NEC

MOS INTEGRATED CIRCUIT

# μPD7228

## PROGRAMMABLE LCD CONTROLLER/DRIVER



### DESCRIPTION

μPD7228 is an LCD controller/driver having all the functions necessary for interfacing an 8-/16-time division dot-matrix LCD with a microprocessor. A single μPD7228 chip is capable of driving a dot-matrix LCD, consisting of up to 50 columns by 8 rows, totalling 400 dots, or an LCD consisting of 42 columns by 16 rows, totalling 800 dots. The LCD controller/driver contains a 5 × 7 dot matrix character generator, conforming to ASCII and JIS (Japan Industrial Standard), so that Japanese characters (katakana) as well as alphanumeric characters can be displayed. The interfacing function can be effected in two modes: serial or 4-bit parallel.

### FEATURES

- Directly drives LCDs
- A single μPD7228 chip can drive two types of LCDs:
  8- and 16-time division LCDs.
  8-time division: 400 (= 50 × 8) dots
  16-time division: 672 (= 42 × 16) dots
- Multiple chip LCD driving system can be configured by using several μPD7228 chips, in which case, the total number of dots that can be driven increases, as follows:
  8-time division: n × 400 (= n × 50 × 8) dots
  16-time division: n × 800 (= n × 50 × 16) dots
- 2 × 50 × 8 bit RAM for display data storage
- Displays dots (graphics) specified by programmer
- Internal character generator displays 64 types of ASCII dot matrices (alphanumeric characters and symbols) with each matrix consisting of 5 × 7 dots. In addition, 96 types of JIS dot matrices (katakana and symbols) can also be displayed.
- Cursor operation commands
- 8-bit serial interface, compatible with 7500, 75X, 87AD, and 78K series
- 4-bit parallel interface, compatible with 7500, 75X, 84/84C series
- Standby function
- CMOS

### ORDERING INFORMATION

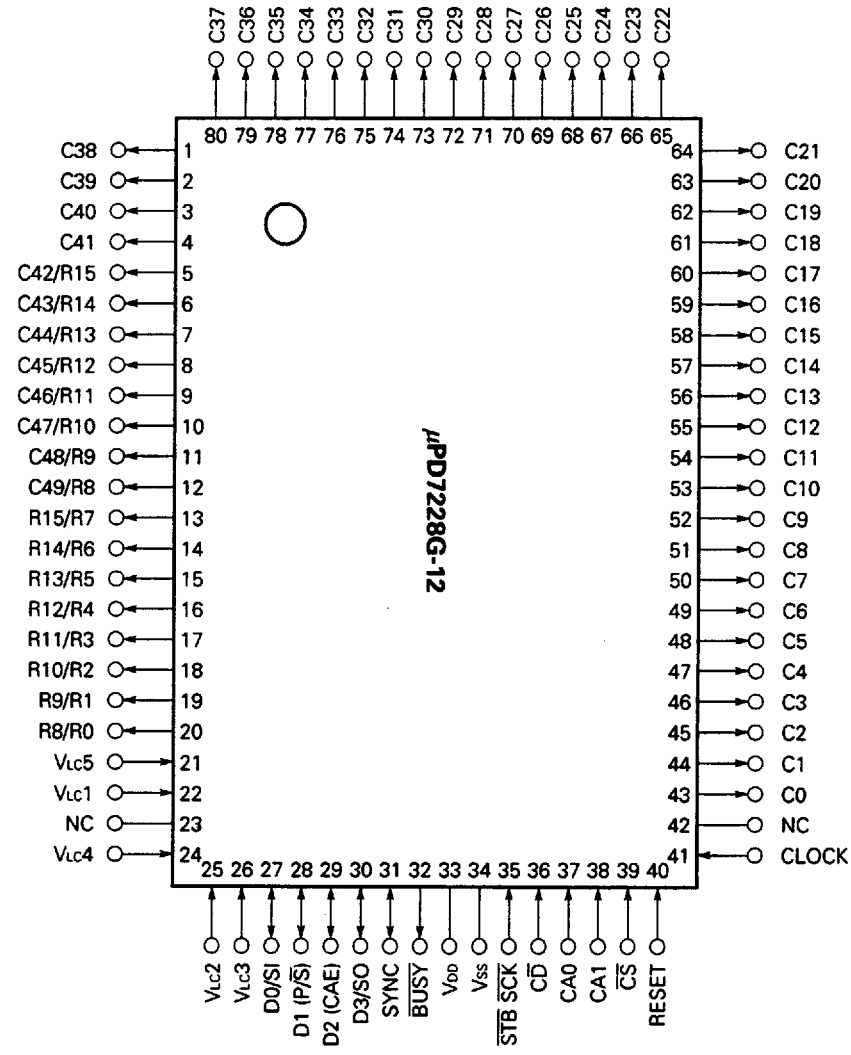
| Part Number | Package Type | Quality Grade |
|---|---|---|
| μPD7228G-12 | 80-pin plastic QFP (14 × 20 mm) | Standard |

Please refer to "Quality grade on NEC Semiconductor Devices" (Document number IEI-1209) published by NEC Corporation to know the specification of quality grade on the devices and its recommended applications.

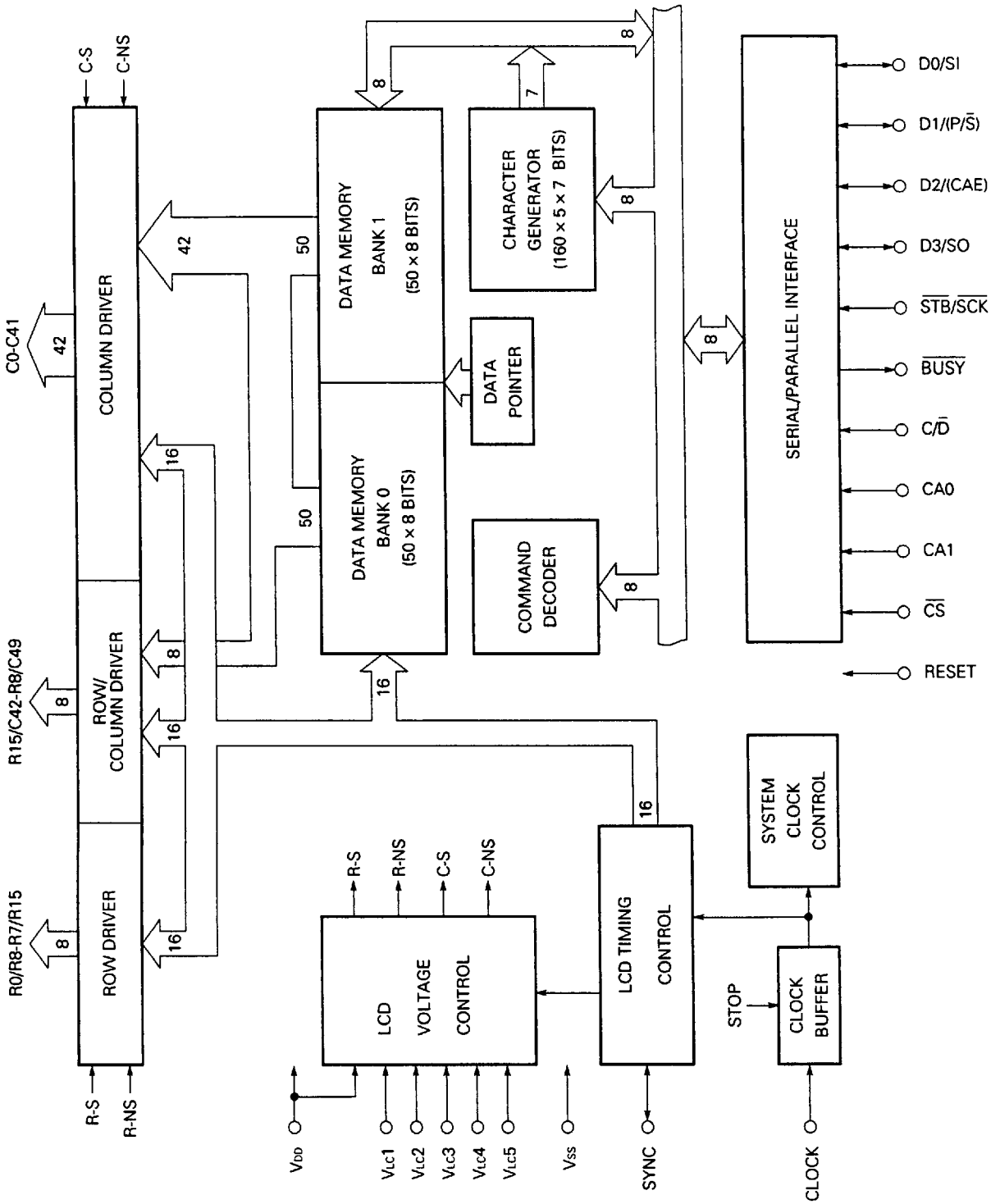


Document No. IC-1534B
Date Published October 1993 P
Printed in Japan

**PIN CONFIGURATIONS (Top View)**



## BLOCK DIAGRAM



CONTENTS

## 1. PIN FUNCTIONS

### 1.1 D0 TO D3 (DATA BUS) ···· 3-STATE I/O

These four pins constitute a data bus in parallel interface mode, and inputs or outputs 4-bit parallel data.

Data on the D0 to D3 pins are read each time the $\overline{STB}$ signal rises. The 4-bit data read at the first rising edge of the $\overline{STB}$ signal is loaded to the higher 4 bits of an internal serial/parallel register, while the data read at the second rising edge of the $\overline{STB}$ signal is loaded to the lower 4 bits of the register.

The contents of the serial/parallel register are output to the D0 to D3 pins each time the $\overline{STB}$ signal falls. When data is read, the higher 4 bits of the register are output to the D0 to D3 pins at the first falling edge of the $\overline{STB}$ signal, and the lower 4 bits of the register are output when the $\overline{STB}$ signal falls second time. In serial interface mode, the D0 pin functions as a serial data input pin (SI) and the D3 pin serves as a serial data output pin (SO).

The D1 pin also selects parallel or serial interface mode (P/$\overline{S}$ pin), and the D2 pin is also used as a chip address enable (CAE) pin.

### 1.2 SI (SERIAL DATA IN) ···· INPUT PIN SHARED BY D0 PIN

In serial interface mode, the D0 pin serves as the SI pin, whose data is loaded to an internal serial/parallel register at the rising edge of the $\overline{SCK}$ signal from the MSB first. The SI pin is internally connected to a Schmitt trigger input circuit having hysteresis characteristics and thus is protected from the adverse effect of noise.

### 1.3 SO (SERIAL DATA OUT) ···· OUTPUT PIN SHARED BY D3 PIN

In serial interface mode, the D3 pin serves as the SO pin that outputs serial data. The contents of an internal serial/parallel register are output to this pin with the MSB first in synchronization with the falling edge of the $\overline{SCK}$ signal.

### 1.4 P/$\overline{S}$ (PARALLEL/SERIAL SELECT) ···· INPUT PIN SHARED BY D1 PIN

If this pin is high, when the RESET signal falls (i.e., when the RESET signal is released), the parallel interface mode is set; if it is low, the serial interface mode is set. This pin is internally connected to a Schmitt trigger input circuit having hysteresis characteristics and thus is protected from the adverse effect of noise.

### 1.5 CAE (CHIP ADDRESS ENABLE) ···· INPUT PIN SHARED BY D2 PIN

The function of this pin is effected when the RESET and P/$\overline{S}$ signals both go low, i.e., when the serial interface mode is set. When this pin goes high in synchronization with the falling of the RESET and P/$\overline{S}$ signals, the chip address function is enabled; when the CAE pin goes low, the chip address function is disabled. This pin is internally connected to a Schmitt trigger input circuit having hysteresis function and thus is protected from the adverse effect of noise.

### 1.6 CA0 AND 1 (CHIP ADDRESS) ···· INPUT

These two pins assign an address to each of several μPD7228 chips, when a multiple chip system is configured. These pins are independent from the CAE input pin in the parallel interface mode, bit, in the serial interface mode, compare the chip address information sent from the CPU when the chip address function of the CAE pin is enabled.

**Table 1-1 Levels of CA1 and 0 Pins**

| Mode | CA1, 0 |
|---|---|
| With chip address function enabled and<br>• always in parallel interface mode<br>• when CAE = 1 in serial interface mode | Set to 00, 01, 10, or 11 (fixed to 00 in a single chip system) |
| With chip address function disabled and when CAE = 0 in serial interface mode | Fixed to 00 |

**Note 1.** When a multiple chip system is configured and used in the serial interface mode, chip address function is enabled by supplying as many decodes $\overline{CS}$ signals as there are μPD7228 chips in the system to each chip in the system, even though the CAE pin is low, disabling the chip address function. At this time, however, the CA1 and 0 pins must both be cleared to 00.

The CA1 and 0 pins are internally connected to a Schmitt trigger input circuit having hysteresis characteristics and thus is protected from the adverse effect of noise.

### 1.7 $\overline{CS}$ (CHIP SELECT) ···· INPUT

This is a low-active chip select signal input pin.

When the chip address function is disabled, inputting a low-level signal to the $\overline{CS}$ pin makes the $\overline{STB}/\overline{SCK}$ input signal and C/$\overline{D}$ input signal valid, enabling command input and data input/output. When the chip address function is enabled, merely lowering the $\overline{CS}$ signal is not sufficient to make the $\overline{STB}/\overline{SCK}$ and C/$\overline{D}$ input signals valid. The chip address information sent from the CPU must also coincide with the signals input to the CA0 and 1 pins.

When the signal input to the $\overline{CS}$ pin is raised to high level, the D3 to D0 and $\overline{BUSY}$ pins unconditionally enter the high-impedance status. The $\overline{CS}$ pin is internally connected to a Schmitt trigger input circuit having hysteresis characteristics and thus is protected from the adverse effect of noise.

### 1.8 $\overline{STB}/\overline{SCK}$ (STROBE/SERIAL CLOCK) ···· INPUT

In the parallel interface mode, this pin inputs a strobe signal that is used to input/output 4-bit parallel data ($\overline{STB}$ pin function). In the serial interface mode, this pin functions as the $\overline{SCK}$ pin that inputs serial clocks necessary for inputting or outputting serial data.

### 1.9 C/$\overline{D}$ (COMMAND/DATA) ···· INPUT

This pin selects commands or data to be input in parallel or serial mode. To input commands, raise the level of this pin; lower the level to input data.

When commands or data are input in the parallel interface mode, the C/$\overline{D}$ pin content is latched at the second rising edge of the STB signal. In the serial interface mode, the C/$\overline{D}$ pin content is latched at the eighth rising edge of the $\overline{SCK}$ signal. Do not change the C/$\overline{D}$ pin level before the $\overline{STB}$ signal rises for the first time in the parallel interface mode.

To output data, always keep the C/$\overline{D}$ pin at the low level, regardless of whether the parallel or serial interface mode is used.

The C/$\overline{D}$ pin is internally connected to a Schmitt trigger input circuit having hysteresis characteristics and thus is protected from the adverse effect of noise.

### 1.10 $\overline{BUSY}$ (BUSY) ···· 3-STATE OUTPUT

This pin outputs an active-low $\overline{BUSY}$ signal to the CPU to inform the CPU that μPD7228 is busy. When the BUSY signal is low, the CPU cannot read or write μPD7228.

The $\overline{BUSY}$ signal goes low at the second rising edge of the $\overline{STB}$ signal in the parallel interface mode; in the serial interface mode, it goes low at the eighth rising edge of the $\overline{SCK}$ signal. When internal processing for μPD7228 is over, the $\overline{BUSY}$ signal goes high.

The $\overline{BUSY}$ signal for a μPD7228 enters the high-impedance status when that μPD7228 is not selected (i.e., when the $\overline{CS}$ pin is high or the chip address sent from the CPU does not coincide with the μPD7228 address).

### 1.11 SYNC (SYNCHRONOUS) ···· 3-STATE OUTPUT

If more than one μPD7228 is used to configure a multiple chip system, where the row drive signal is commonly used, the SYNC pin inputs or outputs a sync signal that synchronizes the phases and frame cycles for the LCD drive AC signals (row/column signals) of all μPD7228s.

In a multiple chip system, one of the several μPD7228s is selected as the master chip. The SYNC pin for the master chip is set in the output mode. All the other chips in the system are regarded as slave chips, whose SYNC pins are set in the input mode.

The SYNC pin for each μPD7228 is set in the input or output mode by the SMM command.

The master chip, whose SYNC pin is set in the output mode, outputs a sync pulse during the last cycle for each frame. The slave chips read the sync pulse output by the master chip and are synchronized with the master chip.

Figs. 1-1 and 1-2 below, respectively show output timing charts for the SYNC signal with 8- and 16-time division LCDs. When only one μPD7228 chip is used in a system, the SYNC pin may be set in either of the input or output modes. However, when the pin is set in the input mode, the potential on the pin must be fixed to Vss. In the output mode, open the SYNC pin.

**Fig. 1-1 SYNC Signal with 8-Time Division LCD**

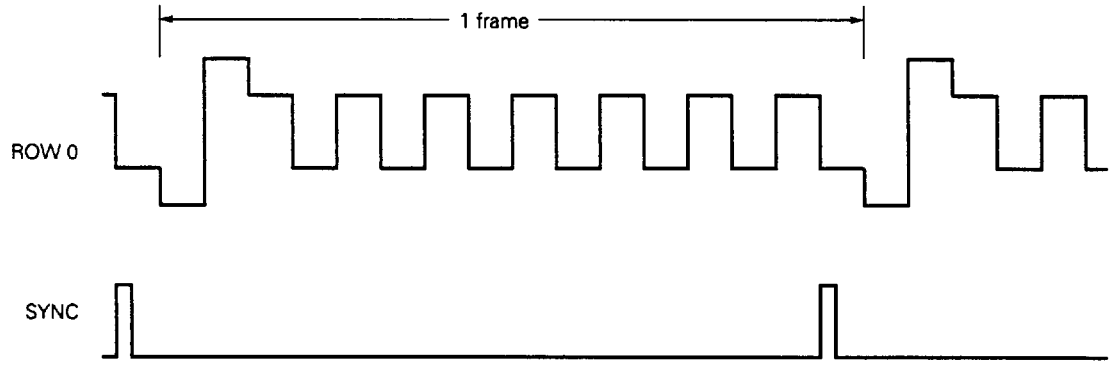
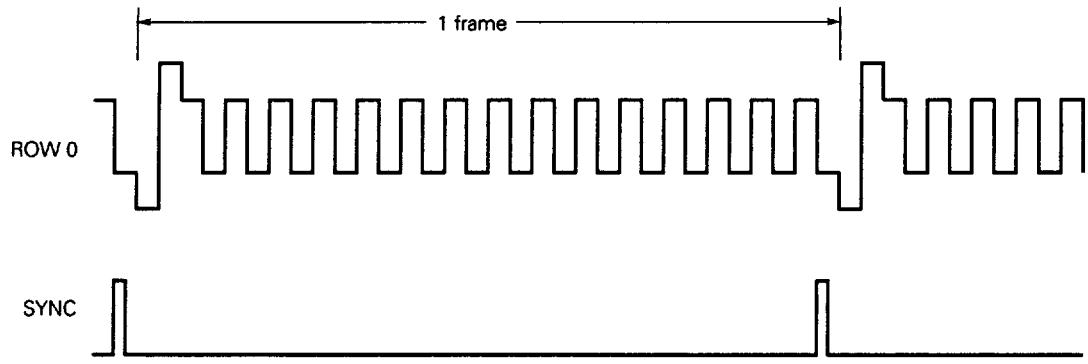


**Fig. 1-2 SYNC Signal with 16-Time Division LCD**




### 1.12 C0 TO C41 (COLUMN) ···· OUTPUT

These pins output LCD column drive signals.

### 1.13 R8/C49 TO R15/C42 (ROW/COLUMN) ···· OUTPUT

These dual-function pins output row drive signals or column drive signals to an LCD. Whether these pins serve as the row drive signal output pins or column drive signal output pins is specified by the SMM command.

### 1.14 R0/R8 TO R7/R15 (ROW) ···· OUTPUT

These eight pins output row drive signals R0 to R7 or R8 to R15. Whether row drive signals R0 to R7 or R8 to R15 are output is specified by the SMM command.

### 1.15 $V_{LC1}$ TO $V_{LC5}$ (LCD DRIVE VOLTAGE SUPPLY) ···· INPUT

These pins input a reference voltage that determines the voltage level for the LCD row and column drive signals.

### 1.16 CLOCK (CLOCK) ···· INPUT

This pin inputs an external clock signal.

### 1.17 RESET (RESET) ···· INPUT

This pin inputs a high-active RESET signal, which takes precedence over any other μPD7228 operations. The RESET signal is also used to release the standby mode or to retain data in the data memory on a low supply voltage.

### 1.18 $V_{DD}$

Positive supply voltage pin

### 1.19 $V_{SS}$

GND pin.

## 2. INTERNAL FUNCTIONAL BLOCKS

### 2.1 SERIAL/PARALLEL INTERFACE

The interface circuit for μPD7228 can function as a serial and parallel interface. Which interface function is effected is determined by the P/$\overline{S}$ pin level, when the RESET signal goes low. That is, the circuit operates as a parallel interface, when the P/$\overline{S}$ pin is high, when the RESET signal goes low, and it functions as a serial interface, when the P/$\overline{S}$ pin is low.

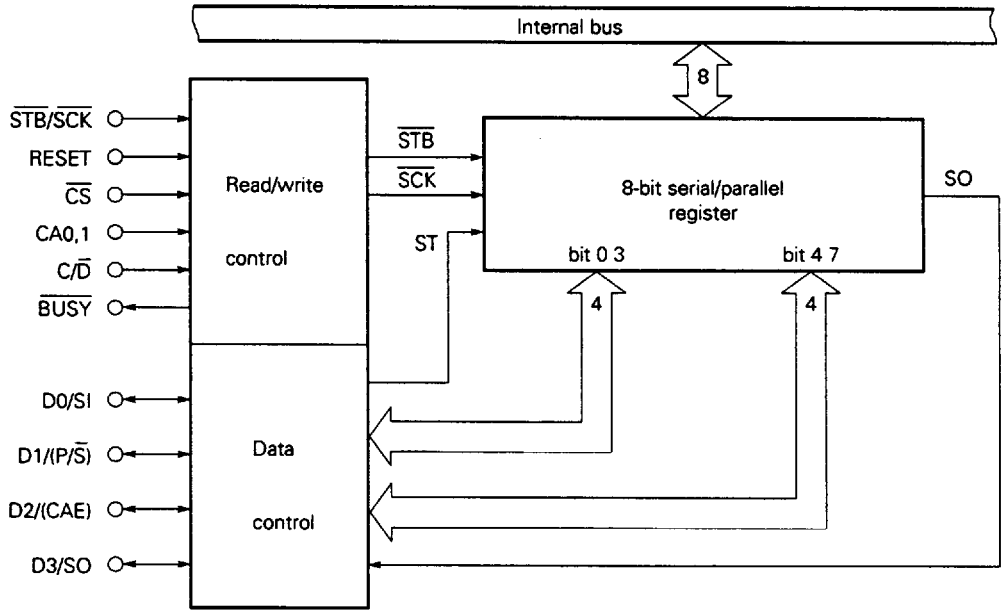
All the commands and data sent from the CPU are written to μPD7228 through this interface circuit. In addition, the interface is also used to output data from μPD7228 to the CPU. The serial/parallel interface operation differs, depending on the data processing mode to be set. When the RESET signal has been input, μPD7228 is set in the write mode, allowing the first command to be input. After that, the LCD controller/driver is set in a write-related or read-related mode by a data processing mode setting command.

When the write, AND, OR, or a character write mode is set as the data processing mode, the serial/parallel interface enters the data input mode and reads data from the SI pin (serial data) or the D3 to D0 pins (4-bit parallel data) in synchronization with the rising edge of the $\overline{SCK}$ or $\overline{STB}$ signal.

When the read mode is set, the serial/parallel interface is set in the data output mode and outputs data from the SO pin (serial data) or D3 to D0 pins (parallel data) in synchronization with the rising edge of the $\overline{SCK}$ or $\overline{STB}$ signal.

The serial/parallel register, shown in Fig. 2-1, serves as a buffer between the 8-bit serial data or two 4-bit parallel data transferred through the serial I/O pins (SI and SO) or parallel I/O pins (D3 to D0), and 8-bit parallel data of the data memory.

**Fig. 2-1 Serial/Parallel Interface**



The data input from the CPU to the serial/parallel interface is sent, if input of command is specified by the C/$\overline{D}$ pin, from the serial/parallel register to a command decoder, where the data is decoded.

When the C/$\overline{D}$ pin specifies input of data, the data loaded to the serial/parallel register is transferred to the data memory as is in the write mode. In the AND or OR mode, the data loaded to the serial/parallel register is ANDed or ORed with the contents of the data memory, and the result of the AND or OR operation is transferred to the data memory. In the character mode, the data loaded to the serial/parallel register is regarded as an ASCII or JIS (Japan Industrial Standard) code, sent to the character generator, decoded into a character display pattern consisting of 5 by 7 bits, and stored in five contiguous addresses of the data memory.

The serial/parallel interface outputs data to the CPU, only when the read mode is set. In the read mode, the serial/parallel interface always read 8-bit data from the data memory in preparation for the next read operation, and sets the data in the serial/parallel register.

In the serial interface mode, the higher 4 bits of the serial/parallel register are output from the D3 to D0 pins at the first rising edge of the $\overline{STB}$ signal, and the lower 4 bits of the register are output at the second rising edge of the $\overline{STB}$ signal. In both the serial and parallel interface modes, when the first 8-bit data has been output, the next 8-bit data is automatically read out from the data memory and is set in the serial/parallel register.

### 2.2 COMMAND DECODER

If the 8-bit data input through the serial/parallel interface is specified by the C/$\overline{D}$ pin to be a command (when the C/$\overline{D}$ pin is 1), the command decoder reads that data as a command, decodes it, and generates an internal control signal.

### 2.3 CHARACTER GENERATOR

The character generator becomes valid when a character mode setting command (SCML or SCMR) has been implemented. It interprets the 8-bit data written to μPD7228 through the serial/parallel interface as ASCII codes (alphanumeric characters and symbols) or JIS codes (katakana and symbols), generates 5×7 dot matrix patterns corresponding to the codes, and transfers the dot matrix patterns to five contiguous addresses of the data memory. Therefore, the character generator transfers 7-bit dot matrix pattern five times to the memory.

The character generator can generate a total of 160 types of characters, which are broken down as follows:

| ASCII | | JIS | |
|---|---|---|---|
| Uppercase characters: | 26 | Katakana: | 55 |
| Lowercase characters: | 26 | Symbols: | 9 |
| Numerals : | 10 | | |
| Symbols : | 34 | | |

Fig. 2-3 shows the correspondence between the character (ASCII or JIS) codes and 5×7 dot display patterns. Ninety-six codes, from 20H to 7FH, correspond to ASCII characters and sixty-four codes, from A0H to DFH, correspond to JIS characters. Fig. 2-2 shows the LCD configuration when the character generator is used. Each character consists of 5 by 7 dots. Because the character generator does not use the most significant bits (bit 7) of data memory addresses, the dots for an LCD (R7 in the 8-time division mode and R7 and R15 in the 16-time division mode) corresponding to the most significant bits can be used as a cursor or the display pattern of an indicator, independently from the character generator.

The most significant bits can be manipulated by cursor manipulation commands (WRCURS and CLCURS).

**Fig. 2-2 LCD Configuration**

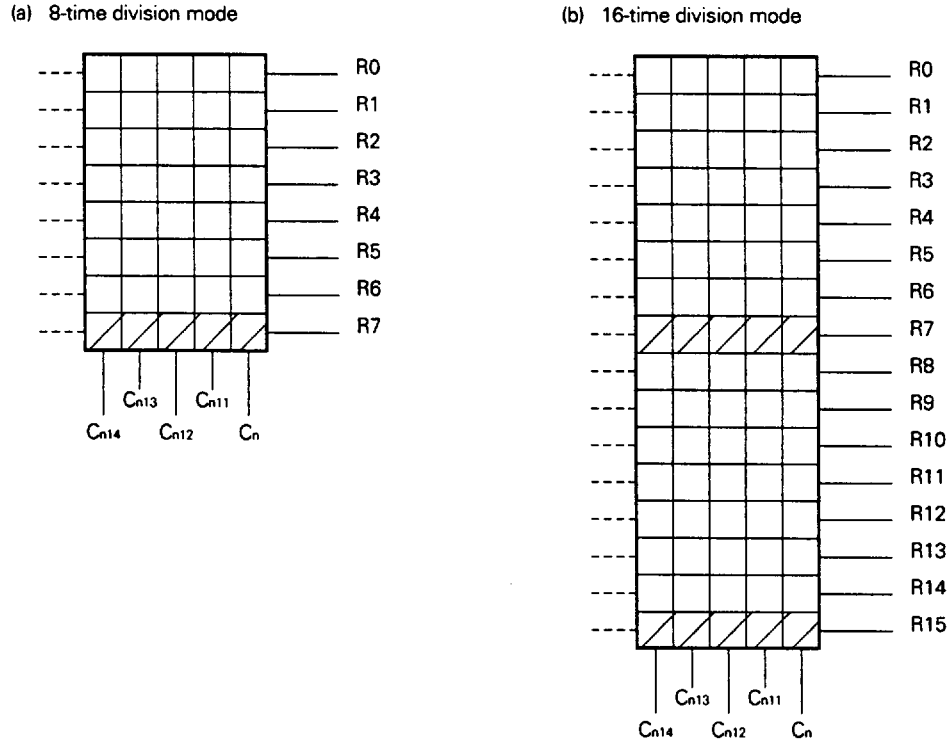


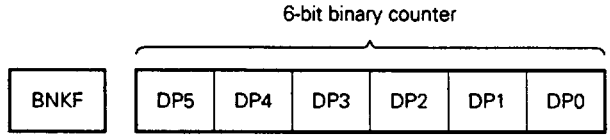
**Fig. 2-3 Character Codes and Displayed Patterns**


| | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| | | | | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| | | | | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | | ! | " | # | $ | % | & | ' | ( | ) | * | + | , | - | . | / |
| 0 | 0 | 1 | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | ? |
| 0 | 1 | 0 | 0 | @ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| 0 | 1 | 0 | 1 | P | Q | R | S | T | U | V | W | X | Y | Z | [ | ¥ | ] | ^ | _ |
| 0 | 1 | 1 | 0 | ` | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| 0 | 1 | 1 | 1 | p | q | r | s | t | u | v | w | x | y | z | { | \| | } | ~ | |
| 1 | 0 | 1 | 0 | | ｡ | ｢ | ｣ | ､ | ･ | ｦ | ｧ | ｨ | ｩ | ｪ | ｫ | ｬ | ｭ | ｮ | ｯ |
| 1 | 0 | 1 | 1 | ｰ | ｱ | ｲ | ｳ | ｴ | ｵ | ｶ | ｷ | ｸ | ｹ | ｺ | ｻ | ｼ | ｽ | ｾ | ｿ |
| 1 | 1 | 0 | 0 | ﾀ | ﾁ | ﾂ | ﾃ | ﾄ | ﾅ | ﾆ | ﾇ | ﾈ | ﾉ | ﾊ | ﾋ | ﾌ | ﾍ | ﾎ | ﾏ |
| 1 | 1 | 0 | 1 | ﾐ | ﾑ | ﾒ | ﾓ | ﾔ | ﾕ | ﾖ | ﾗ | ﾘ | ﾙ | ﾚ | ﾛ | ﾜ | ﾝ | ﾞ | ﾟ |

## 2.4 DATA POINTER

The data pointer comprises a 6-bit binary counter (bits DP5 to DP0) and 1 bit of bank flag (BNKF), as shown in Fig. 2-4, and points to an address of the data memory.

**Fig. 2-4 Data Pointer Configuration**




The contents of the bank flag and the 6-bit binary counter are simultaneously set by the immediate data from the LDPI command. The BNKF content specifies one of two banks of the data memory (i.e., bank 0 is specified when BNKF is 0, and bank 1 is specified when BNKF is 1), while the contents of the 6-bit binary counter specifies an address (00H to 31H) in the bank specified by the BNKF.

The 6-bit binary counter is an up-down counter, whose contents are modified (increased by 1, decreased by 1, or retained) in the read, write, AND, and OR modes as specified by a mode setting command each time 8-bit data is input or output.

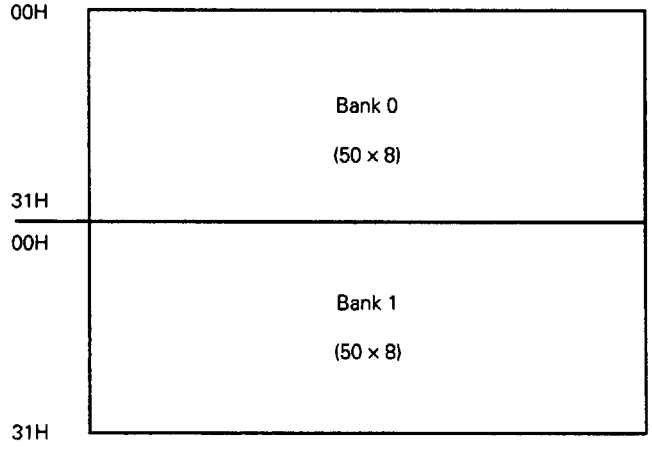
When the BSET or BRESET command is carried out, the 6-bit binary counter contents are modified as directed by the command. In the character mode, the counter contents are increased or decreased by five each time 8-bit data is input or output, or each time a cursor processing command is carried out.

**Note** The 6-bit binary counter contents can be a value outside the address range of the data memory. For example, although the lower limit of the address range is 00H and the upper limit is 31H, it is possible to decrease the counter contents from 00H by one to 3FH, or to increase them from 31H by one to 32H. However, the data memory accomplishes nothing in response to a command that specifies an address from 32H to 3FH.

## 2.5 DATA MEMORY

The data memory is a static RAM, which consists of two banks, with each bank consisting of 50 words by 8 bits, as shown in Fig. 2-5. The memory stores display data.

**Fig. 2-5 Data Memory Configuration**



A data memory bank is specified by the bank flag (BNKF) for the data pointer and is addressed by the 6-bit binary counter for the data pointer.

When the CPU writes 8-bit data to the serial/parallel interface for μPD7228, the data is processed or decoded in accordance with a data processing mode set in advance, and then is written to the data memory.

The data memory contents are directly manipulated by a bit manipulation command.

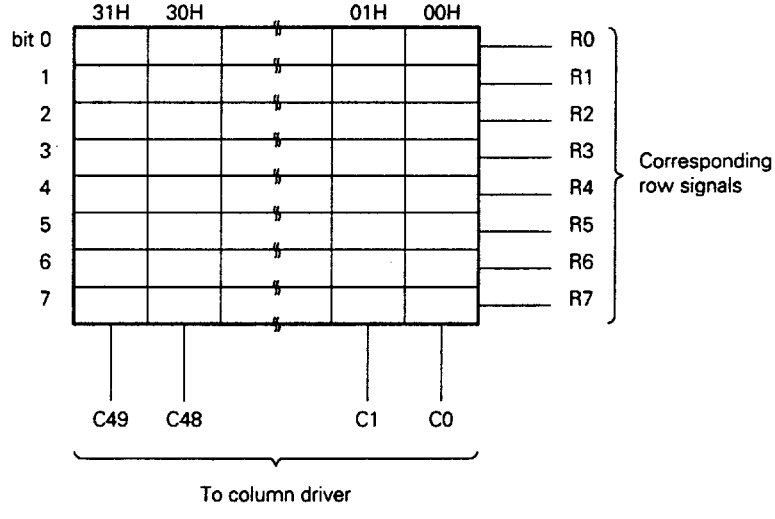
When μPD7228 is set in the read mode, the data memory contents are output to the CPU through the serial/parallel interface. The data memory contents can also be independent from the write and read operations of commands and data, which are transferred between μPD7228 and CPU through the serial/parallel interface, and can be read bitwise in synchronization with the row drive signals and sent to the column driver. How the display data is read out to the column driver differs, depending on the number of divisions for the LCD, more specifically, the number of rows for the LCD.

**(1) When the LCD has 8 rows (in single/multiple chip configuration)**

The display data in one of the two banks specified by the SMM command is read to the column driver.

Each bit in the specified data memory bank corresponds to each dot of the LCD, as illustrated in Fig. 2-6, and when a data memory bank bit corresponding to an LCD dot at the intersection of a row and a column of the LCD is 1, the LCD dot lights; when the bit is 0, the dot remains dark.

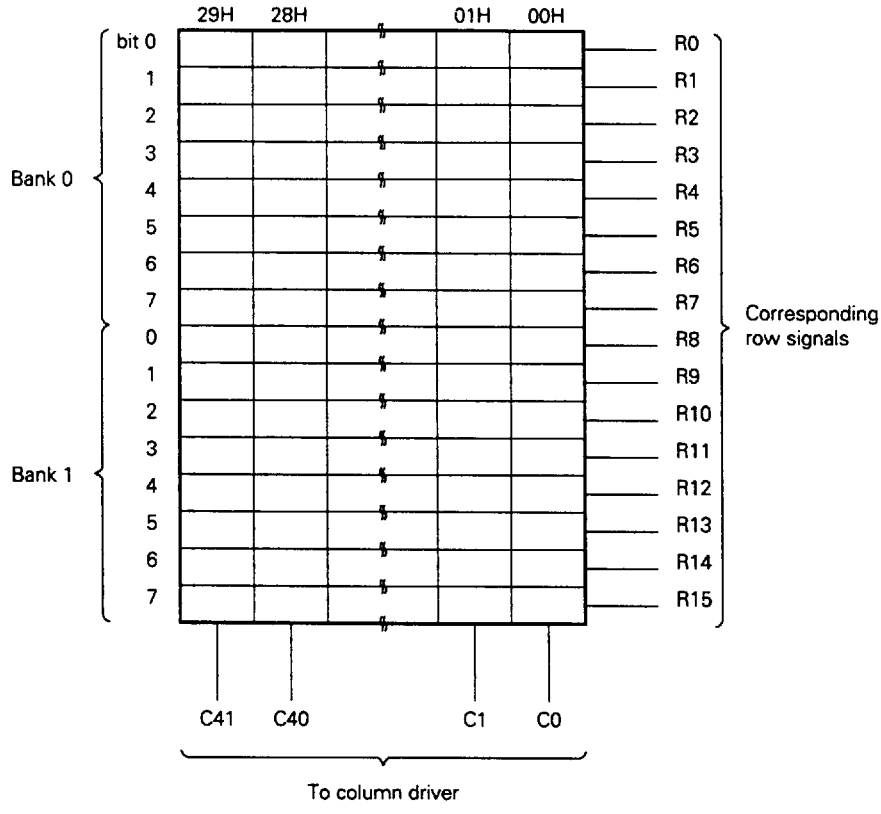
**Fig. 2-6 Data Memory**



**(2) When the LCD has 16 rows (in single chip configuration)**

Both banks 0 and 1 of the data memory are used to drive a total of 672 LCD dots (= 42 words by 16 bits, with data memory addresses 30H and 31H unused). Each bit in the data memory corresponds to each dot of the LCD which, again, is at the intersection of a row and a column of the LCD, as shown in Fig. 2-7.

**Fig. 2-7 Data Memory**



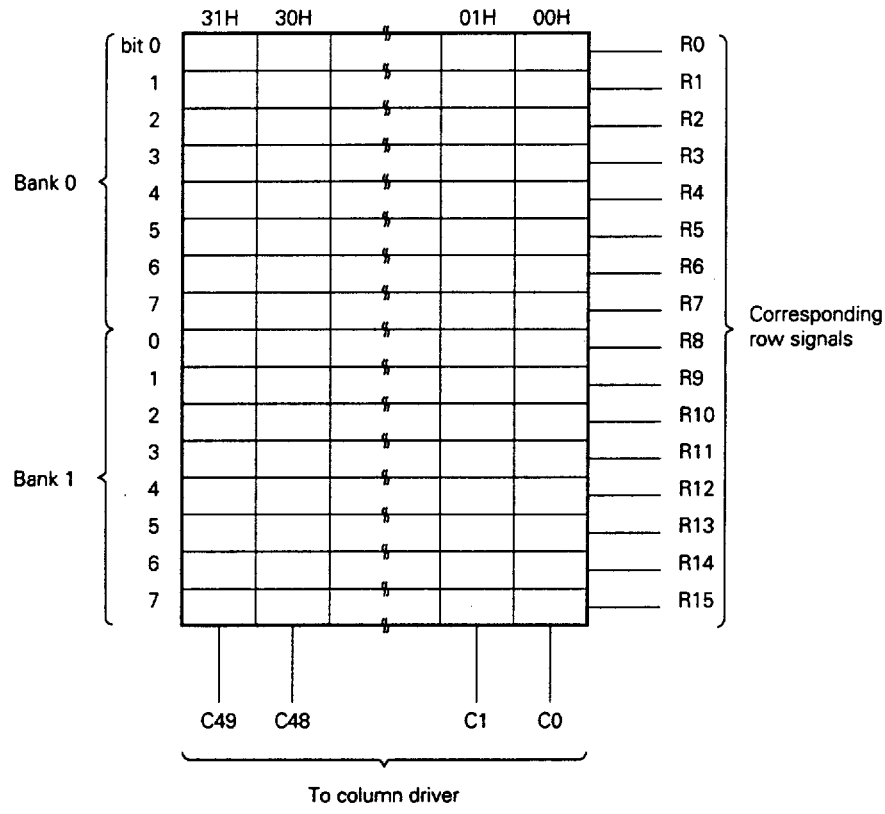

When a bit in the data memory corresponding to an LCD dot at the intersection of a row and a column is 1, the dot lights; when the bit is 0, the dot remains dark.

**(3) When the LCD has 16 rows (in multiple chip configuration)**

Both banks 0 and 1 of the data memory are used to drive a total of 800 LCD dots (= 50 words by 16 bits, because, this time, all the words of the data memory are used). Note that one μPD7228 chip outputs eight row drive signals: R0 to R7, or R8 to R15.

Fig. 2-8 shows the correspondence between the data memory bits and the bits of the row driver and column driver for one μPD7228 chip. Note that, although all the 16 row drive signals, R0 to R15, are shown in this figure, one μPD7228 outputs only eight row drive signals, say, R0 to R7, and that the rest of the signals, R8 to R15, are output by another μPD7228.

**Fig. 2-8 Data Memory**



### 2.6 LCD VOLTAGE CONTROL CIRCUIT

This circuit divides the DC voltage supplied from an external power source through the LCD drive reference voltage input pins ($V_{LC}1$ to $V_{LC}5$) by an AC signal synchronous with the CLOCK and SYNC signals, and supplies a signal to the row and column drivers. The signal supplied to the drivers determines voltage levels at which row and column drive signals should be selected or not, as shown in Table 2-1. The signal also determines the phase of the row and column drive signals.

**Table 2-1**

| | | 8-time division | | 16-time division | |
|---|---|---|---|---|---|
| | | − | + | − | + |
| Row | Selected | $V_{LC}0$ | $V_{LC}5$ | $V_{LC}0$ | $V_{LC}5$ |
| | Not selected | $V_{LC}4$ | $V_{LC}1$ | $V_{LC}4$ | $V_{LC}1$ |
| Column | Selected | $V_{LC}5$ | $V_{LC}0$ | $V_{LC}5$ | $V_{LC}0$ |
| | Not Selected | $V_{LC}2$* | $V_{LC}2$ | $V_{LC}3$ | $V_{LC}2$ |

* $V_{LC}2 = V_{LC}3$

### 2.7 LCD TIMING CONTROL CIRCUIT

This circuit generates timing signals necessary for automatic reading of display data and for driving the LCD in accordance with a frame frequency specified by the SFF command, based on the clock signal and with the number of time divisions specified by the SMM command. The circuit then supplies the timing signals to the data memory, row and column drivers, and LCD voltage control circuit.

If the SYNC signal is set in the output mode by the SMM command, the SYNC signal is output in each frame cycle. If the SYNC signal input mode is set by the command, the SYNC signal supplied from another μPD7228 chip is input and synchronized in frame cycle units to generate the timing signals.

The SYNC signal input/output function is used to synchronize the LCD drive timing for each μPD7228 chip in multiple chip configuration.

### 2.8 ROW AND COLUMN DRIVERS

μPD7228 is provided with three types of drivers. One is a column driver that drives column signals C0 to C41. Another is a row driver that drives row signals R0 to R15. The other is a row/column driver that drives row signals R15 to R8 and column signals C42 to C49.

The functions of the row driver and row/column driver are determined by the SMM command, as shown in Table 2-2.

**Table 2-2**

| $M_2$ | $M_1$ | $M_0$*1 | Number of rows | R0/R8-R7/R15 | R15/C42-R8/C49 |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 8 | R0-R7 | C42-C49 |
| 0 | 0 | 1 | | | |
| 0 | 1 | 0 | | | |
| 0 | 1 | 1 | | | |
| 1 | 0 | 0 | 16 | R8-R15*2 | |
| 1 | 0 | 1 | | | |
| 1 | 1 | 0 | | R0-R7*3 | |
| 1 | 1 | 1 | | | R15-R8 |

*1. $M_2$, $M_1$ and $M_0$ are the specification codes for the SMM command.

*2. R0 to R7 are output by another chip.

*3. R8 to R15 are output by another chip. However, when $M_2M_1M_0$ = 111, pins R15/C42 to R8/C49 are used to output row signals in single chip configuration, so that a single chip for μPD7228 outputs all the 16 row signals.

These drivers select an analog voltage corresponding to the display data read out from the data memory and the timing signal supplied by the LCD timing control circuit, creates row and column drive signals that can directly drive an LCD, and outputs the signals to the row and column pins. The level and phase of a selected analog voltage are determined by the LCD voltage control circuit.

## 3. DATA INPUT/OUTPUT OPERATIONS

Commands and data μPD7228 handles are 1 byte (8 bits) long. A 1-byte command or data is processed each time it has been transferred between the CPU and μPD7228, regardless of whether the serial or parallel interface mode is set.

Whether or not byte data has been correctly transferred is checked by a byte counter (octal/binary counter) that counts eight falling edges of the $\overline{SCK}$ signal or two falling edges of the $\overline{STB}$ signal.

This counter is unconditionally cleared when the $\overline{CS}$ signal or RESET signal goes high, so that the counter gets ready for counting the new byte data. If the $\overline{CS}$ signal or RESET is allowed to go high while a byte is being transferred, the byte may not be correctly transferred.

A transferred byte is processed as 8-bit serial data in the serial interface mode. When eight serial clocks ($\overline{SCK}$) have been supplied to a selected μPD7228, it is assumed that 1-byte data has been input or output, and consequently, the μPD7228 starts its internal processing. The μPD7228 lowers the $\overline{BUSY}$ signal at the eighth rising edge of the $\overline{SCK}$ signal, informing the CPU of its busy state. When the internal processing has finished, the μPD7228 raises the $\overline{BUSY}$ signal to inform the CPU that the next byte can be transferred.

Serial data is input and output with its MSB first (see Figs. 3-1 and 3-2).

In the serial interface mode with the chip address function enabled, 8-bit serial data (through only the lower 2 bits of it have meanings), which indicates a chip address, is written to all the μPD7228s in the system, and only the μPD7228, whose address coincides with the input 8-bit serial data, is allowed to input commands or input or output data (see Figs. 3-3 and 3-4). In the parallel interface mode, 8-bit data is divided into two 4-bit data and input because the data bus of μPD7228 is 4 bits wide (D3 to D0). When the parallel data strobe signal ($\overline{STB}$) is input twice to a selected μPD7228, it is assumed that 1-byte data has been input to or output from that μPD7228. The μPD7228 starts the internal processing, raising the $\overline{BUSY}$ signal at the second rising edge of the $\overline{STB}$ signal to inform the CPU that it is busy. When the internal processing of the μPD7228 has ended, the $\overline{BUSY}$ signal is raised, indicating that the CPU can transfer the next byte to the μPD7228.

The higher 4 bits of 8-bit parallel data are input to or output from μPD7228 at the first rising edge of the $\overline{STB}$ signal, while the lower 4 bits are input/output when the $\overline{STB}$ signal rises the second time.

The parallel interface for μPD7228 is compatible to 8243 I/O expander. Therefore, 8-bit parallel data can be input to μPD7228 in the same manner as when outputting 4-bit data twice to 8243. Moreover, the 8-bit data for μPD7228's serial/parallel register can be read out in the same manner as when reading 4-bit data out twice from 8243.

In the parallel interface mode, the chip address function for μPD7228 is always enabled, and chip address information is input through the D1 and D0 pins at the first falling edge of the $\overline{STB}$ signal after the CS signal has fallen. As the chip address information, the lower 2 bits of a command code, output from the CPU as data selecting ports 4 to 7 of 8243, are used. After the $\overline{CS}$ signal has risen, the CPU outputs a command code at the second falling edge of the $\overline{STB}$ signal, but this command code is meaningless for μPD7228 (see Figs. 3-5 and 3-6). For details on the chip address function, refer to 4.

**Fig. 3-1 Serial Input Timing (with chip address function disabled)**

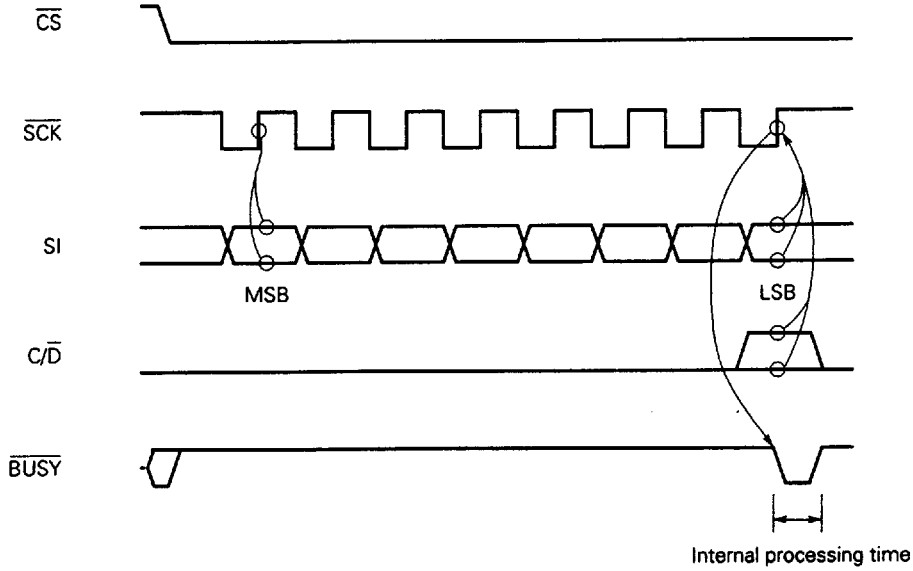
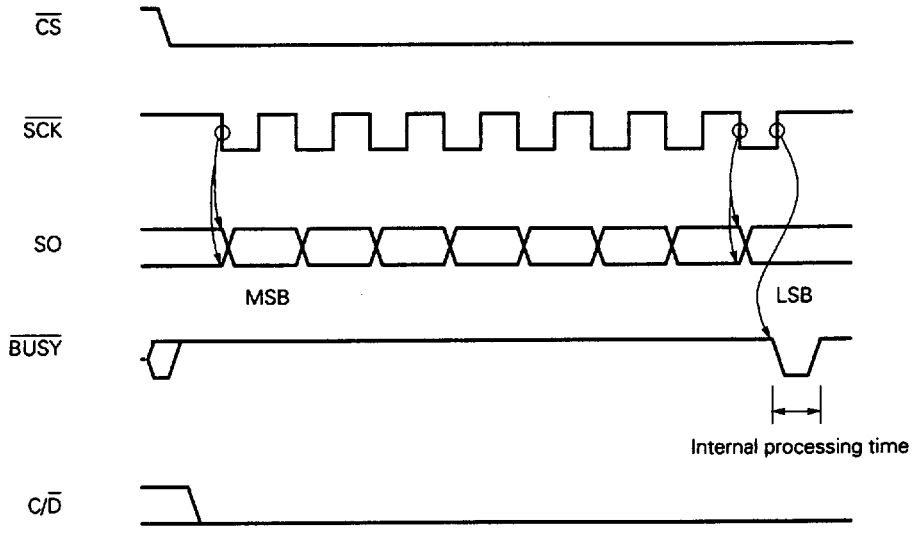



**Fig. 3-2 Serial Output Timing**



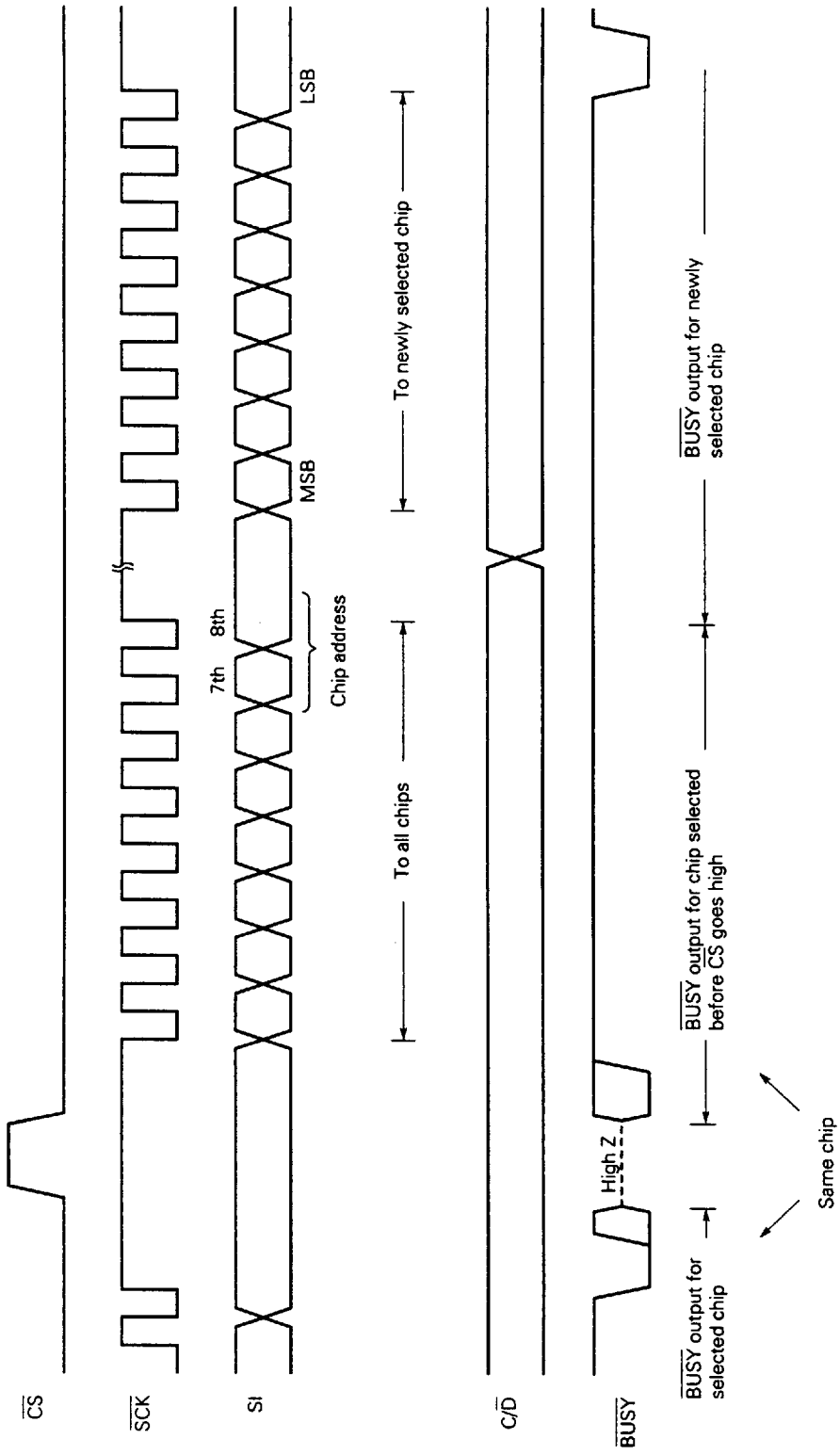



Fig. 3-3 Serial Input Timing (with chip address function enabled)

Fig. 3-4 Serial Output Timing (with chip address function enabled)*

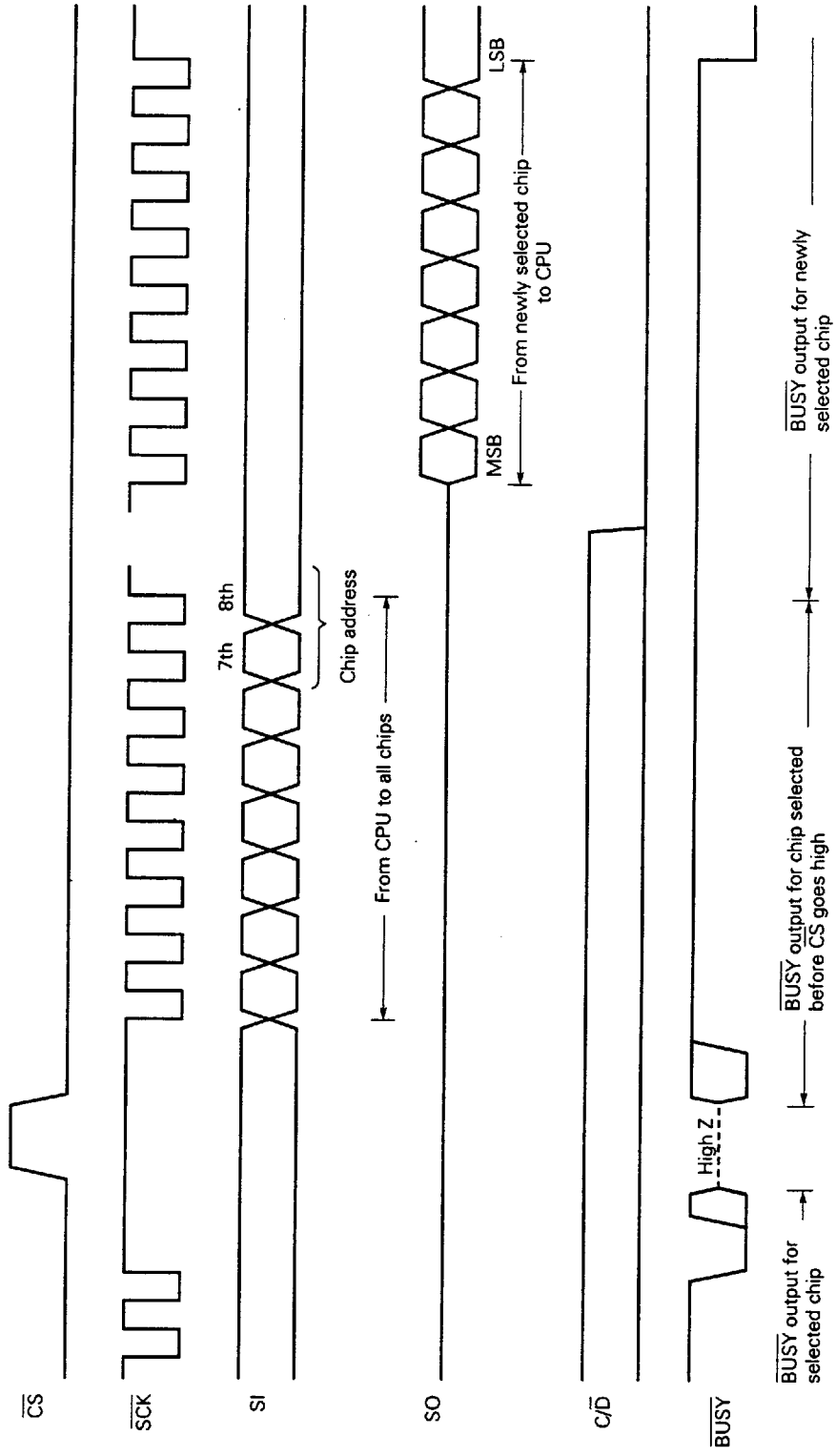


**Fig. 3-5 Parallel Input Timing**

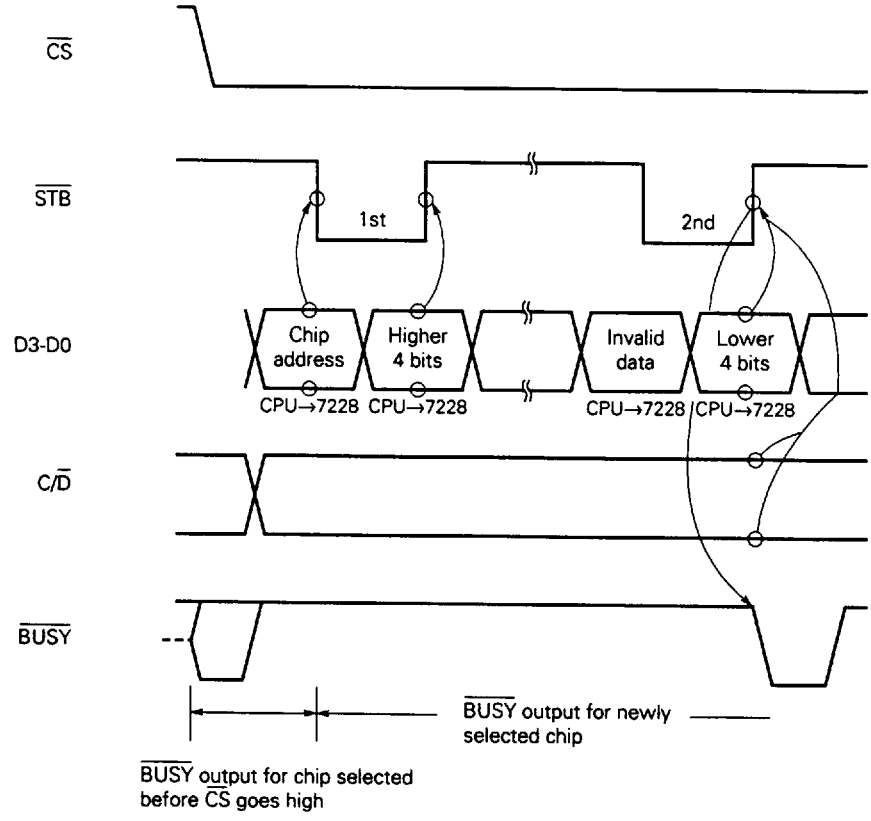
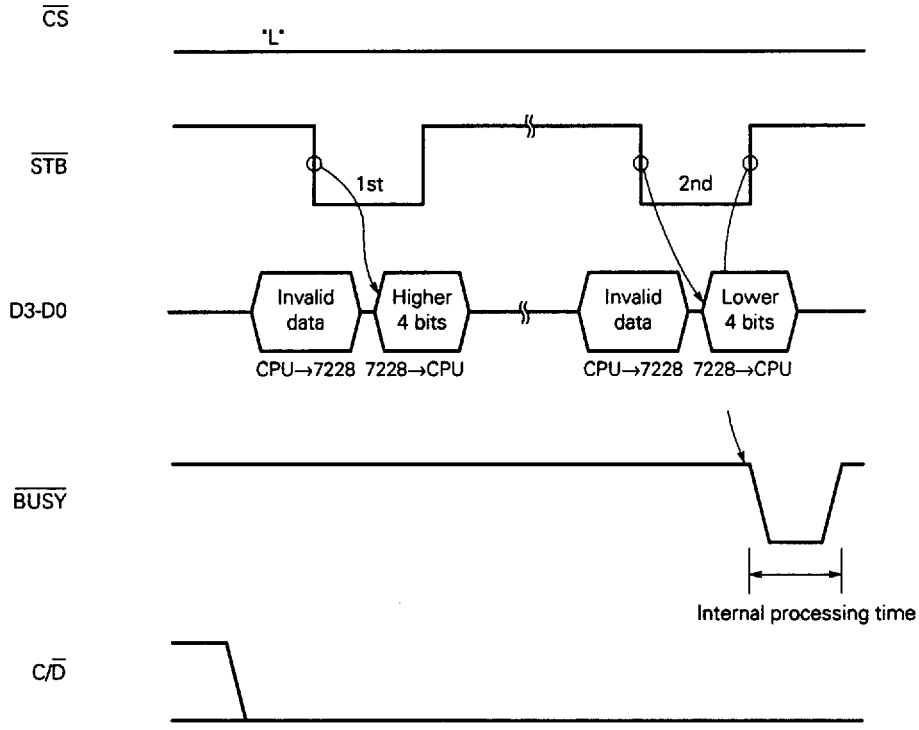


**Fig. 3-6 Parallel Output Timing**




## 4. SELECTING FUNCTION INTERFACING BETWEEN μPD7228 AND CPU

Although commands and data processed by μPD7228 are 8 bits long, μPD7228 communicates with the CPU through an 8-bit serial or 4-bit parallel interface. In addition, μPD7228 has a chip address select function, so that more than one chip for μPD7228 can configure a multiple chip system.

Whether the serial or parallel interface of μPD7228 is used, and whether the chip address select function is used or not are specified by the data sent by the CPU through the D2 (CAE) and D1 (P/$\overline{S}$) pins for μPD7228, when the RESET signal for μPD7228 is released (i.e., falls).

**Fig. 4-1 Interface Specification Timing**

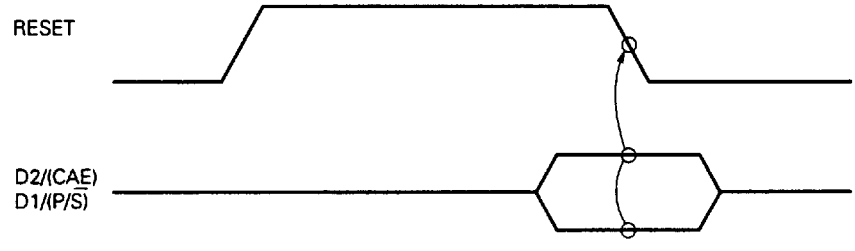


**Table 4-1 Interface Specification Codes**

| D2/(CAE) | D1/(P/$\overline{S}$) | Specifies serial/parallel interface | Specifies chip address select function |
|---|---|---|---|
| 0 | 0 | Serial | Disabled |
| 1 | 0 | | Enabled |
| 0/1 | 1 | Parallel | |

### 4.1 DUAL-FUNCTION PINS FUNCTIONS

Depending on whether the serial interface or parallel interface is specified, the functions for dual-function pins, used to input clock and to input/output data pins ($\overline{STB}$/$\overline{SCK}$, and D3/SO, D0/SI), differ as shown in Table 4-2.

**Table 4-2 Dual-Function Pins Functions**

| Pin | With serial interface (P/$\overline{S}$ = 0) | With parallel interface (P/$\overline{S}$ = 1) |
|---|---|---|
| $\overline{STB}$/$\overline{SCK}$ | $\overline{SCK}$ input | $\overline{STB}$ input |
| D3/SO | SO output | D3 to D0 I/O (4-bit parallel data bus) |
| D2/(CAE) | — | |
| D1/(P/$\overline{S}$) | — | |
| D0/SI | SI input | |

### 4.2 CHIP ADDRESS SELECT FUNCTION

The chip address select function is to select (i.e., allow commands and data to be input or output by) one of several μPD7228s connected to the CPU, so that the CPU does not have to send several chip select ($\overline{CS}$) signals. Each μPD7228 chip is assigned a chip address in advance (by CA0 and CA1 pins). The CPU sends 2-bit chip address information as serial or parallel data. The chip address select function for a μPD7228 compares the chip address for the μPD7228 with the chip address information sent by the CPU, and, if both the addresses coincide, it enables the μPD7228 to input or output commands and data. This function is unconditionally effected in the parallel interface mode. However, in the serial interface mode, it is enabled when the D2/(CAE) signal is set to 1, when the RESET signal has been released.

**(1) In parallel interface mode (See Fig. 3-5 and 3-6.)**

The chip address information is the 2-bit data read to the D1 (corresponding to CA1) and D0 (corresponding to CA0) pins at the first rising edge of the $\overline{STB}$ signal after the CS signal has fallen.

Since the μPD7228 interface is compatible to the interface for 8243 I/O expander, μPD7228 can be connected by using the 8243 interface function for μPD7500 or μPD80C48. The chip address information (0 to 3) for that μPD7228 can be input to the D1 and D0 pins by executing an output or input instruction for the ports 4 to 7 of 8243, when the $\overline{STB}$ signal falls.

**(2) In serial interface mode (See Fig. 3-3 and 3-4.)**

The 2-bit chip address information is the lower 2 bits of the data read to the SI pin at the seventh (corresponding to CA1) and eighth (corresponding to CA0) rising edges of the SCK signal after the CS signal has fallen, i.e., the first 8-bit data input to the SI pin.

**Note** In multiple chip configuration, the chip address comparison data for each μPD7228 is cleared to 00, when the RESET signal has been input. When the CS signal goes low, immediately after the RESET signal has been released, the chip whose CA1 and CA0 pins are cleared to 00 raises its BUSY signal to the high level, informing the CPU that the chip can now be accessed. The CPU starts accessing this chip, when it has detected the high level of the BUSY signal. However, the CA1 and 0 pins of the chip to be accessed first do not necessarily have to be 00, and the chip to be accessed first can be determined by the chip address information the CPU sends, when it accesses a chip.

**Note** In a multiple chip system using a parallel interface, when a chip which should alternate with another chip is continuously selected in the read mode, do not read that chip immediately when it has been selected for the second time. Before reading the chip, implement a data pointer load command to set a restart address. The data pointer load command must also be carried out before reading a chip, if the chip alternating with another chip has contiguous read addresses.

## 5. SUPPLYING LCD DRIVE REFERENCE VOLTAGE

The way the LCD drive reference voltage is supplied to μPD7228 differs, depending on whether μPD7228 controls/drives an 8- or 16-time division LCD, as shown in Fig. 5-1 and 5-2.

**Fig. 5-1 With 8-Time Division LCD**

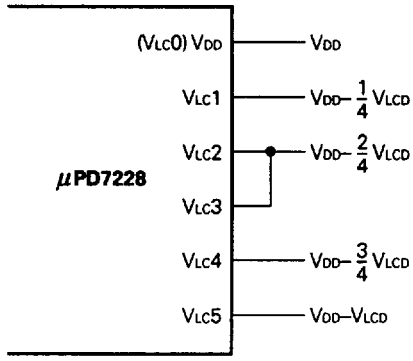
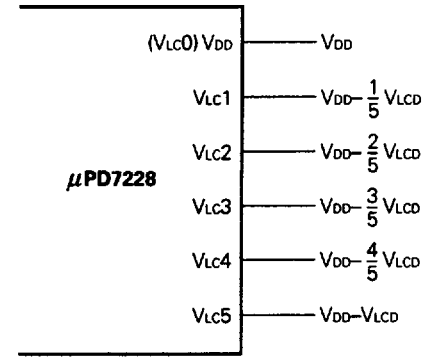



**Fig. 5-2 With 16-Time Division LCD**




**Remarks** Keep the LCD drive voltage ($V_{LCD}$) to $V_{DD}$ or less, regardless of whether an 8- or 16-time division LCD is to be driven.

### 5.1 LCD DRIVE REFERENCE VOLTAGE DIVIDER

The LCD drive reference voltages, shown in Figures 5-1 and 5-2, respectively, can be supplied voltage dividers connected across the $V_{DD}$ and Vss pins, as shown in Fig. 5-3 and 5-4, respectively.

**Fig. 5-3 With 8-Time Division LCD**

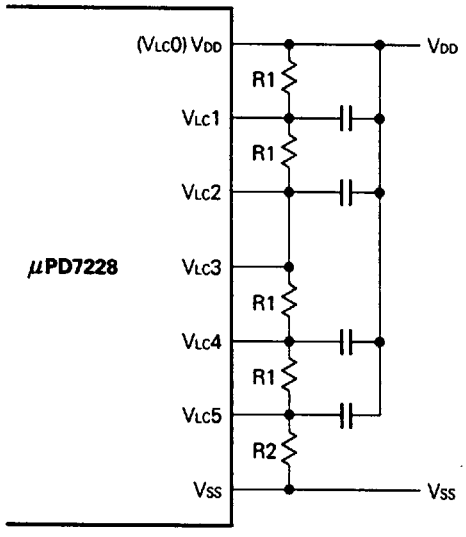
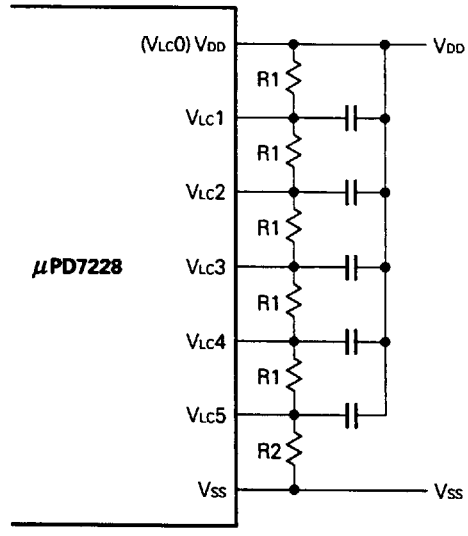



**Fig. 5-4 With 16-Time Division LCD**




In the above figures, the values for resistors R1 and R2 are determined by the following formulas:

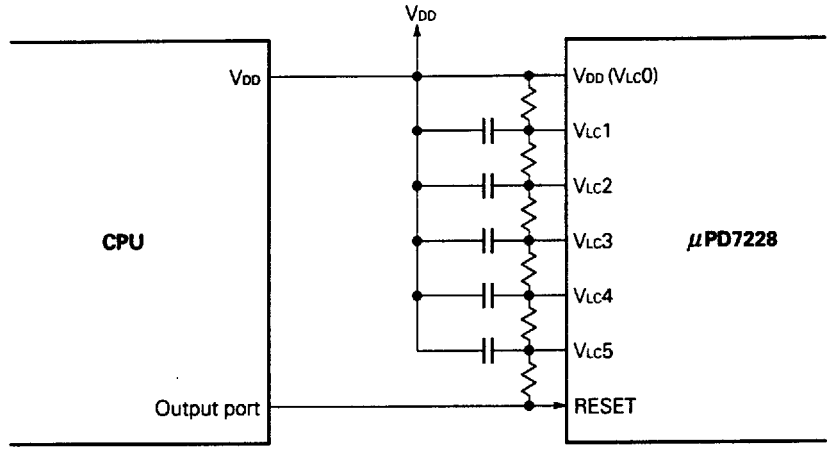
$$R1 = \frac{V_{LCD}}{4\,(V_{DD} - V_{LCD})} \times R2 \quad \text{(8-time division)}$$

$$R1 = \frac{V_{LCD}}{5\,(V_{DD} - V_{LCD})} \times R2 \quad \text{(16-time division)}$$

### 5.2 REDUCING CURRENT DISSIPATION WITH RESET SIGNAL

When the LCD drive reference voltage is supplied through a voltage divider, a slight amount of current leaks through the resistors across the $V_{DD}$ and Vss pins, even when μPD7228 is set in the STOP mode or is reset. In a system such as the one powered by a battery where the leakage current may have an adverse effect, it is necessary to keep the supply current from flowing through the resistors in the voltage divider. This can be done by cutting off the current path by an external circuit, as shown in Fig. 5-5. The circuit shown in this figure uses the RESET signal and prevents the current from flowing into the resistors, when the RESET signal goes high.

**Fig. 5-5 Current Path Control with RESET Signal**




**Remarks** The power source for the CPU and μPD7228 must be the same.

## 6. DISPLAY EXAMPLES

Fig. 6-1 shows the contents for the data memory and the corresponding LCD display patterns, when μPD7228 displays the characters "AEZ" on an 8-time division LCD.

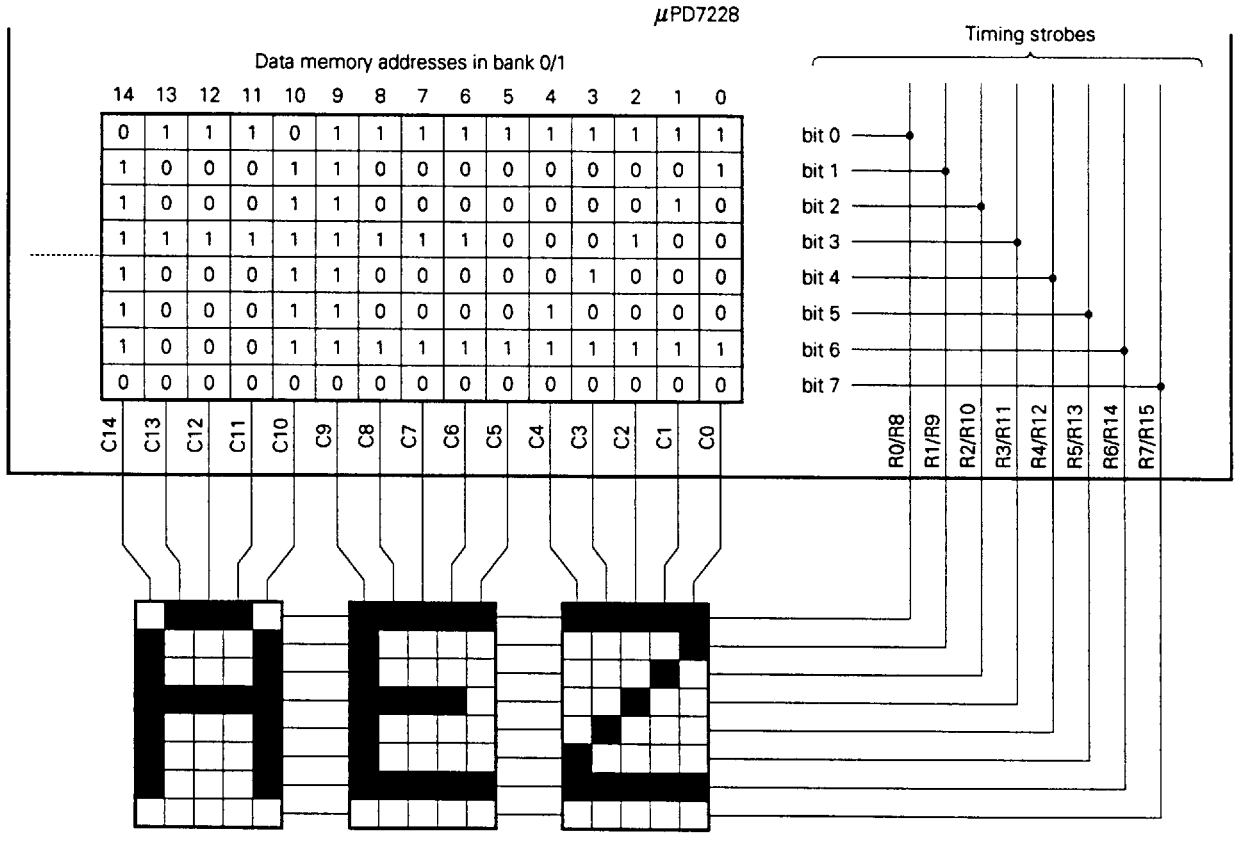
In this example, three digits of 5 × 7 (5 × 8) dot LCDs, data memory addresses 00H to 0EH (0 to 14), and column signals C0 to C14 are used.

Fig. 6-2 shows a timing chart corresponding to the character "A" displayed on columns C14 to C10 in Fig. 6-1. Fig. 6-3 shows the data memory contents and LCD display patterns to display the characters "ANZ" and numerals "8, 5" on a 16-time division LCD.

In this example, six digits of 5 × 7 (5 × 8) dot LCDs, addresses 00H to 0EH (0 to 14) in data memory banks 0 and 1, and column signals C0 to C14 are used.

Fig. 6-4 shows a timing chart corresponding to "A" and "8" displayed on the columns C14 to C10 in Fig. 6-3. In Fig. 6-2 and 6-4, when the difference between the row and column signals in waveform level ranges from $V_{LCD}$ AND $-V_{LCD}$, the LCD dot, corresponding to each waveform level, lights.

**Fig. 6-1 With 8-Time Division LCD**




**Note** Display data is read from either bank 0 or 1.

**Fig. 6-2 With 8-Time Division LCD**

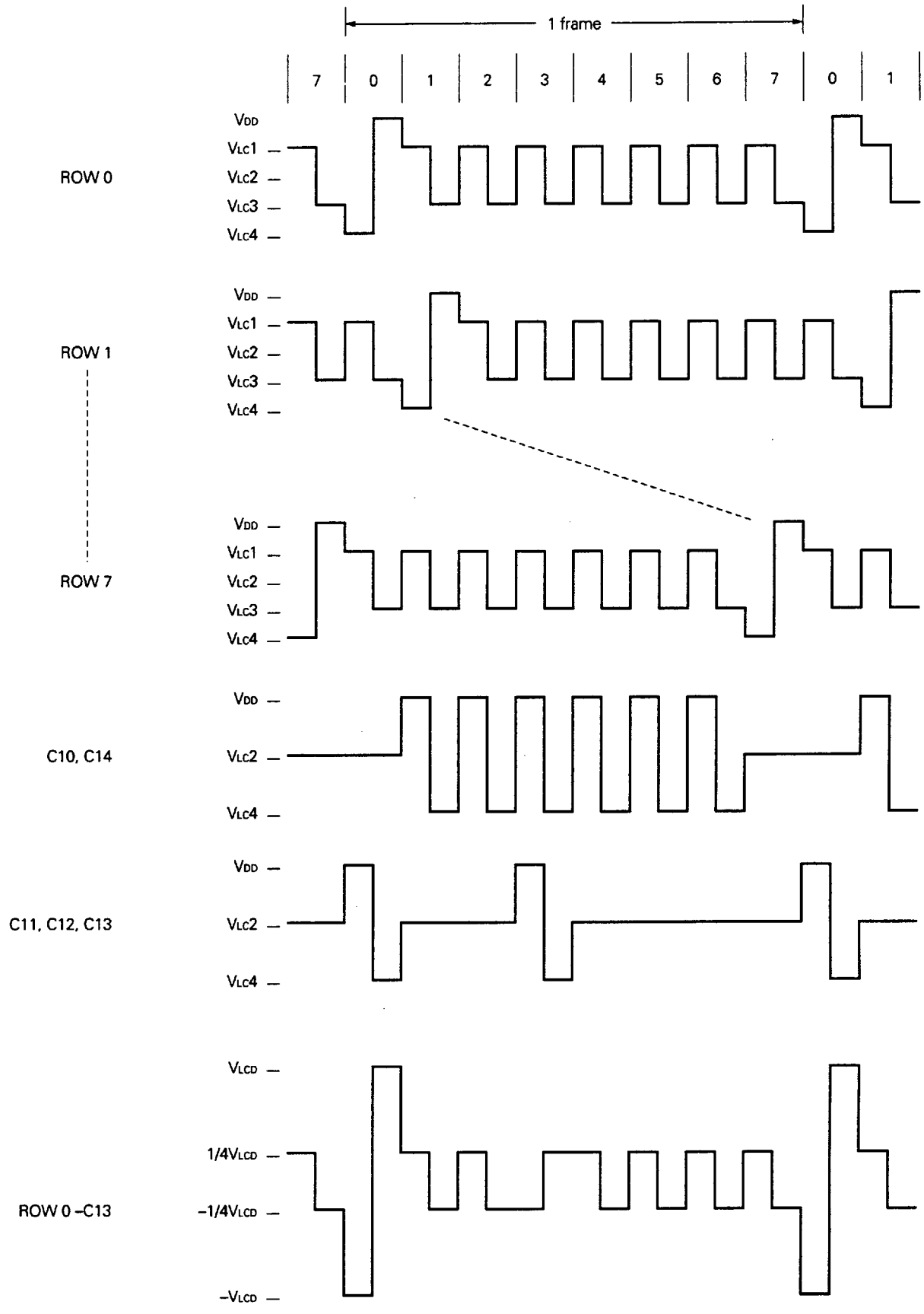


**Fig. 6-3 With 16-Time Division LCD**

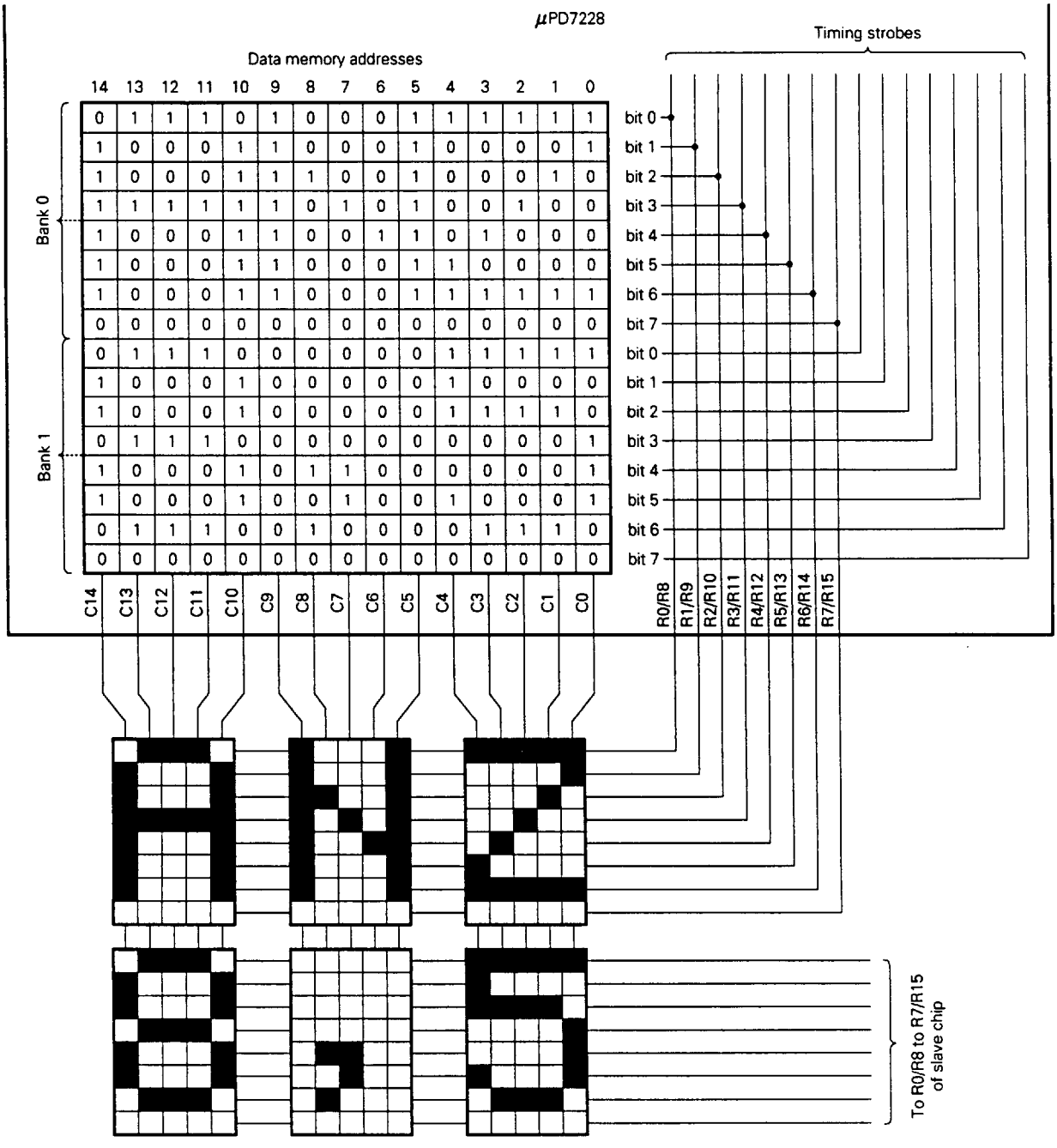
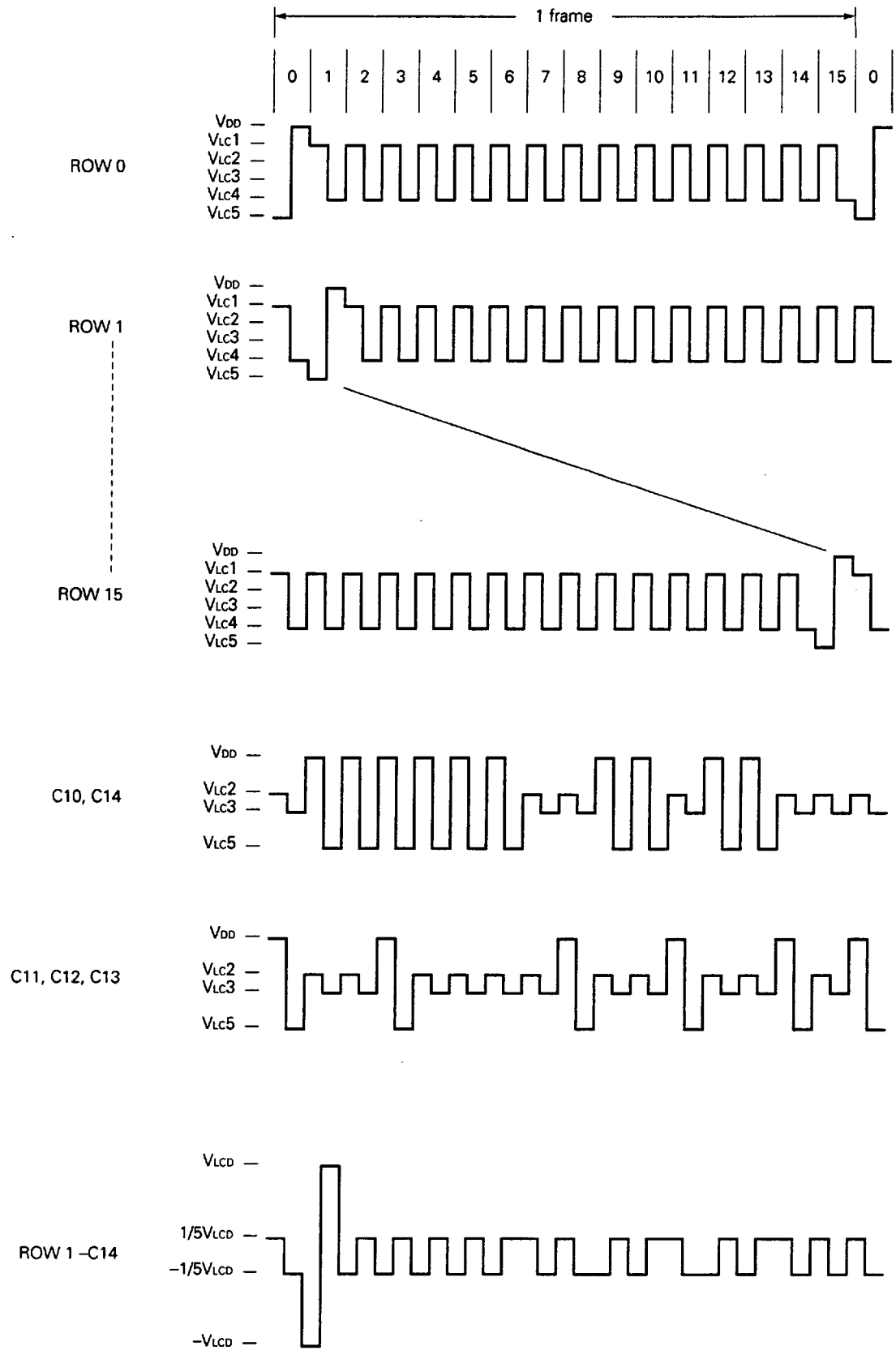


**Fig. 6-4 With 16-Time Division LCD**



## 7. STANDBY MODE

μPD7228 is provided with a standby mode, in which the power dissipation for μPD7228 is reduced.

This mode is set when the STOP command has been carried out and the $\overline{BUSY}$ signal has subsequently gone high. When set in the standby mode, μPD7228 internally masks the CLOCK signal to stop supply of clocks to the LCD timing control circuit and clock control circuit and initializes the data processing mode to the write mode in which autoincrement ($I_{1}I_{0}$ = 00) is accomplished. However, the other modes are not changed and, therefore, the interface mode and display mode are retained.

The standby mode is released when 1-byte data (command or data) has been input or when the RESET signal goes high, though the subsequent operations and processing, while in the standby mode, differ depending on whether the standby mode is released by input of 1-byte data or RESET signal.

During the standby mode, supply of the CLOCK signal to μPD7228 can also be stopped, in which case the power dissipation for μPD7228 can be reduced to a level lower than that when the CLOCK signal is internally masked by μPD7228.

### 7.1 RELEASING STANDBY MODE

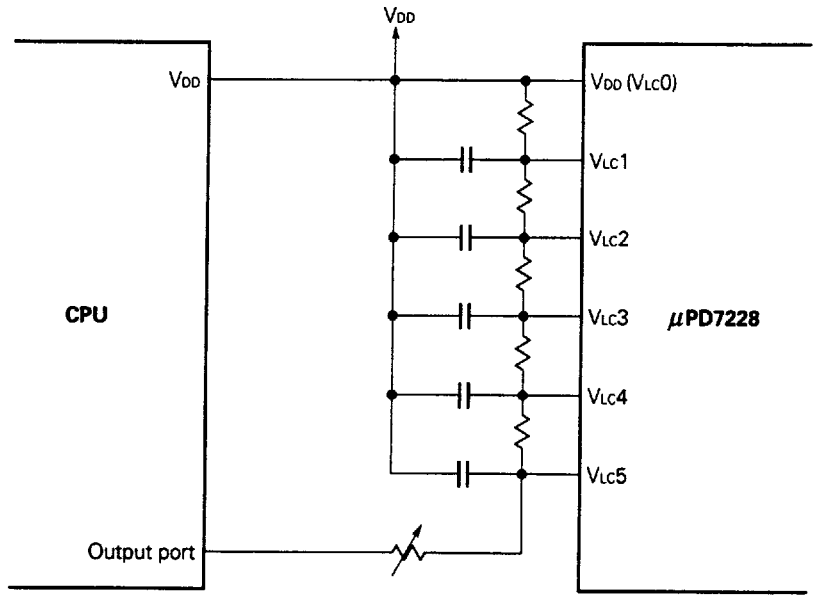
**(1) By writing 1-byte data**

To retain the operation modes (except the data processing mode) immediately before the standby mode, even after the standby mode has been released, release the standby mode by writing 1-byte data (command or data) to μPD7228. The standby mode is released when 8-bit serial data has been written in the serial interface mode (at the eighth rising edge of the $\overline{SCK}$ signal) or when the second 4-bit data has been written in the parallel interface mode (at the second rising edge of the $\overline{STB}$ signal).

However, if the chip address select function is used in the serial interface mode, and if the $\overline{CS}$ signal goes high during the standby mode, the first 8-bit data input after the $\overline{CS}$ signal has gone low is used as chip address information. Therefore, the standby mode is not released until the next 8-bit data is input.

**Note** In the standby mode, the supply of clocks necessary for AC driving of the LCD is stopped. As a result, the LCD drive signals are fixed to the voltage level immediately before the standby mode is set. Consequently, a DC voltage is continuously applied to the LCD. To prevent this, control the voltage on the $V_{LC5}$ pin by using an output port of the CPU, as shown in Fig. 7-1, to output a high-level signal from the output port immediately before the STOP command is implemented, so that no potential difference exists across the $V_{DD}$ and $V_{LC5}$ pins.

**Fig. 7-1 LCD Drive Voltage Control**




**Remarks** The power source for the CPU and μPD7228 must be the same.

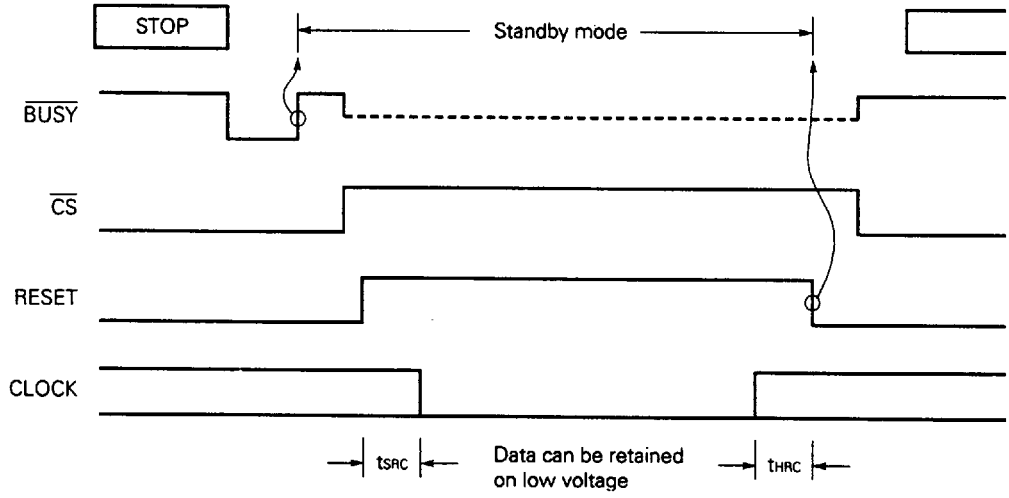
**(2) By RESET signal**

In a system where only the data memory contents have to be retained, when the standby mode has been released, the RESET signal can be used to release the standby mode. To use the RESET signal, the signal must be raised to the high level after the standby mode has been set (which can be confirmed by the high level of the $\overline{BUSY}$ signal). The standby mode is released by lowering the RESET signal. While the RESET signal is high, the LCD drive output pins are in the status same as when they are reset and apply no voltage to the LCD. In this case, however, unlike the ordinary reset operation, the data memory contents do not become undefined, when the RESET signal has been input, and the memory data immediately before the standby mode is set are retained during the standby mode. After the mode has been released, therefore, the data can be used. In a system employing this standby mode releasing method, the current path can be controlled using the RESET signal, as shown in Fig. 5-5.

### 7.2 STOPPING CLOCK SUPPLY AND RETAINING DATA ON LOW VOLTAGE IN STANDBY MODE

In the standby mode, only the data memory contents can be retained on a tiny amount of voltage. In this case, the power dissipation for μPD7228 can be reduced further by stopping the supply of the CLOCK signal to μPD7228. To stop the supply of the CLOCK signal in the standby mode, confirm that μPD7228 has been set in the standby mode (by checking whether the $\overline{BUSY}$ signal has gone high after the STOP command was carried out), raise the RESET signal to the high level, wait for a specified time, and then stop the supply of the CLOCK signal.

**Fig. 7-2 Stopping CLOCK Supply in Standby Mode**



## 8. RESET

When a high-level signal has been input to the RESET pin, μPD7228 is initialized as follows:

- The chip address comparison data (compared with the signals input to the CA1 and CA0 pins) are initialized to 00. Therefore, in multiple chip configuration, the $\overline{BUSY}$ signal operation differs, depending on whether or not the chip address coincides with the address information sent from the CPU, i.e., whether or not the CA1 and 0 signals for the chip are 00.

  Chip with CA1 and 0 = 00 : The $\overline{BUSY}$ signal goes low, if the $\overline{CS}$ signal is 0; if the $\overline{CS}$ signal is 1, the $\overline{BUSY}$ signal enters the high-impedance status.
  Other chips : The $\overline{BUSY}$ signal enters the high-impedance status independently from the input condition for the $\overline{CS}$ signal.

- All the processing operations (command/data processing, reading of timing signal, and display data to the row and column drivers) are stopped.
- A DC signal equivalent to the $V_{LC3}$ level is output to all the LCD drive signal output pins (C0 to C41, R15/C42 to R8/C49, R0/R8 to R7/R15).
- The internal functions are set in the status in which the following commands are implemented.

  SWM ($I_1I_0$ = 00) : autoincrement mode
  LDPI ($D_6$ to $D_0$ = 0000000) : data pointer is cleared to 0
  SMM ($M_2$ to $M_0$ = 000) : 8-time division. R0 to R7 functions of R0/R8 to R7/R15 pins. SYNC pin in input mode. Data memory bank 0
  SFF ($F_2$ to $F_0$ = 000) : frame frequency set to $f_{CL}/2^{14}$

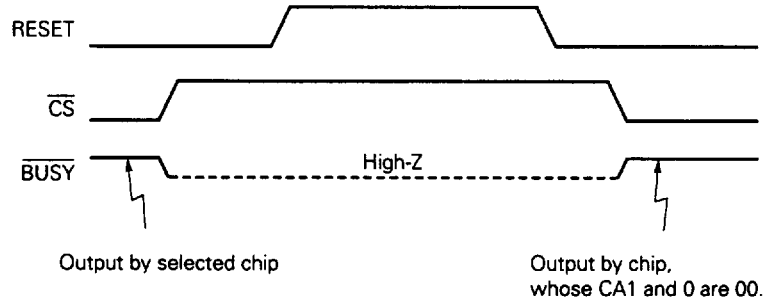
- The byte transfer end counter is cleared.
- If the standby mode has been set, it is retained.
- The data memory contents become undefined.

When the RESET signal is lowered, μPD7228 can start operating from the initialized status. Note that the following processing is accomplished when the RESET signal goes low, and that the display output is in the same status as when the DISP OFF command has been carried out.

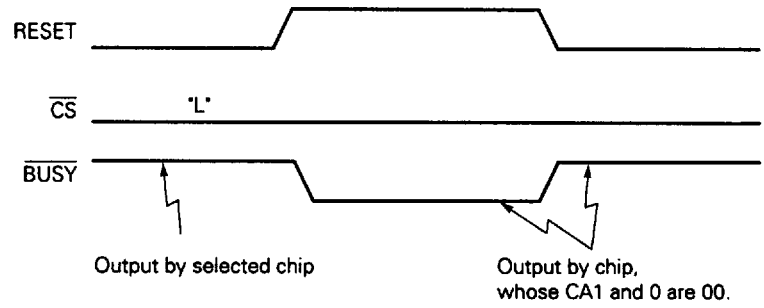
- An interface specification code (specifying parallel or serial mode and whether the chip address select function is enabled or disabled) is read from the DS/(CAE) and D1/(P/$\overline{S}$) pins.
- The chip, whose CA1 and 0 pins are both 0, is selected.
- If μPD7228 is reset while in the standby mode, the standby mode is released. In this case, the data memory contents are retained.

**Fig. 8-1 RESET Timing Example**

**(a)**




**(b)**



## 9. COMMANDS

μPD7228 is provided with the following 16 commands, all of which are 1 byte (8 bits) long.

**Table 9-1 Command List**

| Mnemonic | Operation | Hex. code |
|---|---|---|
| SFF | Set Frame Frequency | 10–14 |
| SMM | Set Multiplexing Mode | 18–1F |
| DISP OFF | Display Off | 08 |
| DISP ON | Display On | 09 |
| LDPI | Load Data Pointer with Immediate | 80–B1, C0–F1 |
| SRM | Set Read Mode | 60–63 |
| SWM | Set Write Mode | 64–67 |
| SORM | Set OR Mode | 68–6B |
| SANDM | Set AND Mode | 6C–6F |
| SCML | Set Character Mode with Left entry | 71 |
| SCMR | Set Character Mode with Right entry | 72 |
| BRESET | Bit Reset | 20–3F |
| BSET | Bit Set | 40–5F |
| CLCURS | Clear Cursor | 7C |
| WRCURS | Write Cursor | 7D |
| STOP | Set Stop Mode | 01 |

### 9.1 LCD DISPLAY MODE SETTING COMMANDS

These four types of LCD display mode setting commands are available:

SFF (Set Frame Frequency)
SMM (Set Multiplexing Mode)
DISP OFF (Display Off)
DISP ON (Display On)

#### (1) SFF (Set Frame Frequency)

| 0 | 0 | 0 | 1 | 0 | $F_2$ | $F_1$ | $F_0$ |
|---|---|---|---|---|---|---|---|

Set a frame frequency, which is the frequency of the clock input from the CLOCK pin and divided by a ratio specified by bits $F_2$ to $F_0$ of this command.

| $F_2$ | $F_1$ | $F_0$ | Frame frequency |
|---|---|---|---|
| 0 | 0 | 0 | $f_{CL}/2^{14}$ |
| 0 | 0 | 1 | $f_{CL}/2^{13}$ |
| 0 | 1 | 0 | $f_{CL}/2^{12}$ |
| 0 | 1 | 1 | $f_{CL}/2^{11}$ |
| 1 | 0 | 0 | $f_{CL}/2^{10}$ |
| 1 | 0 | 1 | Inhibited |
| 1 | 1 | 1 | |

$f_{CL}$: clock frequency

**(2) SMM (Set Multiplexing Mode)**

| 0 | 0 | 0 | 1 | 1 | $M_2$ | $M_1$ | $M_0$ |
|---|---|---|---|---|---|---|---|

Specifies the number of rows, selects the functions of the row and row/column drivers, sets the SYNC pin in input or output mode, and specifies a data memory bank.

| $M_2$ | $M_1$ | $M_0$ | Number of rows | R0/R8-R7-R15 | R15/C42-R8/C49 | SYNC pin | Memory bank |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 8 | R0-R7 | C42-C49 | Input | 0 |
| 0 | 0 | 1 | | | | | 1 |
| 0 | 1 | 0 | | | | Output | 0 |
| 0 | 1 | 1 | | | | | 1 |
| 1 | 0 | 0 | 16 | R8-R15 | | Input | 0, 1 |
| 1 | 0 | 1 | | | | | |
| 1 | 1 | 0 | | R0-R7 | R15-R8 | Output | |
| 1 | 1 | 1 | | | | | |

**(3) DISP OFF (Display Off)**

| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|

Sets the row and column signal at a voltage level at which they are not selected and turns off the display. The effect of this command is independent from the displayed data.

**(4) DISP ON (Display On)**

| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|---|

Accomplishes display operation in accordance with the displayed data.

## 9.2 DATA POINTER LOAD COMMAND

### (1) LDPI (Load Data Pointer with Immediate)

| 1 | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
|---|---|---|---|---|---|---|---|

$D_{6-0}$ = 00H-31H, 40H-71H

Loads 7 bits of immediate data (bits $D_6$ to $D_0$) to the data pointer.

## 9.3 DATA PROCESSING MODE SETTING COMMANDS

The following six types of data processing mode setting commands are provided:

| | |
|---|---|
| SRM | (Set Read Mode) |
| SWM | (Set Write Mode) |
| SORM | (Set OR Mode) |
| SANDM | (Set AND Mode) |
| SCML | (Set Character Mode with Left entry) |
| SCMR | (Set Character Mode with Right entry) |

By writing any of these commands, μPD7228 is set in the mode specified by the command and processes data in the set mode, until the mode is changed by another data processing mode setting command.

The lower 2 bits ($I_1$ and $I_0$) for each data processing mode command specifies how the data pointer should be modified, each time byte data is processed, as shown in the table below.

| $I_1$ | $I_0$ | Data pointer modification |
|---|---|---|
| 0 | 0 | Incremented by 1 each time byte data processed (autoincrement) |
| 0 | 1 | Decremented by 1 each time byte data is processed (autodecrement) |
| 1 | 0 | Inhibited |
| 1 | 1 | Not modified (same address is retained) |

### (1) SRM (Set Read Mode)

| 0 | 1 | 1 | 0 | 0 | 0 | $I_1$ | $I_0$ |
|---|---|---|---|---|---|---|---|

Sets the read mode.

In this mode, the data memory contents addressed by the data pointer are automatically transferred to the serial/parallel register. After that, the data pointer contents are modified as specified by bits $I_1$ and $I_0$ of this command. When all the 8-bit contents of the serial/parallel register have been read by the CPU, the next data memory contents addressed by the modified contents of the data pointer are automatically transferred to the serial/parallel register. In this way, the contents of the data pointer and serial/parallel register are updated each time the CPU reads 8-bit data from the serial/parallel register.

**(2) SWM (Set Write Mode)**

| 0 | 1 | 1 | 0 | 0 | 1 | $I_1$ | $I_0$ |
|---|---|---|---|---|---|---|---|

Sets the write mode.
In this mode, the 8-bit data, written by the CPU to the serial/parallel register, is stored in the data memory addressed by the contents of the data pointer. After that, the contents of the data pointer are modified as specified by bits $I_1$ and $I_0$ of this command. In the same manner, the contents of the data pointer and serial/parallel register are updated each time the CPU writes 8-bit data to the serial/parallel register.

**(3) SORM (Set OR Mode)**

| 0 | 1 | 1 | 0 | 1 | 0 | $I_1$ | $I_0$ |
|---|---|---|---|---|---|---|---|

Sets the OR mode.
In this mode, the 8-bit data written by the CPU to the serial/parallel register are ORed with the data memory contents addressed by the data pointer contents and the OR operation result is stored in the same data memory address. After that, the data pointer contents are modified as specified by bits $I_1$ and $I_0$ of this command. In the same manner, the contents for the data pointer and serial/parallel register are updated each time the CPU writes 8-bit data to the serial/parallel register.

**(4) SANDM (Set AND Mode)**

| 0 | 1 | 1 | 0 | 1 | 1 | $I_1$ | $I_0$ |
|---|---|---|---|---|---|---|---|

Sets the AND mode.
In this mode, the 8-bit data written by the CPU to the serial/parallel register are ANDed with the data memory contents addressed by the data pointer contents and the AND operation result is stored in the same data memory address. After that, the data pointer contents are modified, as specified by bits $I_1$ and $I_0$ for this command. In the same manner, the contents for the data pointer and serial/parallel register are updated, each time the CPU writes 8-bit data to the serial/parallel register.

**(5) SCML (Set Character Mode with Left entry)**

| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|---|

Set the left entry character mode.
In this mode, the 8-bit data written by the CPU to the serial/parallel register is decoded by the character generator into an ASCII or JIS code, consisting of 5 by 7 bits, and is written to five contiguous data memory addresses. The highest address of these five addresses is indicated by the data pointer contents. As a result, the data pointer contents are decremented by five. In the same manner, the contents for the data pointer and serial/parallel register are updated, each time the CPU writes 8-bit data to the serial/parallel register.

**(6) SCML (Set Character Mode with Right entry)**

| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|---|

Set the right entry character mode.
In this mode, the 8-bit data written by the CPU to the serial/parallel register is decoded by the character generator into an ASCII or JIS code, consisting of 5 by 7 bits, and is written to five contiguous data memory addresses. The lowest address of these five addresses is indicated by the data pointer contents. As a result, the data pointer contents are incremented by five. In the same manner, the contents for the data pointer and serial/parallel register are updated, each time the CPU writes 8-bit data to the serial/parallel register.

## 9.4 MEMORY BIT MANIPULATION COMMANDS

The following four types of memory bit manipulation commands are available:

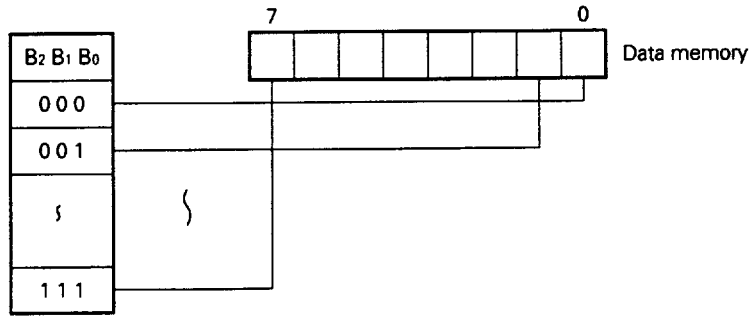
BRESET (Bit Reset)
BSET (Bit set)
CLCURS (Clear Cursor)
WRCURS (Write Cursor)

The BRESET and BSET commands can be accomplished in any data processing mode, and resets and sets a specified bit of the data memory addressed by the data pointer. The bit to be reset or set is specified by the BRESET or BSET command. After the specified bit has been reset or set, the data pointer contents are modified as specified by the lower 2 bits ($J_1$ and $J_0$) of the command byte. The CLCURS and WRCURS commands are used to clear and display the cursor (bit 7 position). When these commands have been executed, the contents of the data pointer are incremented or decremented by five.

The data pointer contents can be manipulated only when any of these commands is executed. After the command has been carried out the data pointer contents are modified, in accordance with the previously set data processing mode.

However, since the data pointer contents have already been modified by the bit manipulation command, further modification of the data pointer contents starts from the value set by the command.

The command bits of the BRESET and BSET commands, $B_2$ to $B_1$, specify a data memory bit as follows:



| $J_1$ | $J_0$ | Data pointer modification |
|---|---|---|
| 0 | 0 | Incremented by 1 |
| 0 | 1 | Decremented by 1 |
| 1 | 0 | Inhibited |
| 1 | 1 | Not modified (the same address is retained) |

**(1) BRESET (Bit Reset)**

| 0 | 0 | 1 | $B_2$ | $B_1$ | $B_0$ | $J_1$ | $J_0$ |
|---|---|---|---|---|---|---|---|

Resets to 0 a bit specified by the bits $B_2$ to $B_0$ of this command in the data memory addressed by the data pointer. After that, the data pointer contents are modified in accordance with bits $J_1$ and $J_0$.

**(2) BSET (Bit Set)**

| 0 | 1 | 0 | $B_2$ | $B_1$ | $B_0$ | $J_1$ | $J_0$ |
|---|---|---|---|---|---|---|---|

Sets to 1 a bit specified by the bits $B_2$ to $B_0$ of this command in the data memory addressed by the data pointer. After that, the data pointer contents are modified in accordance with bits $J_1$ and $J_0$.

**(3) CLCURS (Clear Cursor)**

| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
|---|---|---|---|---|---|---|---|

When this command is carried out in the character mode, only bit 7 of a data memory address higher (SCMR mode) or lower (SCML mode) than the address indicated by the data pointer contents by five addresses is reset to 0.
This command can be used to clear the cursor for display characters consisting of 5 by 7 dots.

**(4) WRCURS (Write Cursor)**

| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
|---|---|---|---|---|---|---|---|

When this command is carried out in the character mode, only the bit 7 of a data memory address higher (SCMR mode) or lower (SCML mode) that the address indicated by the data pointer contents by five addresses is set to 1. This command can be used to display the cursor for display characters consisting of 5 by 7 dots.

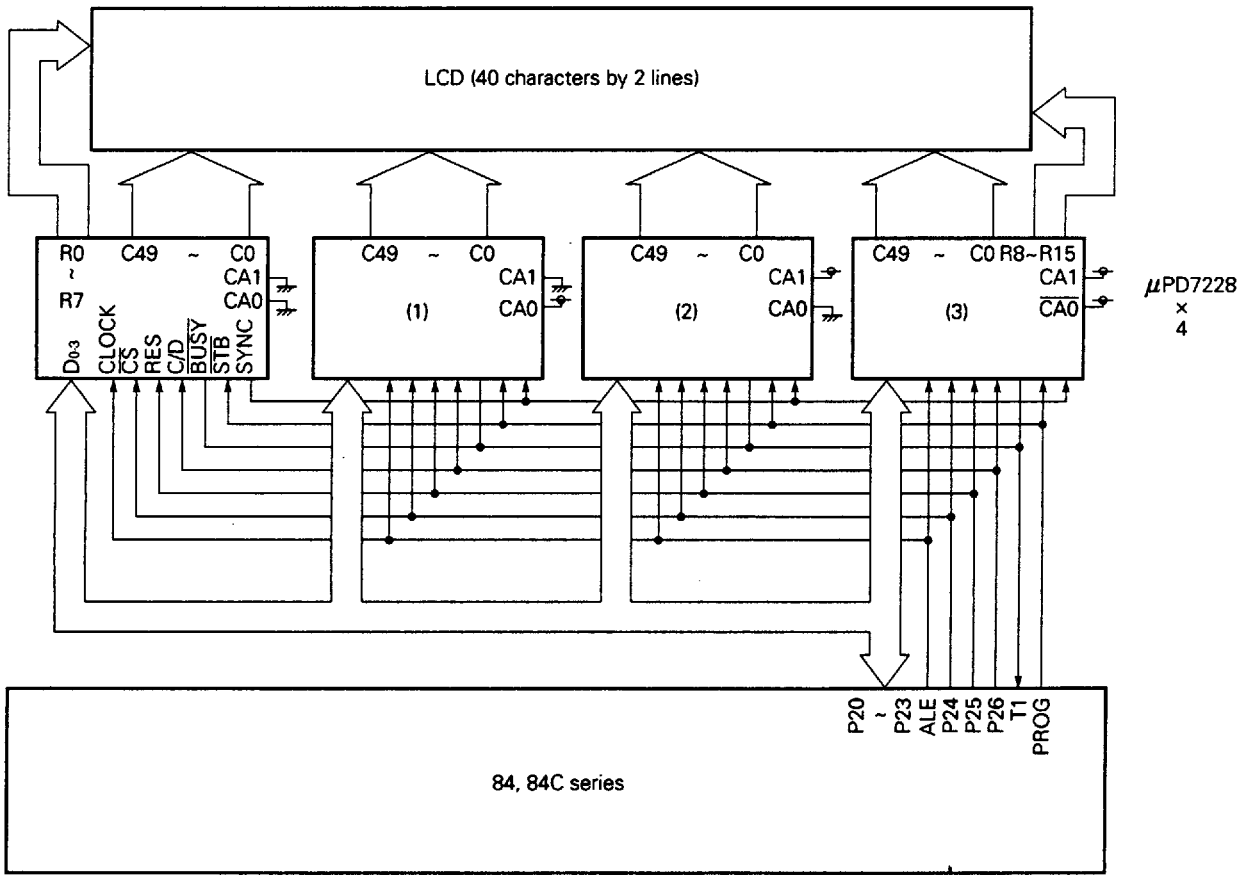
### 9.5 STANDBY MODE SETTING COMMAND

#### (1) STOP (Set Stop Mode)

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|---|

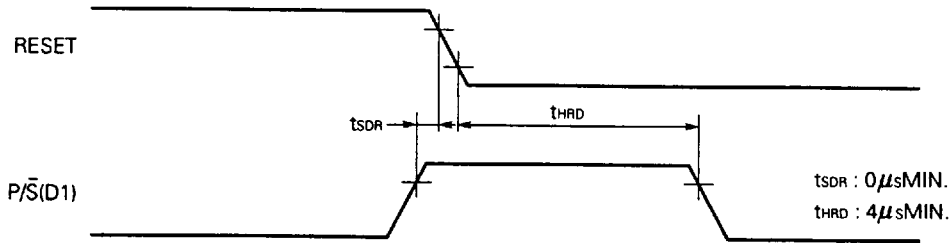
Sets the STOP mode.
The data processing mode is initialized to the write mode of the autoincrement mode ($I_1$ and $I_0$ = 00) when this command has been carried out. The other modes are not affected.

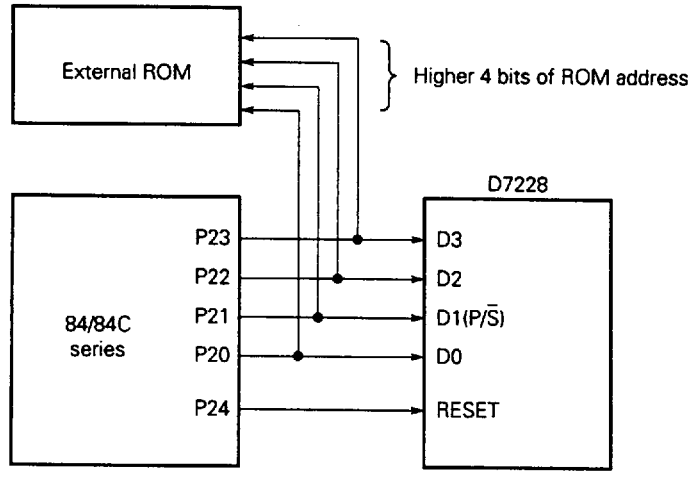
System Configuration Example (Reference)



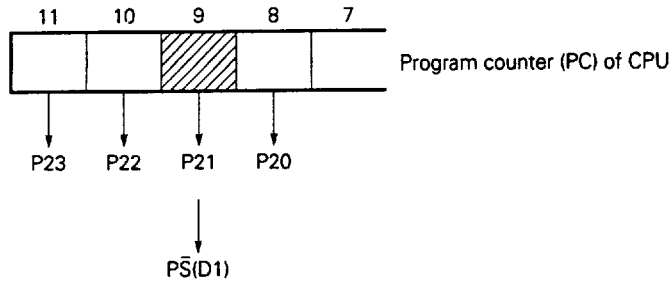
Notes on Interfacing with 84/84C Series CPUs

The interface mode specification data for μPD7228 (P/S signal) must be input to μPD7228 in synchronization with the falling edge of the RESET signal (when the RESET signal is released) and kept at the specified level (high in the parallel interface mode) for at least 4 microseconds after the RESET signal has fallen. Therefore, in a system using an ROM-less product of 84/84C series, or a system using an external ROM with an ROM integrated product of 84/84C series, particular attention must be paid to interfacing these CPUs with μPD7228.




In these systems, data are transferred between the CPU and μPD7228 through the lower 4 bits of the port 2 (P23 to P20 pins) of the CPU, and the higher 4 bits (PCH: PC11 to PC8) of the external ROM address are output to the same pins by means of time division. Therefore, if accessing the external ROM starts immediately after the RESET signal has been released by the port 2 output instruction, the bit of the higher 4 ROM address bits corresponding to the P/S pin μPD7228 (P21 = D1) must be kept at the high level for at least 4 microseconds.




For example, when μPD8039 (that can access up to 4K bytes of external ROM) is used, a port 2 output instruction that releases the RESET signal for μPD7228 and outputs a high-level signal to the P/$\overline{S}$ pin of μPD7228 must be implemented. When this instruction has been accomplished and before 4 microseconds have elapsed after that, instructions that may cause program implementation to jump from the following address ranges (such as jump, call, and return instructions) must not be carried out. In addition, a port 2 output instruction, that changes the level of the P/$\overline{S}$ pin, must not be carried out, either.

200H to 3FFH, 600H to 7FFH
A00H to BFFH, E00H to FFFH

## 10. ELECTRICAL CHARACTERISTICS

### ABSOLUTE MAXIMUM RATINGS (Ta = 25 °C)

| Item | Symbol | Conditions | Ratings | Unit |
|---|---|---|---|---|
| Supply Voltage | $V_{DD}$ | | −0.3 to +7.0 | V |
| Input Voltage | $V_I$ | | −0.3 to $V_{DD}$ +0.3 | V |
| Output Voltage | $V_O$ | | −0.3 to $V_{DD}$ +0.3 | V |
| Operating Temperature | $V_{opt}$ | | −10 to +70 | °C |
| Storage Temperature | $V_{stg}$ | | −65 to +150 | °C |

### DC CHARACTERISTICS (Ta = −10 to +70 °C, $V_{DD}$ = 5 V ± 10 %)

| Item | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| High-Level Input Voltage | $V_{IH1}$ | Except $\overline{SCK}$ | 0.7 $V_{DD}$ | | $V_{DD}$ | V |
| | $V_{IH2}$ | $\overline{SCK}$ | 0.8 $V_{DD}$ | | $V_{DD}$ | V |
| Low-Level Input Voltage | $V_{IL}$ | | 0 | | 0.3 $V_{DD}$ | V |
| High-Level Input Leakage Current | $V_{LIH}$ | VI = $V_{DD}$ | | | 10 | μA |
| Low-Level Input Leakage Current | $V_{LIL}$ | VI = 0 V | | | −10 | μA |
| High-Level Output Voltage | $V_{OH1}$ | $\overline{BUSY}$, D0 – D3<br>$I_{OH}$ = −400 μA | $V_{DD}$ −0.5 | | | V |
| | $V_{OH2}$ | SYNC,<br>$I_{OH}$ = −100 μA | $V_{DD}$ −0.5 | | | V |
| Low-Level Output Voltage | $V_{OL1}$ | $\overline{BUSY}$, D0 – D3<br>$I_{OL}$ = 1.7 mA | | | 0.45 | V |
| | $V_{OL2}$ | SYNC,<br>$I_{OL}$ = 100 μA | | | 0.45 | V |
| High-Level Output Leakage Current | $I_{LOH}$ | $V_O$ = $V_{DD}$ | | | 10 | μA |
| Low-Level Output Leakage Current | $I_{LOL}$ | $V_I$ = 0 V | | | −10 | μA |
| LCD Drive Voltage | $V_{LCD}$ | | 3.0 | | $V_{DD}$ | V |
| Row Output Impedance | $R_{ROW}$ | | | 4 | 8 | kΩ |
| Row/Column Output Impedance | $R_{ROW/COL}$ | | | 5 | 10 | kΩ |
| Column Output Impedance | $R_{COL}$ | | | 10 | 15 | kΩ |
| Supply Current | $I_{DD1}$ | Operation mode,<br>$f_c$ = 400 kHz | | 200 | 400 | μA |
| | $I_{DD2}$ | STOP mode,<br>CLK = 0 V | | | 20 | μA |

**CAPACITANCE (Ta = 25 °C, $V_{DD}$ = 0 V)**

| Item | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Input Capacitance | $C_{IN}$ | f = 1 MHz with pins other than that measured at 0 V | | | 10 | pF |
| Output Capacitance | $C_{OUT}$ | | | | 25 | pF |
| Input/Output Capacitance | $C_{IO}$ | | | | 15 | pF |

**AC CHARACTERISTICS (Ta = −10 to +70 °C, $V_{DD}$ = +5 V ±10 %)**

**Common Operations:**

| Item | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Clock Operation Frequency | fc | | 100 | | 1100 | kHz |
| High-Level Clock Pulse Width | $t_{WHC}$ | | 350 | | | ns |
| Low-Level Clock Pulse Width | $t_{WLC}$ | | 350 | | | ns |
| High-Level $\overline{\text{RESET}}$ Width | $t_{HRS}$ | | 4 | | | μs |
| $\overline{\text{CS}}$ ↓ → $\overline{\text{BUSY}}$ Delay Time | $t_{DCSB}$ | $C_L$ = 50 pF | | | 2 | μs |
| $\overline{\text{CS}}$ ↑ → $\overline{\text{BUSY}}$ Float Delay Time | $t_{DCSBF}$ | $C_L$ = 50 pF | | | 4 | μs |
| High-Level $\overline{\text{CS}}$ Width | $t_{WHCS}$ | | 4 | | | μs |
| SYNC Load Capacitance | $C_{LSY}$ | | | | 100 | pF |
| Data Set Time (vs. RESET ↓) | $t_{SDR}$ | | 0 | | | μs |
| Data Hold Time (vs. RESET ↓) | $t_{HRD}$ | | 4 | | | μs |

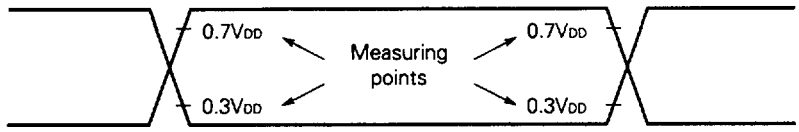
**Serial Input/Output Operations:**

| Item | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| $\overline{\text{SCK}}$ Cylce | $t_{CYK}$ | • | 0.9 | | | μs |
| High-Level $\overline{\text{SCK}}$ Pulse Width | $t_{WHK}$ | | 400 | | | ns |
| Low-Level $\overline{\text{SCK}}$ Pulse Width | $t_{WLK}$ | | 400 | | | ns |
| High-Level $\overline{\text{SCK}}$ Hold Time (vs. $\overline{\text{BUSY}}$ ↑) | $t_{HBK}$ | | 0 | | | ns |
| SI Set Time (vs. $\overline{\text{SCK}}$ ↑) | $t_{SIK}$ | | 100 | | | ns |
| SI Hold Time (vs. $\overline{\text{SCK}}$ ↑) | $t_{HKI}$ | | 250 | | | ns |
| $\overline{\text{SCK}}$ ↓ → SO Delay Time | $t_{DKO}$ | $C_L$ = 50 pF | | | 320 | ns |
| 8th $\overline{\text{SCK}}$ ↑ → $\overline{\text{BUSY}}$ Delay Time | $t_{DKB}$ | $C_L$ = 50 pF | | | 3 | μs |
| Low-Level $\overline{\text{BUSY}}$ Time | $t_{WLB}$ | $C_L$ = 50 pF | 18 | | 64 | 1/fc |
| C/$\overline{\text{D}}$ Set Time (vs. 1st $\overline{\text{SCK}}$ ↑) | $t_{SDK}$ | | 0 | | | μs |
| C/$\overline{\text{D}}$ Hold Time (vs. 8th $\overline{\text{SCK}}$ ↑) | $t_{HKD}$ | | 2 | | | μs |
| $\overline{\text{CS}}$ Hold Time (vs. 8th $\overline{\text{SCK}}$ ↑) | $t_{HKCS}$ | | 2 | | | μs |

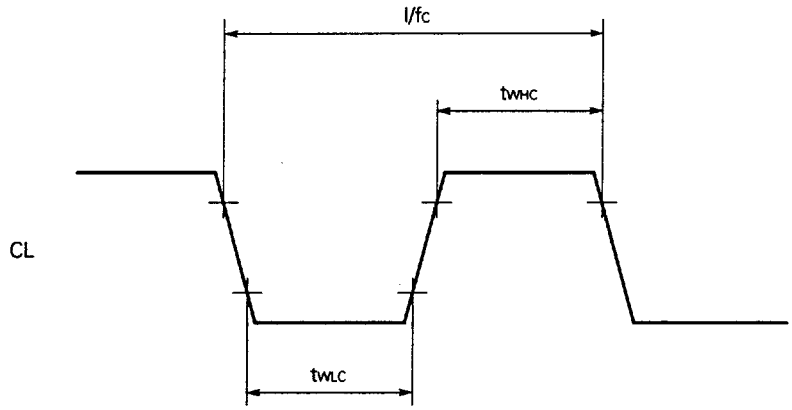
**Parallel Input/Output Operations:**

| Item | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Command Input Set Time (vs. $\overline{STB}$ ↓) | $t_A$ | $C_L$ = 80 pF | 100 | | | ns |
| Command Input Hold Time (vs. $\overline{STB}$ ↓) | $t_B$ | $C_L$ = 20 pF | 90 | | | ns |
| Data Input Set Time (vs. $\overline{STB}$ ↑) | $t_C$ | $C_L$ = 80 pF | 230 | | | ns |
| Data Input Hold Time (vs. $\overline{STB}$ ↑) | $t_D$ | $C_L$ = 20 pF | 50 | | | ns |
| Data Output Delay Time | $t_{ACC}$ | $C_L$ = 80 pF | 90 | | 650 | ns |
| Data Output Hold Time | $t_H$ | $C_L$ = 20 pF | 0 | | 150 | ns |
| $\overline{STB}$ Pulse Width | $t_{SL}$ | | 700 | | | ns |
| High-Level $\overline{STB}$ Time | $t_{SH}$ | | 1 | | | μs |
| High-Level $\overline{STB}$ Hold Time (vs. $\overline{BUSY}$ ↑) | $t_{HBS}$ | | 0 | | | μs |
| 2nd $\overline{STB}$ ↑ → $\overline{BUSY}$ Delay Time | $t_{DSB}$ | | | | 3 | μs |
| Low-Level $\overline{BUSY}$ Time | $t_{WLB}$ | $C_L$ = 50 pF | 18 | | 64 | 1/fc |
| C/$\overline{D}$ Set Time (vs. 1st $\overline{STB}$ ↓) | $t_{SDS}$ | | 0 | | | μs |
| C/$\overline{D}$ Hold Time (vs. 2nd $\overline{STB}$ ↑) | $t_{HSD}$ | | 2 | | | μs |
| $\overline{CS}$ Hold Time (vs. 2nd $\overline{STB}$ ↑) | $t_{HSCS}$ | | 2 | | | μs |

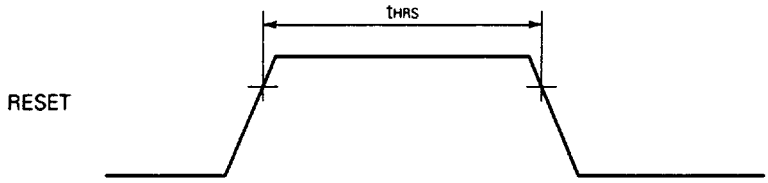
### AC Timing Measured Voltage (excluding STB/SCK and BUSY)



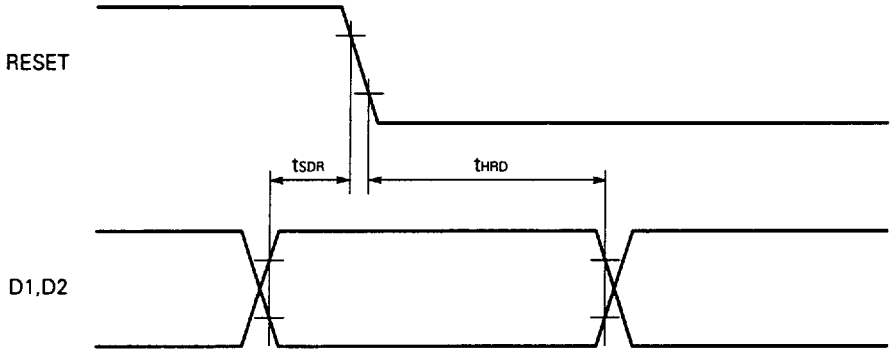

### Clock Timing



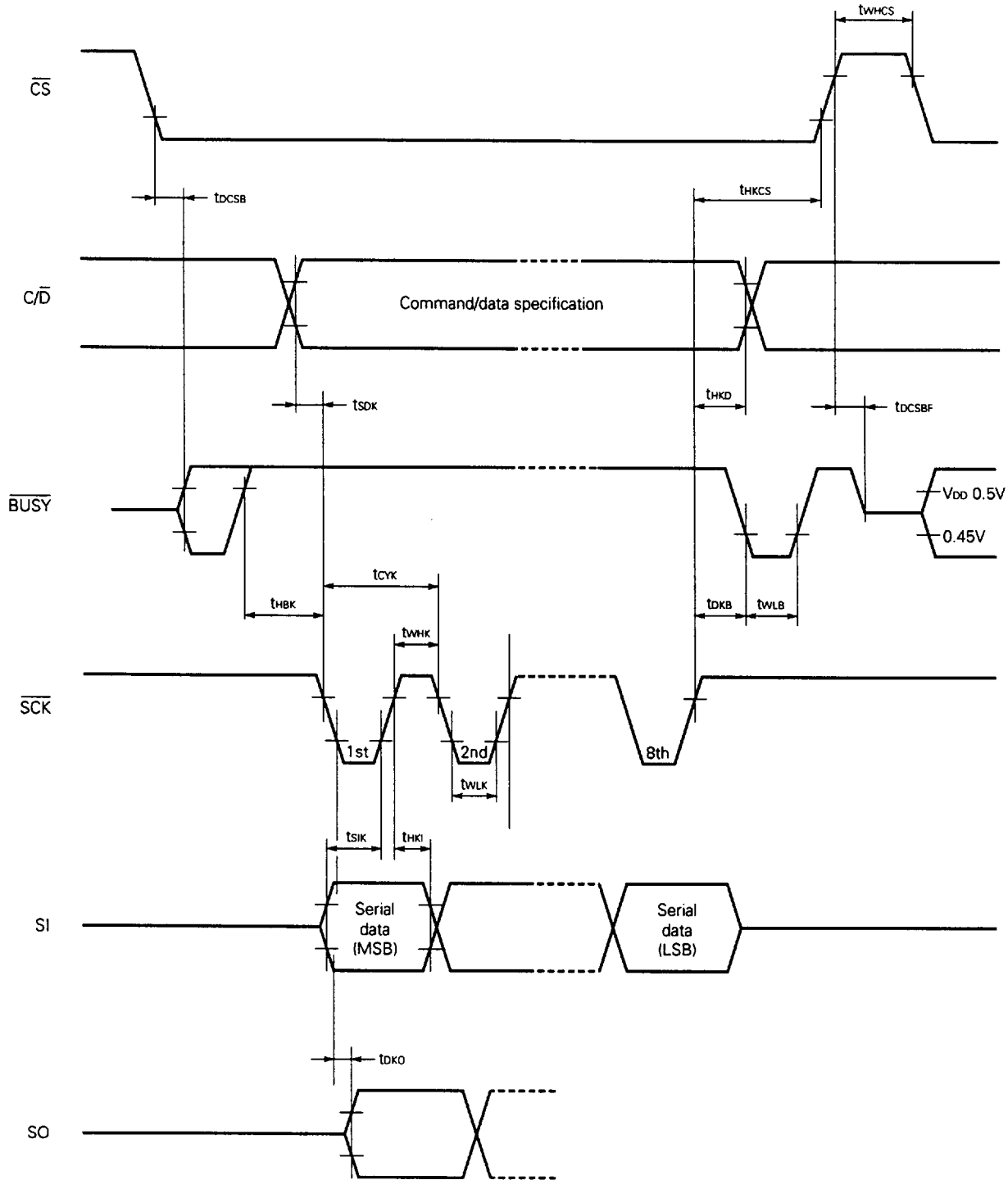

### RESET Input Timing



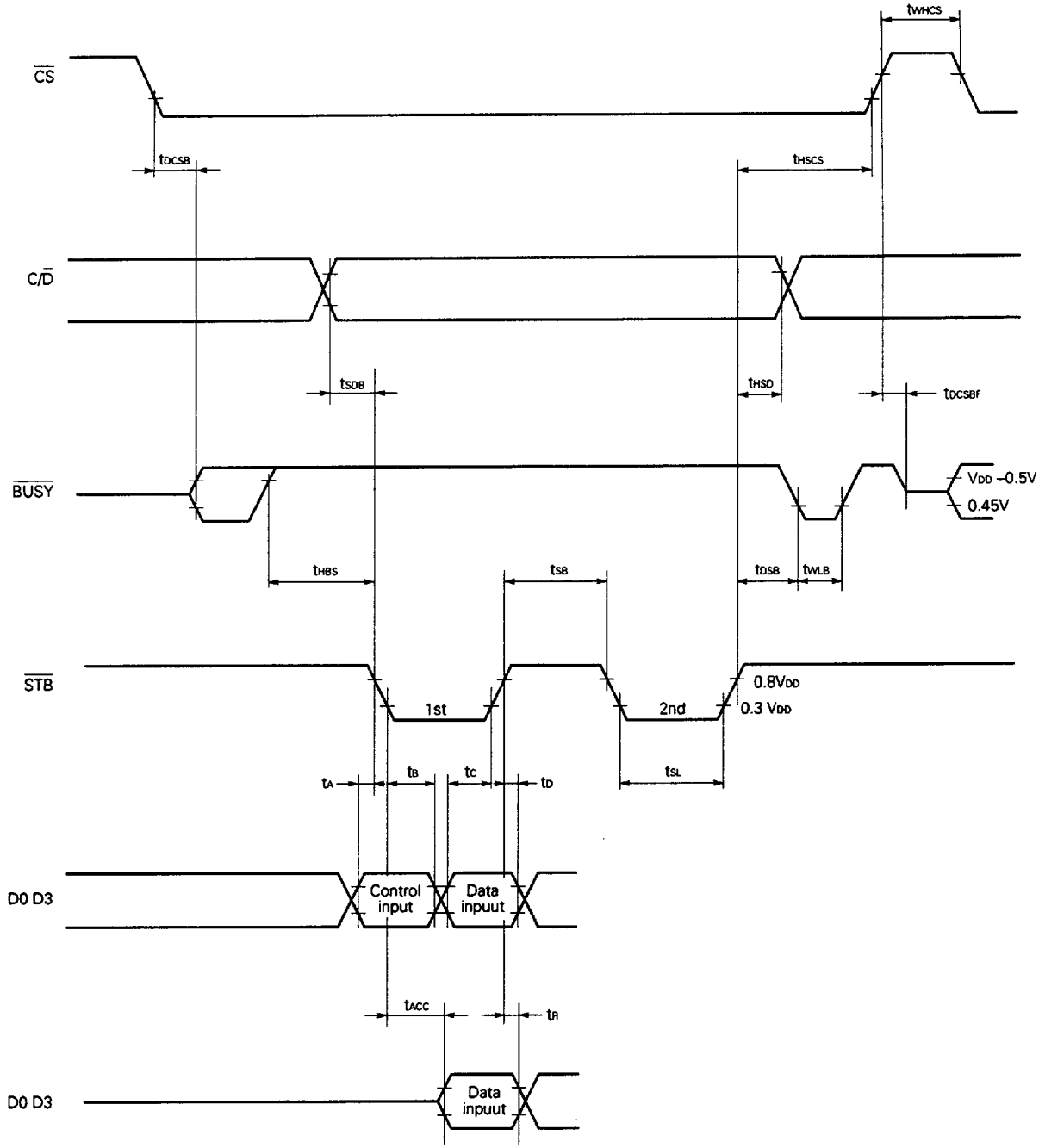

### Interface Mode Specification Timing



**Serial Input/Output Timing**



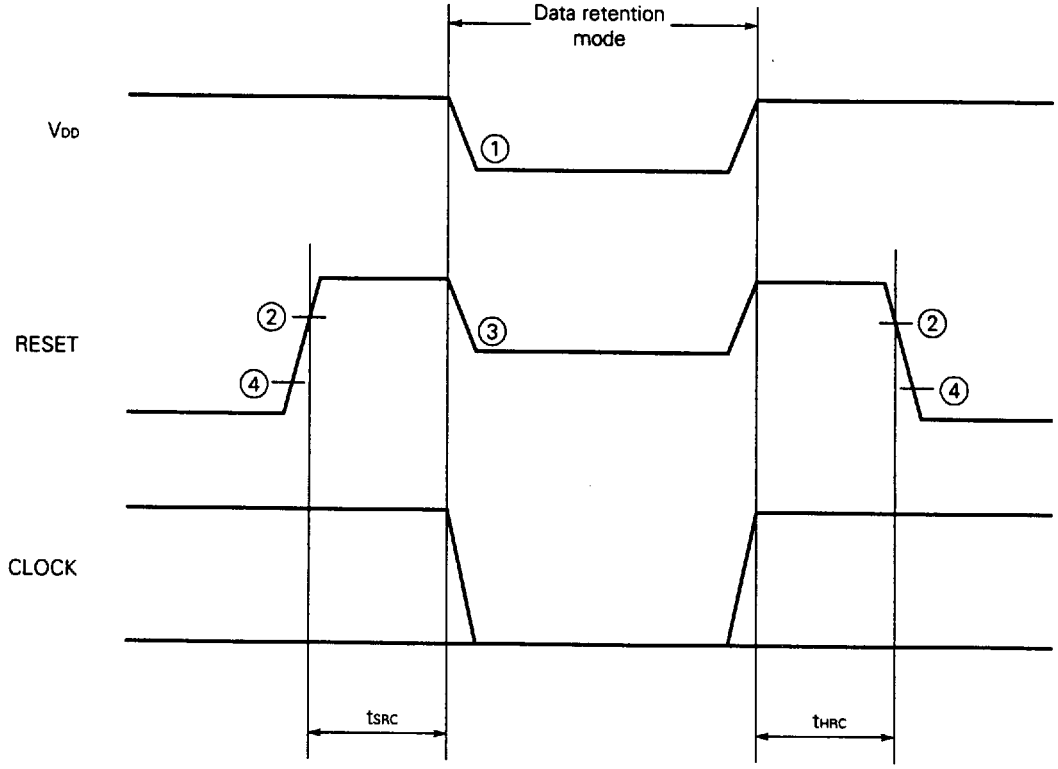
**Parallel Output Timing**



**DATA RETENTION CHARACTERISTICS FOR DATA MEMORY IS STOP MODE AND ON LOW VOLTAGE**
**(Ta = −10 to +70 °C)**

| Item | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Data Retention Supply Voltage | $V_{DDDR}$ | | 2.0 | | | V |
| Data Retention Supply Current | $I_{DDDR}$ | $V_{DDDR}$ = 2.0 V | | | 20 | μA |
| Data Retention High-Level RESET Input Voltage | $V_{IHDR}$ | | 0.9 $V_{DDDR}$ | | $V_{DDDR}$ +0.2 | V |
| RESET & CLOCK Setup Time | $t_{SRC}$ | | 10 | | | μs |
| RESET & CLOCK Hold Time | $t_{HRC}$ | | 10 | | | μs |

**Data Retention Timing**




① $V_{DDDR}$
② $V_{IH1}$
③ $V_{IH1DR}$
④ $V_{IL}$

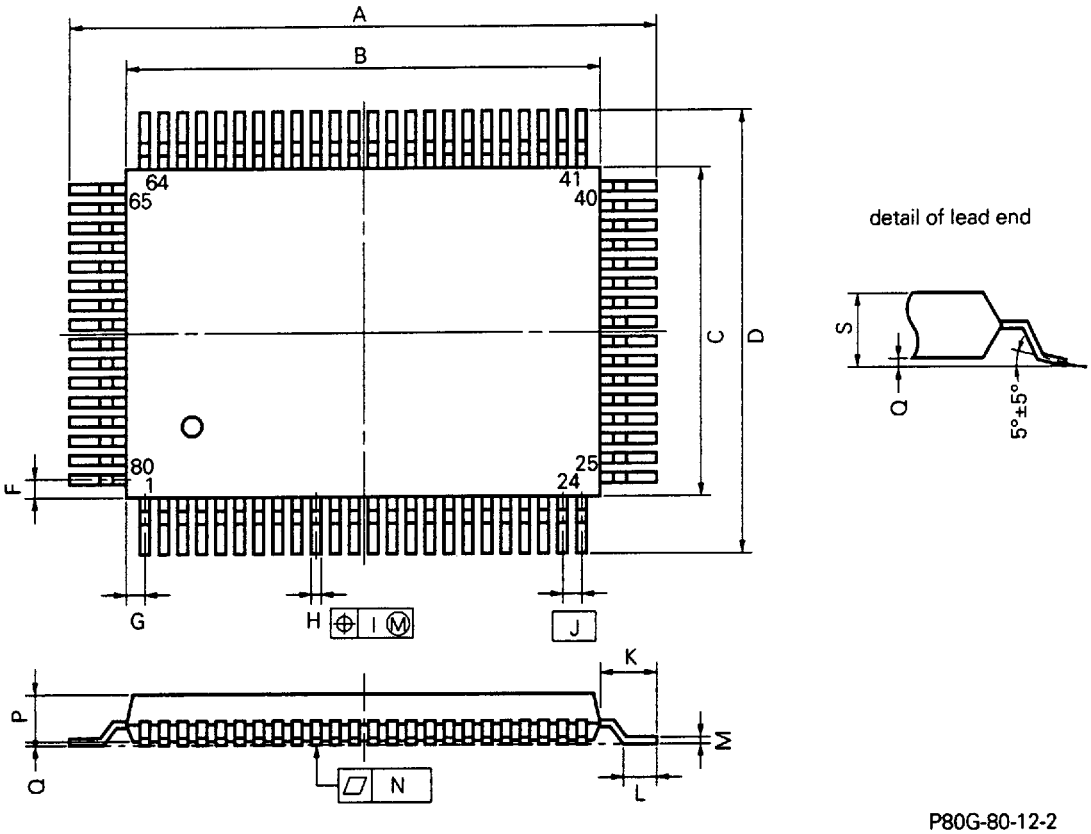
**Note** Keep all the input levels to $V_{DDDR}$ or less in the data retention mode.

## 11. REFERENCE MATERIAL

- SEMICONDUCTOR DEVICE PACKAGE MANUAL (IEI-1213)
- NEC SEMICONDUCTOR DEVICE RELIABILITY/QUALITY (IEI-1212)

## 12. PACKAGE DRAWING

### 80 PIN PLASTIC QFP (14x20)




P80G-80-12-2

**NOTE**

Each lead centerline is located within 0.15 mm (0.006 inch) of its true position (T.P.) at maximum material condition.

| ITEM | MILLIMETERS | INCHES |
|---|---|---|
| A | 24.7±0.4 | $0.972^{+0.017}_{-0.016}$ |
| B | 20.0±0.2 | $0.795^{+0.009}_{-0.008}$ |
| C | 14.0±0.2 | $0.551^{+0.009}_{-0.008}$ |
| D | 18.7±0.4 | 0.736±0.016 |
| F | 1.0 | 0.039 |
| G | 0.8 | 0.031 |
| H | 0.35±0.10 | $0.014^{+0.004}_{-0.005}$ |
| I | 0.15 | 0.006 |
| J | 0.8 (T.P.) | 0.031 (T.P.) |
| K | 2.35±0.2 | $0.093^{+0.008}_{-0.009}$ |
| L | 1.2±0.2 | $0.047^{+0.009}_{-0.008}$ |
| M | $0.15^{+0.10}_{-0.05}$ | $0.006^{+0.004}_{-0.003}$ |
| N | 0.15 | 0.006 |
| P | $2.05^{+0.2}_{-0.1}$ | $0.081^{+0.008}_{-0.005}$ |
| Q | 0.1±0.1 | 0.004±0.004 |
| S | 2.45 MAX. | 0.097 MAX. |

When placing your order for this package, specify package code μPD7228G12.