	$T_{stg}$	-55 to 125	°C

### Notes

1. Open-drain output configuration option
2. All output configuration options

### Recommended Operating Conditions

$T_a = 25\text{ }^\circ\text{C}$ ,  $V_{SS} = 0\text{ V}$

Parameter	Symbol	Rating	Unit
Supply voltage	$V_{DD}$	5	V
Supply voltage range for $1.96 \leq t_{CYC} \leq 10\text{ }\mu\text{s}$ operation	$V_{DD}$	3.0 to 5.5	V
Supply voltage range for $3.92 \leq t_{CYC} \leq 10\text{ }\mu\text{s}$ operation		2.2 to 5.5	
Hold-mode supply voltage range for data retention	$V_{DD}$	1.8 to 5.5	V

### Electrical Characteristics

$V_{DD} = 2.2\text{ to }5.5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ ,  $T_a = -30\text{ to }70\text{ }^\circ\text{C}$  unless otherwise noted

Parameter	Symbol	Condition	Rating			Unit
			min	typ	max	
Supply current	$I_{DD}$	2 MHz ceramic resonator, $V_{DD} = 3.0\text{ to }5.5\text{ V}$	-	1.5	4.0	mA
		2 MHz external clock, $V_{DD} = 3.0\text{ to }5.5\text{ V}$	-	1.5	4.0	
		1 MHz ceramic resonator	-	1.0	4.0	
Halt-mode supply current	$I_{DDHT}$	2 MHz ceramic resonator, $V_{DD} = 3.0\text{ to }5.5\text{ V}$	-	0.8	1.5	mA
		2 MHz external clock, $V_{DD} = 3.0\text{ to }5.5\text{ V}$	-	0.8	1.5	
		1 MHz ceramic resonator	-	0.5	1.5	
Hold-mode supply current	$I_{DDHD}$	$V_{DD} = 1.8\text{ to }5.5\text{ V}$	-	0.01	10.0	$\mu\text{A}$
Ports P2, P3 (excluding P33), P6 and P9, $\overline{\text{RES}}$ and OSC1 LOW-level input voltage	$V_{IL1}$	Output n-channel transistor OFF. See note 1.	$V_{SS}$	-	$0.2V_{DD}$	V
HOLD/P33 LOW-level input voltage	$V_{IL2}$	$V_{DD} = 1.8\text{ to }5.5\text{ V}$	$V_{SS}$	-	$0.2V_{DD}$	V
Ports P0, P1, P4, P5, PC, PD and PE, and TEST LOW-level input voltage	$V_{IL3}$	Output n-channel transistor OFF. See note 1.	$V_{SS}$	-	$0.25V_{DD}$	V
Port PE LOW-level input voltage	$V_{IL4}$	Ternary input levels, $V_{DD} = 2.7\text{ to }5.5\text{ V}$	$V_{SS}$	-	$0.2V_{DD}$	V
Port PE MID-level input voltage	$V_{IM}$	Ternary input levels, $V_{DD} = 2.7\text{ to }5.5\text{ V}$	$0.4V_{DD}$	-	$0.6V_{DD}$	V
Ports P2, P3 (excluding P33) and P6 HIGH-level input voltage	$V_{IH1}$	Output n-channel transistor OFF. See notes 1 and 2.	$0.8V_{DD}$	-	13.5	V
HOLD/P33, P9, $\overline{\text{RES}}$ and OSC1 HIGH-level input voltage	$V_{IH2}$	Output n-channel transistor OFF. See note 2.	$0.8V_{DD}$	-	$V_{DD}$	V
Ports P0, P1, P4, P5, PC, PD and PE HIGH-level input voltage	$V_{IH3}$	Output n-channel transistor OFF. See note 1.	$0.75V_{DD}$	-	$V_{DD}$	V
Port PE HIGH-level input voltage	$V_{IH4}$	Ternary input levels, $V_{DD} = 2.7\text{ to }5.5\text{ V}$	$0.8V_{DD}$	-	$V_{DD}$	V
Ports P0 to P6 (excluding P33), P8 (CMOS), P9 and PC LOW-level output voltage	$V_{OL1}$	$I_{OL} = 1.6\text{ mA}$	-	-	0.4	V
Ports P7, PA and PB LOW-level output voltage	$V_{OL2}$	$I_{OL} = 3\text{ mA}$	-	-	1.5	V
		$I_{OL} = 8\text{ mA}$ , $V_{DD} = 3.0\text{ to }5.5\text{ V}$	-	-	1.5	

LC66556A, LC66558A

Parameter	Symbol	Condition	Rating			Unit
			min	typ	max	
Ports P2, P3 (excluding P33), P6, P8, P9 and PC HIGH-level output voltage	V <sub>OH1</sub>	I <sub>OH</sub> = -1 mA, V <sub>DD</sub> = 3.0 to 5.5 V. See note 3.	V <sub>DD</sub> - 1.0	-	-	V
		I <sub>OH</sub> = -0.5 mA. See note 3.	V <sub>DD</sub> - 1.0	-	-	
		I <sub>OH</sub> = -0.1 mA. See note 3.	V <sub>DD</sub> - 0.5	-	-	
Ports P0, P1, P4, P5, P7, PA and PB HIGH-level output voltage	V <sub>OH2</sub>	I <sub>OH</sub> = -50 μA, V <sub>DD</sub> = 3.0 to 5.5 V. See note 4.	V <sub>DD</sub> - 1.0	-	-	V
		I <sub>OH</sub> = -30 μA, V <sub>DD</sub> = 3.0 to 5.5 V. See note 4.	V <sub>DD</sub> - 0.5	-	-	
		I <sub>OH</sub> = -30 μA. See note 4.	V <sub>DD</sub> - 1.0	-	-	
		I <sub>OH</sub> = -20 μA. See note 4.	V <sub>DD</sub> - 0.5	-	-	
Ports PC2 and PD0 in-phase, comparator input voltage range	V <sub>CMM1</sub>	V <sub>DD</sub> = 2.7 to 5.5 V	1.5	-	V <sub>DD</sub>	V
Ports PC3 and PD1 to PD3 in-phase, comparator input voltage range	V <sub>CMM2</sub>	V <sub>DD</sub> = 2.7 to 5.5 V	V <sub>SS</sub>	-	V <sub>DD</sub> - 1.5	V
Ports PD1 to PD3 comparator offset voltage	V <sub>OS1</sub>	V <sub>I</sub> = V <sub>SS</sub> to V <sub>DD</sub> - 1.5 V, V <sub>DD</sub> = 2.7 to 5.5 V	-	±50	±300	mV
Port PD0 comparator offset voltage	V <sub>OS2</sub>	V <sub>I</sub> = 1.5 V to V <sub>DD</sub> , V <sub>DD</sub> = 2.7 to 5.5 V	-	±50	±300	mV
Ports P2, P3, P6 and P9, and $\overline{RES}$ and OSC1 Schmitt-trigger LOW-level threshold voltage	V <sub>IL</sub>		0.2V <sub>DD</sub>	-	0.5V <sub>DD</sub>	V
Ports P2, P3, P6 and P9, and $\overline{RES}$ and OSC1 Schmitt-trigger HIGH-level threshold voltage	V <sub>IH</sub>		0.5V <sub>DD</sub>	-	0.8V <sub>DD</sub>	V
Ports P2, P3, P6 and P9, $\overline{RES}$ and OSC1 Schmitt-trigger hysteresis voltage	V <sub>HYS</sub>		-	0.1V <sub>DD</sub>	-	V
Ports PC2, PC3, PD and PE LOW-level input current	I <sub>IL1</sub>	V <sub>I</sub> = V <sub>SS</sub> , output n-channel transistor OFF. See note 2.	-1.0	-	-	μA
LOW-level input current for all other inputs	I <sub>IL2</sub>	V <sub>I</sub> = V <sub>SS</sub> , output n-channel transistor OFF. See note 2.	-1.0	-	-	μA
Ports P2, P3 (excluding P33) and P6 HIGH-level input current	I <sub>IH1</sub>	V <sub>I</sub> = 13.5 V, output n-channel transistor OFF. See notes 1 and 2.	-	-	5.0	μA
Ports P0, P1, P33, P4, P5, P9, PC (excluding PC2 and PC3) and $\overline{RES}$ and OSC1 HIGH-level input current	I <sub>IH2</sub>	V <sub>I</sub> = V <sub>DD</sub> , output n-channel transistor OFF. See notes 1 and 2.	-	-	1.0	μA
Ports PC2, PC3, PD and PE HIGH-level input current	I <sub>IH3</sub>	V <sub>I</sub> = V <sub>DD</sub> , output n-channel transistor OFF. See notes 1 and 2.	-	-	1.0	μA
Ports P2, P3, P6, P7 and PA output leakage current	I <sub>OFF1</sub>	V <sub>I</sub> = 13.5 V. Output n-channel transistor OFF. See note 2.	-	-	5.0	μA
Ports P0, P1, P4, P5, P9, PB and PC output leakage current	I <sub>OFF2</sub>	V <sub>I</sub> = V <sub>DD</sub> . Output n-channel transistor OFF. See note 2.	-	-	1.0	μA

## LC66556A, LC66558A

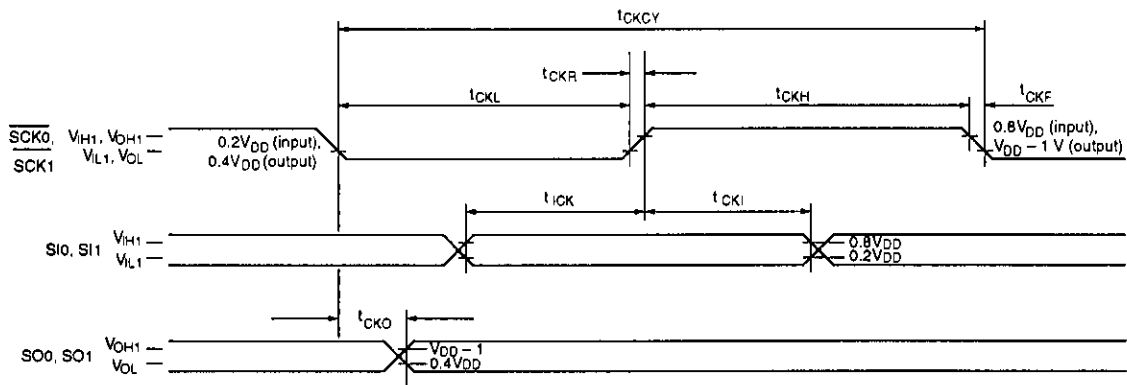
Parameter	Symbol	Condition	Rating			Unit
			min	typ	max	
Port P8 output leakage current	$I_{OFF3}$	$V_i = V_{SS}$ . Output p-channel transistor OFF. See note 2.	-1.0	-	-	$\mu A$
Ports P0, P1, P4, P5, P7, PA and PB output current with pull-up option	$I_{PO}$	$V_i = V_{SS}$ . $V_{DD} = 5.5 V$ . Output n-channel transistor OFF. See note 4.	-1.6	-	-	mA
Ceramic resonator input frequency	$f_{CF}$	$V_{DD} = 3.0$ to $5.5 V$ , 2 MHz resonator	-	2.0	-	MHz
		1 MHz resonator	-	1.0	-	
Ceramic resonator input stabilization time	$t_{CFS}$	$V_{DD} = 3.0$ to $5.5 V$ , 2 MHz resonator	-	-	10	ms
		1 MHz resonator	-	-	10	
OSC1 external clock input frequency	$f_{ext}$	$V_{DD} = 3.0$ to $5.5 V$	0.4	-	2.03	MHz
			0.4	-	1.02	

