



## Four-Bit Single-Chip Microcontrollers with 4, 6, and 8 KB of On-Chip ROM

### Preliminary

### Overview

The LC66354C, LC66356C, and LC66358C are 4-bit CMOS microcontrollers that integrate on a single chip all the functions required in a system controller, including ROM, RAM, I/O ports, a serial interface, comparator inputs, three-value inputs, timers, and interrupt functions. These three microcontrollers are available in a 42-pin package.

These products differ from the earlier LC66358A Series and LC66358B Series in the power-supply voltage range, the operating speed, and other points.

### Features and Functions

- On-chip ROM capacities of 4, 6, and 8 kilobytes, and an on-chip RAM capacity of  $512 \times 4$  bits.
- Fully supports the LC66000 Series common instruction set (128 instructions).
- I/O ports: 36 pins
- 8-bit serial interface: two circuits (can be connected in cascade to form a 16-bit interface)
- Instruction cycle time: 0.92 to 10  $\mu$ s (at 2.5 to 5.5 V)
  - For the earlier LC66358A Series: 1.96 to 10  $\mu$ s (at 3.0 to 5.5 V) and 3.92 to 10  $\mu$ s (at 2.2 to 5.5 V)
  - For the earlier LC66358B Series: 0.92 to 10  $\mu$ s (at 3.0 to 5.5 V)
- Powerful timer functions and prescalers
  - Time limit timer, event counter, pulse width measurement, and square wave output using a 12-bit timer.
  - Time limit timer, event counter, PWM output, and square wave output using an 8-bit timer.
  - Time base function using a 12-bit prescaler.
- Powerful interrupt system with 8 interrupt factors and 8 interrupt vector locations.
  - External interrupts: 3 factors/3 vector locations
  - Internal interrupts: 5 factors/5 vector locations
- Flexible I/O functions  
Comparator inputs, three-value inputs, 20-mA drive outputs, 15-V high-voltage pins, and pull-up/open-drain options.
- Optional runaway detection function (watchdog timer)
- 8-bit I/O functions
- Power saving functions using halt and hold modes.
- Packages: DIP42S, QIP48E (QFP48E)

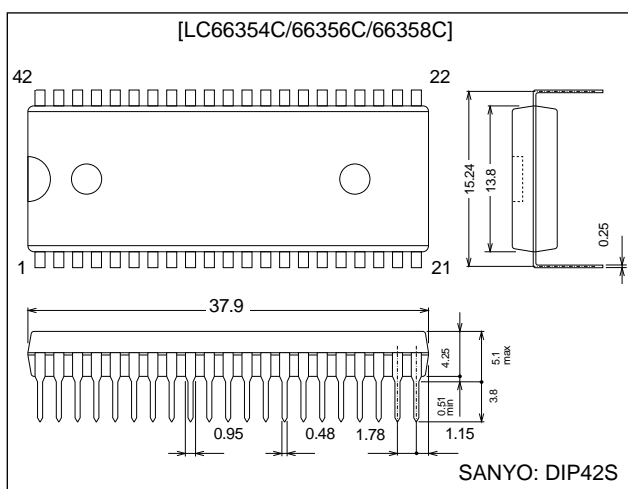
### • Evaluation LSIs

- LC66599 (evaluation chip) + EVA85/800-TB6630X
- LC66E308 (on-chip EPROM microcontroller) used together.

### Package Dimensions

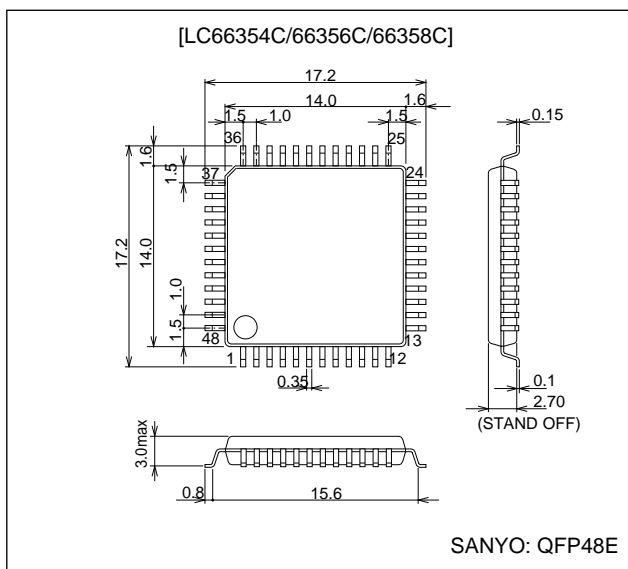
unit: mm

#### 3025B-DIP42S



unit: mm

#### 3156-QFP48E

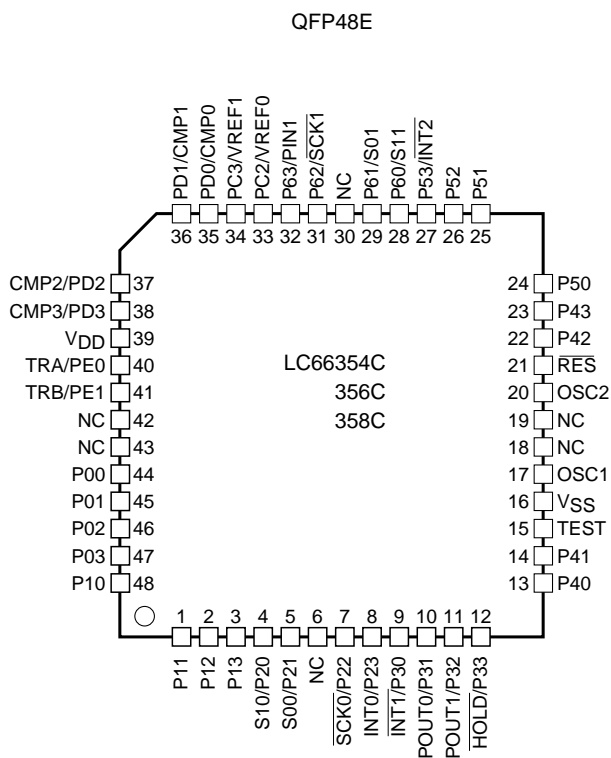
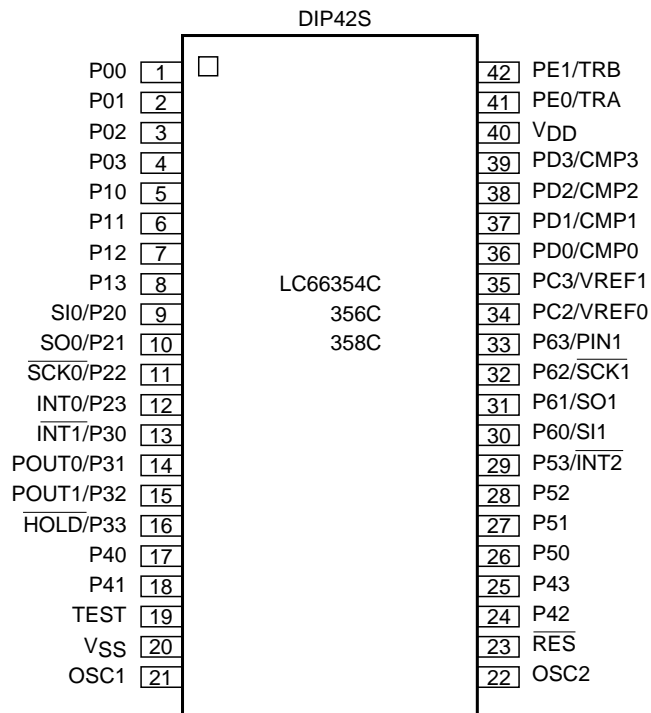


## Series Organization

Type No.	No. of pins	ROM capacity	RAM capacity	Package		Features
LC66304A/306A/308A	42	4 K/6 K/8 KB	512 W	DIP42S	QFP48E	Normal versions 4.0 to 6.0 V/0.92 $\mu$ s
LC66404A/406A/408A	42	4 K/6 K/8 KB	512 W	DIP42S	QFP48E	
LC66506B/508B/512B/516B	64	6 K/8 K/12 K/16 KB	512 W	DIP64S	QFP64A	
LC66354A/356A/358A	42	4 K/6 K/8 KB	512 W	DIP42S	QFP48E	Low-voltage versions 2.2 to 5.5 V/3.92 $\mu$ s
LC66354S/356S/358S	42	4 K/6 K/8 KB	512 W		QFP44M	
LC66556A/558A/562A/566A	64	6 K/8 K/12 K/16 KB	512 W	DIP64S	QFP64E	Low-voltage high-speed versions 3.0 to 5.5 V/0.92 $\mu$ s
LC66354B/356B/358B	42	4 K/6 K/8 KB	512 W	DIP42S	QFP48E	
LC66556B/558B	64	6 K/8 KB	512 W	DIP64S	QFP64E	
LC66562B/566B	64	12 K/16 KB	512 W	DIP64S	QFP64E	----- 2.5 to 5.5 V/0.92 $\mu$ s
LC66354C/356C/358C	42	4 K/6 K/8 KB	512 W	DIP42S	QFP48E	
LC662304A/2306A/2308A	42	4 K/6 K/8 KB	512 W	DIP42S	QFP48E	On-chip DTMF generator versions
LC662312A/2316A	42	12 K/16 KB	512 W	DIP42S	QFP48E	3.0 to 5.5 V/0.95 $\mu$ s
LC665304A/665306A/665308A	48	4 K/6 K/8 KB	512 W	DIP48S	QFP48E	Dual oscillator support 3.0 to 5.5 V/0.95 $\mu$ s
LC665312A/5316A	48	12 K/16 KB	512 W	DIP48S	QFP48E	
LC66E308	42	EPROM 8 KB	512 W	DIC42S with window	QFC48 with window	Window and OTP evaluation versions 4.5 to 5.5 V/0.92 $\mu$ s
LC66P308	42	OTPROM 8 KB	512 W	DIP42S	QFP48E	
LC66E408	42	EPROM 8 KB	512 W	DIC42S with window	QFC48 with window	
LC66P408	42	OTPROM 8 KB	512 W	DIP42S	QFP48E	
LC66E516	64	EPROM 16 KB	512 W	DIC64S with window	QFC64 with window	
LC66P516	64	OTPROM 16 KB	512 W	DIP64S	QFP64E	
LC66E2316	42	EPROM 16 KB	512 W	DIC42S with window	QFC48 with window	4.5 to 5.5 V/0.95 $\mu$ s
LC66E5316	52/48	EPROM 16 KB	512 W	DIC52S with window	QFC48 with window	
LC66P2316*	42	OTPROM 16 KB	512 W	DIP42S	QFP48E	4.0 to 5.5 V/0.95 $\mu$ s
LC66P5316	48	OTPROM 16 KB	512 W	DIP48S	QFP48E	

