The application circuits and their parameters are for reference only and are not intended for use in actual design-ins.

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Corporation. NEC Corporation assumes no responsibility for any errors which may appear in this document.

NEC Corporation does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from use of a device described herein or any other liability arising from use of such device. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Corporation or others.

While NEC Corporation has been making continuous effort to enhance the reliability of its semiconductor devices, the possibility of defects cannot be eliminated entirely. To minimize risks of damage or injury to persons or property arising from a defect in an NEC semiconductor device, customers must incorporate sufficient safety measures in its design, such as redundancy, fire-containment, and anti-failure features.

NEC devices are classified into the following three quality grades: "Standard", "Special", and "Specific". The Specific quality grade applies only to devices developed based on a customer designated "quality assurance program" for a specific application. The recommended applications of a device depend on its quality grade, as indicated below. Customers must check the quality grade of each device before using it in a particular application.

- **Standard**: Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots.
- **Special**: Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).
- **Specific**: Aircrafts, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems or medical equipment for life support, etc.

The quality grade of NEC devices is "Standard" unless otherwise specified in NEC's Data Sheets or Data Books. If customers intend to use NEC devices for applications other than those specified for Standard quality grade, they should contact an NEC sales representative in advance.

Anti-radioactive design is not implemented in this product.
Main revisions in this edition

<table>
<thead>
<tr>
<th>Pages</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 2</td>
<td>• $\mu$PD7507 $\mu$PD7507B</td>
</tr>
<tr>
<td></td>
<td>• Deletion of description related to $\mu$COM-43NA</td>
</tr>
</tbody>
</table>
CONTENTS

CHAPTER 1  GENERAL ................................................................. 1
  1.1 General .................................................................................................................. 1
  1.2 Internal Segment Decoder ..................................................................................... 3
    1.2.1 7-segment decoder .............................................................................................. 3
    1.2.2 14-segment decoder ............................................................................................ 6
    1.2.3 The Input of serial data ....................................................................................... 8
  1.3 The Commands of $\mu\text{PD7225}$ ........................................................................ 9
    1.3.1 MODE SET .......................................................................................................... 9
    1.3.2 SYNCHRONIZED TRANSFER ................................................................................ 9
    1.3.3 UNSYNCHRONIZED TRANSFER........................................................................... 9
    1.3.4 PAUSE TRANSFER ............................................................................................ 10
    1.3.5 BLINKING ON .................................................................................................. 10
    1.3.6 BLINKING OFF ................................................................................................ 10
    1.3.7 DISPLAY ON .................................................................................................... 10
    1.3.8 DISPLAY OFF .................................................................................................... 10
    1.3.9 WITH SEGMENT DECODER ............................................................................... 11
    1.3.10 WITHOUT SEGMENT DECODER ...................................................................... 11
    1.3.11 LOAD DATA POINTER ...................................................................................... 11
    1.3.12 WRITE DATA MEMORY .................................................................................. 11
    1.3.13 OR DATA MEMORY ......................................................................................... 11
    1.3.14 AND DATA MEMORY ..................................................................................... 12
    1.3.15 CLEAR DATA MEMORY .................................................................................. 12
    1.3.16 WRITE BLINKING DATA MEMORY ................................................................. 12
    1.3.17 OR BLINKING DATA MEMORY ......................................................................... 12
    1.3.18 AND BLINKING DATA MEMORY ....................................................................... 12
    1.3.19 CLEAR BLINKING DATA MEMORY .................................................................. 13

CHAPTER 2  EXAMPLES OF APPLICATIONS ......................................................... 15
  2.1 Matters Attended to in a Program .......................................................................... 15
  2.2 System Using $\mu\text{PD7507B}$ ............................................................................. 15
    2.2.1 Interface with $\mu\text{PD7507B}$ and $\mu\text{PD7225}$ ...................................................... 15
    2.2.2 The connection of $\mu\text{PD7225}$ and LCD ......................................................... 16
    2.2.3 Structure of the display data within the program memory of $\mu\text{PD7507B}$ ........ 16
    2.2.4 Program example .............................................................................................. 17
    2.2.5 How to set wait time ......................................................................................... 23

APPENDIX A  BIAS OF LCD AND THE NUMBER OF TIME SHARING DRIVES .......... 25

APPENDIX B  TIME SHARING DRIVE AND THE MAXIMUM NUMBER OF DISPLAY ELEMENTS . 27

APPENDIX C  DISPLAY TIMING AND SEGMENT DRIVE SIGNALS ........................... 29

APPENDIX D  LCD POWER SOURCE CIRCUITS .................................................... 31
1.1 General

μPD7225 is an LCD (Liquid Crystal Display) controller/driver which is programmable by software. Figure 1-1. shows the pin configurations for μPD7225. Figure 1-2. shows a functional block diagram. The μPD7225 interfaces with the CPU through a serial ports, in a microcomputer application system, for directly controlling LCD by static 2-, 3- or 4-time sharing. It also incorporates a segment decoder for generating specific segment patterns and can also control blinking operations.
Figure 1-2. Block Diagram

LCD DRIVER

DISPLAY DATA LATCH

DATA MEMORY

SEGMENT DECODER

DATA MEMORY

COMMAND/DATA REGISTER

SERIAL INTERFACE

COMMAND DECODER

LCD TIMING CONTROL

Vcc1

Vcc2

Vcc3

SYNC

CL1

CL2

Vcc

Vss

RESET

CS

C/D

BUSY

S31 S30 S29

S1 S0

COM COM COM COM
1.2 Internal Segment Decoder

μPD7225 incorporates 7-segment type and 14-segment type decoders, taking in serial data at the SI pin, and generating patterns as shown in Figure 1-4. and Figure 1-5.

