

MCS-51 单片机的系统扩展技术（二）

3 数据存储器的扩展

MCS-51 单片机内部有 128 或 256 个字节的 data 存储器，这些存储器通常被用作工作寄存器、堆栈、临时变量等等，一般已经够用，但是如果系统要存储大量的数据，比如数据采集系统，那么片内的 data 存储器就不够用了，需要进行扩展。

3.1 常用的 data 存储器

单片机中常用的 data 存储器是静态 RAM 存储器（SRAM），图 7 是几种常用的 data 存储器的引脚图，以 62256 为例介绍，其中：

A0~A14：地址输入线；

D0~D7：数据线；

\overline{CE} ：选片信号输入线，低电平有效；

\overline{OE} ：读选通信号输入线，低电平有效；

\overline{WE} ：写选通信号输入线，低电平有效；

CE2：6264 芯片的高有效选通端；

VCC：工作电源，一般接+5V；

GND：工作地。

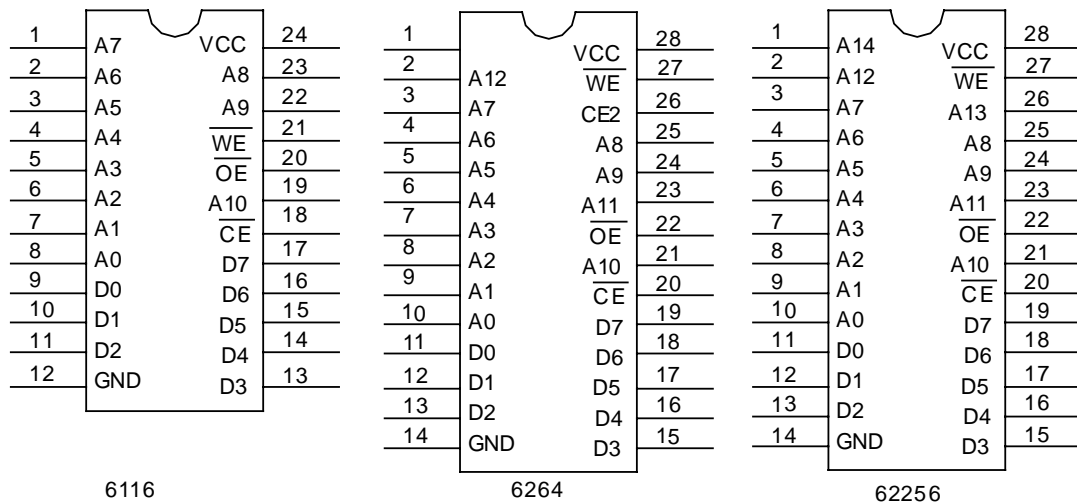


图 7 常用 RAM 引脚图

3.2 数据存储器扩展的方法

图 8 是 MCS-51 扩展 RAM 的示意图，从图中可以看出，MCS-51 单片机扩展 data 存储

器的方法基本同扩展程序存储器 EPROM 相似，仅是扩展 RAM 所用控制信号的 \overline{WE} 和 \overline{OE} 分别与 MCS-51 的 \overline{RD} 和 \overline{WR} 相连，而和 \overline{PSEN} 无关。关于数据存储器的片选信号将在下面详述。

3.3 地址空间的分配

由于 MCS-51 系列单片机的程序存储器和数据存储器截然分开，因此有两个各自的 64KB 的寻址空间，它们的地址都是从 0000H-FFFFH，这是由单片机通过不同的片选信号来确定的，用于 ROM 的是 \overline{PSEN} 信号，而用于 RAM 的则是 \overline{WR} 和 \overline{RD} 信号。其中 \overline{PSEN} 信号是每个读 ROM 的周期都会产生的，而 \overline{WR} 和 \overline{RD} 则是在使用 MOVX 类指令进行输入和输出时产生的。

由于芯片制造技术的提高，已不再使用若干片 ROM 来组合成所需容量的外部程序存储器了，而是直接选用所需容量的 ROM 芯片，这样外部 ROM 的地址空间分配不是一个问题。而外部 RAM 空间却是 RAM 和外部 I/O 口所共有的，其它的一些芯片如外接的定时器，串行口扩展芯片等等也有一些需要读或写的地址空间，所以就有一个如何分配这些空间的问题。