Note: \* Under development

Pin Assignments

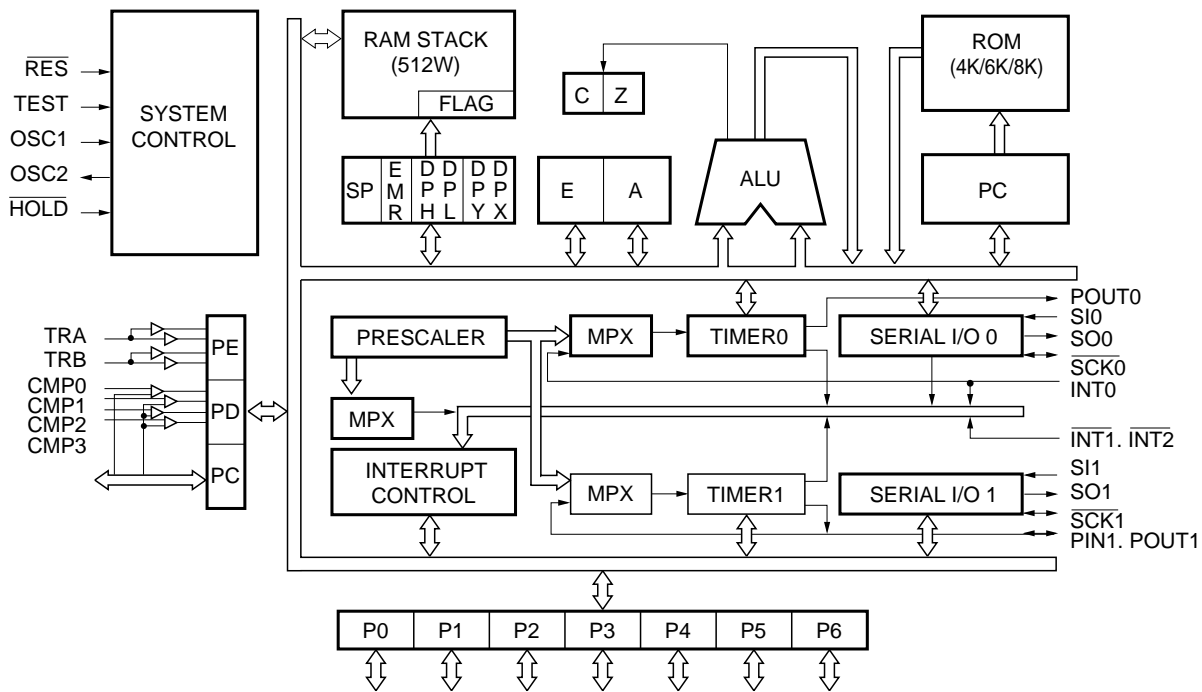


Top view

We recommend the use of reflow-soldering techniques to solder-mount QFP packages.

Please consult with your Sanyo representative for details on process conditions if the package itself is to be directly immersed in a dip-soldering bath (dip-soldering techniques).

**System Block Diagram**



**Differences between the LC66354C, LC66356C, and LC66358C and the LC6630X Series**

Item	LC6630X Series (Including the LC66599 evaluation chip)	LC6635XC Series
System differences Hardware wait time (number of cycles) when hold mode is cleared	65536 cycles About 64 ms at 4 MHz (T <sub>cyc</sub> = 1 μs)	16384 cycles About 16 ms at 4 MHz (T <sub>cyc</sub> = 1 μs)
Value of timer 0 after a reset (Including the value after hold mode is cleared)	Set to FF0.	Set to FFC.
Difference in major features Operating power-supply voltage and operating speed (cycle time)	<ul style="list-style-type: none"> <li>LC66304A/306A/308A 4.0 to 6.0 V/0.92 to 10 μs</li> <li>LC66E308/P308 4.5 to 5.5 V/0.92 to 10 μs</li> </ul>	<ul style="list-style-type: none"> <li>2.5 to 5.5 V/0.92 to 10 μs</li> <li>LC6635XA 2.2 to 5.5 V/3.92 to 10 μs</li> <li>3.0 to 5.5 V/1.96 to 10 μs</li> <li>LC6635XB 3.0 to 5.5 V/0.92 to 10 μs</li> </ul>

- Note: 1. An RC oscillator cannot be used with the LC66354C, LC66356C, and LC66358C.  
 2. There are other differences, including differences in output currents and port input voltages. For details, see the data sheets for the LC66308A, LC66E308, and LC66P308.  
 3. Pay close attention to the differences listed here when using the LC66E308 and LC66P308 for evaluation.

## Pin Function Overview

Pin	I/O	Overview	Output driver type	Options	State after a reset
P00 P01 P02 P03	I/O	I/O ports P00 to P03 <ul style="list-style-type: none"> <li>Input or output in 4-bit or 1-bit units</li> <li>P00 to P03 support the halt mode control function</li> </ul>	<ul style="list-style-type: none"> <li>Pch: Pull-up MOS type</li> <li>Nch: Intermediate sink current type</li> </ul>	<ul style="list-style-type: none"> <li>Pull-up MOS or Nch OD output</li> <li>Output level on reset</li> </ul>	High or low (option)
P10 P11 P12 P13	I/O	I/O ports P10 to P13 <ul style="list-style-type: none"> <li>Input or output in 4-bit or 1-bit units</li> </ul>	<ul style="list-style-type: none"> <li>Pch: Pull-up MOS type</li> <li>Nch: Intermediate sink current type</li> </ul>	<ul style="list-style-type: none"> <li>Pull-up MOS or Nch OD output</li> <li>Output level on reset</li> </ul>	High or low (option)
P20/SI0 P21/SO0 P22/SCK0 P23/INT0	I/O	I/O ports P20 to P23 <ul style="list-style-type: none"> <li>Input or output in 4-bit or 1-bit units</li> <li>P20 is also used as the serial input SI0 pin.</li> <li>P21 is also used as the serial output SO0 pin.</li> <li>P22 is also used as the serial clock SCK0 pin.</li> <li>P23 is also used as the INT0 interrupt request pin, and also as the timer 0 event counting and pulse width measurement input.</li> </ul>	<ul style="list-style-type: none"> <li>Pch: CMOS type</li> <li>Nch: Intermediate sink current type</li> <li>Nch: +15-V handling when OD option selected</li> </ul>	CMOS or Nch OD output	H
P30/INT1 P31/POUT0 P32/POUT1	I/O	I/O ports P30 to P32 <ul style="list-style-type: none"> <li>Input or output in 3-bit or 1-bit units</li> <li>P30 is also used as the INT1 interrupt request.</li> <li>P31 is also used for the square wave output from timer 0.</li> <li>P32 is also used for the square wave output from timer 1.</li> </ul>	<ul style="list-style-type: none"> <li>Pch: CMOS type</li> <li>Nch: Intermediate sink current type</li> <li>Nch: +15-V handling when OD option selected</li> </ul>	CMOS or Nch OD output	H
P33/HOLD	I	Hold mode control input <ul style="list-style-type: none"> <li>Hold mode is set up by the HOLD instruction when HOLD is low.</li> <li>In hold mode, the CPU is restarted by setting HOLD to the high level.</li> <li>This pin can be used as input port P33 along with P30 to P32.</li> <li>When the P33/HOLD pin is at the low level, the CPU will not be reset by a low level on the RES pin. Therefore, applications must not set P33/HOLD low when power is first applied.</li> </ul>			
P40 P41 P42 P43	I/O	I/O ports P40 to P43 <ul style="list-style-type: none"> <li>Input or output in 4-bit or 1-bit units</li> <li>Input or output in 8-bit units when used in conjunction with P50 to P53.</li> <li>Can be used for output of 8-bit ROM data when used in conjunction with P50 to P53.</li> </ul>	<ul style="list-style-type: none"> <li>Pch: Pull-up MOS type</li> <li>Nch: Intermediate sink current type</li> <li>Nch: +15-V handling when OD option selected</li> </ul>	Pull-up MOS or Nch OD output	H
P50 P51 P52 P53/INT2	I/O	I/O ports P50 to P53 <ul style="list-style-type: none"> <li>Input or output in 4-bit or 1-bit units</li> <li>Input or output in 8-bit units when used in conjunction with P40 to P43.</li> <li>Can be used for output of 8-bit ROM data when used in conjunction with P40 to P43.</li> <li>P53 is also used as the INT2 interrupt request.</li> </ul>	<ul style="list-style-type: none"> <li>Pch: Pull-up MOS type</li> <li>Nch: Intermediate sink current type</li> <li>Nch: +15-V handling when OD option selected</li> </ul>	Pull-up MOS or Nch OD output	H

Continued on next page.

Continued from preceding page.

Pin	I/O	Overview	Output driver type	Options	State after a reset
P60/SI0 P61/SO1 P62/SCK1 P63/PIN1	I/O	I/O ports P60 to P63 <ul style="list-style-type: none"> <li>Input or output in 4-bit or 1-bit units</li> <li>P60 is also used as the serial input S11 pin.</li> <li>P61 is also used as the serial output SO1 pin.</li> <li>P62 is also used as the serial clock SCK1 pin.</li> <li>P63 is also used for the event count input to timer 1.</li> </ul>	<ul style="list-style-type: none"> <li>Pch: CMOS type</li> <li>Nch: Intermediate sink current type</li> <li>Nch: +15-V handling when OD option selected</li> </ul>	CMOS or Nch OD output	H
PC2/VREF0 PC3/VREF1	I/O	I/O ports PC2 and PC3 <ul style="list-style-type: none"> <li>Input or output in 2-bit or 1-bit units</li> <li>PC2 is also used as the VREF0 comparator comparison voltage pin.</li> <li>PC3 is also used as the VREF1 comparator comparison voltage pin.</li> </ul>	<ul style="list-style-type: none"> <li>Pch: CMOS type</li> <li>Nch: Intermediate sink current type</li> </ul>	CMOS or Nch OD output	H
PD0/CMP0 PD1/CMP1 PD2/CMP2 PD3/CMP3	I	Dedicated input ports PD0 to PD3 <ul style="list-style-type: none"> <li>These pins can be switched in software to function as comparator inputs.</li> <li>The comparison voltage for PD0 is provided by VREF0.</li> <li>The comparison voltage for PD1 to PD3 is provided by VREF1.</li> <li>Pins PD0 and PD1 can be set to the comparator function individually, but pins PD2 and PD3 are set together.</li> </ul>			Normal input
PE0/TRA PE1/TRB	I	Dedicated input ports These pins can be switched in software to function as three-value inputs.			Normal input
OSC1 OSC2	I O	System clock oscillator connections When an external clock is used, leave OSC2 open and connect the clock signal to OSC1.		Use of either a ceramic oscillator or an external clock can be selected.	
$\overline{\text{RES}}$	I	System reset input When the P33/HOLD pin is at the high level, a low level input to the RES pin will initialize the CPU.			
TEST	I	CPU test pin This pin must be connected to $V_{SS}$ during normal operation.			
$V_{DD}$ $V_{SS}$		Power supply pins			

Note: Pull-up MOS type: The output circuit includes a MOS transistor that pulls the pin up to  $V_{DD}$ .

