

**SANYO**

No. 1965A

**LC7981****CONTROLLER FOR THE LC DOT MATRIX  
GRAPHIC DISPLAY****Summary**

The LC7981 is a controller LSI for the liquid crystal dot matrix graphic display. It stores display data sent from the 8-bit microcomputer in the display RAM attached externally and generates dot matrix LC drive signals.

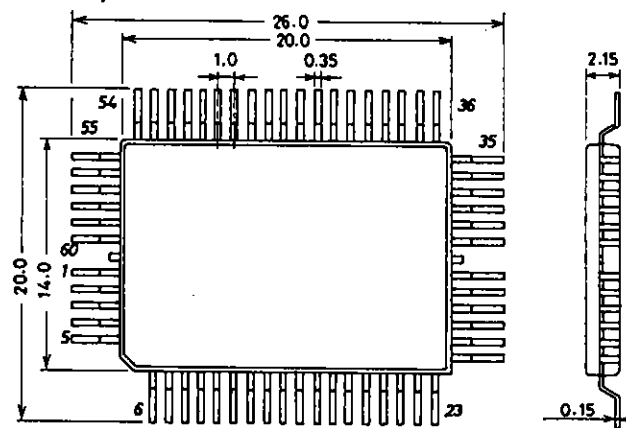
The LC7981 has two modes — the graphic mode, in which each bit of data from the external RAM either lights or does not light a dot in the LCD, and the character mode, in which character codes stored in the external RAM generate dot patterns through the built-in character-generator ROM. These two ways enable the LC7981 to cover a wide variety of applications.

As the LC7981 is fabricated using CMOS process technology, combining it with a CMOS microcomputer produces an LCD device of low power demand.

**Features**

1. Liquid crystal dot matrix graphic display controller
2. Display control capacity.  
Graphic mode ----- 512K dots ( $2^{16}$  bytes)  
Character mode ----- 4096 characters ( $2^{12}$  characters)
3. Character generator ROM ----- 7360 bits  
Character font 5 x 7 dots    160 types  
Character font 5 x 11 dots    32 types } Total 192 types  
(Extendable to 4K bytes with an external ROM)
4. Interfacing allowed with 8-bit MPU
5. Display duty (selectable by program)  
From static to 1/256 duty
6. A variety of instruction functions  
Scroll, cursor on/off/blink, character blink, bit manipulation
7. Display system ----- A system or B system selectable
8. Built-in oscillator (resistor, capacitor attached externally)
9. Low power demand
10. Single +5V power supply

**Package Dimensions 3055A-Q60C1C**  
(unit: mm)



**Absolute Maximum Ratings/ $T_a=25^{\circ}\text{C}$ , GND=0V**

|                             |                     |                              | unit               |
|-----------------------------|---------------------|------------------------------|--------------------|
| Maximum Supply Voltage      | $V_{DD\text{ max}}$ | -0.3to+7.0                   | V                  |
| Input Voltage               | $V_I$               | -0.3to $V_{DD}+0.3$          | V                  |
| Output Voltage              | $V_O$               | -0.3to $V_{DD}+0.3$          | V                  |
| Allowable Power Dissipation | $P_d\text{ max}$    | $T_a=25^{\circ}\text{C}$ 200 | mW                 |
| Operating Temperature       | $T_{opr}$           | -20to+75                     | $^{\circ}\text{C}$ |
| Storage Temperature         | $T_{stg}$           | -55to+125                    | $^{\circ}\text{C}$ |

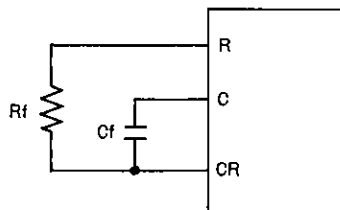
**Allowable Operating Conditions/ $T_a=-20$  to  $+75^{\circ}\text{C}$ , GND=0V**

|                                 |           |  | min          | typ | max          | unit |
|---------------------------------|-----------|--|--------------|-----|--------------|------|
| Supply Voltage                  | $V_{DD}$  |  | 4.75         |     | 5.25         | V    |
| Input "H"-Level Voltage         | $V_{IH1}$ | All input, I/O terminals except for $\overline{\text{SYNC}}$ , CR                          | 2.2          |     | $V_{DD}$     | V    |
| Input "L"-Level Voltage         | $V_{IL1}$ | "  | 0            |     | 0.8          | V    |
| Input "H"-Level Voltage         | $V_{IH2}$ | $\overline{\text{SYNC}}$ , CR  | 0.7 $V_{DD}$ |     | $V_{DD}$     | V    |
| Input "L"-Level Voltage         | $V_{IL2}$ | $\overline{\text{SYNC}}$ , CR  | 0            |     | 0.3 $V_{DD}$ | V    |
| Output "H"-Level Voltage        | $V_{OH1}$ | $I_{OH}=-0.6\text{mA}$<br>DB0to7, $\overline{\text{WE}}$ , MA0to15, MD0to7                 | 2.4          |     | $V_{DD}$     | V    |
| Output "L"-Level Voltage        | $V_{OL1}$ | $I_{OL}=1.6\text{mA}$<br>DB0to7, $\overline{\text{WE}}$ , MA0to15, MD0to7                  | 0            |     | 0.4          | V    |
| Output "H"-Level Voltage        | $V_{OH2}$ | $I_{OH}=-0.6\text{mA}$<br>$\overline{\text{SYNC}}$ , CPO, FLM, CL1, CL2,<br>D1, D2, MA, MB | $V_{DD}-0.4$ |     | $V_{DD}$     | V    |
| Output "L"-Level Voltage        | $V_{OL2}$ | $I_{OL}=0.6\text{mA}$<br>$\overline{\text{SYNC}}$ , CPO, FLM, CL1, CL2,<br>D1, D2, MA, MB  | 0            |     | 0.4          | V    |
| <b>Internal Clock Operation</b> |           |  |              |     |              |      |
| Clock OSC Frequency             | $f_{OSC}$ | $C_f=15\text{pF} \pm 5\%$ , $R_f=39\text{k}\Omega \pm 2\%$ , <b>Note 1</b>                 | 500          | 600 | 700          | kHz  |
| <b>External Clock Operation</b> |           |  |              |     |              |      |
| Clock Frequency                 | $f_{CP}$  | <b>Note 2</b>  |              |     | 2.5          | MHz  |
| Clock Duty                      | Duty      | <b>Note 3</b>  | 47.5         | 50  | 52.5         | %    |
| Clock Rise Time                 | $t_{rcp}$ | <b>Note 3</b>  |              |     | 50           | ns   |
| Clock Fall Time                 | $t_{fcp}$ | <b>Note 3</b>  |              |     | 50           | ns   |

**Electrical Characteristics/ $T_a=-20$  to  $+75^{\circ}\text{C}$ , GND=0V,  $V_{DD}=5\text{V} \pm 5\%$** 

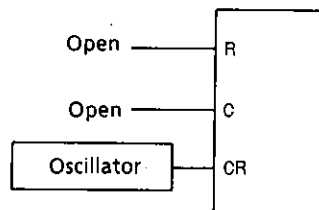
|                     |           |   | min | typ | max | unit          |
|---------------------|-----------|---|-----|-----|-----|---------------|
| Input Leak Current  | $I_{IN}$  | $V_{IN}=0$ to $V_{DD}$ , $\overline{\text{CS}}$ , E, RS, R/W, $\overline{\text{RES}}$ | -5  |     | 5   | $\mu\text{A}$ |
| Current Dissipation | $I_{CC1}$ | CR oscillation, $f_{OSC}=600\text{kHz}$   |     | 2   | 4   | mA            |
| Current Dissipation | $I_{CC2}$ | External clock, $f_{CP}=2.5\text{MHz}$  |     | 3   | 5   | mA            |
| Pull-up Current     | $I_{PL}$  | $V_{IN}=\text{GND}$ , DB0to7, RD0to7, MD0to7  |     | 10  | 20  | $\mu\text{A}$ |

