GT23L24M1Z0-S

标准汉字字库芯片

24X24 点阵

竖置竖排

（SPI 接口/PLII 接口）

用户手册

VER 2.6

2007-Q2

版本修订记录

版本号	修改内容	日期	备注
V2.5	生成数据手册	2007-04	

目　　录

1 概述 4

2 特点 5

3 引脚配置 6

4 引脚描述 6

4.1 SPI 接口引脚描述 7
4.2 PLII 接口引脚描述 7

5 SPI 接口/PLII 接口与主机接口电路示意图 8

5.1 SPI 接口与主机接口电路示意图 8
5.2 PLII 接口与主机接口电路示意图 9

6 操作指令 10

6.1 SPI 接口模式下操作 10
6.2 PLII 接口模式下操作 12

7 电气特性 14

7.1 绝对最大额定值 14
7.2 DC 特性 14
7.3 AC 特性 14

8 封装尺寸：SOP20 18

9 功能描述 19

9.1 汉字点阵排列格式 19
9.2 汉字点阵字库结构 24
9.3 汉字点阵在芯片中的地址计算方法 25

10 附录 28

10.1 GB18030　1 区和 5 区 (字符区) 28
10.2 扩展字符区字符 32

1 概述

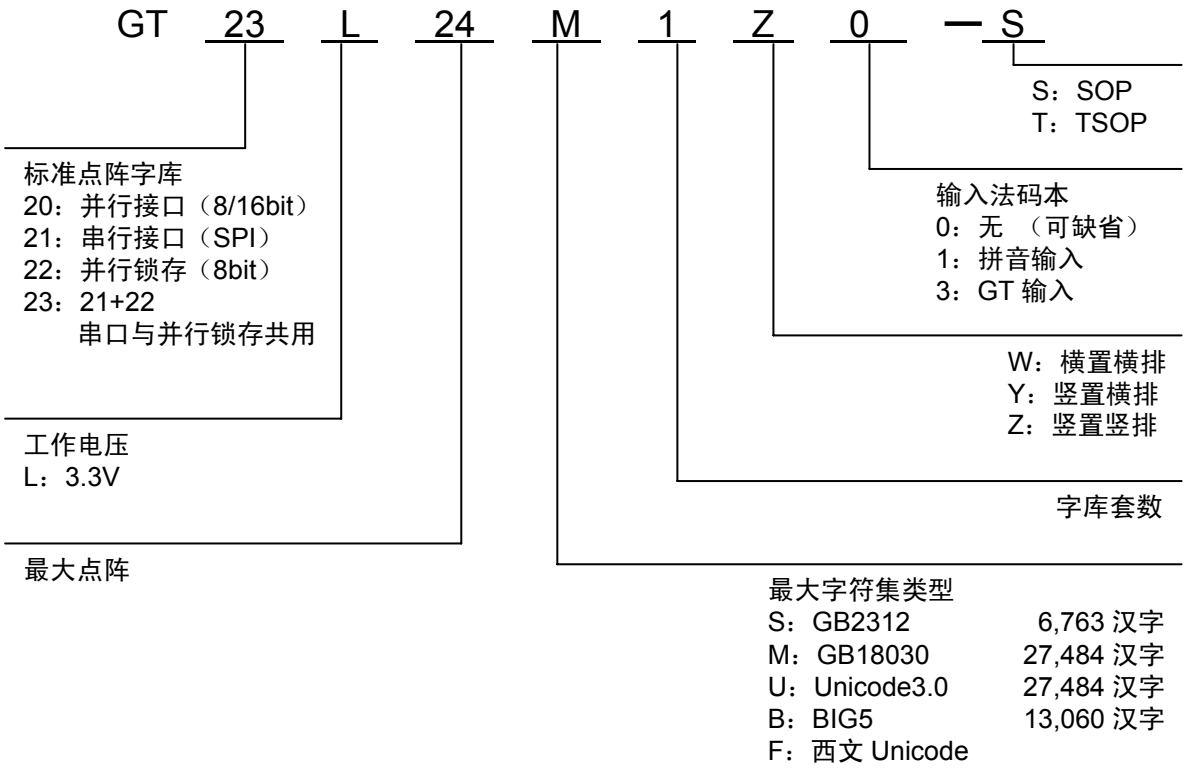
GT23L24M1Z0是一款支持GB18030字符集（27484汉字）的24X24点阵字库芯片，另外还含有2套24点高的半角不等宽ASCII字符、1套12X24点扩展字符。

每个汉字在字库中是以汉字字模的形式存储的，每个点用一个二进制位表示，存 1 的点，当显示时可以在屏幕上显示亮点，存 0 的点，则在屏幕上不显示。点阵排列格式为竖置竖排：即一个字节的高位表示下面的点，低位表示上面的点，排满一列的点后再排下一列。这样把点阵信息用来直接在显示器上按上述规则显示，则将出现对应的汉字。

用户只要知道汉字的内码，就可以通过本手册提供的方法计算出该汉字点阵在芯片中的地址，然后就可从该地址连续读出点阵信息。

GT23L24M1Z0支持2种不同的总线接口：即SPI接口、PLII接口。

命名规则（标准点阵字库芯片）



2 特点

- 数据总线：SPI 总线兼容的串行接口/PLII 接口
- 点阵排列方式：字节竖置横排
- 访问速度：SPI 时钟频率：40MHz(max.)
  PLII 访问速度：70ns(max.) @3.3V
- 工作电压：2.7V – 3.6V
- 电流：
  工作电流：30mA(max.) @3.3V
  待机电流：15uA(max.) @3.3V
- 封装： SOP20
- 字库内容：

汉 字 字 库	
24×24 点	GB18030 （27,484 汉字） （1,038 字符）

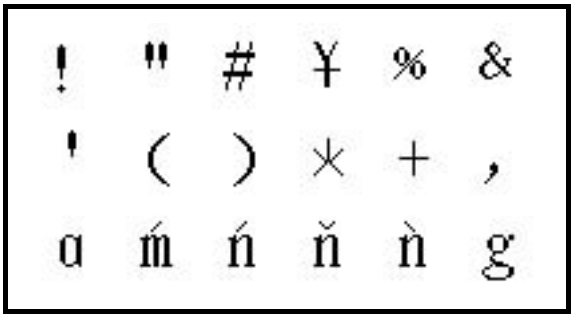
ASCII 字符	
12×24 点	扩展字符（126 字符）
24 点阵	方头（96 字符）
24 点阵	白正（96 字符）

- 字型样张

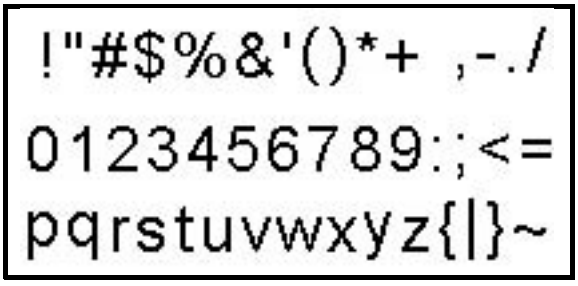
24×24 点 宋体 GB18030（27,484 汉字）



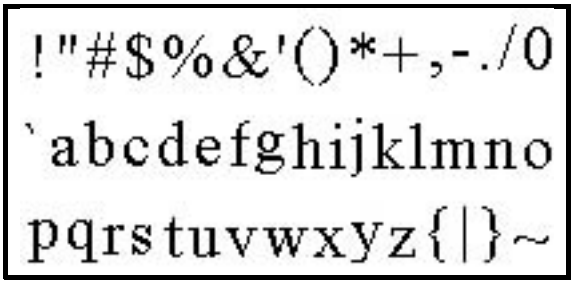
12×24 点 ASCII 扩展字符（126 字符）



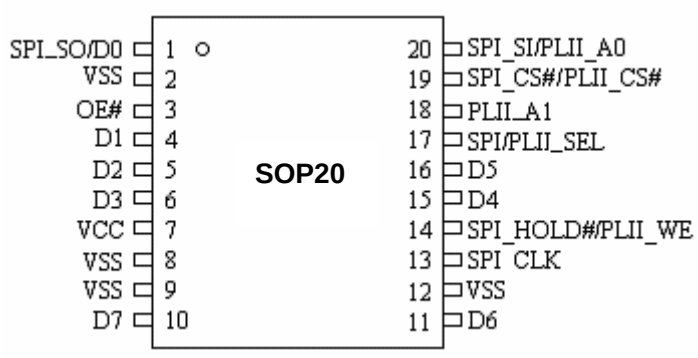
24 点阵 ASCII 方头（96 字符）



24 点阵 ASCII 白正（96 字符）



3 引脚配置



SOP20

4 引脚描述

SOP20	名称	描述	
		GT21L(SPI 接口)	GT22L(PLII 接口)
1	SPI_SO/D0	Serial data output	Data Outputs
2	VSS	Ground	
3	OE#	No Connection	Output Enable Input
4	D1	No Connection	Data Outputs
5	D2	No Connection	Data Outputs
6	D3	No Connection	Data Outputs
7	VCC	Power Supply(3.3V)	
8	VSS	Ground	
9	VSS	Ground	
10	D7	No Connection	Data Outputs
11	D6	No Connection	Data Outputs
12	VSS	Ground	
13	SPI_CLK	Serial clock input	No Connection
14	SPI_HOLD#/PLII_WE	Hold(to pause the device)	Write Enable Input
15	D4	No Connection	Data Outputs
16	D5	No Connection	Data Outputs
17	SPI/PLII_SEL	SPI/PLII SELECT	
		NC: SPI	GND: PLII
18	PLII_A1	No Connection	Address Inputs
19	SPI_CS#/PLII_CS#	Chip enable input	Chip Enable Input
20	SPI_SI/PLII_A0	Serial data input	Address Inputs