### Notes

1. Ports with CMOS output configuration option cannot be used as input ports.
2. Open-drain output configuration option
3. CMOS output configuration option
4. Pull-up output configuration option

### Timing Characteristics

#### Serial input/output timing



$V_{DD} = 2.2$  to  $5.5 V$ ,  $V_{SS} = 0 V$ ,  $T_a = -30$  to  $70 ^\circ C$

Parameter	Symbol	Condition	Rating			Unit
			min	typ	max	
Instruction cycle time	$t_{cyc}$	$f_{CP} = 0.4$ to $2.03 MHz$ , $V_{DD} = 3.0$ to $5.5 V$	10	-	1.96	$\mu s$
		$t_{cyc} = 0.4$ to $1.02 MHz$	10	-	3.92	
$\overline{SCK0}$ and $\overline{SCK1}$ serial clock input cycle time	$t_{CKCY}$	$V_{DD} = 3.0$ to $5.5 V$	1.9	-	-	$\mu s$
			3.9	-	-	
$\overline{SCK0}$ and $\overline{SCK1}$ serial clock output cycle time			$2t_{cyc}$	-	-	

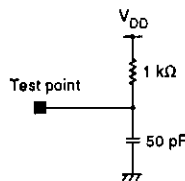


## LC66556A, LC66558A

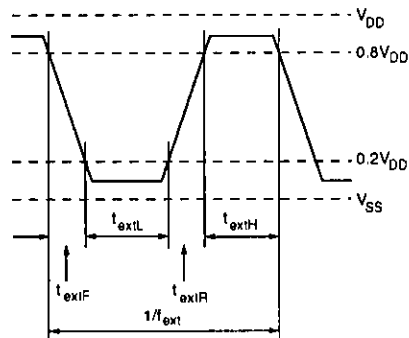
Parameter	Symbol	Condition	Rating			Unit
			min	typ	max	
$\overline{\text{SCK0}}$ and $\overline{\text{SCK1}}$ serial clock input pulsewidth	$t_{\text{ckL}}$	$V_{\text{DD}} = 3.0$ to $5.5$ V	0.9	-	-	$\mu\text{s}$
			1.9	-	-	
$\overline{\text{SCK0}}$ and $\overline{\text{SCK1}}$ serial clock output pulsewidth			$t_{\text{cyc}}$	-	-	
$\overline{\text{SCK0}}$ and $\overline{\text{SCK1}}$ serial clock output rise time	$t_{\text{ckR}}$		-	-	0.1	$\mu\text{s}$
$\overline{\text{SCK0}}$ and $\overline{\text{SCK1}}$ serial clock output fall time	$t_{\text{ckF}}$		-	-	0.1	$\mu\text{s}$
S10 and S11 serial data setup time	$t_{\text{ck}}$		0.6	-	-	$\mu\text{s}$
S10 and S11 serial data hold time	$t_{\text{ckI}}$		0.6	-	-	$\mu\text{s}$
S00 and S01 serial data output delay	$t_{\text{cko}}$	$V_{\text{DD}} = 3.0$ to $5.5$ V	-	-	0.6	$\mu\text{s}$
			-	-	0.9	

### Note

Each test input and output has an RC load as shown in the following figure.



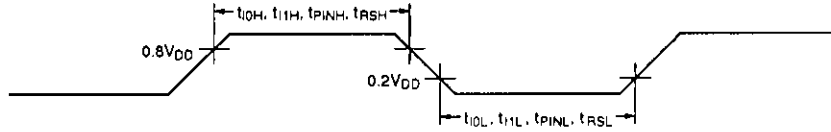
### External clock timing



$V_{\text{DD}} = 2.2$  to  $5.5$  V,  $V_{\text{SS}} = 0$  V,  $T_a = -30$  to  $70$  °C

Parameter	Symbol	Condition	Rating			Unit
			min	typ	max	
OSC1 external clock LOW-level input pulsewidth	$t_{\text{extL}}$	$V_{\text{DD}} = 3.0$ to $5.5$ V	100	-	-	ns
			200	-	-	
OSC1 external clock HIGH-level input pulsewidth	$t_{\text{extH}}$	$V_{\text{DD}} = 3.0$ to $5.5$ V	100	-	-	ns
			200	-	-	
OSC1 external clock input rise time	$t_{\text{extR}}$		-	-	30	ns
OSC1 external clock input fall time	$t_{\text{extF}}$		-	-	30	ns

**Interrupt and reset timing**

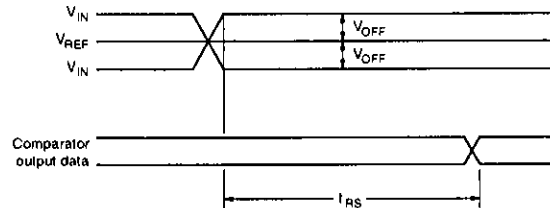


$V_{DD} = 2.2$  to  $5.5$  V,  $V_{SS} = 0$  V,  $T_a = -30$  to  $70$  °C

Parameter	Symbol	Rating			Unit
		min	typ	max	
INT0 LOW-level pulsewidth	$t_{IOL}$	2tcyc	–	–	$\mu$ s
INT0 HIGH-level pulsewidth	$t_{IQH}$	2tcyc	–	–	$\mu$ s
$\overline{INT1}$ to $\overline{INT4}$ and INT5 LOW-level pulsewidth	$t_{IHL}$	2tcyc	–	–	$\mu$ s
$\overline{INT1}$ to $\overline{INT4}$ and INT5 HIGH-level pulsewidth	$t_{IHH}$	2tcyc	–	–	$\mu$ s
PIN1 LOW-level input pulsewidth	$t_{PINL}$	2tcyc	–	–	$\mu$ s
PIN1 HIGH-level input pulsewidth	$t_{PINH}$	2tcyc	–	–	$\mu$ s
$\overline{RES}$ LOW-level input pulsewidth	$t_{RSL}$	3tcyc	–	–	$\mu$ s
$\overline{RES}$ HIGH-level input pulsewidth	$t_{RSH}$	3tcyc	–	–	$\mu$ s

**Comparator output timing**

$V_{DD} = 2.7$  to  $5.5$  V,  $V_{SS} = 0$  V,  $T_a = -30$  to  $70$  °C



Parameter	Symbol	Condition	Rating			Unit
			min	typ	max	
Port PD comparator response time	$t_{RS}$	$V_{DD} = 2.7$ to $5.5$ V	–	–	20	ms

**INPUT AND OUTPUT FUNCTIONS**

The LC66556A and LC66558A have many multiplexed pins whose functions are controlled by software. The function of each pin is described in table 1.

Table 1. Pin functions

Name	Function
P00	Ports P00 to P03 can be addressed as either a 4-bit port or four, single-bit ports. They also have half-mode control functions. Level after reset is set by user option.
P01	
P02	
P03	

LC66556A, LC66558A

Table 1. Pin functions—continued

Name	Function
P10	Ports P10 to P13 can be addressed as either a 4-bit port or four, single-bit ports. Level after reset is set by user option.
P11	
P12	
P13	
SI0/P20	Ports P20 to P23 can be addressed as either a 4-bit port or four, single-bit ports. Port P20 also functions as a serial data input, P21 as a serial data output, P22 as a serial data clock and P23 as an interrupt request, pulsewidth measurement and event counter input using timer 0. HIGH-level after reset
SO0/P21	
SCK0/P22	
INT0/P23	
$\overline{\text{INT1}}$ /P30	Ports P30 to P32 can be addressed as either a 3-bit port, a 4-bit port with P33 or three, single-bit ports. Port P30 also functions as an interrupt request input, P31 as a square-wave output from timer 0, and P32 as a square-wave output and a PWM output from timer 1. HIGH-level after reset
POUT0/P31	
POUT1/P32	
$\overline{\text{HOLD}}$ /P33	Port P33 can be addressed as either a 4-bit port with P30 to P32 or a single-bit port. It functions as the hold-mode control input when P33 is LOW and the HOLD instruction is executed. The CPU restarts when P33 goes HIGH again. Reset signals are ignored whenever $\overline{\text{HOLD}}$ :P33 is LOW, including when not in hold mode.
P40	Ports P40 to P43 can be addressed as either a 4-bit port, four, single-bit ports or an 8-bit port with P50 to P53. HIGH-level after reset
P41	
P42	
P43	
P50	Ports P50 to P53 can be addressed as either a 4-bit port, four, single-bit ports or an 8-bit port with P40 to P43. HIGH-level after reset
P51	
P52	
P53	
SI1/P60	Ports P60 to P63 can be addressed as either a 4-bit port or four, single-bit ports. Port P60 also functions as a serial data input, P61 as a serial data output, P62 as a serial data clock and P63 as the timer 1 event counter input. HIGH-level after reset
SO1/P61	
SCK1/P62	
PIN1/P63	
P70	Ports P70 to P73 can be addressed as either a 4-bit port or four, single-bit ports. Input instructions read the contents of the output latch. HIGH-level after reset
P71	
P72	
P73	
P80	Ports P80 to P83 can be addressed as either a 4-bit port or four, single-bit ports. Input instructions read the contents of the output latch. Level after reset is set by user option. Note that the open-drain option for P8 is p-channel.
P81	
P82	
P83	
$\overline{\text{INT2}}$ /P90	Ports P90 to P93 can be addressed as either a 4-bit port or four, single-bit ports. P90 also functions as interrupt request 2, P91 as interrupt request 3, P92 as interrupt request 4 and P93 as interrupt request 5. HIGH-level after reset
$\overline{\text{INT3}}$ /P91	
$\overline{\text{INT4}}$ /P92	
$\overline{\text{INT5}}$ /P93	

