

|              |          |   |
|--------------|----------|---|
| <b>SANYO</b> | No. 4677 | <b>LC66354B, 66356B, 66358B</b>   |
|              |          | <b>Four-bit Single-Chip Microcontrollers<br/>On-Chip 4 K/6 K/8 K-byte ROM</b> |

## Overview

The LC66354B, LC66356B and LC66358B are 42-pin package four-bit CMOS microcontrollers that integrate on a single chip all functions required in a control microcontroller, including ROM, RAM, I/O ports, serial interfaces, comparator inputs, three-value inputs, timers and an interrupt system. These products differ from the earlier LC66358A series in their power supply voltage range and operating speed specifications.

## Features and Functions

- ROM (with 4 K-, 6 K- and 8 K-byte capacities) and RAM (512 4-bit digits) on chip
- LC66000 series compatible instruction set (128 instructions)
- A total of 36 I/O port pins
- Two eight-bit serial interfaces that can be connected in cascade to form a 16-bit interface
- Instruction cycle time: 0.92 to 10  $\mu$ s (3 to 5.5 V)  
The earlier LC66358A series had instruction cycle times of from 1.96 to 10  $\mu$ s (at 3 to 5.5 V) and from 3.92 to 10  $\mu$ s (at 2.2 to 5.5 V).
- Powerful timer and prescaler functions  
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 eight interrupts and eight vector locations  
External interrupts: three interrupts and three vector locations  
Internal interrupts: five interrupts and five vector locations
- Flexible I/O functions  
Comparator inputs, three-value inputs, 20 mA drive outputs, 15 V withstand voltage, pull-up or open-drain option switching
- Runaway detection function (watchdog timer) option
- Eight-bit I/O function
- Power reduction functions using halt and hold modes
- Packages: DIP42S, QFP48E (QFP48E)

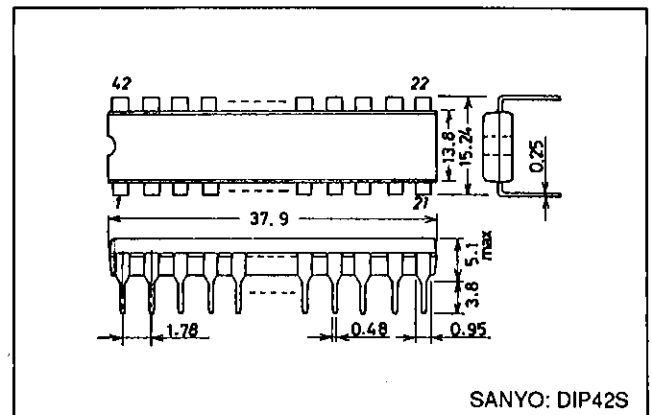
- Evaluation LSI: used together

- LC66599 (evaluation chip) + EVA850/800-TB6630X
- LC66E308 (on-chip EPROM microcontroller)

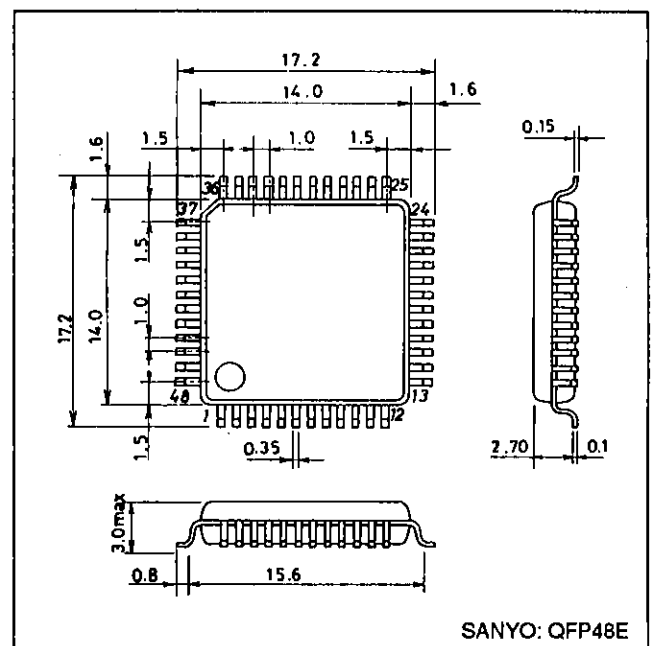
## Package Dimensions

unit: mm

### 3025B-DIP42S



### 3156-QFP48E



**SANYO Electric Co., Ltd. Semiconductor Business Headquarters**

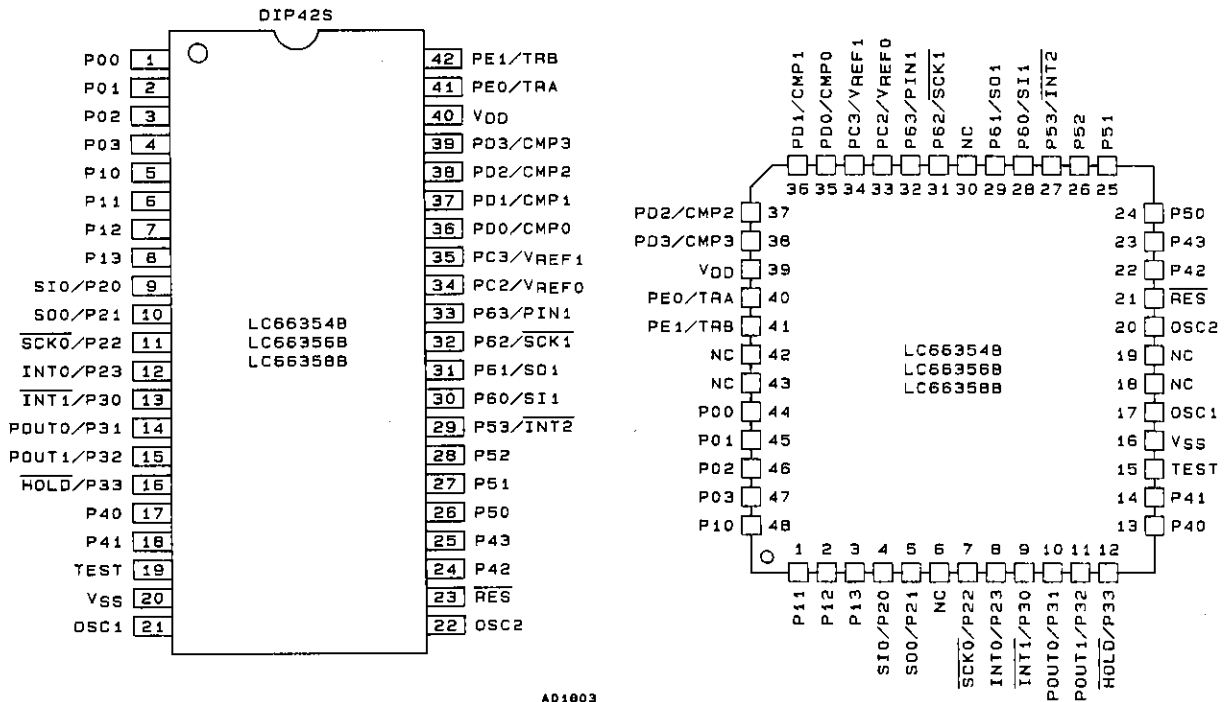
TOKYO OFFICE Tokyo Bldg., 1-10, 1 Chome, Ueno, Taito-ku, TOKYO, 110 JAPAN

## Series Structure

| Product name            | Pins   | ROM capacity            | RAM capacity | Package         |                | Features   |
|-------------------------|--------|-------------------------|--------------|-----------------|----------------|--|
| LC66304A/306A/308A      | 42, 48 | 4 K/6 K/8 K bytes       | 512 W        | DIP42S          | QFP48E         | Normal version<br>4.0 to 6.0 V/0.92 μs                     |
| LC66404A/406A/408A      | 42, 48 | 4 K/6 K/8 K bytes       | 512 W        | DIP42S          | QFP48E         |  |
| LC66506B/508B/512B/516B | 64     | 6 K/8 K/12 K/16 K bytes | 512 W        | DIP64S          | QFP64A         |  |
| LC66354A/356A/358A      | 42, 48 | 4 K/6 K/8 K bytes       | 512 W        | DIP42S          | QFP48E         | Low-voltage version<br>2.2 to 5.5 V/3.92 μs                |
| LC66354S/356S/358S*     | 44     | 4 K/6 K/8 K bytes       | 512 W        |                 | QFP44M         |  |
| LC66556A/558A/562A/566A | 64     | 6 K/8 K/12 K/16 K bytes | 512 W        | DIP64S          | QFP64E         | Low-voltage, high-speed version<br>3.0 to 5.5 V/0.92 μs    |
| LC66354B/356B/358B      | 42, 48 | 4 K/6 K/8 K bytes       | 512 W        | DIP42S          | QFP48E         |  |
| LC66556B/558B*          | 64     | 6 K/8 K bytes           | 512 W        | DIP64S          | QFP64E         |  |
| LC66562B/566B           | 64     | 12 K/16 K bytes         | 512 W        | DIP64S          | QFP64E         |  |
| LC66E308                | 42, 48 | EPROM, 8 K bytes        | 512 W        | DIC42S (window) | QFC48 (window) |  |
| LC66P308                | 42, 48 | OTPROM, 8 K bytes       | 512 W        | DIP42S          | QFP48E         | Evaluation window and OTP versions<br>4.5 to 5.5 V/0.92 μs |
| LC66E408                | 42, 48 | EPROM, 8 K bytes        | 512 W        | DIC42S (window) | QFC48 (window) |  |
| LC66P408                | 42, 48 | OTPROM, 8 K bytes       | 512 W        | DIP42S          | QFP48E         |  |
| LC66E516                | 64     | EPROM 16 K bytes        | 512 W        | DIC64S (window) | QFC64 (window) |  |
| LC66P516                | 64     | OTPROM 16 K bytes       | 512 W        | DIP64S          | QFP64E         |  |