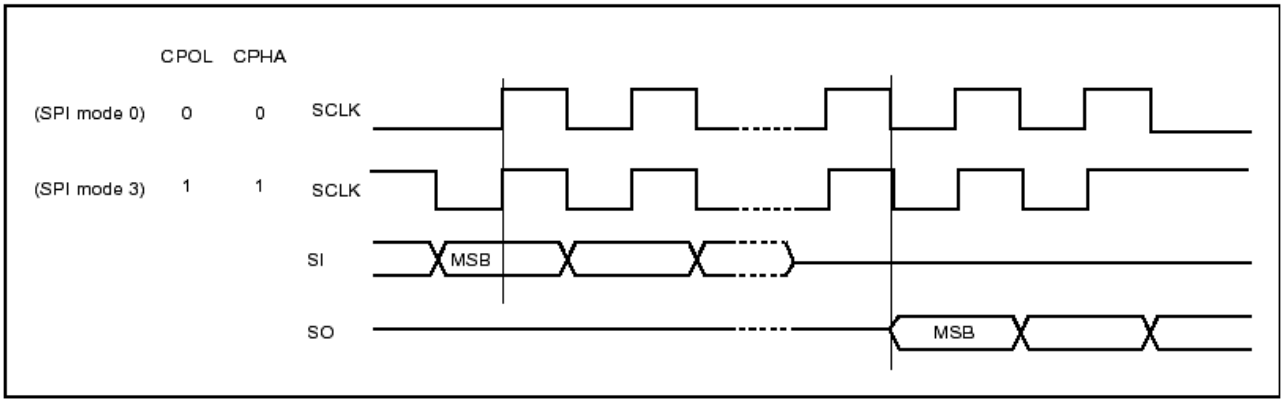
4.1 SPI 接口引脚描述

串行数据输出（SO）：该信号用来把数据从芯片串行输出，数据在时钟的下降沿移出。

串行数据输入（SI）：该信号用来把数据从串行输入芯片，数据在时钟的上升沿移入。

串行时钟输入（SCLK）：数据在时钟上升沿移入，在下降沿移出。

片选输入（CS#）：所有串行数据传输开始于CE#下降沿，CE#在传输期间必须保持为低电平，在两条指令之间保持为高电平。

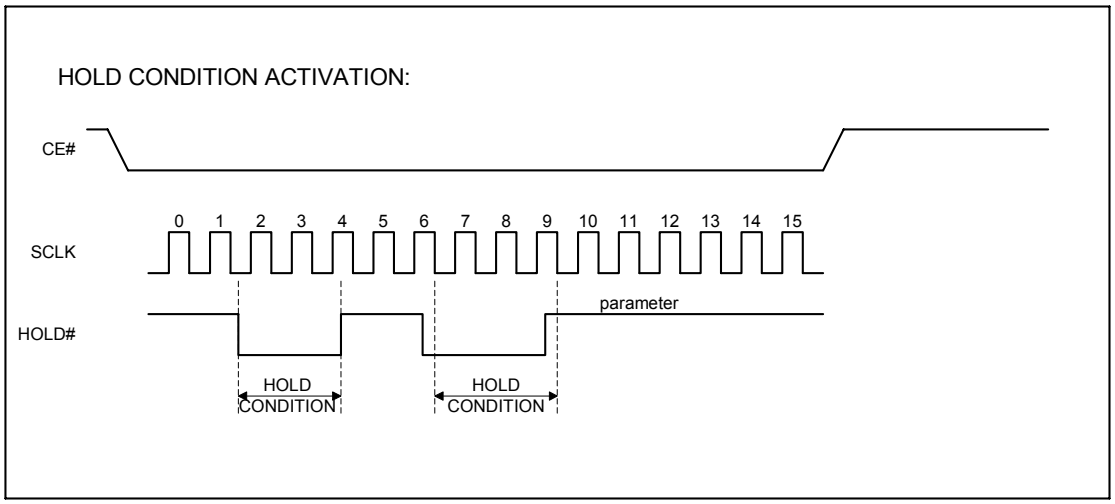


总线挂起输入（HOLD#）：

该信号用于片选信号有效期间暂停数据传输，在总线挂起期间，串行数据输出信号处于高阻态，芯片不对串行数据输入信号和串行时钟信号进行响应。

当HOLD#信号变为低并且串行时钟信号（SCLK）处于低电平时，进入总线挂起状态。

当HOLD#信号变为高并时串行时钟信号（SCLK）处于低电平时，结束总线挂起状态。



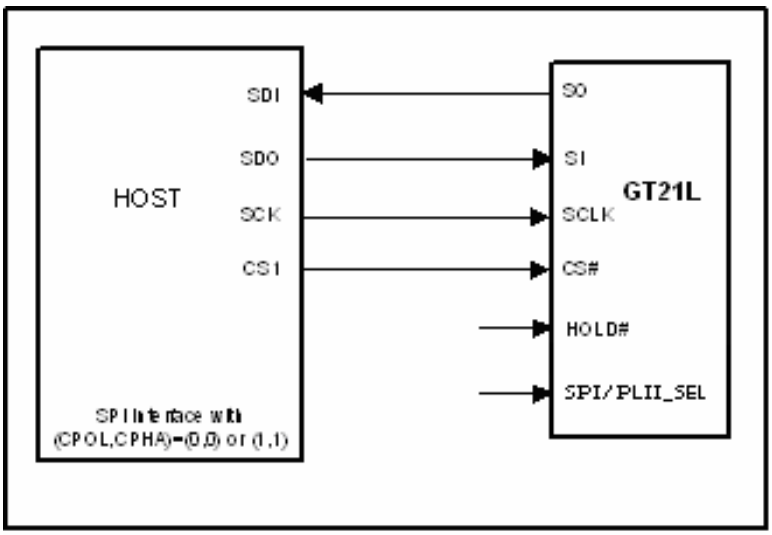
4.2 PLII 接口引脚描述

Pin name	I/O	描述
A[1..0]	I	地址寄存器寻址
D[7..0]	I/O	地址输入/数据输出
CE#	I	片选信号输入，低有效
OE#	I	“输出使能”信号输入，OE# 为低时输出使能
WE#	I	“写使能”信号输入，WE# 为低时写使能