1.2.1 7-segment decoder

A 7-segment decoder, which performs 3- or 4-time sharing drive can generate numeric characters 0 to 9, five kinds of sings and blank display codes.

When the LCD display is made with the segment decoder output (display code), the LCD configuration shown in Figure 1-3. should be used. With LCD that is not configured like this, different display patterns are appeared.

Figure 1-3. 7-segment Type LCD

Make the connection in the following configuration for a 3-time sharing type of LCD.
In the case of 4-time sharing, make the following connection.

- SEG-N
  - COM0
  - COM2
  - COM3
  - SEG-N + 1

- SEG
  - SEG N: a, b, c, DP
  - SEG N + 1: d, e, f, g
  - COM 0: a, f
  - COM 1: b, g
  - COM 2: c, e
  - COM 3: d, DP
**Figure 1-4. 7-Segment**

<table>
<thead>
<tr>
<th>Data (HEX)</th>
<th>Display pattern</th>
<th>3-time sharing</th>
<th>4-time sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>N + 2</td>
<td>N + 1</td>
</tr>
<tr>
<td>00</td>
<td>00</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>01</td>
<td>01</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>02</td>
<td>02</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>03</td>
<td>03</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>04</td>
<td>04</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>05</td>
<td>05</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>06</td>
<td>06</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>07</td>
<td>07</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data (HEX)</th>
<th>Display pattern</th>
<th>3-time sharing</th>
<th>4-time sharing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>N + 2</td>
<td>N + 1</td>
</tr>
<tr>
<td>08</td>
<td>08</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>09</td>
<td>09</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>0A</td>
<td>0A</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0B</td>
<td>0B</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>0C</td>
<td>0C</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>0D</td>
<td>0D</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>0E</td>
<td>0E</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>0F</td>
<td>0F</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
1.2.2 14-segment decoder

A 14-segment decoder, which performs 4-time sharing drive, can generate 36 kinds of alphanumeric characters, 12 kinds of signs and blank display codes.

**Figure 1-5. 14-Segment Type**

In the case of a 14-segment type, only the 4-time sharing can be used. The relation between SEGMENT and COMMON is realized by the following connection.

**Figure 1-6. 14-Segment Type LCD**

Shown next is the configuration of input data, display pattern and the display data written in the data memory. In the case of a 7-segment type, the 4 least significant bits (D3 to D0) are decoded, whereas the 7 least significant bits (D6 to D0) are decoded in the 14-segment type. In this case, the input data and display pattern of the 14-segment type correspond to the 8-bit ASCII code. The beginning write address for the display data should be Address N.
**Figure 1-7. 14-Segment**