Table 1. Pin functions—continued

Name	Function
PA0	Ports PA0 to PA3 can be addressed as either a 4-bit port or four, single-bit ports. Input instructions read the contents of the output latch. HIGH-level after reset
PA1	
PA2	
PA3	
PB0	Ports PB0 to PB3 can be addressed as either a 4-bit port or four, single-bit ports. Input instructions read the contents of the output latch. HIGH-level after reset
PB1	
PB2	
PB3	
PC0	Ports PC0 to PC3 can be addressed as either a 4-bit port or four, single-bit ports. Port PC2 also functions as the reference voltage input for comparator 0, and PC3, as the reference voltage input for comparators 1, 2 and 3. HIGH-level after reset
PC1	
VREF0/PC2	
VREF1/PC3	
CMP0/PD0	Ports PD0 to PD3 can be addressed as either a 4-bit port or four, single-bit ports. They also function as comparator inputs. Normal input after reset
CMP1/PD1	
CMP2/PD2	
CMP3/PD3	
TRA/PE0	Ports PE0 to PE1 can be addressed as either a 2-bit port or two, single-bit ports. They also function as ternary-level inputs. Normal input after reset
TRB/PE1	
OSC1	OSC1 and OSC2 function as the external ceramic resonator connections. When an external clock is used, OSC2 is left open.
OSC2	
$\overline{\text{RES}}$	When $\overline{\text{RES}}$ goes LOW while $\overline{\text{HOLD/P33}}$ is HIGH, the CPU is reset.
TEST	CPU test input. Normally connected to ground

**USER OPTIONS**

**Oscillator Options**

There are two user options for the oscillator—an external clock and a ceramic resonator. An external RC oscillator is not supported. The internal circuits of OSC1 and OSC2 and connections for the external clock and ceramic resonator options are shown in figures 1 and 2, respectively. Note the Schmitt-trigger input for the external clock option.



Figure 1. External clock option

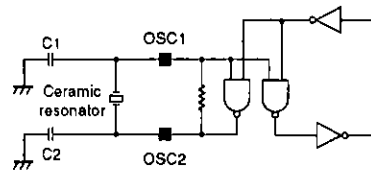


Figure 2. Ceramic resonator option

### Output Options

There are two user options for the output configuration of each port—n-channel open drain and p-channel, active pull-up. The options for all ports, excluding P8, PD and PE, are shown in figures 3 and 5, and for P8, in figures 4 and 6. Note that the open-drain option for port P8 is p-channel. Ports P2, P3, P6 and P9 have Schmitt-trigger inputs in both output configurations.

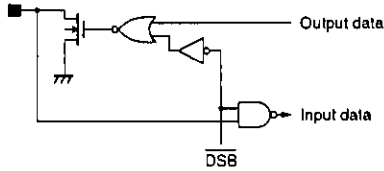


Figure 3. N-channel open-drain option for all ports except P8

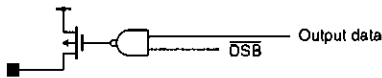


Figure 4. P-channel open-drain option for port P8

The p-channel pull-up option for all ports is identical. However, the ports are classified as pull-up or CMOS according to the drive capability of the p-channel transistor. Ports P0, P1, P4, P5, P7, PA and PB are classified as pull-up, and ports P2, P3, P6, P9 and PC, as CMOS.

### SPECIFYING OPTIONS

The user-addressable memory is in the range 0000H to 2007H. Addresses 0000H to 1FFFH are for user programs, and addresses 2000H to 2007H, for option specification. The option specification is coded using the

Table 2. User options

Address	Data bit	Parameter	Option	
			0	1
2000H	D0	Watchdog timer function	No	Yes
	D1	Port P0 level after reset	LOW	HIGH
	D2	Port P1 level after reset	LOW	HIGH
	D3	Port P8 level after reset	LOW	HIGH
	D4	Oscillator	External clock	Ceramic resonator
	D5 to D7	No function	Set to 0	

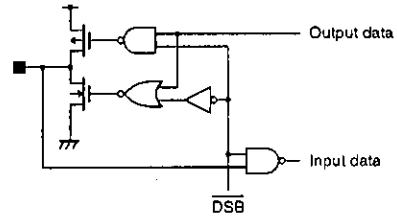


Figure 5. P-channel pull-up option for all ports except P8, PD and PE

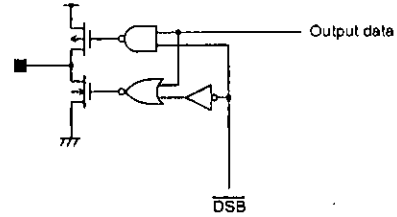


Figure 6. P-channel pull-up option for port P8

Note that the n-channel open-drain outputs for ports P2, P3, P6, P7 and PA have a maximum withstand voltage of 15 V.

### Output Level After Reset Option

The output level of ports P0, P1 and P8 after a CPU reset is user selectable.

### Watchdog Timer Option

A watchdog timer is also available to prevent program runaway.

information shown in table 2. Refer to the *LC66S Jump Optimizing Cross Assembler Manual* for setting information.

LC66556A, LC66558A

Table 2. User options—continued

Address	Data bit	Parameter	Option	
			0	1
2001H	D0	Port P00 output configuration	Open-drain	Pull-up
	D1	Port P01 output configuration		
	D2	Port P02 output configuration		
	D3	Port P03 output configuration		
	D4	Port P10 output configuration	Open-drain	Pull-up
	D5	Port P11 output configuration		
	D6	Port P12 output configuration		
	D7	Port P13 output configuration		
2002H	D0	Port P20 output configuration	Open-drain	CMOS
	D1	Port P21 output configuration		
	D2	Port P22 output configuration		
	D3	Port P23 output configuration		
	D4	Port P30 output configuration	Open-drain	CMOS
	D5	Port P31 output configuration		
	D6	Port P32 output configuration		
	D7	No function		
2003H	D0	Port P40 output configuration	Open-drain	Pull-up
	D1	Port P41 output configuration		
	D2	Port P42 output configuration		
	D3	Port P43 output configuration		
	D4	Port P50 output configuration	Open-drain	Pull-up
	D5	Port P51 output configuration		
	D6	Port P52 output configuration		
	D7	Port P53 output configuration		
2004H	D0	Port P60 output configuration	Open-drain	CMOS
	D1	Port P61 output configuration		
	D2	Port P62 output configuration		
	D3	Port P63 output configuration		
	D4	Port P70 output configuration	Open-drain	Pull-up
	D5	Port P71 output configuration		
	D6	Port P72 output configuration		
	D7	Port P73 output configuration		

Table 2. User options—continued

Address	Data bit	Parameter	Option	
			0	1
2005H	D0	Port P80 output configuration	Open-drain	Pull-up
	D1	Port P81 output configuration		
	D2	Port P82 output configuration		
	D3	Port P83 output configuration		
	D4	Port P90 output configuration	Open-drain	CMOS
	D5	Port P91 output configuration		
	D6	Port P92 output configuration		
	D7	Port P93 output configuration		
2006H	D0	Port PA0 output configuration	Open-drain	Pull-up
	D1	Port PA1 output configuration		
	D2	Port PA2 output configuration		
	D3	Port PA3 output configuration		
	D4	Port PB0 output configuration	Open-drain	Pull-up
	D5	Port PB1 output configuration		
	D6	Port PB2 output configuration		
	D7	Port PB3 output configuration		
2007H	D0	Port PC0 output configuration	Open-drain	CMOS
	D1	Port PC1 output configuration		
	D2	Port PC2 output configuration		
	D3	Port PC3 output configuration		
	D4 to D7	No function	Set to 0	

The assembler execute command when specifying programs and options using a Sanyo cross assembler is LC66S.EXE.

**APPLICATION NOTES**

**Reference Clock**

The external circuit for a ceramic resonator is shown in figure 7, and the corresponding recommended resonator and component values, in table 3. The oscillator stabilization characteristics are shown in figure 8.

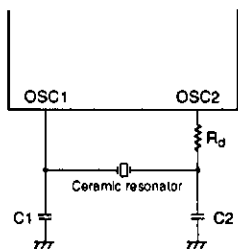


Figure 7. Ceramic resonator

Table 3. Recommended ceramic resonators

Ceramic resonator	R <sub>d</sub>	C1	C2
2 MHz Murata CSA-2.00MG	0 Ω	33 pF ±10%	33 pF ±10%
2 MHz Kyocera KBR-2.0MS	0 Ω	47 pF ±10%	47 pF ±10%
1 MHz Murata CSB1000J	2.2 kΩ ±5%	100 pF ±10%	100 pF ±10%
1 MHz Kyocera KBR1000H	0 Ω	100 pF ±10%	100 pF ±10%

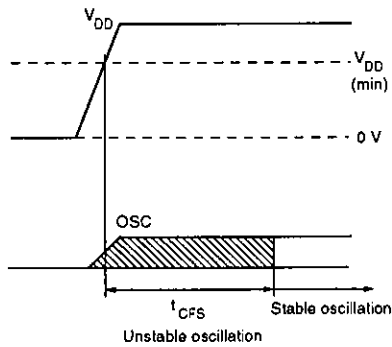


Figure 8. Ceramic resonator stabilization time

The external clock input connection is OSC1. The remaining oscillator connection, OSC2, should be left open as shown in figure 9.

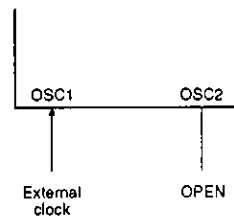


Figure 9. External clock connection

**DEVELOPMENT TOOLS**

Program development for the LC66556A and LC66558A microcontrollers can be performed using a cross assembler running on an IBM-compatible personal computer under MS-DOS. A number of other development tools are available to simplify and speed the

development process—the EVA800/850 debugger, the EVA800/850-TB665XX evaluation chip board, the LC66599 evaluation chip and the LC66E516 EPROM, shown in figure 10.

