Sitronix

# ST7920

Chinese Fonts built in LCD controller/driver

## Main Features

- Voltage operating range:
  - 2.7 to 5.5V
- Support 8 bit, 4 bit, serial bus MPU interface
- 64 x 16-bits character display RAM (max. 16 chars x 4 lines, LCD display range 16 char. X 2 lines
- 64 x 256-bits graphic display RAM（GDRAM）
- 2M-bits Chinese fonts ROM (CGROM) supporting 8192 Chinese fonts (16x16 dot matrix)
- 16K-bits half height ROM (HCGROM) supporting 126 character set (16x8 dot matrix)
- 64 x 16-bits character generation RAM (CGRAM)
- 32-common x 64-segment (2 lines display) LCD drivers
- Automatic power on reset
- External reset pin (XRESET)
- With extension segment drivers display area can up to 16x2 lines
- RC oscillator built in (with external R)
- Low power design
  Normal mode (450uA Typ $V_{DD}$=5V)
  Standby mode (30uA Max $V_{DD}$=5V)
- VLCD (V0~ $V_{SS}$): max 7V
- Graphic and character mix modes display
- Multiple instructions：
  - （Display clear）
  - （Return home）
  - （Display on/off）
  - （Cursor on/off）
  - （Display character blink）
  - （Cursor shift）
  - （Display shift）
  - （Vertical line scroll）
  - （By_line reverse display）
  - （Standby mode）
- Built in voltage booster (2 times)
- 1/32 Duty

## Function Description

ST7920 LCD controller/driver IC can display alphabets, numbers, Chinese fonts and self-defined characters. It supports 3 kinds of bus interface, namely 8 bit/ 4bit and serial. All functions, including display RAM, character generator ROM, LCD display drivers and control circuits are all in a one-chip solution. With a minimum system configuration, a Chinese character display system can easily achieved.

ST7920 includes character ROM with 8192 16X16 dots Chinese fonts and 126 16X8 dots half height alphanumerical fonts. Also for graphic display it supports 64x256 dots graphic display area（GDRAM）. Mix mode display with both character and graphic data is possible. ST7920 has built in 4 sets CGRAM providing software programmable 16X16 font.

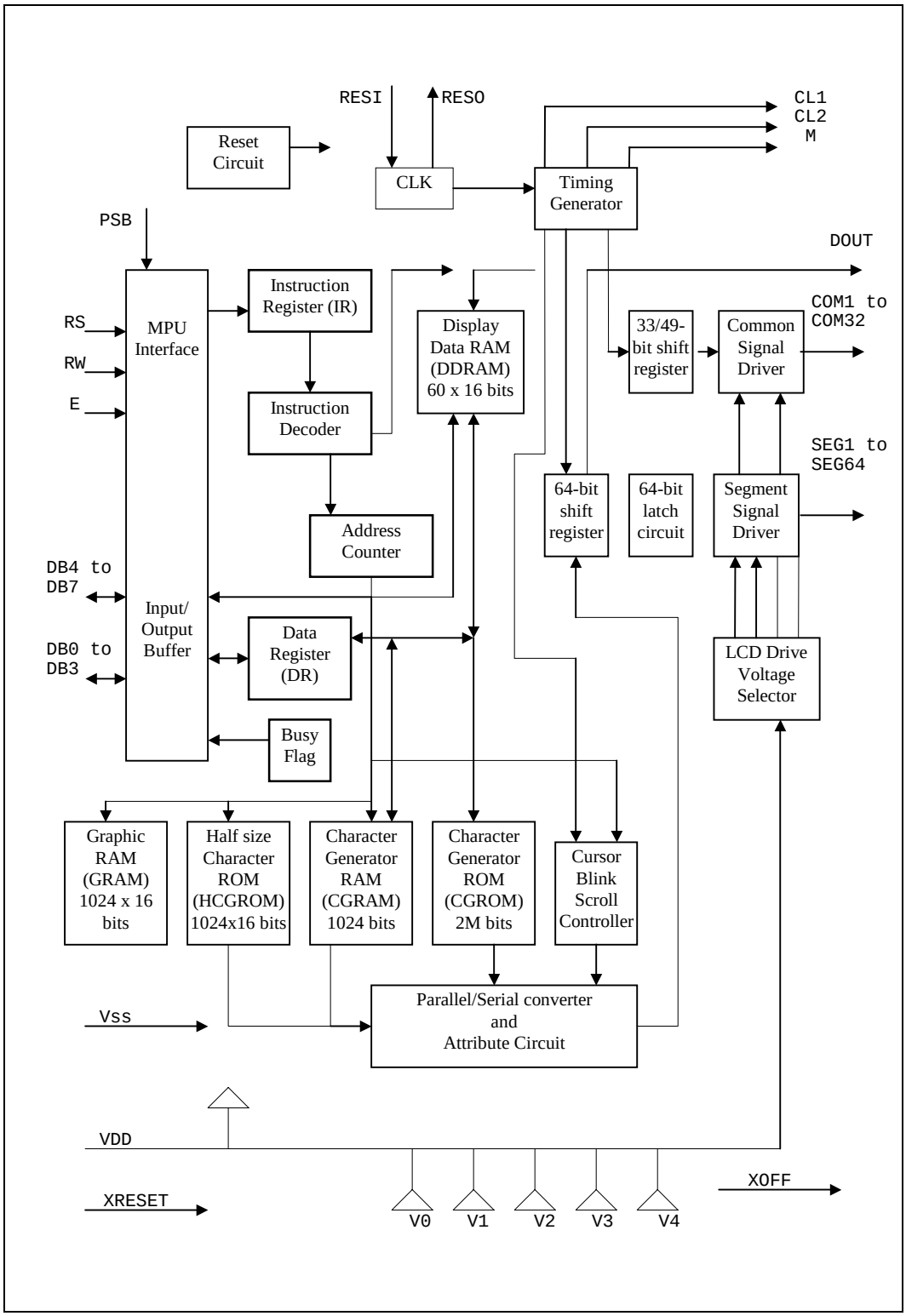
ST7920 has wide operating voltage (2.7V to 5.5V) and low power consumption suitable for battery power portable device.

ST7920 LCD driver consists of 32 common and 64 segments. Together with extension segment driver ST7921, ST7920 can support up to 32 common x 256 segments display.

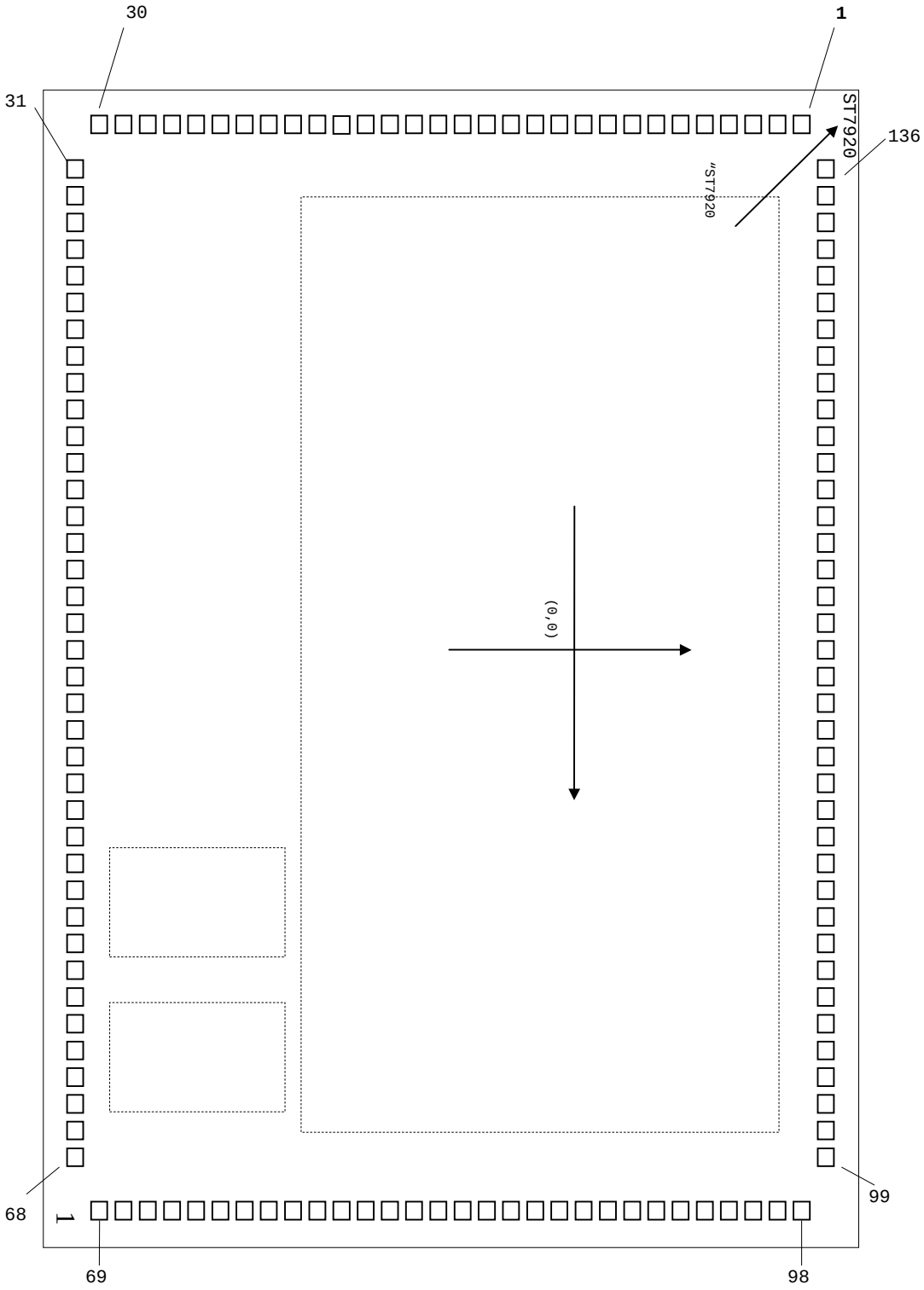
| Product | Font type |
|---|---|
| **ST7920-0A** | **BIG-5 code traditional character set** |
| **ST7920-0B** | **GB code simplified character set** |
| **ST7920-0C** | **GB code,BIG-5 code and Japanese code** |

| ST7920 Specification Revision History | | |
|---|---|---|
| **Version** | **Date** | **Description** |
| C1.7 | 2000/12/15 | 1. VCC changed to VDD.<br>2. VLCD changed from VCC-V5 to V0-VSS.<br>3. DC characteristics input High voltage (Vih) changed to 0.7VDD.<br>4. DC characteristics output High voltage (Voh) changed to 0.8VDD. |
| C1.8 | 2001/03/01 | 1. Chip Size changed.<br>2. ICON 256 dots changed to 240 dots.<br>3. XOFF normal high sleep Low changed to normal low sleep High.<br>4. Added XOFF application.<br>5. Modified application of ST7920 4,5,6 PIN floating. (4,5,6 為 test pin)<br>6. Modified voltage doubler CAP1P, CAP1M, CAP2M capacitors polarity |
| C1.9 | 2001/05/28 | 1. Icon RAM TABLE changed. (TABLE-6)<br>2. Booster description modified. (PAGE-29)<br>3. AC Characteristics modified.<br>4. Added 2Line 16 Chinese Word (32Com X 256Seg) application circuit.<br>5. Added oscillation resistor's relation to power consumption and frequency. |
| C2.0 | 2001/07/03 | 1. Added Register initial values.<br>2. Voltage booster CAP1M CAP1P polarity changed. (PAGE-30) |
| V2.0 | 2001/08/17 | 1. Modified Table 7. (PAGE-14)<br>2. Change to English version. |
| V2.0c | 2001/10/18 | 1. Modified page-38 Serial interface timing diagram |
| V2.0d | 2002/05/09 | 1. Add the standard code (Japan、GB code、BIG-5 code) |
| V3.0 | 2002/10/11 | 1. Delete sleep mode function |
| V3.1 | 2003/04/11 | 1. Modified GDRAM Address (AC5 … AC0 , 00h…3Fh) |
| V3.2 | 2003/09/09 | 1.Add the CGROM and HCGROM test application circuit |
| V3.3 | 2004/03/29 | 1.Updat the using method for ICON. |

## System Block diagram



## Pads diagram




origin: center of chip  coordinates: from pad center

chip size: 5305 X 4074  Pad open: 90 X 90

Pad pitch: 125  unit: μm

*** chip substrate must connect to VSS**

## Pin coordinates

unit: um

| No. | Name | X | Y |
|---|---|---|---|
| 1 | V0 | -2548 | 1812 |
| 2 | V1 | -2548 | 1688 |
| 3 | V2 | -2548 | 1562 |
| 4 | CLK | -2548 | 1438 |
| 5 | TT1 | -2548 | 1312 |
| 6 | TT2 | -2548 | 1188 |
| 7 | V3 | -2548 | 1062 |
| 8 | V4 | -2548 | 938 |
| 9 | VSS | -2548 | 812 |
| 10 | VDD | -2548 | 688 |
| 11 | XRESET | -2548 | 562 |
| 12 | CL1 | -2548 | 438 |
| 13 | CL2 | -2548 | 312 |
| 14 | VDD | -2548 | 188 |
| 15 | M | -2548 | 62 |
| 16 | DOUT | -2548 | -62 |
| 17 | RS | -2548 | -188 |
| 18 | RW | -2548 | -312 |
| 19 | E | -2548 | -438 |
| 20 | VSS | -2548 | -562 |
| 21 | OSC1 | -2548 | -688 |
| 22 | OSC2 | -2548 | -812 |
| 23 | PSB | -2548 | -938 |
| 24 | D0 | -2548 | -1062 |
| 25 | D1 | -2548 | -1188 |
| 26 | D2 | -2548 | -1312 |
| 27 | D3 | -2548 | -1438 |
| 28 | D4 | -2548 | -1562 |
| 29 | D5 | -2548 | -1688 |
| 30 | D6 | -2548 | -1812 |
| 31 | D7 | -2306 | -1933 |
| 32 | XOFF | -2181 | -1933 |
| 33 | VOUT | -2056 | -1933 |
| 34 | CAP3M | -1931 | -1933 |
| 35 | CAP1P | -1806 | -1933 |
| 36 | CAP1M | -1681 | -1933 |
| 37 | CAP2P | -1556 | -1933 |
| 38 | CAP2M | -1431 | -1933 |
| 39 | VD2 | -1306 | -1933 |
| 40 | C[1] | -1181 | -1933 |
| 41 | C[2] | -1056 | -1933 |
| 42 | C[3] | -931 | -1933 |
| 43 | C[4] | -806 | -1933 |
| 44 | C[5] | -681 | -1933 |
| 45 | C[6] | -556 | -1933 |
| 46 | C[7] | -431 | -1933 |
| 47 | C[8] | -306 | -1933 |
| 48 | C[9] | -181 | -1933 |
| 49 | C[10] | -56 | -1933 |
| 50 | C[11] | 69 | -1933 |
| 51 | C[12] | 194 | -1933 |
| 52 | C[13] | 319 | -1933 |
| 53 | C[14] | 444 | -1933 |
| 54 | C[15] | 569 | -1933 |
| 55 | C[16] | 694 | -1933 |
| 56 | C[17] | 819 | -1933 |
| 57 | C[18] | 944 | -1933 |
| 58 | C[19] | 1069 | -1933 |
| 59 | C[20] | 1194 | -1933 |
| 60 | C[21] | 1319 | -1933 |
| 61 | C[22] | 1444 | -1933 |
| 62 | C[23] | 1569 | -1933 |
| 63 | C[24] | 1694 | -1933 |
| 64 | C[25] | 1819 | -1933 |
| 65 | C[26] | 1944 | -1933 |
| 66 | C[27] | 2069 | -1933 |
| 67 | C[28] | 2194 | -1933 |
| 68 | C[29] | 2319 | -1933 |
| 69 | C[30] | 2548 | -1812 |
| 70 | C[31] | 2548 | -1688 |
| 71 | C[32] | 2548 | -1562 |
| 72 | C[33] Not use | 2548 | -1438 |
| 73 | S[64] | 2548 | -1312 |
| 74 | S[63] | 2548 | -1188 |
| 75 | S[62] | 2548 | -1062 |
| 76 | S[61] | 2548 | -938 |

| No. | Name | X | Y |
|---|---|---|---|
| 77 | S[60] | 2548 | -812 |
| 78 | S[59] | 2548 | -688 |
| 79 | S[58] | 2548 | -562 |
| 80 | S[57] | 2548 | -438 |
| 81 | S[56] | 2548 | -312 |
| 82 | S[55] | 2548 | -188 |
| 83 | S[54] | 2548 | -62 |
| 84 | S[53] | 2548 | 62 |
| 85 | S[52] | 2548 | 188 |
| 86 | S[51] | 2548 | 312 |
| 87 | S[50] | 2548 | 438 |
| 88 | S[49] | 2548 | 562 |
| 89 | S[48] | 2548 | 688 |
| 90 | S[47] | 2548 | 812 |
| 91 | S[46] | 2548 | 938 |
| 92 | S[45] | 2548 | 1062 |
| 93 | S[44] | 2548 | 1188 |
| 94 | S[43] | 2548 | 1312 |
| 95 | S[42] | 2548 | 1438 |
| 96 | S[41] | 2548 | 1562 |
| 97 | S[40] | 2548 | 1688 |
| 98 | S[39] | 2548 | 1812 |
| 99 | S[38] | 2319 | 1933 |
| 100 | S[37] | 2194 | 1933 |
| 101 | S[36] | 2069 | 1933 |
| 102 | S[35] | 1944 | 1933 |
| 103 | S[34] | 1819 | 1933 |
| 104 | S[33] | 1694 | 1933 |
| 105 | S[32] | 1569 | 1933 |
| 106 | S[31] | 1444 | 1933 |
| 107 | S[30] | 1319 | 1933 |
| 108 | S[29] | 1194 | 1933 |
| 109 | S[28] | 1069 | 1933 |
| 110 | S[27] | 944 | 1933 |
| 111 | S[26] | 819 | 1933 |
| 112 | S[25] | 694 | 1933 |
| 113 | S[24] | 569 | 1933 |
| 114 | S[23] | 444 | 1933 |
| 115 | S[22] | 319 | 1933 |

| No. | Name | X | Y |
|---|---|---|---|
| 116 | S[21] | 194 | 1933 |
| 117 | S[20] | 69 | 1933 |
| 118 | S[19] | -56 | 1933 |
| 119 | S[18] | -181 | 1933 |
| 120 | S[17] | -306 | 1933 |
| 121 | S[16] | -431 | 1933 |
| 122 | S[15] | -556 | 1933 |
| 123 | S[14] | -681 | 1933 |
| 124 | S[13] | -806 | 1933 |
| 125 | S[12] | -931 | 1933 |
| 126 | S[11] | -1056 | 1933 |
| 127 | S[10] | -1181 | 1933 |
| 128 | S[9] | -1306 | 1933 |
| 129 | S[8] | -1431 | 1933 |
| 130 | S[7] | -1556 | 1933 |
| 131 | S[6] | -1681 | 1933 |
| 132 | S[5] | -1806 | 1933 |
| 133 | S[4] | -1931 | 1933 |
| 134 | S[3] | -2056 | 1933 |
| 135 | S[2] | -2181 | 1933 |
| 136 | S[1] | -2306 | 1933 |