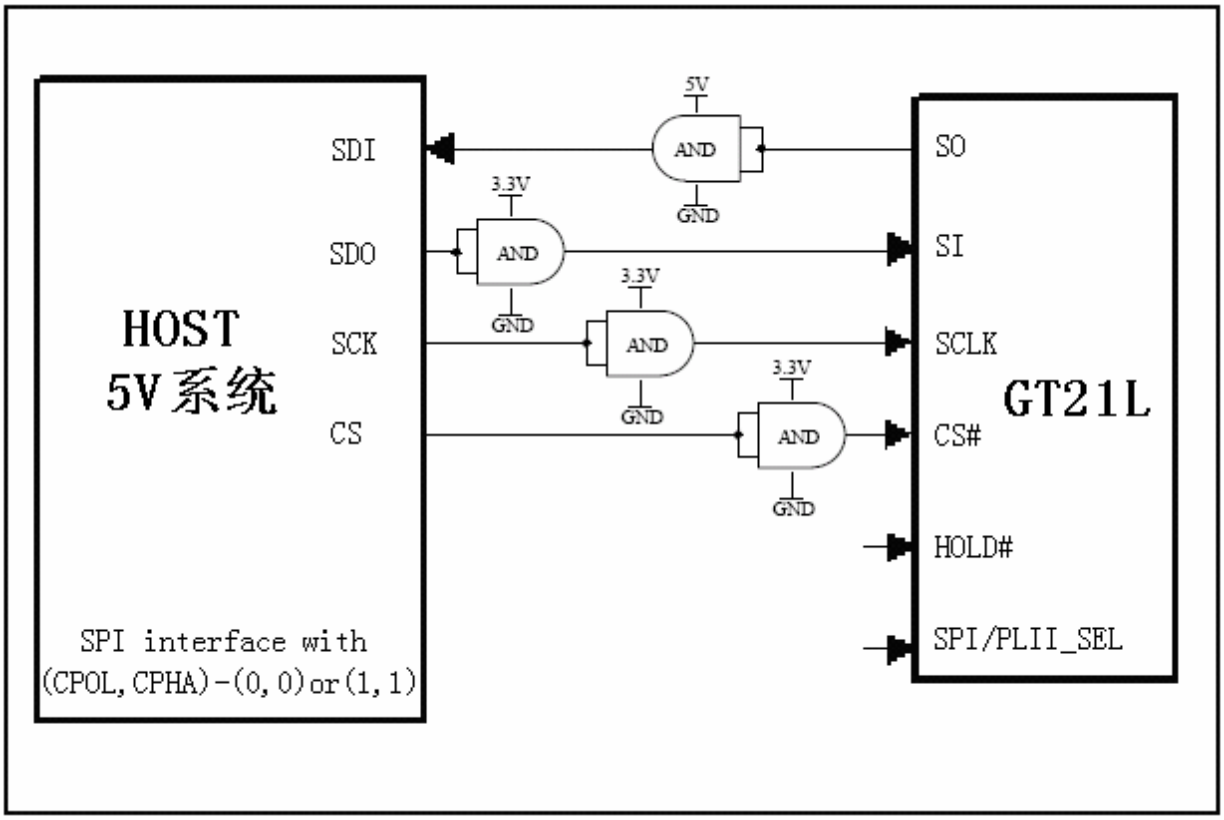
5 SPI 接口/PLII 接口与主机接口电路示意图

5.1 SPI 接口与主机接口电路示意图

SPI/PLII_SEL（管脚内部有 100K 上拉电阻）悬空，字库芯片选择 SPI 接口模式，与主机接口电路连接可以参考下图。

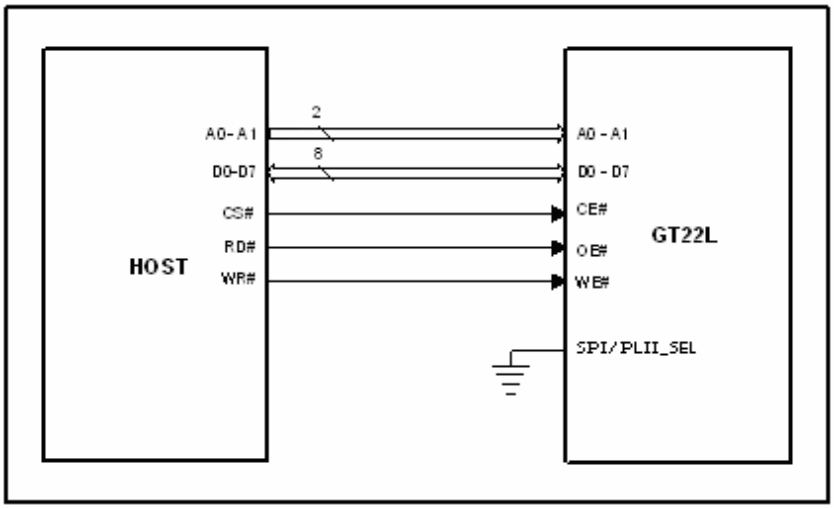


注意：若是采用系统电压为 5V 的，则需要进行电平转换匹配连接 GT21L 库芯片，可以参考下图。

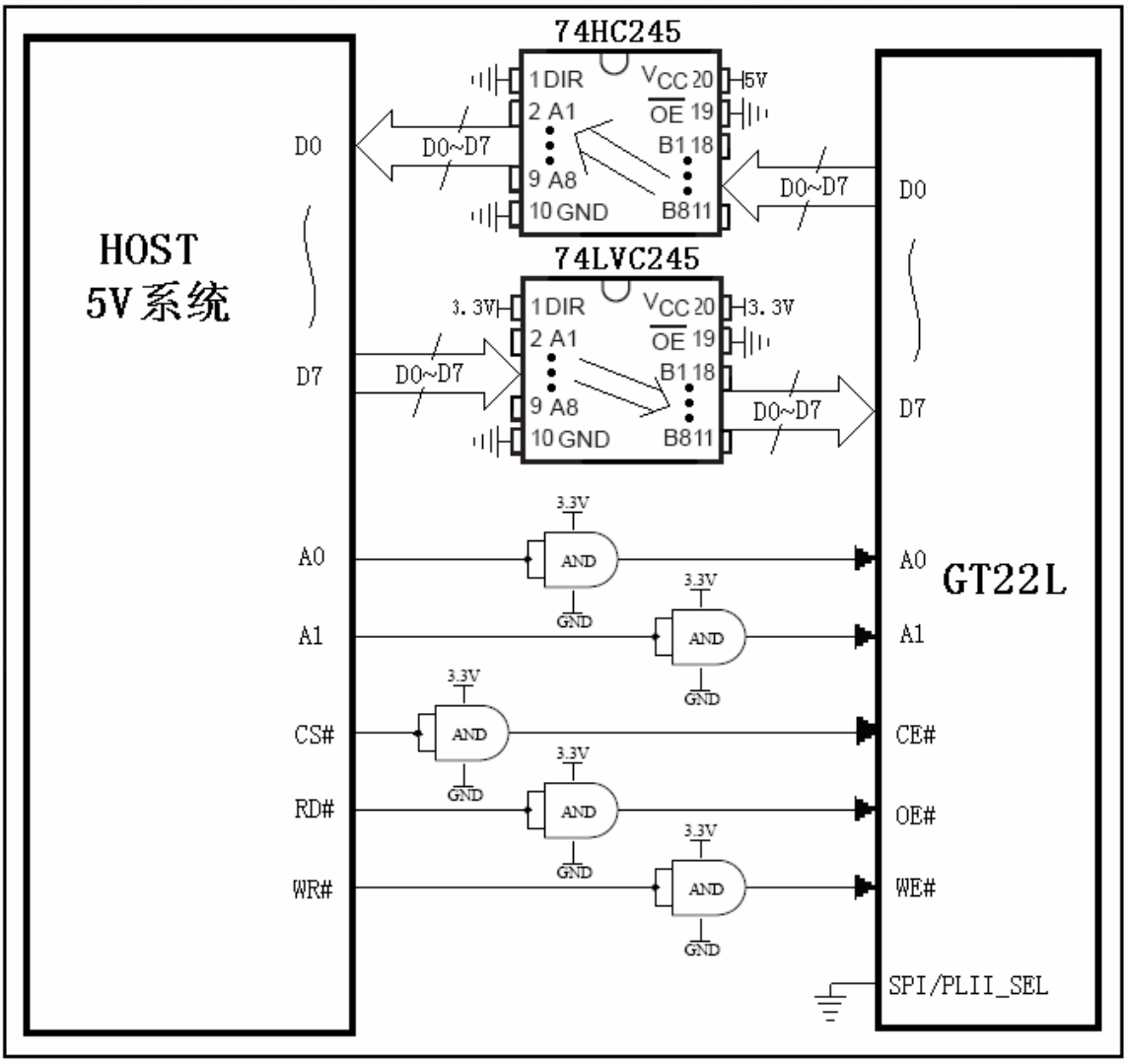


5.2 PLII 接口与主机接口电路示意图

SPI/PLII_SEL（管脚内部有 100K 上拉电阻）接地，字库芯片选择 SPI 接口模式，与主机接口电路连接可以参考下图。



注意：若是采用系统电压为 5V 的，则需要进行电平转换匹配(如:通过 74LVC245 进行电平转换)，连接 GT20L 库芯片，可以参考下图。



6 操作指令

6.1 SPI 接口模式下操作

6.1.1 指令参数

Instruction Set

Instruction	Description	Instruction Code(One-Byte)		Address Bytes	Dummy Bytes	Data Bytes
READ	Read Data Bytes	0000 0011	03 h	3	—	1 to ∞
FAST_READ	Read Data Bytes at Higher Speed	0000 1011	0B h	3	1	1 to ∞

所有对本芯片 SPI 接口的操作只有 2 个，那就是 Read Data Bytes (READ “一般读取”)和 Read Data Bytes at Higher Speed (FAST_READ “快速读取点阵数据”)。.

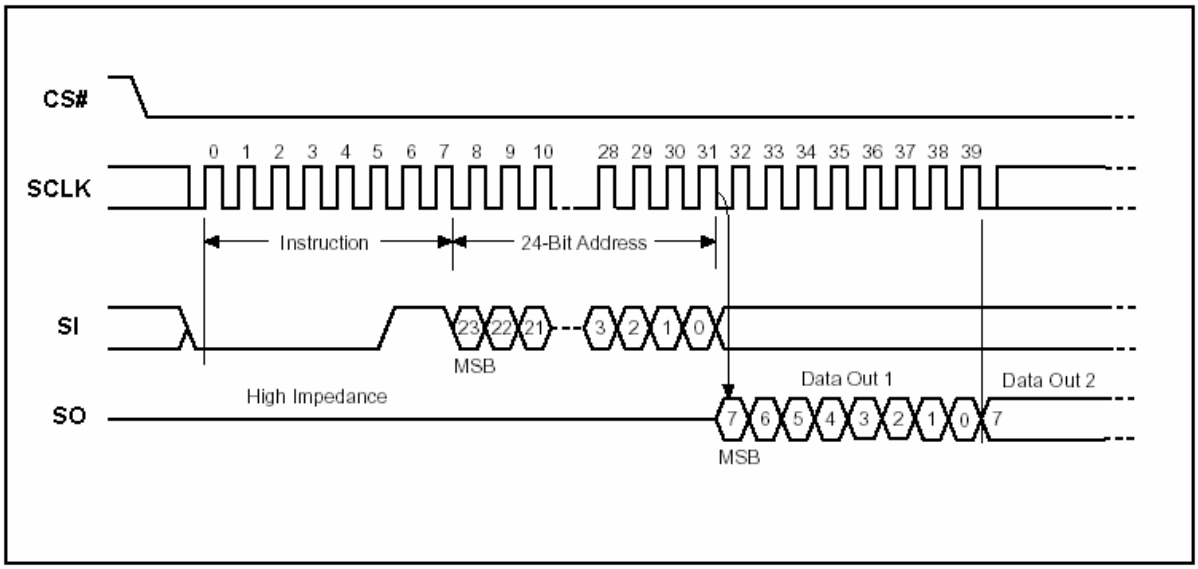
6.1.2 Read Data Bytes（一般读取）

Read Data Bytes 需要用指令码来执行每一次操作。READ 指令的时序如下(图)：

- 首先把片选信号（CS#）变为低，紧跟着的是 1 个字节的命令字（03 h）和 3 个字节的地址和通过串行数据输入引脚（SI）移位输入，每一位在串行时钟（SCLK）上升沿被锁存。
- 然后该地址的字节数据通过串行数据输出引脚（SO）移位输出，每一位在串行时钟（SCLK）下降沿被移出。
- 读取字节数据后，则把片选信号（CS#）变为高，结束本次操作。

如果片选信号（CS#）继续保持为底，则下一个地址的字节数据继续通过串行数据输出引脚（SO）移位输出。

图：Read Data Bytes (READ) Instruction Sequence and Data-out sequence



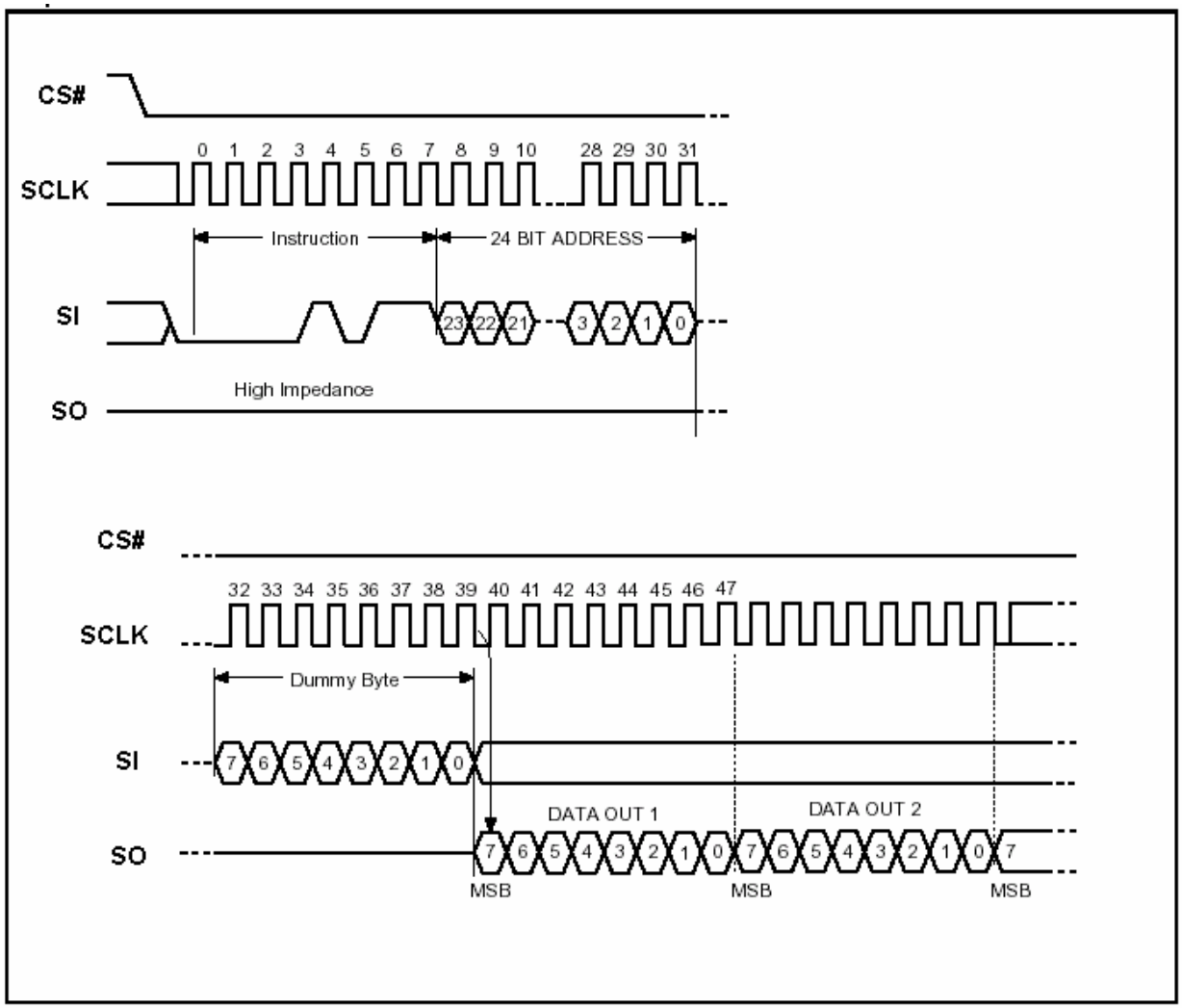
6.1.3 Read Data Bytes at Higher Speed（快速读取点阵数据）

Read Data Bytes at Higher Speed 需要用指令码来执行操作。READ_FAST 指令的时序如下(图)：

- 首先把片选信号（CS#）变为低，紧跟着的是 1 个字节的命令字（0B h）和 3 个字节的地址以及一个字节 Dummy Byte 通过串行数据输入引脚（SI）移位输入，每一位在串行时钟（SCLK）上升沿被锁存。
- 然后该地址的字节数据通过串行数据输出引脚（SO）移位输出，每一位在串行时钟（SCLK）下降沿被移出。
- 如果片选信号（CS#）继续保持为底，则下一个地址的字节数据继续通过串行数据输出引脚（SO）移位输出。例：读取一个 15x16 点阵汉字需要 32Byte，则连续 32 个字节读取后结束一个汉字的点阵数据读取操作。

  如果不需要继续读取数据，则把片选信号（CS#）变为高，结束本次操作。

图：Read Data Bytes at Higher Speed (READ_FAST) Instruction Sequence and Data-out sequence



6.2 PLII 接口模式下操作

在PLII模式下，字库芯片内部有3个地址保持寄存器，HOST 读字库芯片时，字库芯片把地址寄存器对应字库芯片地址内容送给HOST，并且HOST每读取一个字节后，字库芯片内部会把地址寄存器的值增1。当地址寄存器越过最大地址时自动归零。字库芯片可以对地址寄存器进行写操作。字库芯片上电时，内部硬件把地址寄存器清零。