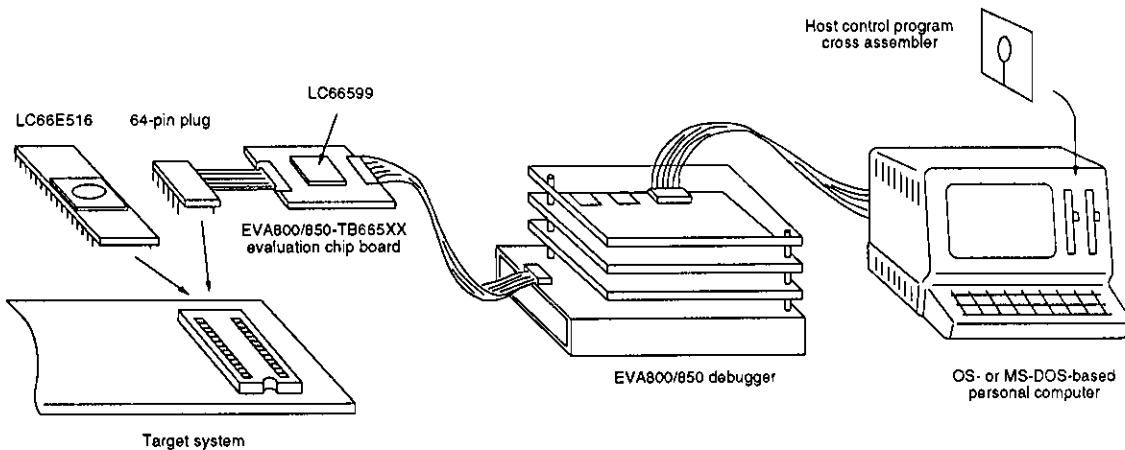


Figure 10. Development tools

**Program Debugger**

The EVA800/850 communicates with an external host computer using a standard serial interface. It performs basic debugging functions, including breakpoints,

single-stepping and tracing under MPM665XX debug monitor software control. It also includes an EPROM programmer.



## Evaluation System

The evaluation chip board, which holds the LC66599 evaluation chip, has a 64-pin connector which plugs directly into the socket of a target system. Jumpers and

switches configure the initial device options and states as shown in tables 4 and 5.

Table 4. Jumper settings

Jumper 1		Jumper 2		Jumper 3	
Oscillator		Reset method		Target system supply source	
EXT	External clock	INT (a)	Reset on a RUN command from the host computer	ON (a)	Supply power from the evaluation chip board output
RC	RC oscillator				
CF	Ceramic filter resonator	EXT (b)	Reset by the target system reset circuit	OFF (b)	Separate evaluation chip and target system supplies

### Note

An RC option is not supported. Accordingly, do not connect jumper 1 to the RC setting.

Table 5. Switch 1

P0S	Port P0 output level on reset	P1S	Port P1 output level on reset	P8S	Port P8 output level on reset	WDC	Watchdog timer enable
ON	All outputs HIGH	ON	All outputs HIGH	ON	All outputs HIGH	ON	Watchdog timer enabled
OFF	All outputs LOW	OFF	All outputs LOW	OFF	All outputs LOW	OFF	Watchdog timer disabled

### Note

RC0 and RC1 should be turned ON together.

Switches 2 to 14 select the internal pull-up resistor option. When a switch is ON, the corresponding output pin has an internal pull-up resistor (pull-down resistor

for port P8 using SW10), and when OFF, the corresponding pin is an open-drain output.

## Cross Assembler

The cross assembler execute file, LC66S.EXE, can be used for the devices shown in table 6. Refer to the

*LC66S Jump Optimizing Cross Assembler Manual* for operating information.

Table 6. Cross assembler compatibility

Cross assembler	Target device	Bank Instructions supported
LC66S.EXE	LC66556A, LC66558A	None
	LC66E516, LC66P516, LC66599	SB0, SB1, SB2, SB3

## Simulation Chip

The LC66E516 is a microprocessor with a 16-Kbyte EPROM which is used for simulation. It can be programmed by a standard EPROM programmer using the W66E516DH (DIP) or W66E516QH (QIP) adapter boards. The LC66E516 is pin- and functionally-compatible with the LC66556A and LC66558A devices. Note that the hold-mode release time and electrical specifications of the LC66E516 differ from those of the LC66556A and LC66558A.

The LC66E516, shown in figure 11, can be configured to match the target device by programming certain EPROM locations. These locations set the reset level of ports 0, 1 and 8, the watchdog timer enable and the port output types. Refer to the LC66E516 datasheet for further operating information.

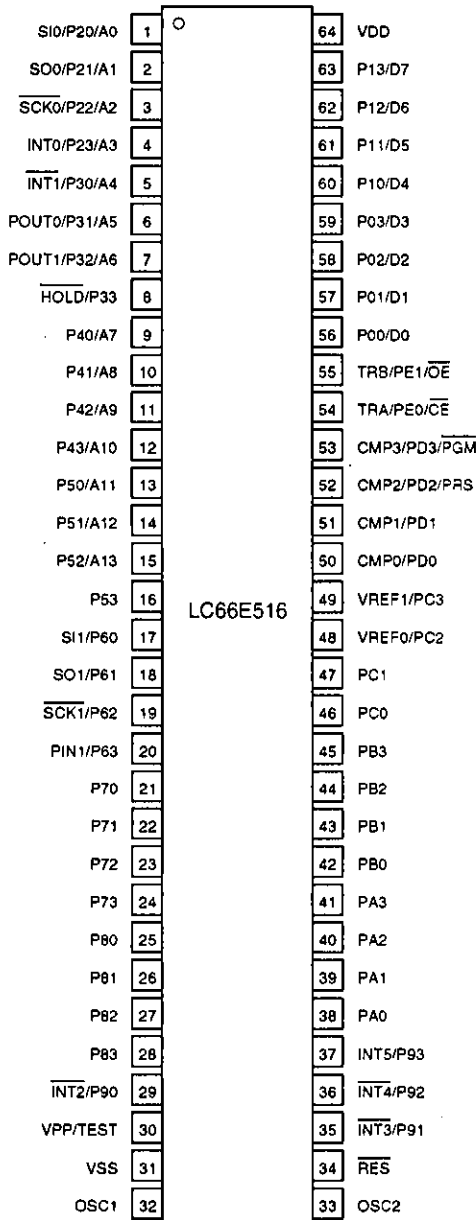


Figure 11. LC66E516 pinout

Series Comparison

A comparison of the LC66508B series characteristics with those of the LC66556A and LC66558A is shown in table 7.

Table 7. Device comparison

Parameter	LC66508B series	LC66556A and LC66558A
Hold-mode release hardware delay	65,536 cycles (approximately 64 ms with a 4 MHz clock)	16,384 cycles (approximately 32 ms with a 2 MHz clock and 64 ms with a 1 MHz clock)
Timer 0 contents after reset	FF0H	FFCH
Operating supply voltage and clock period	4.0 to 6.0 V for 0.92 to 10 μs (LC66506B and LC66508B), 4.5 to 5.5 V for 0.92 to 10 μs (LC66E516 and LC66P516)	2.2 to 5.5 V for 3.92 to 10 ms, 3.0 to 5.5 V for 1.96 to 10 μs

Notes

1. The LC66556A and LC66558A do not support an RC oscillator.
2. Refer to the LC66E516 and LC66P516 datasheets for output drive current and comparator input voltages.

A breakdown of the LC66 series devices is shown in table 8.

Table 8. LC66 series devices

Device	Pins	ROM capacity	RAM capacity	Package type	Type
LC66304A/306A/308A	42	4/6/8 Kbyte ROM	512 words	DIP42S and QIP48E	Normal type 4.0 to 6.0 V/0.92 μs
LC66404A/406A/408A	42	4/6/8 Kbyte ROM	512 words	DIP42S and QIP48E	
LC66506B/508B/512B/516B	64	6/8/12/16 Kbyte ROM	512 words	DIP64S and QIP64A	
LC66354A/356A/358A	42	4/6/8 Kbyte ROM	512 words	DIP42S and QIP48E	Low-voltage type 2.2 to 5.5 V/3.92 μs
LC66354S/356S/358S	42	4/6/8 Kbyte ROM	512 words	QIP44M	
LC66556A/558A/562A/566A	64	6/8/12/16 Kbyte ROM	512 words	DIP64S and QIP64E	Low voltage, high-speed type 3.0 to 5.5 V/0.92 μs
LC66354B/356B/358B	42	4/6/8 Kbyte ROM	512 words	DIP42S and QIP48E	
LC66556B/558B*	64	6/8 Kbyte ROM	512 words	DIP64S and QIP64E	
LC66562B/566B	64	12/16 Kbyte ROM	512 words	DIP64S and QIP64E	

**LC66556A, LC66558A**

Table 8. LC66 series devices—continued

Device	Pins	ROM capacity	RAM capacity	Package type	Type
LC66E308	42	8 Kbyte EPROM	512 words	DIC42S and QIC48	Evaluation ROMs and EPROMs with window 4.5 to 5.5 V/0.92 μs
LC66P308	42	8 Kbyte PROM	512 words	DIP42S and QIP48E	
LC66E408	42	8 Kbyte EPROM	512 words	DIC42S and QIC48	
LC66P408	42	8 Kbyte PROM	512 words	DIP42S and QIP48E	
LC66E516	64	16 Kbyte EPROM	512 words	DIC64S and QIC64	
LC66P516	64	16 Kbyte PROM	512 words	DIP64S and QIP64E	

\* Under development

**INSTRUCTION SET**

The following abbreviations are used in the instruction set table.