CMOS output: Complementary output.

OD output: Open-drain output.

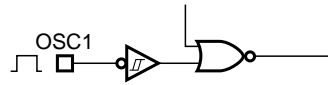
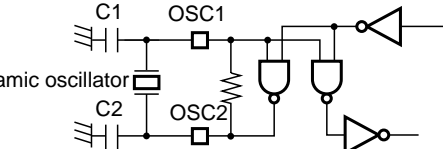
## User Options

### 1. Port 0 and 1 output level at reset option

The output levels at reset for I/O ports 0 and 1, in independent 4-bit groups, can be selected from the following two options.

Option	Conditions and notes
1. Output high at reset	The four bits of ports 0 or 1 are set in a group
2. Output low at reset	The four bits of ports 0 or 1 are set in a group

### 2. Oscillator circuit options

Option	Circuit	Conditions and notes
1. External clock		The input has Schmitt characteristics
2. Ceramic oscillator		

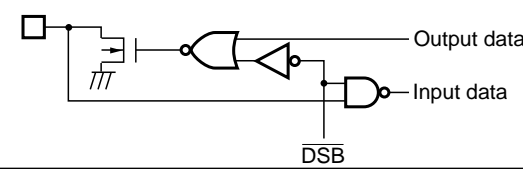
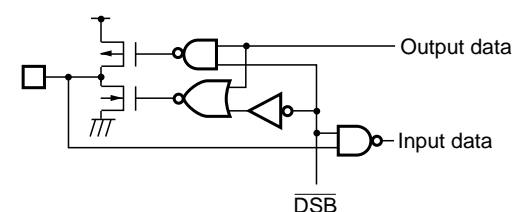
Note: There is no RC oscillator option.

### 3. Watchdog timer option

A runaway detection function (watchdog timer) can be selected as an option.

### 4. Port output type options

- The output type of each bit (pin) in ports P0, P1, P2, P3 (except for the P33/ $\overline{\text{HOLD}}$  pin), P4, P5, P6, and PC can be selected individually from the following two options.

Option	Circuit	Conditions and notes
1. Open-drain output		The port P2, P3, P5, and P6 inputs have Schmitt characteristics.
2. Output with built-in pull-up resistor		The port P2, P3, P5, and P6 inputs have Schmitt characteristics. The CMOS outputs (ports P2, P3, P6, and PC) and the pull-up MOS outputs (P0, P1, P4, and P5) are distinguished by the drive capacity of the p-channel transistor.

- The port PD comparator input and the port PE three-value input are selected in software.

## Specifications

### Absolute Maximum Ratings at $T_a = 25^\circ\text{C}$ , $V_{SS} = 0\text{ V}$

Parameter	Symbol	Conditions	Ratings	Unit	Note	
Maximum supply voltage	$V_{DD\text{ max}}$	$V_{DD}$	-0.3 to +7.0	V		
Input voltage	$V_{IN1}$	P2, P3 (except for the P33/HOLD pin), P4, P5, and P6	-0.3 to +15.0	V	1	
	$V_{IN2}$	All other inputs	-0.3 to $V_{DD} + 0.3$	V	2	
Output voltage	$V_{OUT1}$	P2, P3 (except for the P33/HOLD pin), P4, P5, and P6	-0.3 to +15.0	V	1	
	$V_{OUT2}$	All other inputs	-0.3 to $V_{DD} + 0.3$	V	2	
Output current per pin	$I_{ON}$	P0, P1, P2, P3 (except for the P33/HOLD pin), P4, P5, P6, and PC	20	mA	3	
	$-I_{OP1}$	P0, P1, P4, P5	2	mA	4	
	$-I_{OP2}$	P2, P3 (except for the P33/HOLD pin), P6, and PC	4	mA	4	
Total pin current	$\Sigma I_{ON1}$	P0, P1, P2, P3 (except for the P33/HOLD pin), P40, and P41	75	mA	3	
	$\Sigma I_{ON2}$	P5, P6, P42, P43, PC	75	mA	3	
	$\Sigma I_{OP1}$	P0, P1, P2, P3 (except for the P33/HOLD pin), P40, and P41	25	mA	4	
	$\Sigma I_{OP2}$	P5, P6, P42, P43, PC	25	mA	4	
Allowable power dissipation	$P_d\text{ max}$	$T_a = -30\text{ to }+70^\circ\text{C}$	DIP42S	600	mW	
			QFP48E	430	mW	5
Operating temperature	$T_{opr}$		-30 to +70	$^\circ\text{C}$		
Storage temperature	$T_{stg}$		-55 to +125	$^\circ\text{C}$		

Note: 1. Applies to pins with open-drain output specifications. For pins with other than open-drain output specifications, the ratings in the pin column for that pin apply.

2. For the oscillator input and output pins, levels up to the free-running oscillation level are allowed.

3. Sink current

4. Source current (Applies to pins with pull-up output and CMOS output specifications.)

5. We recommend the use of reflow soldering techniques to solder mount QFP packages.

Please consult with your Sanyo representative for details on process conditions if the package itself is to be directly immersed in a dip-soldering bath (dip-soldering techniques).



**Allowable Operating Ranges at Ta = -30 to +70°C, V<sub>SS</sub> = 0 V, V<sub>DD</sub> = 2.5 to 5.5 V, unless otherwise specified.**

Parameter	Symbol	Conditions	min	typ	max	Unit	Note
Operating supply voltage	V <sub>DD</sub>	V <sub>DD</sub> : 0.92 T <sub>cyc</sub> 10 μs	2.5		5.5	V	
Memory retention supply voltage	V <sub>DDH</sub>	V <sub>DD</sub> : During hold mode	1.8		5.5	V	
Input high-level voltage	V <sub>IH1</sub>	P2, P3 (except for the P33/HOLD pin), P4, P5, and P6: N-channel output transistor off	0.8 V <sub>DD</sub>		+13.5	V	1
	V <sub>IH2</sub>	P33/HOLD, RES, OSC1: N-channel output transistor off	0.8 V <sub>DD</sub>		V <sub>DD</sub>	V	2
	V <sub>IH3</sub>	P0, P1, PC, PD, PE: N-channel output transistor off	0.8 V <sub>DD</sub>		V <sub>DD</sub>	V	3
	V <sub>IH4</sub>	PE: With 3-value input used, V <sub>DD</sub> = 3.0 to 5.5 V	0.8 V <sub>DD</sub>		V <sub>DD</sub>	V	
Mid-level input voltage	V <sub>IM</sub>	PE: With 3-value input used, V <sub>DD</sub> = 3.0 to 5.5 V	0.4 V <sub>DD</sub>		0.6 V <sub>DD</sub>	V	
Common-mode input voltage range	V <sub>CMM1</sub>	PD0, PC2: When the comparator input is used, V <sub>DD</sub> = 3.0 to 5.5 V	1.5		V <sub>DD</sub>	V	
	V <sub>CMM2</sub>	PD1, PD2, PD3, PC3: When the comparator input is used, V <sub>DD</sub> = 3.0 to 5.5 V	V <sub>SS</sub>		V <sub>DD</sub> - 1.5	V	
Input low-level voltage	V <sub>IL1</sub>	P2, P3 (except for the P33/HOLD pin), P5, P6, RES, and OSC1: N-channel output transistor off			0.2 V <sub>DD</sub>	V	2
	V <sub>IL2</sub>	P33/HOLD: V <sub>DD</sub> = 1.8 to 5.5 V			0.2 V <sub>DD</sub>	V	
	V <sub>IL3</sub>	P0, P1, P4, PC, PD, PE, TEST: N-channel output transistor off	V <sub>SS</sub>		0.2 V <sub>DD</sub>	V	3
	V <sub>IL4</sub>	PE: With 3-value input used, V <sub>DD</sub> = 3.0 to 5.5 V	V <sub>SS</sub>		0.2 V <sub>DD</sub>	V	
Operating frequency (instruction cycle time)	f <sub>op</sub> (T <sub>cyc</sub> )		0.4 (10)		4.35 (0.92)	MHz (μs)	
[External clock input conditions]							
Frequency	f <sub>ext</sub>	OSC1: Defined by Figure 1. Input the clock signal to OSC1 and leave OSC2 open. (External clock input must be selected as the oscillator circuit option.)	0.4		4.35	MHz	
Pulse width	t <sub>extH</sub> , t <sub>extL</sub>	OSC1: Defined by Figure 1. Input the clock signal to OSC1 and leave OSC2 open. (External clock input must be selected as the oscillator circuit option.)	100			ns	
Rise and fall times	t <sub>extR</sub> , t <sub>extF</sub>	OSC1: Defined by Figure 1. Input the clock signal to OSC1 and leave OSC2 open. (External clock input must be selected as the oscillator circuit option.)			30	ns	

Note: 1. Applies to pins with open-drain specifications. However, V<sub>IH2</sub> applies to the P33/HOLD pin.

When ports P2, P3, and P6 have CMOS output specifications they cannot be used as input pins.

2. Applies to pins with open-drain specifications.

3. When RE is used as a three-value input, V<sub>IH4</sub>, V<sub>IM</sub>, and V<sub>IL4</sub> apply. When the ports PC pins have CMOS output specifications they cannot be used as input pins.