地址空间的分配，实际是 16 位地址线的具体安排与分配，是应用系统硬件设计中至关重要的一个问题。它与外部扩展的存储器容量及数量、功能接口芯片部件的数量等等有关，必须综合考虑，合理分配。在学习中尤其必须注意所谓地址分配的真实物理意义，这样才能灵活应用。

在外部扩展多片存储和功能部件接口芯片时，主机通过地址总线发出的地址是用来选择某一个存储单元或某一个功能部件接口芯片（或芯片中的某一个寄存器）的。要完成这一功能，必须进行两种选择：一是必须选择出指定的芯片（称之为片选）；二是必须选择出该芯片的某一个存储单元。第二种选择由地址总线来完成，现在讨论第一种选择，即选中该芯片。通常有两种片选的方法：线选法和译码法。

首先了解“选中芯片”的含义，从第一章的学习中得知，很多芯片都是“并联”的，即它们的数据总线、地址总线都是一一对应地连在一起的，比如某块芯片的 D0~D7 和其它的若干块芯片的 D0~D7 联在一起，然后再一起接到单片机的 P0.0~P0.7 上去。这必然带来一个问题：从单片机里送出的数据究竟由哪一块芯片来接收？或者如果单片机要接收某块芯片送出的数据，怎样保证其它芯片不在同一时间也送出数据？所以构成总线式连接的芯片一定要有一个“片选端”，以便单片机可以送出信号来控制确定选中某片芯片，而其它的芯片不被选中，所谓不被选中其实就是指这些芯片的 D0~D7 都处于“高阻”状态，不会对数据的传输产生影响。

一、线选法

线选法就是将多余的地址总线（即除去存储容量所占用的地址总线外）中的某一根地址线作为选择某一片存储或某一个功能部件接口芯片的片选信号线。一定会有一些这样的地址线，否则就不存在所谓的“选片”的问题了。每一块芯片均需占用一根地址线，这种方法适

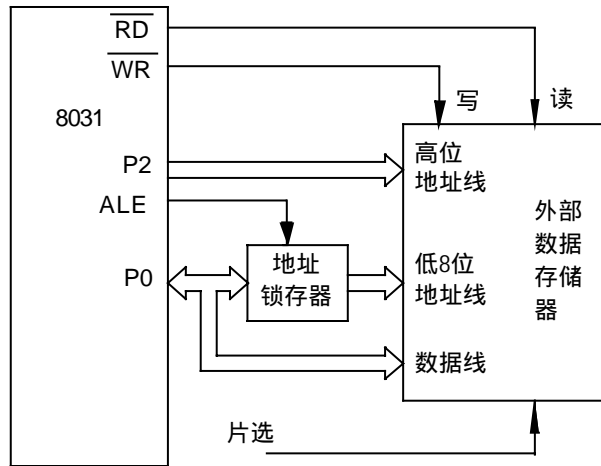


图8 MCS-51 扩展 RAM 的示意图

用于存储容量较小,外扩芯片较少的小系统,其优点是不需地址译码器,硬件节省,成本低。缺点是外扩器件的数量有限,而且地址空间是不连续的。

例如,单片机外扩二块 6264RAM,6264 只有 13 根地址线,还有三根是多余的,这三根线就可以用来作为片选使用,将其中的 A13 接第一片 6264 的 \overline{CE} ,而 A14 接第二片 6264 的 \overline{CE} ,可以分析一下两块芯片的地址。

由于芯片规定, \overline{CE} 端为 0 芯片就能被选中,所以第一片芯片的地址可以是

A15 A14 A13 A12.....A0

1 1 0 ×.....×

其中 X 是接于 6264 芯片的地址引脚上的值,可以是 0 也可以是 1,因此,这一块芯片的地址就是从 1100 0000 0000 0000~1101 1111 1111 1111,也就是 C000H~DFFFH 这一段空间,而第二块芯片的地址则可以是