(Note 1)

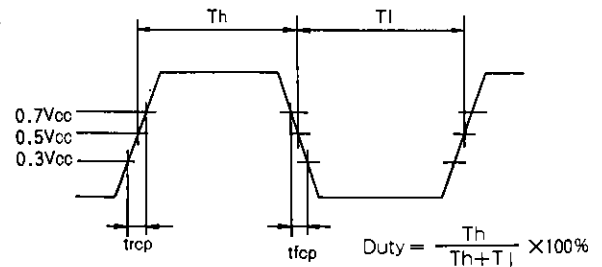


$C_f=15\text{pF} \pm 5\%$   
 $R_f=39\text{k}\Omega \pm 2\%$   
 (at  $f_{OSC}=600\text{kHz typ.}$ )

(Note 2)

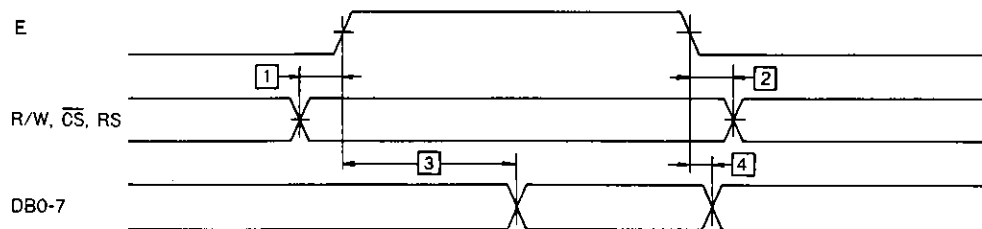
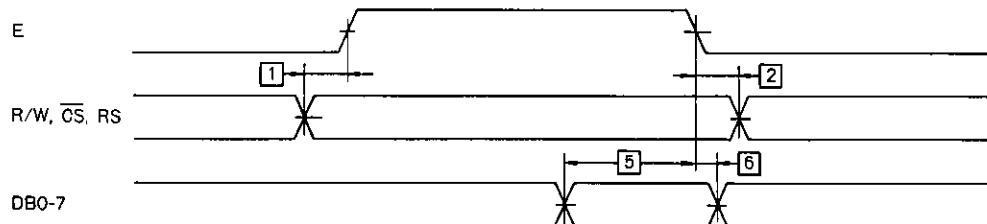


(Note 3)



## Timing Characteristics

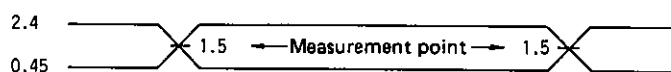
## ● Bus read/write operation 1

READ CYCLEWRITE CYCLE

$T_a = -20$  to  $+75^{\circ}\text{C}$ ,  $V_{DD} = 5\text{ V} \pm 5\%$ ,  $GND = 0\text{ V}$

| No | Item                     | Symbol | min | typ | max | unit | Conditions          |
|----|--------------------------|--------|-----|-----|-----|------|---------------------|
| 1  | Address set-up time      | tAS    | 90  |     |     | ns   |                     |
| 2  | Address hold time        | tAH    | 10  |     |     | ns   |                     |
| 3  | Data delay time (read)   | tDDR   |     |     | 140 | ns   | $C_L = 50\text{pF}$ |
| 4  | Data hold time (read)    | tDHR   | 10  |     |     | ns   |                     |
| 5  | Data set-up time (write) | tDSW   | 220 |     |     | ns   |                     |
| 6  | Data hold time (write)   | tDHW   | 20  |     |     | ns   |                     |

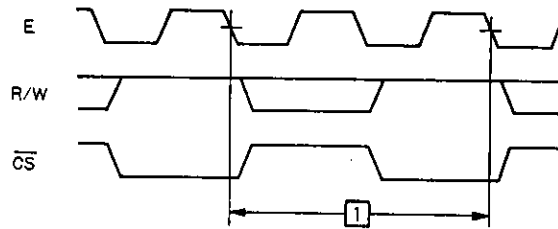
Note: Definition of the test waveform



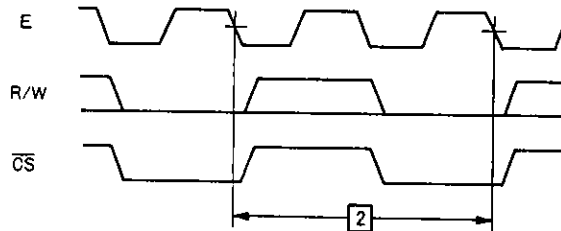
The input terminals are driven at 2.4V and 0.45V. Timing is measured at 1.5V.

• Bus read/write operation 2

DATA READ CYCLE



DATA WRITE CYCLE



$T_a = -20$  to  $+75^{\circ}\text{C}$ ,  $V_{DD} = 5\text{ V} \pm 5\%$ ,  $\text{GND} = 0\text{ V}$

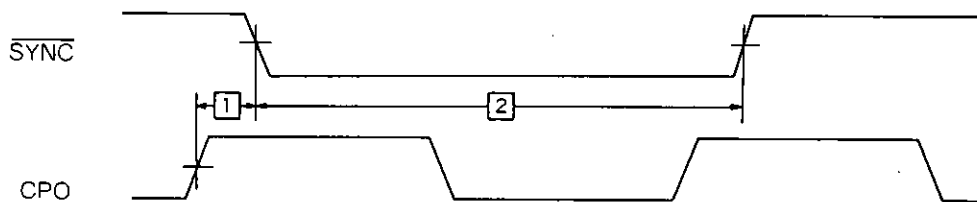
| No | Item             | Symbol     | min | typ | max   | unit | Instruction register value                        |
|----|------------------|------------|-----|-----|---|------|---|
| 1  | Read cycle time  | $t_{RCY}$  |     |     | $\frac{(H_P + 2) \times 10^3}{F_{OSC}} + 200$   | ns   | 0 DH  |
| 2  | Write cycle time | $t_{WCY1}$ |     |     | $\frac{(2 H_P + 2) \times 10^3}{F_{OSC}} + 200$ | ns   | 0 EH, 0 FH  |
| 2  | Write cycle time | $t_{WCY2}$ |     |     | $\frac{(H_P + 2) \times 10^3}{F_{OSC}} + 200$   | ns   | 0 CH  |
| 2  | Write cycle time | $t_{WCY3}$ |     |     | $\frac{2000}{F_{OSC}} + 200$                    | ns   | 00H, 01H, 02H, 03H<br>04H, 08H, 09H<br>0 AH, 0 BH |

Notes: (1) In the character mode,  $H_P$  is the number of horizontal dots per character in a character display. In the graphic mode,  $H_P$  indicates how many bits from RAM appear in a 1-byte display.

(2)  $F_{OSC}$  is the oscillating frequency, expressed in MHz.

(3) All measurement points are at 1.5V.