6.2.1 信号描述

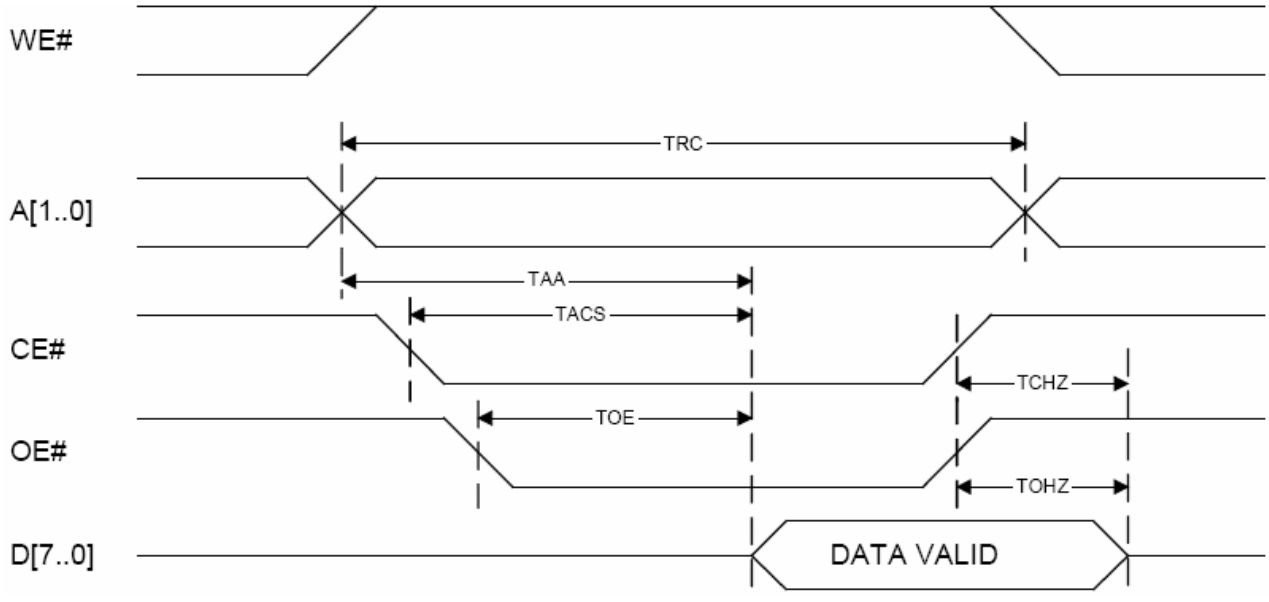
Pin name	I/O	描述
A[1..0]	I	地址寄存器寻址
D[7..0]	I/O	地址输入/数据输出
CE#	I	片选信号输入，低有效
OE#	I	“输出使能”信号输入，OE# 为低时输出使能
WE#	I	“写使能”信号输入，WE# 为低时写使能

真值表

Mode	CE#	OE#	WE#	D[7..0]
Other	H	X	X	High-Z
Read	L	L	H	Data Out
write	L	H	L	Addr In

6.2.2 读操作

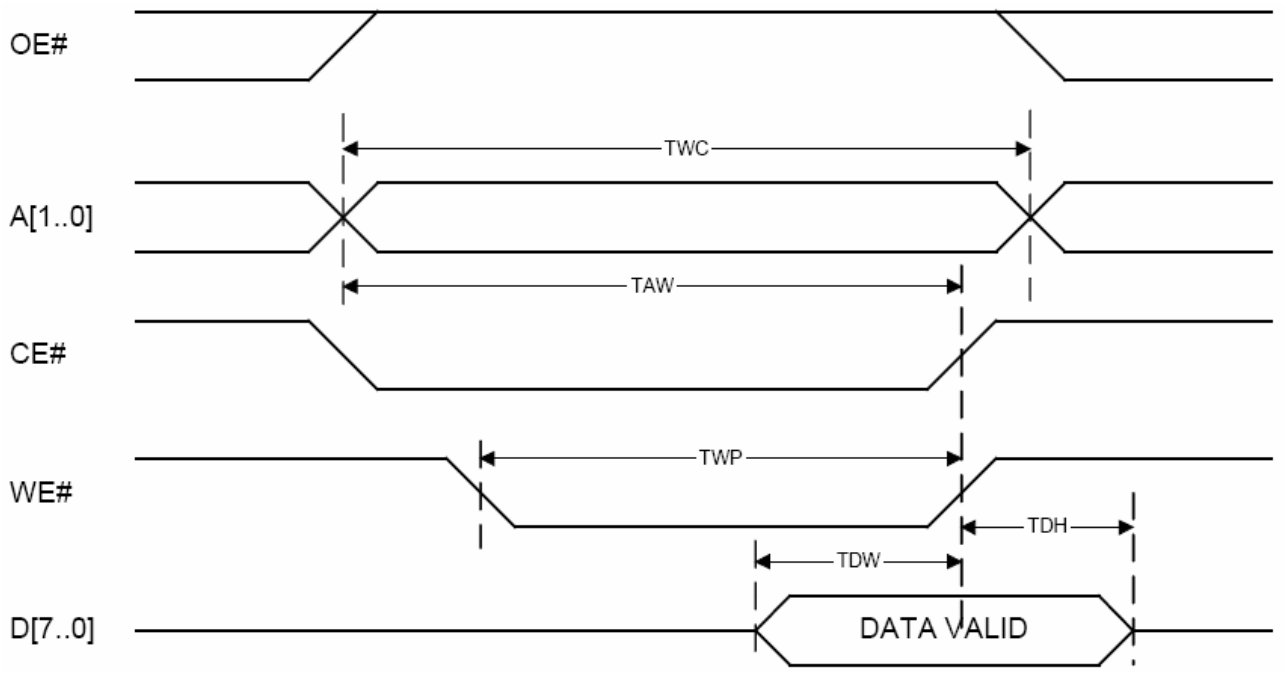
读字库芯片内部的点阵字库数据时，HOST 写字库芯片的 3 个地址寄存器，字库芯片把对应地址的数据送给 HOST。在 OE#和 CE#同时为低电平、WE#为高电平的情况下，HOST 可以从数据总线（D[7..0]）读出字库芯片的个字节的数据。当出现 OE# 或 CE# 中的一个信号变高，则字库芯片内部在此时刻把地址寄存器增 1，并保持地址寄存器的值。



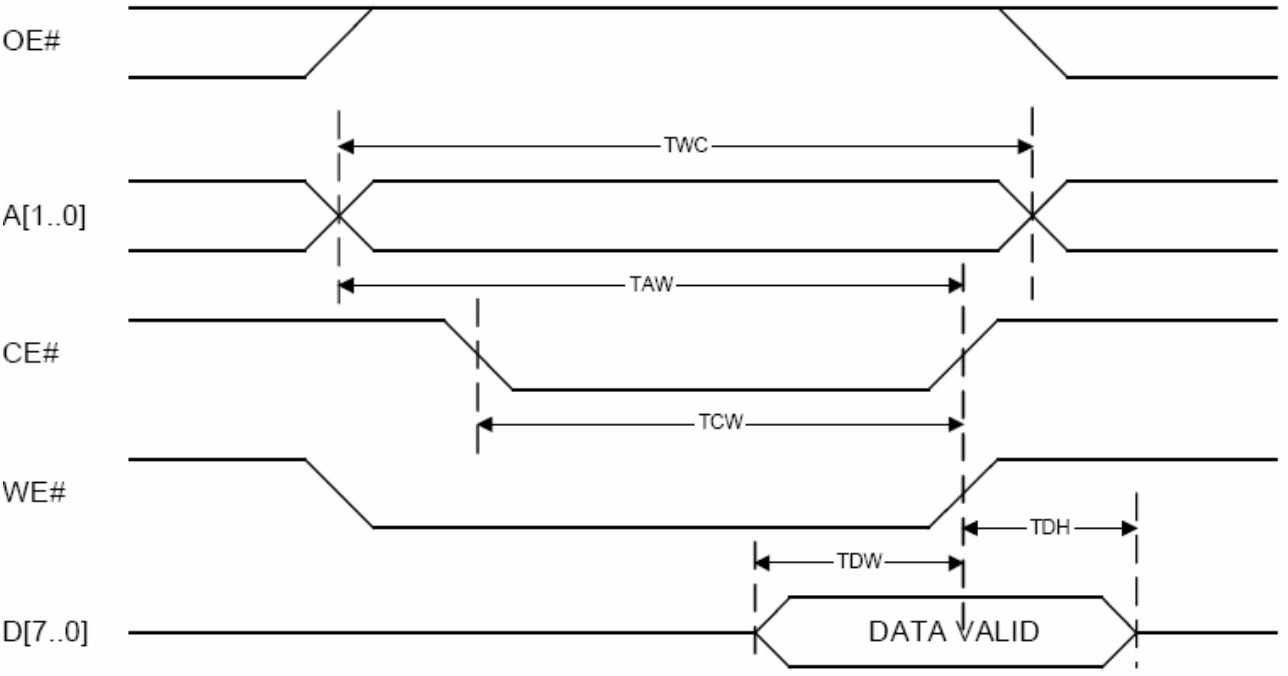
读周期时序波形图

6.2.3 写操作

字库芯片内部有 3 个地址保持寄存器，HOST 读字库芯片时需要把地址写到字库芯片中。在 WE#和 CE#同时为低电平、OE#为高电平的情况下，HOST 可以通过数据总线（D[7..0]）写 1 个字节数据到字库芯片。



写周期时序波形图（WE#控制的时序）



写周期时序波形图（CE#控制的时序）

7 电气特性

7.1 绝对最大额定值

Symbol	Parameter	Min.	Max.	Unit	Condition
T_{OP}	Operating Temperature	0	70	℃	
T_{STG}	Storage Temperature	-65	125	℃	
VCC	Supply Voltage	2.7	3.6	V	
V_{IN}	Input Voltage	-0.5	VCC+0.5	V	
GND	Power Ground	0	0	V	

7.2 DC 特性

Condition：T_{OP} = 0℃ to 70℃，GND=0V

Symbol	Parameter	Min.	Max.	Unit	Condition
I_{DD}	VCC Supply Current(active)		40	mA	
I_{SB}	VCC Standby Current		10	uA	
V_{IL}	Input LOW Voltage	-0.3	0.6	V	VCC=3V
V_{IH}	Input HIGH Voltage	2.2	VCC+0.3	V	
V_{OL}	Output LOW Voltage		0.4 (I_{OL}=1.6mA)	V	
V_{OH}	Output HIGH Voltage	0.8VCC (I_{OH}=-0.4mA)		V	
I_{LI}	Input Leakage Current	0	+10	uA	
I_{LO}	Output Leakage Current	0	+10	uA	

Note：I_{IL}：Input LOW Current，I_{IH}：Input HIGH Current，
I_{OL}：Output LOW Current，I_{OH}：Output HIGH Current，

7.3 AC 特性

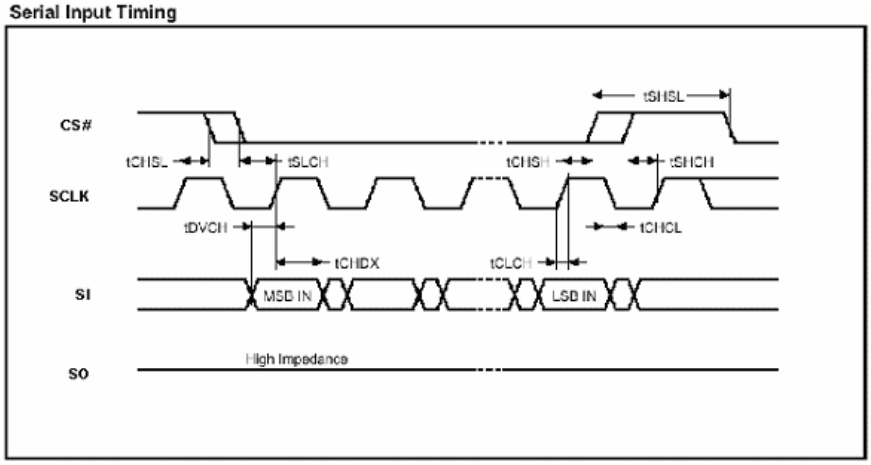
7.3.1 SPI 接口模式下 AC 特性

Condition：T_{OP} = 0℃ to 70℃，VCC= 2.7V to 3.6V

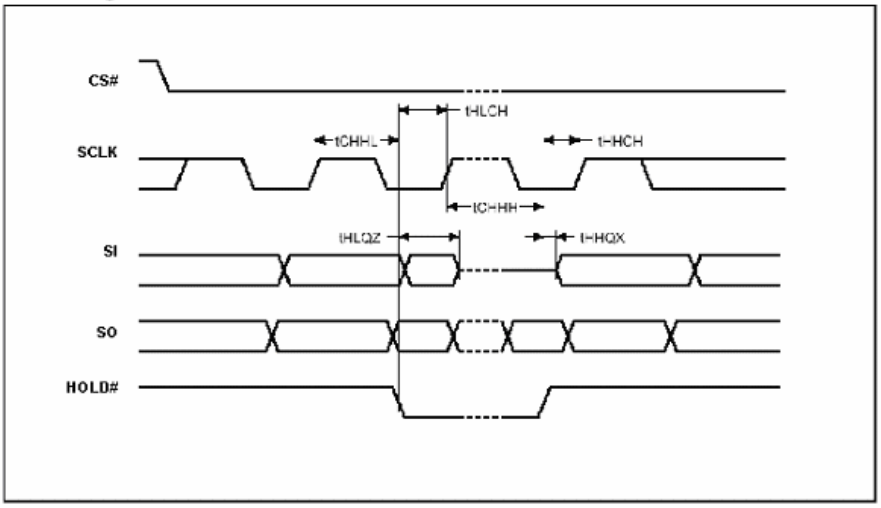
Symbol	Alt.	Parameter	Min.	Max.	Unit
Fc	Fc	Clock Frequency	D.C.	40	MHz
t_{CH}	t_{CLH}	Clock High Time	11		ns
t_{CL}	t_{CLL}	Clock Low Time	11		ns
t_{CLCH}		Clock Rise Time(peak to peak)	0.1		V/ns
t_{CHCL}		Clock Fall Time (peak to peak)	0.1		V/ns
t_{SLCH}	t_{CSS}	CS# Active Setup Time (relative to SCLK)	5		ns
t_{CHSL}		CS# Not Active Hold Time (relative to SCLK)	5		ns
t_{DVCH}	t_{DSU}	Data In Setup Time	2		ns
t_{CHDX}	t_{DH}	Data In Hold Time	5		ns
t_{CHSH}		CS# Active Hold Time (relative to SCLK)	5		ns
t_{SHCH}		CS# Not Active Setup Time (relative to SCLK)	5		ns
t_{SHSL}	t_{CSH}	CS# Deselect Time	100		ns
t_{SHQZ}	t_{DIS}	Output Disable Time		9	ns
t_{CLQV}	t_V	Clock Low to Output Valid		9	ns

t CLQX	tHO	Output Hold Time	0		ns
t HLCH		HOLD# Setup Time (relative to SCLK)	5		ns
t CHHH		HOLD# Hold Time (relative to SCLK)	5		ns
t HHCH		HOLD Setup Time (relative to SCLK)	5		ns
t CHHL		HOLD Hold Time (relative to SCLK)	5		ns
t HHQX	tLZ	HOLD to Output Low-Z		9	ns
t HLQZ	tHZ	HOLD# to Output High-Z		9	ns

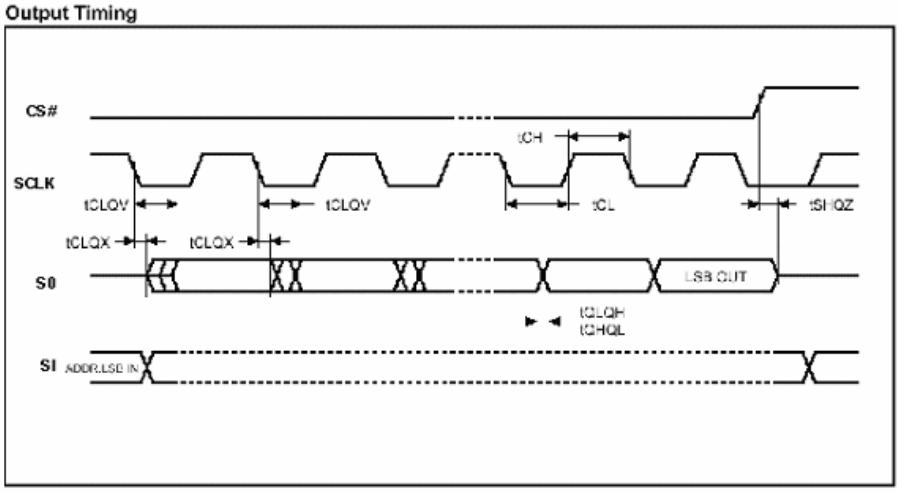
Serial Input Timing



Hold Timing



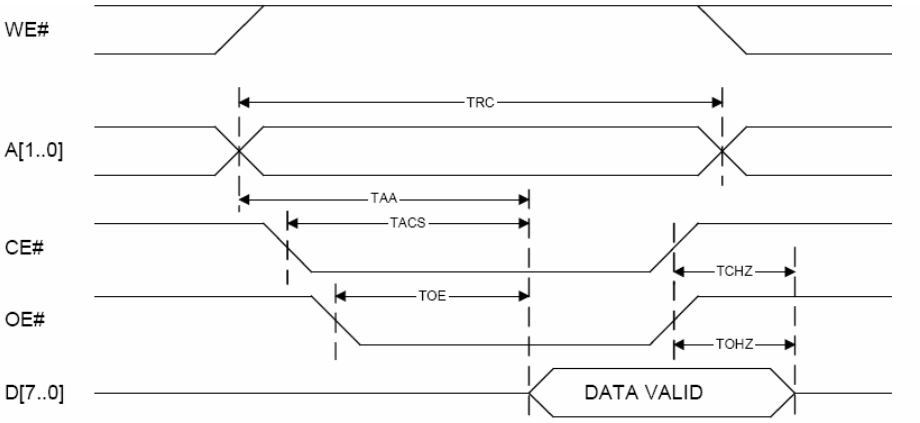
Output Timing



7.3.2 PLII 接口模式下 AC 特性

7.3.2.1 读周期时间特性

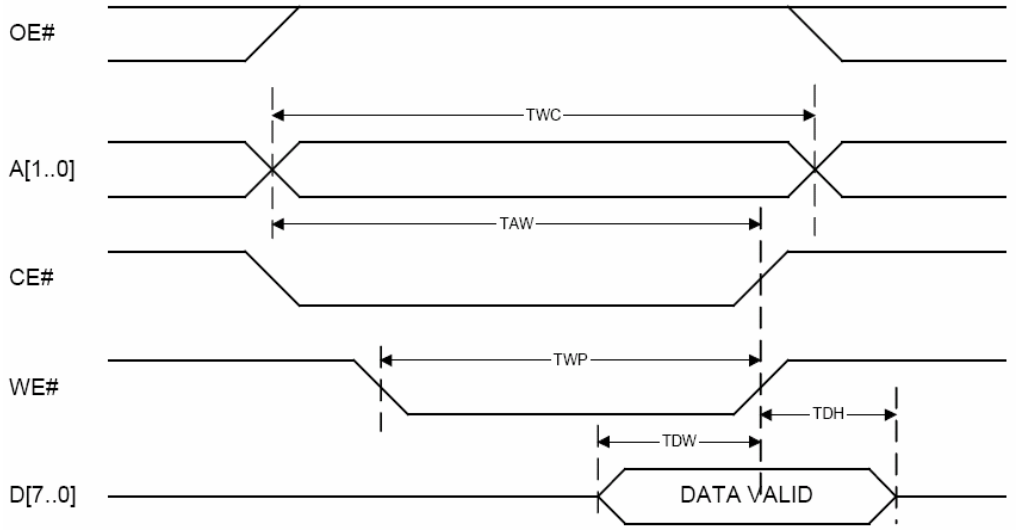
Symbol	Parameter	Min.	Max.	Unit
TRC	Read Cycle Time	70	-	ns
TAA	Address Access Time	-	60	ns
TACS	Chip Select Access Time	-	60	ns
TOE	Output Enable to Output Valid	-	30	ns
TCHZ	Chip Deselect to Output in High-Z	-	10	ns
TOHZ	Output Disable to Output in High-Z	-	10	ns



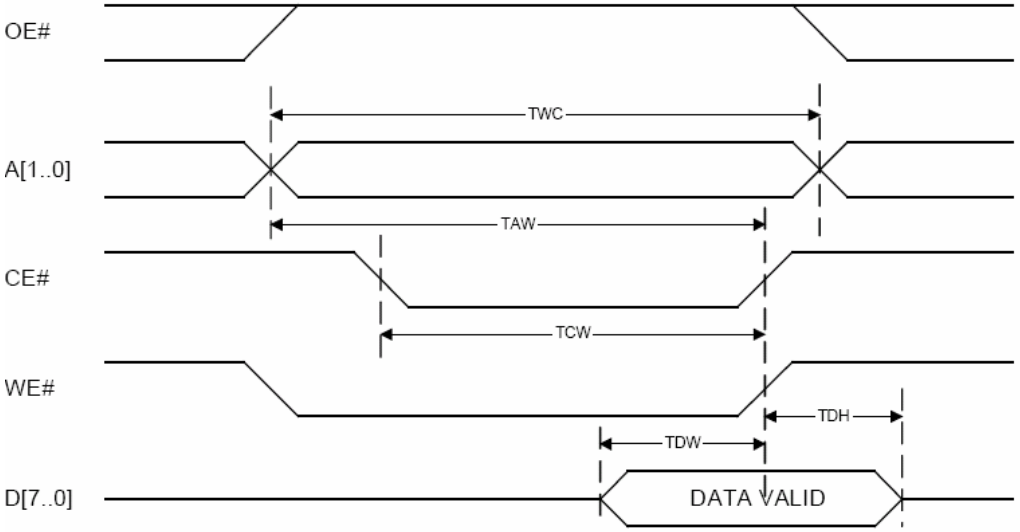
读周期时序波形图

7.3.2.2 写周期时间特性

Symbol	Parameter	Min.	Max.	Unit
TWC	Write Cycle Time	70		ns
TAW	Address Valid to End-of-Write	60		ns
TCW	Chip Select to End-of-Write	35		ns
TWP	Write Pulse Width	35		ns
TDW	Dada to Write Time Overlap	30		ns
TDH	Data Hold from Write Time	5		ns



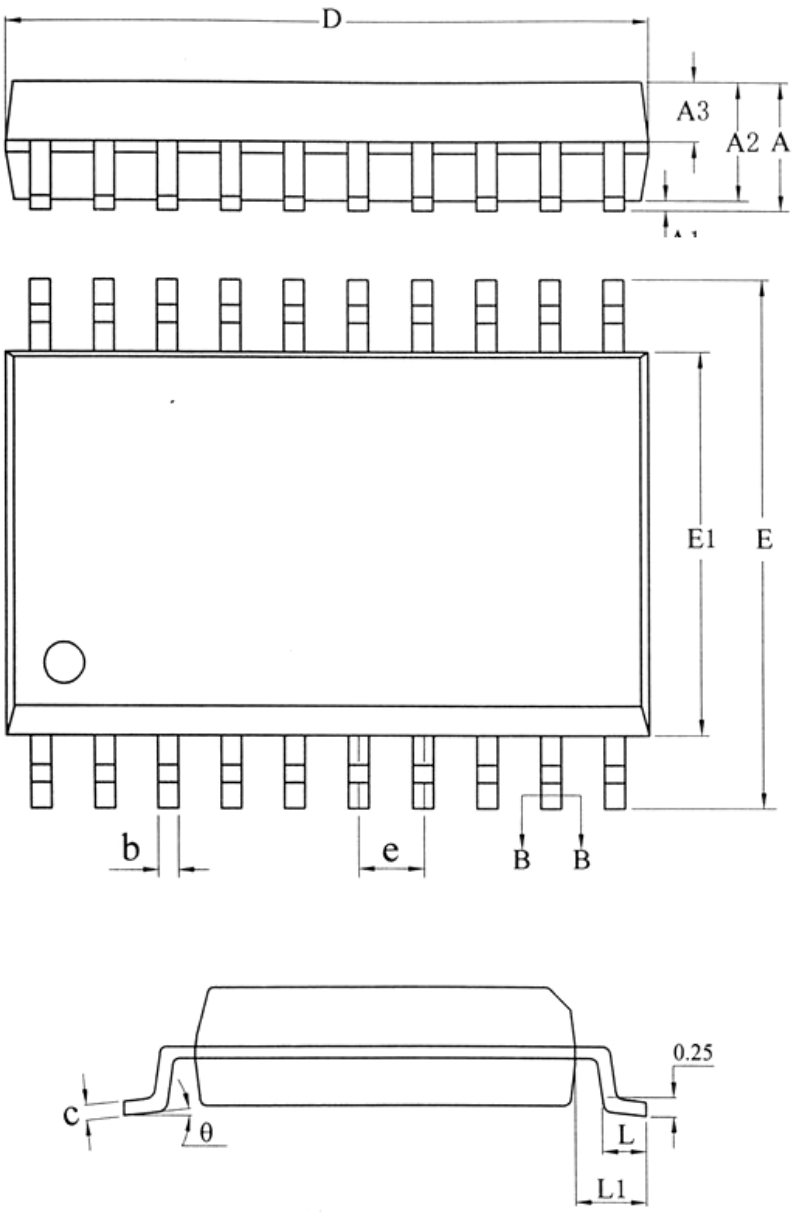
写周期时序波形图（WE#控制的时序）



写周期时序波形图（CE#控制的时序）

8 封装尺寸：SOP20

单位: mm



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	2.70
A1	0.10	0.20	0.30
A2	2.10	2.30	2.50
A3	0.92	1.02	1.12
b	0.35	—	0.44
b1	0.34	0.37	0.39
c	0.26	—	0.31
c1	0.24	0.25	0.26
D	12.60	12.80	13.00
E	10.10	10.30	10.50
E1	7.30	7.50	7.70
e	1.27BSC		
L	0.70	0.85	1.00
L1	1.40BSC		
θ	0	—	8°

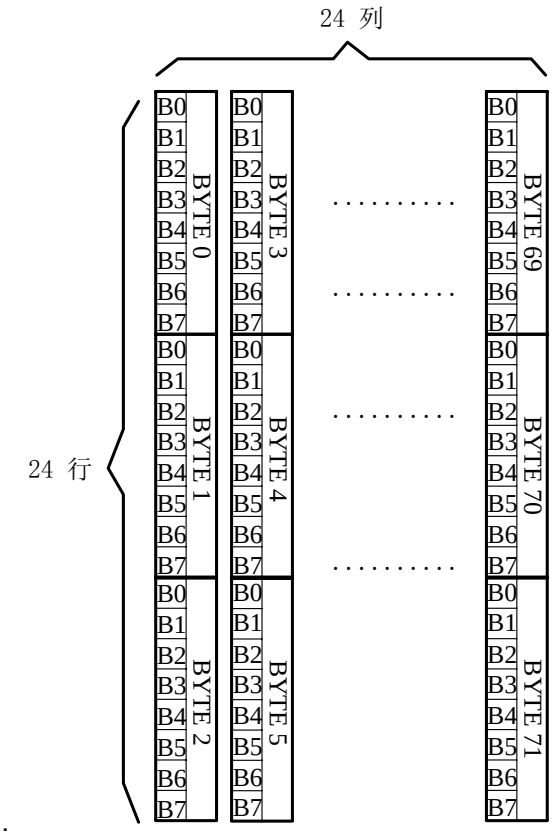
9 功能描述

9.1 汉字点阵排列格式

每个汉字在字库中是以汉字字模的形式存储的，每个点用一个二进制位表示，存 1 的点，当显示时可以在屏幕上显示亮点，存 0 的点，则在屏幕上不显示。点阵排列格式为竖置竖排：即一个字节的高位表示下面的点，低位表示上面的点，（如果用户按 word mode 读取点阵数据，请注意高低字节的顺序）排满一列的点后再排下一列。这样把点阵信息用来直接在显示器上按上述规则显示，则将出现对应的汉字。

9.1.1 24X24 点汉字排列格式

24X24 点汉字的信息需要 72 个字节（BYTE 0 – BYTE 71）来表示。该 24X24 点汉字的点阵数据是竖置横排的，其具体排列结构如下图：.



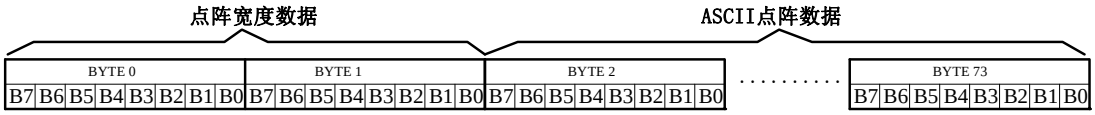
24X24 点汉字排列格式图

9.1.2 24 点阵 ASCII 方头字符排列格式

24 点阵 ASCII 方头字符的信息需要 74 个字节（BYTE 0 – BYTE73）来表示。

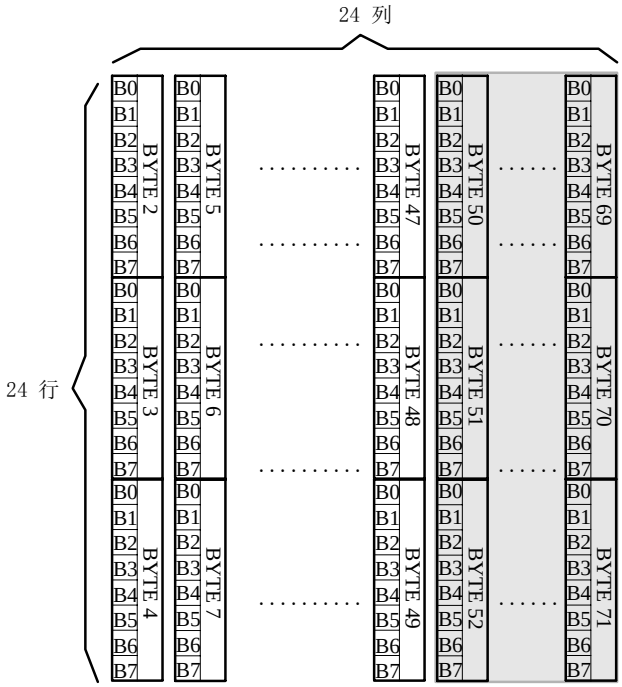
■ 存储格式

由于 ASCII 方头字符是不等宽的，因此在存储格式中 BYTE0~ BYTE1 存放点阵宽度数据，BYTE2-73 存放点阵数据。具体格式见下图：



■ 存储结构

不等宽 ASCII 字符的存储结构是以宽度为 BYTE 取整的，根据不同字符宽度会出现相应的空白区。根据 BYTE0~ BYTE1 所存放点阵的宽度数据，可以对还原下一个字的显示或排版留作参考。



24 点阵 ASCII 方头字符排列格式图

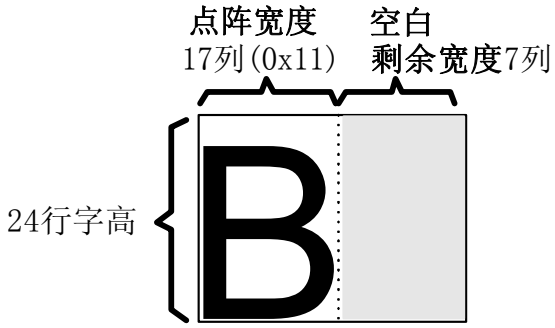
例如：ASCII 方头字符 B

0-73 BYTE 的点阵数据是: 00 11 00 00 00 00 00 00 C0 FF 7F C0 FF 7F C0 FF 7F C0 E1 70 C0 E1 70 C0 E1 70 C0 E1 70 C0 E1 70 C0 E1 70 C0 E1 70 C0 F3 70 C0 FF 79 80 FF 3F 00 9E 3F 00 00 0F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

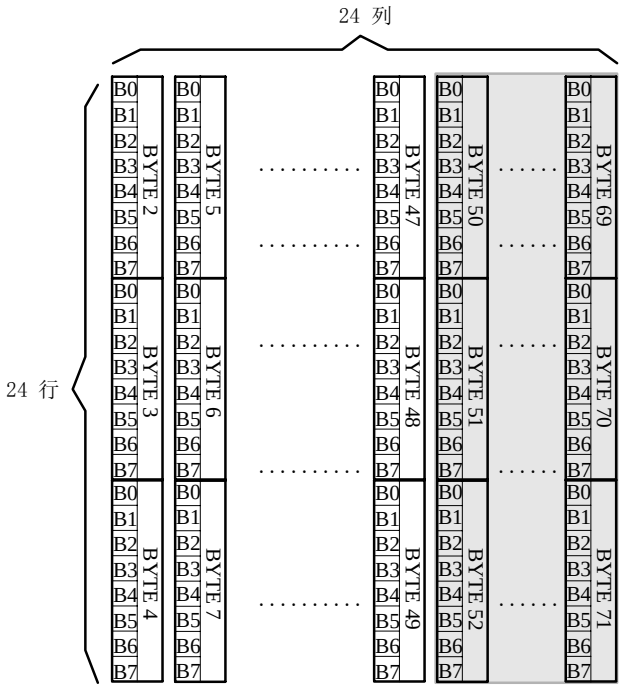
其中：

BYTE0~ BYTE1: 00 11

为 ASCII 方头字符 B 的点阵宽度数据，即：17 位宽度。字符后面有 7 位空白区，可以在排版下一个字时考虑到这一点，将下一个字的起始位置前移。（见下图）



BYTE2-73: 00 00 00 00 00 00 C0 FF 7F C0 FF 7F C0 FF 7F C0 E1 70 C0 E1 70 C0 E1 70 C0 E1 70 C0 E1 70 C0 E1 70 C0 F3 70 C0 FF 79 80 FF 3F 00 9E 3F 00 00 0F 00 00 00 00 00 00