Note: \* Under development

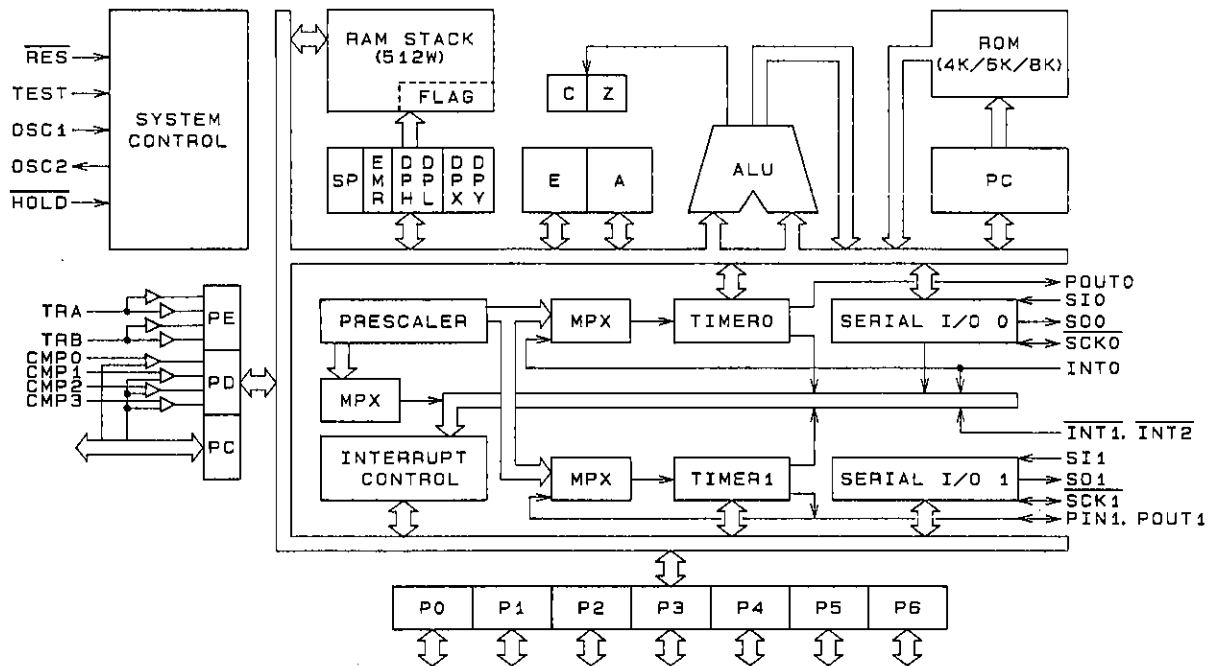
## Pin Assignment (Top view)



We recommend using reflow soldering methods to mount the QFP package version.

Contact your Sanyo sales representative to discuss process conditions if techniques in which the whole package is immersed in a solder bath (solder dip or spray techniques) are used.

System Block Diagram



A01805

Differences between the LC66354B, LC66356B and LC66358B and the LC6630X Series

| Parameter  | LC6630X series<br>(including the LC66599 evaluation chip)  | LC6635XB series   |
|--|--|---|
| System Differences   |  |   |
| • Hardware wait time (number of cycles) when HOLD mode is cleared            | 65536 cycles<br>At 4 MHz (T <sub>cy</sub> = 1 μs): About 64 ms   | 16384 cycles<br>At 4 MHz (T <sub>cy</sub> = 1 μs): About 16 ms                                    |
| • Value of timer 0 on reset (including the value after HOLD mode is cleared) | The value FFO is loaded.   | The value FFC is loaded.  |
| Main differences in product characteristics                                  |  |   |
| • Operating power supply voltage/operating speed (cycle time)                | LC66304A, 66306A, 66308A<br>4.0 to 6.0 V/0.92 to 10 μs<br>LC66E308, 66P308<br>4.5 to 5.5 V/0.92 to 10 μs | 3.0 to 5.5 V/0.92 to 10 μs<br>LC6635XA, 2.2 to 5.5 V/3.92 to 10 μs,<br>3.0 to 5.5 V/1.96 to 10 μs |

- Note: 1. An RC oscillator cannot be used with the LC66354B, LC66356B and LC66358B.  
 2. In addition, there are differences in the output currents, comparator input voltages and other aspects. For details, refer to the individual catalogs for the LC66308A, LC66E308 and the LC66P308.  
 3. These points require care when using the LC66E308 or LC66P308 for evaluation purposes.

Pin Function Overview

| Pin  | I/O | Overview   | Output drive type  | Option   | Value on reset             |
|--|-----|--|--|--|----------------------------|
| P00<br>P01<br>P02<br>P03                   | I/O | I/O ports P00 to P03<br><ul style="list-style-type: none"> <li>Input or output in 4-bit or 1-bit units</li> <li>P00 to P03 have control functions in HALT mode.</li> </ul>   | <ul style="list-style-type: none"> <li>P-channel: pull-up MOS type</li> <li>N-channel: intermediate sink current type</li> </ul>                                 | <ul style="list-style-type: none"> <li>Either with pull-up MOS or n-channel OD output</li> <li>Reset output level</li> </ul> | High or low level (option) |
| P10<br>P11<br>P12<br>P13                   | I/O | I/O ports P10 to P13<br><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>P-channel: pull-up MOS type</li> <li>N-channel: intermediate sink current type</li> </ul>                                 | <ul style="list-style-type: none"> <li>Either with pull-up MOS or n-channel OD output</li> <li>Reset output level</li> </ul> | High or low level (option) |
| P20/SI0<br>P21/SO0<br>P22/SCK0<br>P23/INT0 | I/O | I/O ports P20 to P23<br><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, the timer 0 event counter and pulse width measurement input.</li> </ul>   | <ul style="list-style-type: none"> <li>P-channel: CMOS type</li> <li>N-channel: intermediate sink current type (+15 V withstand voltage in OD)</li> </ul>        | <ul style="list-style-type: none"> <li>Either CMOS or n-channel OD output</li> </ul>   | H                          |
| P30/INT1<br>P31/POUT0<br>P32/POUT1         | I/O | I/O ports P30 to P32<br><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 square wave output from timer 0.</li> <li>P32 is also used for square wave output from timer 1 and PWM output.</li> </ul>  | <ul style="list-style-type: none"> <li>P-channel: CMOS type</li> <li>N-channel: intermediate sink current type (+15 V withstand voltage in OD)</li> </ul>        | <ul style="list-style-type: none"> <li>Either CMOS or n-channel OD output</li> </ul>   | H                          |
| P33/HOLD                                   | I   | Hold mode control input<br><ul style="list-style-type: none"> <li>Hold mode is entered if a HOLD instruction is executed when HOLD is low.</li> <li>When in hold mode, the CPU is reactivated by setting HOLD to the high level.</li> <li>P33 can also be used as an input port along with P30 to P32.</li> <li>When P33/HOLD is low, the CPU will not be reset by a low level on RES. Therefore, RES cannot be used in applications that set P33/HOLD low when power is first applied.</li> </ul> |  |  |                            |
| P40<br>P41<br>P42<br>P43                   | I/O | I/O ports P40 to P43<br><ul style="list-style-type: none"> <li>Input or output in 3-bit or 1-bit units</li> <li>I/O in 8-bit units when used in conjunction with P50 to P53</li> <li>Output of 8-bit ROM data when used in conjunction with P50 to P53</li> </ul>  | <ul style="list-style-type: none"> <li>P-channel: pull-up MOS type</li> <li>N-channel: intermediate sink current type (+15 V withstand voltage in OD)</li> </ul> | <ul style="list-style-type: none"> <li>Either with pull-up MOS or n-channel OD output</li> </ul>                             | H                          |

Continued on next page.

LC66354B, 66356B, 66358B

Continued from preceding page.

| Pin  | I/O    | Overview   | Output drive type  | Option  | Value on reset |
|--|--------|--|--|---|----------------|
| P50<br>P51<br>P52<br>P53/INT2                | I/O    | I/O ports P50 to P53<br><ul style="list-style-type: none"> <li>Input or output in 4-bit or 1-bit units</li> <li>I/O in 8-bit units when used in conjunction with P40 to P43</li> <li>Output of 8-bit ROM data when used in conjunction with P40 to P43</li> <li>P53 is also used for the INT2 interrupt request.</li> </ul>                            | <ul style="list-style-type: none"> <li>P-channel: pull-up MOS type</li> <li>N-channel: intermediate sink current type (+15 V withstand voltage in OD)</li> </ul> | <ul style="list-style-type: none"> <li>Either with pull-up MOS or n-channel OD output</li> </ul>                  | H              |
| P60/SI1<br>P61/SO1<br>P62/SCK1<br>P63/PIN1   | I/O    | I/O ports P60 to P63<br><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 SI1 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 as the timer 1 event counter input.</li> </ul> | <ul style="list-style-type: none"> <li>P-channel: CMOS type</li> <li>N-channel: intermediate sink current type (+15 V withstand voltage in OD)</li> </ul>        | <ul style="list-style-type: none"> <li>Either CMOS or n-channel OD output</li> </ul>                              | H              |
| PC2/VREF0<br>PC3/VREF1                       | I/O    | I/O ports PC2 and PC3<br><ul style="list-style-type: none"> <li>Output in 4-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>P-channel: CMOS type</li> <li>N-channel: intermediate sink current type</li> </ul>  | <ul style="list-style-type: none"> <li>Either CMOS or n-channel OD output</li> </ul>                              | H              |
| PD0/CMP0<br>PD1/CMP1<br>PD2/CMP2<br>PD3/CMP3 | I      | Dedicated input ports PD0 to PD3<br><ul style="list-style-type: none"> <li>Can be switched to use as comparator inputs under program control.</li> </ul> <p>The PD0 comparison voltage is VREF0.</p> <p>The PD1 to PD3 comparison voltage is VREF1.</p> <p>Comparisons can be specified in units of PD0, PD2, and PD2 and PD3 together.</p>            |  |   | Normal input   |
| PE0/TRA<br>PE1/TRB                           | I      | Dedicated input ports<br><ul style="list-style-type: none"> <li>Can be switched to function as three-value inputs under program control.</li> </ul>  |  |   | Normal input   |
| OSC1<br>OSC2                                 | I<br>O | System clock oscillator external connection<br>When an external clock is used, leave OSC2 open and input the clock signal to OSC1.   |  | <ul style="list-style-type: none"> <li>Selection of either ceramic oscillator or external clock input.</li> </ul> |                |
| $\overline{\text{RES}}$                      | I      | System reset input<br>The CPU is initialized if a low level is input to $\overline{\text{RES}}$ when the P33/HOLD pin is high.   |  |   |                |
| TEST   | I      | CPU test pin<br>This pin must be connected to $V_{SS}$ during normal operation.  |  |   |                |
| $V_{DD}$<br>$V_{SS}$                         |        | Power supply connections   |  |   |                |

Note: Pull-up MOS output:.....A pull-up MOS transistor is connected to the output circuit.