• Parallel operation (at master mode)



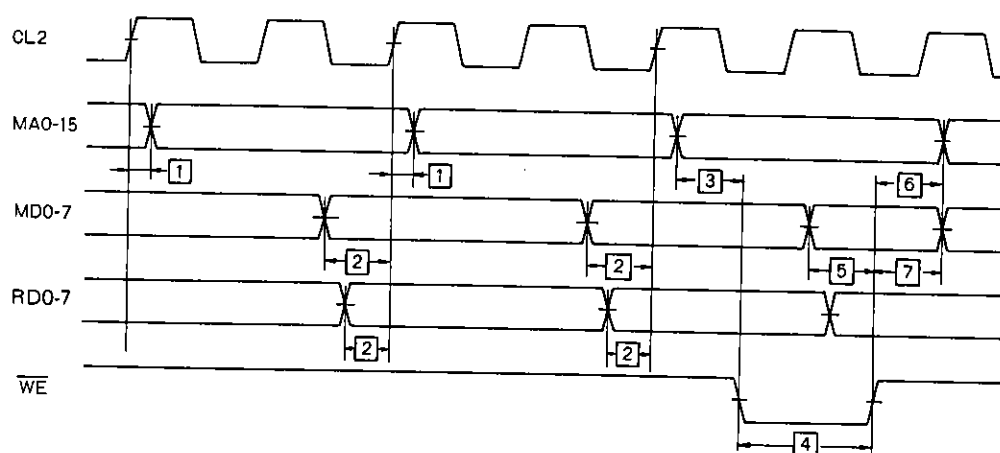
$T_a = -20$  to  $+75^{\circ}\text{C}$ ,  $V_{DD} = 5\text{ V} \pm 5\%$ ,  $\text{GND} = 0\text{ V}$

| No | Item             | Symbol    | min | typ | max | unit | Conditions |
|----|------------------|-----------|-----|-----|-----|------|------------|
| 1  | SYNC delay time  | $t_{DSY}$ |     |     | 100 | ns   |            |
| 2  | SYNC pulse width | $t_{WSY}$ | 350 |     |     | ns   |            |

Notes: (1) All output terminals are under no load.

(2) All measurement points are at  $0.5V_{DD}$ .

● Interface with external RAM and ROM



READ CYCLE  $T_a = -20^{\circ}\text{C to } +75^{\circ}\text{C}$ ,  $V_{DD} = 5\text{ V} \pm 5\%$ ,  $GND = 0\text{ V}$

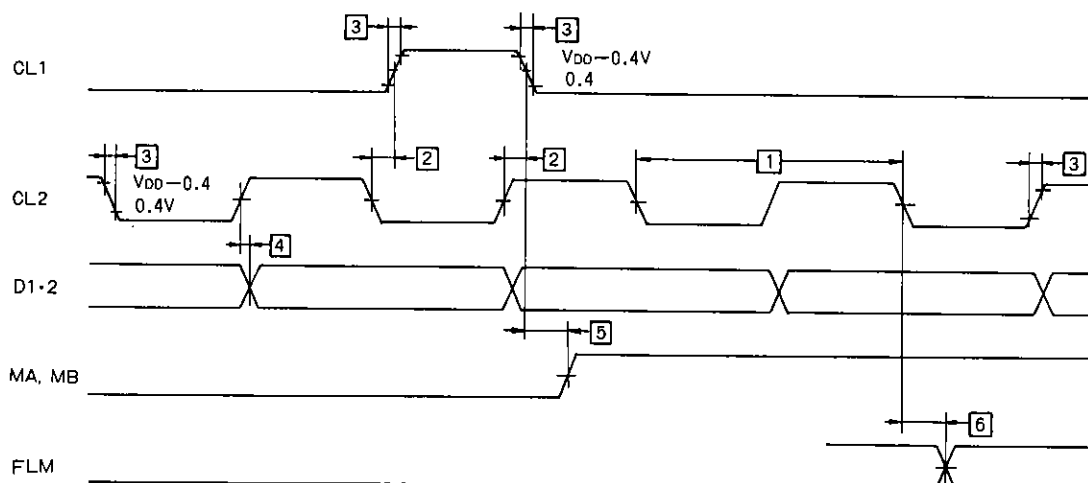
| No | Item                           | Symbol | min | typ | max | unit | Conditions |
|----|--------------------------------|--------|-----|-----|-----|------|------------|
| 1  | MA0-15 read address delay time | tDMAR  |     |     | 95  | ns   |            |
| 2  | MD0-7 · RD0-7 set-up time      | tSMDR  | 105 |     |     | ns   |            |

WRITE CYCLE  $T_a = -20^{\circ}\text{C to } +75^{\circ}\text{C}$ ,  $V_{DD} = 5\text{ V} \pm 5\%$ ,  $GND = 0\text{ V}$

| No | Item                               | Symbol | min | typ | max | unit | Conditions |
|----|------------------------------------|--------|-----|-----|-----|------|------------|
| 3  | Memory address set-up time         | tSMAW  | 50  |     |     | ns   |            |
| 4  | $\overline{\text{WE}}$ pulse width | tWWE   | 350 |     |     | ns   |            |
| 5  | Memory data set-up time            | tSMDW  | 250 |     |     | ns   |            |
| 6  | Memory address hold time           | tHMAW  | 50  |     |     | ns   |            |
| 7  | Memory data hold time              | tHMDW  | 50  |     |     | ns   |            |

Notes: (1) All output terminals are under no load.  
(2) All measurement points are at 1.5V.

● Interface with the driver LSI



$T_a = -20$  to  $+75^{\circ}\text{C}$ ,  $V_{DD} = 5\text{ V} \pm 5\%$ ,  $\text{GND} = 0\text{ V}$

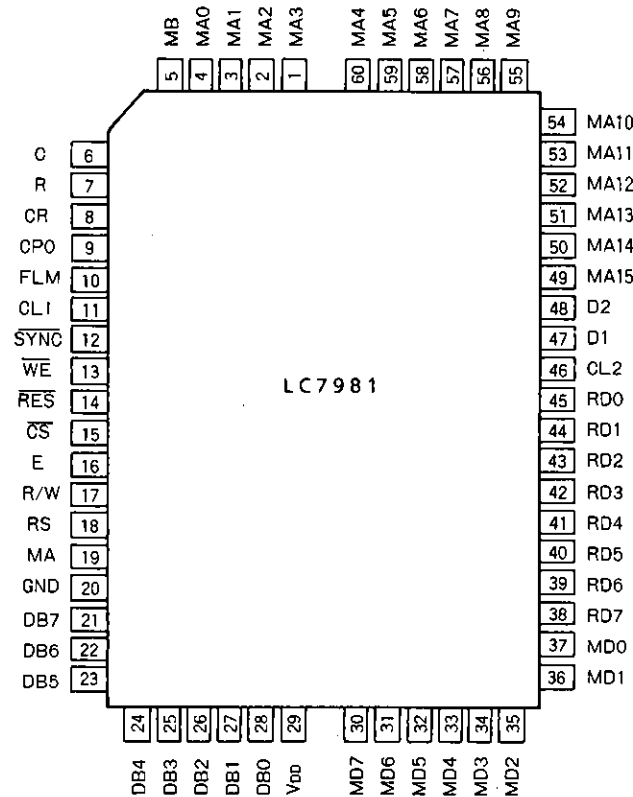
| No | Item                    | Symbol | min | typ | max | unit | Conditions |
|----|-------------------------|--------|-----|-----|-----|------|------------|
| 1  | Clock cycle time        | tcyc   | 400 |     |     | ns   |            |
| 2  | Clock phase difference  | tdcl   |     |     | 100 | ns   |            |
| 3  | Clock rise/fall time    | tcrrf  |     |     | 30  | ns   |            |
| 4  | D1-2 phase difference   | tdd    |     |     | 100 | ns   |            |
| 5  | MA, MB phase difference | tdma   |     |     | 200 | ns   |            |
| 6  | FLM phase difference    | tdfm   |     |     | 200 | ns   |            |

Notes: (1) All output terminals are under no load.