<table>
<thead>
<tr>
<th>Data (HEX)</th>
<th>Display pattern</th>
<th>Data memory N+3 N+2 N+1 N</th>
<th>Display pattern</th>
<th>Data memory N+3 N+2 N+1 N</th>
<th>Display pattern</th>
<th>Data memory N+3 N+2 N+1 N</th>
<th>Display pattern</th>
<th>Data memory N+3 N+2 N+1 N</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td><img src="image1" alt="" /></td>
<td><img src="image2" alt="" /></td>
<td><img src="image3" alt="" /></td>
<td><img src="image4" alt="" /></td>
<td><img src="image5" alt="" /></td>
<td><img src="image6" alt="" /></td>
<td><img src="image7" alt="" /></td>
<td><img src="image8" alt="" /></td>
</tr>
<tr>
<td>1</td>
<td><img src="image9" alt="" /></td>
<td><img src="image10" alt="" /></td>
<td><img src="image11" alt="" /></td>
<td><img src="image12" alt="" /></td>
<td><img src="image13" alt="" /></td>
<td><img src="image14" alt="" /></td>
<td><img src="image15" alt="" /></td>
<td><img src="image16" alt="" /></td>
</tr>
<tr>
<td>2</td>
<td><img src="image17" alt="" /></td>
<td><img src="image18" alt="" /></td>
<td><img src="image19" alt="" /></td>
<td><img src="image20" alt="" /></td>
<td><img src="image21" alt="" /></td>
<td><img src="image22" alt="" /></td>
<td><img src="image23" alt="" /></td>
<td><img src="image24" alt="" /></td>
</tr>
<tr>
<td>3</td>
<td><img src="image25" alt="" /></td>
<td><img src="image26" alt="" /></td>
<td><img src="image27" alt="" /></td>
<td><img src="image28" alt="" /></td>
<td><img src="image29" alt="" /></td>
<td><img src="image30" alt="" /></td>
<td><img src="image31" alt="" /></td>
<td><img src="image32" alt="" /></td>
</tr>
<tr>
<td>4</td>
<td><img src="image33" alt="" /></td>
<td><img src="image34" alt="" /></td>
<td><img src="image35" alt="" /></td>
<td><img src="image36" alt="" /></td>
<td><img src="image37" alt="" /></td>
<td><img src="image38" alt="" /></td>
<td><img src="image39" alt="" /></td>
<td><img src="image40" alt="" /></td>
</tr>
<tr>
<td>5</td>
<td><img src="image41" alt="" /></td>
<td><img src="image42" alt="" /></td>
<td><img src="image43" alt="" /></td>
<td><img src="image44" alt="" /></td>
<td><img src="image45" alt="" /></td>
<td><img src="image46" alt="" /></td>
<td><img src="image47" alt="" /></td>
<td><img src="image48" alt="" /></td>
</tr>
<tr>
<td>6</td>
<td><img src="image49" alt="" /></td>
<td><img src="image50" alt="" /></td>
<td><img src="image51" alt="" /></td>
<td><img src="image52" alt="" /></td>
<td><img src="image53" alt="" /></td>
<td><img src="image54" alt="" /></td>
<td><img src="image55" alt="" /></td>
<td><img src="image56" alt="" /></td>
</tr>
<tr>
<td>7</td>
<td><img src="image57" alt="" /></td>
<td><img src="image58" alt="" /></td>
<td><img src="image59" alt="" /></td>
<td><img src="image60" alt="" /></td>
<td><img src="image61" alt="" /></td>
<td><img src="image62" alt="" /></td>
<td><img src="image63" alt="" /></td>
<td><img src="image64" alt="" /></td>
</tr>
<tr>
<td>8</td>
<td><img src="image65" alt="" /></td>
<td><img src="image66" alt="" /></td>
<td><img src="image67" alt="" /></td>
<td><img src="image68" alt="" /></td>
<td><img src="image69" alt="" /></td>
<td><img src="image70" alt="" /></td>
<td><img src="image71" alt="" /></td>
<td><img src="image72" alt="" /></td>
</tr>
<tr>
<td>9</td>
<td><img src="image73" alt="" /></td>
<td><img src="image74" alt="" /></td>
<td><img src="image75" alt="" /></td>
<td><img src="image76" alt="" /></td>
<td><img src="image77" alt="" /></td>
<td><img src="image78" alt="" /></td>
<td><img src="image79" alt="" /></td>
<td><img src="image80" alt="" /></td>
</tr>
<tr>
<td>A</td>
<td><img src="image81" alt="" /></td>
<td><img src="image82" alt="" /></td>
<td><img src="image83" alt="" /></td>
<td><img src="image84" alt="" /></td>
<td><img src="image85" alt="" /></td>
<td><img src="image86" alt="" /></td>
<td><img src="image87" alt="" /></td>
<td><img src="image88" alt="" /></td>
</tr>
<tr>
<td>B</td>
<td><img src="image89" alt="" /></td>
<td><img src="image90" alt="" /></td>
<td><img src="image91" alt="" /></td>
<td><img src="image92" alt="" /></td>
<td><img src="image93" alt="" /></td>
<td><img src="image94" alt="" /></td>
<td><img src="image95" alt="" /></td>
<td><img src="image96" alt="" /></td>
</tr>
<tr>
<td>C</td>
<td><img src="image97" alt="" /></td>
<td><img src="image98" alt="" /></td>
<td><img src="image99" alt="" /></td>
<td><img src="image100" alt="" /></td>
<td><img src="image101" alt="" /></td>
<td><img src="image102" alt="" /></td>
<td><img src="image103" alt="" /></td>
<td><img src="image104" alt="" /></td>
</tr>
<tr>
<td>D</td>
<td><img src="image105" alt="" /></td>
<td><img src="image106" alt="" /></td>
<td><img src="image107" alt="" /></td>
<td><img src="image108" alt="" /></td>
<td><img src="image109" alt="" /></td>
<td><img src="image110" alt="" /></td>
<td><img src="image111" alt="" /></td>
<td><img src="image112" alt="" /></td>
</tr>
<tr>
<td>E</td>
<td><img src="image113" alt="" /></td>
<td><img src="image114" alt="" /></td>
<td><img src="image115" alt="" /></td>
<td><img src="image116" alt="" /></td>
<td><img src="image117" alt="" /></td>
<td><img src="image118" alt="" /></td>
<td><img src="image119" alt="" /></td>
<td><img src="image120" alt="" /></td>
</tr>
<tr>
<td>F</td>
<td><img src="image121" alt="" /></td>
<td><img src="image122" alt="" /></td>
<td><img src="image123" alt="" /></td>
<td><img src="image124" alt="" /></td>
<td><img src="image125" alt="" /></td>
<td><img src="image126" alt="" /></td>
<td><img src="image127" alt="" /></td>
<td><img src="image128" alt="" /></td>
</tr>
</tbody>
</table>
1.2.3 The Input of serial data

Serial data is synchronized by the serial clock in units of 8 bits and inputted to the SI pin at the top of MSB. As BUSY becomes low when CS becomes low, synchronization is made with SCK when the BUSY signal becomes high after internal processing (SCK counter and data pointer are cleared), and the first bit (MSB) is transferred. Serial data is transferred to the serial register in units of 1 bit by the rise of SCK. Inputting eight serial clock transitions causes all 8-bit data to be transferred to the serial register. Upon the rise of the 8th serial clock BUSY becomes low, the state of C/D pin is fetched, and it is determined whether the 8-bit data is a command or data.