## Pin Description

| Name | No. | I/O | Connects to | Function |
|---|---|---|---|---|
| XRESET | 11 | I | — | System reset low active |
| PSB | 23 | I | — | Interface selection:<br>0: serial mode<br>1: 8/4-bits parallel bus mode |
| RS(CS*) | 17 | I | MPU | Register select<br>0: select instruction write, busy flag read, address counter read<br>1: select data write, read<br>(Chip select) for serial mode<br>1: chip enable<br>0: chip disable |
| RW(SID*) | 18 | I | MPU | Read write control<br>0: write<br>1: read<br>(serial data input) |
| E(SCLK*) | 19 | I | MPU | Enable trigger<br>(serial clock) |
| D4 to D7 | 28 31 | I/O | MPU | Higher nibble data bus for 8 bit interface and data bus for 4 bit interface |
| D0 to D3 | 24 27 | I/O | MPU | Lower nibble data bus for 8 bit interface |
| CL1 | 12 | O | Extension segment drv. | Latch signal for extension segment drivers |
| CL2 | 13 | O | Extension segment drv. | Shift clock for extension segment drivers |
| M | 15 | O | Extension segment drv. | AC signal for extension segment drivers voltage inversion |
| DOUT | 16 | O | Extension segment drv. | Data output for extension segment drivers |
| COM1 to COM32 | 40 71 | O | LCD | Common signals |
| SEG1 to SEG64 | 136 73 | O | LCD | Segment signals |
| V0 to V4 | 1 3<br>7,8 | — | — | LCD bias voltage<br>$V_0$ - V4 ≦ 7 V |
| $V_{DD}$ | 10,14 | I | Power | $V_{DD}$: 2.7V to 5.5V |
| Vss | 9,20 | I | Power | VSS: 0V |
| OSC1, OSC2 | 21,22 | I, O | Resistors | For internal oscillation resistor<br>5.0V R=33K<br>2.7V R=18K<br>use OSC1 for external clock input |
| VOUT | 33 | O | Resistors | LCD voltage doubler output |

***note: The OSC pin must have the shortest wiring pattern of all other pins.To prevent noise from other signal lines , it should also be enclosed with the largest GND pattern possible. Poor noise characteristics on the OSC line will result in malfunction , or adversely affect the clock's duty ratio.**

## Pin description

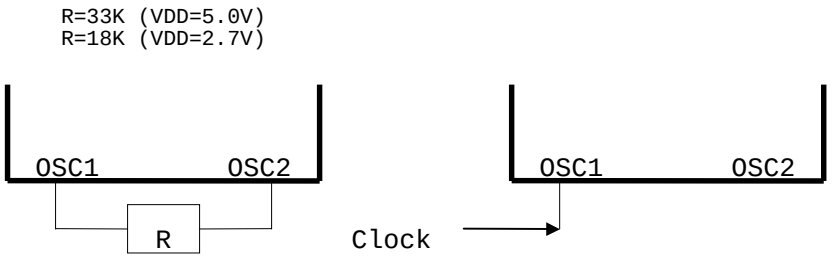
| Name | No. | I/O | Connects to | Description |
|---|---|---|---|---|
| CAP3M<br>CAP1P<br>CAP1M<br>CAP2M | 34<br>35<br>36<br>38 | I/O | Capacitors | Capacitor pins for voltage doubler |
| XOFF | 32 | O | — | Reserved (no connection) |
| CAP2P | 37 | — | — | Reserved (no connection) |
| VD2 | 39 | I | Reference voltage | Voltage doubler reference voltage |
| N.C.<br>N.C.<br>N.C. | 4<br>5<br>6 | I | —<br>—<br>— | Test pins (no connection) |

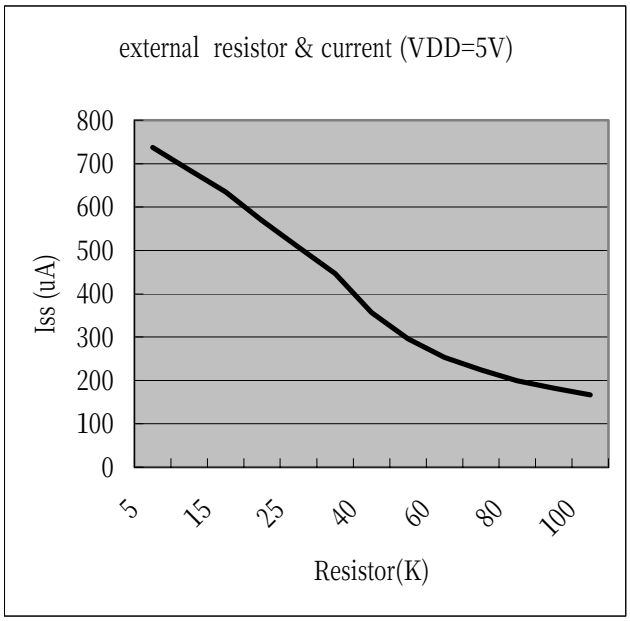
Note:

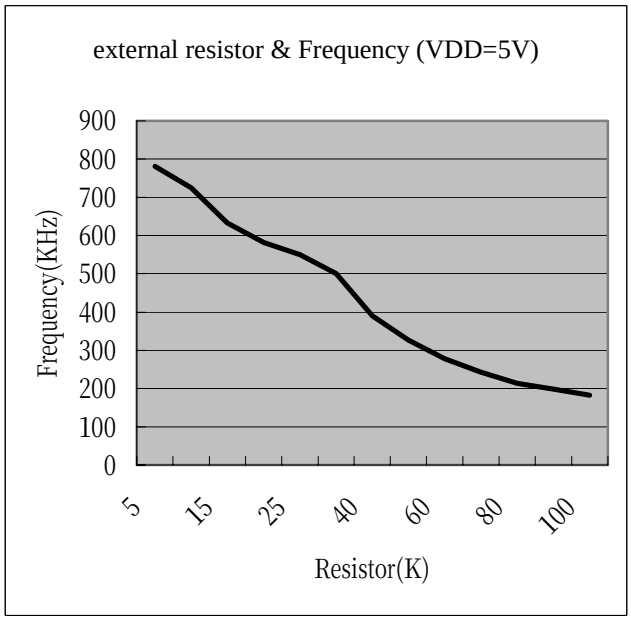
1. VDD>=V0>=V1>=V2>=V3>=V4 must be maintained

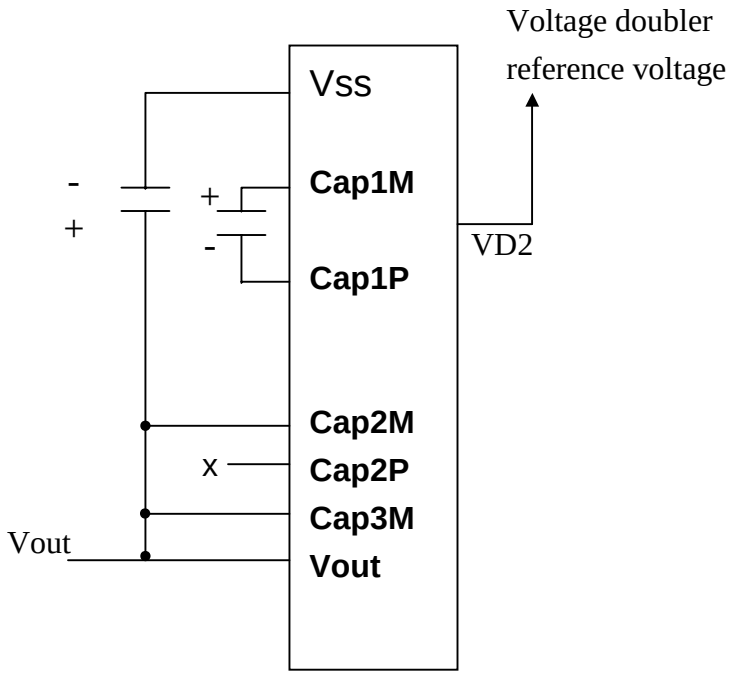
2. Two clock options:

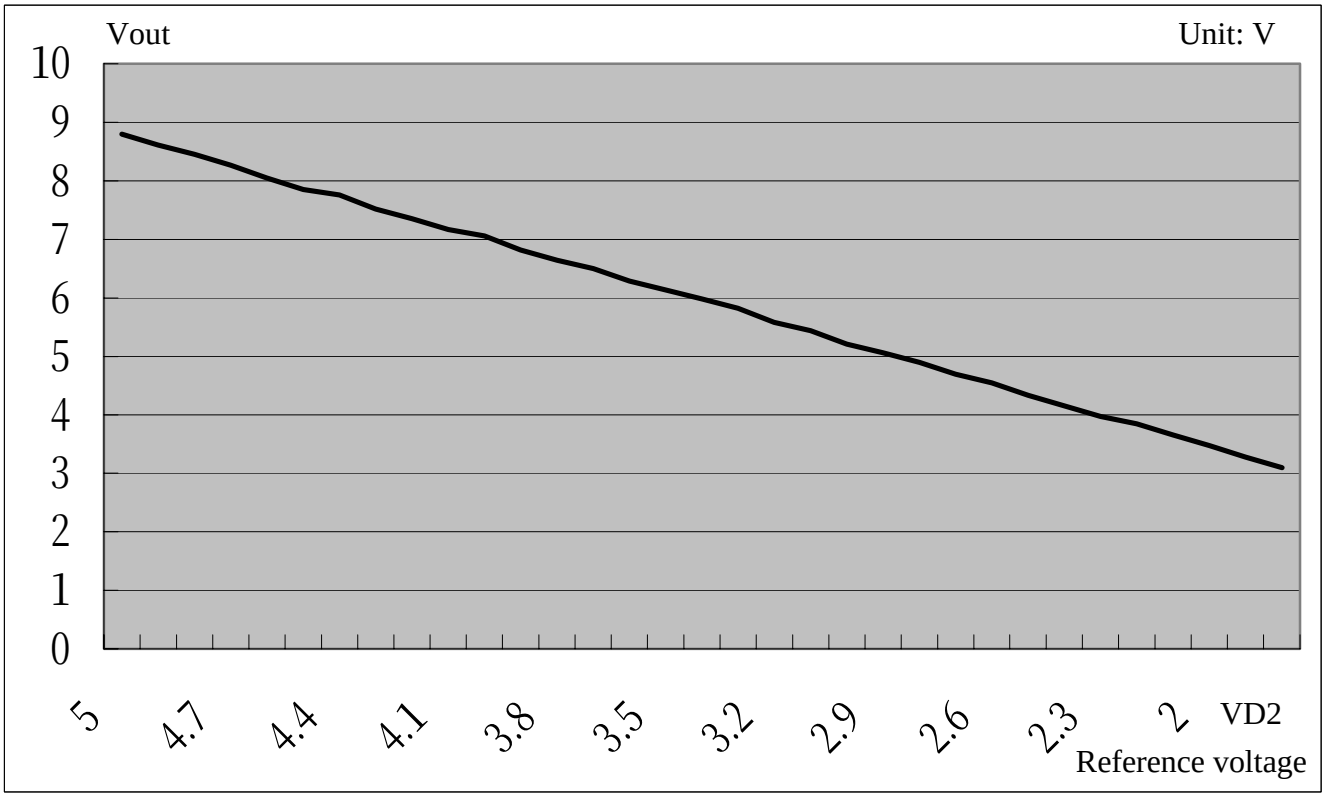
3.When using voltage doubler for VOUT it is recommended that the total sum of bleeder resistors R1~R5 should be larger than 20K Ohm






Doubler voltage mode VD2 & Vout output characteristic

Notes:
Follower loading resistor total 20k(ohm)
Boostaer Cap use 4.7uf
Panel size 80mm * 28mm (check display)

## Function Description :

**System interface**
ST7920 supports 3 kinds of bus interface to MPU. 8 bits parallel, 4 bits parallel and clock synchronized serial interface. Parallel interface is selected by PSB="1" and serial interface by PSB="0". 8 bit / 4 bit interface is selected by function set instruction DL bit.
Two 8 bit registers (data register DR, instruction register IR) are used in ST7920's write and read operation. Data Register（DR）can access DDRAM/CGRAM/GDRAM and IRAM's data through the address pointer implemented by Address Counter (AC). Instruction Register (IR) stores the instruction by MPU to ST7920.

4 modes of read/write operation specified by RS and RW：

| RS | RW | Description |
|---|---|---|
| L | L | MPU write instruction to instruction register（IR） |
| L | H | MPU read busy flag（BF）and address counter（AC） |
| H | L | MPU write data to data register（DR） |
| H | H | MPU read data from data register（DR） |

**Busy Flag（BF）**
Internal operation is in progress when BF="1", ST7920 is in busy state. No new instruction will be accepted until BF="0". MPU must check BF to determine whether the internal operation is finished and new instruction can be sent.

**Address counter（AC）**
Address counter（AC）is used for address pointer of DDRAM/CGRAM/IRAM/GDRAM. (AC) can be set by instruction and after data read or write to the memories (AC) will increase or decrease by 1 according to the setting in "entry mode set". When RS= "0" and RW= "1" and E="1" the value of（AC）will output to DB6　DB0.

**16x16 character generation ROM (CGROM) and 8x16 half height ROM（HCGROM）**

ST7920 provides character generation ROM supporting 8192 16 x 16 character fonts and 126 8 x 16 alphanumeric characters. It is easy to support multi languages application such as Chinese and English. Two consecutive bytes are used to specify one 16x16 character or two 8x16 half-height characters. Character codes are written into DDRAM and the corresponding fonts are mapped from CGROM or HCGROM to the display drivers.

**Character generation RAM (CGRAM)**

ST7920 provides RAM to support user-defined fonts. Four sets of 16x16 bit map area are available. These user-defined fonts are displayed the same ways as CGROM fonts through writing character cod data to DDRAM.

**Display data RAM（DDRAM）**

There are 64x2 bytes for display data RAM area. Can store display data for 16 characters(16x16) by 4 lines or 32 characters(8x16) by 4 lines. However, only 2 lines can be displayed at a time. Character codes stored in DDRAM point to the fonts specified by CGROM, HCGROM and CGRAM. ST7920 display half height HCGROM fonts, user-defined CGRAM fonts and full 16x16 CGROM fonts. Data codes 0000H 0006H are for CGRAM user-defined fonts. Data codes 02H 7FH are for half height alpha numeric fonts. Data codes（A140 D75F）are for BIG5 code and (A1A0 F7FF) are for GB code.