CMOS output:.....Complementary output

OD output:.....Open drain output

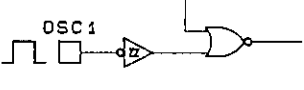
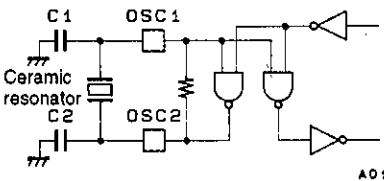
**User Option Types**

1. Port 0 and 1 reset time output level option

The output levels of ports 0 and 1 can be selected from the following two options in 4-bit units.

| Option                          | Conditions and notes           |
|---------------------------------|--------------------------------|
| High level output at reset time | Ports 0 and/or 1 in 4-bit sets |
| Low level output at reset time  | Ports 0 and/or 1 in 4-bit sets |

2. Oscillator circuit option

| Option             | Circuit  | Conditions and notes                         |
|--------------------|--|--|
| External clock     |  <p style="text-align: right;">A01805</p> | This input is a Schmitt specification input. |
| Ceramic oscillator |  <p style="text-align: right;">A01807</p> |  |

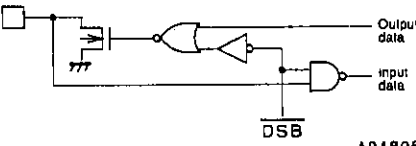
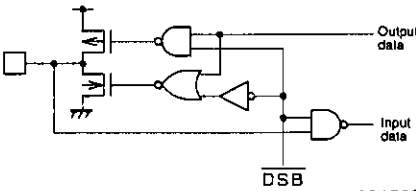
Note: There is no RC oscillator option.

3. Watchdog timer option

The presence or absence of a watchdog timer can be selected as an option.

4. Port output type option

- One of the following two output circuit options can be selected for each bit in ports P0, P1, P2, P3 (except for the P33/HOLD pin), P4, P5, P6 and PC.

| Option                           | Circuit  | Conditions and notes  |
|----------------------------------|--|---|
| Open drain output                |  <p style="text-align: right;">A01808</p> | P2, P3, P5 and P6 are Schmitt inputs.   |
| Built-in pull-up resistor output |  <p style="text-align: right;">A01809</p> | P2, P3, P5 and P6 are Schmitt inputs.<br>CMOS outputs (P2, P3, P6 and PC) and pull-up MOS outputs (P0, P1, P4 and P5) are differentiated according to the drive capacity of the p-channel transistor. |

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

## Specifications

Absolute Maximum Ratings at Ta = 25°C, VSS = 0 V

| Parameter                   | Symbol   | Applicable pins, notes                                       | Conditions | Ratings           | Unit | Note |
|-----------------------------|----------|--|------------|-------------------|------|------|
| Maximum supply voltage      | VDD max  | VDD  |            | -0.3 to +7.0      | V    |      |
| Input voltage               | VIN (1)  | P2, P3 (except for the P33/HOLD pin), P4, P5, P6             |            | -0.3 to +15.0     | V    | 1    |
|                             | VIN (2)  | Other inputs   |            | -0.3 to VDD + 0.3 | V    | 2    |
| Output voltage              | VOU (1)  | P2, P3 (except for the P33/HOLD pin), P4, P5, P6             |            | -0.3 to +15.0     | V    | 1    |
|                             | VOU (2)  | Other outputs  |            | -0.3 to VDD + 0.3 | V    | 2    |
| Output current per pin      | ION      | P0, P1, P2, P3 (except for the P33/HOLD pin), P4, P5, P6, PC |            | 20                | mA   | 3    |
|                             | -IOP (1) | P0, P1, P4, P5   |            | 2                 | mA   | 4    |
|                             | -IOP (2) | P2, P3 (except for the P33/HOLD pin), P6, PC                 |            | 4                 | mA   | 4    |
| Total pin current           | ΣION (1) | P0, P1, P2, P3, (except for the P33/HOLD pin), P40, P41      |            | 75                | mA   | 3    |
|                             | ΣION (2) | P5, P6, P42, P43, PC   |            | 75                | mA   | 3    |
|                             | ΣIOP (1) | P0, P1, P2, P3 (except for the P33/HOLD pin), P40, P41       |            | 25                | mA   | 4    |
|                             | ΣIOP (2) | P5, P6, P42, P43, PC   |            | 25                | mA   | 4    |
| Allowable power dissipation | Pd max   | Ta = -30 to +70°C  | DIP42S     | 600               | mW   |      |
|                             |          |  | QFP48E     | 430               | mW   | 5    |
| Operating temperature       | Topr     |  |            | -30 to +70        | °C   |      |
| Storage temperature         | Tstg     |  |            | -55 to +125       | °C   |      |

Note: 1. Applies to open drain output specification pins. The rating from the "other pin" entry applies for specifications other than the open drain output specification.

2. Levels up to the free-running oscillation level are allowed for the oscillator input and output pins.

3. Inflow current

4. Outflow current (Applies to the pull-up output specification and CMOS output specification pins.)

5. We recommend using reflow soldering methods to mount the QFP package version.

Contact your Sanyo sales representative to discuss process conditions if techniques in which the whole package is immersed in a solder bath (solder dip or spray techniques) are used.

Allowable Operating Ranges at Ta = -30 to +70°C, VSS = 0 V, VDD = 3.0 to 5.5 V unless otherwise specified

| Parameter                                    | Symbol     | Applicable pins   | Conditions                               | min      | typ | max         | Unit     | Note |
|--|------------|---|--|----------|-----|-------------|----------|------|
| Operating supply voltage                     | VDD        | VDD   | 0.92 ≤ Tcyc ≤ 10 μs                      | 3.0      |     | 5.5         | V        |      |
| Memory hold supply voltage                   | VDD (H)    | VDD   | In HOLD mode                             | 1.8      |     | 5.5         | V        |      |
| Input high level Voltage                     | VIH (1)    | P2, P3 (except for the P33/HOLD pin), P4, P5, P6        | With the output n-channel transistor off | 0.8 VDD  |     | 13.5        | V        | 1    |
|  | VIH (2)    | P33/HOLD, RES, OSC1                                     | With the output n-channel transistor off | 0.8 VDD  |     | VDD         | V        | 2    |
|  | VIH (3)    | P0, P1, PC, PD, PE                                      | With the output n-channel transistor off | 0.75 VDD |     | VDD         | V        | 3    |
|  | VIH (4)    | PE  | Using three-value input                  | 0.8 VDD  |     | VDD         | V        |      |
| Middle level input voltage                   | VIM        | PE  | Using three-value input                  | 0.4 VDD  |     | 0.6 VDD     | V        |      |
| Common mode input voltage range              | VCMM (1)   | PD0, PC2  | Using comparator input                   | 1.5      |     | VDD         | V        |      |
|  | VCMM (2)   | PD1, PD2, PD3, PC3                                      |  | VSS      |     | VDD - 1.5   | V        |      |
| Input low level voltage                      | VIL (1)    | P2, P3 (except for the P33/HOLD pin), P5, P6, RES, OSC1 | With the output n-channel transistor off |          |     | 0.2 VDD     | V        | 1    |
|  | VIL (2)    | P33/HOLD  | VDD = 1.8 to 5.5 V                       |          |     | 0.2 VDD     | V        |      |
|  | VIL (3)    | P0, P1, P4, PC, PD, PE, TEST                            | With the output n-channel transistor off | VSS      |     | 0.25 VDD    | V        | 3    |
|  | VIL (4)    | PE  | Using comparator input                   | VSS      |     | 0.2 VDD     | V        |      |
| Operating frequency (instruction cycle time) | fOP (Tcyc) |   |  | 0.4 (10) |     | 4.35 (0.92) | MHz (μs) |      |

Note: 1. Applies to open drain specification pins. However, the rating for VIH (2) applies to the P33/HOLD pin. Ports P2, P3 and P6 cannot be used as input pins when CMOS output specifications are used.

2. Applies to open drain specification pins.

3. When PE is used as a three-value input, VIH (4), VIM and VIL (4) apply. Port P3 cannot be used as input pins when CMOS output specifications are used.

Continued on next page.

Continued from preceding page.

| Parameter                       | Symbol          | Applicable pins          | Conditions | min  | typ | max | Unit | Note |
|---------------------------------|-----------------|--------------------------|------------|--|-----|-----|------|------|
| External clock input conditions | Frequency       | $f_{ext}$                | OSC1       | See Figure 1.<br>With the signal input to OSC1 and with OSC2 open (with external clock input selected for the oscillator circuit option) | 0.4 |     | 4.35 | MHz  |
|                                 | Pulse width     | $t_{extH}$<br>$t_{extL}$ |            | See Figure 1.<br>With the signal input to OSC1 and with OSC2 open (with external clock input selected for the oscillator circuit option) | 100 |     |      | ns   |
|                                 | Rise/fall times | $t_{extR}$<br>$t_{extF}$ |            | See Figure 1.<br>With the signal input to OSC1 and with OSC2 open (with external clock input selected for the oscillator circuit option) |     |     | 30   | ns   |