A15 A14 A13 A12.....A0

1 0 1 ×.....×

即这一块芯片的地址可以是 1010 0000 0000 0000~1011 1111 1111 1111,也就是 A000H-BFFFH。

可是为什么第一块芯片的地址要是从 C000H-DFFFH 呢,难道不可以是这样吗:

A15 A14 A13 A12.....A0

0 1 0 ×.....×

当然可以,所以如果用 0100 0000 0000 0000~0101 1111 1111 1111 即 4000H-5FFFH 来访问第一块 6264 也是可以的,而第二块 6264 也可以用 2000H-3FFFH 来访问,也说是说一个芯片有多个地址与其相对应。

那么如果把第一块芯片的地址这样定:

A15 A14 A13 A12.....A0

1 0 0 ×.....×

是否可行呢?

如果只有一块芯片,这也是没有问题的,但这里有两块芯片,如果 A13 为 0 的同时 A14 也为 0,则必然会在选中第一块芯片时,也选中第二块芯片,这就会造成冲突,也就是说如果这样定地址的话,就人为造成了同时选中两块芯片的情况,这当然是不允许的,所以并不是单片机不让你用这组地址,而是你不会去这么用(除非你不想你的系统能正常工作)。

从上面的分析中可以看出,线选法会使得一块芯片拥有多组地址,这实际上意味着这些地址空间被浪费掉了,不能够使用了。以上面例子而言,只有 3 根(A13、A14、A15)地址线可以用于片选,所以只能扩展 3 片 6264 芯片,也就是只能够扩展到 3*8=24K 的 RAM 空间,采用这种芯片就没有办法扩展更多的 RAM 了。

二、译码法

由于线选法中一根高位地址线只能选通一个部件,每个部件占用了许多重复的地址空间,从而限制了外部扩展部件的数量。采用译码法的目的是减少各部件所占用的地址空间,以增加扩展部件的数量。译码法必须要采用译码芯片,下面先对译码芯片作一个介绍。

常用的译码芯片有 74LS138、74LS139 等。这里介绍 74LS138 的工作原理。图 9 是 74LS138 的引脚图。表 1 是 74LS138 的真值表。

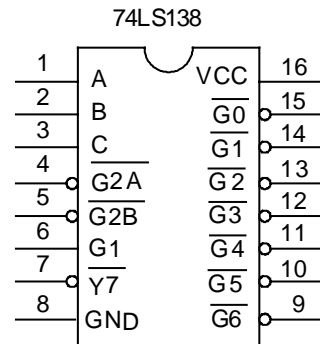


图 9 74LS138 引脚图

表 1 74LS138 译码器的真值表

译码器输入						译码器输出								
控制端			编码端			$\overline{Y0}$	$\overline{Y1}$	$\overline{Y2}$	$\overline{Y3}$	$\overline{Y4}$	$\overline{Y5}$	$\overline{Y6}$	$\overline{Y7}$	
G1	$\overline{G2A}$	$\overline{G2B}$	A	B	C									
1	0	0	0	0	0	0	1	1	1	1	1	1	1	
			0	0	1	1	0	1	1	1	1	1	1	
			0	1	0	1	1	0	1	1	1	1	1	
			0	1	1	1	1	1	1	0	1	1	1	
			1	0	0	1	1	1	1	1	0	1	1	
			1	0	1	1	1	1	1	1	1	0	1	
			1	1	0	1	1	1	1	1	1	1	0	1
			1	1	1	1	1	1	1	1	1	1	1	1
0	×	×				$\overline{Y0} \sim \overline{Y7}$ 均为 1								
×	1	×	×	×										
×	×	1												