Then, the contents of the serial register are fetched by the command/data register.

When two bytes or more of serial data are inputted continuously, CS should be left low until the input of all the bytes is completed. Every time the input of one byte is completed, BUSY becomes low. As BUSY becomes high when serial data is fetched by the command/data register from the serial register, the next serial data can be inputted.

When CS is raised after the input of all serial data is completed, the contents of the data memory are displayed.

Do not raise CS while a byte is being transferred (i.e., in the state where the serial clock is not input through eight transitions).

When it is necessary to make the temporarily stop transfer due to a CPU interruption while several bytes are being transferred, make CS high after executing the PAUSE TRANSFER command. If CS is made high in this case, transfer is not made from the data memory to the data display latch.

To start the transfer of the serial data again, CS is made low like an ordinary transfer start. In this case, however, only the SCK counter is cleared and the data pointer keeps the contents before interruption. Therefore, when the start of the next serial data starts, the transferred data is processed as the subsequent data.

Figure 1-8. One-byte Input

![One-byte Input Diagram](attachment:one-byte-input.png)

Figure 1-9. The Continuous Input of 5 Bytes

![Continuous Input of 5 Bytes Diagram](attachment:continuous-input-5-bytes.png)
1.3 The Commands of $\mu$PD7225

1.3.1 MODE SET

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>0</th>
<th>M2</th>
<th>M1</th>
<th>M0</th>
<th>F1</th>
<th>F0</th>
</tr>
</thead>
</table>

This command sets the time sharings number for LCD display, the bias method and frame frequency.

(a) M1 and M0 specify time sharing.

<table>
<thead>
<tr>
<th>M1</th>
<th>M0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

(b) M2 specifies the bias method.

<table>
<thead>
<tr>
<th>M2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1/1</td>
</tr>
</tbody>
</table>

(c) F1 and F0 sets the frame frequency.

<table>
<thead>
<tr>
<th>F1</th>
<th>F0</th>
<th>Scale ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>$2^0$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>$2^1$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>$2^0$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>$2^1$</td>
</tr>
</tbody>
</table>

1.3.2 SYNCHRONIZED TRANSFER

| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |

This command controls the re-writing of display data.

Raising the CS signal usually re-writes the display data (i.e. the transfer of display data from the data memory to the display data latch). After this command is executed, the display data is re-written at the beginning of the alternating drive period (frame frequency times the number of time sharings) when the CS signal is raised.

1.3.3 UNSYNCHRONIZED TRANSFER

| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |

This command controls the re-writing of display data.

After this command is executed, the display data is re-written upon raising the CS pin.
1.3.4 PAUSE TRANSFER

0 0 1 1 1 0 0 0

This command inhibits the re-writing of display data. After this command is executed, data displayed at the rise of the first CS pin is not re-written, but it is retained until the second CS pin is raised. Furthermore, raising the first CS pin does not clear the data pointer. This command is used when the CS pin must be raised temporarily due to a CPU interrupt that occurred while the serial data is being inputted.

1.3.5 BLINKING ON

0 0 0 1 1 0 1 K0

This command sets the blinking state. The least significant bit K0 is used to set the blinking frequency.

K0
0 ----------- fosc/2^17 (Hz)
1 ----------- fosc/2^16 (Hz) fosc: Oscillation frequency

1.3.6 BLINKING OFF

0 0 0 1 1 0 0 0

Execution of this command stops the blinking operation.

1.3.7 DISPLAY ON

0 0 0 1 0 0 0 1

After this command is executed, the LCD display begins according to the display data of the display data latch.

1.3.8 DISPLAY OFF

0 0 0 1 0 0 0 0

Execution of this command makes non-selective the relation between all the common drive signals and the segment drive signals.
As a result, the display is extinguished. This command does not affect the transfer of display data from the data memory to the display data latch.

**1.3.9 WITH SEGMENT DECODER**

\[
\begin{array}{cccccc}
0 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 & 1
\end{array}
\]

Data inputted after this command is executed is sent to the segment decoder, and the code obtained by the segment decoder is written into the data memory.

**1.3.10 WITHOUT SEGMENT DECODER**

\[
\begin{array}{cccccc}
0 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 & 0
\end{array}
\]

Data inputted after this command is executed is written into the data memory without passing the segment decoder.

**1.3.11 LOAD DATA POINTER**

\[
\begin{array}{cccccc}
1 & 1 & 1 & \text{D4} & \text{D3} & \text{D2} \\
1 & \text{D1} & \text{D0}
\end{array}
\]

This command sets the immediate data D4-D0 in the data pointer.

**1.3.12 WRITE DATA MEMORY**

\[
\begin{array}{cccccc}
1 & 1 & 0 & 1 & \text{D3} & \text{D2} \\
1 & \text{D1} & \text{D0}
\end{array}
\]

This command stores the immediate data D3-D0 in the data memory addressed by the data pointer, incrementing by (+1) the contents of the data pointer.

**1.3.13 OR DATA MEMORY**

\[
\begin{array}{cccccc}
1 & 0 & 1 & 1 & \text{D3} & \text{D2} \\
1 & \text{D1} & \text{D0}
\end{array}
\]

The contents of the data memory addressed by the data pointer and the immediate data D3-D0 are ORed, and the result is stored in the data memory. The contents of the data pointer are incremented by (+1).
1.3.14 AND DATA MEMORY

\[
\begin{array}{cccccccc}
1 & 0 & 0 & 1 & D3 & D2 & D1 & D0 \\
\end{array}
\]

The contents of the data memory addressed by the data pointer and the immediate data D3-D0 are ANDed; the result is stored in the data memory. The contents of the data pointer are incremented by (+1).

1.3.15 CLEAR DATA MEMORY

\[
\begin{array}{cccccccc}
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\end{array}
\]

This command clears the contents of the data memory and the data pointer.

1.3.16 WRITE BLINKING DATA MEMORY

\[
\begin{array}{cccccccc}
1 & 1 & 0 & 0 & D3 & D2 & D1 & D0 \\
\end{array}
\]

This command stores the immediate data D3-D0 in the blinking data memory addressed by the data pointer, incrementing the contents of the data pointer by (+1).

1.3.17 OR BLINKING DATA MEMORY

\[
\begin{array}{cccccccc}
1 & 0 & 1 & 0 & D3 & D2 & D1 & D0 \\
\end{array}
\]

The contents of the blinking data memory addressed by the data pointer and the immediate data D3-D0 are ORed, the result is stored in the blinking data memory. The contents of the data pointer are incremented by (+1).

1.3.18 AND BLINKING DATA MEMORY

\[
\begin{array}{cccccccc}
1 & 0 & 0 & 0 & D3 & D2 & D1 & D \\
\end{array}
\]

The contents of the blinking data memory addressed by the data pointer and the immediate data D3-D0 are ANDed, the result is stored in the blinking data memory. The contents of the data pointer are incremented by (+1).
1.3.19 CLEAR BLINKING DATA MEMORY

This command clears the contents of the blinking data memory and the data pointer.
2.1 Matters Attended to in a Program

(a) The high-level signal to the CS pin must have 8 clock cycles or more at the clock frequency.

(b) When the data is transferred, be sure to check the BUSY pin to confirm whether or not the data is transferable.

(c) Be sure not to raise the chip select signal while the data is being transferred. Raising it during transfer may cause a malfunction.

2.2 System Using μPD7507B

This section introduces an example where a CMOS 4-bit one-chip microcomputer (μPD7507B) is used as the control system for the μPD7225. In this example, it is presumed that the display panel which uses a 14-segment 8-digit LCD panel writes the data in the mode to use the segment decoder. In addition, it is presumed that this program does not test the BUSY signal outputted by μPD7225, and that the serial data is sent.

2.2.1 Interface with μPD7507B and μPD7225

Figure 2-1. Interface with μPD7507B

![Diagram of interface with μPD7507B and μPD7225]
Shown below are the pin functions of µPD7507B in this interface.

- **P30:** Used for the chip Select signal.
- **P31:** Specifies command/data for the serial data to be written.
- **P02/SO:** Used for serial data output (command/data).
- **P01/SCK:** Used for the serial clock output.

The clock for µPD7225 can be easily made only by connecting resistance to clock the pins (CL1, CL2).

Example:  \( R = 180k\Omega \sim f_{\text{CLK}} \sim 130 \text{ kHz} \) (TYP.)
\[ V_{DD} = 5.0 \text{ V} \]

When it is necessary to adjust the clock with a variable resistance, use \( R = 180 \text{ k}\Omega \pm 5 \% \).

### 2.2.2 The connection of µPD7225 and LCD

**Figure 2-2. Example of Connection for LCD Display**

![Figure 2-2](image)

### 2.2.3 Structure of the display data within the program memory of µPD7507B

This section shows the structure of the data written in the program memory of µPD7507B to display “D7225NEC”, as shown in the example.