(2) All measurement points other than those specified are at  $0.5V_{DD}$ .

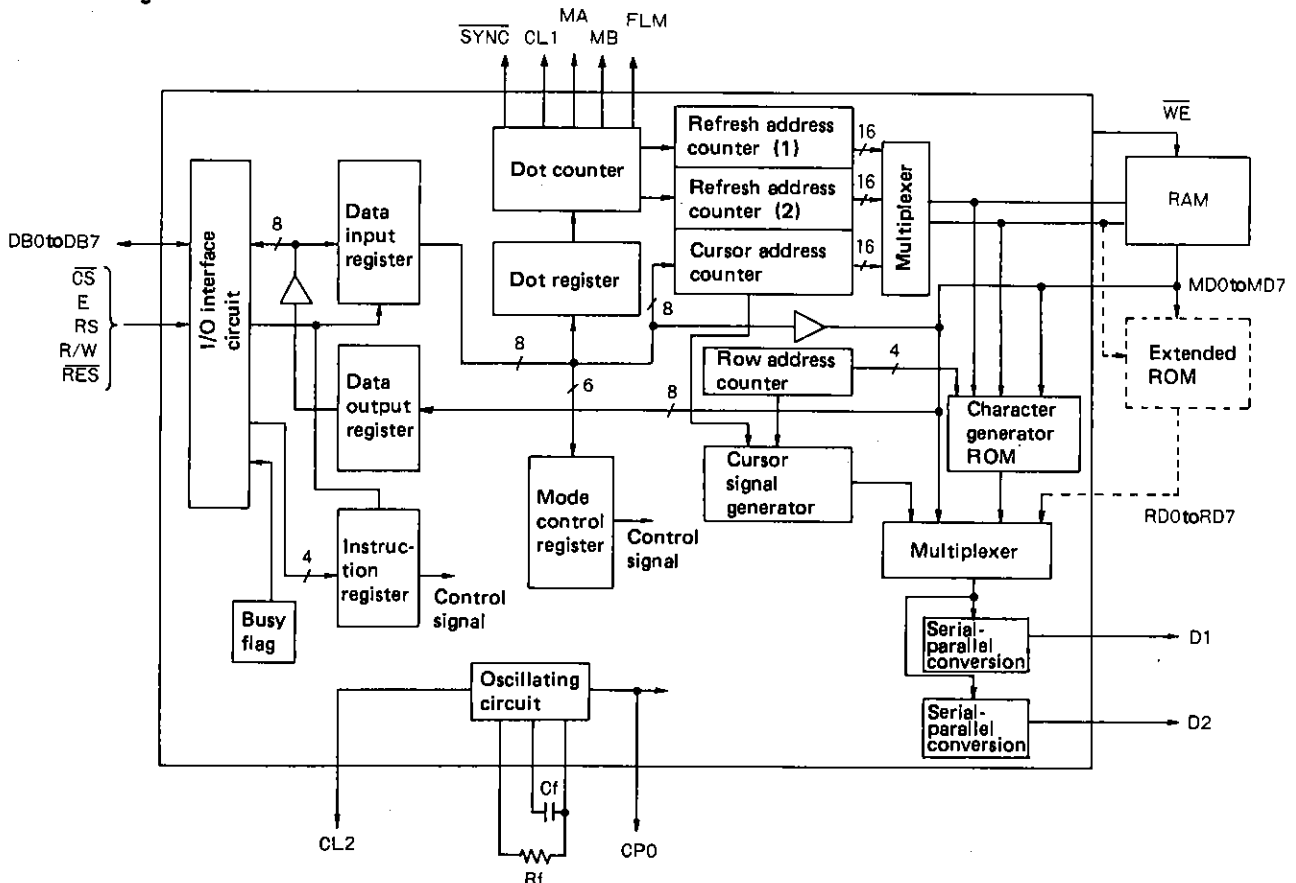
# LC7981

## Pin Assignment



(Top view)

## Block Diagram



\* When the ROM is extended, MA0 to MA11 are used for RAM addresses and MA12 to MA15 are used for the extended ROM.

**Function of each block**

- Register

The LC7981 has 5 types of registers — the instruction register, data input register, data output register, dot register, and mode control register.

The instruction register stores such instruction codes as the start address, cursor address specification, etc. It consists of 4 bits, and the lower 4 bits of the data bus, DB0 to DB3, are written into it.

The data input register temporarily stores data to be written into the external RAM, dot register, and mode control register. It consists of 8 bits.

The data output register temporarily stores data to be read from external RAM, and consists of 8 bits. When the cursor address is written into the cursor address counter via the data input register and the memory read instruction is set in the instruction register, data in external RAM is read into the data output register by internal operation. With the next instruction, the MPU reads the data output register, and completes data transfer to the MPU.

The dot register stores dot information such as the character pitch, the number of vertical dots, etc. Data sent from the MPU is written into the dot register via the data input register.

The mode control register stores LCD status information such as display on/off and cursor on/off/blink. It consists of 6 bits. Data sent from the MPU is written into this register via the data input register.

- Busy flag

When the busy flag is "1", the LC7981 is operating internally. At this time, the next instruction cannot be accepted. The busy flag is output to DB7 when RS=1, RW=1. The next instruction must be written after ensuring that the busy flag is "0". When the maximum value of the read cycle time or write cycle time has been passed after the execution of the preceding data read instruction or data write instruction, the next instruction can be executed without checking the busy flag.

- Dot counter

The dot counter generates LC display timing according to the contents of the dot register.

- Refresh address counter

The refresh address counter controls addresses of the external RAM, character generator ROM, extended ROM and is available in two types — refresh address counter (1) and refresh address counter (2). The former is for the upper screen, and the latter for the lower screen. In the graphic mode, 16-bit data is output and used as the address signal for the external RAM. In the character mode, the high-order 4 bits are ignored and 4 bits of the row address counter are output instead. The 4 bits are used as the address of the extended ROM.

- Character generator ROM

The character generator ROM has a total of 7360 bits and stores data on 192 kinds of characters. Character codes from the external RAM and row codes from the row address counter are added to address signals, and ROM outputs 5-bit dot data.

There are 192 kinds of character fonts, of which 160 are 5 x 7 and 32 are 5 x 11. With extended ROM, character fonts can be increased to 256 kinds sized 8 x 16.

- Cursor address counter

The cursor address counter is a 16-bit counter which can be preset by instruction. When data is read from or written into external RAM (i.e., read/write of display dot data or character codes), the counter retains the addresses. The value indicated on the cursor address counter is automatically incremented by 1 when instructions to read/write display data and to perform bit set/clear are issued.

- Cursor signal generator

In the character mode, the cursor can be displayed by means of instructions. The cursor is generated automatically when the cursor address counter and the row address counter reach the specified value.

- Parallel-serial conversion

The two parallel-serial conversion circuits simultaneously transfer parallel data from the external RAM, character generator, and extended ROM to the upper and lower LC screen drive circuits as serial data.



# LC7981

## Pin Function

| Description       | Pin No.            | Function  |
|-------------------|--------------------|---|
| DB0 to 7          | 21 to 28           | Data bus ----- Three-state I/O common terminal, terminal for transmitting/receiving data to/from the MPU.   |
| $\overline{CS}$   | 15                 | Chip select ----- Selection allowed when $\overline{CS}=0$  |
| R/W               | 17                 | Read/write ----- R/W=1 ----- MPU $\leftarrow$ LC7981<br>R/W=0 ----- MPU $\rightarrow$ LC7981  |
| RS                | 18                 | Register select ----- RS=1 ----- instruction register<br>RS=0 ----- data register   |
| E                 | 16                 | Enable ----- Data is written on the negative transition of E.<br>Data can be read while E=1.  |
| CR, R, C          | 6, 7, 8            | Terminals for the CR oscillator   |
| $\overline{RES}$  | 14                 | Reset ----- Setting RES to 0 selects display OFF, slave mode, and Hp=6.   |
| MA0 to 15         | 1 to 4<br>49 to 60 | Address output for the display RAM. For character display, raster address for external CG is output at MA12 to 15.  |
| MD0 to 7          | 30 to 37           | Display data bus ----- Three-state I/O common terminals.  |
| RD0 to 7          | 38 to 45           | ROM data input ----- Dot data from the external character generator is input.   |
| $\overline{WE}$   | 13                 | Write enable ----- Display RAM write signal.  |
| CL2               | 46                 | Display data shift clock signal.  |
| CL1               | 11                 | Display data latch signal.  |
| FLM               | 10                 | Frame signal.   |
| MA                | 19                 | LC drive signal ----- AC signal ----- A system  |
| MB                | 5                  | LC drive signal ----- AC signal ----- B system  |
| D1, D2            | 47, 48             | Display data serial output ----- D1----- for the upper screen<br>D2 ----- for the lower screen  |
| CPO               | 9                  | Clock for slave   |
| $\overline{SYNC}$ | 12                 | Sync signal for parallel operation ----- Three-state input/output common terminal<br>Master mode ----- Sync signal is output.<br>Slave mode ----- Sync signal is input. |

- **Display control instruction**

Display is controlled by writing data into the instruction register and 13 data registers. The instruction register and the data register are distinguished by the RS signal. First, write 4-bit data in the instruction register when RS=1, then specify the code of the data register. Next, with RS=0, write 8-bit data in the data register, which executes the specified instruction.

A new instruction cannot be accepted while an old instruction is being executed. As the Busy flag is set under this condition, write an instruction only after reading the Busy flag and making sure that it is 0.

However, the next instruction can be executed without checking the busy flag when the maximum read cycle time or the write cycle time has been exceeded after execution of the previous data read instruction or the data write instruction. The busy flag does not change when data is written into the instruction register (RS=1). Therefore, the busy flag need not be checked immediately after writing data into the instruction register.

### 1) Mode control

Write code "00H" (in hexadecimal notation) in the instruction register and specify the mode control register.

| Register              | R/W | RS | DB7 | DB6 | DB5       | DB4 | DB3 | DB2 | DB1 | DB0 |
|-----------------------|-----|----|-----|-----|-----------|-----|-----|-----|-----|-----|
| Instruction register  | 0   | 1  | 0   | 0   | 0         | 0   | 0   | 0   | 0   | 0   |
| Mode control register | 0   | 0  | 0   | 0   | MODE Data |     |     |     |     |     |

| DB5         | DB4         | DB3            | DB2          | DB1   | DB0    | Cursor/blink               | CG                          | Graphic/character display |
|-------------|-------------|----------------|--------------|-------|--------|----------------------------|-----------------------------|---------------------------|
| 1<br>/<br>0 | 1<br>/<br>0 | 0              | 0            | 0     | 0      | Cursor OFF                 | Built-in CG                 | Character display         |
|             |             | 0              | 1            |       |        | Cursor ON                  |                             |                           |
|             |             | 1              | 0            |       |        | Cursor OFF character blink |                             |                           |
|             |             | 1              | 1            |       |        | Cursor blink               |                             |                           |
|             |             | 0              | 0            |       | 1      | Cursor OFF                 | External CG                 |                           |
|             |             | 0              | 1            |       |        | Cursor ON                  |                             |                           |
|             |             | 1              | 0            |       |        | Cursor OFF character blink |                             |                           |
|             |             | 1              | 1            |       |        | Cursor blink               |                             |                           |
|             |             | 0              | 0            | 1     | 0      |                            |                             | Graphic mode              |
|             |             | Display ON/OFF | Master/slave | Blink | Cursor | Mode                       | External/<br>built-in<br>CG |                           |

1 : master mode  
0 : slave mode

1 : display ON  
0 : display OFF

## 2) Setting the character pitch

| Register                 | R/W | RS | DB7           | DB6 | DB5 | DB4 | DB3 | DB2           | DB1 | DB0 |
|--------------------------|-----|----|---------------|-----|-----|-----|-----|---------------|-----|-----|
| Instruction register     | 0   | 1  | 0             | 0   | 0   | 0   | 0   | 0             | 0   | 1   |
| Character pitch register | 0   | 0  | (Vp-1) Binary |     |     |     | 0   | (Hp-1) Binary |     |     |

Vp is the number of vertical dots per character. Determine Vp with the pitch between two vertically placed characters taken into consideration. This value is meaningful only in the character display mode: It is invalid in the graphic mode.

In character mode, Hp indicates the number of horizontal dots per character, from the leftmost part of one character to the leftmost part of the next. In the graphic mode, Hp indicates how many bits (or dots) from RAM appear in a 1-byte display.

Hp must take one of the following three values.

| HP | DB2 | DB1 | DB0 |                              |
|----|-----|-----|-----|------------------------------|
| 6  | 1   | 0   | 1   | Horizontal character pitch 6 |
| 7  | 1   | 1   | 0   | 7                            |
| 8  | 1   | 1   | 1   | 8                            |

## 3) Setting the number of characters

| Register                  | R/W | RS | DB7           | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---------------------------|-----|----|---------------|-----|-----|-----|-----|-----|-----|-----|
| Instruction register      | 0   | 1  | 0             | 0   | 0   | 0   | 0   | 0   | 1   | 0   |
| Character number register | 0   | 0  | (HN-1) Binary |     |     |     |     |     |     |     |

In the character display mode, HN indicates the number of characters in the horizontal direction. In the graphic mode, it indicates the number of bytes in the horizontal direction. The total number of dots positioned horizontally on the screen n is given by the formula

$$n = H_p \times H_N$$

Even numbers in the range 2 to 256 (decimal) can be set as HN.

## 4) Setting the time division number (display duty)

| Register               | R/W | RS | DB7           | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|------------------------|-----|----|---------------|-----|-----|-----|-----|-----|-----|-----|
| Instruction register   | 0   | 1  | 0             | 0   | 0   | 0   | 0   | 0   | 1   | 1   |
| Time division register | 0   | 0  | (Nx-1) Binary |     |     |     |     |     |     |     |

Consequently, 1/Nx is the display duty.

Decimal numbers within the range 1 to 256 can be set as Nx.