**Electrical Characteristics at  $T_a = -30$  to  $+70^\circ\text{C}$ ,  $V_{SS} = 0\text{ V}$ ,  $V_{DD} = 3.5$  to  $5.5\text{ V}$  unless otherwise specified**

| Parameter                  | Symbol                       | Applicable pins  | Conditions  | min            | typ          | max          | Unit          | Note |
|----------------------------|------------------------------|--|---|----------------|--------------|--------------|---------------|------|
| Input high level current   | $I_{IH(1)}$                  | P2, P3 (except for the P33/HOLD pin), P4, P5, P6             | $V_{IN} = 13.5$ ,<br>With the output n-channel transistor off   |                |              | 5.0          | $\mu\text{A}$ | 1    |
|                            | $I_{IH(2)}$                  | P0, P1, PC, OSC1, RES, P33/HOLD                              | $V_{IN} = V_{DD}$ ,<br>With the output n-channel transistor off |                |              | 1.0          | $\mu\text{A}$ | 1    |
|                            | $I_{IH(3)}$                  | PD, PE, PC2, PC3   | $V_{IN} = V_{DD}$ ,<br>With the output n-channel transistor off |                |              | 1.0          | $\mu\text{A}$ | 1    |
| Input low level current    | $I_{IL(1)}$                  | Inputs other than PD, PE, PC2 and PC3                        | $V_{IN} = V_{SS}$ ,<br>With the output n-channel transistor off | -1.0           |              |              | $\mu\text{A}$ | 2    |
|                            | $I_{IL(2)}$                  | PC2, PC3, PD, PE   | $V_{IN} = V_{SS}$ ,<br>With the output n-channel transistor off | -1.0           |              |              | $\mu\text{A}$ | 2    |
| Output high level voltage  | $V_{OH(1)}$                  | P2, P3 (except for the P33/HOLD pin) P6, PC                  | $I_{OH} = -1\text{ mA}$   | $V_{DD} - 1.0$ |              |              | V             | 3    |
|                            |                              |  | $I_{OH} = -0.1\text{ mA}$                                       | $V_{DD} - 0.5$ |              |              |               |      |
| Output pull-up current     | $V_{OH(2)}$                  | P0, P1, P4, P5   | $I_{OH} = -50\text{ }\mu\text{A}$                               | $V_{DD} - 1.0$ |              |              | V             | 4    |
|                            |                              |  | $I_{OH} = -30\text{ }\mu\text{A}$                               | $V_{DD} - 0.5$ |              |              |               |      |
| Output low level voltage   | $V_{OL(1)}$                  | P0, P1, P2, P3, P4, P5, P6, PC (except for the P33/HOLD pin) | $I_{OL} = 1.6\text{ mA}$  |                |              | 0.4          | V             | 5    |
| Output off leakage current | $V_{OL(2)}$                  | P0, P1, P2, P3, P4, P5, P6, PC (except for the P33/HOLD pin) | $I_{OL} = 8\text{ mA}$  |                |              | 1.5          | V             |      |
|                            |                              |  |   |                |              |              |               |      |
| Comparator offset voltage  | $I_{OFF(1)}$                 | P2, P3, P4, P5, P6   | $V_{IN} = 13.5\text{ V}$  |                |              | 5.0          | $\mu\text{A}$ | 5    |
|                            | $I_{OFF(2)}$                 | P0, P1, PC   | $V_{IN} = V_{DD}$   |                |              | 1.0          | $\mu\text{A}$ | 5    |
| Schmitt characteristics    | $V_{OFF(1)}$                 | PD1, PD2, PD3  | $V_{IN} = V_{SS}$ to $V_{DD} - 1.5\text{ V}$                    |                | $\pm 50$     | $\pm 300$    | mV            |      |
|                            | $V_{OFF(2)}$                 | PD0  | $V_{IN} = 1.5$ to $V_{DD}$                                      |                | $\pm 50$     | $\pm 300$    | mV            |      |
| Schmitt characteristics    | Hysteresis voltage           | $V_{HIS}$  |   |                | $0.1 V_{DD}$ |              | V             |      |
|                            | High level thresHOLD voltage | $V_{IH}$   | P2, P3, P5, P6, OSC1 (EXT), RES                                 |                | $0.5 V_{DD}$ | $0.8 V_{DD}$ | V             |      |
|                            | Low level thresHOLD voltage  | $V_{iL}$   |   |                | $0.2 V_{DD}$ | $0.5 V_{DD}$ | V             |      |

Continued on next page.

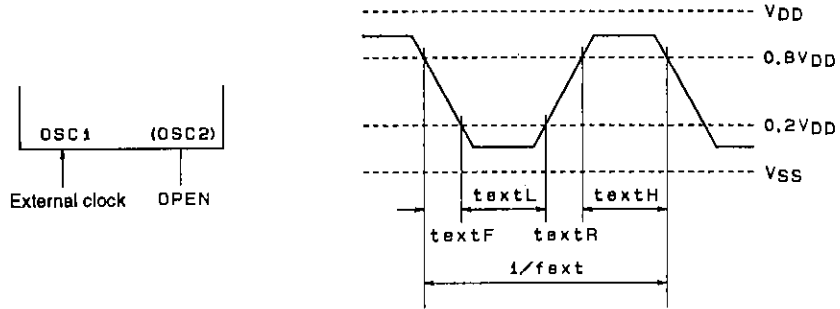


LC66354B, 66356B, 66358B

Continued from preceding page.

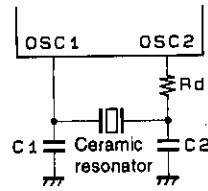
| Parameter                    |  | Symbol                   | Applicable pins                    | Conditions   | min   | typ  | max | Unit             | Note             |         |
|------------------------------|--|--------------------------|------------------------------------|--|---|------|-----|------------------|------------------|---------|
| Ceramic oscillator           | Oscillator frequency   | $f_{CF}$                 | OSC1, OSC2                         | Figure 2, 4 MHz  |   | 4.0  |     | MHz              |                  |         |
|                              | Oscillator stabilization time                                    | $t_{CFS}$                |                                    | Figure 3, 4 MHz  |   |      | 10  | ms               |                  |         |
| Serial clock                 | Cycle time   | Input                    | $\overline{SCK0}, \overline{SCK1}$ | The timing from Figure 4 and the test load from Figure 5   | 0.9   |      |     | $\mu s$          |                  |         |
|                              |  | Output                   |                                    |  | 2.0   |      |     | T <sub>cyc</sub> |                  |         |
|                              | Low level/high level pulse widths                                | Input                    |                                    |  | 0.4   |      |     | $\mu s$          |                  |         |
|                              |  | Output                   |                                    |  | 1.0   |      |     | T <sub>cyc</sub> |                  |         |
|                              | Rise/fall times  | Output                   |                                    |  | $t_{CKR}$<br>$t_{CKF}$  |      |     |                  | 0.1              | $\mu s$ |
| Serial input                 | Data setup time  | $t_{CK}$                 | SIO, SI1                           | Stipulated with respect to the rising edge timing for $\overline{SCK0}$ and $\overline{SCK1}$ from Figure 4  | 0.3   |      |     | $\mu s$          |                  |         |
|                              | Data hold time   | $t_{CKI}$                |                                    |  | 0.3   |      |     | $\mu s$          |                  |         |
| Serial output                | Output delay time  | $t_{CKO}$                | SO0, SO1                           | Stipulated with respect to the rising edge timing for $\overline{SCK0}$ and $\overline{SCK1}$ from Figure 4 and the test load shown in Figure 5  |   |      | 0.3 | $\mu s$          |                  |         |
| Pulse conditions             | INT0 high/low level pulse widths                                 | $t_{IOH}$<br>$t_{IOL}$   | INT0                               | Figure 6<br><ul style="list-style-type: none"> <li>Conditions such that the INT0 interrupt is accepted</li> <li>Conditions such that timer 0 event counter and pulse width measurement inputs are accepted.</li> </ul> | 2   |      |     | T <sub>cyc</sub> |                  |         |
|                              | High/low level pulse widths for interrupt inputs other than INT0 | $t_{IIH}$<br>$t_{IIL}$   | $\overline{INT1}, \overline{INT2}$ |  | <ul style="list-style-type: none"> <li>Conditions such that all interrupts are accepted</li> </ul>                | 2    |     |                  | T <sub>cyc</sub> |         |
|                              | PIN1 high/low level pulse widths                                 | $t_{PINH}$<br>$t_{PINL}$ | PIN1                               |  | <ul style="list-style-type: none"> <li>Conditions such that timer 1 event counter inputs are accepted.</li> </ul> | 2    |     |                  | T <sub>cyc</sub> |         |
|                              | $\overline{RES}$ high/low level pulse widths                     | $t_{RSH}$<br>$t_{RSL}$   | $\overline{RES}$                   |  | <ul style="list-style-type: none"> <li>Conditions such that reset occurs</li> </ul>                               | 3    |     |                  | T <sub>cyc</sub> |         |
| Comparator response speed    |  | $T_{RS}$                 | PD                                 | Figure 7   |   |      | 20  | ms               |                  |         |
| Operating mode current drain |  | $I_{DDOP}$               | $V_{DD}$                           | Using a 4 MHz ceramic oscillator   |   | 3.0  | 5.0 | mA               | 8                |         |
|                              |  |                          |                                    | Using a 4 MHz external clock   |   | 3.0  | 5.0 | mA               |                  |         |
| HALT mode current drain      |  | $I_{DDHALT}$             | $V_{DD}$                           | Using a 4 MHz ceramic oscillator   |   | 1.0  | 2.0 | mA               |                  |         |
|                              |  |                          |                                    | Using a 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. Common input and output ports with open-drain output specifications are specified for the state with the output n-channel transistor turned off. These pins cannot be used for input when the CMOS output specification option is selected.
2. Common input and output ports with open-drain output specifications are specified for the state with the output n-channel transistor turned off. Ratings for pull-up output specification pins are stipulated for the output pull-up current  $I_{PO}$ . These pins cannot be used for input when the CMOS output specification option is selected.
3. Stipulated for CMOS output specifications with the output n-channel transistor in the off state.
4. Stipulated for pull-up output specifications with the output n-channel transistor in the off state.
5. Stipulated for open-drain output specifications with the output n-channel transistor in the off state.
6. In the reset state