AC	Accumulator	PCh	Bits 8 to 11 of the program counter
E	E register	PCm	Bits 4 to 7 of the program counter
CF	Carry flag	PCI	Bits 0 to 3 of the program counter
ZF	Zero flag	F <sub>n</sub>	User flags, n = 0 to 15
HL	Data pointer DP <sub>H</sub> and DP <sub>L</sub>	TIMER0	Timer 0
XY	Data pointer DP <sub>X</sub> and DP <sub>Y</sub>	TIMER1	Timer 1
M	Data memory	SIO	Serial port register
M (HL)	Data memory pointed to by DP <sub>HL</sub>	P	Port
M (XY)	Data memory pointed to by DP <sub>XY</sub>	P (i4)	Port specified by 4-bit immediate data
M2 (HL)	Two-word location of data memory at even address pointed to by DP <sub>HL</sub>	INT	Interrupt enable flag
SP	Stack pointer	( ), [ ]	Contents
M2 (SP)	Two-word location of data memory pointed to by SP	←	Direction of transfer, result
M4 (SP)	Four-word location of data memory pointed to by SP	⊕	Logical exclusive-OR
i <sub>n</sub>	n-bit immediate data	·	Logical AND
i <sub>2</sub>	Bit specifier	+	Logical OR
		+	Arithmetic addition
		-	Arithmetic subtraction

Mnemonic	Operation	Instruction code								Cycles	Notation	Description	Status flags
		D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0				
<b>Accumulator instructions</b>													
CLA	Clear AC	1	0	0	0	0	0	0	0	1	AC ← 0 (equivalent to LAI 0)	Clears the contents of the accumulator. Vertical skip function available	ZF
DAA	Decimal adjust AC after addition	1	1	0	0	1	1	1	1	2	AC ← (AC) + 6 (equivalent to ADI 6)	Adds 6 to the contents of the accumulator	ZF
DAS	Decimal adjust AC after subtraction	1	1	0	0	1	1	1	1	2	AC ← (AC) + 10 (equivalent to ADI 0H)	Adds 10 to the contents of the accumulator	ZF
CLC	Clear CF	0	0	0	1	1	1	1	0	1	CF ← 0	Clears the carry flag	CF
STC	Set CF	0	0	0	1	1	1	1	1	1	CF ← 1	Sets the carry flag	CF
CMA	Complement AC	0	0	0	1	1	0	0	0	1	AC ← $\overline{(AC)}$	Takes the 1s complement of the contents of the accumulator	ZF
IA	Increment AC	0	0	0	1	0	1	0	0	1	AC ← (AC) + 1	Increments the contents of the accumulator by 1	ZF, CF
DA	Decrement AC	0	0	1	0	0	1	0	0	1	AC ← (AC) - 1	Decrements the contents of the accumulator by 1	ZF, CF

LC66556A, LC66558A

Mnemonic	Operation	Instruction code								Cycles	Notation	Description	Status flags
		D7	D6	D5	D4	D3	D2	D1	D0				
RAR	Rotate AC right through CF	0	0	0	1	0	0	0	0	1	$AC_3 \leftarrow (CF)$ $AC_n \leftarrow (AC_n + 1)$ $CF \leftarrow (AC_0)$	Shifts the contents of the accumulator right through the carry flag	CF
RAL	Rotate AC left through CF	0	0	0	0	0	0	0	1	1	$AC_0 \leftarrow (CF)$ $AC_n + 1 \leftarrow (AC_n)$ $CF \leftarrow (AC_3)$	Shifts the contents of the accumulator left through the carry flag	CF, ZF
TAE	Transfer AC to E	0	1	0	0	0	1	0	1	1	$E \leftarrow (AC)$	Copies the contents of the accumulator into register E	
TEA	Transfer E to AC	0	1	0	0	0	1	1	0	1	$AC \leftarrow (E)$	Copies the contents of register E into the accumulator	ZF
XAE	Exchange AC with E	0	1	0	0	0	1	0	0	1	$(AC) \leftrightarrow (E)$	Exchanges the contents of the accumulator and register E	
<b>Memory Instructions</b>													
IM	Increment M	0	0	0	1	0	0	1	0	1	$M(HL) \leftarrow [M(HL)] + 1$	Increments the contents of memory location HL by 1	ZF, CF
DM	Decrement M	0	0	1	0	0	0	1	0	1	$M(HL) \leftarrow [M(HL)] - 1$	Decrements the contents of memory location HL by 1	ZF, CF
IMDR i8	Increment M direct	1 i7	1 i6	0 i5	0 i4	0 i3	1 i2	1 i1	1 i0	2	$M(i8) \leftarrow [M(i8)] + 1$	Increments the contents of the memory location specified by immediate data i0 to i7 by 1	ZF, CF
OMDR i8	Decrement M direct	1 i7	1 i6	0 i5	0 i4	0 i3	0 i2	1 i1	1 i0	2	$M(i8) \leftarrow [M(i8)] - 1$	Decrements the contents of the memory location specified by immediate data i0 to i7 by 1	ZF, CF
SMB i2	Set M data bit	0	0	0	0	1	1	i1	i0	1	$[M(HL), i2] \leftarrow 1$	Sets the bit in memory location HL specified by i0 and i1	
RMB i2	Reset M data bit	0	0	1	0	1	1	i1	i0	1	$[M(HL), i2] \leftarrow 0$	Clears the bit in memory location HL specified by i0 and i1	ZF
<b>Arithmetic instructions</b>													
AD	Add M to AC	0	0	0	0	0	1	1	0	1	$AC \leftarrow (AC) + [M(HL)]$	Adds the contents of memory location HL to the contents of the accumulator and stores the result in the accumulator	ZF, CF
ADDR i8	Add M direct to AC	1 i7	1 i6	0 i5	0 i4	1 i3	0 i2	0 i1	1 i0	2	$AC \leftarrow (AC) + [M(i8)]$	Adds the contents of the memory location specified by immediate data i0 to i7 to the contents of the accumulator and stores the result in the accumulator	ZF, CF
ADC	Add M to AC with CF	0	0	0	0	0	0	1	0	1	$AC \leftarrow (AC) + [M(HL)] + (CF)$	Adds the contents of memory location HL to the contents of the accumulator with carry and stores the result in the accumulator	ZF, CF
ADI i4	Add immediate data to AC	1 0	1 0	0 1	0 0	1 i3	1 i2	1 i1	1 i0	2	$AC \leftarrow (AC) + i3 i2 i1 i0$	Adds immediate data i0 to i3 to the contents of the accumulator and stores the result in the accumulator	ZF
SUBC	Subtract AC from M with CF	0	0	0	1	0	1	1	1	1	$AC \leftarrow [M(HL)] - (AC) - (\overline{CF})$	Subtracts the contents of the accumulator from the contents of memory location HL with carry and stores the result in the accumulator	ZF, CF
ANDA	AND M with AC then store in AC	0	0	0	0	0	1	1	1	1	$AC \leftarrow (AC) \cdot [M(HL)]$	Takes the logical AND of the contents of the accumulator with the contents of memory location HL and stores the result in the accumulator	ZF
ORA	OR M with AC then store in AC	0	0	0	0	0	1	0	1	1	$AC \leftarrow (AC) + [M(HL)]$	Takes the logical OR of the contents of the accumulator with the contents of memory location HL and stores the result in the accumulator	ZF
EXL	Exclusive-OR M with AC then store in AC	0	0	0	1	0	1	0	1	1	$AC \leftarrow (AC) \oplus [M(HL)]$	Takes the logical exclusive-OR of the contents of the accumulator with the contents of memory location HL and stores the result in the accumulator	ZF
ANDM	AND M with AC then store in M	0	0	0	0	0	0	1	1	1	$M(HL) \leftarrow (AC) \cdot [M(HL)]$	Takes the logical AND of the contents of the accumulator with the contents of memory location HL and stores the result in the accumulator	ZF
ORM	OR M with AC then store in M	0	0	0	0	0	1	0	0	1	$M(HL) \leftarrow (AC) + [M(HL)]$	Takes the logical OR of the contents of the accumulator with the contents of memory location HL and stores the result in memory location HL	ZF

LC66556A, LC66558A

Mnemonic	Operation	Instruction code								Cycles	Notation	Description	Status flags
		D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0				
CM	Compare AC with M	0	0	0	1	0	1	1	0	1	$\overline{[M(HL)]} - (AC) - 1$	Compares the contents of the accumulator with the contents of memory location HL and sets the condition flags as shown below	ZF, CF
		Comparison		CF	ZF								
		[M(HL)] > (AC)		0	0								
		[M(HL)] = (AC)		1	1								
		[M(HL)] < (AC)		1	0								
CI i4	Compare AC with immediate data	1	1	0	0	1	1	1	1	2	$\overline{I_3 I_2 I_1 I_0} - (AC) - 1$	Compares the contents of the accumulator with immediate data I <sub>0</sub> to I <sub>3</sub> and sets the condition flags as shown below	ZF, CF
		Comparison		CF	ZF								
		I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub> > (AC)		0	0								
		I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub> = (AC)		1	1								
		I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub> < (AC)		1	0								
CLI i4	Compare DPL with immediate data	1	1	0	0	1	1	1	1	2	ZF ← 1 if (DPL) = I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub> , ZF ← 0 if (DPL) ≠ I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	Compares the contents of the lower nibble of data pointer HL with immediate data I <sub>0</sub> to I <sub>3</sub> . Sets the zero flag when equal and clears the zero flag when unequal	ZF
CMB i2	Compare AC bit with M data bit	1	1	0	0	1	1	1	1	2	ZF ← 1 if (AC, i2) = [M(HL), i2], ZF ← 0 if (AC, i2) ≠ [M(HL), i2]	Compares the bit of the accumulator specified by I <sub>0</sub> and I <sub>1</sub> with the bit of memory location HL specified by I <sub>0</sub> and I <sub>1</sub> . Sets the zero flag when equal and clears the zero flag when unequal	ZF
Load and store instructions													
LAE	Load AC and E from M2(HL)	0	1	0	1	1	1	0	0	1	AC ← M(HL), E ← M(HL + 1)	Loads the contents of memory location HL into the accumulator, and the contents of memory location HL + 1, into register E	
LAI i4	Load AC with immediate data	1	0	0	0	I <sub>3</sub>	I <sub>2</sub>	I <sub>1</sub>	I <sub>0</sub>	1	AC ← I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	Loads immediate data I <sub>0</sub> to I <sub>3</sub> into the accumulator. Vertical skip function available	ZF
LADR i8	Load AC from M direct	1	1	0	0	0	0	0	1	2	AC ← [M(i8)]	Loads the contents of the memory location specified by immediate data I <sub>0</sub> to I <sub>7</sub> into the accumulator	ZF
S	Store AC to M	0	1	0	0	0	1	1	1	1	M(HL) ← (AC)	Stores the contents of the accumulator in memory location HL	
SAE	Store AC and E to M2(HL)	0	1	0	1	1	1	1	0	1	M(HL) ← (AC), M(HL + 1) ← (E)	Stores the contents of the accumulator in memory location HL, and the contents of register E, in memory location HL + 1	
LA reg	Load AC from M(reg)	0	1	0	0	1	0	I <sub>0</sub>	0	1	AC ← [M(reg)]	Loads the contents of the memory location specified by I <sub>0</sub> into the accumulator	ZF
		Register		I <sub>0</sub>									
		HL		0									
		XY		1									
LA reg, I	Load AC from M(reg) then increment reg	0	1	0	0	1	0	I <sub>0</sub>	1	2	AC ← [M(reg)], DPL ← (DPL) + 1 or DPY ← (DPY) + 1	Loads the contents of the memory location specified by I <sub>0</sub> into the accumulator and increments the lower nibble of the corresponding memory location data pointer.	ZF
		Register		I <sub>0</sub>									
		HL		0									
		XY		1									
		The zero flag is set according to the data pointer increment operation.											