**Electrical Characteristics at Ta = -30 to +70°C, V<sub>SS</sub> = 0 V, V<sub>DD</sub> = 2.5 to 5.5 V unless otherwise specified.**

Parameter	Symbol	Conditions	min	typ	max	Unit	Note
Input high-level current	I <sub>IH1</sub>	P2, P3 (except for the P33/HOLD pin), P4, P5, and P6: V <sub>IN</sub> = 13.5 V, with the output Nch transistor off			5.0	μA	1
	I <sub>IH2</sub>	P0, P1, PC, OSC1, $\overline{\text{RES}}$ , P33/HOLD: V <sub>IN</sub> = V <sub>DD</sub> , with the output Nch transistor off			1.0	μA	1
	I <sub>IH3</sub>	PD, PE, PC2, PC3: V <sub>IN</sub> = V <sub>DD</sub> , with the output Nch transistor off			1.0	μA	1
Input low-level current	I <sub>IL1</sub>	Input ports other than PD, PE, PC2, and PC3: V <sub>IN</sub> = V <sub>SS</sub> , with the output Nch transistor off	-1.0			μA	2
	I <sub>IL2</sub>	PC2, PC3, PD, PE: V <sub>IN</sub> = V <sub>SS</sub> , with the output Nch transistor off	-1.0			μA	2
Output high-level voltage	V <sub>OH1</sub>	P2, P3 (except for the P33/HOLD pin), P6, and PC: I <sub>OH</sub> = -1 mA	V <sub>DD</sub> - 1.0			V	3
		P2, P3 (except for the P33/HOLD pin), P6, and PC: I <sub>OH</sub> = -0.1 mA	V <sub>DD</sub> - 0.5				
	V <sub>OH2</sub>	P0, P1, P4, P5: I <sub>OH</sub> = -50 μA	V <sub>DD</sub> - 1.0			V	4
		P0, P1, P4, P5: I <sub>OH</sub> = -30 μA	V <sub>DD</sub> - 0.5				
Output pull-up current	I <sub>PO</sub>	P0, P1, P4, P5: V <sub>IN</sub> = V <sub>SS</sub> , V <sub>DD</sub> = 5.5 V	-1.6			mA	4
Output low-level voltage	V <sub>OL1</sub>	P0, P1, P2, P3, P4, P5, P6, and PC (except for the P33/HOLD pin): I <sub>OL</sub> = 1.6 mA			0.4	V	5
	V <sub>OL2</sub>	P0, P1, P2, P3, P4, P5, P6, and PC (except for the P33/HOLD pin): I <sub>OL</sub> = 8 mA			1.5	V	
Output off leakage current	I <sub>OFF1</sub>	P2, P3, P4, P5, P6: V <sub>IN</sub> = 13.5 V			5.0	μA	5
	I <sub>OFF2</sub>	P0, P1, PC: V <sub>IN</sub> = V <sub>DD</sub>			1.0	μA	5
Comparator offset voltage	V <sub>OFF1</sub>	PD1 to PD3: V <sub>IN</sub> = V <sub>SS</sub> to V <sub>DD</sub> - 1.5 V, V <sub>DD</sub> = 3.0 to 5.5 V		±50	±300	mV	
	V <sub>OFF2</sub>	PD0: V <sub>IN</sub> = 1.5 to V <sub>DD</sub> , V <sub>DD</sub> = 3.0 to 5.5 V		±50	±300	mV	
[Schmitt characteristics]							
Hysteresis voltage	V <sub>HIS</sub>			0.1 V <sub>DD</sub>			
High-level threshold voltage	V <sub>tH</sub>	P2, P3, P5, P6, OSC1 (EXT), $\overline{\text{RES}}$	0.5 V <sub>DD</sub>		0.8 V <sub>DD</sub>	V	
Low-level threshold voltage	V <sub>tL</sub>		0.2 V <sub>DD</sub>		0.5 V <sub>DD</sub>	V	
[Ceramic oscillator]							
Oscillator frequency	f <sub>CF</sub>	OSC1, OSC2: Figure 2, 4 MHz		4.0		MHz	
Oscillator stabilization time	f <sub>CFS</sub>	Figure 3, 4 MHz			10	ms	
[Serial clock]							
Cycle time	Input	SCK0, SCK1: With the timing of Figure 4 and the test load of Figure 5.	t <sub>CKCY</sub>	0.9			μs
	Output			2.0			T <sub>cyc</sub>
Low-level and high-level pulse widths	Input		t <sub>CKL</sub>	0.4			μs
	Output		t <sub>CKH</sub>	1.0			T <sub>cyc</sub>
Rise and fall times	Output	t <sub>CKR</sub> , t <sub>CKF</sub>			0.1	μs	
[Serial input]							
Data setup time	t <sub>ICK</sub>	S10, S11: With the timing of Figure 4. Stipulated with respect to the rising edge (↑) of SCK0 or SCK1.	0.3			μs	
Data hold time	t <sub>ICKI</sub>		0.3			μs	
[Serial output]							
Output delay time	t <sub>CKO</sub>	SO0, SO1: With the timing of Figure 4 and the test load of Figure 5. Stipulated with respect to the falling edge (↓) of SCK0 or SCK1.			0.3		

Continued on next page.

Continued from preceding page.

Parameter	Symbol	Conditions	min	typ	max	Unit	Note
[Pulse conditions]							
INT0 high and low-level	$t_{IOH}, t_{IOL}$	INT0: Figure 6, conditions under which the INT0 interrupt can be accepted, conditions under which the timer 0 event counter or pulse width measurement input can be accepted	2			Tcyc	
High and low-level pulse widths for interrupt inputs other than INT0	$t_{IIH}, t_{IIL}$	INT1, INT2: Figure 6, conditions under which the corresponding interrupt can be accepted	2			Tcyc	
PIN1 high and low-level pulse widths	$t_{PINH}, t_{PINL}$	PIN1: Figure 6, conditions under which the timer 1 event counter input can be accepted	2			Tcyc	
RES high and low-level pulse widths	$t_{RSH}, t_{RSL}$	RES: Figure 6, conditions under which reset can be applied.	3			Tcyc	
Comparator response speed	$T_{RS}$	PD: Figure 7, $V_{DD} = 3.0$ to $5.5$ V			20	ms	
Operating current drain	$I_{DDOP}$	$V_{DD}$ : 4-MHz ceramic oscillator		3.0	5.0	mA	6
		$V_{DD}$ : 4-MHz external clock		3.0	5.0	mA	
Halt mode current drain	$I_{DDHALT}$	$V_{DD}$ : 4-MHz ceramic oscillator		1.0	2.0	mA	
		$V_{DD}$ : 4-MHz external clock		1.0	2.0	mA	
Hold mode current drain	$I_{DDHOLD}$	$V_{DD}$ : $V_{DD} = 1.8$ to $5.5$ V		0.01	10	$\mu$ A	

- Note: 1. With the output Nch transistor off in shared I/O ports with the open-drain output specifications. These pins cannot be used as input pins if the CMOS output specifications are selected.  
 2. With the output Nch transistor off in shared I/O ports with the open-drain output specifications. The rating for the pull-up output specification pins is stipulated in terms of the output pull-up current IPO. These pins cannot be used as input pins if the CMOS output specifications are selected.  
 3. With the output Nch transistor off for CMOS output specification pins.  
 4. With the output Nch transistor off for pull-up output specification pins.  
 5. With the output Nch transistor off for open-drain output specification pins.  
 6. Reset state

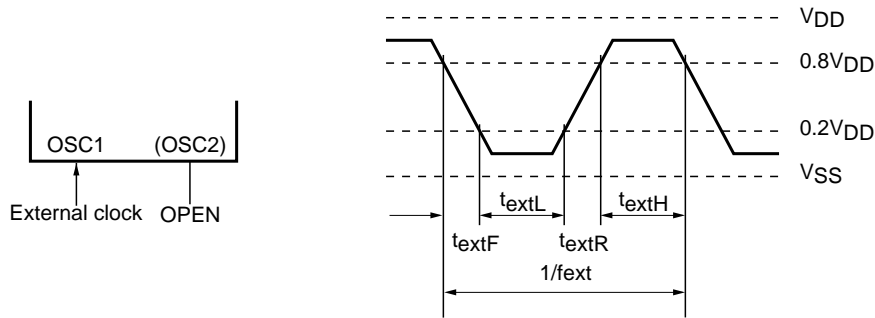


Figure 1 External Clock Input Waveform

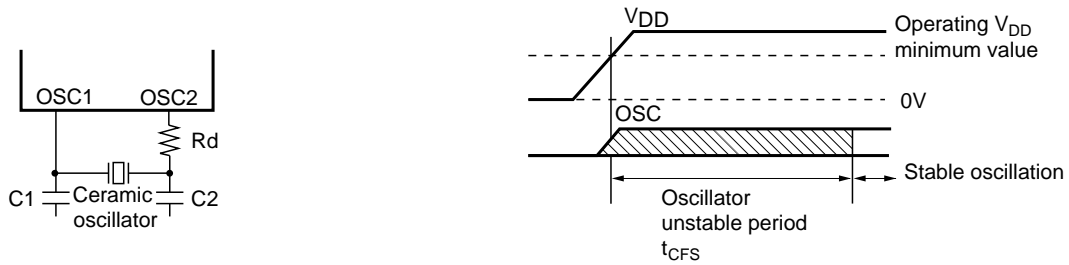


Figure 2 Ceramic Oscillator Circuit

Figure 3 Oscillator Stabilization Period

Table 1 Guaranteed Ceramic Oscillator Constants

4 MHz (Murata Mfg. Co., Ltd.) CSA4.00MG	C1 = 33 pF ± 10%	4 MHz (Kyocera Corporation) KBR4.0MS	C1 = 33 pF ± 10%
	C2 = 33 pF ± 10%		C2 = 33 pF ± 10%
	Rd = 0		Rd = 0