对照表和图，可以看出 74LS138 一共有 6 个输入端，其中 G1, $\overline{G2A}$ 、 $\overline{G2B}$ 用于选通本片芯片 相当于是 74LS138 的“片选端”，如果要 74LS138 起作用，G1 必须接高电平，而 $\overline{G2A}$ 、 $\overline{G2B}$ 则必须接低电平，这三个引脚可以用作 74LS138 的级联，即在系统中有多个 74LS138 时的情况。另外的三个输入端是编码端 A、B、C，它们的状态决定了译码器的输出 $\overline{Y0} \sim \overline{Y7}$ 的状态，第一章时学过，三根线有 8 种状态 ($2^3=8$)，而输出也正好是 8 根线，因此这 8 根线的状态就与这三根线的 8 种状态相对应。注意，这里的关键是三根线可以是 0 和 1 的任意组合，而输出的 8 根线却是任意时刻只有一根线是 0，而其余都是 1。74LS138 被称之为“三一八译码器”。

在上面的例子中，系统扩展了两片 6264 芯片，共用去地址线 13 根，还有 3 根地址线没有用，如果用译码法，就可以把这三根地址线分别接到 74LS138 的译码端，如将 A15 接 A 端，A14 接 B 端，A13 接 C 端，然后用译码器的输出端接 6264 的片选端，例如用 Y0 接到第一片 6264，而用 Y5 接到第二片 6264。由于系统中只有一块 74LS138，所以把 G1 接+5V， $\overline{G2A}$ 、 $\overline{G2B}$ 则接地以选中该块芯片。下面分析一下这样这法这两片 6264 的地址范围。

要选中第一片 6264，就是要求 Y0 为 0，查一下表，当 Y0 是 0 时，要求 A、B、C 均为 0，因此，可以写出第一片 6264 的地址范围是：

A B C
A15 A14 A13 A12.....A0
0 0 0 ×.....×

即从 0000 0000 0000 0000B~0001 1111 1111 1111B，也就是 0000H~1FFFFH。第二片 6264 的片选端是接的 Y5，逆查表格可以看到，要 Y5 等于 0，就是要求 A=1、B=0、C=1，同样可以写出第二片的地址范围是：

A B C
A15 A14 A13 A12.....A0
1 0 1 ×.....×

即第二片的地址范围是 1010 0000 0000 0000~1011 1111 1111 1111，也就是 A000H~BFFFH。

由于一块 74LS138 有 8 个输出端，这里只用了两个，所以还可以再接 6 个，因此，采用这种方法可以扩展的地址空间是 $8K \times 8 = 64K$ ，即能够利用全部的 64K 空间。

如果所用的芯片是 2K 的，如 6116，那么就只需要用到 11 根地址线，也就是有 5 根高

位的地址线是空余的,这时再用 74LS138 进行扩展也会出现地址重叠的现象,比如,将 A13, A12 和 A11 接到一片 74LS138 的 A, B, C 三个输入端,那么第一个输出 Y0 对应的地址就可以是:

	A	B	C		
A15	A14	A13	A12	A11	A10.....A0
1	1	0	0	0	×.....×

即地址范围是 1100 0000 0000 0000~1100 0111 1111 1111,也就是 C000H~C7FFH。

很容易想到,并非一定要规定 A15 A14 是 1,也可以是 0,所以实际上 Y0 对应的有四段地址,分别是 0000 0000 0000 0000~0000 0111 1111 1111 (0000H~07FFH) 0100 0000 0000 0000~0100 0111 1111 1111 (4000H~47FFH) 8000H~87FFH 和 C000H~C7FFH。

同样,其余的每一个输出端也会对应 4 段地址,事实上,由于一片 74LS138 只有 8 个输出端,而每一个输出端接的是一片 2K 的芯片,所以用一片 74LS138 芯片只能译出 2K×8 即 16K 的地址。

在用一片 74LS138 不能译出所有的地址的情况下,可以使用多块 74LS138 进行译码,利用该芯片的片选端进行控制,例如在这个系统中可以扩展两块 74LS138 芯片,把 A15 (P2.7) 接第一块 74LS138 的 G1 端,把 A14 (P2.6) 接第二块 74LS138 的 G1 端,这样一共有 16 个输出端 (8×2), 作为练习,可以自己写出这 16 个输出端所对应的地址段。

需要说明的是,如果实际的系统中并不需要扩展很多芯片,那么就没有必要把所有的码都译出来——重叠就重叠吧。

图 10 是一个系统扩展 RAM 芯片 6264 的图,如果读者需要扩展 ROM 或 RAM,可以根据这个图来进行。

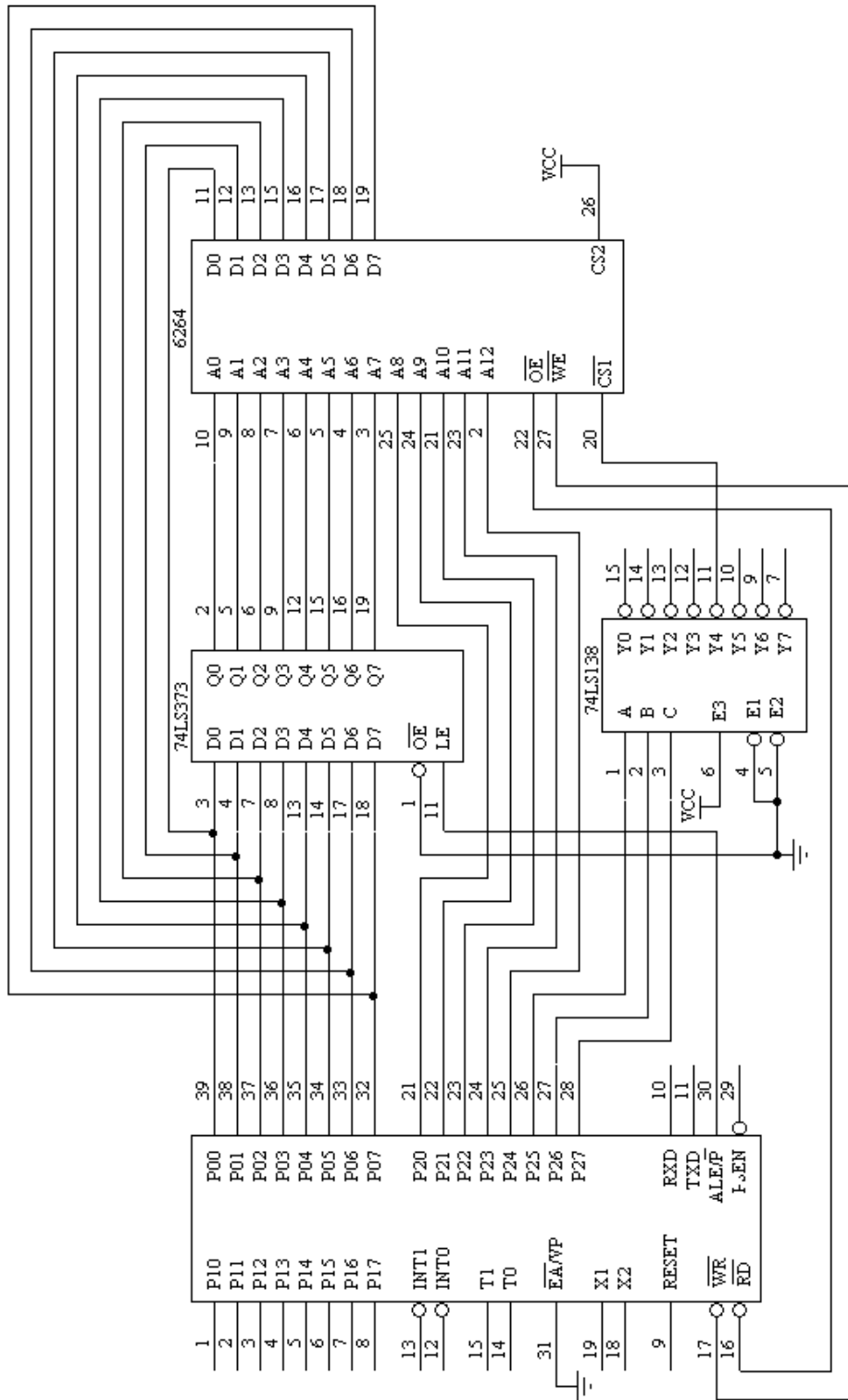


图 10 MCS-51 单片机扩展 6264 芯片