LC66556A, LC66558A

Mnemonic	Operation	Instruction code								Cycles	Notation	Description	Status flags						
		D7	D6	D5	D4	D3	D2	D1	D0										
LA reg. D	Load AC from M(reg) then decrement reg	0	1	0	1	1	0	1	0	2	$AC \leftarrow [M(\text{reg})]$ $DPL \leftarrow (DPL) - 1$ or $DPY \leftarrow (DPY) - 1$	<table border="1"> <tr> <td>Register</td> <td>t0</td> </tr> <tr> <td>HL</td> <td>0</td> </tr> <tr> <td>XY</td> <td>1</td> </tr> </table> <p>The zero flag is set according to the data pointer decrement operation.</p>	Register	t0	HL	0	XY	1	ZF
Register	t0																		
HL	0																		
XY	1																		
XA reg	Exchange AC with M(reg)	0	1	0	0	1	1	1	0	1	$(AC) \leftrightarrow [M(\text{reg})]$	<table border="1"> <tr> <td>Register</td> <td>t0</td> </tr> <tr> <td>HL</td> <td>0</td> </tr> <tr> <td>XY</td> <td>1</td> </tr> </table>	Register	t0	HL	0	XY	1	
Register	t0																		
HL	0																		
XY	1																		
XA reg. I	Exchange AC with M(reg) then increment reg	0	1	0	0	1	1	1	0	2	$(AC) \leftrightarrow [M(\text{reg})]$ $DPL \leftarrow (DPL) + 1$ or $DPY \leftarrow (DPY) + 1$	<table border="1"> <tr> <td>Register</td> <td>t0</td> </tr> <tr> <td>HL</td> <td>0</td> </tr> <tr> <td>XY</td> <td>1</td> </tr> </table> <p>The zero flag is set according to the data pointer increment operation.</p>	Register	t0	HL	0	XY	1	ZF
Register	t0																		
HL	0																		
XY	1																		
XA reg. D	Exchange AC with M(reg) then decrement reg	0	1	0	1	1	1	1	0	2	$(AC) \leftrightarrow [M(\text{reg})]$ $DPL \leftarrow (DPL) - 1$ or $DPY \leftarrow (DPY) - 1$	<table border="1"> <tr> <td>Register</td> <td>t0</td> </tr> <tr> <td>HL</td> <td>0</td> </tr> <tr> <td>XY</td> <td>1</td> </tr> </table> <p>The zero flag is set according to the data pointer decrement operation.</p>	Register	t0	HL	0	XY	1	ZF
Register	t0																		
HL	0																		
XY	1																		
XADR iB	Exchange AC with M direct	1	1	0	0	1	0	0	0	2	$(AC) \leftrightarrow [M(iB)]$	Exchanges the contents of the accumulator with the contents of the memory location specified by immediate data t0 to t7							
LEAI iB	Load E and AC with immediate data	1	1	0	0	0	1	1	0	2	$E \leftarrow [17:16:15:14]$ $AC \leftarrow [13:12:11:10]$	Loads immediate data t4 to t7 into register E, and immediate data t0 to t3, into the accumulator							
RTBL	Read table data from program ROM	0	1	0	1	1	0	1	0	2	$E, AC \leftarrow [ROM(PCh, E, AC)]$	Loads the upper nibble of the memory location specified by the program counter (the lower 8 bits are replaced by the contents of register E and the accumulator) into register E, and the upper nibble, into the accumulator							
RTBLP	Read table data from program ROM then output to P4 and P5	0	1	0	1	1	0	0	0	2	Ports 4 and 5 $\leftarrow [ROM(PCh, E, AC)]$	Loads the upper nibble of the memory location specified by the program counter (the lower 8 bits are replaced by the contents of register E and the accumulator) into port P4, and the upper nibble, into port P5							
<b>Data pointer instructions</b>																			
LDZ i4	Load DPH with zero and DPL with immediate data	0	1	1	0	1	1	1	0	1	$DPH \leftarrow 0, DPL \leftarrow [13:12:11:10]$	Clears the contents of the upper nibble of data pointer DPH, and loads immediate data t0 to t3 into the lower nibble of data pointer DPH							
LHI i4	Load DPH with immediate data	1	1	0	0	1	1	1	1	2	$DPH \leftarrow [13:12:11:10]$	Loads immediate data t0 to t3 into the upper nibble of data pointer DPH							
LLI i4	Load DPL with immediate data	1	0	0	0	1	1	1	1	2	$DPL \leftarrow [13:12:11:10]$	Loads immediate data t0 to t3 into the lower nibble of data pointer DPH							
LHLI iB	Load DPH, DPL with immediate data	1	1	0	0	0	0	0	0	2	$DPH \leftarrow [17:16:15:14]$ $DPL \leftarrow [13:12:11:10]$	Loads immediate data t0 to t7 into the data pointer DPH							
LXYI iB	Load DPX, DPY with immediate data	1	1	0	0	0	0	1	0	2	$DPX \leftarrow [17:16:15:14]$ $DPY \leftarrow [13:12:11:10]$	Loads immediate data t0 to t7 into the auxiliary data pointer DPXY							

LC66556A, LC66558A

Mnemonic	Operation	Instruction code								Cycles	Notation	Description	Status flag				
		D7	D6	D5	D4	D3	D2	D1	D0								
IL	Increment DPL	0	0	0	1	0	0	0	1	1	$DPL \leftarrow (DPL) + 1$	Increments the contents of the lower nibble of data pointer DP <sub>HL</sub> by 1	ZF				
DL	Decrement DPL	0	0	1	0	0	0	0	1	1	$DPL \leftarrow (DPL) - 1$	Decrements the contents of the lower nibble of data pointer DP <sub>HL</sub> by 1	ZF				
IY	Increment DPY	0	0	0	1	0	0	1	1	1	$DPY \leftarrow (DPY) + 1$	Increments the contents of the lower nibble of auxiliary data pointer DP <sub>XY</sub>	ZF				
DY	Decrement DPY	0	0	1	0	0	0	1	1	1	$DPY \leftarrow (DPY) - 1$	Decrements the contents of the lower nibble of auxiliary data pointer DP <sub>XY</sub>	ZF				
TAH	Transfer AC to DPH	1	1	0	0	1	1	1	0	2	$DPH \leftarrow (AC)$	Copies the contents of the accumulator into the upper nibble of data pointer DP <sub>HL</sub>					
THA	Transfer DPH to AC	1	1	0	0	1	1	1	0	2	$AC \leftarrow (DPH)$	Copies the contents of the upper nibble of data pointer DP <sub>HL</sub> into the accumulator	ZF				
XAH	Exchange AC with DPH	0	1	0	0	0	0	0	0	1	$(AC) \leftrightarrow (DPH)$	Exchanges the contents of the accumulator with the contents of the upper nibble of data pointer DP <sub>HL</sub>					
TAL	Transfer AC to DPL	1	1	0	0	1	1	1	1	2	$DPL \leftarrow (AC)$	Copies the contents of the lower nibble of data pointer DP <sub>HL</sub> into the accumulator					
TLA	Transfer DPL to AC	1	1	0	0	1	1	1	1	2	$AC \leftarrow (DPL)$	Copies the contents of the accumulator into the lower nibble of data pointer DP <sub>HL</sub>	ZF				
XAL	Exchange AC with DPL	0	1	0	0	0	0	0	1	1	$(AC) \leftrightarrow (DPL)$	Exchanges the contents of the accumulator with the contents of the lower nibble of data pointer DP <sub>HL</sub>					
TAX	Transfer AC to DPX	1	1	0	0	1	1	1	0	2	$DPX \leftarrow (AC)$	Copies the contents of the accumulator into the upper nibble of auxiliary data pointer DP <sub>XY</sub>					
TXA	Transfer DPX to AC	1	1	0	0	1	1	1	0	2	$AC \leftarrow (DPX)$	Copies the contents of the upper nibble of auxiliary data pointer DP <sub>XY</sub> into the accumulator	ZF				
XAX	Exchange AC with DPX	0	1	0	0	0	0	1	0	1	$(AC) \leftrightarrow (DPX)$	Exchanges the contents of the accumulator with the contents of the upper nibble of auxiliary data pointer DP <sub>XY</sub>					
TAY	Transfer AC to DPY	1	1	0	0	1	1	1	1	2	$DPY \leftarrow (AC)$	Copies the contents of the accumulator into the lower nibble of auxiliary data pointer DP <sub>XY</sub>					
TYA	Transfer DPY to AC	1	1	0	0	1	1	1	1	2	$AC \leftarrow (DPY)$	Copies the contents of the lower nibble of auxiliary data pointer DP <sub>XY</sub> into the accumulator	ZF				
XAY	Exchange AC with DPY	0	1	0	0	0	0	1	1	1	$(AC) \leftrightarrow (DPY)$	Exchanges the contents of the accumulator with the contents of the lower nibble of auxiliary data pointer DP <sub>XY</sub>					
<b>Flag instructions</b>																	
SFB n4	Set flag bit	0	1	1	1	n3	n2	n1	n0	1	$F_n \leftarrow 1$	Sets the flag specified by n <sub>0</sub> to n <sub>3</sub>					
RFB n4	Reset flag bit	0	0	1	1	n3	n2	n1	n0	1	$F_n \leftarrow 0$	Clears the flag specified by n <sub>0</sub> to n <sub>3</sub>	ZF				
<b>Jump and call instructions</b>																	
JMP addr	Jump in the current bank	1	1	1	0	P7	P6	P5	P4	P3	P2	P1	P0	2	$PC_{13,12} \leftarrow PC_{13,12}$ , $PC_{11} \text{ to } 0 \leftarrow P_{11} \text{ to } P_0$	Jumps within the same bank to the address specified by the program counter (the lower 12 bits are replaced by immediate data P <sub>0</sub> to P <sub>11</sub> ). After a bank instruction, PC <sub>12</sub> is complemented.	
JPEA	Jump to the address stored at E and AC in the current page	0	0	1	0									1	$PC_{13,12} \text{ to } PC_8 \leftarrow PC_{13,12}$ , $PC_7 \text{ to } 4 \leftarrow (E)$ , $PC_3 \text{ to } 0 \leftarrow (AC)$	Jumps within the current page to the address specified by the program counter (the lower 8 bits are replaced by the contents of register E and the accumulator)	
CAL addr	Call subroutine	0	1	0	1	P7	P6	P5	P4	P3	P2	P1	P0	2	$PC_{13} \text{ and } 11 \leftarrow 0$ , $PC_{10} \text{ to } 0 \leftarrow P_{10} \text{ to } P_0$ , $M4(SP) \leftarrow (CF, ZF, PC_{13} \text{ to } 0)$ , $SP \leftarrow (SP) - 4$	Calls a subroutine at the address specified by P <sub>0</sub> to P <sub>10</sub>	
CZP addr	Call subroutine in the zero page	1	0	1	0	P3	P2	P1	P0					2	$PC_{13} \text{ to } 6, PC_1 \text{ to } 0 \leftarrow 0$ , $PC_5 \text{ to } 2 \leftarrow P_3 \text{ to } P_0$ , $M4(SP) \leftarrow (CF, ZF, PC_{12} \text{ to } 0)$ , $SP \leftarrow SP - 4$	Calls a subroutine within page 0 of bank 0 at the address specified by P <sub>0</sub> to P <sub>3</sub>	
BANK	Change bank	0	0	0	1									1		Changes memory bank or register bank	