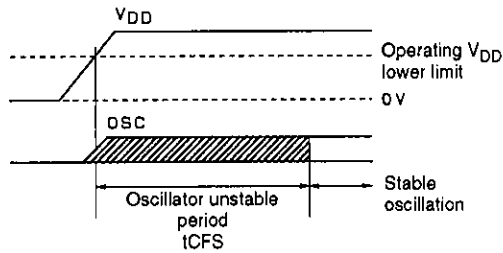
A01810

Figure 1 External Clock Input Waveform



A01811

Figure 2 Ceramic Oscillator Circuit

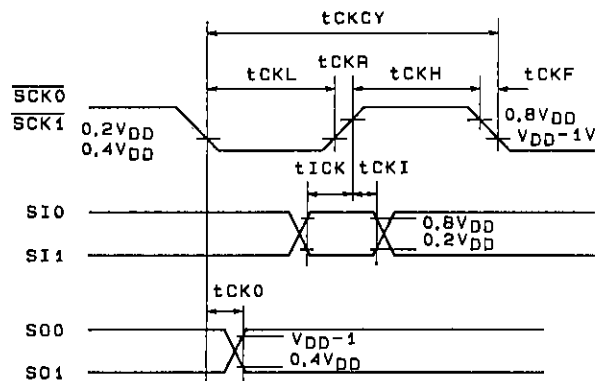


A01812

Figure 3 Oscillator Stabilization Time

Table 1 Ceramic Oscillator Guaranteed Constants

|                           |                          |                  |                          |                  |
|---------------------------|--------------------------|------------------|--------------------------|------------------|
| External capacitance type | 2 MHz (Murata) CSA2.00MG | C1 = 33 pF ± 10% | 2 MHz (Kyocera) KBR2.0MS | C1 = 47 pF ± 10% |
|                           |                          | C2 = 33 pF ± 10% |                          | C2 = 47 pF ± 10% |
|                           |                          | Rd = 0 Ω         |                          | Rd = 0 Ω         |
|                           | 4 MHz (Murata) CSA4.00MG | C1 = 33 pF ± 10% | 4 MHz (Kyocera) KBR4.0MS | C1 = 33 pF ± 10% |
|                           | C2 = 33 pF ± 10%         |                  | C2 = 33 pF ± 10%         |                  |
|                           | Rd = 0 Ω                 |                  | Rd = 0 Ω                 |                  |



A01813

Figure 4 Serial I/O Timing

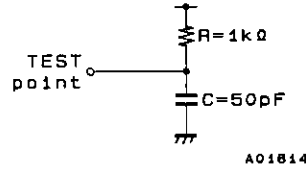


Figure 5 Timing Load

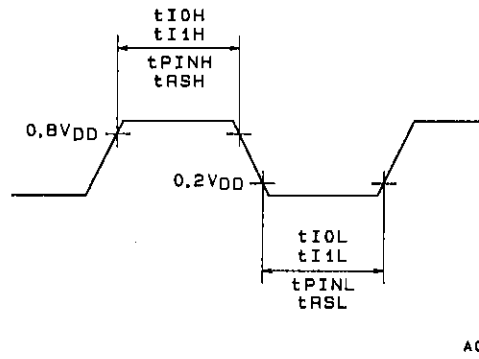


Figure 6 Input Timing for INT0, INT1, INT2, PIN1 and RES

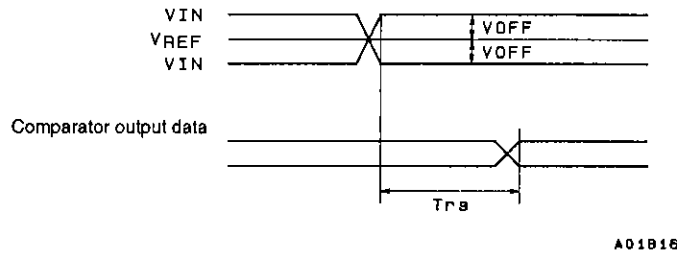
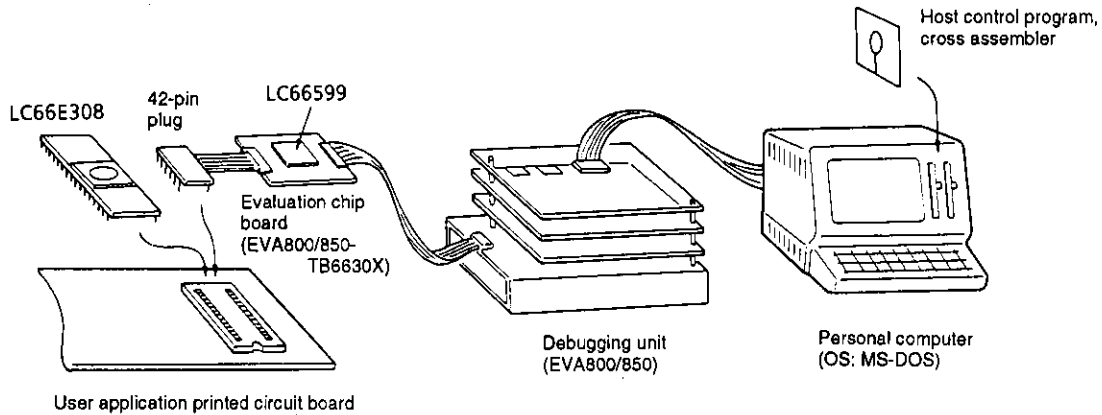


Figure 7 Comparator Response Speed  $T_{rs}$  Timing

**Application Development Tools**

Programs for the LC66354B, LC66356B and LC66358B microprocessors are developed on an IBM-PC compatible personal computer running the MS-DOS operating system. A cross assembler and other tools are available. To make application development more convenient, Sanyo also provides a program debugging unit (EVA800/850), an evaluation board (EVA800/850-TB6630X), an evaluation chip (LC66599) and an on-chip EPROM microprocessor (LC66E308).



**Structure of the Application Development Tools**

**1. Program debugging unit (EVA800/850)**

This is an emulator that provides functions for EPROM writing and serial data communications with external equipment (such as a host computer). It supports application development in machine language and program modification. Its main debugging functions include breaking, stepping and tracing. (The MPM6630X is used for the EVA800/850 monitor ROM.)

**2. Evaluation chip board (EVA800/850-TB6630X)**

The evaluation chip signals and ports are output to the 42-pin connector and when the output cable is connected, the evaluation chip board converts these signals to the same pin assignments as those on the mass production chip. The evaluation chip board includes jumpers for setting options and other states, and these jumper settings allow the evaluation chip to implement the same I/O circuit types and functions as the mass production chip. However, there are differences in the HOLD mode clear timing and the electrical characteristics.

**Jumper**

| Type                    | OSC           |                                      | Reset method       |   | Power supply to the user application board |  |
|-------------------------|---------------|--------------------------------------|--------------------|---|--|--|
| Jumper                  | Jumper 1 (J1) |                                      | Jumper 2 (J2: RES) |   | Jumper 3 (J3: V <sub>DD</sub> )            |  |
| Jumper setting and mode | EXT           | External oscillator (external clock) | INT (a)            | Reset by a RUN instruction from the host computer.                        | ON (a)                                     | V <sub>DD</sub> is supplied to the user application printed circuit board through the evaluation chip board. |
|                         | RC            | RC oscillator                        |                    |   |  |  |
|                         | CF            | CF oscillator                        | EXT (b)            | Reset by the reset circuit on the user application printed circuit board. | OFF (b)                                    | Separate power supplies on the user application printed circuit board and the evaluation chip board.         |

**Switches (SW9, SW10 and SW11)**

| Type                    | Port 0 and 1 output levels on reset |             |            |             | Watchdog timer presence or absence setting |                        |
|-------------------------|-------------------------------------|-------------|------------|-------------|--|------------------------|
| Switch                  | SW11: P0HL                          |             | SW10: P1HL |             | SW9: WDC                                   |                        |
| Switch setting and mode | ON                                  | Port 0 high | ON         | Port 1 high | ON   | Watchdog timer present |
|                         | OFF                                 | Port 0 low  | OFF        | Port 1 low  | OFF  | Watchdog timer absent  |

**Switches SW1 to SW8: Pull-up resistor option settings**

- Set the corresponding switch to the on position for built-in pull-up resistors, and set the switch to the off position for open drain output.
- These settings can be specified for individual pins.

3. Cross Assembler

| Cross assembler (file name) | Object microprocessors                                      | Limitations on program creation  |
|-----------------------------|---|--|
| LC66S.EXE                   | LC66354B, 66356B, 66358B<br>(LC66E308, 66P308)<br>(LC66599) | SB instruction limitations<br>• LC66354B : Only SB0 can be used.<br>• LC66356B, 66358B : Only SB0 and SB1 can be used.<br>(LC66E308, 66P308)<br>• LC66599 : SB0, SB1, SB2 and SB3 can be used. |

4. Simulation chip (See the LC66E308 individual product catalog for more details.)

The LC66E308 simulation chip is an on-chip EPROM microprocessor. Mounted configuration operation can be confirmed in the application product by using a dedicated conversion board (the W66EP308D/408D for DIP products and the W66EP308Q/408Q for QFP products) and writing programs with a commercial PROM writer.

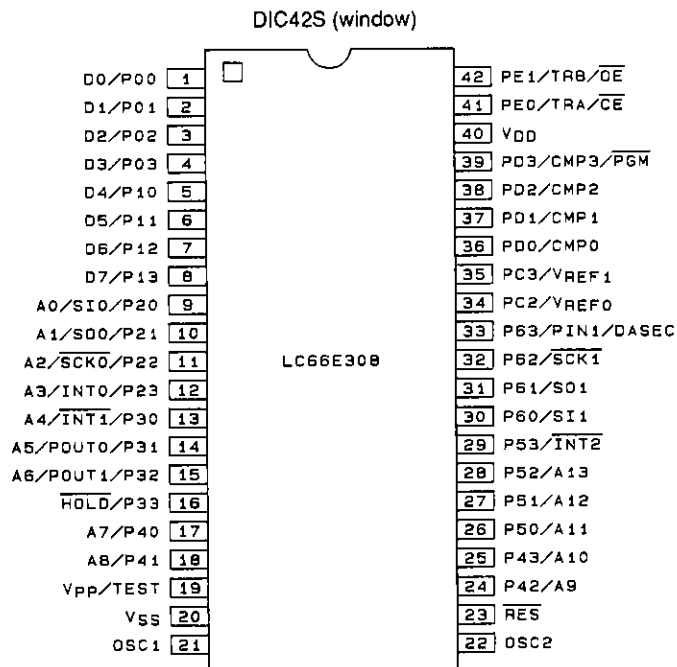
• Form

The LC66E308 has a pin arrangement and functions identical to those of the LC66354B, LC66356B and LC66358B. However, there are differences in the HOLD mode clear timing and the electrical characteristics. The figure below shows the pin assignment.

• Options

The options (the port 0 and 1 level at reset, the watchdog timer and the port output circuit types) for the microprocessor to be evaluated can be specified by EPROM data. (The next item describes the option data area and definitions.) This allows evaluation with the same peripheral circuits as those that will be used in the mass production product.