## 5) Setting the cursor position

| Register                 | R/W | RS | DB7 | DB6 | DB5 | DB4 | DB3           | DB2 | DB1 | DB0 |
|--------------------------|-----|----|-----|-----|-----|-----|---------------|-----|-----|-----|
| Instruction register     | 0   | 1  | 0   | 0   | 0   | 0   | 0             | 1   | 0   | 0   |
| Cursor position register | 0   | 0  | 0   | 0   | 0   | 0   | (CP-1) Binary |     |     |     |

In the character display mode, Cp indicates the line at which the cursor is displayed. For example, when Cp=8 (decimal) is specified, the cursor is displayed beneath the character of the 5 x 7 dot-font. The horizontal length of the cursor equals Hp (the horizontal character pitch). Decimal values in the range 1 to 16 can be assigned to Cp. When the value is less than the vertical character pitch Vp ( $C_p \leq V_p$ ), display priority is given to the cursor (provided the cursor display is ON). The cursor is not displayed when  $C_p < V_p$ . The horizontal length of the cursor equals Hp.

## 6) Setting the display start lower address

| Register                                    | R/W | RS | DB7                               | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|-----|----|-----------------------------------|-----|-----|-----|-----|-----|-----|-----|
| Instruction register                        | 0   | 1  | 0                                 | 0   | 0   | 0   | 1   | 0   | 0   | 0   |
| Display start address register (lower byte) | 0   | 0  | (start address lower byte) binary |     |     |     |     |     |     |     |

## 7) Setting the display start upper address

| Register                                    | R/W | RS | DB7                               | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|-----|----|-----------------------------------|-----|-----|-----|-----|-----|-----|-----|
| Instruction register                        | 0   | 1  | 0                                 | 0   | 0   | 0   | 1   | 0   | 0   | 1   |
| Display start address register (upper byte) | 0   | 0  | (start address upper byte) binary |     |     |     |     |     |     |     |

This instruction writes the display start address value in the display start address register. The display start address is the RAM address at which data to be displayed at the leftmost position of the top line of the screen is stored. The start address consists of 16 bits (upper and lower).

## 8) Setting the cursor (lower) address (RAM read/write lower address)

| Register                            | R/W | RS | DB7                                | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|-------------------------------------|-----|----|------------------------------------|-----|-----|-----|-----|-----|-----|-----|
| Instruction register                | 0   | 1  | 0                                  | 0   | 0   | 0   | 1   | 0   | 1   | 0   |
| Cursor address counter (lower byte) | 0   | 0  | (cursor address lower byte) binary |     |     |     |     |     |     |     |

## 9) Setting the cursor (upper) address (RAM read/write upper address)

| Register                            | R/W | RS | DB7                                | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|-------------------------------------|-----|----|------------------------------------|-----|-----|-----|-----|-----|-----|-----|
| Instruction register                | 0   | 1  | 0                                  | 0   | 0   | 0   | 1   | 0   | 1   | 1   |
| Cursor address counter (upper byte) | 0   | 0  | (cursor address upper byte) binary |     |     |     |     |     |     |     |

This instruction writes the cursor address value in the cursor address counter. The cursor address indicates the address for exchanging display data and character codes with RAM. In other words, data at the address specified by the cursor address is read from or written into RAM. In character display, the cursor is displayed at the position specified by the cursor address.

The cursor address is divided into a lower address (8 bits) and an upper address (8 bits). It should be set in accordance with the following rules.

|   |  |  |
|---|--|--|
| 1 | To rewrite (set) both lower and upper addresses: | First set the lower address, then the upper.                       |
| 2 | To rewrite the lower address:                    | Always reset the upper address after setting the lower address.    |
| 3 | To rewrite the upper address only:               | Set the upper address. It is necessary to reset the lower address. |

The cursor address counter is a 16-bit up-counter with set/reset functions: when the Nth bit goes from 1 to 0, the count of the (N + 1)th bit increments by one. Accordingly, when the lower address is set so that the lower MSB (8th bit) changes from 1 to 0, the LSB (1st bit) of the upper counter must increment by one. When setting the cursor address, set the lower and upper addresses as a 2-byte continuous instruction.

## 10) Writing display data

| Register             | R/W | RS | DB7                                    | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|----------------------|-----|----|--|-----|-----|-----|-----|-----|-----|-----|
| Instruction register | 0   | 1  | 0                                      | 0   | 0   | 0   | 1   | 1   | 0   | 0   |
| RAM                  | 0   | 0  | MSB (pattern data, character code) LSB |     |     |     |     |     |     |     |

Write code "0CH" in the instruction register. Then, write 8-bit data with RS=0, and the data is written into RAM as display data or character codes at the address specified by the cursor address counter. After writing, the count of the cursor address counter increments by 1.

## 11) Reading display data

| Register             | R/W | RS | DB7                                    | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|----------------------|-----|----|--|-----|-----|-----|-----|-----|-----|-----|
| Instruction register | 0   | 1  | 0                                      | 0   | 0   | 0   | 1   | 1   | 0   | 1   |
| RAM                  | 1   | 0  | MSB (pattern data, character code) LSB |     |     |     |     |     |     |     |

Write "0DH" in the instruction register. Then, establish the read status with RS=0, and data in the RAM can be read. The procedure for reading data is as follows:

This instruction outputs the contents of the data output register to DB0 to 7, then transfers the RAM data indicated by the cursor address to the data output register. It then increments the cursor address by 1, which means that correct data cannot be read in the first read operation. The specified value is output in the second read operation. Accordingly, a dummy read operation must be performed once when reading data after setting the cursor address.

## 12) Bit clear

| Register             | R/W | RS | DB7 | DB6 | DB5 | DB4 | DB3 | DB2           | DB1 | DB0 |
|----------------------|-----|----|-----|-----|-----|-----|-----|---------------|-----|-----|
| Instruction register | 0   | 1  | 0   | 0   | 0   | 0   | 1   | 1             | 1   | 0   |
| Bit clear            | 0   | 0  | 0   | 0   | 0   | 0   | 0   | (NB-1) Binary |     |     |

## 13) Bit set

| Register             | R/W | RS | DB7 | DB6 | DB5 | DB4 | DB3 | DB2           | DB1 | DB0 |
|----------------------|-----|----|-----|-----|-----|-----|-----|---------------|-----|-----|
| Instruction register | 0   | 1  | 0   | 0   | 0   | 0   | 1   | 1             | 1   | 1   |
| Bit set              | 0   | 0  | 0   | 0   | 0   | 0   | 0   | (NB-1) Binary |     |     |

As the bit-clear or bit-set instruction, 1 bit of a 1 byte of data in display RAM is set to 0 or 1. The bit specified by NB is set to 0 for the bit-clear instruction and 1 for the bit-set instruction. The RAM address is specified by the cursor address, which is automatically incremented by 1 at the completion of the instruction. NB is a value in the range from 1 to 8. The LSB is indicated by Ng=1, and the MSB by Ng=8.

## 14) Reading the BUSY flag

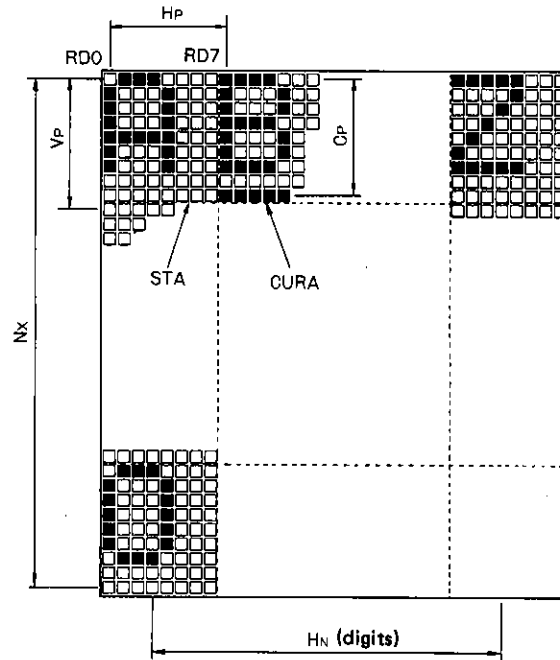
| Register  | R/W | RS | DB7   | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|-----------|-----|----|-------|-----|-----|-----|-----|-----|-----|-----|
| Busy flag | 1   | 1  | 1 / 0 | *   |     |     |     |     |     |     |

The busy flag is output to DB7 when read mode is established with RS=1. The busy flag is set to 1 while any of the instructions 1) through 13) is being executed. It is set to 0 at the completion of the execution, allowing the next instruction to be accepted. No other instruction can be accepted when the busy flag is 1. Accordingly, before writing an instruction and data, it is necessary to ensure that the busy flag is 0. However, the next instruction can be executed without checking the busy flag when the maximum read cycle time or the write cycle time has been exceeded after execution of the previous data read instruction or the data write instruction.

The busy flag does not change when data is written into the instruction register (RS=1). Therefore, the busy flag need not be checked immediately after writing data into the instruction register.

Specification of the instruction register is unnecessary to read the busy flag.

The relation between the LCD panel display and  $H_p$ ,  $H_N$ ,  $V_p$ ,  $C_p$ , and  $N_x$ .



| Symbol | Description                                      | Contents  | Value                             |
|--------|--|---|-----------------------------------|
| $H_p$  | Horizontal character pitch                       | Character pitch in the horizontal direction   | 6 to 8 dots                       |
| $H_N$  | Number of characters in the horizontal direction | Number of characters (digits) per horizontal line or the number of words per line (graphic) | Even digits in the range 2 to 256 |
| $V_p$  | Vertical character pitch                         | Character pitch in the vertical direction   | 1 to 16 dots                      |
| $C_p$  | Cursor position                                  | The line number at which the cursor is to be displayed                                      | 1 to 16 lines                     |
| $N_x$  | Number of lines in the vertical direction        | Display duty  | 1 to 256 lines                    |

Note)

When the number of vertical dots on the screen is  $m$  and that of horizontal dots is  $n$ ,

$$1/m = 1/N_x = \text{display duty}$$

$$n = H_p \times H_N$$

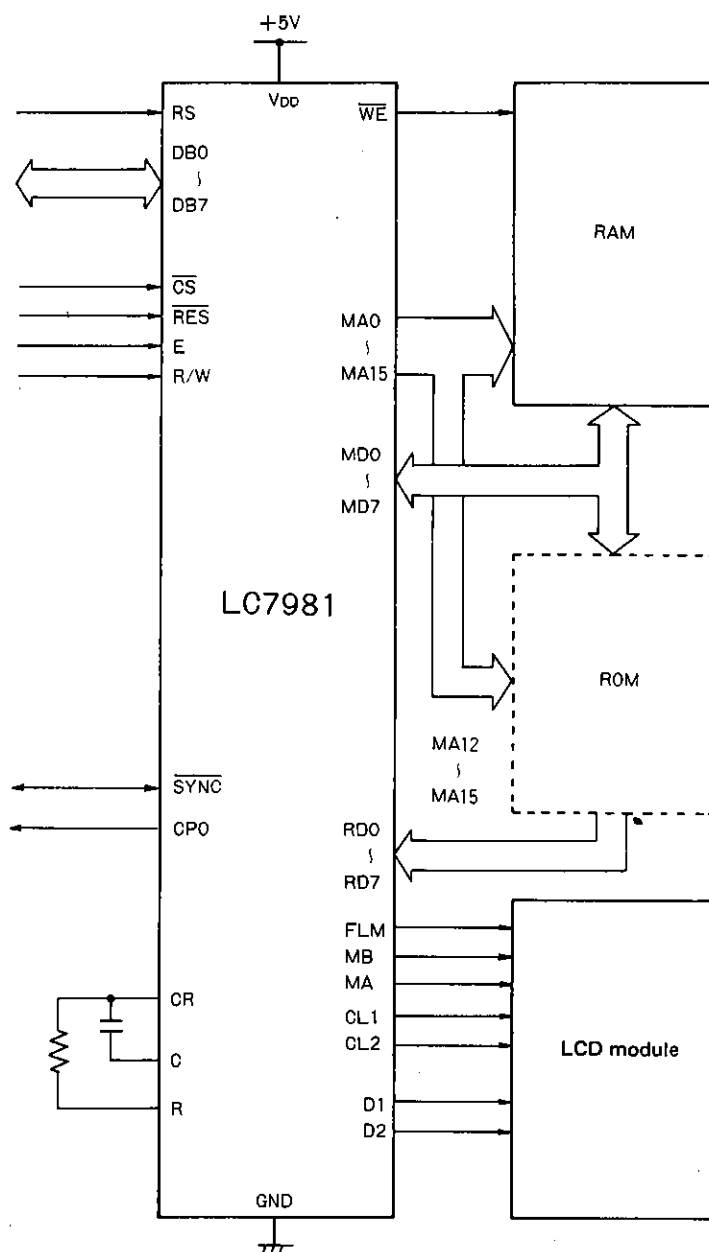
$$m/V_p = \text{number of display lines}$$

$$C_p \leq V_p$$

Display mode

| Display mode      | Display data from the MPU | RAM                  | LCD panel                  |
|-------------------|---------------------------|----------------------|----------------------------|
| Character display | Display pattern (8 bits)  | <p>Start address</p> | <p>Hp : 6, 7 or 8 dots</p> |
| Graphic           | Character code (8 bits)   | <p>Start address</p> | <p>Hp : 8 dots</p>         |

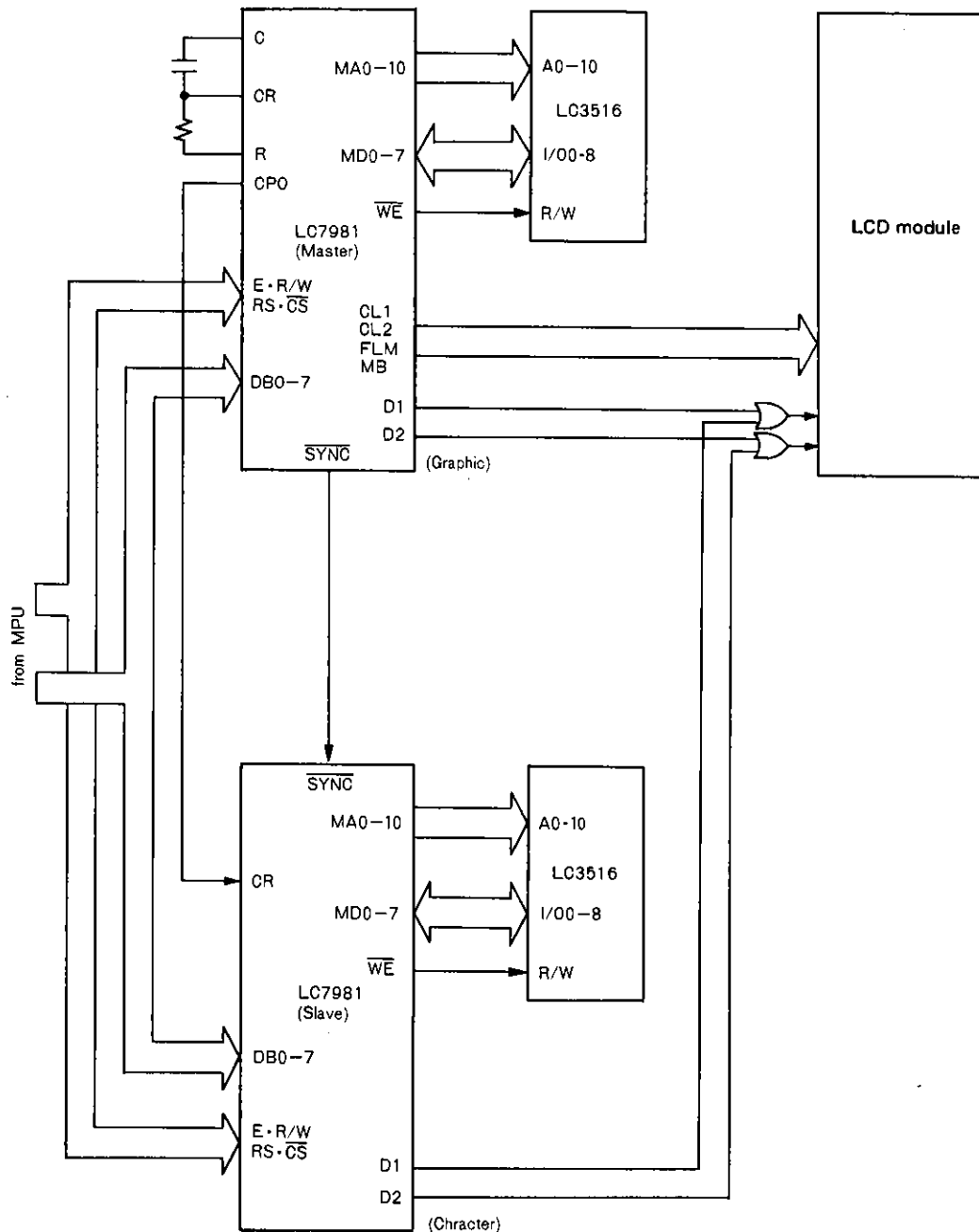
Sample application circuit (1)



- 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.

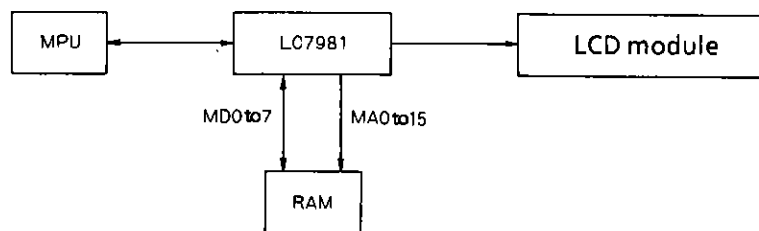


Sample application circuit (2) [Composition of graphic display and character display]

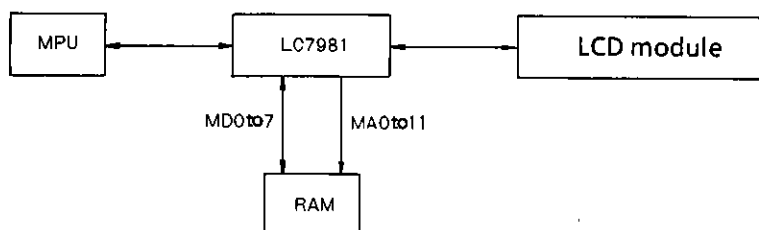


## Examples of configuration

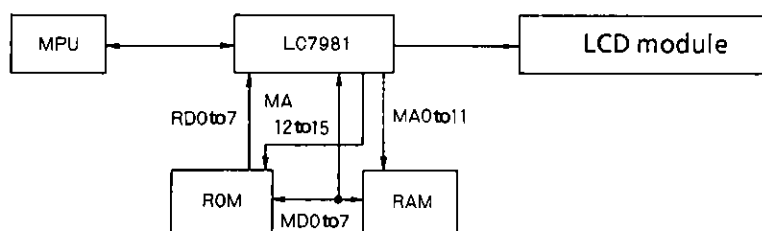
- Graphic mode



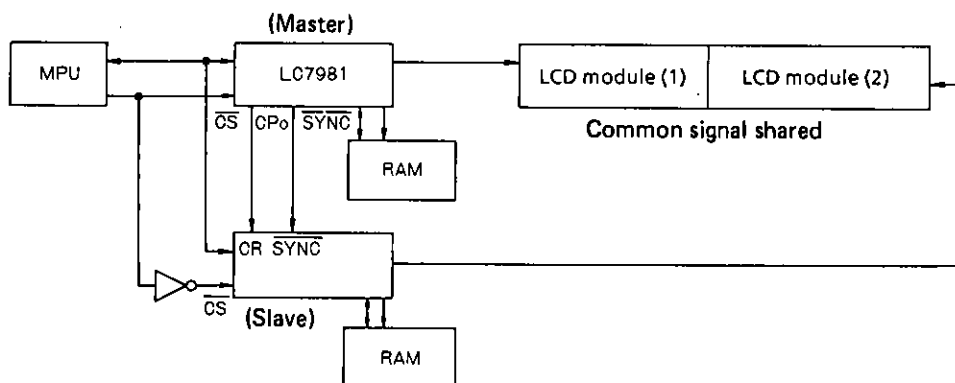
- Character display mode (1) (built-in character generator)



- Character display mode (2) (external character generator)



- Parallel operation



## Built-in character generator

| Upper<br>4bit<br>Lower<br>4bit | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
|--------------------------------|------|------|------|------|------|------|------|------|------|------|------|------|
| xxxx0000                       |      | 0    | a    | P    | `    | P    |      | -    | 3    | E    | a    | P    |
| xxxx0001                       | !    | 1    | A    | Q    | a    | 9    | .    | 7    | +    | 4    | a    | 9    |
| xxxx0010                       | "    | 2    | B    | R    | b    | r    | "    | イ    | ウ    | ×    | P    | a    |
| xxxx0011                       | #    | 3    | C    | S    | c    | s    | ,    | ウ    | 7    | E    | e    | ∞    |
| xxxx0100                       | \$   | 4    | D    | T    | d    | t    | \    | I    | t    | +    | H    | a    |
| xxxx0101                       | %    | 5    | E    | U    | e    | u    | .    | ア    | +    | 1    | e    | U    |
| xxxx0110                       | &    | 6    | F    | V    | f    | v    | 7    | カ    | =    | 3    | P    | Σ    |
| xxxx0111                       | '    | 7    | G    | W    | g    | w    | 7    | +    | 7    | ウ    | g    | π    |
| xxxx1000                       | (    | 8    | H    | X    | h    | x    | 4    | ウ    | *    | リ    | 7    | X    |
| xxxx1001                       | )    | 9    | I    | Y    | i    | y    | ウ    | 7    | リ    | W    | "    | y    |
| xxxx1010                       | *    | :    | J    | Z    | j    | z    | ±    | コ    | n    | レ    | j    | 7    |
| xxxx1011                       | +    | :    | K    | L    | k    | l    | (    | ア    | 7    | ヒ    | *    | ア    |
| xxxx1100                       | ,    | <    | L    | *    | l    | l    | +    | 3    | 7    | 7    | +    | ア    |
| xxxx1101                       | -    | =    | M    | J    | m    | j    | 1    | 7    | ^    | 2    | +    | +    |
| xxxx1110                       | .    | >    | N    | ^    | n    | +    | 3    | ヒ    | *    | イ    | ア    |      |
| xxxx1111                       | /    | ?    | O    | _    | o    | +    | ウ    | リ    | 7    | "    | 6    |      |