24 点阵 ASCII 白正字符排列格式图

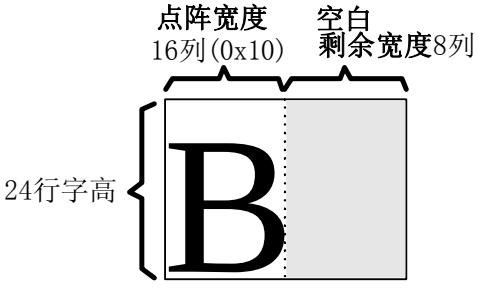
例如：ASCII 白正字符 B

0-73BYTE 的点阵数据是: 00 10 00 00 00 20 00 10 20 00 10 E0 FF 1F E0 FF 1F 20 10 10 20 10 10 20 10 10 20 10 10 20 10 10 60 30 10 C0 38 18 C0 6F 0C 80 E7 0F 00 80 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

其中：

BYTE0~ BYTE1: 00 10

为 ASCII 白正字符 B 的点阵宽度数据，即：16 位宽度。字符后面有 8 位空白区，可以在排版下一个字时考虑到这一点，将下一个字的起始位置前移。（见下图）

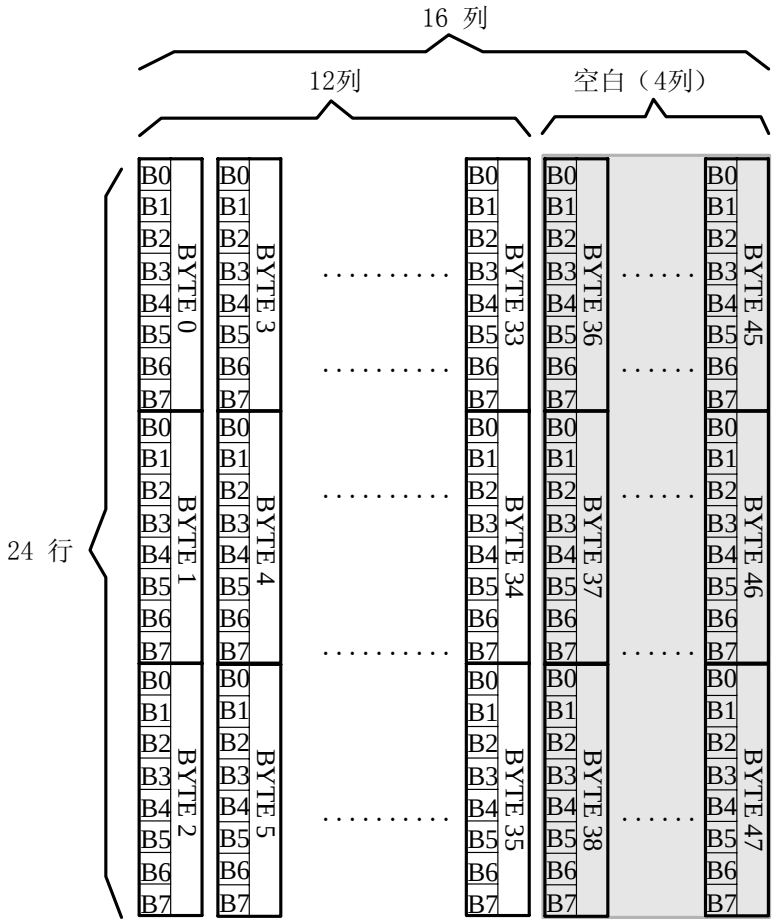


BYTE2-73: 00 00 00 20 00 10 20 00 10 E0 FF 1F E0 FF 1F 20 10 10 20 10 10 20 10 10 20 10 10 20 10 10 60 30 10 C0 38 18 C0 6F 0C 80 E7 0F 00 80 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

为 ASCII 白正字符 B 的点阵数据。

9.1.4 12X24 点扩展字符排列格式

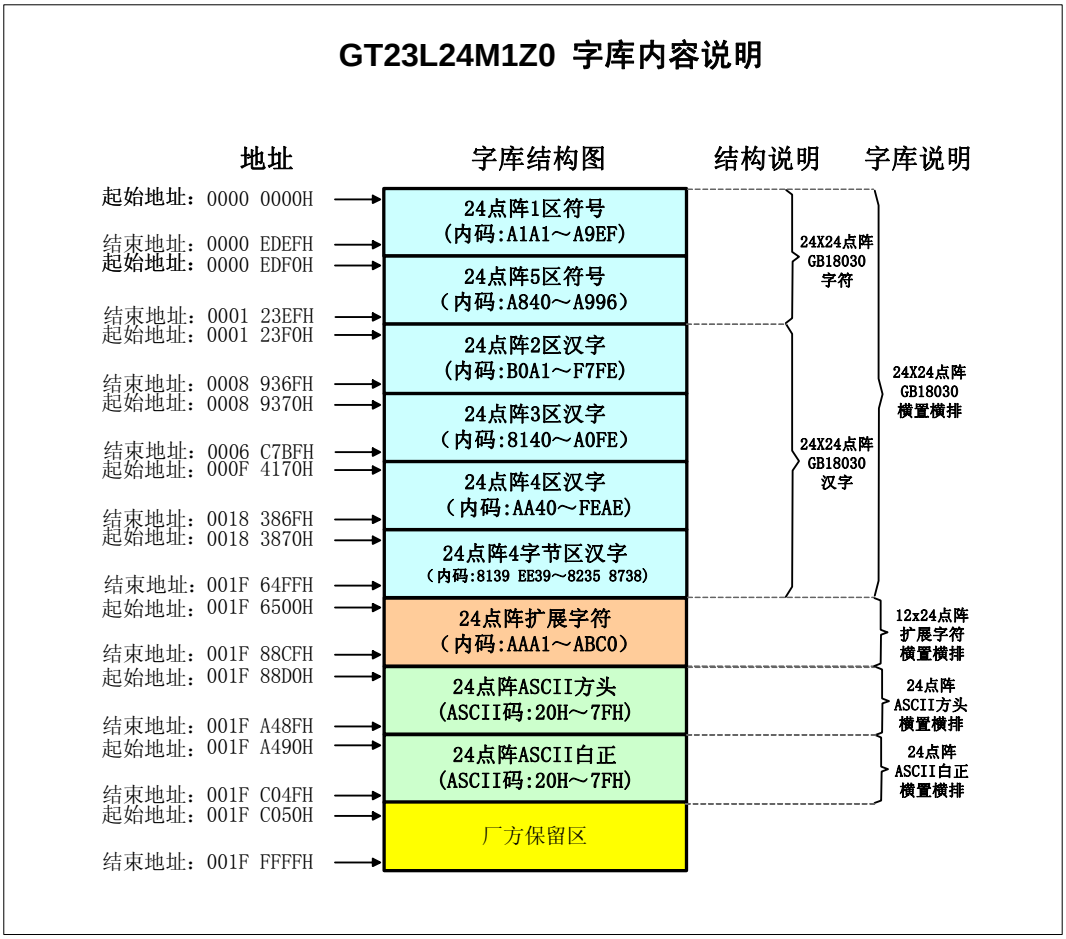
12X24 点扩展字符的信息需要 48 个字节（BYTE 0 – BYTE 47）来表示。但是由于该字符为标准的 12X24 点格式，而存储空间是按照 12X24 点 BYTE 取整进行存储的（即 16x24 点的容量为 48 BYTES），注意在排版时作相应的调整。



12X24 点扩展字符排列格式

9.2 汉字点阵字库结构

- GB18030 标准点阵字库
  标准点阵字符 1 区和 5 区分别对应 GB18030 码位的 A1A1~A996 共计 1038 个字符；
  点阵字库中的 2 区、3 区、4 区分别对应 GB18030 双字节码位 B0A1~FEAE 共计 21003 个汉字；
  点阵字库中的四字节区汉字码位范围 8139 EE39~8235 8738 共计 6530 个汉字。
- ASCII 方头为不等宽 ASCII，所对应码位 20~7E 共计 96 个字符。
- ASCII 白正为不等宽 ASCII，所对应码位 20~7E 共计 96 个字符。
- 1 套 12X24 点扩展字符对应 GB18030 码位的 AAA1~ABC0 共计 126 个字符。



标准点阵字库数据在汉字芯片中的地址速查表

序号	标准点阵字库数据	在汉字芯片地址
1	24X24 点阵 GB18030	0000 0000H
2	12×24 点　　扩展字符	001F 6500H
3	24 点阵　方头（96 字符）	001F 88D0H
4	24 点阵　白正（96 字符）	001F A490H

9.3 汉字点阵在芯片中的地址计算方法

用户只要知道汉字的内码，就可以计算出该汉字点阵在芯片中的地址，然后就可从该地址连续读出点阵信息用于显示。

24X24 点阵字库地址分配(字节地址)：0000 0000 H ~ 001F 64FF H,

9.3.1 24X24 汉字点阵地址的计算

地址的计算由下面的函数实现（ANSI C 语言编写）

```
/*****************************************************************************************************
函数：unsigned long gt(unsigned char c1, unsigned char c2, unsigned char c3, unsigned char c4)
功能：计算汉字点阵在芯片中的地址
参数：c1,c2,c3,c4：4 字节汉字内码通过参数 c1,c2,c3,c4 传入，双字节内码通过参数 c1,c2 传入，c3=0,c4=0

返回：汉字点阵的字节地址(byte address)。如果用户是按 word mode 读取点阵数据，则其地址(word address)为字节地址除以 2，即：word address = byte address / 2 .

例如：BaseAdd: 说明汉字点阵数据在字库芯片中的起始地址，即 BaseAdd＝0x0000 0000;
      “啊”字的内码为 0xb0a1,则 byte address = gt(0xb0,0xa1,0x00,0x00) *72+BaseAdd
                                word address = byte address / 2
      “ 上”字的内码为 0x8139ee39,则 byte address = gt(0xee,0x39,0x81,0x39) *72+ BaseAdd
                                word address = byte address / 2
*****************************************************************************************************/
 unsigned long gt (unsigned char c1, unsigned char c2, unsigned char c3, unsigned char c4)
 {
      unsigned long h=0;
      if(c2==0x7f) return (h);
      if(c1>=0xA1 && c1 <= 0xAB && c2>=0xa1)          //Section 1
           h= (c1 - 0xA1) * 94 + (c2 - 0xA1);
      else if(c1>=0xa8 && c1 <= 0xa9 && c2<0xa1)      //Section 5
      {
           if(c2>0x7f)
                c2--;
           h=(c1-0xa8)*96 + (c2-0x40)+846;
      }
      if(c1>=0xb0 && c1 <= 0xf7 && c2>=0xa1)          //Section 2
           h= (c1 - 0xB0) * 94 + (c2 - 0xA1)+1038;
      else if(c1<0xa1 && c1>=0x81)                    //Section 3
      {
           if(c2>0x7f)
                c2--;
           h=(c1-0x81)*190 + (c2-0x40) + 1038 +6768;
      }
      else if(c1>=0xaa && c2<0xa1)                    //Section 4
      {
```

```
            if(c2>0x7f)
                c2--;
            h=(c1-0xaa)*96 + (c2-0x40) + 1038 +12848;
        }
        if(c2>=0x30 && c2<=0x39)                    //Extended Section (With 4 BYTES InCode)
        {
            if(c4<0x30 || c4>0x39)
                return(h=0);
            h=(c3-0x81)*12600+(c4-0x39)*1260+(c1-0xee)*10+(c2-0x39)+22046;
            if(h>=22046 && h<=22046+6530)
                return(h);
            else
            {   h=0;
                return(h);
            }
        }
        return(h);
    }
```