1. display HCGROM fonts：Write 2 bytes data to DDRAM to display two 8x16 fonts. Each byte represents 1 character font. The data of each byte is 02H 7FH.
2. display CGRAM fonts：Write 2 bytes data to DDRAM to display one 16x16 font. Only 0000H, 0002H, 0004H, 0006H are allowed.
3. display CGROM fonts：Write 2 bytes data to DDRAM to display one 16x16 font.
   A140H D75FH are for (BIG5) code, A1A0H F7FFH are for (GB) code.

Higher byte（D15 D8）are written first and then lower byte（D7 D0）.

Refer to Table 5 for address map

CGRAM fonts and CGROM fonts can only be displayed in the start position of each address. (Refer toTable 4)

| 80 | | 81 | | 82 | | 83 | | 84 | | 85 | | 86 | | 87 | | 88 | | 89 | | 8A | | 8B | | 8C | | 8D | | 8E | | 8F | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| H | L | H | L | H | L | H | L | H | L | H | L | H | L | H | L | H | L | H | L | H | L | H | L | H | L | H | L | H | L | H | L |
| **S** | **i** | **t** | **r** | **o** | **n** | **i** | **x** | | **S** | **T** | **7** | **9** | **2** | **0** | | | | | | | | | | | | | | | | | |
| 矽 | | 創 | | 電 | | 子 | | . | . | 中 | | 文 | | 編 | | 碼 | | | ( | 正 | | 確 | | ) | | | | | | | |
| 矽 | | 創 | | 電 | | 子 | | . | . | . | 中 | | 文 | | 編 | | 碼 | | | | | | | | | | | | | | |

Incorrect position

**Table 4**

**Graphic RAM（GDRAM）**

Graphic display RAM supports 64x256 bits bit-mapped memory space. GDRAM address is set by writing 2 consecutive bytes for vertical address and horizontal address. Two-bytes data write to GDRAM for one address. Address counter will automatically increase by one for the next two-byte data. The procedure is as followings.

1. Set vertical address（Y）for GDRAM
2. Set horizontal address（X）for GDRAM
3. Write D15　D8 to GDRAM 中(first byte)
4. Write D7　D0 to GDRAM 中(second byte)

Graphic display memory map please refer to Table-8

**LCD driver**

LCD driver have 33 common and 64 segments to drive the LCD panel. Segment data from CGRAM /CGROM /HCGROM are shifted into the 64 bits segment latches to display. Extended segment driver ST7921 can be used to extend the segment drivers to 256.

| DDRAM data (char. code) | | | | | CGRAM Addr. | | | | | | CGRAM data (higher byte) | | | | | | | | CGRAM data (lower byte) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| B15~ B4 | B3 | B2 | B1 | B0 | B5 | B4 | B3 | B2 | B1 | B0 | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 0 | X | 00 | X | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| | | | | | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| | | | | | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| | | | | | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| | | | | | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| | | | | | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| | | | | | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| | | | | | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| | | | | | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| | | | | | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| | | | | | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| | | | | | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | X | 01 | X | 01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| | | | | | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| | | | | | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| | | | | | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| | | | | | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| | | | | | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| | | | | | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| | | | | | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| | | | | | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| | | | | | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| | | | | | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| | | | | | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| | | | | | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| | | | | | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| | | | | | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| | | | | | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 5： DDRAM data（character code）， CGRAM data / address map**

Note：

1. DDRAM data (character code) bit1 and bit2 are the same as CGRAM address bit4 and bit5.
2. CGRAM address bit0 to bit3 specify total 16 rows. Row16 is for cursor display. The data in row 16 will be logical OR to the cursor.
3. CGRAM data for each address is 16 bits.
4. DDRAM data to select CGRAM bit4 to bit15 must be “0”. Bit0 and bit3 value are “don’t care”.

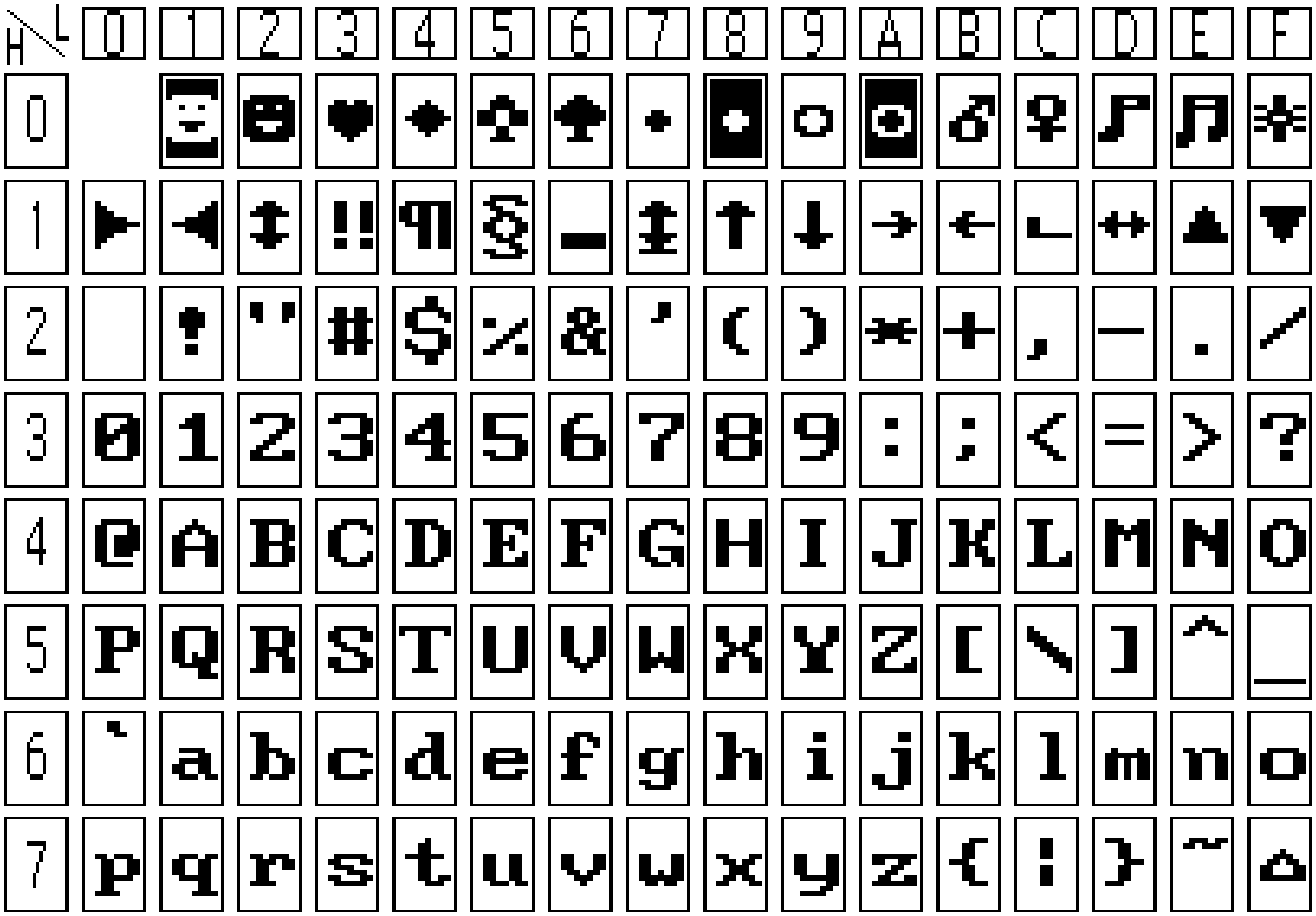


**Table 6 16x8 half-height characters**

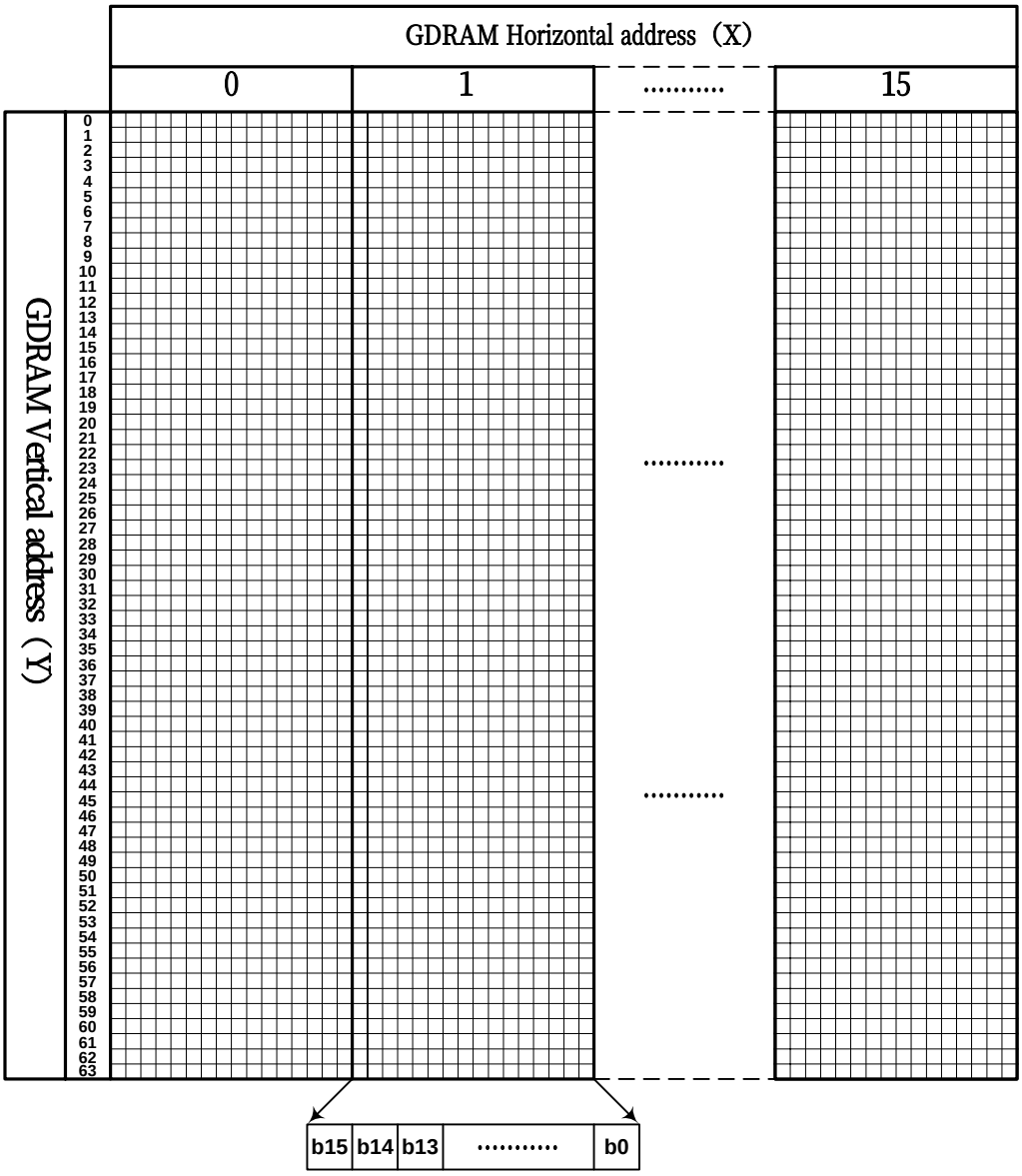



**Table 7 GDRAM display coordinates and corresponding address**

## Instructions

ST7920 offers basic instruction set and extended instruction set：

### Instruction set 1: (RE=0: basic instruction)

| Ins | code | | | | | | | | | | Description | Exec time (540KHZ) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RS | RW | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 | | |
| CLEAR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Fill DDRAM with "20H", and set DDRAM address counter （AC） to "00H" | 1.6 ms |
| HOME | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | X | Set DDRAM address counter（AC）to "00H", and put cursor to origin ；the content of DDRAM are not changed | 72us |
| ENTRY MODE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | I/D | S | Set cursor position and display shift when doing write or read operation | 72us |
| DISPLAY ON/OFF | 0 | 0 | 0 | 0 | 0 | 0 | 1 | D | C | B | D=1: display ON<br>C=1: cursor ON<br>B=1: blink ON | 72 us |
| CURSOR DISPLAY CONTROL | 0 | 0 | 0 | 0 | 0 | 1 | S/C | R/L | X | X | Cursor position and display shift control ；the content of DDRAM are not changed | 72 us |
| FUNCTION SET | 0 | 0 | 0 | 0 | 1 | DL | X | 0<br>RE | X | X | DL=1 8-BIT interface<br>DL=0 4-BIT interface<br>**RE=1: extended instruction**<br>**RE=0: basic instruction** | 72 us |
| SET CGRAM ADDR. | 0 | 0 | 0 | 1 | AC5 | AC4 | AC3 | AC2 | AC1 | AC0 | Set CGRAM address to address counter（AC）<br>**Make sure that in extended instruction SR=0 (scroll or RAM address select)** | 72 us |
| SET DDRAM ADDR. | 0 | 0 | 1 | 0<br>AC6 | AC5 | AC4 | AC3 | AC2 | AC1 | AC0 | Set DDRAM address to address counter（AC）<br>AC6 is fixed to 0 | 72 us |
| READ BUSY FLAG（BF） & ADDR. | 0 | 1 | BF | AC6 | AC5 | AC4 | AC3 | AC2 | AC1 | AC0 | Read busy flag（BF）for completion of internal operation, also Read out the value of address counter（AC） | 0 us |
| WRITE RAM | 1 | 0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Write data to internal RAM (DDRAM/CGRAM/GDRAM) | 72 us |
| READ RAM | 1 | 1 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Read data from internal RAM (DDRAM/CGRAM/GDRAM) | 72 us |

## Instruction set 2: (RE=1: extended instruction)

| Inst. | code | | | | | | | | | | description | Exec. time (540KHZ) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RS | RW | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 | | |
| STAND BY | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Enter stand by mode, any other instruction can terminate (Com1..32 halted) | 72 us |
| SCROLL or RAM ADDR. SELECT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | SR | SR=1: enable vertical scroll position<br>SR=0: enable CGRAM address**(basic instruction)** | 72 us |
| REVERSE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | R1 | R0 | Select 1 out of 4 line ( in DDRAM) and decide whether to reverse the display by toggling this instruction<br>**R1,R0 initial value is 00** | 72 us |
| EXTENDED FUNCTION SET | 0 | 0 | 0 | 0 | 1 | DL | X | 1<br>RE | G | 0 | DL=1 8-BIT interface<br>DL=0 4-BIT interface<br>**RE=1: extended instruction set**<br>**RE=0: basic instruction set**<br>G=1 :graphic display ON<br>G=0 :graphic display OFF | 72 us |
| SET IRAM or SCROLL ADDR | 0 | 0 | 0 | 1 | AC5 | AC4 | AC3 | AC2 | AC1 | AC0 | SR=1: AC5~AC0 the address of vertical scroll | 72 us |
| SET GRAPHIC RAM ADDR. | 0 | 0 | 1 | 0<br>0 | 0<br>AC5 | 0<br>AC4 | AC3<br>AC3 | AC2<br>AC2 | AC1<br>AC1 | AC0<br>AC0 | Set GDRAM address to address counter（AC）<br>First set vertical address and the horizontal address by consecutive writing<br>Vertical address range AC5...AC0<br>Horizontal address range AC3…AC0 | 72 us |

**Note：**

1. Make sure that ST7920 is not in busy state by reading the busy flag before sending instruction or data. If use delay loop instead please make sure the delay time is enough. Please refer to the instruction execution time.
2. “RE”is the selection bit of basic and extended instruction set. Each time when altering the value of RE it will remain. There is no need to set RE every time when using the same group of instruction set.

## Initial setting(Register flag) (RE=0: basic instruction)

| Inst. | code | | | | | | | | | | Description |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | RS | RW | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 | |
| ENTRY MODE SET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | I/D | S | Cursor move to right ,DDRAM address counter（AC） plus 1 |
| | | | | | | | | | **1** | **0** | |
| DISPLAY STATUS | 0 | 0 | 0 | 0 | 0 | 0 | 1 | D | C | B | Display, cursor and blink ALL OFF |
| | | | | | | | | **0** | **0** | **0** | |
| CURSOR DISPLAY SHIFT | 0 | 0 | 0 | 0 | 0 | 1 | S/C | R/L | X | X | No cursor or display shift operation |
| | | | | | | | **X** | **X** | | | |
| FUNCTION SET | 0 | 0 | 0 | 0 | 1 | DL | X | 0 RE | X | X | **8 BIT** MPU interface , basic instruction set |
| | | | | | | **1** | | **0** | | | |

## Initial setting(Register flag) (RE=1: extended instruction set)

| Inst. | code | | | | | | | | | | description |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | RS | RW | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 | |
| SCROLL OR RAM ADDR. SELECT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | SR | Allow IRAMaddress or set CGRAM address |
| | | | | | | | | | | **0** | |
| REVERSE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | R1 | R0 | Begin with normal and toggle to reverse |
| | | | | | | | | | **0** | **0** | |
| EXTENDED FUNCTION SET | 0 | 0 | 0 | 0 | 1 | DL | X | 1 RE | G | 0 | Graphic display OFF |
| | | | | | | | | | **0** | | |

## Description of basic instruction set

- **CLEAR**

| | RS | RW | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Code | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Fill DDRAM with "20H"(space code). And set DDRAM address counter （AC）to"00H". Set entry mode I/D bit to be "1". Cursor moves right and AC adds 1 after write or read operation.