Pin Assignment



A01817

Option Data Area and Definitions

| ROM area | Bit    | Option item           | Relation between option and data             |   |
|----------|--------|-----------------------|--|---|
| 2000H    | 7      | Unused                | Must be set to zeros.                        |   |
|          | 6      |                       |  |   |
|          | 5      |                       |  |   |
|          | 4      | Oscillator option     | 1 = ceramic oscillator<br>0 = external clock |   |
|          | 3      | Unused                | Must be set to zero.                         |   |
|          | 2      | P1                    | Level of reset                               | 1 = high level<br>0 = low level           |
| 1        | P0     |                       |  |   |
|          | 0      | Watchdog timer option | 1 = present, 0 = absent                      |   |
| 2001H    | 7      | P13                   | Output circuit type<br><br>1 = PU, 0 = OD    |   |
|          | 6      | P12                   |  |   |
|          | 5      | P11                   |  |   |
|          | 4      | P10                   |  |   |
|          | 3      | P03                   |  |   |
|          | 2      | P02                   |  |   |
|          | 1      | P01                   |  |   |
|          | 0      | P00                   |  |   |
| 2002H    | 7      | Unused                | Must be set to zero.                         |   |
|          | 6      | P32                   |  |   |
|          | 5      | P31                   |  |   |
|          | 4      | P30                   |  | Output circuit type<br><br>1 = PU, 0 = OD |
|          | 3      | P23                   |  |   |
|          | 2      | P22                   |  |   |
|          | 1      | P21                   |  |   |
|          | 0      | P20                   |  |   |
| 2003H    | 7      | P53                   | Output circuit type<br><br>1 = PU, 0 = OD    |   |
|          | 6      | P52                   |  |   |
|          | 5      | P51                   |  |   |
|          | 4      | P50                   |  |   |
|          | 3      | P43                   |  |   |
|          | 2      | P42                   |  |   |
|          | 1      | P41                   |  |   |
|          | 0      | P40                   |  |   |
| 2004H    | 7 to 4 | Unused                | Must be set to zero.                         |   |
|          | 3      | P63                   |  |   |
|          | 2      | P62                   |  | Output circuit type<br><br>1 = PU, 0 = OD |
|          | 1      | P61                   |  |   |
|          | 0      | P60                   |  |   |
| 2005H    | 7 to 0 | Unused                | Must be set to zero.                         |   |
| 2006H    | 7 to 0 | Unused                | Must be set to zero.                         |   |
| 2007H    | 7 to 4 | Unused                | Must be set to zero.                         |   |
|          | 3      | PC3                   |  | Output circuit type<br><br>1 = PU, 0 = OD |
|          | 2      | PC2                   |  |   |
|          | 1      | Unused                |  | Must be set to zero.                      |
| 0        |        |                       |  |   |

**LC663XX 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 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
- v: Logical or
- +: Addition
- : Subtraction
- : Taking the one's complement

Instructions

| Instruction group                             | 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                          | $AC \leftarrow 0$<br>(Equivalent to LAI0.)   | Clear AC.   | ZF             | 1              |
|   | DAA                | Decimal adjust AC in addition    | 1                | 1              | 0              | 0              | 0               | 0                | 1              | 0              | 2                    | 2                          | $AC \leftarrow (AC) + 6$<br>(Equivalent to ADI6.)                                    | Add six to AC.  | ZF             |                |
|   | DAS                | Decimal adjust AC in subtraction | 1                | 1              | 0              | 0              | 1               | 1                | 1              | 1              | 2                    | 2                          | $AC \leftarrow (AC) + 10$<br>(Equivalent to ADIOAH.)                                 | Add 10 to AC.   | ZF             |                |
|   | CLC                | Clear CF                         | 0                | 0              | 0              | 1              | 1               | 1                | 1              | 0              | 1                    | 1                          | $CF \leftarrow 0$  | Clear CF to 0.  | CF             |                |
|   | STC                | Set CF                           | 0                | 0              | 0              | 1              | 1               | 1                | 1              | 1              | 1                    | 1                          | $CF \leftarrow 1$  | Set CF to 1.  | CF             |                |
|   | CMA                | Complement AC                    | 0                | 0              | 0              | 1              | 1               | 0                | 0              | 0              | 1                    | 1                          | $AC \leftarrow (\overline{AC})$  | Take the one's complement of AC.  | ZF             |                |
|   | IA                 | Increment AC                     | 0                | 0              | 0              | 1              | 0               | 1                | 0              | 0              | 1                    | 1                          | $AC \leftarrow (AC) + 1$   | Increment AC.   | ZF, CF         |                |
|   | DA                 | Decrement AC                     | 0                | 0              | 1              | 0              | 0               | 1                | 0              | 0              | 1                    | 1                          | $AC \leftarrow (AC) - 1$   | Decrement AC.   | ZF, CF         |                |
|   | RAR                | Rotate AC right through CF       | 0                | 0              | 0              | 1              | 0               | 0                | 0              | 0              | 1                    | 1                          | $AC_3 \leftarrow (CF)$ ,<br>$AC_n \leftarrow (AC_n + 1)$ ,<br>$CF \leftarrow (AC_0)$ | Shift AC (including CF) right.  | CF             |                |
|   | RAL                | Rotate AC left through CF        | 0                | 0              | 0              | 0              | 0               | 0                | 0              | 1              | 1                    | 1                          | $AC_0 \leftarrow (CF)$ ,<br>$AC_n + 1 \leftarrow (AC_n)$ ,<br>$CF \leftarrow (AC_3)$ | Shift AC (including CF) left.   | CF, ZF         |                |
|   | TAE                | Transfer AC to E                 | 0                | 1              | 0              | 0              | 0               | 1                | 0              | 1              | 1                    | 1                          | $E \leftarrow (AC)$  | Move the contents of AC to E.   |                |                |
|   | TEA                | Transfer E to AC                 | 0                | 1              | 0              | 0              | 0               | 1                | 1              | 0              | 1                    | 1                          | $AC \leftarrow (E)$  | Move the contents of E to AC.   | ZF             |                |
| XAE   | Exchange AC with E | 0                                | 1                | 0              | 0              | 0              | 1               | 0                | 0              | 1              | 1                    | $(AC) \leftrightarrow (E)$ | Exchange the contents of AC and E.   |   |                |                |
| Memory manipulation instruction               | IM                 | Increment M                      | 0                | 0              | 0              | 1              | 0               | 0                | 1              | 0              | 1                    | 1                          | $M(HL) \leftarrow [M(HL)] + 1$   | Increment M (HL).   | ZF, CF         |                |
|   | DM                 | Decrement M                      | 0                | 0              | 1              | 0              | 0               | 0                | 1              | 0              | 1                    | 1                          | $M(HL) \leftarrow [M(HL)] - 1$   | Decrement M (HL).   | ZF, CF         |                |
|   | IMDR i8            | Increment M direct               | 1                | 1              | 0              | 0              | 0               | 1                | 1              | 1              | 2                    | 2                          | $M(i8) \leftarrow [M(i8)] + 1$   | Increment M (i8).   | ZF, CF         |                |
|   | DMDR i8            | Decrement M direct               | 1                | 1              | 0              | 0              | 0               | 0                | 1              | 1              | 2                    | 2                          | $M(i8) \leftarrow [M(i8)] - 1$   | 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                          | $[M(HL), t2] \leftarrow 1$   | 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                          | $[M(HL), t2] \leftarrow 0$   | 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                          | $AC \leftarrow (AC) + [M(HL)]$   | 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              | 1               | 0                | 0              | 1              | 2                    | 2                          | $AC \leftarrow (AC) + [M(i8)]$   | 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                          | $AC \leftarrow (AC) + [M(HL)] + (CF)$  | 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              | 2                    | 2                          | $AC \leftarrow (AC) + i_3, i_2, i_1, i_0$  | 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                          | $AC \leftarrow [M(HL)] - (AC) - (CF)$  | Subtract the contents of AC and CF from M (HL) as two's complement values and store the result in AC. | ZF, CF         | 2              |
|   | ANDA               | And M with AC then store AC      | 0                | 0              | 0              | 0              | 0               | 1                | 1              | 1              | 1                    | 1                          | $AC \leftarrow (AC) \wedge [M(HL)]$  | Take the logical and of AC and M (HL) and store the result in AC.                                     | ZF             |                |

Note: 1. Has a vertical skip function.  
2. CF will be zero if there was a borrow and one otherwise.

Continued on next page.



Continued from preceding page.