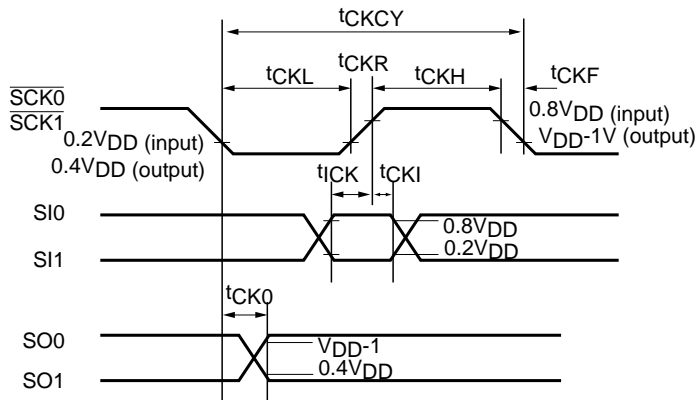


Figure 4 Serial I/O Timing

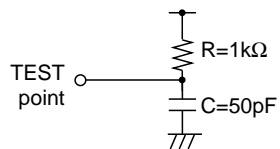


Figure 5 Timing Load

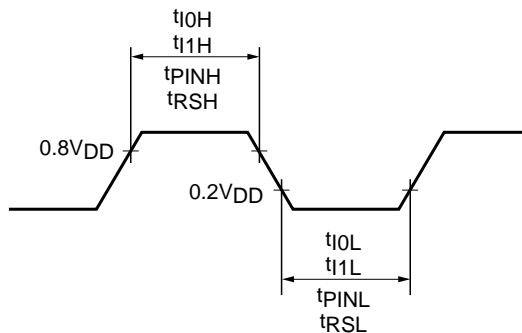


Figure 6 Input Timing for the INT0, INT1, INT2, PIN1, and RES pins

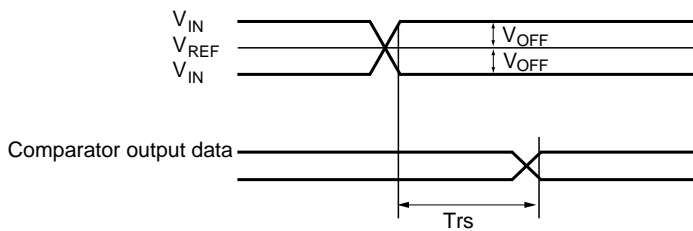


Figure 7 Comparator Response Speed Trs Timing

**LC66XXX Series Instruction Table (by function)**

## Abbreviations:

AC:	Accumulator
E:	E register
CF:	Carry flag
ZF:	Zero flag
HL:	Data pointer DPH, DPL
XY:	Data pointer DPX, DPY
M:	Data memory
M (HL):	Data memory pointed to by the DPH, DPL data pointer
M (XY):	Data memory pointed to by the DPX, DPY auxiliary data pointer
M2 (HL):	Two words of data memory (starting on an even address) pointed to by the DPH, DPL data pointer
SP:	Stack pointer
M2 (SP):	Two words of data memory pointed to by the stack pointer
M4 (SP):	Four words of data memory pointed to by the stack pointer
in:	n bits of immediate data
t2:	Bit specification

t2	11	10	01	00
Bit	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>

PCh:	Bits 8 to 11 in the PC
PCm:	Bits 4 to 7 in the PC
PCl:	Bits 0 to 3 in the PC
Fn:	User flag, n = 0 to 15
TIMER0:	Timer 0
TIMER1:	Timer 1
SIO:	Serial register
P:	Port
P (i4):	Port indicated by 4 bits of immediate data
INT:	Interrupt enable flag
( ), [ ]:	Indicates the contents of a location
←:	Transfer direction, result
∨:	Exclusive or
∧:	Logical and
∨:	Logical or
+:	Addition
-:	Subtraction
—:	Taking the one's complement

Mnemonic	Instruction code		Number of bytes	Number of cycles	Operation	Description	Affected status bits	Note	
	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub>	D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>							
[Accumulator manipulation instructions]									
CLA	Clear AC	1 0 0 0	0 0 0 0	1	1	AC ← 0 (Equivalent to LAI 0.)	Clear AC to 0.	ZF	Has a vertical skip function.
DAA	Decimal adjust AC in addition	1 1 0 0 0 0 1 0	1 1 1 1 0 1 1 0	2	2	AC ← (AC) + 6 (Equivalent to ADI 6.)	Add six to AC.	ZF	
DAS	Decimal adjust AC in subtraction	1 1 0 0 0 0 1 0	1 1 1 1 1 0 1 0	2	2	AC ← (AC) + 10 (Equivalent to ADI 0AH.)	Add 10 to AC.	ZF	
CLC	Clear CF	0 0 0 1	1 1 1 0	1	1	CF ← 0	Clear CF to 0.	CF	
STC	Set CF	0 0 0 1	1 1 1 1	1	1	CF ← 1	Set CF to 1.	CF	
CMA	Complement AC	0 0 0 1	1 0 0 0	1	1	AC ← $\overline{(AC)}$	Take the one's complement of AC.	ZF	
IA	Increment AC	0 0 0 1	0 1 0 0	1	1	AC ← (AC) + 1	Increment AC.	ZF, CF	
DA	Decrement AC	0 0 1 0	0 1 0 0	1	1	AC ← (AC) - 1	Decrement AC.	ZF, CF	
RAR	Rotate AC right through CF	0 0 0 1	0 0 0 0	1	1	AC <sub>3</sub> ← (CF), AC <sub>n</sub> ← (AC <sub>n</sub> + 1), CF ← (AC <sub>0</sub> )	Shift AC (including CF) right.	CF	
RAL	Rotate AC left through CF	0 0 0 0	0 0 0 1	1	1	AC <sub>0</sub> ← (CF), AC <sub>n</sub> + 1 ← (AC <sub>n</sub> ), CF ← (AC <sub>3</sub> )	Shift AC (including CF) left.	CF, ZF	
TAE	Transfer AC to E	0 1 0 0	0 1 0 1	1	1	E ← (AC)	Transfer the contents of AC to E.		
TEA	Transfer E to AC	0 1 0 0	0 1 1 0	1	1	AC ← (E)	Transfer the contents of E to AC.	ZF	
XAE	Exchange AC with E	0 1 0 0	0 1 0 0	1	1	(AC) ↔ (E)	Exchange the contents of AC and E.		
[Memory manipulation instructions]									
IM	Increment M	0 0 0 1	0 0 1 0	1	1	M (HL) ← [M (HL)] + 1	Increment M (HL).	ZF, CF	
DM	Decrement M	0 0 1 0	0 0 1 0	1	1	M (HL) ← [M (HL)] - 1	Decrement M (HL).	ZF, CF	
IMDR i8	Increment M direct	1 1 0 0 l <sub>7</sub> l <sub>6</sub> l <sub>5</sub> l <sub>4</sub>	0 1 1 1 l <sub>3</sub> l <sub>2</sub> l <sub>1</sub> l <sub>0</sub>	2	2	M (i8) ← [M (i8)] + 1	Increment M (i8).	ZF, CF	
DMDR i8	Decrement M direct	1 1 0 0 l <sub>7</sub> l <sub>6</sub> l <sub>5</sub> l <sub>4</sub>	0 0 1 1 l <sub>3</sub> l <sub>2</sub> l <sub>1</sub> l <sub>0</sub>	2	2	M (i8) ← [M (i8)] - 1	Decrement M (i8).	ZF, CF	
SMB t2	Set M data bit	0 0 0 0	1 1 t <sub>1</sub> t <sub>0</sub>	1	1	[M (HL), t2] ← 1	Set the bit in M (HL) specified by t0 and t1 to 1.		
RMB t2	Reset M data bit	0 0 1 0	1 1 t <sub>1</sub> t <sub>0</sub>	1	1	[M (HL), t2] ← 0	Clear the bit in M (HL) specified by t0 and t1 to 0.	ZF	
[Arithmetic, logic and comparison instructions]									
AD	Add M to AC	0 0 0 0	0 1 1 0	1	1	AC ← (AC) + [M (HL)]	Add the contents of AC and M (HL) as two's complement values and store the result in AC.	ZF, CF	
ADDR i8	Add M direct to AC	1 1 0 0 l <sub>7</sub> l <sub>6</sub> l <sub>5</sub> l <sub>4</sub>	1 0 0 1 l <sub>3</sub> l <sub>2</sub> l <sub>1</sub> l <sub>0</sub>	2	2	AC ← (AC) + [M (i8)]	Add the contents of AC and M (i8) as two's complement values and store the result in AC.	ZF, CF	
ADC	Add M to AC with CF	0 0 0 0	0 0 1 0	1	1	AC ← (AC) + [M (HL)] + (CF)	Add the contents of AC, M (HL) and C as two's complement values and store the result in AC.	ZF, CF	
ADI i4	Add immediate data to AC	1 1 0 0 0 0 1 0	1 1 1 1 l <sub>3</sub> l <sub>2</sub> l <sub>1</sub> l <sub>0</sub>	2	2	AC ← (AC) + l <sub>3</sub> , l <sub>2</sub> , l <sub>1</sub> , l <sub>0</sub>	Add the contents of AC and the immediate data as two's complement values and store the result in AC.	ZF	
SUBC	Subtract AC from M with CF	0 0 0 1	0 1 1 1	1	1	AC ← [M (HL)] - (AC) - (CF)	Subtract the contents of AC and CF from M (HL) as two's complement values and store the result in AC.	ZF, CF	CF will be zero if there was a borrow and one otherwise.
ANDA	And M with AC then store AC	0 0 0 0	0 1 1 1	1	1	AC ← (AC) ∧ [M (HL)]	Take the logical and of AC and M (HL) and store the result in AC.	ZF	
ORA	Or M with AC then store AC	0 0 0 0	0 1 0 1	1	1	AC ← (AC) ∨ [M (HL)]	Take the logical or of AC and M (HL) and store the result in AC.	ZF	

Continued from preceding page.