- **HOME**

| | RS | RW | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Code | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | x |

Set DDRAM address counter（AC）to "00H". Cursor moves to origin. Then content of DDRAM is not changed.

- **ENTRY MODE SET**

| | RS | RW | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Code | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | I/D | S |

Set the cursor movement and display shift direction when doing write or read operation.

**I/D :address counter increase / decrease**

When I/D = "1", cursor moves right, DRAM address counter（AC）add by 1.

When I/D = "0", cursor moves left, DRAM address counter（AC）subtract by 1.

**S: Display shift**

| S | I/D | DESCRIPTION |
|---|---|---|
| H | H | Entire display shift left by 1 |
| H | L | Entire display shift right by 1 |

- **DISPLAY STATUS**

| | RS | RW | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Code | 0 | 0 | 0 | 0 | 0 | 0 | 1 | D | C | B |

Controls display, cursor and blink ON/OFF.

**D : Display ON/OFF control bit**

When D = "1", display ON

When D = "0",display OFF , the content of DDRAM is not changed

**C : Cursor ON/OFF control bit**

When C = "1", cursor ON.

When C = "0", cursor OFF.

**B : Blink ON/OFF control bit**

When B = "1", cursor position blink ON. Then display data in cursor position will blink.

When B = "0", cursor position blink OFF

- **CURSOR AND DISPLAY SHIFT CONTROL**

| | RS | RW | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Code | 0 | 0 | 0 | 0 | 0 | 1 | S/C | R/L | x | x |

Instruction to move the cursor or shift the entire display. The content of DDRAM is not changed.

| S/C | R/L | Description | AC Value |
|---|---|---|---|
| L | L | Cursor moves left by 1 | AC=AC-1 |
| L | H | Cursor moves right by 1 | AC=AC+1 |
| H | L | Display shift left by 1, cursor also follows to shift. | AC=AC |
| H | H | Display shift right by 1, cursor also follows to shift. | AC=AC |

- **FUNCTION SET**

| | RS | RW | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Code | 0 | 0 | 0 | 0 | 1 | DL | X | RE | x | x |

**DL : 4/8 BIT interface control bit**

When DL = "1", **8 BIT** MPU bus interface

When DL = "0", 為 **4 BIT** MPU bus interface

**RE : extended instruction set control bit**

When RE = "1", extended instruction set

When RE = "0", basic instruction set

**In same instruction cannot alter DL and RE at once. Make sure that change DL first then RE.**

- **SET CGRAM ADDRESS**

| | RS | RW | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Code | 0 | 0 | 0 | 1 | AC5 | AC4 | AC3 | AC2 | AC1 | AC0 |

Set CGRAM address to address counter（AC）

AC range is 00H..3FH

**Make sure that in extended instruction SR=0 (scroll address or RAM address select)**

- **SET DDRAM ADDRESS**

| | RS | RW | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Code | 0 | 0 | 1 | AC6 | AC5 | AC4 | AC3 | AC2 | AC1 | AC0 |

Set DDRAM address to address counter（AC）.

First line AC range is 80H..8FH

Second line AC range is 90H..9FH

Third line AC range is A0H..AFH

Fourth line AC range is B0H..BFH

Please note that only 2 lines can be display at a time.

- **READ BUSY FLAG（BF）AND ADDRESS**

| | RS | RW | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Code | 0 | 1 | BF | AC6 | AC5 | AC4 | AC3 | AC2 | AC1 | AC0 |

Read busy flag（BF）can check whether internal operation is finished. At the same time the value of address counter

（AC）is also read. When BF = “1” new instruction will not be accepted. Must wait for BF = “0” for new instruction.

- **WRITE DATA TO RAM**

| | RS | RW | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Code | 1 | 0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

Write data to internal RAM and alter the (AC) by 1

Each RAM address (CGRAM,DDRAM,IRAM…..) must write 2 consecutive bytes for 16 bit data. After the second byte the address counter will add or subtract by 1 according to the entry mode set control bit.

- **READ RAM DATA**

| | RS | RW | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Code | 1 | 1 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

Read data from internal RAM and alter the (AC) by 1

After address set to read (CGRAM,DDRAM,IRAM…..)a DUMMY READ is required.

There is no need to DUMMY READ for the following bytes unless a new address set instruction is issued.

## Description of extended instruction set

- **STAND BY**

| | RS | RW | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Code | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Instruction to enter stand by mode. Any other instruction follows this instruction can terminate stand by.

The content of DDRAM remain the same.

- **VERTICAL SCROLL OR RAM ADDRESS SELECT**

| | RS | RW | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Code | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | SR |

When SR = "1", the vertical scroll address set is enabled.

When SR = "0", the IRAM address set **(extended instruction)** and CGRAM address set**(basic instruction)** is enabled.

- **REVERSE**

| | RS | RW | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Code | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | R1 | R0 |

Select 1 out of 4 lines to reverse the display and to toggle the reverse condition by repeating this instruction.

R1,R0 initial vale is 00. When set the first time the display is reversed and set the second time the display become normal.

| R1 | R0 | Description |
|---|---|---|
| L | L | First line normal or reverse |
| L | H | Second line normal or reverse |
| H | L | Third line normal or reverse |
| H | H | Fourth line normal or reverse |

Please note that only 2 lines out of 4 line display data can be displayed.

- **EXTENED FUNCTION SET**

| | RS | RW | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Code | 0 | 0 | 0 | 0 | 1 | DL | x | RE | G | x |

**DL : 4/8 BIT interface control bit**

When DL = "1", **8 BIT** MPU interface

When DL = "0", **4 BIT** MPU interface

**RE : extended instruction set control bit**

When RE = "1", extended instruction set

When RE = "0", basic instruction set

**G : Graphic display control bit**

When G = "1", graphic display ON

When G = "0", Graphic display OFF

**In same instruction cannot alter DL, RE and G at once. Make sure that change DL or G first and then RE.**

- **SET SCROLL ADDRESS**

| | RS | RW | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Code | 0 | 0 | 0 | 1 | AC5 | AC4 | AC3 | AC2 | AC1 | AC0 |

SR=1: AC5~AC0 is vertical scroll displacement address

- **SET GRAPHIC RAM ADDRESS**

| | RS | RW | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Code | 0 | 0 | 1 | 0 | AC5 | AC4 | AC3 | AC2 | AC1 | AC0 |

Set GDRAM address to address counter（AC）.

First set vertical address and then horizontal address(write 2 consecutive bytes to complete vertical and horizontal address set)

Vertical address range is AC5...AC0

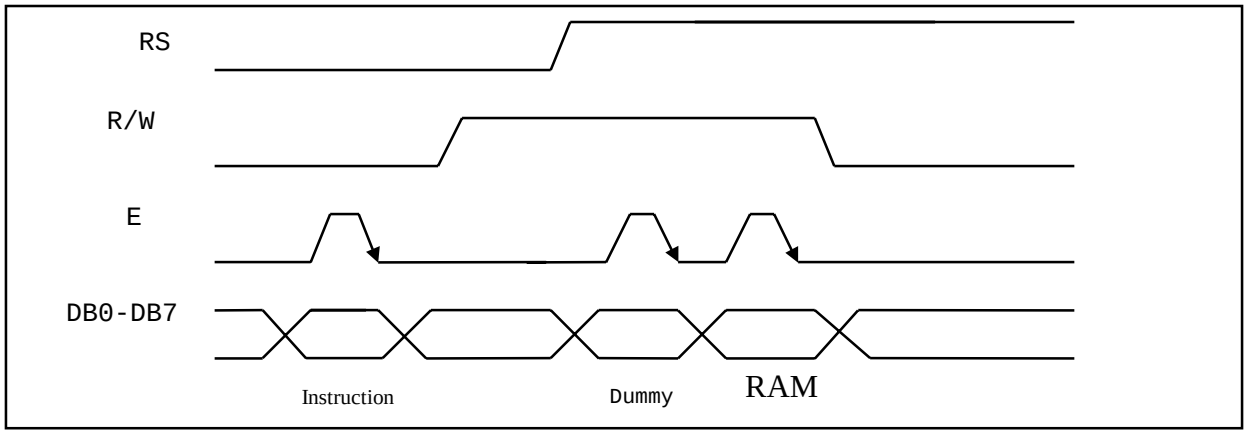
Horizontal address range is AC3…AC0

The address counter（AC）of graphic RAM(GRAM) only increment after write for horizontal address. After horizontal address =0FH it will automatically back to 00H. However, the vertical address will not increase as the result of the same action.

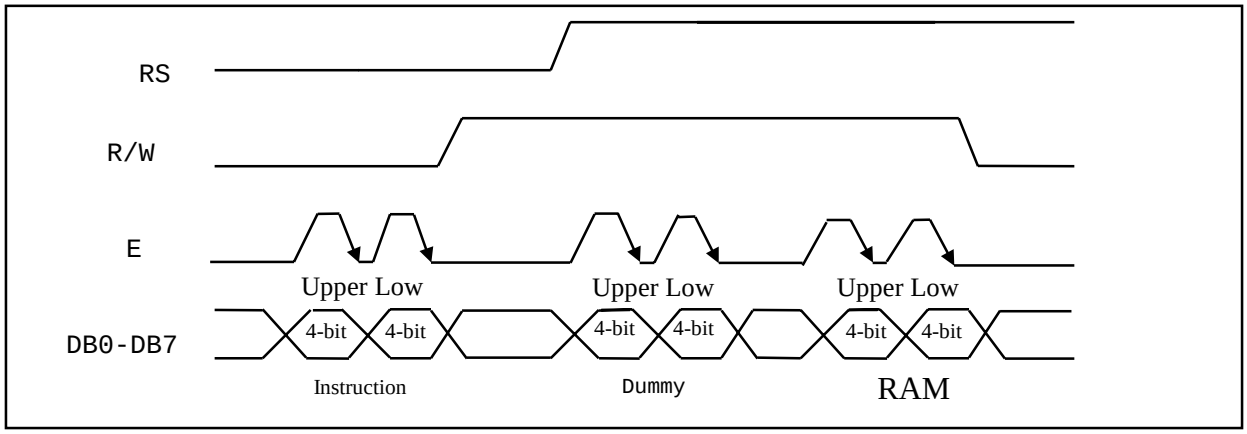
## Parallel interface :

ST7920 is in parallel mode by pulling up PSB pin. And can select 8 bit or 4-bit bus interface by function set instruction DL control bit. MPU can control ( RS , RW , E , and DB0..DB7 ) pins to complete the data transmission.

In 4-bit transfer mode, every 8 bits data or instruction is separated into 2 parts. Higher 4 bits（DB7~DB4）data will transfer First and placed into data pins（DB7~DB4）. Lower 4 bits（DB3~DB0）data will transfer second and placed into data pins （DB7~DB4）. (DB3~DB0) data pins are not used.




Timing Diagram of 8-bit Parallel Bus Mode Data Transfer




Timing Diagram of 4-bit Parallel Bus Mode Data Transfer

## Serial interface :

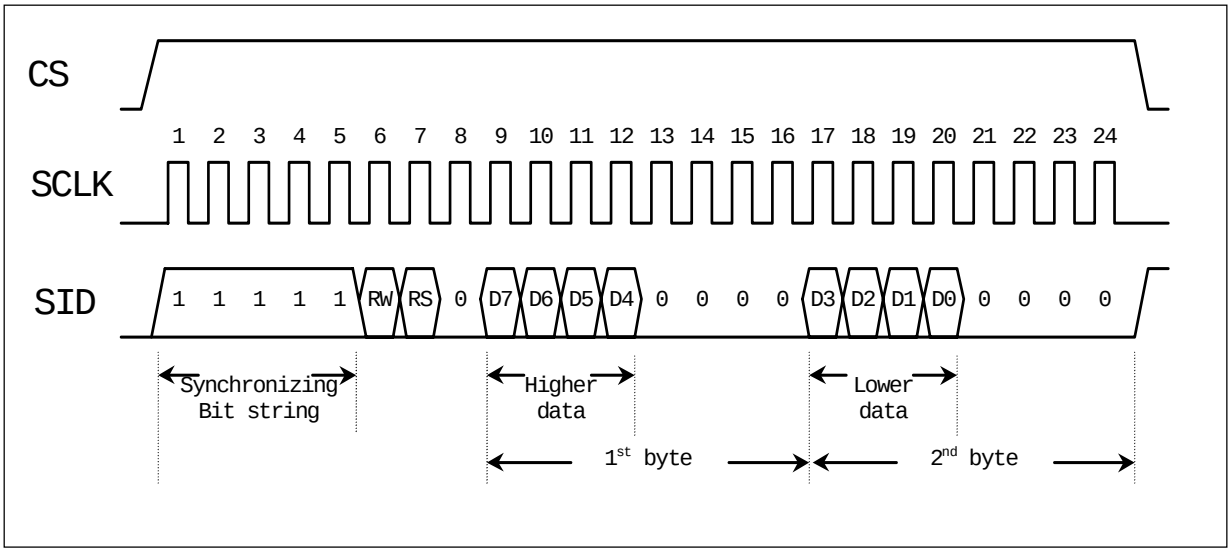
ST7920 is in serial interface mode when pull down PSB pin. Two pins (SCLK and SID) are used to complete the data transfer. Only write data is available.

When connecting several ST7920, chip select(CS)must be used. Only when(CS)is high the serial clock(SCLK)can be accepted. On the other hand, when chip select（CS） is low ST7920 serial counter and data will be reset. Transmission will be terminated and data will be cleared. Serial transfer counter is set to the first bit. For a minimal system with only one ST7920 and one MPU, only SCLK and SID pins are necessary. CS pin should pull to high.

ST7920's serial clock（SCLK） is asynchronous to the internal clock and is generated by MPU. When multiple instruction/data is transferred instruction execution time must be considered. Must wait for the previous instruction to finish before sending the next. ST7920 has no internal instruction buffer area.

When starting a transmission a start byte is required. It consists of 5 consecutive 〝1〞 （sync character）. Serial transfer counter will be reset and synchronized. Following 2 bits for read/write（RW）and register/data select（RS）. Last 4 bits is filled by 〝0〞。

After receiving the sync character and RW and RS bits, every 8 bits instruction/data will be separated into 2 groups. Higher 4 bits（DB7~DB4）will be placed in first section followed by 4 〝0〞 . And lower 4 bits（DB3~DB0）will be placed in second section followed by 4 〝0〞 .




Timing Diagram of Serial Mode Data Transfer

## 8051 demo program for serial interface

```
;---------------------------------------------------------------
;   Write data from A into INSTRUCTION Register
;---------------------------------------------------------------
WRINS:
        SETB    CS
        SETB    SID         ; SID = 1
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        CLR     SID         ; SID = 0
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        MOVBIT  SID, A.7    ; SID = A.7
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        MOVBIT  SID, A.6    ; SID = A.6
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        MOVBIT  SID, A.5    ; SID = A.5
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        MOVBIT  SID, A.4    ; SID = A.4
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        CLR     SID         ; SID = 0
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        MOVBIT  SID, A.3    ; SID = A.3
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        MOVBIT  SID, A.2    ; SID = A.2
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        MOVBIT  SID, A.1    ; SID = A.1
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        MOVBIT  SID, A.0    ; SID = A.0
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        CLR     SID         ; SID = 0
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        CLR     CS
        CALL    DLY8
        RET
```

```
;-------------------------------------------------
;   Write data from A into DATA Register
;-------------------------------------------------
WRDATA:
        SETB    CS
        SETB    SID         ; SID = 1
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        CLR     SID         ; SID = 0
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SID         ; SID = 1
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        CLR     SID         ; SID = 0
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        MOVBIT  SID, A.7    ; SID = A.7
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        MOVBIT  SID, A.6    ; SID = A.6
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        MOVBIT  SID, A.5    ; SID = A.5
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        MOVBIT  SID, A.4    ; SID = A.4
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        CLR     SID         ; SID = 0
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        MOVBIT  SID, A.3    ; SID = A.3
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        MOVBIT  SID, A.2    ; SID = A.2
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        MOVBIT  SID, A.1    ; SID = A.1
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        MOVBIT  SID, A.0    ; SID = A.0
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        CLR     SID         ; SID = 0
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        SETB    SCLK        ; READ DATA FROM SID
        CLR     SCLK
        CLR     CS
        CALL    DLY8
        RET
```

**Application circuit for testing CGROM and HCGROM:**

We can use the function of “CHECK SUM” to check the CGROM is right or error.

See the following notes: Useing IC Pad (Pin4→CLK, Pin5→TT1, Pin6→TT2) to do the “CHECK SUM” function.