LC66556A, LC66558A

Mnemonic	Operation	Instruction code								Cycles	Notation	Description	Status flags															
		D7	D6	D5	D4	D3	D2	D1	D0																			
PUSH reg	Push reg on M2(SP)	1	1	0	0	1	1	1	1	2	M2(SP) ← (reg). SP ← (SP) - 2	Pushes the contents of the register specified by i <sub>0</sub> and i <sub>1</sub> onto the stack and decrements the stack pointer by 2																
		1	1	1	1	1	1	1	0			<table border="1"> <thead> <tr> <th>Register</th> <th>i<sub>1</sub></th> <th>i<sub>0</sub></th> </tr> </thead> <tbody> <tr> <td>HL</td> <td>0</td> <td>0</td> </tr> <tr> <td>XY</td> <td>0</td> <td>1</td> </tr> <tr> <td>AE</td> <td>1</td> <td>0</td> </tr> <tr> <td>Illegal</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	Register	i <sub>1</sub>	i <sub>0</sub>	HL	0	0	XY	0	1	AE	1	0	Illegal	1	1	
Register	i <sub>1</sub>	i <sub>0</sub>																										
HL	0	0																										
XY	0	1																										
AE	1	0																										
Illegal	1	1																										
POP reg	Pop reg off M2(SP)	1	1	0	0	1	1	1	1	2	SP ← (SP) + 2. reg ← [M2(SP)]	Increases the stack pointer by 2, pops the contents off the stack and moves the contents into the register specified by i <sub>0</sub> and i <sub>1</sub>																
		1	1	1	0	1	1	1	0			<table border="1"> <thead> <tr> <th>Register</th> <th>i<sub>1</sub></th> <th>i<sub>0</sub></th> </tr> </thead> <tbody> <tr> <td>HL</td> <td>0</td> <td>0</td> </tr> <tr> <td>XY</td> <td>0</td> <td>1</td> </tr> <tr> <td>AE</td> <td>1</td> <td>0</td> </tr> <tr> <td>Illegal</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	Register	i <sub>1</sub>	i <sub>0</sub>	HL	0	0	XY	0	1	AE	1	0	Illegal	1	1	
Register	i <sub>1</sub>	i <sub>0</sub>																										
HL	0	0																										
XY	0	1																										
AE	1	0																										
Illegal	1	1																										
RT	Return from subroutine	0	0	0	1	1	1	0	0	2	SP ← (SP) + 4. PC ← [M4(SP)]	Returns from a subroutine or an interrupt service routine, but does not restore the state of the zero and carry flags																
RTI	Return from interrupt subroutine	0	0	0	1	1	1	0	1	2	SP ← (SP) + 4. PC ← [M4(SP)], CF, ZF ← [M4(SP)]	Returns from a subroutine or an interrupt service routine, including the state of the zero and carry flags	ZF, CF															
Branch Instructions																												
BAI2 addr	Branch on AC bit	1	1	0	1	0	0	1	1	2	PC7 to 0 ← P7 P6 P5 P4 P3 P2 P1 P0 if (AC, i <sub>2</sub> ) = 1	Tests the bit specified by immediate data i <sub>0</sub> and i <sub>1</sub> of the accumulator and, if set, branches within the same page to the address specified by P <sub>0</sub> to P <sub>7</sub>																
		P	P	P	P	P	P	P	P																			
		7	6	5	4	3	2	1	0																			
BNA2 addr	Branch on no AC bit	1	0	0	1	0	0	1	1	2	PC7 to 0 ← P7 P6 P5 P4 P3 P2 P1 P0 if (AC, i <sub>2</sub> ) = 0	Tests the bit specified by immediate data i <sub>0</sub> and i <sub>1</sub> of the accumulator and, if not set, branches within the same page to the address specified by P <sub>0</sub> to P <sub>7</sub>																
		P	P	P	P	P	P	P	P																			
		7	6	5	4	3	2	1	0																			
BMI2 addr	Branch on M bit	1	1	0	1	0	1	1	1	2	PC7 to 0 ← P7 P6 P5 P4 P3 P2 P1 P0 if [M(HL), i <sub>2</sub> ] = 1	Tests the bit specified by immediate data i <sub>0</sub> and i <sub>1</sub> of memory location HL and, if set, branches within the same page to the address specified by P <sub>0</sub> to P <sub>7</sub>																
		P	P	P	P	P	P	P	P																			
		7	6	5	4	3	2	1	0																			
BNM2 addr	Branch on no M bit	1	0	0	1	0	1	1	1	2	PC7 to 0 ← P7 P6 P5 P4 P3 P2 P1 P0 if [M(HL), i <sub>2</sub> ] = 0	Tests the bit specified by immediate data i <sub>0</sub> and i <sub>1</sub> of memory location HL and, if not set, branches within the same page to the address specified by P <sub>0</sub> to P <sub>7</sub>																
		P	P	P	P	P	P	P	P																			
		7	6	5	4	3	2	1	0																			
BPI2 addr	Branch on port bit	1	1	0	1	1	0	1	1	2	PC7 to 0 ← P7 P6 P5 P4 P3 P2 P1 P0 if [P(DP <sub>L</sub> ), i <sub>2</sub> ] = 1	Tests the bit specified by immediate data i <sub>0</sub> and i <sub>1</sub> of the port specified by the contents of the lower nibble of data pointer DP <sub>HL</sub> and, if set, branches within the same page to the address specified by P <sub>0</sub> to P <sub>7</sub> . When followed by bank instructions, the internal control registers are also valid (read-only registers).																
		P	P	P	P	P	P	P	P																			
		7	6	5	4	3	2	1	0																			
BNP2 addr	Branch on no port bit	1	0	0	1	1	0	1	1	2	PC7 to 0 ← P7 P6 P5 P4 P3 P2 P1 P0 if [P(DP <sub>L</sub> ), i <sub>2</sub> ] = 0	Tests the bit specified by immediate data i <sub>0</sub> and i <sub>1</sub> of the port specified by the contents of the lower nibble of data pointer DP <sub>HL</sub> and, if not set, branches within the same page to the address specified by P <sub>0</sub> to P <sub>7</sub> . When followed by bank instructions, the internal control registers are also valid (read-only registers).																
		P	P	P	P	P	P	P	P																			
		7	6	5	4	3	2	1	0																			
BC addr	Branch on CF	1	1	0	1	1	1	0	0	2	PC7 to 0 ← P7 P6 P5 P4 P3 P2 P1 P0 if (CF) = 1	Tests the carry flag and, if set, branches within the same page to the address specified by P <sub>0</sub> to P <sub>7</sub>																
		P	P	P	P	P	P	P	P																			
		7	6	5	4	3	2	1	0																			
BNC addr	Branch on no CF	1	0	0	1	1	1	0	0	2	PC7 to 0 ← P7 P6 P5 P4 P3 P2 P1 P0 if (CF) = 0	Tests the carry flag and, if not set, branches within the same page to the address specified by P <sub>0</sub> to P <sub>7</sub>																
		P	P	P	P	P	P	P	P																			
		7	6	5	4	3	2	1	0																			
BZ addr	Branch on ZF	1	1	0	1	1	1	0	1	2	PC7 to 0 ← P7 P6 P5 P4 P3 P2 P1 P0 if (ZF) = 1	Tests the zero flag and, if set, branches within the same page to the address specified by P <sub>0</sub> to P <sub>7</sub>																
		P	P	P	P	P	P	P	P																			
		7	6	5	4	3	2	1	0																			
BNZ addr	Branch on no ZF	1	0	0	1	1	1	0	1	2	PC7 to 0 ← P7 P6 P5 P4 P3 P2 P1 P0 if (ZF) = 0	Tests the zero flag and, if not set, branches within the same page to the address specified by P <sub>0</sub> to P <sub>7</sub>																
		P	P	P	P	P	P	P	P																			
		7	6	5	4	3	2	1	0																			



LC66556A, LC66558A

Mnemonic	Operation	Instruction code								Cycles	Notation	Description	Status flags
		D7	D6	D5	D4	D3	D2	D1	D0				
BFn4 addr	Branch on flag bit	1 P 7	1 P 6	1 P 5	1 P 4	n 3 P 3	n 2 P 2	n 1 P 1	n 0 P 0	2	PC7 to 0 ← P7 P6 P5 P4 P3 P2 P1 P0 if (Fn) = 1	Tests the flag specified by n0 to n3 and, if set, branches within the same page to the address specified by P0 to P7	
BNFn4 addr	Branch on no flag bit	1 P 7	0 P 6	1 P 5	1 P 4	n 3 P 3	n 2 P 2	n 1 P 1	n 0 P 0	2	PC7 to 0 ← P7 P6 P5 P4 P3 P2 P1 P0 if (Fn) = 0	Tests the flag specified by n0 to n3 and, if not set, branches within the same page to the address specified by P0 to P7	
Input and output instructions													
IP0	Input port 0 to AC	0	0	1	0	0	0	0	0	1	AC ← (P0)	Loads the contents of input port P0 into the accumulator	ZF
IP	Input port to AC	0	0	1	0	0	1	1	0	1	AC ← [P(DPL)]	Loads the contents of the input port specified by the contents of the lower nibble of data pointer DP <sub>HL</sub> into the accumulator	ZF
IPM	Input port to M	0	0	0	1	1	0	0	1	1	M(HL) ← [P(DPL)]	Loads the contents of the input port specified by the contents of the lower nibble of data pointer DP <sub>HL</sub> into memory location HL	
IPDR i4	Input port to AC direct	1 0	1 1	0 1	0 1	1 1 <sub>3</sub>	1 1 <sub>2</sub>	1 1 <sub>1</sub>	1 1 <sub>0</sub>	2	AC ← [P(i4)]	Loads the contents of the input port specified by immediate data i0 to i3 into the accumulator	ZF
IP45	Input ports 4 and 5 to E and AC, respectively	1 1	1 1	0 0	0 1	1 0	1 1	1 0	1 0	2	E ← [P(4)], AC ← [P(5)]	Loads the contents of input ports P4 and P5 into register E and the accumulator, respectively	
OP	Output AC to port	0	0	1	0	0	1	0	1	1	P(DPL) ← (AC)	Loads the contents of the accumulator into the output port specified by the contents of the lower nibble of data pointer DP <sub>HL</sub>	
OPM	Output M to port	0	0	0	1	1	0	1	0	1	P(DPL) ← [M(HL)]	Loads the contents of memory location HL into the output port specified by the lower nibble of data pointer DP <sub>HL</sub>	
OPDR i4	Output AC to port direct	1 0	1 1	0 1	0 1	1 1 <sub>3</sub>	1 1 <sub>2</sub>	1 1 <sub>1</sub>	1 1 <sub>0</sub>	2	P(i4) ← (AC)	Loads the contents of the accumulator into the output port specified by immediate data i0 to i3	
OP45	Output E and AC to ports 4 and 5, respectively	1 1	1 1	0 0	0 1	1 0	1 1	1 0	1 1	2	P(4) ← (E), P(5) ← (AC)	Loads the contents of register E and the accumulator into ports P4 and P5, respectively	
SPB i2	Set port bit	0	0	0	0	1	0	1 <sub>1</sub>	1 <sub>0</sub>	1	[P(DPL), i2] ← 1	Sets the bit specified by immediate data i0 and i1 of the output port specified by the lower nibble of data pointer DP <sub>HL</sub>	
RPB i2	Reset port bit	0	0	1	0	1	0	1 <sub>1</sub>	1 <sub>0</sub>	1	[P(DPL), i2] ← 0	Clears the bit specified by immediate data i0 and i1 of the output port specified by the lower nibble of data pointer DP <sub>HL</sub>	ZF
ANDPDR i4, p4	AND port with immediate data then output	1 i3	1 i2	0 i1	0 i0	0 P 3	1 P 2	0 P 1	1 P 0	2	P(P3 to P0) ← [P(P3 to 0)] + i3 to 0	Takes the logical AND of the contents of the port specified by P0 to P3 with immediate data i0 to i3 and loads the result into the port	ZF
ORPDR i4, p4	OR port with immediate data then output	1 i3	1 i2	0 i1	0 i0	0 P 3	1 P 2	0 P 1	0 P 0	2	P(P3 to P0) ← [P(P3 to 0)] + i3 to 0	Takes the logical OR of the contents of the port specified by P0 to P3 with immediate data i0 to i3 and loads the result into the port	ZF
Timer control instructions													
WTTM0	Write timer 0	1	1	0	0	1	0	1	0	2	TIMER0 ← [M2(HL)] and (AC)	Writes the contents of memory locations HL and HL + 1 and the contents of the accumulator into the timer 0 reload register	
WTTM1	Write timer 1	1 1	1 1	0 1	0 1	1 0	1 1	1 0	1 0	2	TIMER1 ← (E) and (AC)	Writes the contents of register E and the accumulator into the timer 0 reload register	
RTIM0	Read timer 0	1	1	0	0	1	0	1	1	2	M2(HL) and AC ← (TIMER0)	Reads the contents of the timer 0 counter into memory locations HL and HL + 1 and the accumulator	
RTIM1	Read timer 1	1 1	1 1	0 1	0 1	1 0	1 1	1 0	1 1	2	E and AC ← (TIMER1)	Reads the contents of the timer 1 counter into register E and the accumulator	
START0	Start timer 0	1 1	1 1	0 1	0 0	1 0	1 1	1 1	1 0	2	Start timer 0 counter	Starts the timer 0 counter	
START1	Start timer 1	1 1	1 1	0 1	0 0	1 0	1 1	1 1	1 1	2	Start timer 1 counter	Starts the timer 1 counter	
STOP0	Stop timer 0	1 1	1 1	0 1	0 1	1 0	1 1	1 1	1 0	2	Stop timer 0 counter	Stops the timer 0 counter	
STOP1	Stop timer 1	1 1	1 1	0 1	0 1	1 0	1 1	1 1	1 1	2	Stop timer 1 counter	Stops the timer 1 counter	

LC66556A, LC66558A

Mnemonic	Operation	Instruction code								Cycles	Notation	Description	Status flags
		D7	D6	D5	D4	D3	D2	D1	D0				
<b>Interrupt control instructions</b>													
MSET	Set Interrupt Master Enable flag	1 0	1 1	0 0	0 1	1 0	1 0	0 0	1 0	2	$MSE \leftarrow 1$	Sets the interrupt master enable flag	
MRESET	Reset Interrupt Master Enable flag	1 1	1 0	0 0	0 1	1 0	1 0	0 0	1 0	2	$MSE \leftarrow 0$	Clears the interrupt master enable flag	
EIH i4	Enable interrupt HIGH	1 0	1 1	0 0	0 1	1 i <sub>3</sub>	1 i <sub>2</sub>	0 i <sub>1</sub>	1 i <sub>0</sub>	2	$EDIH \leftarrow (EDIH) + i4$	Takes the logical OR of the interrupt mask with immediate data i <sub>0</sub> to i <sub>3</sub> and, for bits that are set, enables the corresponding active-HIGH interrupts	
EIL i4	Enable interrupt LOW	1 0	1 1	0 0	0 1	1 i <sub>3</sub>	1 i <sub>2</sub>	0 i <sub>1</sub>	1 i <sub>0</sub>	2	$EDIL \leftarrow (EDIL) + i4$	Takes the logical OR of the interrupt mask with immediate data i <sub>0</sub> to i <sub>3</sub> and, for bits that are set, enables the corresponding active-LOW interrupts	
DIH i4	Disable interrupt HIGH	1 1	1 0	0 0	0 1	1 i <sub>3</sub>	1 i <sub>2</sub>	0 i <sub>1</sub>	1 i <sub>0</sub>	2	$EDIH \leftarrow (EDIH) \cdot \bar{i4}$	Takes the logical AND of the interrupt mask with the 1s complement of immediate data i <sub>0</sub> to i <sub>3</sub> and, for bits that are set, disables the corresponding active-HIGH interrupts	ZF
DIL i4	Disable interrupt LOW	1 1	1 0	0 0	0 1	1 i <sub>3</sub>	1 i <sub>2</sub>	0 i <sub>1</sub>	1 i <sub>0</sub>	2	$EDIL \leftarrow (EDIL) \cdot \bar{i4}$	Takes the logical AND of the interrupt mask with the 1s complement of immediate data i <sub>0</sub> to i <sub>3</sub> and, for bits that are set, disables the corresponding active-LOW interrupts	ZF
WTSP	Write SP	1 1	1 1	0 0	0 1	1 1	1 0	1 1	1 0	2	$SP \leftarrow (E), (AC)$	Writes the contents of register E and the accumulator into the stack pointer	
RSP	Read SP	1 1	1 1	0 0	0 1	1 1	1 0	1 1	1 1	2	$E, AC \leftarrow (SP)$	Reads the contents of the stack pointer into register E and the accumulator	
<b>Standby control instructions</b>													
HALT	Halt	1 1	1 1	0 0	0 1	1 1	1 1	1 1	1 0	2	Halt	Selects halt mode	
HOLD	Hold	1 1	1 1	0 0	0 1	1 1	1 1	1 1	1 1	2	Hold	Selects hold mode	
<b>Serial input/output control instructions</b>													
STARTS	Start serial I/O	1 1	1 1	0 1	0 1	1 1	1 1	1 1	1 0	2	START SIO	Starts serial input/output operation	
WTSIO	Write serial I/O	1 1	1 1	0 1	0 1	1 1	1 1	1 1	1 1	2	$SIO \leftarrow (E) \text{ and } (AC)$	Writes the contents of register E and the accumulator into the serial input/output port	
RSIO	Read serial I/O	1 1	1 1	0 1	0 1	1 1	1 1	1 1	1 1	2	$E \text{ and } AC \leftarrow (SIO)$	Reads the serial input/output port into register E and the accumulator	
<b>Miscellaneous instructions</b>													
NOP	No operation	0	0	0	0	0	0	0	0	1	No operation	No operation for one cycle	
SB i2	Select bank	1 1	1 1	0 0	0 0	1 0	1 0	1 i <sub>1</sub>	1 i <sub>0</sub>	2	$PC13, 12 \leftarrow i1 \ i0$	Selects the memory bank specified by immediate data i <sub>0</sub> and i <sub>1</sub>	

Note

The LC66556A and LC66558A do not support the SB instruction. Only the LC66E516, LC66P516, LC66599 and LC66PG5XX support the SB instruction.

■ No products described or contained herein are intended for use in surgical implants, life-support systems, aerospace equipment, nuclear power control systems, vehicles, disaster/crime-prevention equipment and the like, the failure of which may directly or indirectly cause injury, death or property loss.

■ Anyone purchasing any products described or contained herein for an above-mentioned use shall:

- ① Accept full responsibility and indemnify and defend SANYO ELECTRIC CO., LTD., its affiliates, subsidiaries and distributors and all their officers and employees, jointly and severally, against any and all claims and litigation and all damages, cost and expenses associated with such use;
- ② Not impose any responsibility for any fault or negligence which may be cited in any such claim or litigation on SANYO ELECTRIC CO., LTD., its affiliates, subsidiaries and distributors or any of their officers and employees jointly or severally.

■ Information (including circuit diagrams and circuit parameters) herein is for example only; it is not guaranteed for volume production. SANYO believes information herein is accurate and reliable, but no guarantees are made or implied regarding its use or any infringements of intellectual property rights or other rights of third parties.