**Program memory**

<table>
<thead>
<tr>
<th>Address</th>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>150H</td>
<td>C4</td>
<td>&quot;D&quot;</td>
</tr>
<tr>
<td>151H</td>
<td>B7</td>
<td>&quot;7&quot;</td>
</tr>
<tr>
<td>152H</td>
<td>B2</td>
<td>&quot;2&quot;</td>
</tr>
<tr>
<td>153H</td>
<td>B2</td>
<td>&quot;2&quot;</td>
</tr>
<tr>
<td>154H</td>
<td>B5</td>
<td>&quot;5&quot;</td>
</tr>
<tr>
<td>155H</td>
<td>CE</td>
<td>&quot;N&quot;</td>
</tr>
<tr>
<td>156H</td>
<td>C5</td>
<td>&quot;E&quot;</td>
</tr>
<tr>
<td>157H</td>
<td>C3</td>
<td>&quot;C&quot;</td>
</tr>
</tbody>
</table>
Write the data in $\mu$PD7225 in the order $157\text{H} \rightarrow 150\text{H}$ of the program memory.

### 2.2.4 Program example

This section shows a program to control the $\mu$PD7225 by dividing one into the INITIALIZE routine, data routine, the display routine of DP, weight routine and serial data transfer routine.

This section shows a program which only controls $\mu$PD7225. It is assumed, therefore, that the data memory of $\mu$PD7507B has been initialized or the stack pointer has already been set.

#### (1) INITIALIZE Routine

In the INITIALIZE routine, each mode is set after the RESET for $\mu$PD7225 is removed, the display RAM is cleared and the command of display start is transferred.

**Flowchart**

```
INIT
  \_ C/D = 1
  \_ CS = 0

WAIT
    ; BUSY = 0 \_ 1 Wait

MODE SET

SIOUT
    ; Serial data transfer
    (Command issue)

WITH SEGMENT DECODER

SIOUT
    ; Segment decoder use

CLEAR DATA MEMORY

SIOUT
    ; Command issue

DISPLAY ON
    ; Display start

SIOUT
    ; Command issue

END
```
Program list

INIT: LAI 0EH
OP 3 ; C/D = 1 CS = 0
CALL WAIT ; WAIT BUSY 0 → 1
LHLI 7FH
LAI 2 ; MODE SET
ST ; COMMAND SET
LAI 4 ;
CALL SIOUT ; COMMAND OUT
LAI 5 ;
ST ; WITH SEGMENT DECODER
LAI 1 ;
CALL SIOUT ;
LAI 0 ;
ST ; CLEAR DATA MEMORY
LAI 2 ;
CALL SIOUT
LAI 1 ;
ST ; DISPLAY ON
CALL SIOUT

; INIT END

In this program, CS is not raised after issuing the DISPLAY ON command. If it is necessary to raise CS, be sure to hold CS = 1 for 48 clocks or more. (The μPD7225 clock) (In the case of UNSYNCHRONIZED TRANSFER MODE)
(2) Data routine
The data routine is used to write display data.

Flowchart

```
DATA
  C/D = 1
  CS = 0 ; Command specification
  Chip Select setting
  WAIT
  LOAD DATA POINTER
  SIOUT ; Data pointer setting
  C/D = 0 ; Data specification
  DATA ADDRESS SET
      ; Setting of the table
      ; address of the display data
  DATA SET ; Setting of the display data
  SIOUT ; Display data issue
  DATA ADDRESS-1 ; Display data address decrement
      NO
      TRANSFER END
      YES
  C/D = 1
  CS = 0 ; Command specification
  END
```
## Program list

**DATA:**

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAI 0EH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OP 3</td>
<td>;</td>
<td>$C/D = 1, CS = 0$</td>
</tr>
<tr>
<td>CALL WAIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LHLI 7FH</td>
<td>;</td>
<td></td>
</tr>
<tr>
<td>LAI 0</td>
<td>;</td>
<td>LOAD DATA POINTER</td>
</tr>
<tr>
<td>ST</td>
<td>;</td>
<td>(DP = 00)</td>
</tr>
<tr>
<td>LAI 0EH</td>
<td>;</td>
<td></td>
</tr>
<tr>
<td>CALL SIOUT</td>
<td>;</td>
<td>COMMAND OUT</td>
</tr>
<tr>
<td>LAI 0CH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OP 3</td>
<td>;</td>
<td>$C/D = 0, CS = 0$</td>
</tr>
<tr>
<td>LAI 7</td>
<td>;</td>
<td>DATA NO.</td>
</tr>
<tr>
<td>LHLI 71H</td>
<td>;</td>
<td>(71) DATA ADDRESS</td>
</tr>
<tr>
<td>ST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TABL:**

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>LADR 71H</td>
<td>;</td>
<td></td>
</tr>
<tr>
<td>ST</td>
<td>;</td>
<td>DATA TABLE ADDRESS SET</td>
</tr>
<tr>
<td>LAI 5</td>
<td>;</td>
<td></td>
</tr>
<tr>
<td>LAMTL</td>
<td>;</td>
<td>TABLE LOOK UP</td>
</tr>
<tr>
<td>CALL SIOUT</td>
<td>;</td>
<td>DATA OUT</td>
</tr>
<tr>
<td>DDRS 71H</td>
<td>;</td>
<td>DATA ADDRESS-1</td>
</tr>
<tr>
<td>JCP TABL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAI 0FH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OP 3</td>
<td>;</td>
<td>$C/D = 1, CS = 1$</td>
</tr>
</tbody>
</table>

;DATA END
(3) DP display routine
The DP display routine displays "." between "5" and "N" as shown in Figure 2-12. The use of this routine is not restricted to DP, but it can be applied to the change of each character.

Flowchart

Program list

DP:  
LAI 0EH  
OP 3 ; C/D = 1, CS = 0  
CALL WAIT  
LAI 0CH ; LOAD DATA POINTER  
ST ; (DP = 0C)  
LAI 0EH ;  
CALL SIOUT  
LAI 8 ; OR DATA MEMORY  
LAI 0BH ;  
CALL SIOUT ;  
LAI 0FH  
OP 3 ; C/D = 1, CS = 1  
RT

;DP END
(4) Wait routine
Because the BUSY signal outputted by μPD7225 is not tested here, it is necessary to allow enough time until BUSY = 1 is obtained before starting serial data transfer.

Flowchart

Program list

WAIT:  LAI 05H ; WAIT TIME SET
TAL
LOOP:  LAI 0FH ; WAIT TIME SET
TAE
DES ; WAIT
JCP $-1
DLS ; WAIT
JCP LOOP
JHLT 7FH ; "HL" REG SET
RT
(5) Serial data transfer routine

This routine transfers 8-bit serial data to the µPD7225.

Flowchart

```
SIOUT
(ST) ← DATA
DATA TRANSFER
WAIT
RT
```

; Data is set to the shift register
; Shift operation start
; Wait routine

Program list

```
SIOUT: TAMSIO ; (ST) ← DATA
SIO ; DATA TRANSFER
CALL WAIT
RT
```

2.2.5 How to set wait time

The BUSY low level time \( t_{WL} \) for the µPD7225 is shown in the table below.

<table>
<thead>
<tr>
<th>( t_{WL} )</th>
<th>MIN.</th>
<th>MAX.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4</td>
<td>44</td>
<td>1/fc</td>
</tr>
<tr>
<td></td>
<td>(57)*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Where:
- \( fc \) is the clock frequency of the µPD7225.
- 14-segment decoder is used.
- \( \overline{CS} = 1 \geq 48/fc \) : (UNSYNCHRONIZED TRANSFER MODE)
- \( \overline{CS} = 1 \geq (48 + \text{Alternating drive frequency})/fc \) : (SYNCHRONIZED TRANSFER MODE)

In this case, alternating drive frequency = frame frequency \( x \) the number of time-shares
- * signifies the case where BLINKING is used.

When serial data is sent from CPU to µPD7225, the BUSY signal need not be tested if the data is sent after Wait time of \( t_{WL} \) MAX.

The example below shows the setting of the actual Wait time in the above-mentioned Wait routine.

Although µPD7225 generates the clock through an internal oscillator (i.e. R connected to CL1 and CL2), there is a drift in the clock frequency. (See Figure 2-11.)
In considering the MAX. time of BUSY = 0, it is necessary to consider the case where the frequency is the lowest. Therefore the MAX. time of BUSY = 0 under the conditions

\[
R = 180 \text{ k}\Omega \pm 5 \%
\]

\[
V_{DD} = 5 \text{ V} \pm 10 \%
\]

BLINKING unused

14-segment decoder used

UNSYNCHRONIZED TRANSFER mode

is obtained according to the following expression:

\[
t_{\text{WLB}}(\text{MAX.}) \times \frac{1}{f_c(\text{MIN.})} = 44 \times \frac{1}{85} \times 10^3
\]

\[
= 520 (\mu\text{s})
\]

On the other hand, when Wait time is set without using a timer by the software of \(\mu\text{PD7507B}\), it is necessary to consider the case where the clock is the fastest.

On the assumption that the system clock is generated by CR in \(\mu\text{PD7507B}\), the oscillation frequency has the following deviations.

<table>
<thead>
<tr>
<th>Condition</th>
<th>MIN.</th>
<th>MAX.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>f_{CC}</td>
<td>120</td>
<td>280</td>
<td>kHz</td>
</tr>
<tr>
<td>R = 82 \text{ k}\Omega \pm 2 %</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C = 33pF \pm 5 %</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(V_{DD} = 5 \text{ V} \pm 10 %)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

It is therefore necessary to set the Wait routine with respect to \(f_{CC} = 280 \text{ kHz}\).

Because one machine cycle is \(7 [\mu\text{s}]\) at \(f_c = 280 \text{ kHz}\) in \(\mu\text{PD7507B}\), Wait of 75 (machine cycles) becomes necessary in order to pression indicates:

\[
520 [\mu\text{s}] / 7 [\mu\text{s}] = 74.3 = 75 \text{ (machine cycles)}
\]

In order to set Wait time, the lowest clock of the \(\mu\text{PD7225}\) is combined with the fastest system clock of the CPU.
APPENDIX A  BIAS OF LCD AND THE NUMBER OF TIME SHARING DRIVES

Figure A-1. shows the relationship between LCD bias and the number of time sharing drives.