The application circuit is at Page49.

**Timing Diagram for checking CGROM (TT1=0, TT2=1)**

The ST7920 check sum process:

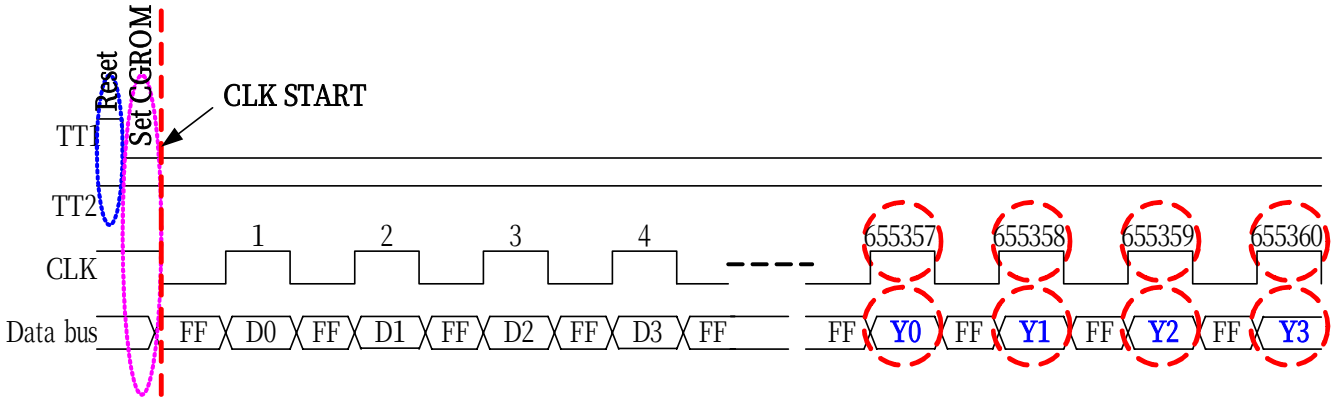
In the first place: Resetting the internal counter (set TT1 and TT2 to Height)

In the second place: Setting CGROM mode (set TT1 to Low, TT2 to Height).

In the third place: CLK starts to count 655360 times.

In the final place: Finishing the counting, read the last four bytes to CHECK SUM (reading only when the CLK is Height).

ST7920 check sum circuit: Data is available when CLK is height; if CLK is low then the data is always FFH. The last four bytes are Y0, Y1, Y2, and Y3.




The table below is a comparing table of CGROM for different versions.

| | Version (Font) | CGROM Last four bytes | | | |
|---|---|---|---|---|---|
| | | Y0 | Y1 | Y2 | Y3 |
| 1 | Big5 (0A) | 38 | 88 | CC | F1 |
| 2 | GB (0B) | 9D | 81 | 79 | 29 |
| 3 | 0C | FD | 6F | B5 | 85 |

**Timing Diagram for checking HCGROM (TT1=1, TT2=0)**

The ST7920 check sum process:

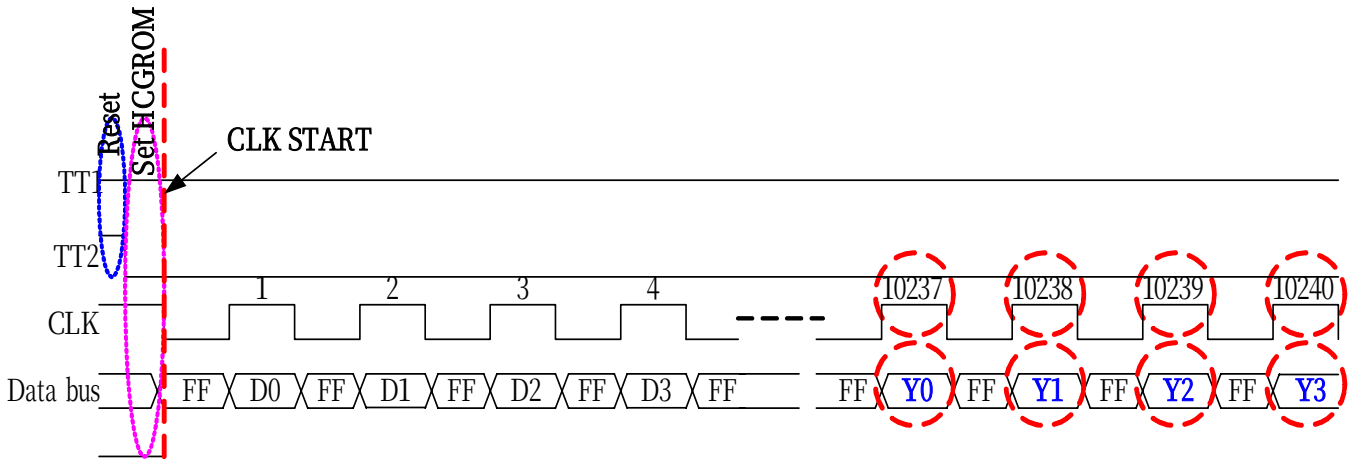
In the first place: Resetting the internal counter (set TT1 and TT2 to Height)

In the second place: Setting CGROM mode (set TT1 to Height, TT2 to Low).

In the third place: CLK starts to count 10240 times.

In the final place: Finishing the counting, read the last four bytes to CHECK SUM (reading only when the CLK is Height).

ST7920 check sum circuit: Data is available when CLK is height; if CLK is low then the data is always FFH. The last four bytes are Y0, Y1, Y2, and Y3.



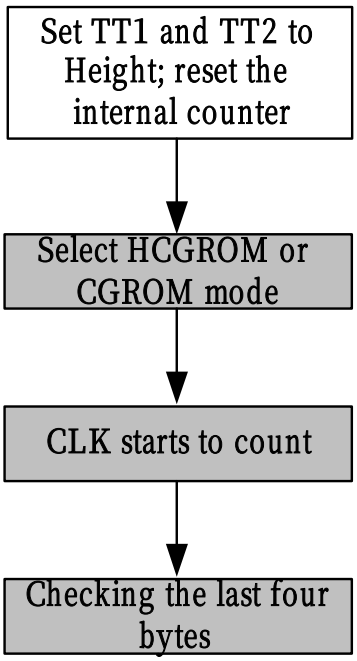

The table below is a comparing table of HCGROM for different versions.

| | Version (Font) | HCGROM last four bytes | | | |
|---|---|---|---|---|---|
| | | Y0 | Y1 | Y2 | Y3 |
| 1 | Big5 (0A) | B5 | 11 | B5 | 11 |
| 2 | GB (0B) | B5 | 11 | B5 | 11 |
| 3 | 0C | B5 | 11 | B5 | 11 |

Testing Step:
1. Composing TT1 and TT2 to make the ‘Reset’ action, and clear the internal counter.
2. Selecting the test mode by setting TT1 and TT2 (CGROM or HCGROM).
3. After setp1 and setp2, entering some impulse signals through Pin4 (CLK).
4. Reading the CHECK SUM data through D0 to D7.
5. Comparing CHECK SUM with the Code Table (upper table) to check if the data is correct or not.

| TT1 | TT2 | No. of counts | Status |
|---|---|---|---|
| 1 | 1 | -- | RESET |
| 0 | 1 | 655360 | CGROM |
| 1 | 0 | 10240 | HGROM |

Test process flow:



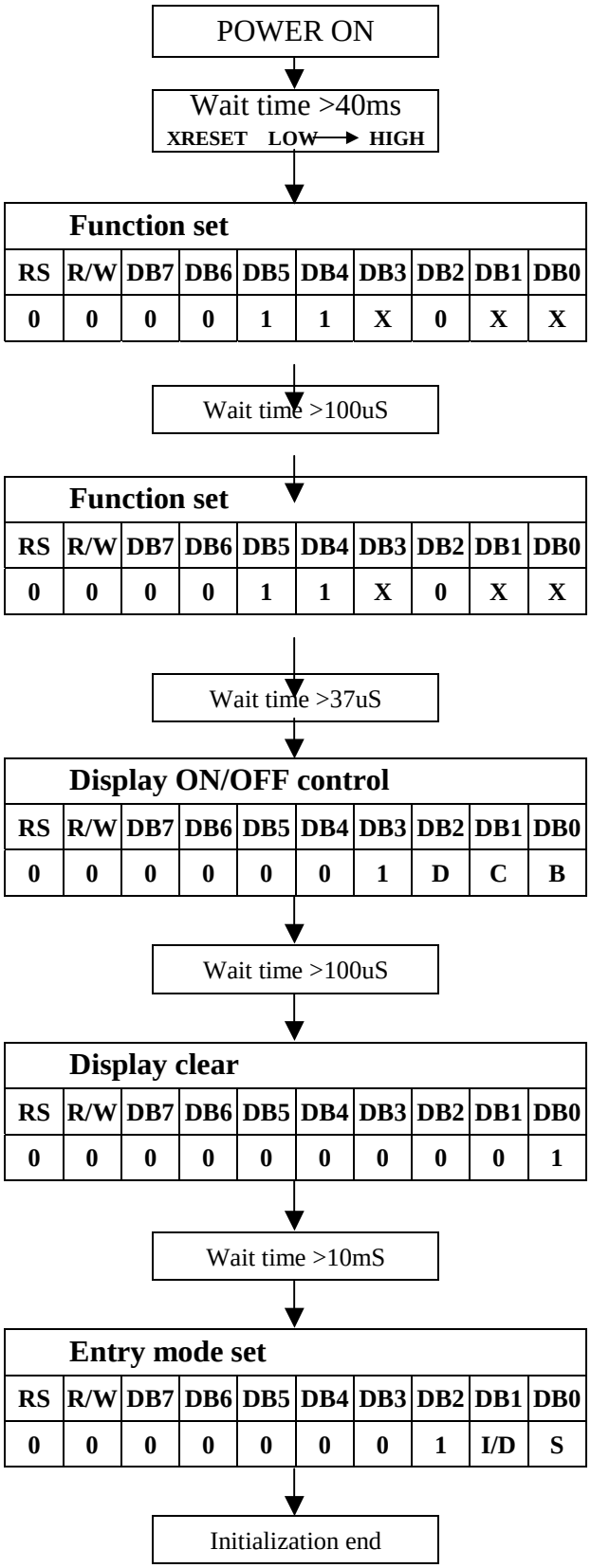
8051 CGROM、HCGROM illustrative test program

```
        ;*********************************;
        ;*      CHECK_ROM                *;
        ;*********************************;
        ;*********************************;
        ;*   Definition of outside Pin *;
        ;*********************************;
CLK     REG     P3.5                    ;
TT1     REG     P3.0                    ;
TT2     REG     P3.1                    ;
TT3     REG     P3.2                    ;CHECK CGROM FLAG
TT4     REG     P3.3                    ;CHECK HCGROM FLAG
TT5     REG     P3.4                    ;ERROR FLAG
        ;*********************************;
        ;*  Definition of internal RAM *;
        ;*********************************;
STACK           EQU     6FH             ;
FUNC            EQU     20H             ;
        ;*********************************;
        ;*         Interrupt set        *;
        ;*********************************;
        ORG     00H                     ;
        AJMP    RESET                   ;
        ;*********************************;
        ;*      PROGRAM START           *;
        ;*********************************;
RESET:  MOV     SP,#STACK               ;
        MOV     P1,#FFH                 ;
        MOV     P3,#FFH                 ;
        ;*********************************;
        ;*      CHECK_CGROM             *;
        ;*********************************;
        ;*********************************;
        ;*     Initial setting          *;
        ;*********************************;
CGROM:  SETB    TT1                     ;
        SETB    TT2                     ;TT1,TT2 SET HIGH (RESET)
        CALL    DELAY_100US             ;Wait Reset 100us
        CLR     TT1                     ;TT1=LOW TT2=HIGH ( CHECK CGROM)
        SETB    CLK                     ;
        CALL    DELAY_100US             ;
        ;*********************************;
        ;*      start counter           *;
        ;*********************************;
        MOV     R3,#9                   ;
CN4:    MOV     R2,#0                   ;<----
CN3:    MOV     R1,#0                   ;     |
CN2:    CLR     CLK                     ;     |
        SETB    CLK                     ;     |
        DJNZ    R1,CN2                  ;     |
        DJNZ    R2,CN3                  ;     |
        DJNZ    R3,CN4                  ;     |
                                        ;     |
        MOV     R3,#0                   ;     |
CN5:    MOV     R2,#255                 ;     |
CN6:    CLR     CLK                     ;     |
        SETB    CLK                     ;     |
        DJNZ    R2,CN6                  ;     |
        DJNZ    R3,CN5                  ;     |
                                        ;     |
        MOV     R3,#63                  ;     |
CN7:    MOV     R2,#2                   ;     |
CN8:    MOV     R1,#2                   ;     |
CN9:    CLR     CLK                     ;     |
        SETB    CLK                     ;     |
        DJNZ    R1,CN9                  ;     |
        DJNZ    R2,CN8                  ;     |
        DJNZ    R3,CN7                  ;     |
        CLR     CLK                     ;     |
```

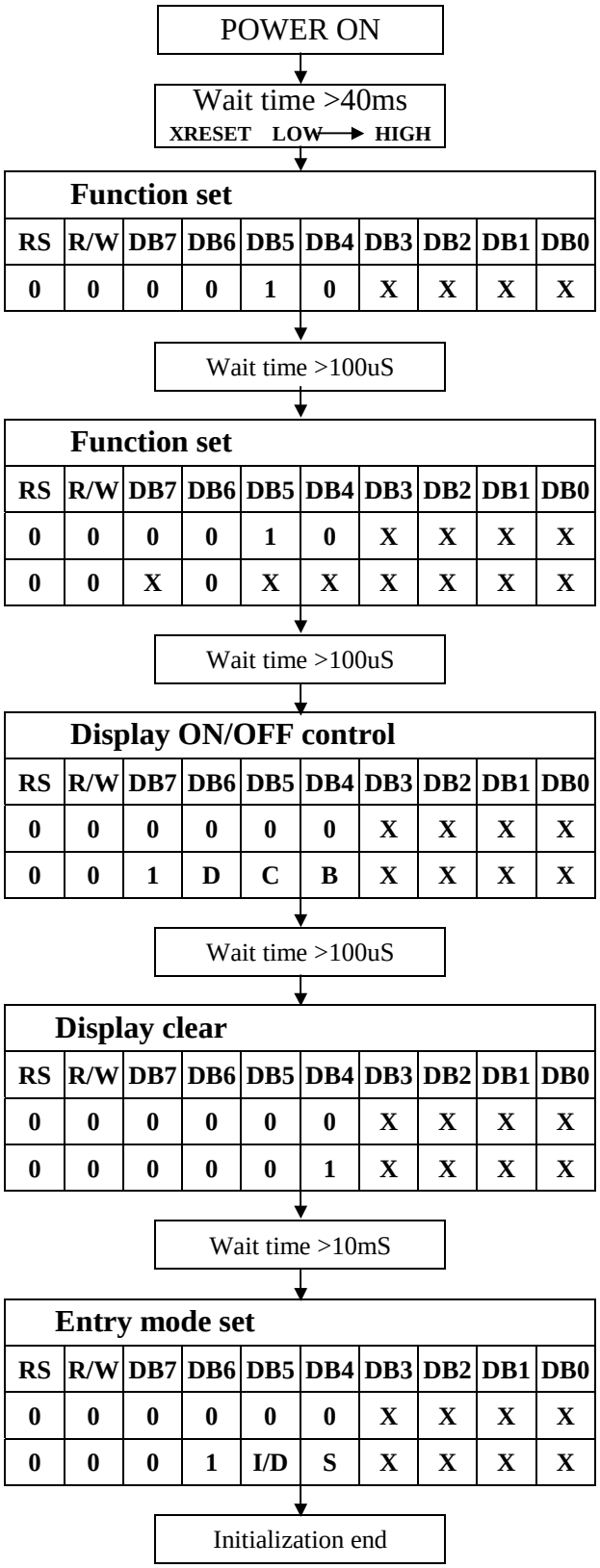
```
        SETB    CLK                     ;     |
        CLR     CLK                     ;     |
        SETB    CLK                     ;<----  Counter 655356
        ;-------------------------------;
        CLR     CLK                     ;Counter 655357
        SETB    CLK                     ;
        MOV     A,P1                    ;A=Y0
        CJNE    A,#FDH,ERRORC           ;COMPARE Y0 DATA
        CLR     CLK                     ;Counter 655358
        SETB    CLK                     ;
        MOV     A,P1                    ;A=Y1
        CJNE    A,#6FH,ERRORC           ;COMPARE Y1 DATA
        CLR     CLK                     ;Counter 655359
        SETB    CLK                     ;
        MOV     A,P1                    ;A=Y2
        CJNE    A,#B5H,ERRORC           ;COMPARE Y2 DATA

        CLR     CLK                     ;Counter 655360
        SETB    CLK                     ;
        MOV     A,P1                    ;A=Y3
        CJNE    A,#85H,ERRORC           ;COMPARE Y3 DATA
        CLR     CLK                     ;
        CLR     TT3                     ;IF OK CLR TT3
        CALL    HCGROM                  ;
ERRORC:                                 ;
        CLR     TT5                     ;IF CGROM CHECK ERROR CLR TT5
;---------------------------------------;
        ;*********************************;
        ;*      CHECK_HCGROM           *;
        ;*********************************;
        ;*********************************;
        ;*      Initial setting        *;
        ;*********************************;
HCGROM: SETB    TT1                     ;
        SETB    TT2                     ;TT1,TT2 SET HIGH (RESET)
        CALL    DELAY_100US             ;Wait Reset 100us
        CLR     TT2                     ;TT2=LOW TT1=HIGH ( CHECK HCGROM)
        SETB    CLK                     ;
        CALL    DELAY_100US             ;
        ;*********************************;
        ;*      start counter          *;
        ;*********************************;
        MOV     R3,#9                   ;
N4:     MOV     R2,#32                  ;<----
N3:     MOV     R1,#32                  ;     |
N2:     CLR     CLK                     ;     |
        SETB    CLK                     ;     |
        DJNZ    R1,N2                   ;     |
        DJNZ    R2,N3                   ;     |
        DJNZ    R3,N4                   ;     |
                                        ;     |
        MOV     R3,#32                  ;     |
N5:     MOV     R2,#31                  ;     |
N6:     CLR     CLK                     ;     |
        SETB    CLK                     ;     |
        DJNZ    R2,N6                   ;     |
        DJNZ    R3,N5                   ;     |
                                        ;     |
        MOV     R2,#30                  ;     |
N7:     CLR     CLK                     ;     |
        SETB    CLK                     ;     |
        DJNZ    R2,N7                   ;     |
;---------------------------------------;<----  Counter 10236
        CLR     CLK                     ;Counter 10237
        SETB    CLK                     ;
        MOV     A,P1                    ;A=Y0
        CJNE    A,#B5H,ERROR            ;COMPARE Y0 DATA
        CLR     CLK                     ;Counter 10238
        SETB    CLK                     ;
        MOV     A,P1                    ;A=Y1
```

```
        CJNE    A,#11H,ERROR            ;COMPARE Y1 DATA
        CLR     CLK                     ;Counter 10239
        SETB    CLK                     ;
        MOV     A,P1                    ;A=Y2
        CJNE    A,#B5H,ERROR            ;COMPARE Y2 DATA
        CLR     CLK                     ;Counter 10240
        SETB    CLK                     ;
        MOV     A,P1                    ;A=Y3
        CJNE    A,#11H,ERROR            ;COMPARE Y3 DATA
        CLR     CLK                     ;
        CLR     TT4                     ;IF HCGROM CHECK OK THEN CLR TT4
        AJMP    $                       ;
ERROR:                                  ;
        CLR     TT5                     ;IF HCGROM CHECK ERROR THEN CLR TT5
        AJMP    $                       ;
        ;*******************************;
        ;*      DELAY TIME 100US       *;
        ;*******************************;
DELAY_100US                             ;
DEL_10  MOV     R6,#5                   ;
DEL_9   MOV     R7,#3                   ;
        DJNZ    R7,$                    ;
        DJNZ    R6,DEL_9                ;
        RET                             ;
END                                     ;
```