Mnemonic	Instruction code		Number of bytes	Number of cycles	Operation	Description	Affected status bits	Note													
	D7 D6 D5 D4	D3 D2 D1 D0																			
[Arithmetic, logic and comparison instructions]																					
EXL	Exclusive or M with AC then store AC	0 0 0 1	0 1 0 1	1	1	$AC \leftarrow (AC) \oplus [M (HL)]$	Take the logical exclusive or of AC and M (HL) and store the result in AC.	ZF													
ANDM	And M with AC then store M	0 0 0 0	0 0 1 1	1	1	$M (HL) \leftarrow (AC) \wedge [M (HL)]$	Take the logical and of AC and M (HL) and store the result in M (HL).	ZF													
ORM	Or M with AC then store M	0 0 0 0	0 1 0 0	1	1	$M (HL) \leftarrow (AC) \vee [M (HL)]$	Take the logical or of AC and M (HL) and store the result in M (HL).	ZF													
CM	Compare AC with M	0 0 0 1	0 1 1 0	1	1	$\overline{[M (HL)]} + (AC) + 1$	Compare the contents of AC and M (HL) and set or clear CF and ZF according to the result. <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Magnitude comparison</td> <td>CF</td> <td>ZF</td> </tr> <tr> <td><math>[M (HL)] &gt; (AC)</math></td> <td>0</td> <td>0</td> </tr> <tr> <td><math>[M (HL)] = (AC)</math></td> <td>1</td> <td>1</td> </tr> <tr> <td><math>[M (HL)] &lt; (AC)</math></td> <td>1</td> <td>0</td> </tr> </table>	Magnitude comparison	CF	ZF	$[M (HL)] > (AC)$	0	0	$[M (HL)] = (AC)$	1	1	$[M (HL)] < (AC)$	1	0	ZF, CF	
Magnitude 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 0 1 0	1 1 1 1 $i_3 i_2 i_1 i_0$	2	2	$\overline{i_3 i_2 i_1 i_0} + (AC) + 1$	Compare the contents of AC and the immediate data $i_3 i_2 i_1 i_0$ and set or clear CF and ZF according to the result. <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Magnitude comparison</td> <td>CF</td> <td>ZF</td> </tr> <tr> <td><math>i_3 i_2 i_1 i_0 &gt; AC</math></td> <td>0</td> <td>0</td> </tr> <tr> <td><math>i_3 i_2 i_1 i_0 = AC</math></td> <td>1</td> <td>1</td> </tr> <tr> <td><math>i_3 i_2 i_1 i_0 &lt; AC</math></td> <td>1</td> <td>0</td> </tr> </table>	Magnitude comparison	CF	ZF	$i_3 i_2 i_1 i_0 > AC$	0	0	$i_3 i_2 i_1 i_0 = AC$	1	1	$i_3 i_2 i_1 i_0 < AC$	1	0	ZF, CF	
Magnitude comparison	CF	ZF																			
$i_3 i_2 i_1 i_0 > AC$	0	0																			
$i_3 i_2 i_1 i_0 = AC$	1	1																			
$i_3 i_2 i_1 i_0 < AC$	1	0																			
CLI i4	Compare $DP_L$ with immediate data	1 1 0 0 1 0 1 1	1 1 1 1 $i_3 i_2 i_1 i_0$	2	2	$ZF \leftarrow 1$ if $(DP_L) = i_3 i_2 i_1 i_0$ $ZF \leftarrow 0$ if $(DP_L) \neq i_3 i_2 i_1 i_0$	Compare the contents of $DP_L$ with the immediate data. Set ZF if identical and clear ZF if not.	ZF													
CMB t2	Compare AC bit with M data bit	1 1 0 0 1 1 0 1	1 1 1 1 $0 0 t_1 t_0$	2	2	$ZF \leftarrow 1$ if $(AC, t_2) = [M (HL), t_2]$ $ZF \leftarrow 0$ if $(AC, t_2) \neq [M (HL), t_2]$	Compare the corresponding bits specified by $t_0$ and $t_1$ in AC and M (HL). Set ZF if identical and clear ZF if not.	ZF													
[Load and store instructions]																					
LAE	Load AC and E from M2 (HL)	0 1 0 1	1 1 0 0	1	1	$AC \leftarrow M (HL), E \leftarrow M (HL + 1)$	Load the contents of M2 (HL) into AC, E.														
LAI i4	Load AC with immediate data	1 0 0 0	$i_3 i_2 i_1 i_0$	1	1	$AC \leftarrow i_3 i_2 i_1 i_0$	Load the immediate data into AC.	ZF	Has a vertical skip function												
LADR i8	Load AC from M direct	1 1 0 0 $i_7 i_6 i_5 i_4$	0 0 0 1 $i_3 i_2 i_1 i_0$	2	2	$AC \leftarrow [M (i8)]$	Load the contents of M (i8) into AC.	ZF													
S	Store AC to M	0 1 0 0	0 1 1 1	1	1	$M (HL) \leftarrow (AC)$	Store the contents of AC into M (HL).														
SAE	Store AC and E to M2 (HL)	0 1 0 1	1 1 1 0	1	1	$M (HL) \leftarrow (AC)$ $M (HL + 1) \leftarrow (E)$	Store the contents of AC, E into M2 (HL).														
LA reg	Load AC from M (reg)	0 1 0 0	1 0 $t_0$ 0	1	1	$AC \leftarrow [M (reg)]$	Load the contents of M (reg) into AC. The reg is either HL or XY depending on $t_0$ . <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>reg</td> <td><math>T_0</math></td> </tr> <tr> <td>HL</td> <td>0</td> </tr> <tr> <td>XY</td> <td>1</td> </tr> </table>	reg	$T_0$	HL	0	XY	1	ZF							
reg	$T_0$																				
HL	0																				
XY	1																				

Continued on next page.

Continued from preceding page.

Mnemonic	Instruction code		Number of bytes	Number of cycles	Operation	Description	Affected status bits	Note						
	D7 D6 D5 D4	D3 D2 D1 D0												
[Load and store instructions]														
LA reg, I	Load AC from M (reg) then increment reg	0 1 0 0	1 0 t <sub>0</sub> 1	1	2	$AC \leftarrow [M(\text{reg})]$ $DP_L \leftarrow (DP_L) + 1$ or $DP_Y \leftarrow (DP_Y) + 1$	Load the contents of M (reg) into AC. (The reg is either HL or XY.) Then increment the contents of either DP <sub>L</sub> or DP <sub>Y</sub> . The relationship between t <sub>0</sub> and reg is the same as that for the LA reg instruction.	ZF	ZF is set according to the result of incrementing DP <sub>L</sub> or DP <sub>Y</sub> .					
LA reg, D	Load AC from M (reg) then decrement reg	0 1 0 1	1 0 t <sub>0</sub> 1	1	2	$AC \leftarrow [M(\text{reg})]$ $DP_L \leftarrow (DP_L) - 1$ or $DP_Y \leftarrow (DP_Y) - 1$	Load the contents of M (reg) into AC. (The reg is either HL or XY.) Then decrement the contents of either DP <sub>L</sub> or DP <sub>Y</sub> . The relationship between t <sub>0</sub> and reg is the same as that for the LA reg instruction.	ZF	ZF is set according to the result of decrementing DP <sub>L</sub> or DP <sub>Y</sub> .					
XA reg	Exchange AC with M (reg)	0 1 0 0	1 1 t <sub>0</sub> 0	1	1	(AC) ↔ [M (reg)]	Exchange the contents of M (reg) and AC. The reg is either HL or XY depending on t <sub>0</sub> .							
						<table border="1"> <tr> <td>reg</td> <td>T<sub>0</sub></td> </tr> <tr> <td>HL</td> <td>0</td> </tr> <tr> <td>XY</td> <td>1</td> </tr> </table>	reg	T <sub>0</sub>	HL	0	XY	1		
reg	T <sub>0</sub>													
HL	0													
XY	1													
XA reg, I	Exchange AC with M (reg) then increment reg	0 1 0 0	1 1 t <sub>0</sub> 1	1	2	$(AC) \leftrightarrow [M(\text{reg})]$ $DP_L \leftarrow (DP_L) + 1$ or $DP_Y \leftarrow (DP_Y) + 1$	Exchange the contents of M (reg) and AC. (The reg is either HL or XY.) Then increment the contents of either DP <sub>L</sub> or DP <sub>Y</sub> . The relationship between t <sub>0</sub> and reg is the same as that for the XA reg instruction.	ZF	ZF is set according to the result of incrementing DP <sub>L</sub> or DP <sub>Y</sub> .					
XA reg, D	Exchange AC with M (reg) then decrement reg	0 1 0 1	1 1 t <sub>0</sub> 1	1	2	$(AC) \leftrightarrow [M(\text{reg})]$ $DP_L \leftarrow (DP_L) - 1$ or $DP_Y \leftarrow (DP_Y) - 1$	Exchange the contents of M (reg) and AC. (The reg is either HL or XY.) Then decrement the contents of either DP <sub>L</sub> or DP <sub>Y</sub> . The relationship between t <sub>0</sub> and reg is the same as that for the XA reg instruction.	ZF	ZF is set according to the result of decrementing DP <sub>L</sub> or DP <sub>Y</sub> .					
XADR i8	Exchange AC with M direct	1 1 0 0 i <sub>7</sub> i <sub>6</sub> i <sub>5</sub> i <sub>4</sub>	1 0 0 0 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	(AC) ↔ [M (i8)]	Exchange the contents of AC and M (i8).							
LEAI i8	Load E & AC with immediate data	1 1 0 0 i <sub>7</sub> i <sub>6</sub> i <sub>5</sub> i <sub>4</sub>	0 1 1 0 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	$E \leftarrow i_7 i_6 i_5 i_4$ $AC \leftarrow i_3 i_2 i_1 i_0$	Load the immediate data i8 into E, AC.							
RTBL	Read table data from program ROM	0 1 0 1	1 0 1 0	1	2	E, AC ← [ROM (PCh, E, AC)]	Load into E, AC the ROM data at the location determined by replacing the lower 8 bits of the PC with E, AC.							
RTBLP	Read table data from program ROM then output to P4, 5	0 1 0 1	1 0 0 0	1	2	Port 4, 5 ← [ROM (PCh, E, AC)]	Output from ports 4 and 5 the ROM data at the location determined by replacing the lower 8 bits of the PC with E, AC.							
[Data pointer manipulation instructions]														
LDZ i4	Load DP <sub>H</sub> with zero and DP <sub>L</sub> with immediate data respectively	0 1 1 0	i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	1	1	$DP_H \leftarrow 0$ $DP_L \leftarrow i_3 i_2 i_1 i_0$	Load zero into DP <sub>H</sub> and the immediate data i4 into DP <sub>L</sub> .							
LHI i4	Load DP <sub>H</sub> with immediate data	1 1 0 0 0 0 0 0	1 1 1 1 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	$DP_H \leftarrow i_3 i_2 i_1 i_0$	Load the immediate data i4 into DP <sub>H</sub> .							
LLI i4	Load DP <sub>L</sub> with immediate data	1 1 0 0 0 0 0 1	1 1 1 1 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	$DP_L \leftarrow i_3 i_2 i_1 i_0$	Load the immediate data i4 into DP <sub>L</sub> .							
LHLI i8	Load DP <sub>H</sub> , DP <sub>L</sub> with immediate data	1 1 0 0 i <sub>7</sub> i <sub>6</sub> i <sub>5</sub> i <sub>4</sub>	0 0 0 0 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	$DP_H \leftarrow i_7 i_6 i_5 i_4$ $DP_L \leftarrow i_3 i_2 i_1 i_0$	Load the immediate data into DP <sub>H</sub> , DP <sub>L</sub> .							
LXYI i8	Load DP <sub>X</sub> , DP <sub>Y</sub> with immediate data	1 1 0 0 i <sub>7</sub> i <sub>6</sub> i <sub>5</sub> i <sub>4</sub>	0 0 0 0 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	$DP_X \leftarrow i_7 i_6 i_5 i_4$ $DP_Y \leftarrow i_3 i_2 i_1 i_0$	Load the immediate data into DP <sub>X</sub> , DP <sub>Y</sub> .							

Continued on next page.



Continued from preceding page.