| Instruction group                             | 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>       |                |                        |                  |    |        |                  |   |   |                  |   |   |        |  |
| Arithmetic, logic and comparison instructions | ORA   | Or M with AC then store AC           | 0  | 0              | 0              | 0  | 0   | 1                    | 0              | 1                    | 1                      | 1              | $AC \leftarrow (AC) \vee [M(HL)]$                     | Take the logical or of AC and M (HL) and store the result in AC.  | ZF                   |                |                        |                  |    |        |                  |   |   |                  |   |   |        |  |
|   | EXL   | Exclusive or M with AC then store AC | 0  | 0              | 0              | 1  | 0   | 1                    | 0              | 1                    | 1                      | 1              | $AC \leftarrow (AC) \nabla [M(HL)]$                   | 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              | $[M(HL)] + (AC) + 1$                                  | Compare the contents of AC and M (HL) and set or clear CF and ZF according to the result.<br><table border="1"> <thead> <tr> <th>Magnitude comparison</th> <th>CF</th> <th>ZF</th> </tr> </thead> <tbody> <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> </tbody> </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<br>1 0 1 0                   | 1 1 1 1<br>i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub> | 2              | 2              | $i_3 i_2 i_1 i_0 + (AC) + 1$   | Compare the contents of AC and the immediate data i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub> and set or clear CF and ZF according to the result.<br><table border="1"> <thead> <tr> <th>Magnitude comparison</th> <th>CF</th> <th>ZF</th> </tr> </thead> <tbody> <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> </tbody> </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 <sub>L</sub> with immediate data | 1 1 0 0<br>1 0 1 1                   | 1 1 1 1<br>i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub> | 2              | 2              | ZF ← 1<br>if (DP <sub>L</sub> ) = i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub><br>ZF ← 0<br>if (DP <sub>L</sub> ) = i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub> | Compare the contents of DP <sub>L</sub> with the immediate data. Set ZF if identical and clear ZF if not.   | ZF                   |                |                      |                        |                |   |   |                      |                |                        |                  |    |        |                  |   |   |                  |   |   |        |  |
| CMB i2  | Compare AC bit with M data bit              | 1 1 0 0<br>1 1 0 1                   | 1 1 1 1<br>0 0 i <sub>1</sub> i <sub>0</sub>                           | 2              | 2              | ZF ← 1<br>if (AC, i <sub>2</sub> ) = [M(HL), i <sub>2</sub> ]<br>ZF ← 0<br>if (AC, i <sub>2</sub> ) = [M(HL), i <sub>2</sub> ]   | Compare the corresponding bits specified by i <sub>0</sub> and i <sub>1</sub> 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              | $AC \leftarrow M(HL)$<br>$E \leftarrow M(HL + 1)$     | Load the contents of M2 (HL) into AC, E.  |                      |                |                        |                  |    |        |                  |   |   |                  |   |   |        |  |
|   | LAI i4                                      | Load AC with immediate data          | 1  | 0              | 0              | 0  | i <sub>3</sub>  | i <sub>2</sub>       | i <sub>1</sub> | i <sub>0</sub>       | 1                      | 1              | $AC \leftarrow i_3 i_2 i_1 i_0$                       | Load the immediate data into AC.  | ZF                   | 3              |                        |                  |    |        |                  |   |   |                  |   |   |        |  |
|   | LADR i8                                     | Load AC from M direct                | 1  | 1              | 0              | 0  | i <sub>7</sub>  | i <sub>6</sub>       | i <sub>5</sub> | i <sub>4</sub>       | 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              | $M(HL) \leftarrow (AC)$<br>$M(HL + 1) \leftarrow (E)$ | Store the contents of AC, E into M2(HL).  |                      |                |                        |                  |    |        |                  |   |   |                  |   |   |        |  |
|   | LA reg                                      | Load AC from M (reg)                 | 0  | 1              | 0              | 0  | 1   | 0                    | i <sub>0</sub> | 0                    | 1                      | 1              | $AC \leftarrow [M(reg)]$                              | Load the contents of M (reg) into AC. The reg is either HL or XY depending on i <sub>0</sub> .<br><table border="1"> <thead> <tr> <th>reg</th> <th>i<sub>0</sub></th> </tr> </thead> <tbody> <tr> <td>HL</td> <td>0</td> </tr> <tr> <td>XY</td> <td>1</td> </tr> </tbody> </table>  | reg                  | i <sub>0</sub> | HL                     | 0                | XY | 1      | ZF               |   |   |                  |   |   |        |  |
| reg   | i <sub>0</sub>                              |                                      |  |                |                |  |   |                      |                |                      |                        |                |   |   |                      |                |                        |                  |    |        |                  |   |   |                  |   |   |        |  |
| HL  | 0   |                                      |  |                |                |  |   |                      |                |                      |                        |                |   |   |                      |                |                        |                  |    |        |                  |   |   |                  |   |   |        |  |
| XY  | 1   |                                      |  |                |                |  |   |                      |                |                      |                        |                |   |   |                      |                |                        |                  |    |        |                  |   |   |                  |   |   |        |  |

Note: 3. Has a vertical skip function.

Continued on next page.

Continued from preceding page.

| Instruction group           | 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> |    |   |    |   |  |  |
| 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                    | 2    | AC ← [M (reg)]<br>DP <sub>L</sub> ← (DP <sub>L</sub> ) + 1<br>or DP <sub>Y</sub> ← (DP <sub>Y</sub> ) + 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  | 4              |    |   |    |   |  |  |
|                             | LA reg, D   | Load AC from M (reg) then decrement reg     | 0              | 1              | 0              | 1               | 1                | 0         | t <sub>0</sub> | 1                    | 2    | AC ← [M (reg)]<br>DP <sub>L</sub> ← (DP <sub>L</sub> ) - 1<br>or DP <sub>Y</sub> ← (DP <sub>Y</sub> ) - 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  | 5              |    |   |    |   |  |  |
|                             | 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                    | 2    | (AC) ↔ [M (reg)]<br>DP <sub>L</sub> ← (DP <sub>L</sub> ) + 1<br>or DP <sub>Y</sub> ← (DP <sub>Y</sub> ) + 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 as that for the XA reg instruction.   | ZF  | 6              |    |   |    |   |  |  |
|                             | XA reg, D   | Exchange AC with M (reg) then decrement reg | 0              | 1              | 0              | 1               | 1                | 1         | t <sub>0</sub> | 1                    | 2    | (AC) ↔ [M (reg)]<br>DP <sub>L</sub> ← (DP <sub>L</sub> ) - 1<br>or DP <sub>Y</sub> ← (DP <sub>Y</sub> ) - 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 as that for the XA reg instruction.   | ZF  | 7              |    |   |    |   |  |  |
| XADR i8                     | Exchange AC with M direct                             | 1   | 1              | 0              | 0              | 1               | 0                | 0         | 0              | 2                    | 2    | (AC) ↔ [M (i8)]   | Exchange the contents of AC with M (i8).   |   |                |    |   |    |   |  |  |
| LEAI i8                     | Load E & AC with immediate data                       | 1   | 1              | 0              | 0              | 0               | 1                | 1         | 0              | 2                    | 2    | E ← i <sub>7</sub> i <sub>6</sub> i <sub>5</sub> i <sub>4</sub><br>AC ← i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub> | 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.  |   |                |    |   |    |   |  |  |

Note: 4. ZF is set according to the result of incrementing DP<sub>L</sub> or DP<sub>Y</sub>.  
 5. ZF is set according to the result of decrementing DP<sub>L</sub> or DP<sub>Y</sub>.  
 6. ZF is set according to the result of incrementing DP<sub>L</sub> or DP<sub>Y</sub>.  
 7. ZF is set according to the result of decrementing DP<sub>L</sub> or DP<sub>Y</sub>.

Continued on next page.

Continued from preceding page.

| Instruction group                      | 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> |                 |                  |           |                           |  |   |    |  |
| 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 <sub>H</sub> ← 0<br>DP <sub>L</sub> ← i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>   | 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              | 1              | 1              | 1               | 1                | 2         | 2                         | DP <sub>H</sub> ← i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>  | Load the immediate data i4 into DP <sub>H</sub> .                               |    |  |
|  | LLI i4                           | Load DP <sub>L</sub> with immediate data  |                | 1              | 1              | 0              | 0              | 1              | 1              | 1               | 1                | 2         | 2                         | DP <sub>L</sub> ← i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub>  | 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              | 0              | 0              | 0               | 0                | 2         | 2                         | DP <sub>H</sub> ← i <sub>7</sub> i <sub>6</sub> i <sub>5</sub> i <sub>4</sub><br>DP <sub>L</sub> ← i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub> | 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              | 0              | 0              | 1               | 0                | 2         | 2                         | DP <sub>X</sub> ← i <sub>7</sub> i <sub>6</sub> i <sub>5</sub> i <sub>4</sub><br>DP <sub>Y</sub> ← i <sub>3</sub> i <sub>2</sub> i <sub>1</sub> i <sub>0</sub> | Load the immediate data into DP <sub>X</sub> , DP <sub>Y</sub> .                |    |  |
|  | 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                | 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               | 1                | 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                | 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               | 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                | 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               | 1                | 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                | 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               | 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                         | F <sub>n</sub> ← 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                         | F <sub>n</sub> ← 0   | Clear the flag specified by n4 to 0.  | ZF |  |

Continued on next page.

Continued from preceding page.

| Instruction group                | 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> |                 |   |  |   |                      |      |
| Jump and subroutine instructions | JMP addr                      | Jump in the current bank                                   | 1                | 1              | 1              | 0              | P <sub>11</sub> | P <sub>10</sub> | P <sub>9</sub> | P <sub>8</sub> | 2               | 2   | PC12 ← PC12<br>PC11 to 0 ← P <sub>11</sub> to P <sub>0</sub>   | Jump to the location in the same bank specified by the immediate data P12.  |                      | 8    |
|                                  | JPEA                          | Jump to the address stored at E and AC in the current page | 0                | 0              | 1              | 0              | 0               | 1               | 1              | 1              | 1               | 1   | PC12 to PC8 ← PC12 to PC8<br>PC7 to 4 ← (E)<br>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              | 0               | P <sub>10</sub> | P <sub>9</sub> | P <sub>8</sub> | 2               | 2   | PC12, 11 ← 0<br>PC10 to 0 ← P <sub>10</sub> to P <sub>0</sub><br>M4 (SP) ← (CF, ZF, PC12 to 0)<br>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   | PC12 to 6, PC1 to 0 ← 0<br>PC5 to 2 ← P <sub>3</sub> to P <sub>0</sub><br>M4 (SP) ← (CF, ZF, PC12 to 0)<br>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.   |                      |      |
|                                  | PUSH reg                      | Push reg on M2 (SP)  | 1                | 1              | 0              | 0              | 1               | 1               | 1              | 1              | 2               | 2   | M2 (SP) ← (reg)<br>SP ← (SP) - 2   | Store the contents of reg in M2 (SP). Subtract 2 from SP after the store.   |                      |      |
|                                  | POP reg                       | Pop reg off M2 (SP)  | 1                | 1              | 0              | 0              | 1               | 1               | 1              | 1              | 2               | 2   | SP ← (SP) + 2<br>reg ← [M2 (SP)]   | Add 2 to SP and then load the contents of M2 (SP) into reg. The relation between i <sub>1</sub> i <sub>0</sub> 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<br>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<br>PC ← [M4 (SP)]<br>CF, ZF ← [M4 (SP)] | Return from a subroutine or interrupt handling routine. ZF and CF are restored.  | ZF, CF  |                      |      |
| Branch instructions              | BAI2 addr                     | Branch on AC bit   | 1                | 1              | 0              | 1              | 0               | 0               | t <sub>1</sub> | t <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><br>if (AC, t <sub>2</sub> ) = 1     | Branch to the location in the same page specified by P <sub>0</sub> to P <sub>7</sub> if the bit in AC specified by the immediate data t <sub>1</sub> t <sub>0</sub> is one.      |                      |      |
|                                  | MNAI2 addr                    | Branch on no AC bit  | 1                | 0              | 0              | 1              | 0               | 0               | t <sub>1</sub> | t <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><br>if (AC, t <sub>2</sub> ) = 0     | Branch to the location in the same page specified by P <sub>0</sub> to P <sub>7</sub> if the bit in AC specified by the immediate data t <sub>1</sub> t <sub>0</sub> is zero.     |                      |      |
|                                  | BMI2 addr                     | Branch on M bit  | 1                | 1              | 0              | 1              | 0               | 1               | t <sub>1</sub> | t <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><br>if [M (HL), t <sub>2</sub> ] = 1 | Branch to the location in the same page specified by P <sub>0</sub> to P <sub>7</sub> if the bit in M (HL) specified by the immediate data t <sub>1</sub> t <sub>0</sub> is one.  |                      |      |
|                                  | BNMI2 addr                    | Branch on no M bit   | 1                | 0              | 0              | 1              | 0               | 1               | t <sub>1</sub> | t <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><br>if [M (HL), t <sub>2</sub> ] = 0 | Branch to the location in the same page specified by P <sub>0</sub> to P <sub>7</sub> if the bit in M (HL) specified by the immediate data t <sub>1</sub> t <sub>0</sub> is zero. |                      |      |

Note: 8. This becomes PC12 + (PC12) immediately following a BANK instruction.

Continued on next page.

Continued from preceding page.

| Instruction group   | 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>            |  |   |                 |  |  |             |                      |      |
| Branch instructions | BPT2 addr  | Branch on port bit  | 1 1 0 1<br>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><br>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><br>P <sub>4</sub> P <sub>3</sub> P <sub>2</sub><br>P <sub>1</sub> P <sub>0</sub><br>if [P<br>(DP <sub>L</sub> ), t <sub>2</sub> ]<br>= 1 | Branch to the location in the same page specified by P <sub>0</sub> to P <sub>7</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.          |             | 9                    |      |
|                     | BNPT2 addr | Branch on no port bit                                       | 1 0 0 1<br>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><br>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><br>P <sub>4</sub> P <sub>3</sub> P <sub>2</sub><br>P <sub>1</sub> P <sub>0</sub><br>if [P<br>(DP <sub>L</sub> ), t <sub>2</sub> ]<br>= 0 | Branch to the location in the same page specified by P <sub>0</sub> to P <sub>7</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.         |             | 9                    |      |
|                     | BC addr    | Branch on CF  | 1 1 0 1<br>P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub> | 1 1 0 0<br>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><br>P <sub>4</sub> P <sub>3</sub> P <sub>2</sub><br>P <sub>1</sub> P <sub>0</sub><br>if (CF)<br>= 1                                       | Branch to the location in the same page specified by P <sub>0</sub> to P <sub>7</sub> if CF is one.  |             |                      |      |
|                     | BNC addr   | Branch on no CF   | 1 0 0 1<br>P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub> | 1 1 0 0<br>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><br>P <sub>4</sub> P <sub>3</sub> P <sub>2</sub><br>P <sub>1</sub> P <sub>0</sub><br>if (CF)<br>= 0                                       | Branch to the location in the same page specified by P <sub>0</sub> to P <sub>7</sub> if CF is zero.   |             |                      |      |
|                     | BZ addr    | Branch on ZF  | 1 1 0 1<br>P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub> | 1 1 0 1<br>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><br>P <sub>4</sub> P <sub>3</sub> P <sub>2</sub><br>P <sub>1</sub> P <sub>0</sub><br>if (ZF)<br>= 1                                       | Branch to the location in the same page specified by P <sub>0</sub> to P <sub>7</sub> if ZF is one.  |             |                      |      |
|                     | BNZ addr   | Branch on no ZF   | 1 0 0 1<br>P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub> | 1 1 0 1<br>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><br>P <sub>4</sub> P <sub>3</sub> P <sub>2</sub><br>P <sub>1</sub> P <sub>0</sub><br>if (ZF)<br>= 0                                       | Branch to the location in the same page specified by P <sub>0</sub> to P <sub>7</sub> if ZF is zero.   |             |                      |      |
|                     | BFn4 addr  | Branch on flag bit  | 1 1 1 1<br>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><br>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><br>P <sub>4</sub> P <sub>3</sub> P <sub>2</sub><br>P <sub>1</sub> P <sub>0</sub><br>if (Fn)<br>= 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<br>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><br>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><br>P <sub>4</sub> P <sub>3</sub> P <sub>2</sub><br>P <sub>1</sub> P <sub>0</sub><br>if (Fn)<br>= 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 ← (P <sub>0</sub> )   | 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<br>0 1 1 0   | 1 1 1 1<br>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<br>1 1 0 1   | 1 1 1 1<br>0 1 0 0   | 2 | 2               | E ← [P (4)]<br>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<br>0 1 1 1   | 1 1 1 1<br>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<br>1 1 0 1   | 1 1 1 1<br>0 1 0 1   | 2 | 2               | P (4) ← (E)<br>P (5) ← (AC)  | Output the contents of E and AC to ports P (4) and P (5) respectively.   |             |                      |      |

Note: 9. 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.

LC66354B, 66356B, 66358B

Continued from preceding page.

| Instruction group              | 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> |                 |                  |                              |  |   |  |    |  |
| IO instructions                | SPB t2        | Set port bit                             | 0                | 0              | 0              | 0              | 1              | 0              | I <sub>1</sub> | I <sub>0</sub> | 1               | 1                | $[P(DP_L), t2] \leftarrow 1$ | Set to one the bit in port P (DP <sub>L</sub> ) specified by the immediate data I <sub>1</sub> I <sub>0</sub> .    |   |  |    |  |
|                                | RPB t2        | Reset port bit                           | 0                | 0              | 1              | 0              | 1              | 0              | I <sub>1</sub> | I <sub>0</sub> | 1               | 1                | $[P(DP_L), t2] \leftarrow 0$ | Clear to zero the bit in port P (DP <sub>L</sub> ) specified by the immediate data I <sub>1</sub> I <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> | P <sub>3</sub>  | P <sub>2</sub>   | P <sub>1</sub>               | P <sub>0</sub>   | $P(P_3 \text{ to } P_0) \leftarrow [P(P_3 \text{ to } 0)] \vee I_{3 \text{ to } 0}$ | 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> | P <sub>3</sub>  | P <sub>2</sub>   | P <sub>1</sub>               | P <sub>0</sub>   | $P(P_3 \text{ to } P_0) \leftarrow [P(P_3 \text{ to } 0)] \vee I_{3 \text{ to } 0}$ | 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 |  |
| 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               | 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               | 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              | 1              | 1               | 2                | Start timer 0 counter        | Start the timer 0 counter.   |   |  |    |  |
|                                | START1        | Start timer 1                            | 1                | 1              | 0              | 0              | 1              | 1              | 1              | 1              | 1               | 2                | Start timer 1 counter        | Start the timer 1 counter.   |   |  |    |  |
|                                | STOP0         | Stop timer 0                             | 1                | 1              | 0              | 0              | 1              | 1              | 1              | 1              | 1               | 2                | Stop timer 0 counter         | Stop the timer 0 counter.  |   |  |    |  |
|                                | STOP1         | Stop timer 1                             | 1                | 1              | 0              | 0              | 1              | 1              | 1              | 1              | 1               | 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              | 0               | 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              | 0               | 2                | MSE ← 0                      | Clear the interrupt master enable flag to zero.  |   |  |    |  |
|                                | EIH i4        | Enable interrupt high                    | 1                | 1              | 0              | 0              | 0              | 1              | 0              | 1              | I <sub>3</sub>  | I <sub>2</sub>   | I <sub>1</sub>               | I <sub>0</sub>   | $EDIH \leftarrow (EDIH) \vee i4$  | Set the interrupt enable flag to one.  |    |  |
|                                | EIL i4        | Enable interrupt low                     | 1                | 1              | 0              | 0              | 0              | 1              | 0              | 0              | I <sub>3</sub>  | I <sub>2</sub>   | I <sub>1</sub>               | I <sub>0</sub>   | $EDIL \leftarrow (EDIL) \vee i4$  | Set the interrupt enable flag to one.  |    |  |
|                                | DIH i4        | Disable interrupt high                   | 1                | 1              | 0              | 0              | 1              | 0              | 0              | 1              | I <sub>3</sub>  | I <sub>2</sub>   | I <sub>1</sub>               | I <sub>0</sub>   | $EDIH \leftarrow (EDIL) \wedge \bar{i4}$  | Clear the interrupt enable flag to zero.   | ZF |  |
|                                | DIL i4        | Disable interrupt low                    | 1                | 1              | 0              | 0              | 1              | 0              | 0              | 0              | I <sub>3</sub>  | I <sub>2</sub>   | I <sub>1</sub>               | I <sub>0</sub>   | $EDIL \leftarrow (EDIL) \wedge \bar{i4}$  | Clear the interrupt enable flag to zero.   | ZF |  |
|                                | WTSP          | Write SP                                 | 1                | 1              | 0              | 0              | 1              | 1              | 1              | 1              | 1               | 2                | SP ← (E), (AC)               | Transfer the contents of E, AC to SP.  |   |  |    |  |
|                                | RSP           | Read SP                                  | 1                | 1              | 0              | 0              | 1              | 1              | 1              | 1              | 1               | 2                | E, AC ← (SP)                 | Transfer the contents of SP to E, AC.  |   |  |    |  |
| Standby control instructions   | HALT          | HALT                                     | 1                | 1              | 0              | 0              | 1              | 1              | 1              | 1              | 1               | 2                | HALT                         | Enter halt mode.   |   |  |    |  |
|                                | HOLD          | HOLD                                     | 1                | 1              | 0              | 0              | 1              | 1              | 1              | 1              | 1               | 2                | HOLD                         | Enter HOLD mode.   |   |  |    |  |

Continued on next page.

LC66354B, 66356B, 66358B

Continued from preceding page.

| Instruction group               | 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> |                 |                  |                                      |   |                      |      |
| Serial I/O control instructions | STARTS   | Start serial I/O | 1                | 1              | 0              | 0              | 1              | 1              | 1              | 1              | 2               | 2                | START SIO                            | Start SIO operation.  |                      |      |
|                                 | WTSIO    | Write serial I/O | 1                | 1              | 0              | 0              | 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              | 2               | 2                | E, AC ← (SIO)                        | Read 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              | 1              | 1              | 2               | 2                | PC12 ← I <sub>1</sub> I <sub>0</sub> | Specify the memory bank.                                    |                      |      |

- 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.