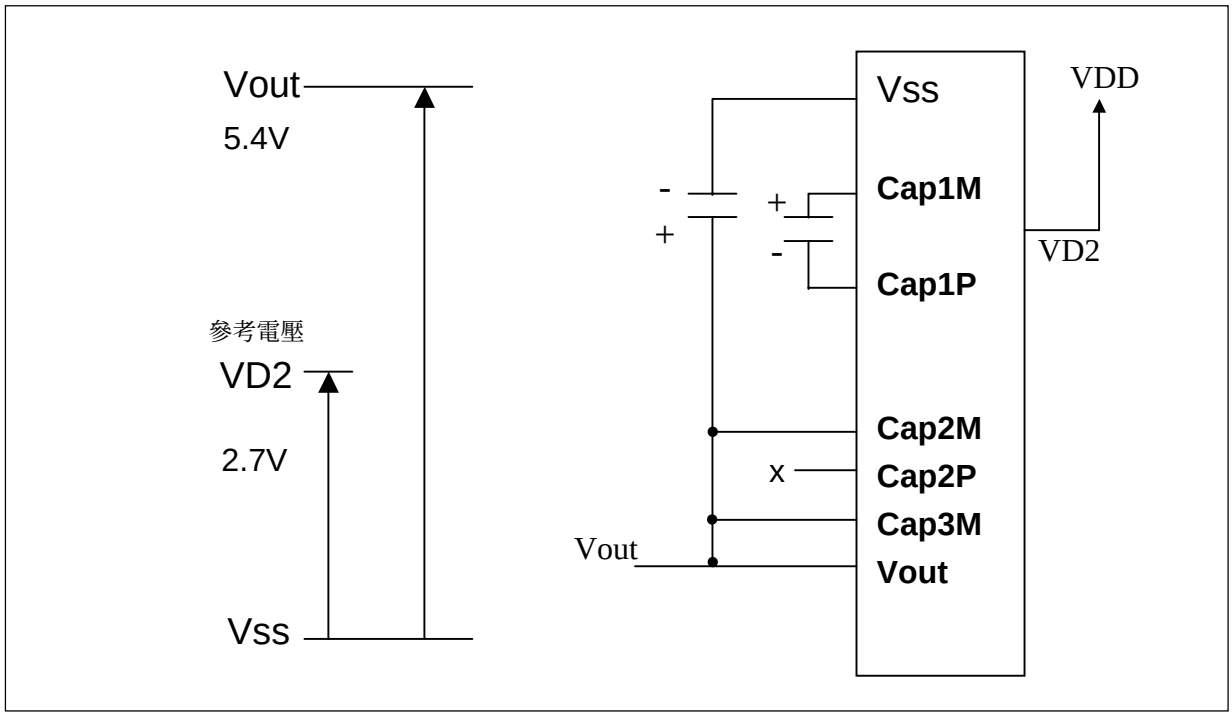
**8 bit interface：**



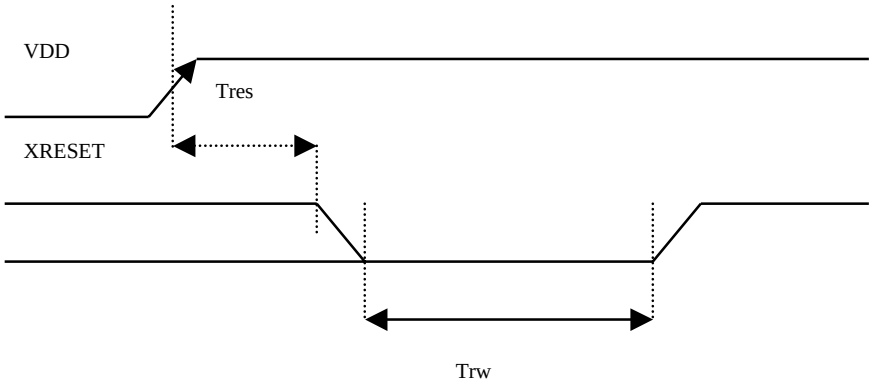
**4 bit interface：**



**Built in voltage booster**




**External reset timing**

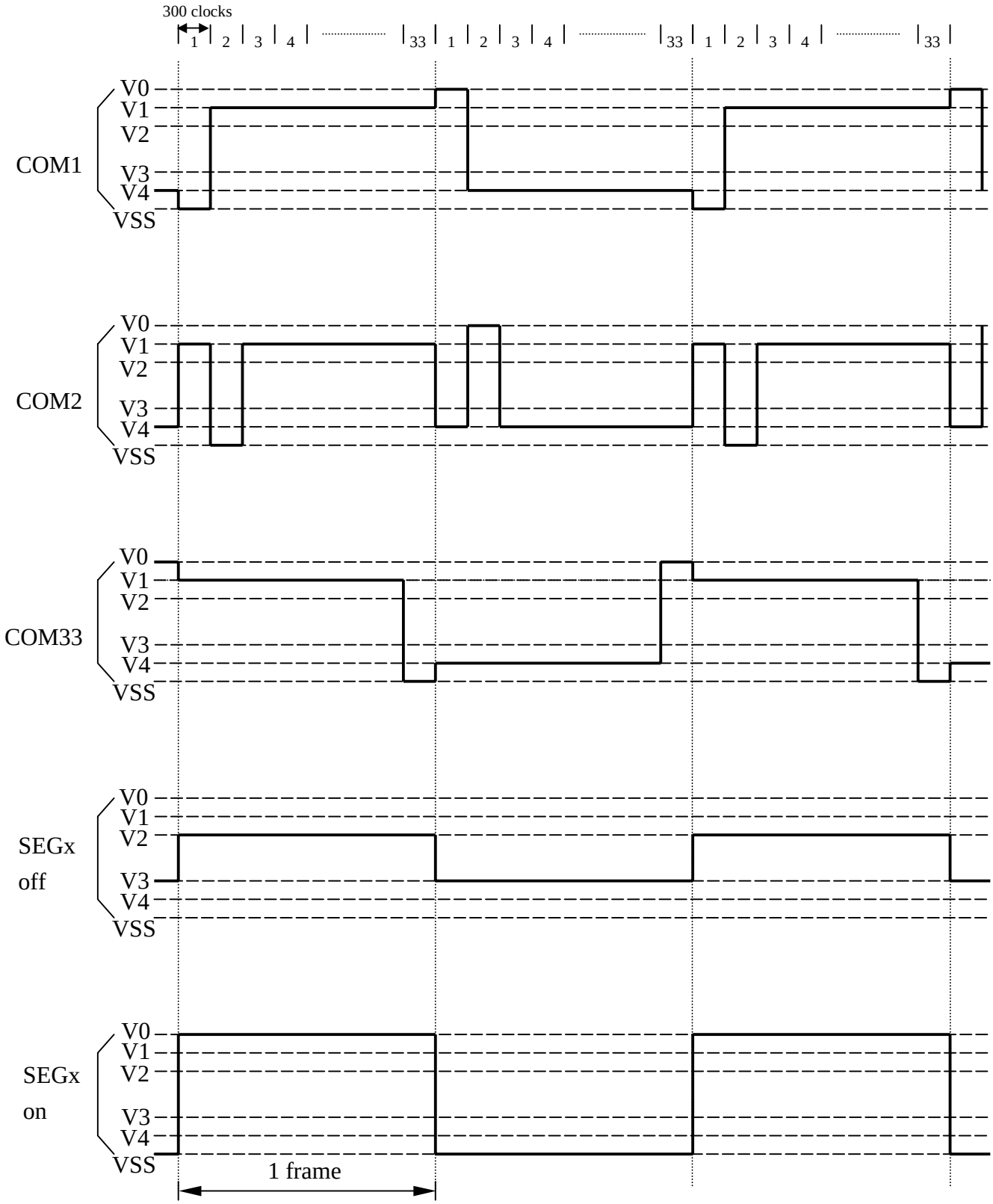



| | | |
|---|---|---|
| XRESET pulse width | Trw | 10us |
| RESET start time | Tres | 50ns |

## LCD driving wave form (1/33 duty , 1/5 bias )

When oscillation frequency is 540KHZ, 1 clock cycle time = 1.85us

1 frame = 1.85us x 300 x 33 = 18315us=18.3ms



## Absolute Maximum Ratings

| Characteristics | Symbol | Value |
|---|---|---|
| Power Supply Voltage | $V_{DD}$ | -0.3V to +5.5V |
| LCD Driver Voltage | $V_{LCD}$ | -0.3V to +7.0V |
| Input Voltage | $V_{IN}$ | -0.3V to $V_{DD}$+0.3V |
| Operating Temperature | $T_A$ | -20°C to + 85°C |
| Storage Temperature | $T_{STO}$ | -55°C to + 125°C |

## DC Characteristics ( $T_A$ = 25°C , $V_{DD}$ = 2.7 V – 4.5 V )

| Symbol | Characteristics | Test Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| $V_{DD}$ | Operating Voltage | - | 2.7 | - | 5.5 | V |
| $V_{LCD}$ | LCD Voltage | V0-$V_{SS}$ | 3.0 | - | 7 | V |
| $I_{CC}$ | Power Supply Current | $f_{OSC}$ = 530KHz, $V_{DD}$=3.0V Rf=18KΩ | - | 0.20 | 0.45 | mA |
| $V_{IH1}$ | Input High Voltage (Except OSC1) | - | 0.7$V_{DD}$ | - | $V_{DD}$ | V |
| $V_{IL1}$ | Input Low Voltage (Except OSC1) | - | - 0.3 | - | 0.6 | V |
| $V_{IH2}$ | Input High Voltage (OSC1) | - | $V_{DD}$ – 1 | - | $V_{DD}$ | V |
| $V_{IL2}$ | Input Low Voltage (OSC1) | - | - | - | 1.0 | V |
| $V_{OH1}$ | Output High Voltage (DB0 - DB7) | $I_{OH}$ = -0.1mA | 0.8$V_{DD}$ | - | $V_{DD}$ | V |
| $V_{OL1}$ | Output Low Voltage (DB0 - DB7) | $I_{OL}$ = 0.1mA | - | - | 0.1 | V |
| $V_{OH2}$ | Output High Voltage (Except DB0 - DB7) | $I_{OH}$ = -0.04mA | 0.8$V_{DD}$ | - | $V_{DD}$ | V |
| $V_{OL2}$ | Output Low Voltage (Except DB0 - DB7) | $I_{OL}$ = 0.04mA | - | - | 0.1$V_{DD}$ | V |
| $I_{LEAK}$ | Input Leakage Current | $V_{IN}$ = 0V to $V_{DD}$ | -1 | - | 1 | µA |
| $I_{PUP}$ | Pull Up MOS Current | $V_{DD}$ = 3V | 22 | 27 | 32 | µA |

## DC Characteristics ( $T_A = 25^oC$, $V_{DD} = 4.5\ V - 5.5\ V$ )

| Symbol | Characteristics | Test Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| $V_{DD}$ | Operating Voltage | - | 4.5 | - | 5.5 | V |
| $V_{LCD}$ | LCD Voltage | V0-$V_{SS}$ | 3.0 | - | 7 | V |
| $I_{CC}$ | Power Supply Current | $f_{OSC}$ = 540KHz, $V_{DD}$=5V Rf=33KΩ | - | 0.45 | 0.75 | mA |
| $V_{IH1}$ | Input High Voltage (Except OSC1) | - | 0.7$V_{DD}$ | - | $V_{DD}$ | V |
| $V_{IL1}$ | Input Low Voltage (Except OSC1) | - | -0.3 | - | 0.6 | V |
| $V_{IH2}$ | Input High Voltage (OSC1) | - | $V_{DD}$-1 | - | $V_{DD}$ | V |
| $V_{IL2}$ | Input Low Voltage (OSC1) | - | - | - | 1.0 | V |
| $V_{OH1}$ | Output High Voltage (DB0 - DB7) | $I_{OH}$ = -0.1mA | 0.8$V_{DD}$ | - | $V_{DD}$ | V |
| $V_{OL1}$ | Output Low Voltage (DB0 - DB7) | $I_{OL}$ = 0.1mA | - | - | 0.4 | V |
| $V_{OH2}$ | Output High Voltage (Except DB0 - DB7) | $I_{OH}$ = -0.04mA | 0.8$V_{DD}$ | - | $V_{DD}$ | V |
| $V_{OL2}$ | Output Low Voltage (Except DB0 - DB7) | $I_{OL}$ = 0.04mA | - | - | 0.1$V_{DD}$ | V |
| $I_{LEAK}$ | Input Leakage Current | $V_{IN}$ = 0V to $V_{DD}$ | -1 | - | 1 | μA |
| $I_{PUP}$ | Pull Up MOS Current | $V_{DD}$ = 5V | 75 | 80 | 85 | μA |