9.3.2 12X24 点扩展字符地址的计算

12X24 点扩展字符在芯片中的字节地址的计算方法如下：

参数：c1,c2：2 字节汉字内码通过参数 c1,c2 传入,如:C1=AA;C2=A3;。

BaseAdd: 说明点阵数据在字库芯片中的起始地址。

```
BaseAdd=1F6500H;
if (C1 >= AAH) and (C2>=A1H)and(C2<=FEH) then          //内码为 AAA1~AAFE 的 94 个扩展字符
   Address = (C2– A1H ) * 48+ BaseAdd
if (C1 >= ABH) and (C2>=A1H)and(C2<=C0H) then          //内码为 ABA1~ABC0 的 32 个扩展字符
   Address = (C2– A1H +96) * 48+ BaseAdd
```

若作为标准半角 ASCII （码位 20H~7E 95 个）的调用可以参考以下算法：

```
LSB 为 ASCII 码
BaseAdd=0x1F6500;
if (LSB>=0x 20 && LSB<=0x 7E) then    //内码为 20H~7E 的 95 个 ASC 字符
     Address = (LSB–0x 20 ) * 48+ BaseAdd
```

9.3.3 ASCII 字符地址的计算

ASCIICode 表示 ASCII 码（8bits）

Address 表示汉字或 ASCII 字符点阵在芯片中的字节地址。

BaseAdd: 说明点阵数据在字库芯片中的起始地址。

(1) 24 点阵 ASCII 方头字符在芯片中的字节地址的计算方法如下：

```
BaseAdd=1F88D0H
if (ASCIICode >= 20H) and (ASCIICode <= 7EH) then
     Address = (ASCIICode – 20H ) * 74+BaseAdd
```

(2) 24 点阵 ASCII 白正字符在芯片中的字节地址的计算方法如下：

```
BaseAdd=1FA490H H
if (ASCIICode >= 20H) and (ASCIICode <= 7EH) then
     Address = (ASCIICode – 20H ) * 74+BaseAdd
```

10 附录

10.1 GB18030　1 区和 5 区 (字符区)

GB18030 标准点阵字符 1 区和 5 区分别对应码位的 A1A1~A996 共计 1038 个字符；

GB18030　1 区

A1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A			、	。	·	ˉ	ˇ	¨	〃	々	—	～	‖	…	‘	’
B	“	”	〔	〕	〈	〉	《	》	「	」	『	』	〖	〗	【	】
C	±	×	÷	∶	∧	∨	∑	∏	∪	∩	∈	∷	√	⊥	∥	∠
D	⌒	⊙	∫	∮	≡	≌	≈	∽	∝	≠	≮	≯	≤	≥	∞	∵
E	∴	♂	♀	°	′	″	℃	$	¤	¢	£	‰	§	№	☆	★
F	○	●	◎	◇	◆	□	■	△	▲	※	→	←	↑	↓	〓	

A2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		ⅰ	ⅱ	ⅲ	ⅳ	ⅴ	ⅵ	ⅶ	ⅷ	ⅸ	ⅹ					
B		1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.
C	16.	17.	18.	19.	20.	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
D	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)	①	②	③	④	⑤	⑥	⑦
E	⑧	⑨	⑩	€		㈠	㈡	㈢	㈣	㈤	㈥	㈦	㈧	㈨	㈩	
F		Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ	Ⅶ	Ⅷ	Ⅸ	Ⅹ	Ⅺ	Ⅻ			

A3	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		！	＂	＃	￥	％	＆	＇	（	）	＊	＋	，	－	．	／
B	０	１	２	３	４	５	６	７	８	９	：	；	＜	＝	＞	？
C	＠	Ａ	Ｂ	Ｃ	Ｄ	Ｅ	Ｆ	Ｇ	Ｈ	Ｉ	Ｊ	Ｋ	Ｌ	Ｍ	Ｎ	Ｏ
D	Ｐ	Ｑ	Ｒ	Ｓ	Ｔ	Ｕ	Ｖ	Ｗ	Ｘ	Ｙ	Ｚ	［	＼	］	＾	＿
E	｀	ａ	ｂ	ｃ	ｄ	ｅ	ｆ	ｇ	ｈ	ｉ	ｊ	ｋ	ｌ	ｍ	ｎ	ｏ
F	ｐ	ｑ	ｒ	ｓ	ｔ	ｕ	ｖ	ｗ	ｘ	ｙ	ｚ	｛	｜	｝	￣	

GB18030　1 区

A4	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		あ	あ	ぃ	い	う	う	ぇ	え	ぉ	お	か	が	き	ぎ	く
B	ぐ	け	げ	こ	ご	さ	ざ	し	じ	す	ず	せ	ぜ	そ	ぞ	た
C	だ	ち	ぢ	っ	つ	づ	て	で	と	ど	な	に	ぬ	ね	の	は
D	ば	ぱ	ひ	び	ぴ	ふ	ぶ	ぷ	へ	べ	ぺ	ほ	ぼ	ぽ	ま	み
E	む	め	も	ゃ	や	ゅ	ゆ	ょ	よ	ら	り	る	れ	ろ	ゎ	わ
F	ゐ	ゑ	を	ん												

A5	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		ァ	ア	ィ	イ	ゥ	ウ	ェ	エ	ォ	オ	カ	ガ	ヰ	ギ	ク
B	グ	ケ	ゲ	コ	ゴ	サ	ザ	シ	ジ	ス	ズ	セ	ゼ	ソ	ゾ	タ
C	ダ	チ	ヂ	ッ	ツ	ヅ	テ	デ	ト	ド	ナ	ニ	ヌ	ネ	ノ	ハ
D	バ	パ	ヒ	ビ	ピ	フ	ブ	プ	ヘ	ベ	ペ	ホ	ボ	ポ	マ	ミ
E	ム	メ	モ	ャ	ヤ	ュ	ユ	ョ	ヨ	ラ	リ	ル	レ	ロ	ヮ	ワ
F	ヰ	ヱ	ヲ	ン	ヴ	ヵ	ヶ									

A6	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
B	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω							
C		α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
D	π	ρ	σ	τ	υ	φ	χ	ψ	ω	’	°	`	:	;	!	?
E	︵	︶	︷	︸	︿	﹀	︽	︾	﹁	﹂	﹃	﹄	﹅	﹆	﹇	﹈
F	︿	﹀	｜	⋮	｜	⦚										

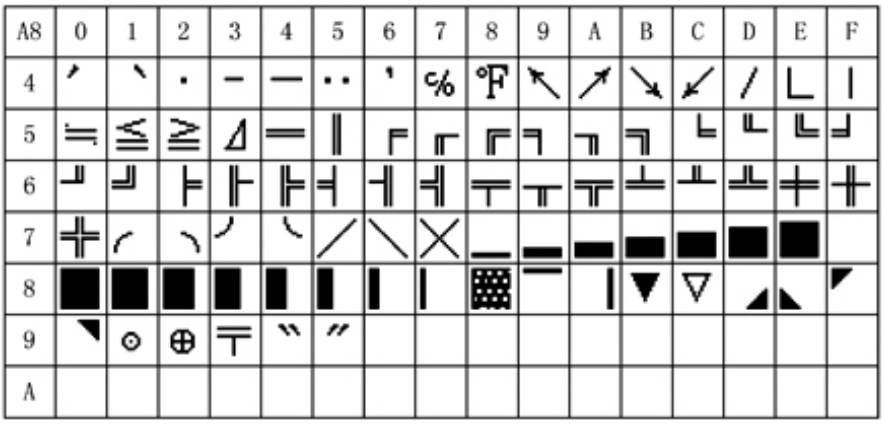
GB18030　1 区

A7	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н
B	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э
C	Ю	Я														
D		а	б	в	г	д	е	ё	ж	з	и	й	к	л	м	н
E	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э
F	ю	я														

A8	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		ā	á	ǎ	à	ē	é	ě	è	ī	í	ǐ	ì	ō	ó	ǒ
B	ò	ū	ú	ǔ	ù	ǖ	ǘ	ǚ	ǜ	ü	ê	ɑ	ḿ	ń	ň	ǹ
C	ɡ					ㄅ	ㄆ	ㄇ	ㄈ	ㄉ	ㄊ	ㄋ	ㄌ	ㄍ	ㄎ	ㄏ
D	ㄐ	ㄑ	ㄒ	ㄓ	ㄔ	ㄕ	ㄖ	ㄗ	ㄘ	ㄙ	ㄚ	ㄛ	ㄜ	ㄝ	ㄞ	ㄟ
E	ㄠ	ㄡ	ㄢ	ㄣ	ㄤ	ㄥ	ㄦ	ㄧ	ㄨ	ㄩ						
F																

A9	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A					—	━	|	┃	┄	┅	┆	┇	┈	┉	┊	┋
B	┌	┍	┎	┏	┐	┑	┒	┓	└	┕	┖	┗	┘	┙	┚	┛
C	├	┝	┞	┟	┠	┡	┢	┣	┤	┥	┦	┧	┨	┩	┪	┫
D	┬	┭	┮	┯	┰	┱	┲	┳	┴	┵	┶	┷	┸	┹	┺	┻
E	┼	┽	┾	┿	╀	╁	╂	╃	╄	╅	╆	╇	╈	╉	╊	╋
F																

GB18030　5 区



A8	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	′	ˋ	·	‒	―	‥	'	℅	℉	↖	↗	↘	↙	/	∟	∣
5	≒	≦	≧	⊿	═	‖	╒	╓	╔	╕	╖	╗	╘	╙	╚	╛
6	╜	╝	╞	╟	╠	╡	╢	╣	╤	╥	╦	╧	╨	╩	╪	╫
7	╬	╭	╮	╯	╰	╱	╲	╳	▁	▂	▃	▄	▅	▆	▇	
8	▇	▇	▇	▇	▇	▇	▇	▇	▩	▔	▕	▼	▽	◢	◣	◤
9	◥	⊙	⊕	╤	‶	″										
A																

A9	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	∣	∥	∣∣∣	ㄨ	ㄩ	⊥	⏊	⏊	夂	㊣	mg	kg	mm	cm	km	m²
5	cc	KM	ln	log	mil	:	¬	¦		TEL	㈱		‒			
6	⌒	‶	°	ˋ	⌇	⌇	⌇	⌇	⋯	⋯	⌇	⌇	⋯	⋯	⌇	，
7	、	.	;	:	?	!	(	)	{	}	〔	〕	#	&	火	
8	+	−	<	>	=	＼	$	%	@	▣	▣	▣	▣	▣	▣	▣
9	▣	▣	▣	▣	▣	▣	○									
A																

10.2 扩展字符区字符

扩展字符区的内码组成为 AAA1~ABC0 共计 126 个字符。

扩展字符区

AA	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		!	"	#	¥	%	&	'	(	)	*	+	,	-	.	/
B	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
C	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
D	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
E	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
F	p	q	r	s	t	u	v	w	x	y	z	{	|	}	¯	

AB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		ā	á	ǎ	à	ē	é	ě	è	ī	í	ǐ	ì	ō	ó	ǒ
B	ò	ū	ú	ǔ	ù	ǖ	ǘ	ǚ	ǜ	ü	ê	ɑ	ḿ	ń	ň	ǹ
C	g															
D																
E																
F																