**Figure A-1. Relation between the Bias and the Number of Time Sharing Drives**

<table>
<thead>
<tr>
<th>Bias</th>
<th>Number of time-shares</th>
<th>Static</th>
<th>2-time shares</th>
<th>3-time shares</th>
<th>4-time shares</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static</td>
<td></td>
<td>○</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/2</td>
<td></td>
<td>○</td>
<td>○</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/3</td>
<td></td>
<td></td>
<td></td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>
APPENDIX B  TIME SHARING DRIVE AND THE MAXIMUM NUMBER OF DISPLAY ELEMENTS

Shown below are each time sharing drive for the µPD7225 and the maximum number of elements displayable in the static display.

<table>
<thead>
<tr>
<th>Number of time-sharing drives</th>
<th>Maximum number of displayable elements</th>
<th>Structure of LCD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static drive</td>
<td>32</td>
<td>COM0, S0-S31</td>
</tr>
<tr>
<td>2-time sharing drive</td>
<td>64</td>
<td>COM0, COM1, S0-S31</td>
</tr>
<tr>
<td>3-time sharing drive</td>
<td>96</td>
<td>COM0-COM2, S0-S31</td>
</tr>
<tr>
<td>4-time sharing drive</td>
<td>128</td>
<td>COM0-COM3, S0-S31</td>
</tr>
</tbody>
</table>

Shown below is the maximum number of display digits for a 7-segment type or a 14-segment type.

<table>
<thead>
<tr>
<th>Number of time-sharing drives</th>
<th>7-segment LCD</th>
<th>14-segment LCD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static drive</td>
<td>4 digits + 4 elements*</td>
<td>2 digits + 4 elements*</td>
</tr>
<tr>
<td>2-time sharing drive</td>
<td>8 digits + 16 elements*</td>
<td>4 digits + 8 elements*</td>
</tr>
<tr>
<td>3-time sharing drive</td>
<td>10 digits + 26 elements*</td>
<td>6 digits + 12 elements*</td>
</tr>
<tr>
<td>4-time sharing drive</td>
<td>16 digits + 16 elements*</td>
<td>8 digits + 16 elements*</td>
</tr>
</tbody>
</table>

Remarks: * These elements can be used as decimal points or indicators.

As shown above, an increase in the number of time shares results in an increase in the number of controllable elements.

Therefore, select the number of time shares according to what display is made in the application system and how many display elements are required. An insufficient number of display elements can be coped with by using several µPD7225s.

In the case of a static drive, there is no problem of cross-talk in a time sharing drive, and better display quality can be obtained.
APPENDIX C DISPLAY TIMING AND SEGMENT DRIVE SIGNALS

An LCD requires alternating drive by its nature. However, because an alternating drive loses balance temporarily the instant the display state changes due to a change in timing, the relation may be disturbed between a common drive signal and a segment drive signal. Therefore, a small direct portion may be imposed. (See Figure C-1.)

Figure C-1. Segment Drive Signal Waveform

Although an LCD’s life shortens when DC voltage is imposed for a long time, the life of LCD’s being used today is not significantly affected by the instantaneous imbalance in an alternating drive as shown in Figure C-1. Therefore, the LCD can be driven by the UNSYNCHRONIZED TRANSFER command.

To synchronize a change in the display state with alternating timing, the SYNCHRONIZED TRANSFER command is used. After CS is raised and after this command is executed, a change is made in the display data at the beginning of the next alternating timing.

Figure C-2. Segment Drive Signal Waveform
[MEMO]
Figure D-1(a), (b) and (c) show LCD power source circuits in Static, ½ bias and 1/3 bias.

**Figure. D-1 (a)**  
Static

<table>
<thead>
<tr>
<th>V_{DD}</th>
<th>V_{LC1}</th>
<th>R_x</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure. D-1 (b)**  
1/2 Bias

<table>
<thead>
<tr>
<th>V_{DD}</th>
<th>V_{LC1}</th>
<th>R_2</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure. D-1 (c)**  
1/3 Bias

<table>
<thead>
<tr>
<th>V_{DD}</th>
<th>V_{LC1}</th>
<th>R_3</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Remark:**  
\[ V_{LCD} = V_{DD} - V_{LC3} \]  
\[ V_{LCD} : V_{LC3} = R_1 : 2 : 3 : R_x \]

**Example:** In case of R_1 = R_2 = R_3 = 5 kΩ, bias in 1.3 of 3-V drive and 4-time sharing drive becomes R_x = 10 kΩ

**Figure D-2. LCD Equivalent Circuit**

\[ \mu PD7225G \]

**Remark:**  
\[ C_P = \text{pattern capacity} \]

When represented in an equivalent circuit, LCD drive can be compared with the charging/discharging of C_P as shown in Figure D-2.

Where drive voltage V_{LC0} to 3 is made from an external split resistance, the charging/discharging waveform becomes dulled when this split resistance is large. Therefore, the effective value of voltage drops and contrast becomes poor.
To better the contrast of LCD and improve visibility in such a case, connect a condenser, lower the impedance of the circuit and adjust the waveform.