## AC Characteristics ($T_A$ = 25°C, $V_{DD}$ = 4.5V) Parallel Mode Interface

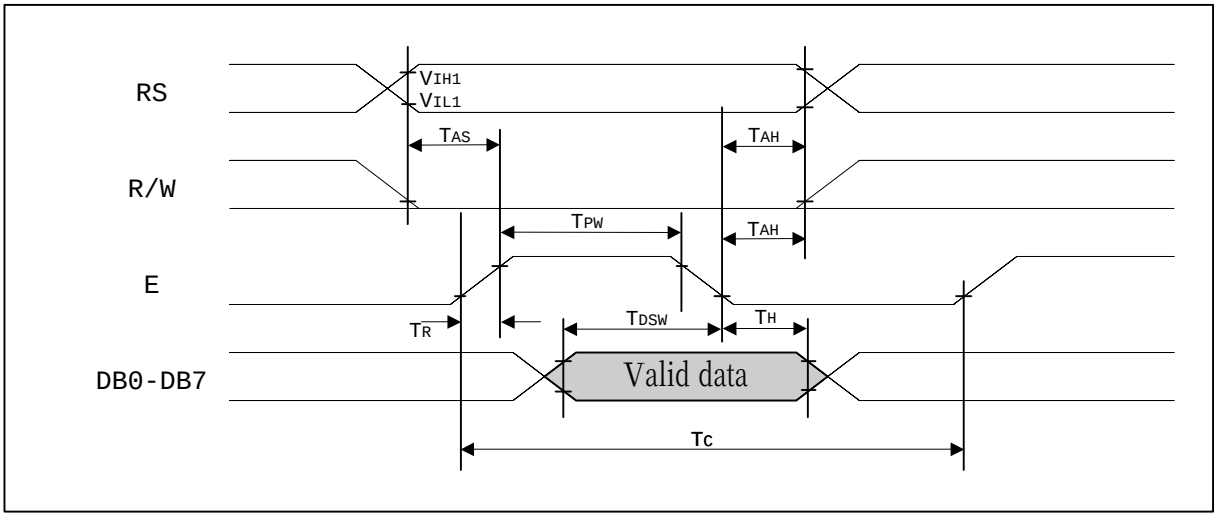
| Symbol | Characteristics | Test Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| *Internal Clock Operation* | | | | | | |
| $f_{OSC}$ | OSC Frequency | R = 33KΩ | 480 | 540 | 600 | KHz |
| *External Clock Operation* | | | | | | |
| $f_{EX}$ | External Frequency | - | 480 | 540 | 600 | KHz |
| | Duty Cycle | - | 45 | 50 | 55 | % |
| $T_R$,$T_F$ | Rise/Fall Time | - | - | - | 0.2 | μs |
| *Write Mode (Writing data from MPU to ST7920)* | | | | | | |
| $T_C$ | Enable Cycle Time | Pin E | 1200 | - | - | ns |
| $T_{PW}$ | Enable Pulse Width | Pin E | 140 | - | - | ns |
| $T_R$,$T_F$ | Enable Rise/Fall Time | Pin E | - | - | 25 | ns |
| $T_{AS}$ | Address Setup Time | Pins: RS,RW,E | 10 | - | - | ns |
| $T_{AH}$ | Address Hold Time | Pins: RS,RW,E | 20 | - | - | ns |
| $T_{DSW}$ | Data Setup Time | Pins: DB0 - DB7 | 40 | - | - | ns |
| $T_H$ | Data Hold Time | Pins: DB0 - DB7 | 20 | - | - | ns |
| *Read Mode (Reading Data from ST7920 to MPU)* | | | | | | |
| $T_C$ | Enable Cycle Time | Pin E | 1200 | - | - | ns |
| $T_{PW}$ | Enable Pulse Width | Pin E | 140 | - | - | ns |
| $T_R$,$T_F$ | Enable Rise/Fall Time | Pin E | - | - | 25 | ns |
| $T_{AS}$ | Address Setup Time | Pins: RS,RW,E | 10 | - | - | ns |
| $T_{AH}$ | Address Hold Time | Pins: RS,RW,E | 20 | - | - | ns |
| $T_{DDR}$ | Data Delay Time | Pins: DB0 - DB7 | - | - | 100 | ns |
| $T_H$ | Data Hold Time | Pins: DB0 - DB7 | 20 | - | - | ns |
| *Interface Mode with LCD Driver(ST7921)* | | | | | | |
| $T_{CWH}$ | Clock Pulse with High | Pins: CL1, CL2 | 800 | - | - | ns |
| $T_{CWL}$ | Clock Pulse with Low | Pins: CL1, CL2 | 800 | - | - | ns |
| $T_{CST}$ | Clock Setup Time | Pins: CL1, CL2 | 500 | - | - | ns |
| $T_{SU}$ | Data Setup Time | Pin: D | 300 | - | - | ns |
| $T_{DH}$ | Data Hold Time | Pin: D | 300 | - | - | ns |
| $T_{DM}$ | M Delay Time | Pin: M | -1000 | - | 1000 | ns |

## AC Characteristics ($T_A$ = 25°C, $V_{DD}$ = 2.7V) Parallel Mode Interface

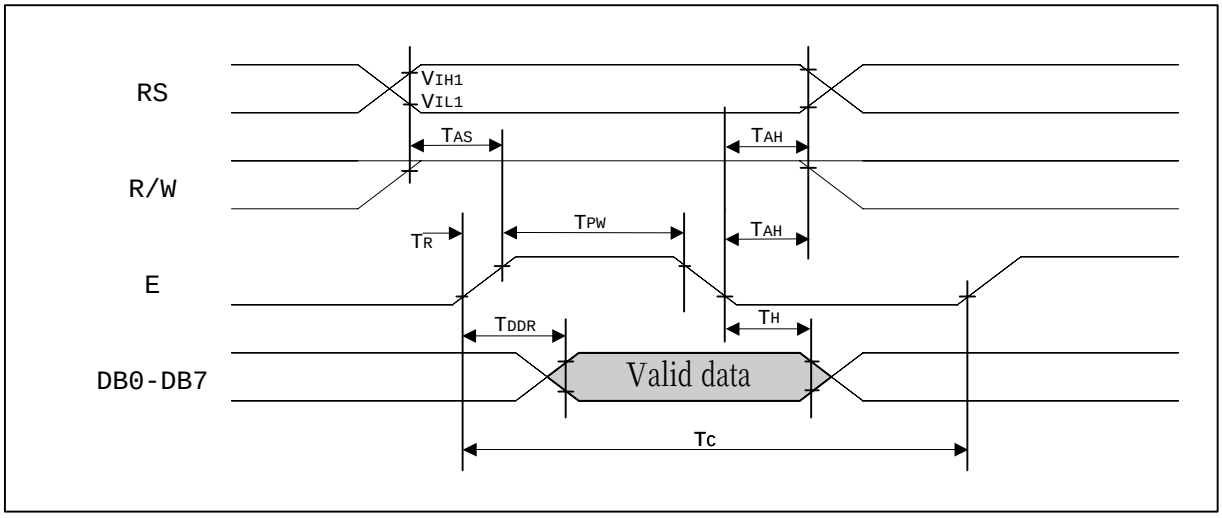
| Symbol | Characteristics | Test Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| *Internal Clock Operation* | | | | | | |
| $f_{OSC}$ | OSC Frequency | R = 18KΩ | 470 | 530 | 590 | KHz |
| *External Clock Operation* | | | | | | |
| $f_{EX}$ | External Frequency | - | 470 | 530 | 590 | KHz |
| | Duty Cycle | - | 45 | 50 | 55 | % |
| $T_R$,$T_F$ | Rise/Fall Time | - | - | - | 0.2 | μs |
| *Write Mode (Writing data from MPU to ST7920)* | | | | | | |
| $T_C$ | Enable Cycle Time | Pin E | 1800 | - | - | ns |
| $T_{PW}$ | Enable Pulse Width | Pin E | 160 | - | - | ns |
| $T_R$,$T_F$ | Enable Rise/Fall Time | Pin E | - | - | 25 | ns |
| $T_{AS}$ | Address Setup Time | Pins: RS,RW,E | 10 | - | - | ns |
| $T_{AH}$ | Address Hold Time | Pins: RS,RW,E | 20 | - | - | ns |
| $T_{DSW}$ | Data Setup Time | Pins: DB0 - DB7 | 40 | - | - | ns |
| $T_H$ | Data Hold Time | Pins: DB0 - DB7 | 20 | - | - | ns |
| *Read Mode (Reading Data from ST7920 to MPU)* | | | | | | |
| $T_C$ | Enable Cycle Time | Pin E | 1800 | - | - | ns |
| $T_{PW}$ | Enable Pulse Width | Pin E | 320 | - | - | ns |
| $T_R$,$T_F$ | Enable Rise/Fall Time | Pin E | - | - | 25 | ns |
| $T_{AS}$ | Address Setup Time | Pins: RS,RW,E | 10 | - | - | ns |
| $T_{AH}$ | Address Hold Time | Pins: RS,RW,E | 20 | - | - | ns |
| $T_{DDR}$ | Data Delay Time | Pins: DB0 - DB7 | - | - | 260 | ns |
| $T_H$ | Data Hold Time | Pins: DB0 - DB7 | 20 | - | - | ns |
| *Interface Mode with LCD Driver(ST7921)* | | | | | | |
| $T_{CWH}$ | Clock Pulse with High | Pins: CL1, CL2 | 800 | - | - | ns |
| $T_{CWL}$ | Clock Pulse with Low | Pins: CL1, CL2 | 800 | - | - | ns |
| $T_{CST}$ | Clock Setup Time | Pins: CL1, CL2 | 500 | - | - | ns |
| $T_{SU}$ | Data Setup Time | Pin: D | 300 | - | - | ns |
| $T_{DH}$ | Data Hold Time | Pin: D | 300 | - | - | ns |
| $T_{DM}$ | M Delay Time | Pin: M | -1000 | - | 1000 | ns |

## 8 bit interface timing diagram

- **MPU write data to ST7920**




- **MPU read data from ST7920**



## AC Characteristics ($T_A$ = 25°C, $V_{DD}$ = 4.5V) Serial Mode Interface

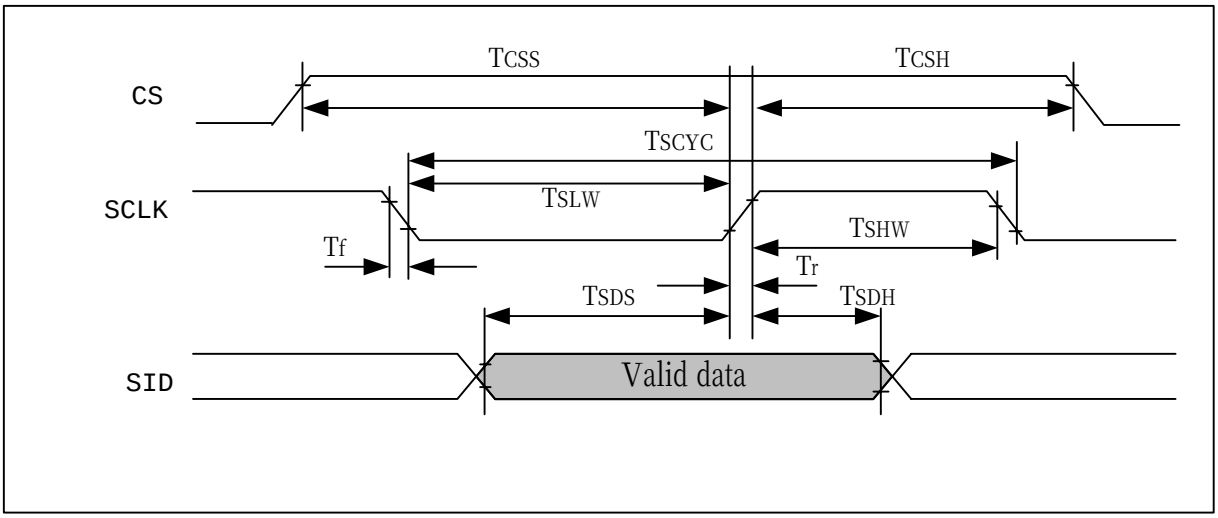
| Symbol | Characteristics | Test Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| | | *Internal Clock Operation* | | | | |
| $f_{OSC}$ | OSC Frequency | R = 33KΩ | 470 | 530 | 590 | KHz |
| | | *External Clock Operation* | | | | |
| $f_{EX}$ | External Frequency | - | 470 | 530 | 590 | KHz |
| | Duty Cycle | - | 45 | 50 | 55 | % |
| $T_R$,$T_F$ | Rise/Fall Time | - | - | - | 0.2 | μs |
| TSCYC | Serial clock cycle | Pin E | 400 | - | - | ns |
| TSHW | SCLK high pulse width | Pin E | 200 | - | - | ns |
| TSLW | SCLK low pulse width | Pin E | 200 | - | - | ns |
| TSDS | SID data setup time | Pins RW | 40 | - | - | ns |
| TSDH | SID data hold time | Pins RW | 40 | - | - | ns |
| TCSS | CS setup time | Pins RS | 60 | - | - | ns |
| TCSH | CS hold time | Pins RS | 60 | - | - | ns |

## AC Characteristics ($T_A$ = 25°C, $V_{DD}$ = 2.7V) Serial Mode Interface

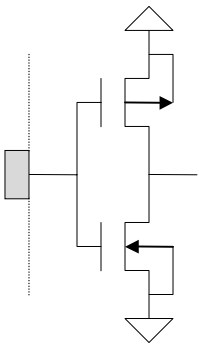
| Symbol | Characteristics | Test Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| | | *Internal Clock Operation* | | | | |
| $f_{OSC}$ | OSC Frequency | R = 18KΩ | 470 | 530 | 590 | KHz |
| | | *External Clock Operation* | | | | |
| $f_{EX}$ | External Frequency | - | 470 | 530 | 590 | KHz |
| | Duty Cycle | - | 45 | 50 | 55 | % |
| $T_R$,$T_F$ | Rise/Fall Time | - | - | - | 0.2 | μs |
| TSCYC | Serial clock cycle | Pin E | 600 | - | - | ns |
| TSHW | SCLK high pulse width | Pin E | 300 | - | - | ns |
| TSLW | SCLK low pulse width | Pin E | 300 | - | - | ns |
| TSDS | SID data setup time | Pins RW | 40 | - | - | ns |
| TSDH | SID data hold time | Pins RW | 40 | - | - | ns |
| TCSS | CS setup time | Pins RS | 60 | - | - | ns |
| TCSH | CS hold time | Pins RS | 60 | - | - | ns |

## Serial interface timing diagram

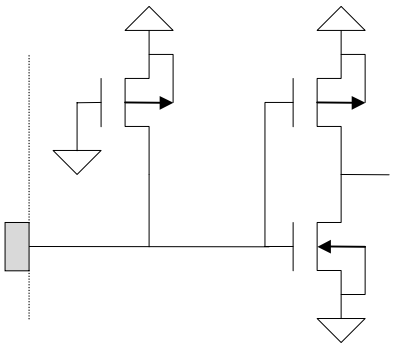
- **MPU write data to ST7920**



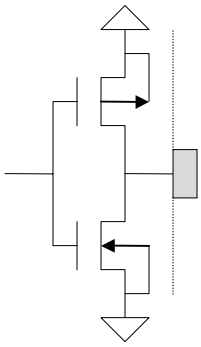
# I/O pin diagram



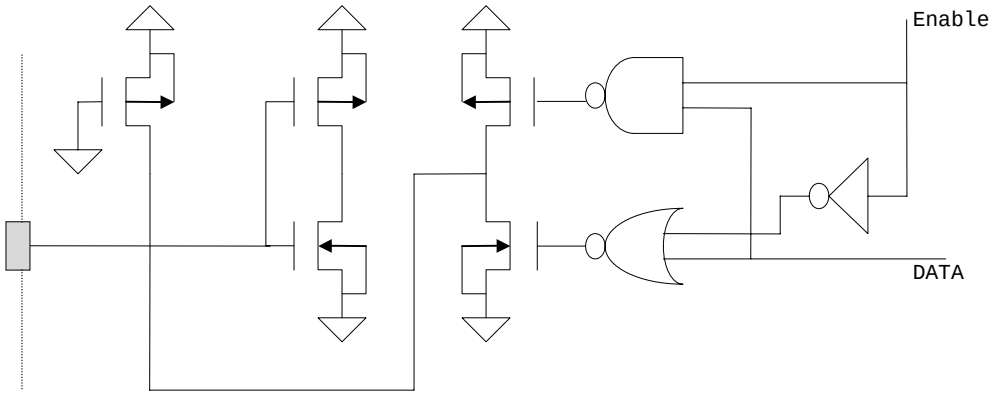
Input PAD: E (No Pull-up)



Input PAD: RS, RW(with Pull-up)