Mnemonic	Instruction code	D7 D6 D5 D4	D3 D2 D1 D0	Number of bytes	Number of cycles	Operation	Description	Affected status bits	Note
[Data pointer manipulation instructions]									
IL	Increment DP <sub>L</sub>	0 0 0 1	0 0 0 1	1	1	DP <sub>L</sub> ← (DP <sub>L</sub> ) + 1	Increment the contents of DP <sub>L</sub> .	ZF	
DL	Decrement DP <sub>L</sub>	0 0 1 0	0 0 0 1	1	1	DP <sub>L</sub> ← (DP <sub>L</sub> ) - 1	Decrement the contents of DP <sub>L</sub> .	ZF	
IY	Increment DP <sub>Y</sub>	0 0 0 1	0 0 1 1	1	1	DP <sub>Y</sub> ← (DP <sub>Y</sub> ) + 1	Increment the contents of DP <sub>Y</sub> .	ZF	
DY	Decrement DP <sub>Y</sub>	0 0 1 0	0 0 1 1	1	1	DP <sub>Y</sub> ← (DP <sub>Y</sub> ) - 1	Decrement the contents of DP <sub>Y</sub> .	ZF	
TAH	Transfer AC to DP <sub>H</sub>	1 1 0 0 1 1 1 1	1 1 1 1 0 0 0 0	2	2	DP <sub>H</sub> ← (AC)	Transfer the contents of AC to DP <sub>H</sub> .		
THA	Transfer DP <sub>H</sub> to AC	1 1 0 0 1 1 1 0	1 1 1 1 0 0 0 0	2	2	AC ← (DP <sub>H</sub> )	Transfer the contents of DP <sub>H</sub> to AC.	ZF	
XAH	Exchange AC with DP <sub>H</sub>	0 1 0 0	0 0 0 0	1	1	(AC) ↔ (DP <sub>H</sub> )	Exchange the contents of AC and DP <sub>H</sub> .		
TAL	Transfer AC to DP <sub>L</sub>	1 1 0 0 1 1 1 1	1 1 1 1 0 0 0 1	2	2	DP <sub>L</sub> ← (AC)	Transfer the contents of AC to DP <sub>L</sub> .		
TLA	Transfer DP <sub>L</sub> to AC	1 1 0 0 1 1 1 0	1 1 1 1 0 0 0 1	2	2	AC ← (DP <sub>L</sub> )	Transfer the contents of DP <sub>L</sub> to AC.	ZF	
XAL	Exchange AC with DP <sub>L</sub>	0 1 0 0	0 0 0 1	1	1	(AC) ↔ (DP <sub>L</sub> )	Exchange the contents of AC and DP <sub>L</sub> .		
TAX	Transfer AC to DP <sub>X</sub>	1 1 0 0 1 1 1 1	1 1 1 1 0 0 1 0	2	2	DP <sub>X</sub> ← (AC)	Transfer the contents of AC to DP <sub>X</sub> .		
TXA	Transfer DP <sub>X</sub> to AC	1 1 0 0 1 1 1 0	1 1 1 1 0 0 1 0	2	2	AC ← (DP <sub>X</sub> )	Transfer the contents of DP <sub>X</sub> to AC.	ZF	
XAX	Exchange AC with DP <sub>X</sub>	0 1 0 0	0 0 1 0	1	1	(AC) ↔ (DP <sub>X</sub> )	Exchange the contents of AC and DP <sub>X</sub> .		
TAY	Transfer AC to DP <sub>Y</sub>	1 1 0 0 1 1 1 1	1 1 1 1 0 0 1 1	2	2	DP <sub>Y</sub> ← (AC)	Transfer the contents of AC to DP <sub>Y</sub> .		
TYA	Transfer DP <sub>Y</sub> to AC	1 1 0 0 1 1 1 0	1 1 1 1 0 0 1 1	2	2	AC ← (DP <sub>Y</sub> )	Transfer the contents of DP <sub>Y</sub> to AC.	ZF	
XAY	Exchange AC with DP <sub>Y</sub>	0 1 0 0	0 0 1 1	1	1	(AC) ↔ (DP <sub>Y</sub> )	Exchange the contents of AC and DP <sub>Y</sub> .		
[Flag manipulation instructions]									
SFB n4	Set flag bit	0 1 1 1	n <sub>3</sub> n <sub>2</sub> n <sub>1</sub> n <sub>0</sub>	1	1	Fn ← 1	Set the flag specified by n4 to 1.		
RFB n4	Reset flag bit	0 0 1 1	n <sub>3</sub> n <sub>2</sub> n <sub>1</sub> n <sub>0</sub>	1	1	Fn ← 0	Reset the flag specified by n4 to 0.	ZF	
[Jump and subroutine instructions]									
JMP addr	Jump in the current bank	1 1 1 0 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	P <sub>11</sub> P <sub>10</sub> P <sub>9</sub> P <sub>8</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC13, 12 ← PC13, 12 PC11 to 0 ← P <sub>11</sub> to P <sub>8</sub>	Jump to the location in the same bank specified by the immediate data P12.		This becomes PC12 + (PC12) immediately following a BANK instruction.
JPEA	Jump to the address stored at E and AC in the current page	0 0 1 0	0 1 1 1	1	1	PC13 to 8 ← PC13 to 8, PC7 to 4 ← (E), PC3 to 0 ← (AC)	Jump to the location determined by replacing the lower 8 bits of the PC by E, AC.		
CAL addr	Call subroutine	0 1 0 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	0 P <sub>10</sub> P <sub>9</sub> P <sub>8</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC13 to 11 ← 0, PC10 to 0 ← P <sub>10</sub> to P <sub>0</sub> , M4 (SP) ← (CF, ZF, PC13 to 0), SP ← (SP)-4	Call a subroutine.		
CZP addr	Call subroutine in the zero page	1 0 1 0	P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	1	2	PC13 to 6, PC10 ← 0, PC5 to 2 ← P <sub>3</sub> to P <sub>0</sub> , M4 (SP) ← (CF, ZF, PC12 to 0), SP ← SP-4	Call a subroutine on page 0 in bank 0.		
BANK	Change bank	0 0 0 1	1 0 1 1	1	1		Change the memory bank and register bank.		

Continued on next page.

Continued from preceding page.

Mnemonic	Instruction code		Number of bytes	Number of cycles	Operation	Description	Affected status bits	Note																
	D7 D6 D5 D4	D3 D2 D1 D0																						
[Jump and subroutine instructions]																								
PUSH reg	Push reg on M2 (SP)	1 1 0 0 1 1 1 1	1 1 1 1 1 i <sub>1</sub> i <sub>0</sub> 0	2	2	M2 (SP) ← (reg) SP ← (SP) - 2	Store the contents of reg in M2 (SP). Subtract 2 from SP after the store. <table border="1" style="margin-left: 20px;"> <tr> <td>reg</td> <td>i<sub>1</sub></td> <td>i<sub>0</sub></td> </tr> <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 value</td> <td>1</td> <td>1</td> </tr> </table>	reg	i <sub>1</sub>	i <sub>0</sub>	HL	0	0	XY	0	1	AE	1	0	Illegal value	1	1		
reg	i <sub>1</sub>	i <sub>0</sub>																						
HL	0	0																						
XY	0	1																						
AE	1	0																						
Illegal value	1	1																						
POP reg	Pop reg off M2 (SP)	1 1 0 0 1 1 1 0	1 1 1 1 1 i <sub>1</sub> i <sub>0</sub> 0	2	2	SP ← (SP) + 2 reg ← [M2 (SP)]	Add 2 to SP and then load the contents of M2(SP) into reg. The relation between i1i0 and reg is the same as that for the PUSH reg instruction.																	
RT	Return from subroutine	0 0 0 1	1 1 0 0	1	2	SP ← (SP) + 4 PC ← [M4 (SP)]	Return from a subroutine or interrupt handling routine. ZF and CF are not restored.																	
RTI	Return from interrupt routine	0 0 0 1	1 1 0 1	1	2	SP ← (SP) + 4 PC ← [M4 (SP)] CF, ZF ← [M4 (SP)]	Return from a subroutine or interrupt handling routine. ZF and CF are restored.	ZF, CF																
[Branch instructions]																								
BAt2 addr	Branch on AC bit	1 1 0 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	0 0 t <sub>1</sub> t <sub>0</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC7 to 0 ← P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> if (AC, t2) = 1	Branch to the location in the same page specified by P <sub>7</sub> to P <sub>0</sub> if the bit in AC specified by the immediate data t <sub>1</sub> t <sub>0</sub> is one.																	
BNAt2 addr	Branch on no AC bit	1 0 0 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	0 0 t <sub>1</sub> t <sub>0</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC7 to 0 ← P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> if (AC, t2) = 0	Branch to the location in the same page specified by P <sub>7</sub> to P <sub>0</sub> if the bit in AC specified by the immediate data t <sub>1</sub> t <sub>0</sub> is zero.																	
BMt2 addr	Branch on M bit	1 1 0 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	0 1 t <sub>1</sub> t <sub>0</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC7 to 0 ← P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> if [M (HL), t2] = 1	Branch to the location in the same page specified by P <sub>7</sub> to P <sub>0</sub> if the bit in M (HL) specified by the immediate data t <sub>1</sub> t <sub>0</sub> is one.																	
BNMt2 addr	Branch on no M bit	1 0 0 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	0 1 t <sub>1</sub> t <sub>0</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC7 to 0 ← P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> if [M (HL), t2] = 0	Branch to the location in the same page specified by P <sub>7</sub> to P <sub>0</sub> if the bit in M (HL) specified by the immediate data t <sub>1</sub> t <sub>0</sub> is zero.																	
BPt2 addr	Branch on Port bit	1 1 0 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	1 0 t <sub>1</sub> t <sub>0</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC7 to 0 ← P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> if [P (DP <sub>L</sub> ), t2] = 1	Branch to the location in the same page specified by P <sub>7</sub> to P <sub>0</sub> if the bit in port (DP <sub>L</sub> ) specified by the immediate data t <sub>1</sub> t <sub>0</sub> is one.	Internal control registers can also be tested by executing this instruction immediately after a BANK instruction. However, this is limited to registers that can be read out.																
BNPt2 addr	Branch on no Port bit	1 0 0 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	1 0 t <sub>1</sub> t <sub>0</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC7 to 0 ← P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> if [P (DP <sub>L</sub> ), t2] = 0	Branch to the location in the same page specified by P <sub>7</sub> to P <sub>0</sub> if the bit in port (DP <sub>L</sub> ) specified by the immediate data t <sub>1</sub> t <sub>0</sub> is zero.	Internal control registers can also be tested by executing this instruction immediately after a BANK instruction. However, this is limited to registers that can be read out.																

Continued on next page.

Continued from preceding page.

Mnemonic		Instruction code		Number of bytes	Number of cycles	Operation	Description	Affected status bits	Note
		D7 D6 D5 D4	D3 D2 D1 D0						
[Branch instructions]									
BC addr	Branch on CF	1 1 0 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	1 1 0 0 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC7 to 0 ← P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> if (CF) = 1	Branch to the location in the same page specified by P <sub>7</sub> to P <sub>0</sub> if CF is one.		
BNC addr	Branch on no CF	1 0 0 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	1 1 0 0 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC7 to 0 ← P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> if (CF) = 0	Branch to the location in the same page specified by P <sub>7</sub> to P <sub>0</sub> if CF is zero.		
BZ addr	Branch on ZF	1 1 0 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	1 1 0 1 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC7 to 0 ← P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> if (ZF) = 1	Branch to the location in the same page specified by P <sub>7</sub> to P <sub>0</sub> if ZF is one.		
BNZ addr	Branch on no ZF	1 0 0 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	1 1 0 1 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC7 to 0 ← P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> if (ZF) = 0	Branch to the location in the same page specified by P <sub>7</sub> to P <sub>0</sub> if ZF is zero.		
BFn4 addr	Branch on flag bit	1 1 1 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	n <sub>3</sub> n <sub>2</sub> n <sub>1</sub> n <sub>0</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC7 to 0 ← P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> if (Fn) = 1	Branch to the location in the same page specified by P <sub>0</sub> to P <sub>7</sub> if the flag (of the 16 user flags) specified by n <sub>3</sub> n <sub>2</sub> n <sub>1</sub> n <sub>0</sub> is one.		
BNFn4 addr	Branch on no flag bit	1 0 1 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub>	n <sub>3</sub> n <sub>2</sub> n <sub>1</sub> n <sub>0</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	PC7 to 0 ← P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> if (Fn) = 0	Branch to the location in the same page specified by P <sub>0</sub> to P <sub>7</sub> if the flag (of the 16 user flags) specified by n <sub>3</sub> n <sub>2</sub> n <sub>1</sub> n <sub>0</sub> is zero.		
[I/O instructions]									
IP0	Input port 0 to AC	0 0 1 0	0 0 0 0	1	1	AC ← (P0)	Input the contents of port 0 to AC.	ZF	
IP	Input port to AC	0 0 1 0	0 1 1 0	1	1	AC ← [P (DP <sub>L</sub> )]	Input the contents of port P (DP <sub>L</sub> ) to AC.	ZF	
IPM	Input port to M	0 0 0 1	1 0 0 1	1	1	M (HL) ← [P (DP <sub>L</sub> )]	Input the contents of port P (DP <sub>L</sub> ) to M (HL).		
IPDR i4	Input port to AC direct	1 1 0 0 0 1 1 0	1 1 1 1 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	AC ← [P (i4)]	Input the contents of P (i4) to AC.	ZF	
IP45	Input port 4, 5 to E, AC respectively	1 1 0 0 1 1 0 1	1 1 1 1 0 1 0 0	2	2	E ← [P (4)] AC ← [P (5)]	Input the contents of ports P (4) and P (5) to E and AC respectively.		
OP	Output AC to port	0 0 1 0	0 1 0 1	1	1	P (DP <sub>L</sub> ) ← (AC)	Output the contents of AC to port P (DP <sub>L</sub> ).		
OPM	Output M to port	0 0 0 1	1 0 1 0	1	1	P (DP <sub>L</sub> ) ← [M (HL)]	Output the contents of M (HL) to port P (DP <sub>L</sub> ).		
OPDR i4	Output AC to port direct	1 1 0 0 0 1 1 1	1 1 1 1 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	P (i4) ← (AC)	Output the contents of AC to P (i4).		
OP45	Output E, AC to port 4, 5 respectively	1 1 0 0 1 1 0 1	1 1 1 1 0 1 0 1	2	2	P (4) ← (E) P (5) ← (AC)	Output the contents of E and AC to ports P (4) and P (5) respectively.		
SPB t2	Set port bit	0 0 0 0	1 0 t <sub>1</sub> t <sub>0</sub>	1	1	[P (DP <sub>L</sub> ), t2] ← 1	Set to one the bit in port P (DP <sub>L</sub> ) specified by the immediate data t <sub>1</sub> t <sub>0</sub> .		
RPB t2	Reset port bit	0 0 1 0	1 0 t <sub>1</sub> t <sub>0</sub>	1	1	[P (DP <sub>L</sub> ), t2] ← 0	Clear to zero the bit in port P (DP <sub>L</sub> ) specified by the immediate data t <sub>1</sub> t <sub>0</sub> .	ZF	
ANDPDR i4, p4	And port with immediate data then output	1 1 0 0 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	0 1 0 1 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	P (P <sub>3</sub> to P <sub>0</sub> ) ← [P (P <sub>3</sub> to P <sub>0</sub> )] ∨ i <sub>3</sub> to i <sub>0</sub>	Take the logical AND of P (P <sub>3</sub> to P <sub>0</sub> ) and the immediate data i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub> and output the result to P (P <sub>3</sub> to P <sub>0</sub> ).	ZF	
ORPDR i4, p4	Or port with immediate data then output	1 1 0 0 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	0 1 0 0 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2	2	P (P <sub>3</sub> to P <sub>0</sub> ) ← [P (P <sub>3</sub> to P <sub>0</sub> )] ∨ i <sub>3</sub> to i <sub>0</sub>	Take the logical OR of P (P <sub>3</sub> to P <sub>0</sub> ) and the immediate data i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub> and output the result to P (P <sub>3</sub> to P <sub>0</sub> ).	ZF	

Continued on next page.

Continued from preceding page.

Mnemonic		Instruction code		Number of bytes	Number of cycles	Operation	Description	Affected status bits	Note
		D7 D6 D5 D4	D3 D2 D1 D0						
[Timer control instructions]									
WTTM0	Write timer 0	1 1 0 0	1 0 1 0	1	2	TIMER0 ← [M2 (HL)], (AC)	Write the contents of M2 (HL), AC into the timer 0 reload register.		
WTTM1	Write timer 1	1 1 0 0 1 1 1 1	1 1 1 1 0 1 0 0	2	2	TIMER1 ← (E), (AC)	Write the contents of E, AC into the timer 1 reload register A.		
RTIM0	Read timer 0	1 1 0 0	1 0 1 1	1	2	M2 (HL), AC ← (TIMER0)	Read out the contents of the timer 0 counter into M2 (HL), AC.		
RTIM1	Read timer 1	1 1 0 0 1 1 1 1	1 1 1 1 0 1 0 1	2	2	E, AC ← (TIMER1)	Read out the contents of the timer 1 counter into E, AC.		
START0	Start timer 0	1 1 0 0 1 1 1 0	1 1 1 1 0 1 1 0	2	2	Start timer 0 counter	Start the timer 0 counter.		
START1	Start timer 1	1 1 0 0 1 1 1 0	1 1 1 1 0 1 1 1	2	2	Start timer 1 counter	Start the timer 1 counter.		
STOP0	Stop timer 0	1 1 0 0 1 1 1 1	1 1 1 1 0 1 1 0	2	2	Stop timer 0 counter	Stop the timer 0 counter.		
STOP1	Stop timer 1	1 1 0 0 1 1 1 1	1 1 1 1 0 1 1 1	2	2	Stop timer 1 counter	Stop the timer 1 counter.		
[Interrupt control instructions]									
MSET	Set interrupt master enable flag	1 1 0 0 0 1 0 1	1 1 0 1 0 0 0 0	2	2	MSE ← 1	Set the interrupt master enable flag to one.		
MRESET	Reset interrupt master enable flag	1 1 0 0 1 0 0 1	1 1 0 1 0 0 0 0	2	2	MSE ← 0	Clear the interrupt master enable flag to zero.		
EIH i4	Enable interrupt high	1 1 0 0 0 1 0 1	1 1 0 1 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	EDIH ← (EDIH) ∨ i4	Set the interrupt enable flag to one.		
EIL i4	Enable interrupt low	1 1 0 0 0 1 0 0	1 1 0 1 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	EDIL ← (EDIL) ∨ i4	Set the interrupt enable flag to one.		
DIH i4	Disable interrupt high	1 1 0 0 1 0 0 1	1 1 0 1 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	EDIH ← (EDIH) ∧ i <sub>4</sub>	Clear the interrupt enable flag to zero.	ZF	
DIL i4	Disable interrupt low	1 1 0 0 1 0 0 0	1 1 0 1 i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>	2	2	EDIL ← (EDIL) ∧ i <sub>4</sub>	Clear the interrupt enable flag to zero.	ZF	
WTSP	Write SP	1 1 0 0 1 1 0 1	1 1 1 1 1 0 1 0	2	2	SP ← (E), (AC)	Transfer the contents of E, AC to SP.		
RSP	Read SP	1 1 0 0 1 1 0 1	1 1 1 1 1 0 1 1	2	2	E, AC ← (SP)	Transfer the contents of SP to E, AC.		
[Standby control instructions]									
HALT	HALT	1 1 0 0 1 1 0 1	1 1 1 1 1 1 1 0	2	2	HALT	Enter halt mode.		
HOLD	HOLD	1 1 0 0 1 1 0 1	1 1 1 1 1 1 1 1	2	2	HOLD	Enter hold mode.		
[Serial I/O control instructions]									
STARTS	Start serial I O	1 1 0 0 1 1 1 0	1 1 1 1 1 1 1 0	2	2	START SIO	Start SIO operation.		
WTSIO	Write serial I O	1 1 0 0 1 1 1 0	1 1 1 1 1 1 1 1	2	2	SIO ← (E), (AC)	Write the contents of E, AC to SIO.		
RSIO	Read serial I O	1 1 0 0 1 1 1 1	1 1 1 1 1 1 1 1	2	2	E, AC ← (SIO)	Read out the contents of SIO into E, AC.		
[Other instructions]									
NOP	No operation	0 0 0 0	0 0 0 0	1	1	No operation	Consume one machine cycle without performing any operation.		
SB i2	Select bank	1 1 0 0 1 1 0 0	1 1 1 1 0 0 i <sub>1</sub> i <sub>0</sub>	2	2	PC12 ← i <sub>1</sub> i <sub>0</sub>	Specify the memory bank.		

Note: The range of for i2 in SB instruction varies according to device. Refer to User's Manual for that.

- 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.

This catalog provides information as of February, 1997. Specifications and information herein are subject to change without notice.