GT21L16S1Z3-S

标准汉字字库芯片

15X16 点阵

竖置竖排

GT-三维部件输入法

（SPI 接口）

用户手册

VER 2.7

2007-Q3

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	2006-6	
V2.51	修正了扩展字符区的 8X16ASCII 对应传统 ASCII 的算法	2006-8	
V2.6	统一规范，修改错误	2007-4	详见修改说明
V2.7	增加 GT 输入法，并更名为 GT21L16S1Z3-S	2007-9	

目　　录

1 概述.....4
2 特点.....5
3 引脚配置.....6
4 引脚描述与接口连接.....6
4.1 引脚描述.....6
4.2 HOST CPU 主机 SPI 接口电路示意图.....7
5 操作指令.....9
5.1 指令参数.....9
5.2 Read Data Bytes（一般读取）.....9
5.3 Read Data Bytes at Higher Speed（快速读取点阵数据）.....10
6 电气特性.....11
6.1 绝对最大额定值.....11
6.2 DC 特性.....11
6.3 AC 特性.....11
7 封装尺寸.....13
8 功能描述.....14
8.1 汉字点阵排列格式.....14
8.1.1 15X16 点汉字排列格式.....14
8.1.2 5X7 点 ASCII 字符排列格式.....15
8.1.3 8X16 点 ASCII 扩展字符排列格式.....15
8.1.4 Unicode 拉丁文、里尔文字符排列格式.....16
8.1.5 16 点阵 ASCII 方头字符排列格式.....16
8.2 汉字点阵字库结构.....18
8.3 汉字点阵在芯片中的地址计算方法.....19
8.3.1 15X16 汉字点阵地址的计算.....19
8.3.2 5X7 ASCII 字符地址的计算.....20
8.3.3 8X16 ASCII 扩展字符地址的计算.....20
8.3.4 16 点阵 ASCII 方头字符在芯片中的字节地址的计算方法如下：.....20
8.3.5 8X16 自定义特殊字符地址的计算.....21
8.3.6 8X16 自定义希腊字符区地址的计算.....21
8.3.7 8X16 拉丁语、西里尔语 Unicode 字符在芯片中的字节地址的计算方法.....21
9 附录.....22
9.1 GB2312 1 区 (字符区).....22
9.2 Unicode 字符区字符.....25
9.3 8×16 点扩展字符区 ASCII 字符（126 字符）.....28
9.4 5×7 点 ASCII 字符（96 字符）.....29
9.5 8×16 自定义特殊字符区 ASCII 字符（64 字符）.....30
9.6 8×16 自定义希腊字符区 ASCII 字符（16 字符）.....30
9.7 国家语言文系对照表.....31

1 概述

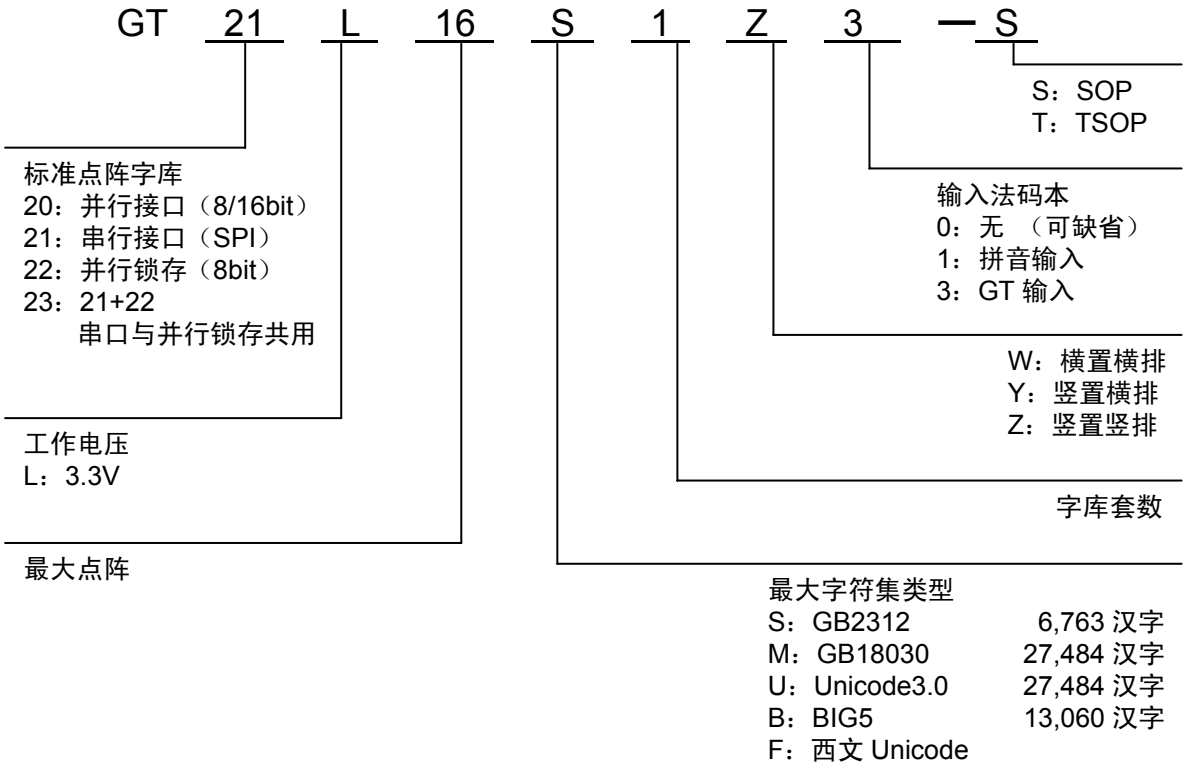
GT21L16S1Z3-S是一款支持GB2312字符集（6763汉字）的15X16点阵字库SPI串行接口芯片，排列格式为竖置竖排。同时芯片内含1套8X16点扩展ASCII字符（见附件：9.3）、1套5X7点ASCII字符（见附件：9.4），1套16点ASCII方头字符，以及Unicode字符区拉丁文系（LATIN）、西里尔文系（CYRILLIC）文系字符（见附件：9.2）基本可以涵盖81个国家及地区（见附件：9.7）。同时还有非常丰富的自定义区的各种图形符号（见附件：9.5），以及常用希腊字符（见附件：9.6）。

每个汉字在芯片中是以汉字点阵字模的形式存储的，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。

本字库芯片内含 GT 输入法码本，该输入法适用于数字小键盘，如手机、遥控器、电话终端、税控收款机等。

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
- 点阵排列方式：字节竖置竖排
- 时钟频率：40MHz(max.) @3.3V
- 工作电压：2.7V – 3.6V
- 电流：
  工作电流：10mA
  待机电流：10uA
- 封装：SOP8
- 字库内容：

汉 字 字 库	
15×16 点	GB2312（6,763 汉字） （846 字符）

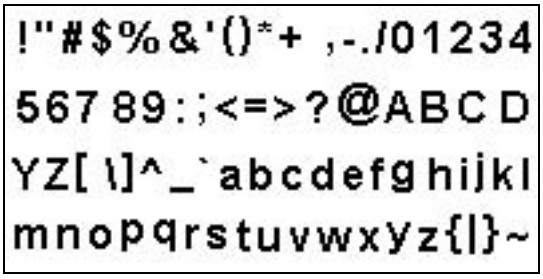
ASCII 字符	
5×7 点	ASCII 字符（96 字符）
8×16 点	ASCII 扩展字符（126 字符）
16 点阵	方头（96字符）
8×16 点阵	Unicode 字符区（626 字符） －拉丁文系（376 字符） －西里尔文系（250 字符） 自定义字符区（64 字符） 希腊字字符区（16 字符）

- 字型样张

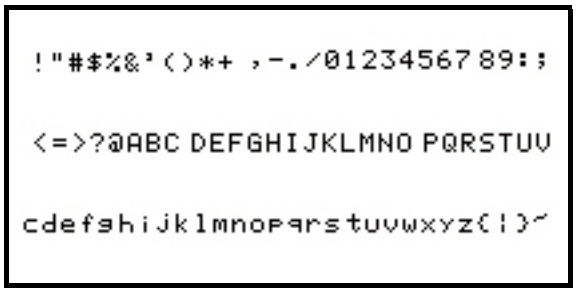
15×16 点 GB2312（6,763 汉字）



16 点阵 ASCII 方头（96 字符）

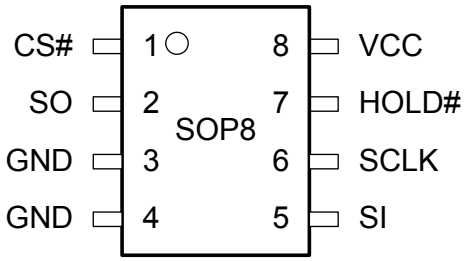


5×7 点 ASCII 字符（96 字符）



其它各种字符样张 详见附件

3 引脚配置



SOP8

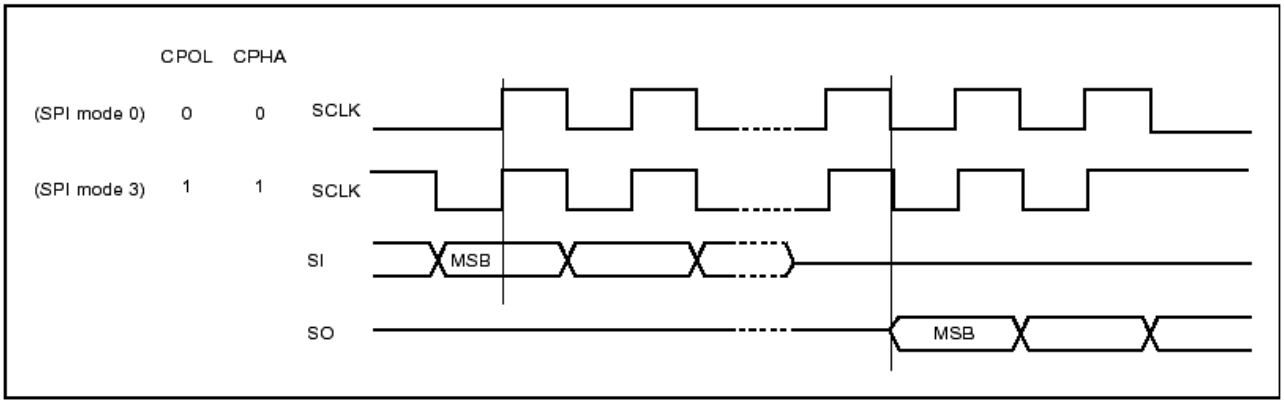
4 引脚描述与接口连接

4.1 引脚描述

8-PIN SOP

SOP8	名称	I/O	描述
1	CS#	I	片选输入（Chip enable input）
2	SO	O	串行数据输出 (Serial data output)
3	GND		地(Ground)
4	GND		地(Ground)
5	SI	I	串行数据输入 (Serial data input)
6	SCLK	I	串行时钟输入（Serial clock input）
7	HOLD#	I	总线挂起（Hold, to pause the device without）
8	VCC		电源(+ 3.3V Power Supply)

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。

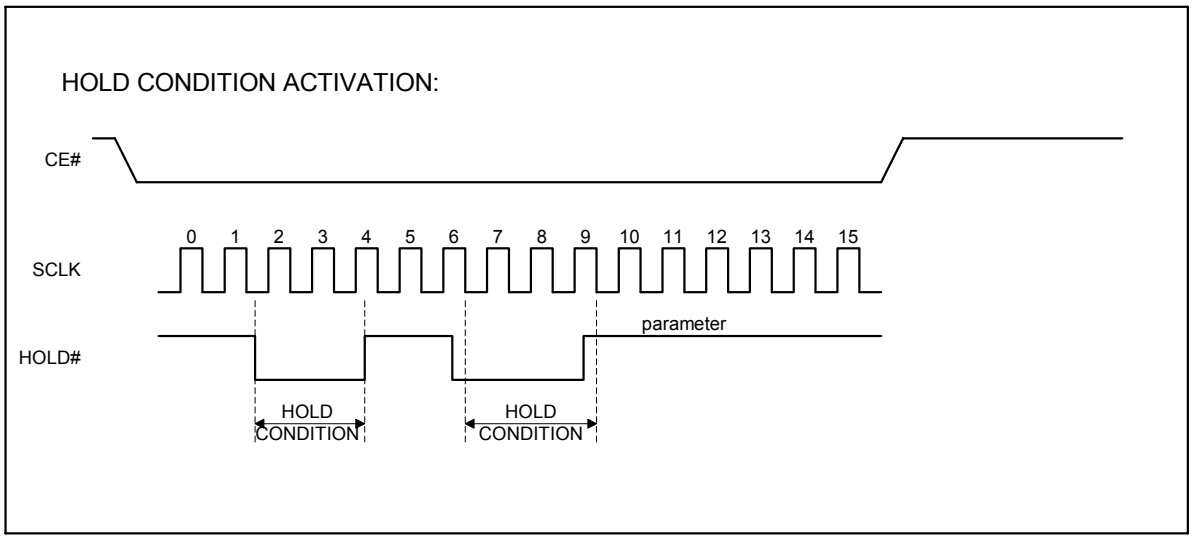


总线挂起输入（HOLD#）：

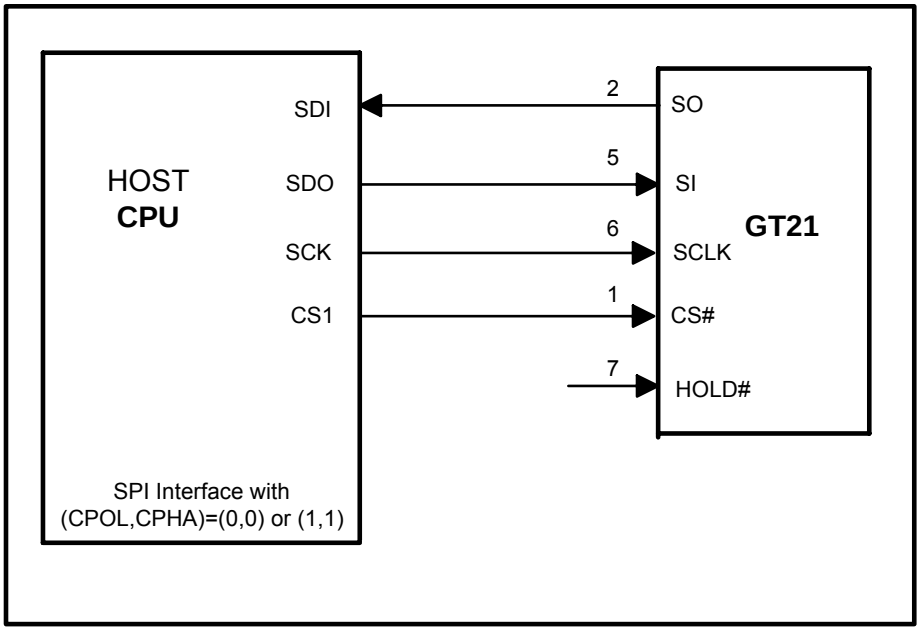
该信号用于片选信号有效期间暂停数据传输，在总线挂起期间，串行数据输出信号处于高阻态，芯片不对串行数据输入信号和串行时钟信号进行响应。

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。

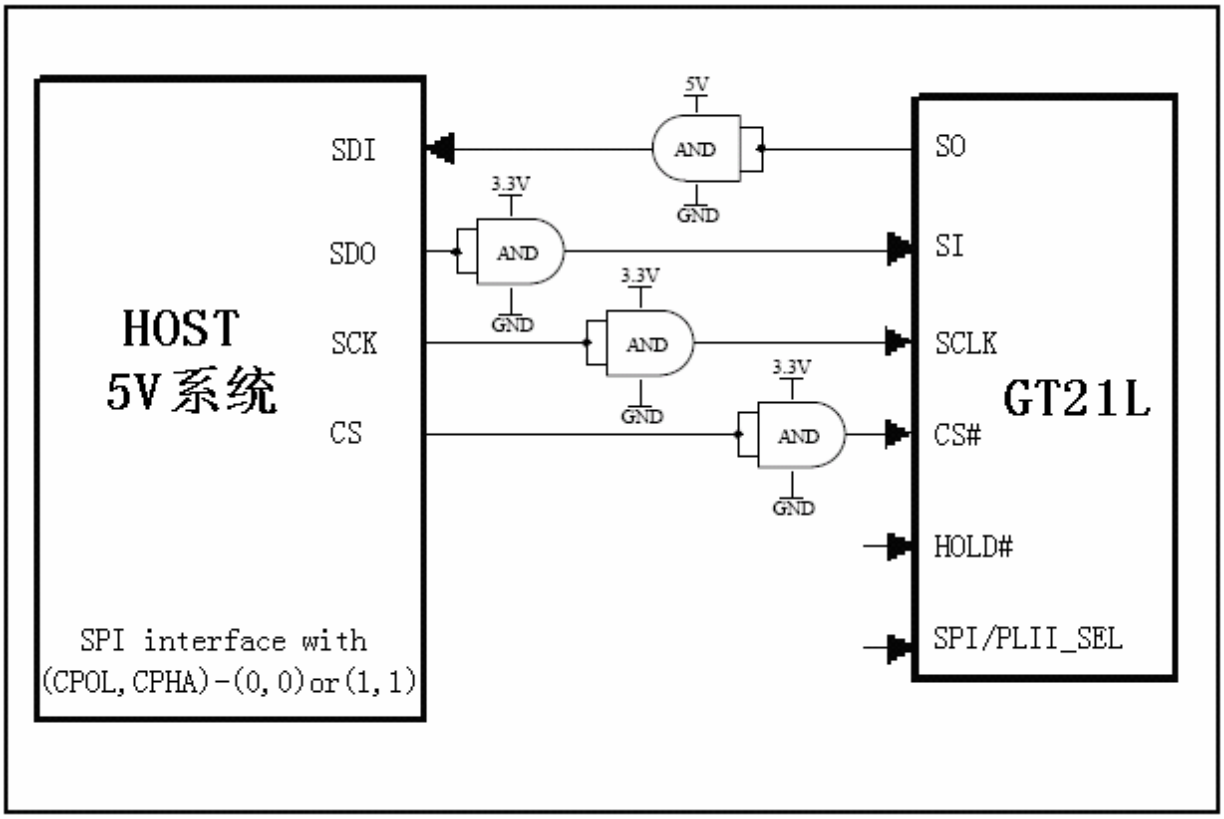


4.2 HOST CPU 主机 SPI 接口电路示意图



HOST CPU 主机 SPI 接口电路示意图

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 操作指令

5.1 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”)。.

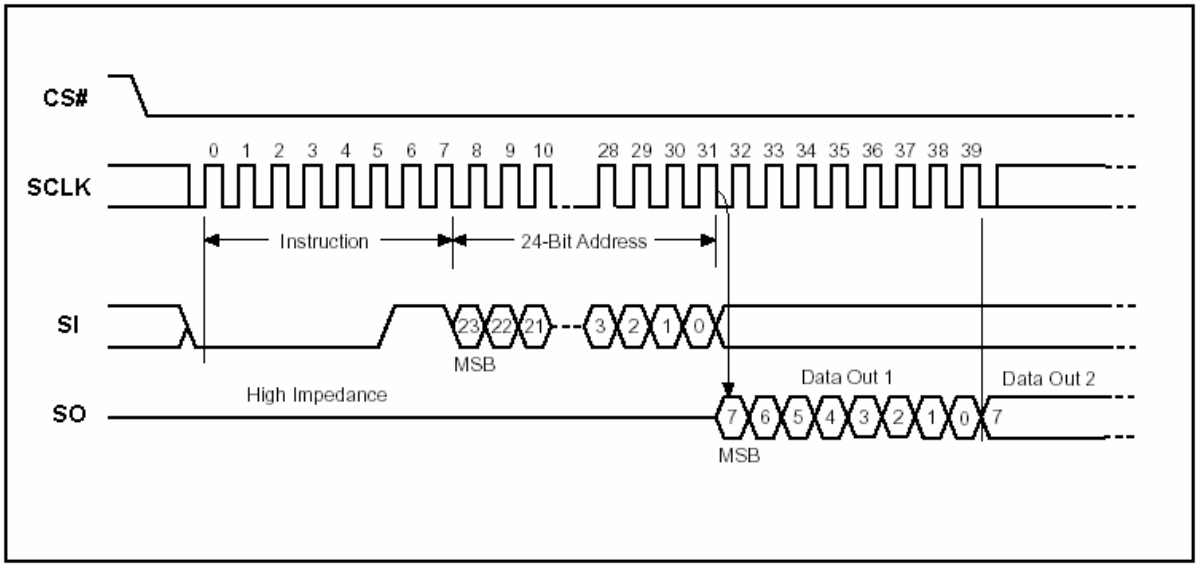
5.2 Read Data Bytes（一般读取）

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:



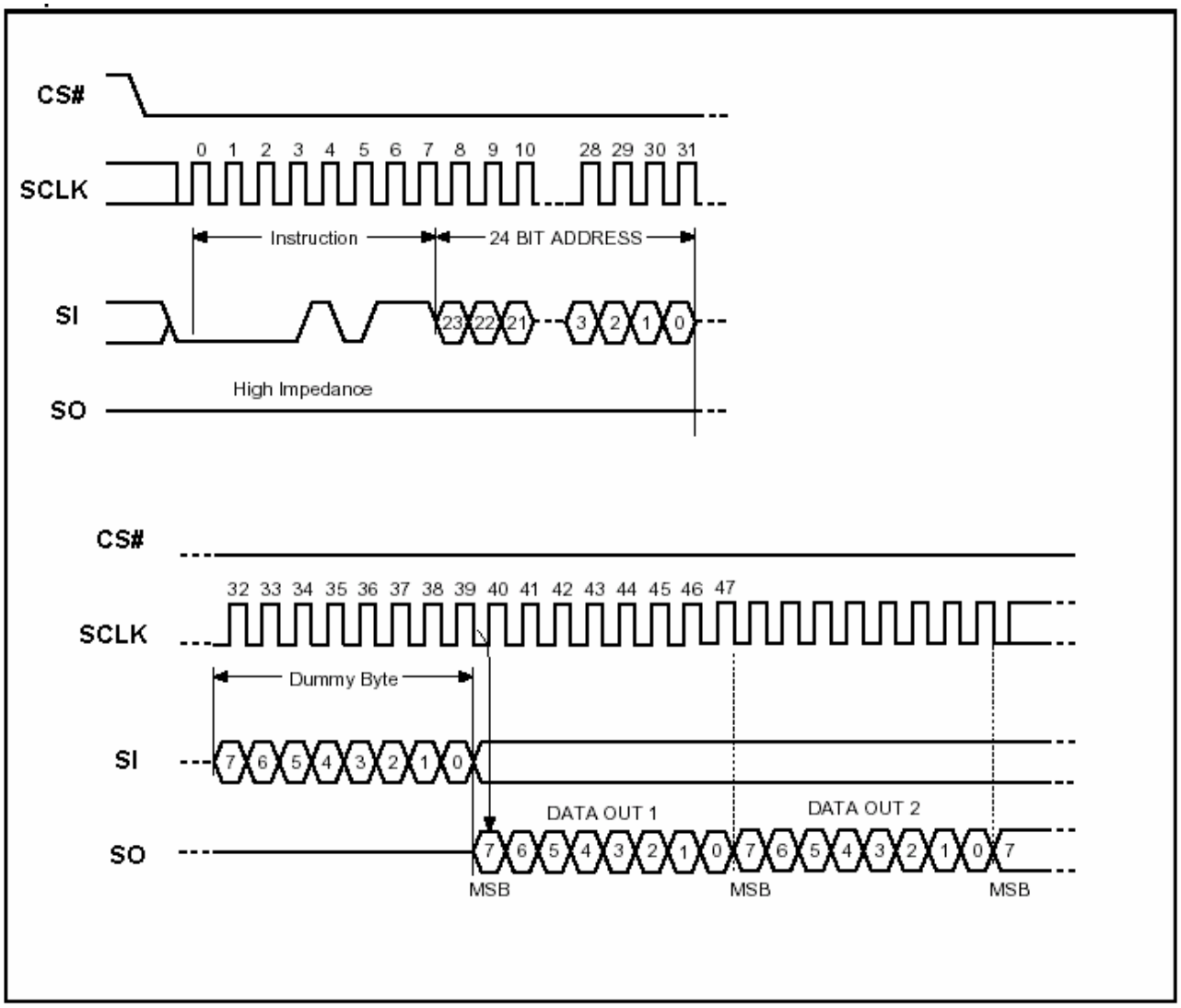
5.3 Read Data Bytes at Higher Speed（快速读取点阵数据）

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 电气特性

6.1 绝对最大额定值

Symbol	Parameter	Min.	Max.	Unit	Condition
T_{OP}	Operating Temperature	-10	70	℃	
T_{STG}	Storage Temperature	-65	150	℃	
VCC	Supply Voltage	-0.3	3.6	V	
V_{IN}	Input Voltage	-0.3	VCC+0.3	V	
GND	Power Ground	-0.3	0.3	V	

6.2 DC 特性

Condition：T_{OP} = -10℃ to 70℃，GND=0V

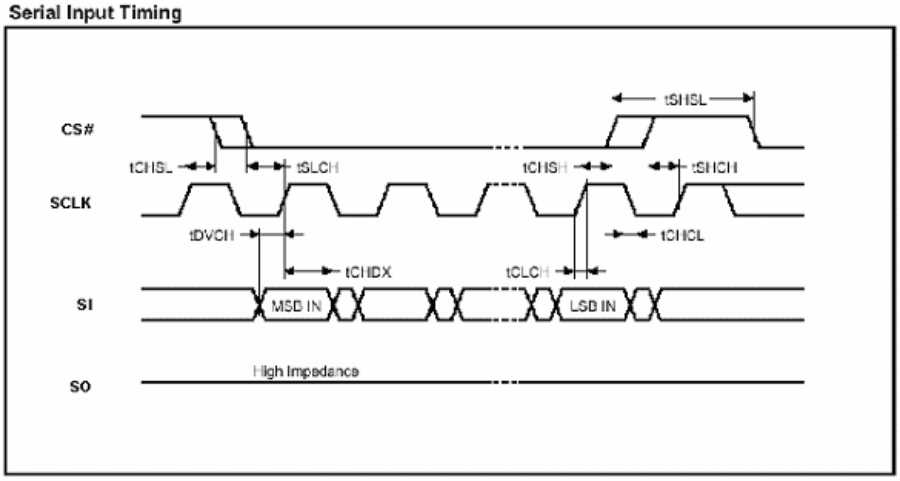
Symbol	Parameter	Min.	Max.	Unit	Condition
I_{DD}	VCC Supply Current(active)		10	mA	
I_{SB}	VCC Standby Current		10	uA	
V_{IL}	Input LOW Voltage	-0.3	0.3VCC	V	VCC=2.7-3.6V
V_{IH}	Input HIGH Voltage	0.7VCC	VCC+0.4	V	
V_{OL}	Output LOW Voltage		0.4 (I_{OL}=1.6mA)	V	
V_{OH}	Output HIGH Voltage	0.8VCC (I_{OH}=-100uA)		V	
I_{LI}	Input Leakage Current	0	2	uA	
I_{LO}	Output Leakage Current	0	2	uA	

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

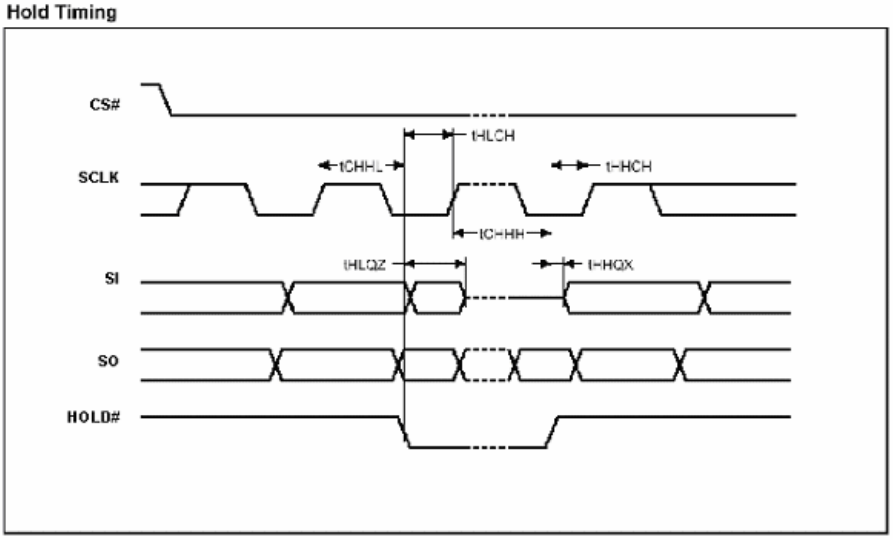
6.3 AC 特性

Symbol	Alt.	Parameter	**Min.**	Max.	Unit
Fc	Fc	**Clock Frequency**	D.C.	40	MHz
tCH	tCLH	Clock High Time	11		ns
tCL	tCLL	Clock Low Time	11		ns
tCLCH		Clock Rise Time(peak to peak)	0.1		V/ns
tCHCL		Clock Fall Time (peak to peak)	0.1		V/ns
tSLCH	tCSS	CS# Active Setup Time (relative to SCLK)	5		ns
tCHSL		CS# Not Active Hold Time (relative to SCLK)	5		ns
tDVCH	tDSU	Data In Setup Time	2		ns
tCHDX	tDH	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	tCSH	CS# Deselect Time	100		ns
t SHQZ	tDIS	Output Disable Time		9	ns
t CLQV	tV	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

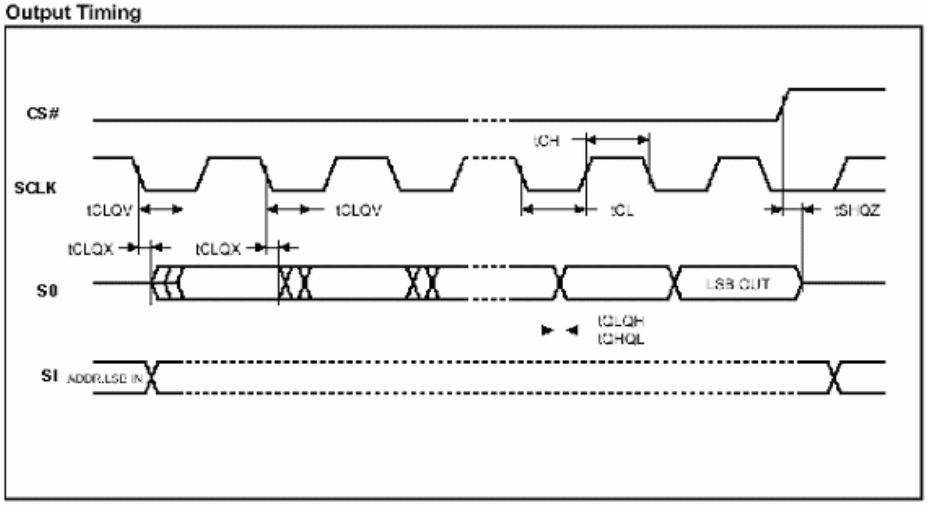
Serial Input Timing



Hold Timing

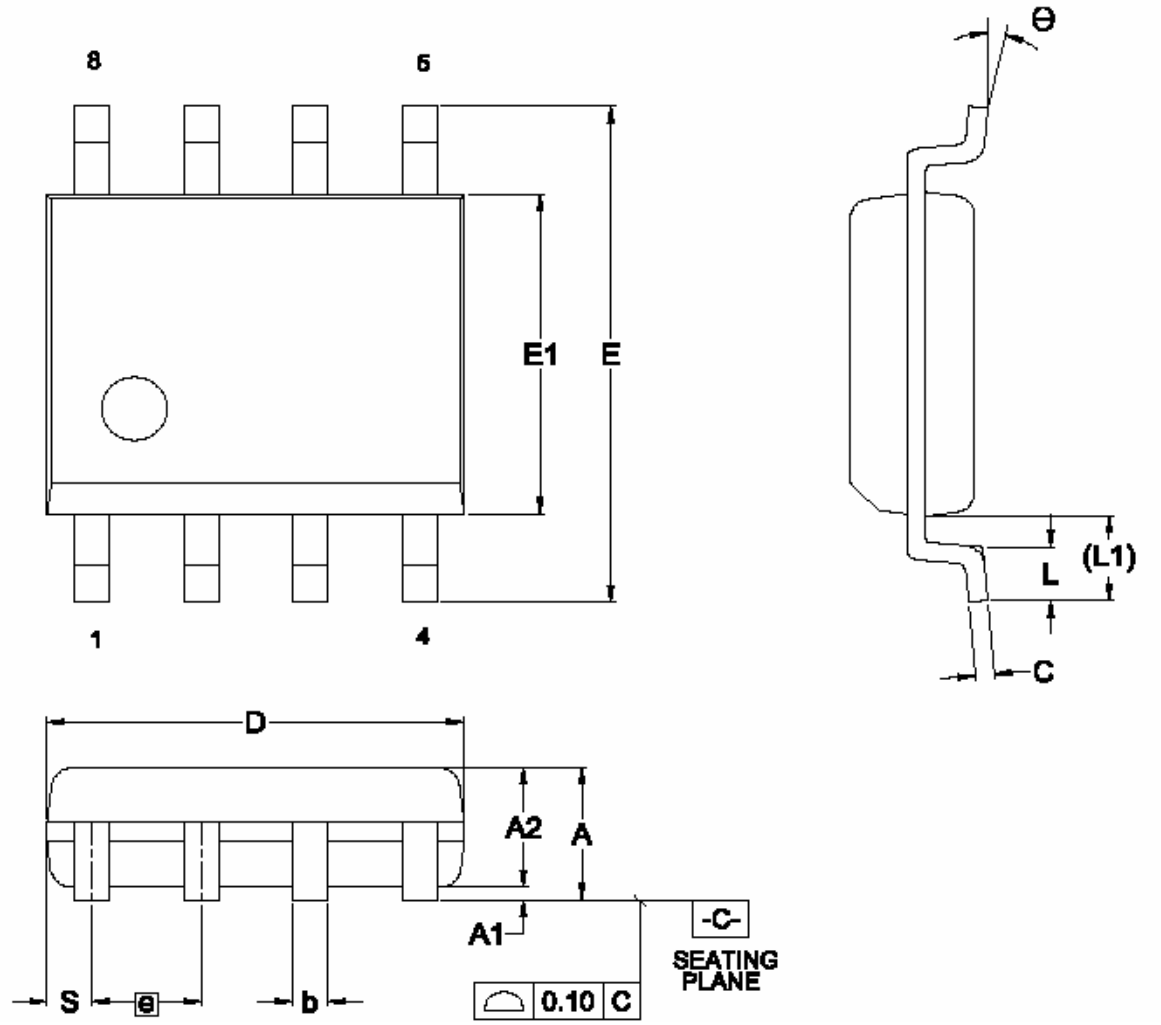


Output Timing



7 封装尺寸

SOP8 Package
Unit :mm



Dimensions (inch dimensions are derived from the original mm dimensions)

SYMBOL / UNIT		A	A1	A2	b	C	D	E	E1	e	L	L1	S	θ
mm	Min.	—	0.10	1.35	0.36	0.15	4.77	5.80	3.80		0.46	0.85	0.41	0
	Nom.	—	0.15	1.45	0.41	0.20	4.90	5.99	3.90	1.27	0.66	1.05	0.54	5
	Max.	1.75	0.20	1.55	0.51	0.25	5.03	6.20	4.00		0.86	1.25	0.67	8
Inch	Min.	—	0.004	0.053	0.014	0.008	0.188	0.228	0.150		0.018	0.033	0.016	0
	Nom.	—	0.006	0.057	0.016	0.008	0.193	0.236	0.154	0.050	0.026	0.041	0.021	5
	Max.	0.069	0.008	0.061	0.020	0.010	0.198	0.244	0.158		0.034	0.049	0.026	8

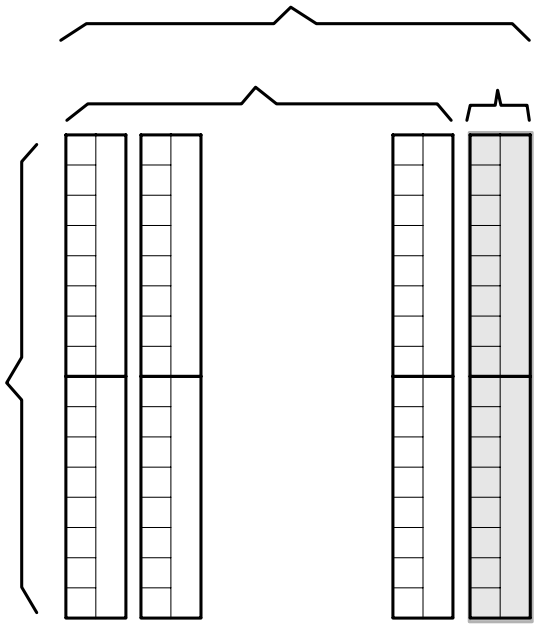
8 功能描述

8.1 汉字点阵排列格式

每个汉字在芯片中是以汉字点阵字模的形式存储的，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。

8.1.1 15X16 点汉字排列格式

15X16 点汉字的信息需要 32 个字节（BYTE 0 – BYTE 31）来表示。该 15X16 点汉字的点阵数据具体排列结构如下图：



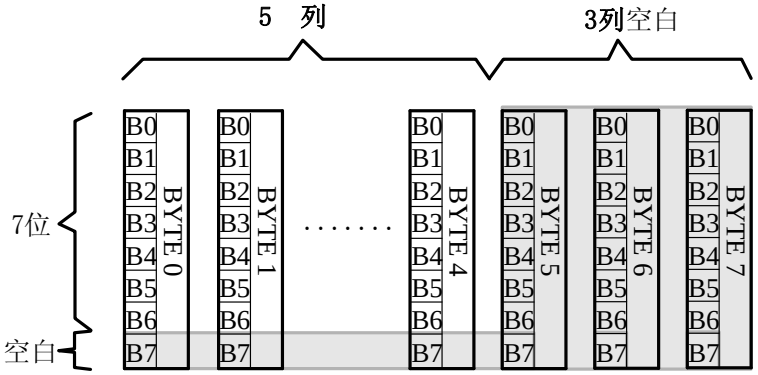
15X16 点汉字排列格式-竖置竖排图

点阵宽
（不等

B0 B0
B1 B1
B2 B2

8.1.2 5X7 点 ASCII 字符排列格式

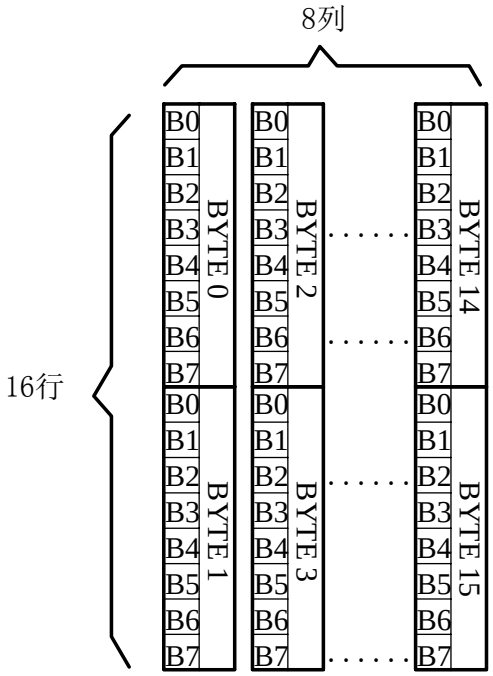
5X7 点 ASCII 的信息需要 8 个字节（BYTE 0 – BYTE7）来表示。该 ASCII 点阵数据是竖置竖排的，其具体排列结构如下图：



5X7 点 ASCII 字符排列格式图

8.1.3 8X16 点 ASCII 扩展字符排列格式

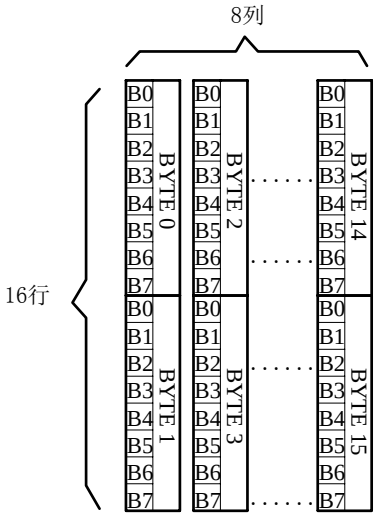
8X16 点 ASCII 的信息需要 16 个字节（BYTE 0 – BYTE15）来表示。



8X16 点 ASCII 字符排列格式图

8.1.4 Unicode 拉丁文、里尔文字符排列格式

8X16 点 ASCII 的信息需要 16 个字节（BYTE 0 – BYTE15）来表示。

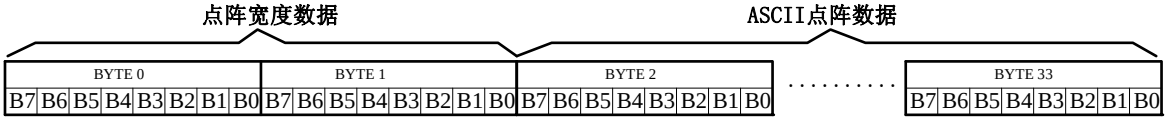


8.1.5 16 点阵 ASCII 方头字符排列格式

16 点阵 ASCII 方头字符的信息需要 34 个字节（BYTE 0 – BYTE33）来表示。

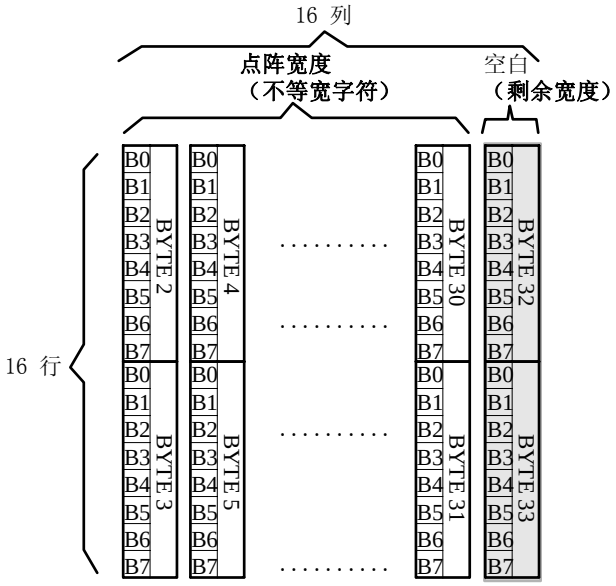
■ 存储格式

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，BYTE2-33 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。



16 点阵 ASCII 方头字符排列格式图

例如：ASCII 方头字符 B

0-33BYTE 的点阵数据是: 00 0C 00 00 F8 7F F8 7F 18 63 18 63 18 63 18 63 F8 67 F0 3E 00 1C 00 00 00 00 00 00 00 00 00 00

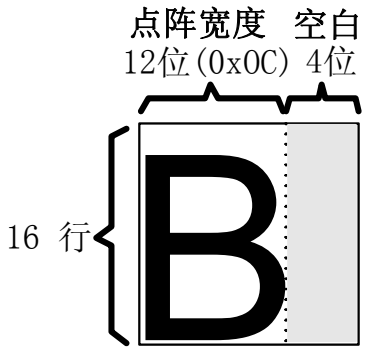
其中：

BYTE0~ BYTE1: 00 0C

为 ASCII 方头字符 B 的点阵宽度数据，即：12 位宽度。字符后面有 4 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）

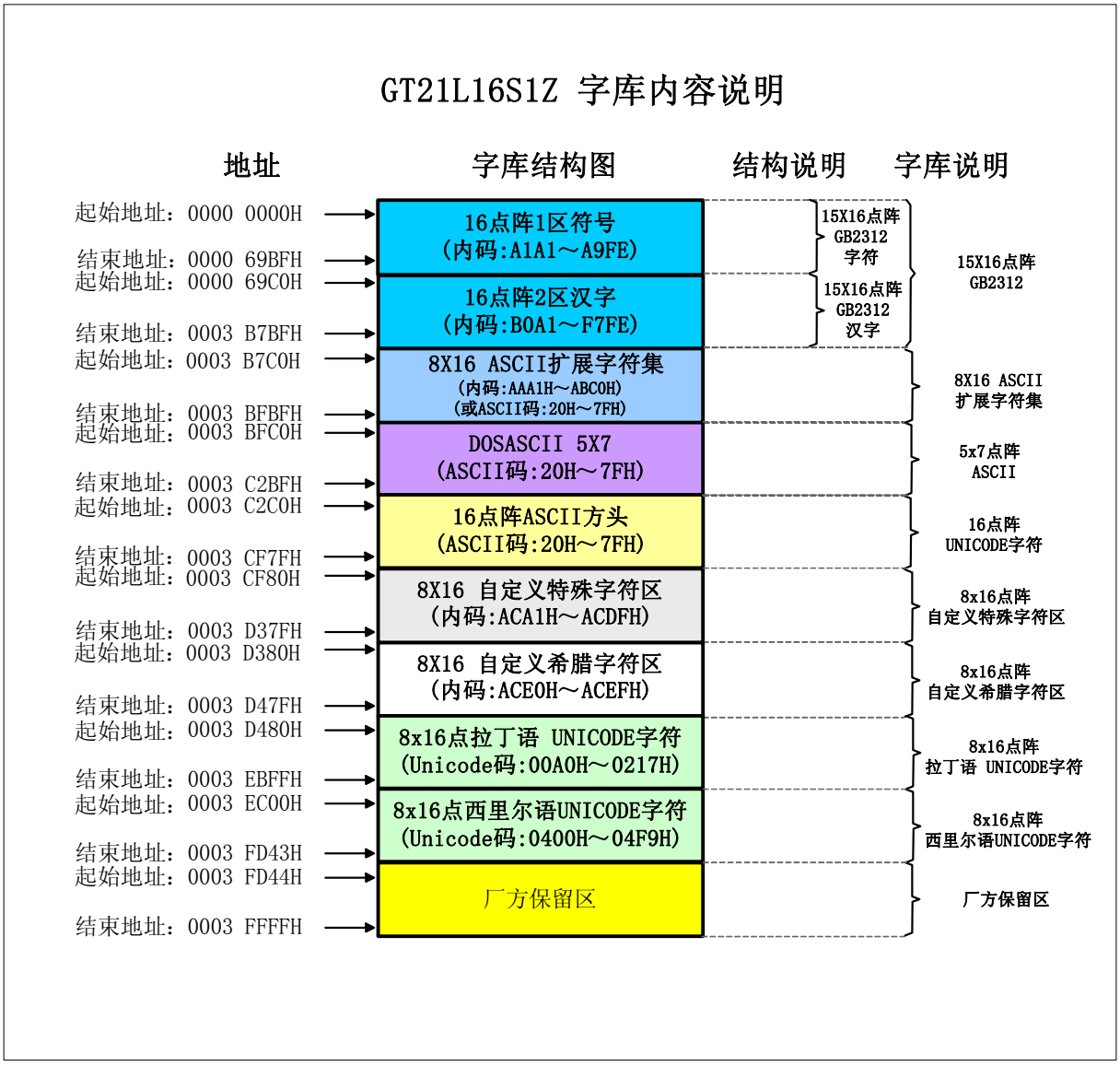
BYTE2-33: 00 00 F8 7F F8 7F 18 63 18 63 18 63 18 63 F8 67 F0 3E 00 1C 00 00 00 00 00 00 00 00 00 00

为 ASCII 方头字符 B 的点阵数据。



8.2 汉字点阵字库结构

- GB2312 标准点阵字库
  标准点阵字符 1 区对应 GB2312 码位的 A1A1~A9FE 共计 846 个字符；
  标准点阵字库中的 2 区对应 GB2312 码位 B0A1~F7FE 共计 6763 个汉字。
- 8X16 点 ASCII 扩展字符集合 对应内码 AAA1~ABC0 共计 126 个字符 (或者对应 ASCII 码位 20~7E 共计 96 个字符,其余忽略) 。
- 5X7 点 ASCII 字符对应 ASCII 码位的 20~7E 共计 96 个字符。
- ASCII 方头为不等宽 ASCII 字符所对应码位 20~7E 共计 96 个字符。
- 8X16 点自定义特殊字符区对应码位的 ACA1~ACDF 共计 64 个字符。
- 8X16 点自定义希腊字符区对应码位的 ACE0~ACEF 共计 16 个字符。
- 8X16 Unicode 字符为等宽字符，其中对应拉丁文系 Unicode 码 00A0H~04E9H 共计 376 个字符，对应西里尔文系 Unicode 码 0400H~04F9H 共计 250 个字符。



8.3 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。

15X16 点阵字库地址分配(字节地址)：0000 0000 H ~ 0003 B7BF H,

8.3.1 15X16 汉字点阵地址的计算

GBCode表示汉字内码。

MSB 表示汉字内码GBCode 的高8bits。

LSB 表示汉字内码GBCode 的低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。

15x16 汉字点阵在芯片中的字节地址的计算方法如下：

```
BaseAdd=0x0000 ;
if(MSB >=0xA1 && MSB <= 0xAB && LSB >=0xA1)
        Address =( (MSB - 0xA1) * 94 + (LSB - 0xA1))*32+ BaseAdd;
else if(MSB >=0xB0 && MSB <= 0xF7 && LSB >=0xA1)
   Address = ((MSB - 0xB0) * 94 + (LSB - 0xA1)+ 846)*32+ BaseAdd;
```

8.3.2 5X7 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。

5X7 ASCII 字符点阵在芯片中的字节地址的计算方法如下：

```
BaseAdd=0x 3BFC0
if (ASCIICode >= 0x 20) and (ASCIICode <= 0x 7E) then
        Address = (ASCIICode –0x 20 ) * 8+BaseAdd
```

8.3.3 8X16 ASCII 扩展字符地址的计算

1. 8X16 **ASCII 扩展字符使用传统的 ASCII 编码的计算方法如下：**

说明：

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。

计算方法：

```
BaseAdd=0x 3B7C0
if (ASCIICode >= 0x 20) and (ASCIICode <= 0x 7E) then
   if(ASCIICode >= 0x 60) then
          Address = (ASCIICode –0x 20+1 ) * 16+BaseAdd
   else
   Address = (ASCIICode –0x 20 ) * 16+BaseAdd
```

2. 8X16 **ASCII 扩展字符使用汉字内码传统的编码的计算方法如下：**

说明：

GBCODE 表示 GB 内码（16bits）
Address 表示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。

计算方法：

```
BaseAdd=0x 3B7C0
if (GBCODE > 0x AAA0) and (GBCODE <=0x ABC0 ) then
   Address = (GBCODE –0x AAA0 ) * 16+BaseAdd
```

8.3.4 16 点阵 ASCII 方头字符在芯片中的字节地址的计算方法如下：

```
BaseAdd=0x3C2C0
if (ASCIICode >= 0x 20) and (ASCIICode <= 0x 7E) then
        Address = (ASCIICode –0x 20 ) * 34+ BaseAdd
```

8.3.5 8X16 自定义特殊字符地址的计算

8X16 自定义特殊字符计算方法如下：

GBCODE 表示 GB 内码（16bits）

Address 表示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。

BaseAdd=0x 3CF80

if (GBCODE > 0x ACA0) and (GBCODE <=0x ACDF) then

Address = (GBCODE –0x ACA0) * 16+BaseAdd

8.3.6 8X16 自定义希腊字符区地址的计算

8X16 自定义希腊字符计算方法如下：

GBCODE 表示 GB 内码（16bits）

Address 表示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。

BaseAdd=0x 3D380

if (GBCODE > 0x ACE0) and (GBCODE <=0x ACEF) then

Address = (GBCODE –0x ACE0) * 16+BaseAdd

8.3.7 8X16 拉丁语、西里尔语 Unicode 字符在芯片中的字节地址的计算方法

计算方法如下：

unicode表示unicode内码。

Address 表示unicode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.字节地址的计算函数调用方法－拉丁文系

BaseAdd= 0x 3D480

if (Unicode > = 0x 00A0) and (unicode <=0x 0217) then

Address = (unicode –0x 00A0) * 16+BaseAdd

2.字节地址的计算函数调用方法－西里尔文系

BaseAdd= 0x 3EC00

if (Unicode > = 0x 0400) and (unicode <=0x 04F9) then

Address = (unicode –0x 0400) * 16+BaseAdd

9 附录

9.1 GB2312　1 区 (字符区)

GB2312 标准点阵字符 1 区和 5 区分别对应码位的 A1A1~A9EF 共计 846 个字符；

GB2312　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	｛	｜	｝	￣	

GB2312　1 区

A4	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		あ	あ	ぃ	い	ぅ	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	︿	﹀	｜	⋮	｜	⦚										

GB2312　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																

9.2 Unicode 字符区字符

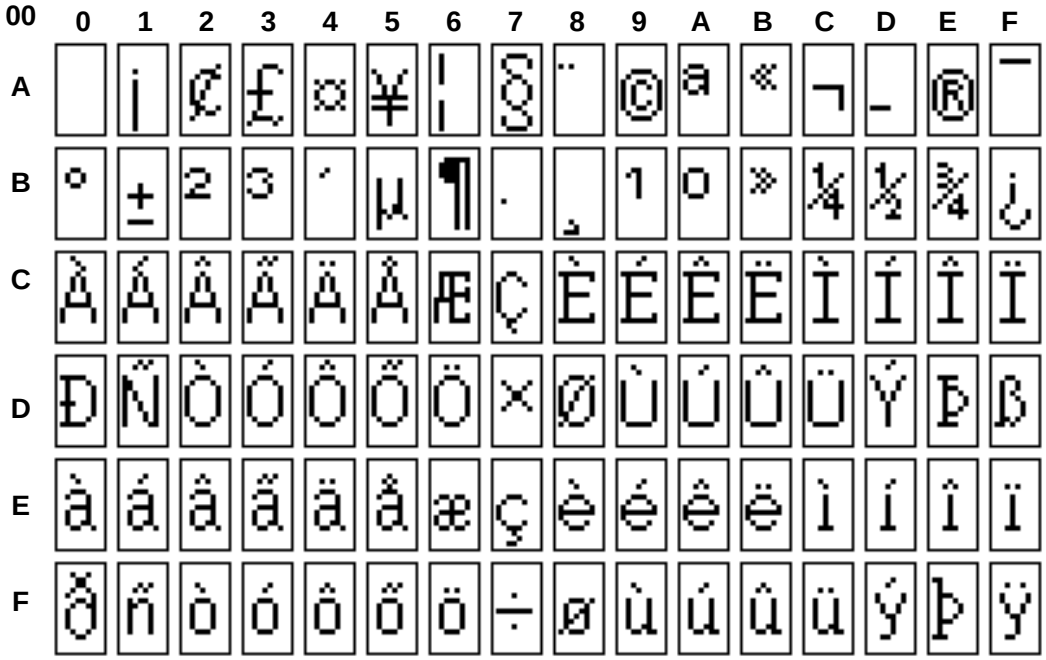
Unicode 字符区共收录　拉丁文系（LATIN）、希腊文系（GREEK）、西里尔文系（CYRILLIC）3 大文系，共计收录字符 456 个

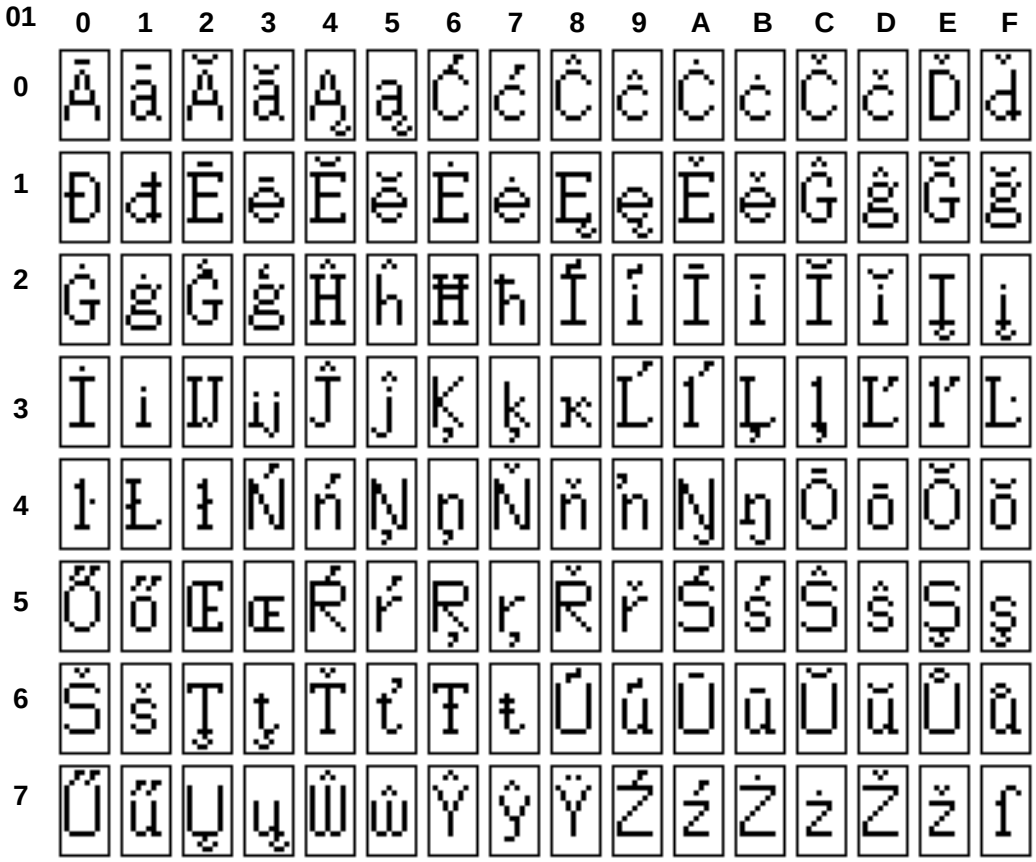
1）拉丁文系（8×16 点 LATIN）

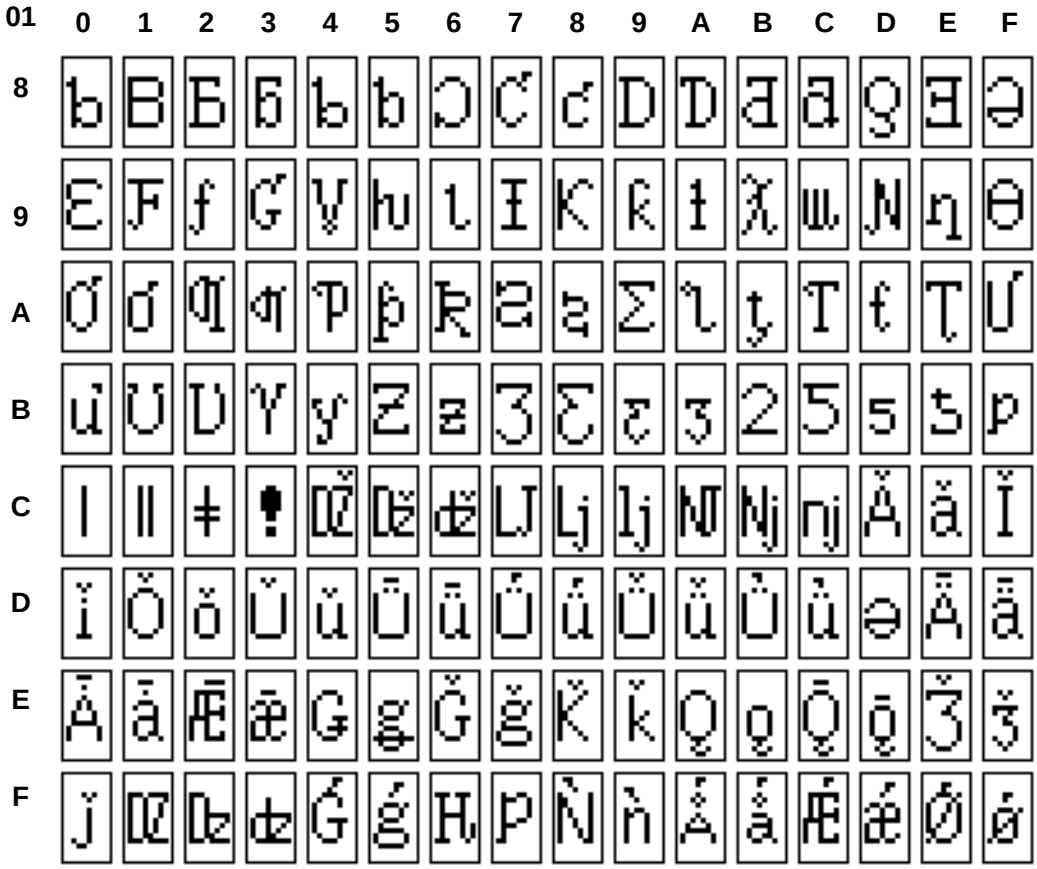
Unicode 字符区-拉丁文系的内码范围为 00A0~0217 共计收录 376 个字符。

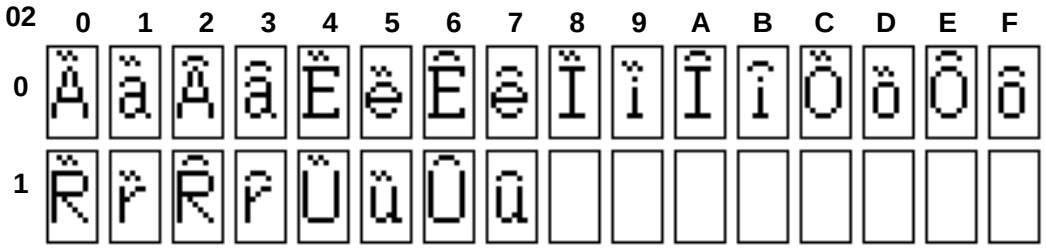
***说明**：由于拉丁文系（BASIC LATIN）的内码范围 0020~007E 与基本 ASCII 编码一致（可以共用），因此未列入 Unicode 字符区-拉丁文系中（见附件：10.2）。

Unicode 字符区-拉丁文系




01
0 1 2 3 4 5 6 7 8 9 A B C D E F
0
1
2
3
4
5
6
7

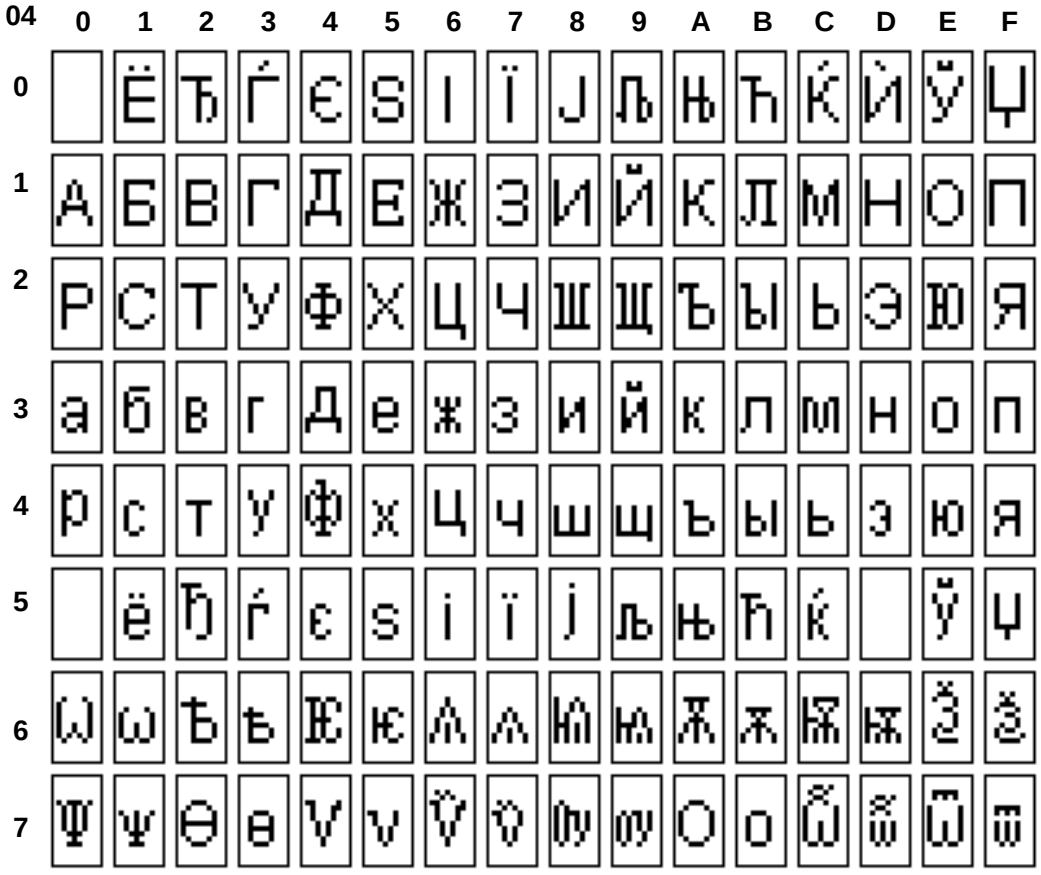

01
0 1 2 3 4 5 6 7 8 9 A B C D E F
8
9
A
B
C
D
E
F

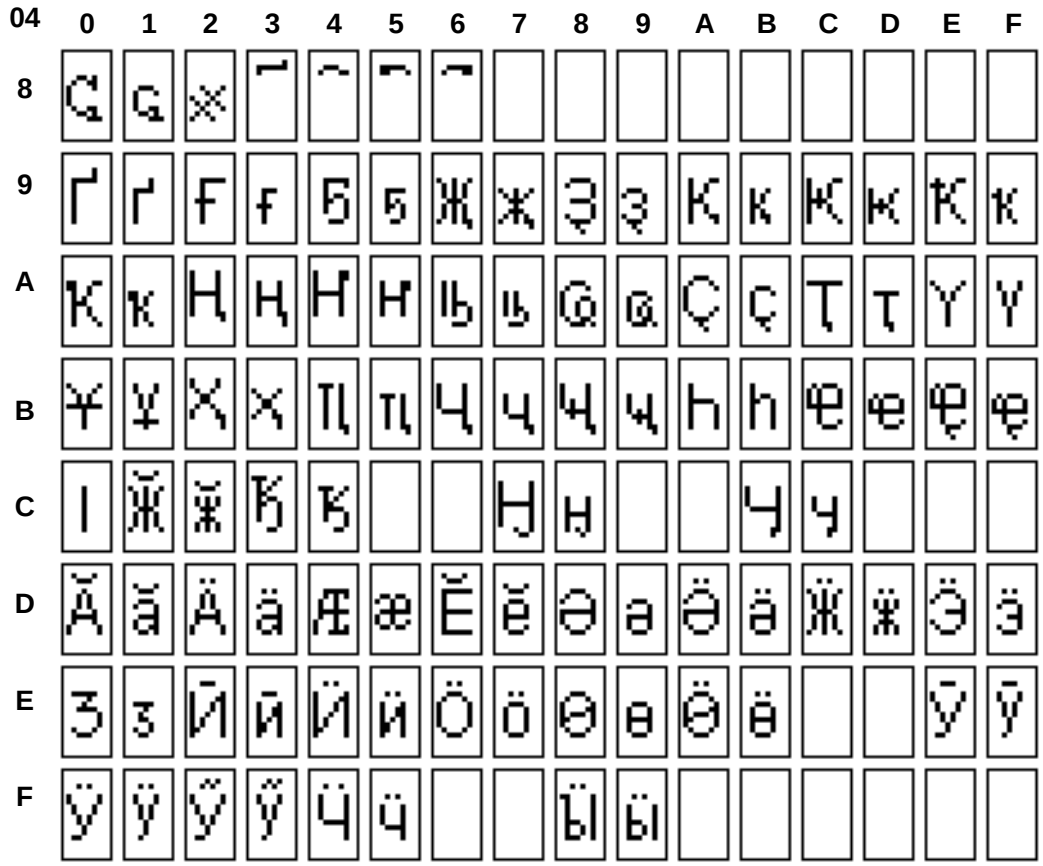


2） 西里尔文系（8×16 点 CYRILLIC）

Unicode 字符区-希腊文系的内码范围为 0400~04F9 共计收录 250 个字符。

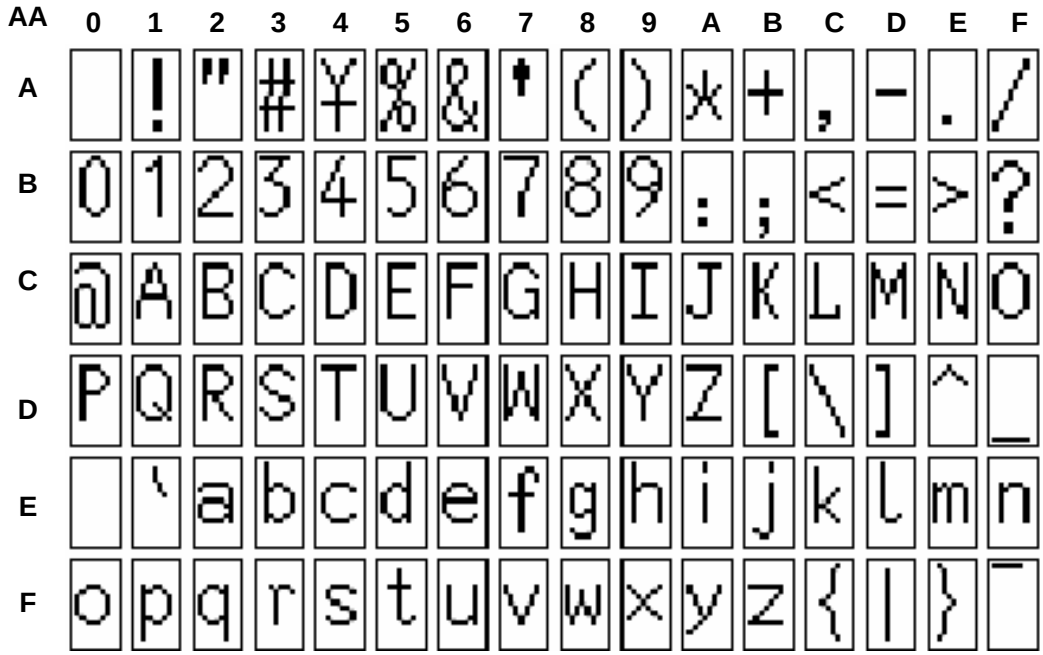
Unicode 字符区-西里尔文系

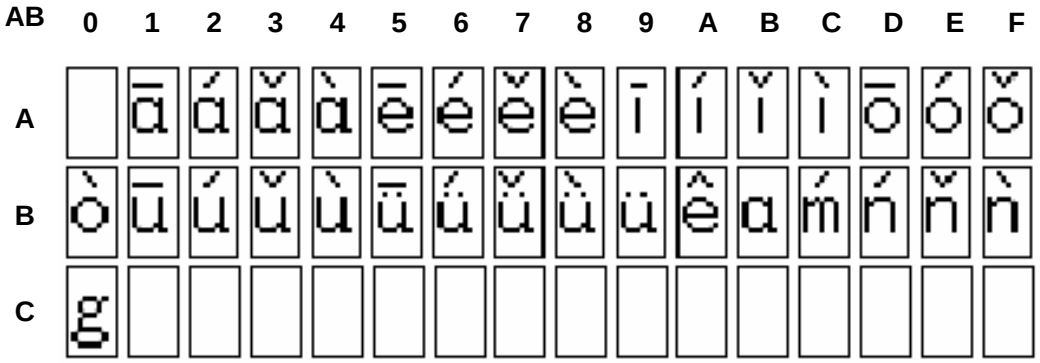




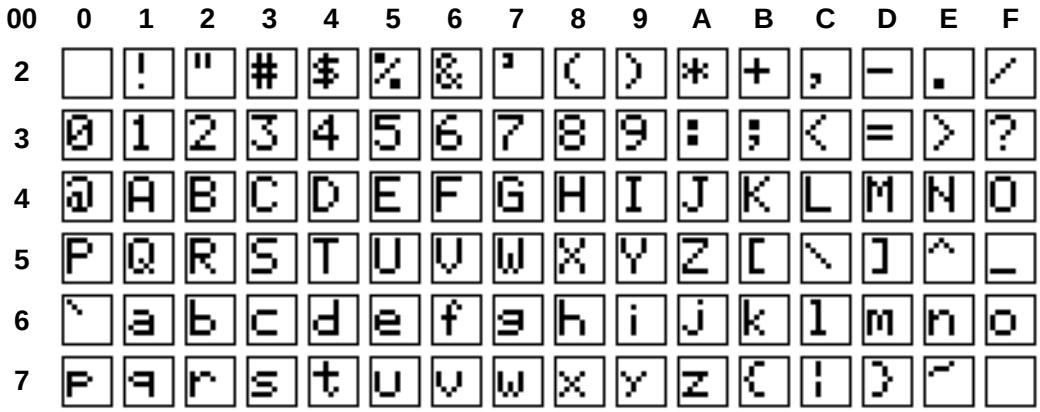
9.3 8×16 点扩展字符区 ASCII 字符（126 字符）

内码组成为 AAA1~ABC0 共计 126 个字符



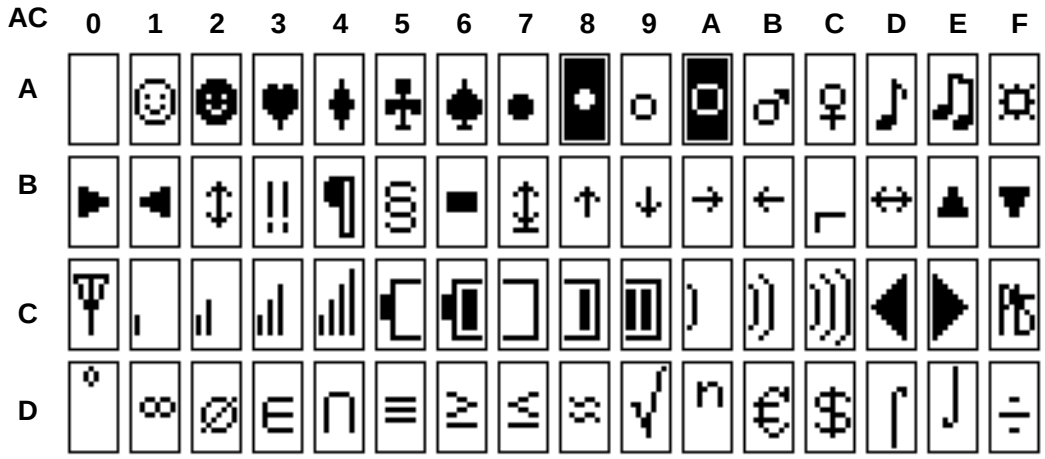


9.4 5×7 点 ASCII 字符（96 字符）



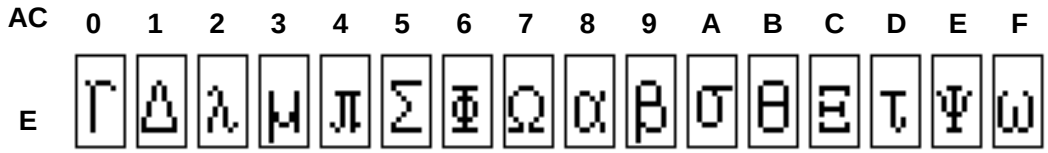
9.5 8×16 自定义特殊字符区 ASCII 字符（64 字符）

内码组成为 ACA1~ACDF 共计 64 个字符



9.6 8×16 自定义希腊字符区 ASCII 字符（16 字符）

内码组成为 ACE0~ACEF 共计 16 个字符



9.7 国家语言文系对照表

国家地区语言文系 Unicode 对照表

序号	国家或地区	语言	语言缩写	语言文系
1	中华人民共和国	中文中国	CHS	简体中文 GB18030
1A	中国香港特别行政区	中文香港特别行政区	ZHH	繁体中文 Big5
1B	中国澳门特别行政区	中文澳门特别行政区	ZHM	繁体中文 Big5
1C	中国台湾	中文台湾	CHT	繁体中文 Big5
2	新加坡	中文新加坡	ZHI	简体中文 GB18030
3	希腊	希腊语	ELL	希腊文系
4	西班牙	加泰隆语	CAT	拉丁文系
5	丹麦	丹麦语	DAN	拉丁文系
6	德国	德语德国	DEU	拉丁文系
7	美国	英语美国	ENU	拉丁文系
8	西班牙	西班牙语传统	ESP	拉丁文系
9	芬兰	芬兰语	FIN	拉丁文系
10	法国	法语法国	FRA	拉丁文系
11	冰岛	冰岛语	ISL	拉丁文系
12	意大利	意大利语意大利	ITA	拉丁文系
13	荷兰	荷兰语荷兰	NLD	拉丁文系
14	挪威	挪威语伯克梅尔	NOR	拉丁文系
15	巴西	葡萄牙语巴西	PTB	拉丁文系
16	瑞典	瑞典语	SVE	拉丁文系
17	印度尼西亚	印度尼西亚语	IND	拉丁文系
18	西班牙	巴士克语	EUQ	拉丁文系
19	南非	南非语	AFK	拉丁文系
20	法罗群岛	法罗语	FOS	拉丁文系
21	马来西亚	马来语马来西亚	MSL	拉丁文系
22	肯尼亚	斯瓦希里语	SWK	拉丁文系
23	西班牙	加里西亚语	GLC	拉丁文系
24	瑞士	德语瑞士	DES	拉丁文系
25	英国	英语英国	ENG	拉丁文系
26	墨西哥	西班牙语墨西哥	ESM	拉丁文系
27	比利时	法语比利时	FRB	拉丁文系
28	瑞士	意大利语瑞士	ITS	拉丁文系
29	比利时	荷兰语比利时	NLB	拉丁文系
30	挪威	挪威语尼诺斯克	NON	拉丁文系
31	葡萄牙	葡萄牙语葡萄牙	PTG	拉丁文系
32	芬兰	瑞典语芬兰	SVF	拉丁文系

33	文莱达鲁萨兰	马来语文莱达鲁萨兰	MSB	拉丁文系
34	奥地利	德语奥地利	DEA	拉丁文系
35	澳大利亚	英语澳大利亚	ENA	拉丁文系
36	西班牙	西班牙语国际	ESN	拉丁文系
37	加拿大	法语加拿大	FRC	拉丁文系
38	卢森堡	德语卢森堡	DEL	拉丁文系
39	加拿大	英语加拿大	ENC	拉丁文系
40	危地马拉	西班牙语危地马拉	ESG	拉丁文系
41	瑞士	法语瑞士	FRS	拉丁文系
42	列支敦士登	德语列支敦士登	DEC	拉丁文系
43	新西兰	英语新西兰	ENZ	拉丁文系
44	哥斯达黎加	西班牙语哥斯达黎加	ESC	拉丁文系
45	卢森堡	法语卢森堡	FRL	拉丁文系
46	爱尔兰	英语爱尔兰	ENI	拉丁文系
47	巴拿马	西班牙语巴拿马	ESA	拉丁文系
48	摩纳哥公国	法语摩纳哥	FRM	拉丁文系
49	南非	英语南非	ENS	拉丁文系
50	多米尼加共和国	西班牙语多米尼加共和国	ESD	拉丁文系
51	牙买加	英语牙买加	ENJ	拉丁文系
52	委内瑞拉	西班牙语委内瑞拉	ESV	拉丁文系
53	加勒比海	英语加勒比海	ENB	拉丁文系
54	哥伦比亚	西班牙语哥伦比亚	ESO	拉丁文系
55	伯利兹	英语伯利兹	ENL	拉丁文系
56	秘鲁	西班牙语秘鲁	ESR	拉丁文系
57	特立尼达和多巴哥	英语特立尼达	ENT	拉丁文系
58	阿根廷	西班牙语阿根廷	ESS	拉丁文系
59	津巴布韦	英语津巴布韦	ENW	拉丁文系
60	厄瓜多尔	西班牙语厄瓜多尔	ESF	拉丁文系
61	菲律宾共和国	英语菲律宾	ENP	拉丁文系
62	智利	西班牙语智利	ESL	拉丁文系
63	乌拉圭	西班牙语乌拉圭	ESY	拉丁文系
64	巴拉圭	西班牙语巴拉圭	ESZ	拉丁文系
65	玻利维亚	西班牙语玻利维亚	ESB	拉丁文系
66	萨尔瓦多	西班牙语萨尔瓦多	ESE	拉丁文系
67	洪都拉斯	西班牙语洪都拉斯	ESH	拉丁文系
68	尼加拉瓜	西班牙语尼加拉瓜	ESI	拉丁文系
69	波多黎各美	西班牙语波多黎各美	ESU	拉丁文系
70	保加利亚	保加利亚语	BGR	西里尔文系
71	俄罗斯	俄语	RUS	西里尔文系

72	乌克兰	乌克兰语	UKR	西里尔文系
73	比利时	比利时语	BEL	西里尔文系
74	前南斯拉夫马其顿共和国	马其顿语 FYROM	MKI	西里尔文系
75	吉尔吉斯坦	哈萨克语	KKZ	西里尔文系
76	吉尔吉斯斯坦	吉尔吉斯语 西里尔文系	KYR	西里尔文系
77	鞑靼斯坦	鞑靼语	TTT	西里尔文系
78	蒙古	蒙古语西里尔文系	MON	西里尔文系
79	阿塞拜疆	阿塞拜疆语西里尔文系	AZE	西里尔文系
80	乌兹别克斯坦	乌兹别克语西里尔文系	UZB	西里尔文系
81	塞尔维亚	塞尔维亚语西里尔文系	SRB	西里尔文系