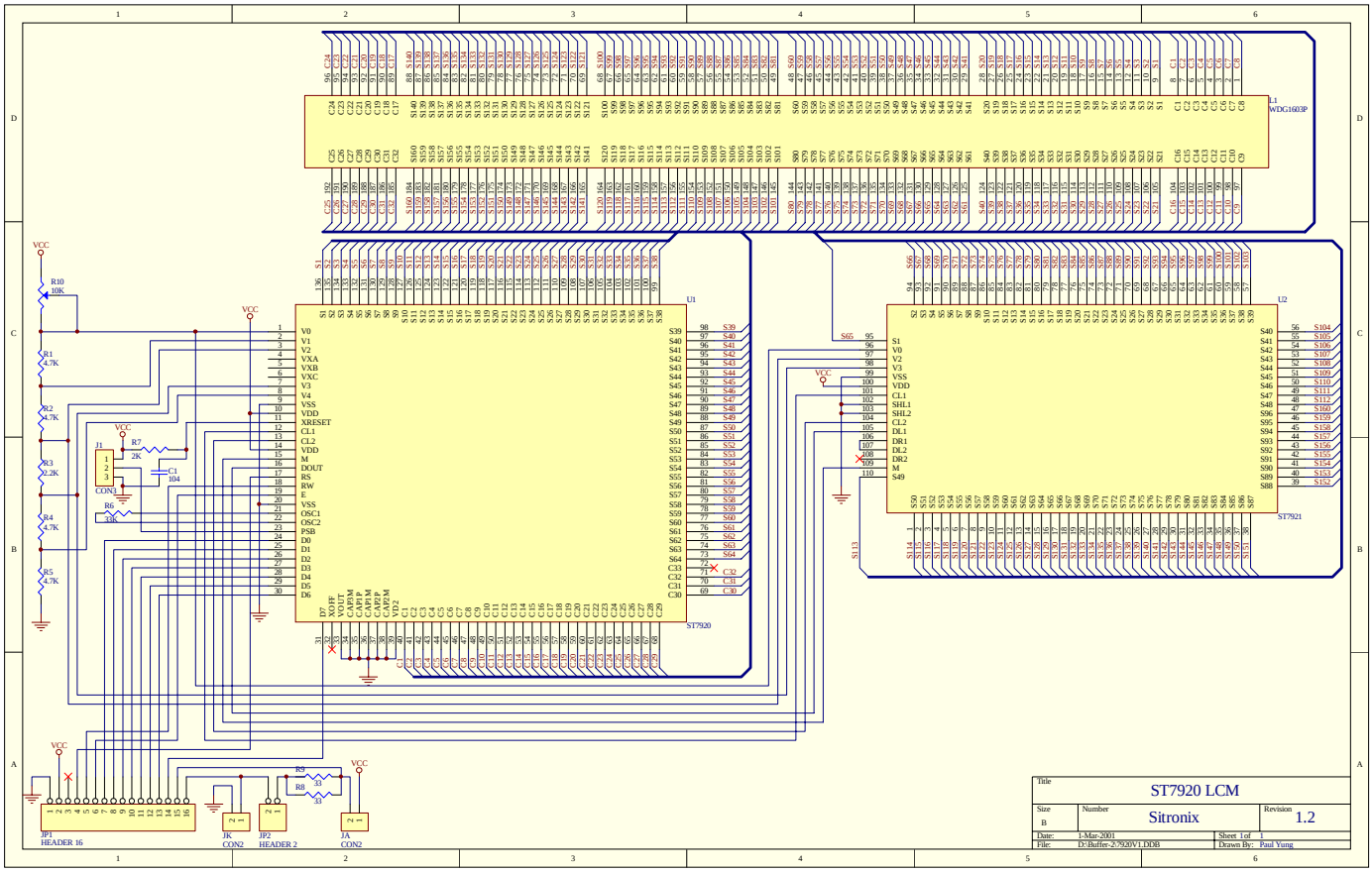
Output PAD: CL1, CL2, M, D




I/O PAD: DB0 – DB7

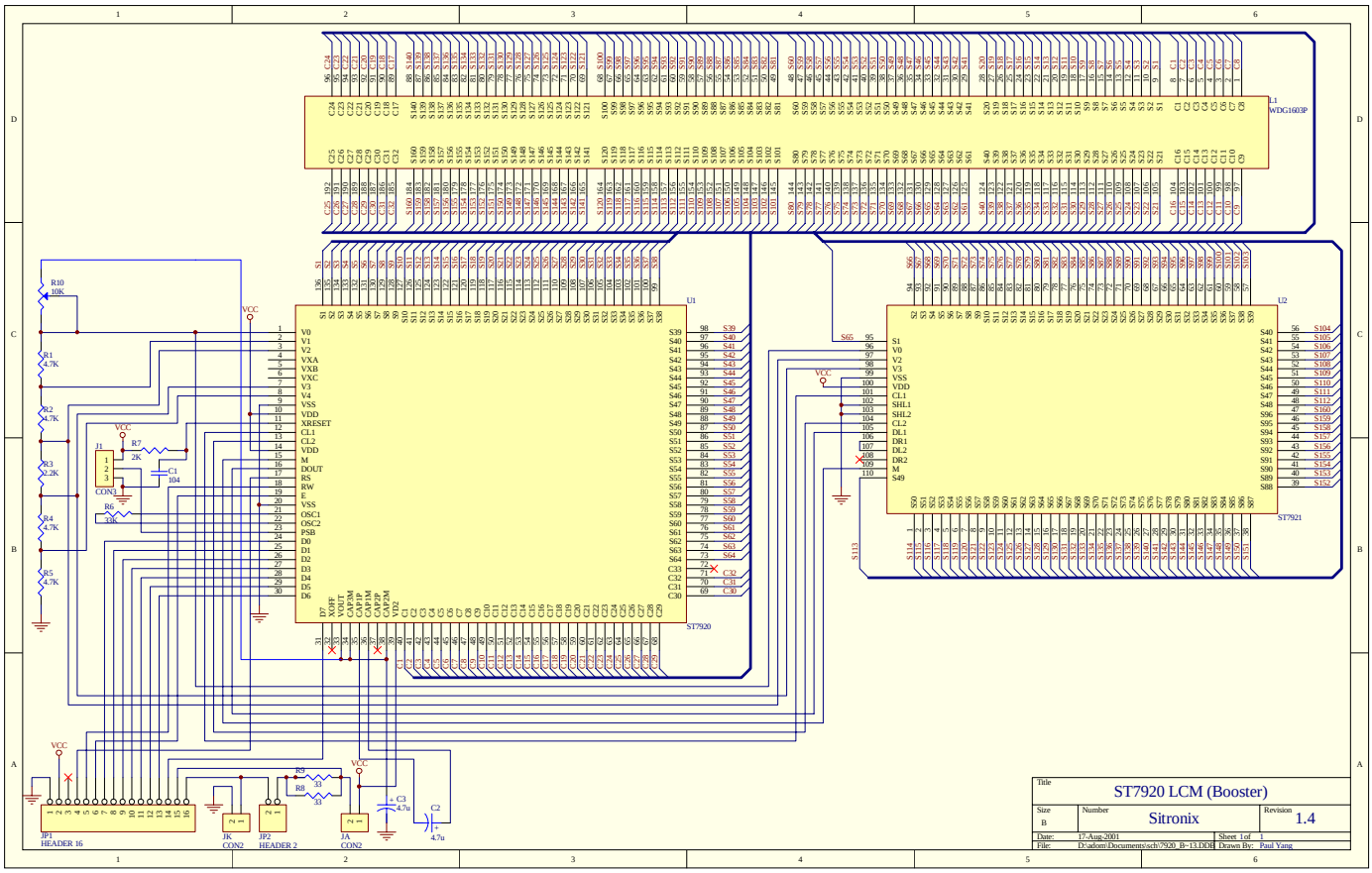
## Application circuit 1:

LCD : 32 COM x 160 SEG
LCD Voltage : VCC



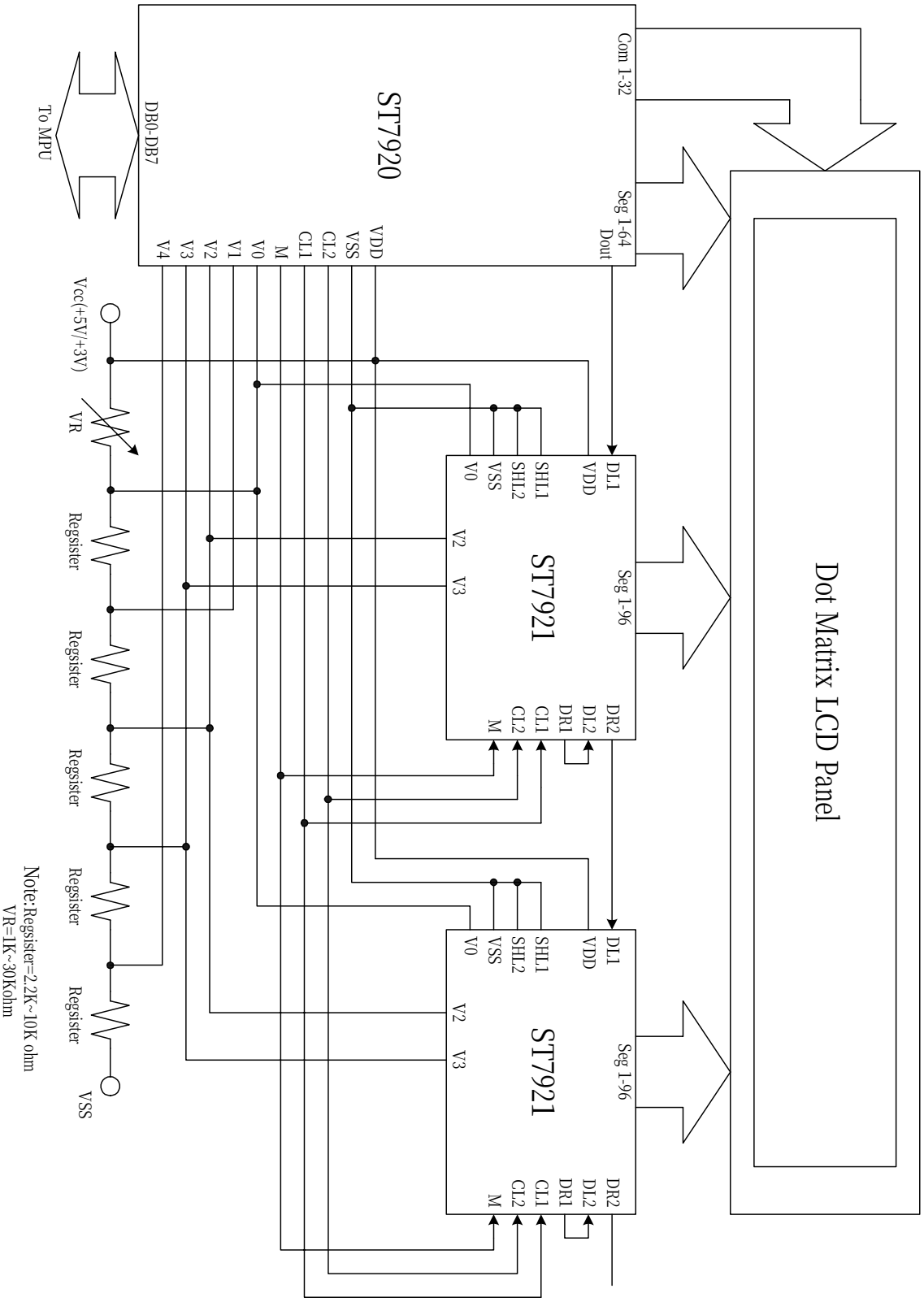
## Application circuit 2：

LCD ： 32 COM x 160 SEG
LCD Voltage ： VCC x 2（Voltage doubler is used）*Vlcd should not over 7v.



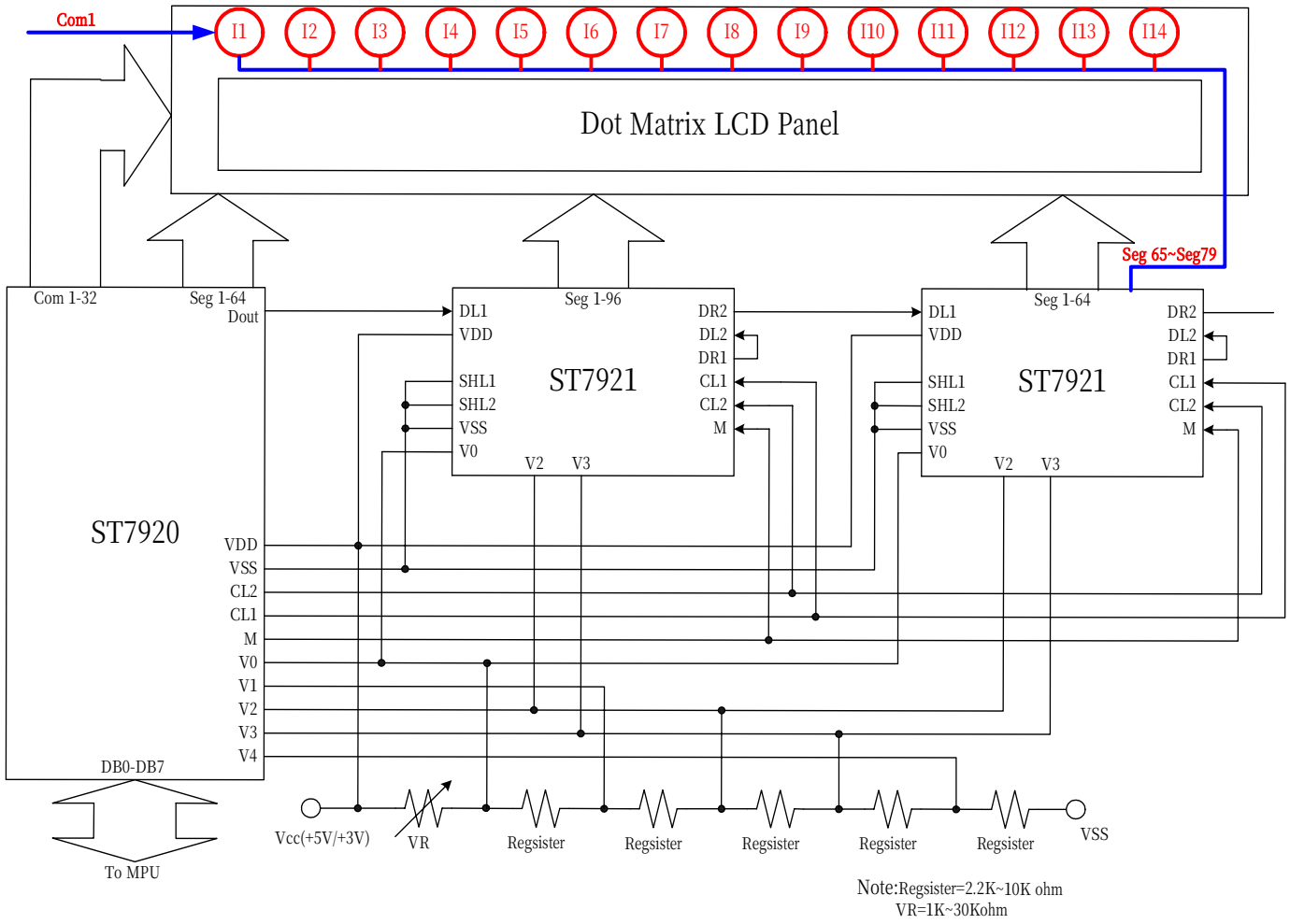
Application circuit 3：

LCD ： 2Line 16Chinese Word (32 COM x 256 SEG)

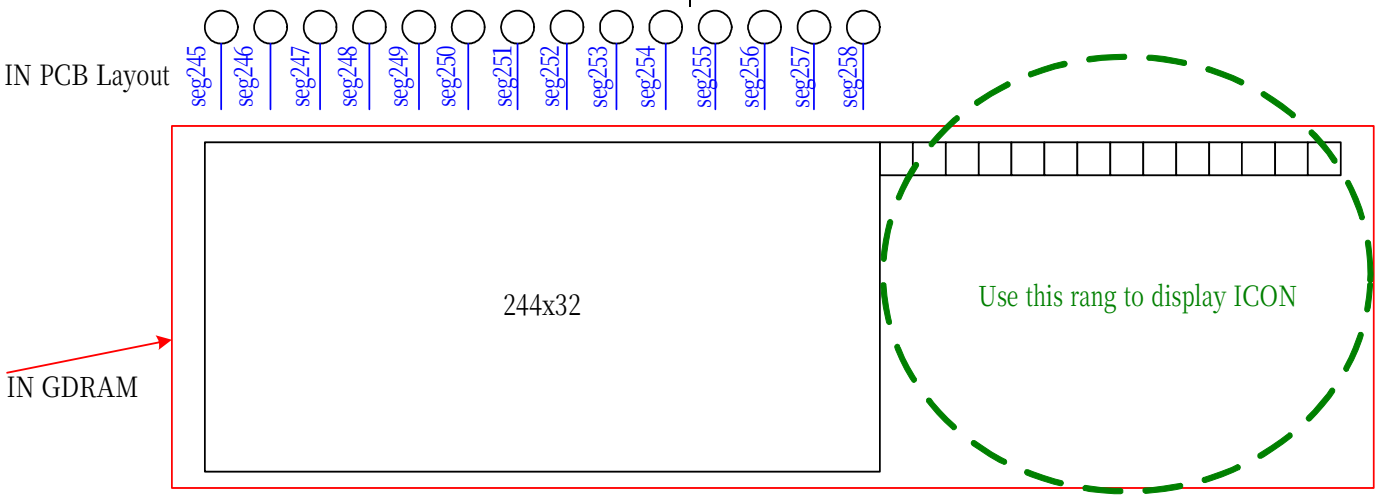


## Application circuit 4：

LCD ： 2Line 12 Chinese Word (32 COM x 244 SEG) +14 ICON (USE14 SEG and 1 COM)




EX：



Application circuit for testing CGROM and HCGROM: