

用 EPROM 设计数字电路

华北石油管理局钻井工艺研究所 廖善榕

EPROM的最主要用途是用作计算机的程序存储器。除此之外,它也可以代替普通的门电路,用来设计数字电路。

用 EPROM 设计数字电路,与可编程逻辑阵列 PAL、GAL 有异曲同工之妙。使用 PAL、GAL,必须有专门的开发工具,而且从目前来看,这种开发工具价格较高,对于用量较小的单位不太适合;采用 EPROM,只需用一般的 EPROM 编程器,容易实现,修改起来也方便。

EPROM 种类很多,如 2708、2716、2732、2764、27128、27256 等。它们除了容量不同外,工作方式都是相同的。下面仅以最常用的 2716 为例,说明用 EPROM 设计数字电路的一般方法。

2716 的管脚图如图 1 所示。

$A_9 \sim A_0$	地址线
$D_7 \sim D_0$	数据线
$\overline{CE}/PGM$	片选/编程信号,读时应接地
$\overline{OE}$	输出使能信号,读时应为低电平
V_{pp}	编程电压,读时应为 +5V
V_{cc}	+5V
GND	地

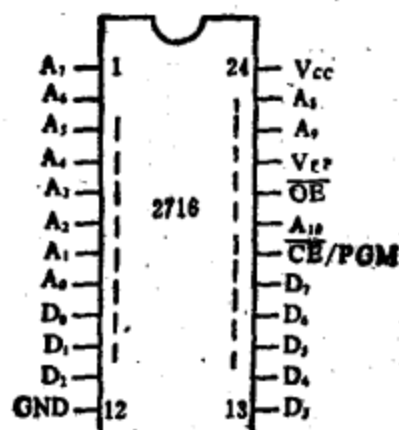


图1 2716管脚图

数字电路分为两类,即组合电路和时序电路。组合电路在任何时刻产生的稳定输出信号仅取决于此时刻电路的输入信号。而时序电路在某一时刻产生的稳定输出信号,则不仅取决于此时刻电路的输入信号,而且也取决于电路过去的输入信号。正因为如此,用 EPROM 设计

组合电路和时序电路的方法有所不同。

一、用 EPROM 设计组合电路

组合电路可以用式(1)表达。

$$O_n = F(I_n) \quad (1)$$

式中, I_n 是 n 时刻输入变量集合, $I_n = \{I_{1n}, I_{2n}, \dots, I_{in}\}$ 。 I_{in} 为第 i 个输入变量在第 n 时刻的状态。 O_n 是 n 时刻输出变量集合,各输出变量都是输入变量的函数。

在用 EPROM 设计组合电路时,只要以输入变量 I_i 作为地址信号,用数据线的输出作为各输出变量,并根据 EPROM 在读工作方式下的要求连接好 $\overline{CE}/PGM$, $\overline{OE}$, V_{pp} 等,然后根据组合逻辑的函数关系列出真值表,把该真值表写入 EPROM 中,这样,用一片 2716 就可以实现有 11 个输入变量 8 个输出变量的组合电路网络。这对于设计码制变换、标度变换等电路具有特别的优越性。

这里,我们用一片 2716 设计一个把 BCD 码转换成 Hamming 码的电路。BCD 码与 Hamming 码的对应关系如表 1 所示⁽¹⁾。

表1 BCD码与Hamming码的对应关系

BCD码				Hamming码							
A_3	A_2	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	1	1	1
0	0	1	0	0	0	0	1	1	0	0	1
0	0	1	1	0	0	0	1	1	1	1	0
0	1	0	0	0	0	1	0	1	0	1	0
0	1	0	1	0	0	1	0	1	1	0	1
0	1	1	0	0	0	1	1	0	0	1	1
0	1	1	1	0	0	1	1	0	1	0	0
1	0	0	0	0	1	0	0	1	0	1	1
1	0	0	1	0	1	0	0	1	1	0	0
1	0	1	0	1	x	x	x	x	x	x	x
1	0	1	1	1	x	x	x	x	x	x	x
1	1	0	0	1	x	x	x	x	x	x	x
1	1	0	1	1	x	x	x	x	x	x	x
1	1	1	0	1	x	x	x	x	x	x	x
1	1	1	1	1	x	x	x	x	x	x	x

根据表 1, $A_3A_2A_1A_0$ 作为 BCD 码输入, $A_{10} \sim A_4$ 未用,可接地。 $D_7D_6D_5D_4D_3D_2D_1D_0$ 作为 Hamming 码输出。由于 BCD 码只有十种状态,而 $A_3 \sim A_0$ 可以出现十六种状态,因此可以用 D_7 来指示输入代码是否正确。输入正确 BCD 码时, $D_7 = 0$, $D_6 \sim D_0$ 为相应的 Hamming 码;输入错误代码时, $D_7 = 1$,表明输入出错, $D_6 \sim D_0$ 无效。在 $A_3A_2A_1A_0 = 1010 \sim 1111$ 时, $D_6 \sim D_0$ 为无关态,可以设定为任意代码,如 0000000。这样,该 2716 就形成了一个 BCD-Hamming 码变换器。

二、用 EPROM 设计时序电路

如前所述,时序电路的输出,不仅与此刻的输入有关,还和以往的输入有关。而上次的输出又是由上

次的输入和以往的输入所决定的。因此,也可以说,时序电路的输出取决于此时此刻的输入和上一时刻的输出状态,用式(2)来表达。

$$O_n = F(I_n, O_{n-1}) \quad (2)$$

式中, I_n 、 O_n 的含义同式(1), O_{n-1} 为上一时刻输出变量的集合,也可以是其子集。

在用 EPROM 设计时序电路时,仍以地址线作为输入,数据线作为输出, $\overline{OE}$, $\overline{CE}$, V_{PP} 连接方法与组合电路相同。所不同的是,由于本次输出还与上次输出有关,因此某些数据线必须通过外部连线与地址线相连。下面,举例说明。

为了检测某旋转体的旋转方向,在旋转体上找到一个凸出点或者粘上一个铁块,在旋转体外邻近该小铁块固定两个接近开关,这两个接近开关与小铁块的相对位置如图 2 所示。

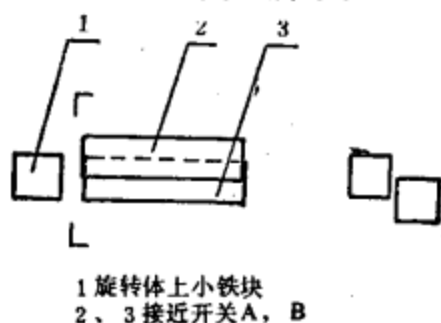


图2 接近开关安装示意图

在旋转体转动时,接近开关输出两组信号 A 和 B,其波形如图 3 所示。

显然,如果物体正转时,BA 的状态变化序列如图 4 中实线箭头所示,

而物体反转时,BA 状态变化序列如图 4 中虚线箭头所示。

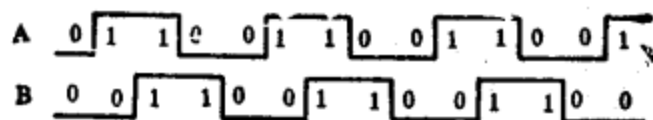


图3

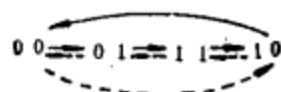


图4 BA 状态变化图

为此,我们用 2716 的 $D_0 = 1$ 表示物体正转, $D_0 = 0$ 表示物体反转,用 $D_1 D_2$ 记忆本次 A、B 状态, A_0 、 A_1 分别与接近开关输出 A、B 相连,并且把 D_1 、

D_2 与 A_2 、 A_3 相连, D_0 与 A_4 相连,如图 5。据此,在 2716 中写入如表 2 所示编码。

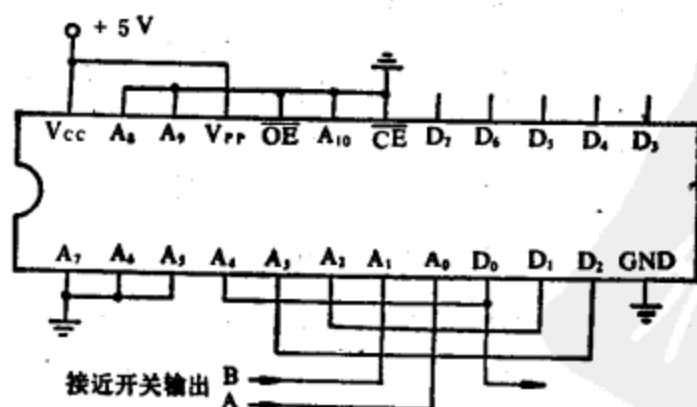


图5 运动方向检测电路图

表 2

$A_{10} \sim A_5$	A_4	A_3	A_2	A_1	A_0	$D_7 \sim D_3$	D_2	D_1	D_0
000000	0	0	0	0	0	×××××	0	0	0
000001	0	0	0	0	1	×××××	0	1	1
000010	0	0	0	1	0	×××××	1	0	0
000011	0	0	0	1	1	×××××	1	1	×
000100	0	0	1	0	0	×××××	0	0	0
000101	0	0	1	0	1	×××××	0	1	0
000110	0	0	1	1	0	×××××	1	0	×
000111	0	0	1	1	1	×××××	1	1	1
001000	0	1	0	0	0	×××××	0	0	1
001001	0	1	0	0	1	×××××	0	1	×
001010	0	1	0	1	0	×××××	1	0	0
001011	0	1	0	1	1	×××××	1	1	0
001100	0	1	1	0	0	×××××	0	0	×
001101	0	1	1	0	1	×××××	0	1	0
001110	0	1	1	1	0	×××××	1	0	1
001111	0	1	1	1	1	×××××	1	1	0
010000	1	0	0	0	0	×××××	0	0	1
010001	1	0	0	0	1	×××××	0	1	1
010010	1	0	0	1	0	×××××	1	0	0
010011	1	0	0	1	1	×××××	1	1	×
010100	1	0	1	0	0	×××××	0	0	0
010101	1	0	1	0	1	×××××	0	1	1
010110	1	0	1	1	0	×××××	1	0	×
010111	1	0	1	1	1	×××××	1	1	1
011000	1	1	0	0	0	×××××	0	0	0
011001	1	1	0	0	1	×××××	0	1	×
011010	1	1	0	1	0	×××××	1	0	1
011011	1	1	0	1	1	×××××	1	1	0
011100	1	1	1	0	0	×××××	0	0	×
011101	1	1	1	0	1	×××××	0	1	0
011110	1	1	1	1	0	×××××	1	0	1
011111	1	1	1	1	1	×××××	1	1	1

表中 $A_{10} \sim A_5$ 因未用,设为全 0。表中 $\times$ 为无关项,其中 $D_7 \sim D_3$ 不用, $D_2 \sim D_0$ 中的 $\times$ 是实际使用中不可能出现的情况,如 $A_4 \sim A_0 = 00011$,其含义是上次 BA 状态为 00,本次状态为 11,从波形图中可知这是不可能的。因此 $D_2 \sim D_0$ 中的 $\times$ 可以置为 0 也可置为 1。

在图 5 中 $\overline{CE}$ 接地,2716 一直处于工作状态, $D_7 \sim D_0$ 直接反映了 $A_{10} \sim A_0$ 的变化, D_0 输出的是一个电平信号。但在某种情况下,后续电路希望得到脉冲信号。这时可把 $\overline{CE}$ 接一脉冲信号(时钟)。当时钟信号在高电平时,2716 处于备用状态, $D_7 \sim D_0$ 呈高阻状态,若将它们接一上拉电阻,则呈高电平,而当时钟信号为低电平时,如果此时 $D_i = 0$ ($i \in \{0 \sim 7\}$),则在一个时钟周期中该脚输出了一个负脉冲。利用这个特点,

可以设计出脉冲输出的数字电路。

三、用EPROM设计数字电路时 应该注意的几个问题

1. 响应速度

在高速工作的数字电路中, 必须注意各部分电路之间的时序配合问题, 否则电路或者不能满足要求, 或者是无法正常工作, 出现险象。在使用EPROM时, 应查阅相应厂家的器件手册, 选择适用的EPROM。不同EPROM的速度差别较大。表3列出了Intel公司生产的几种EPROM的最大读出时间, 可供参考。

表3⁽²⁾

型 号	最大读出时间 (Max. t_{acc})
2716, 2732A-4, 2764-4 27128-4, 27256-4	450 ns
2764-1	180 ns
2732A-2, 2764-2	200 ns
2732A, 2764A, 27128, 27256	250 ns
2732A-3, 2764-3, 27128-3 27256-3	300 ns

2. 在电路设计中, 要注意无论 $\overline{OE}$, $\overline{CE}$ 或是地址信号中, 任一信号的改变, 都将引起输出的数据信号改变, 因此在读取数据信号时应保证读取瞬间以上信号的稳定。

3. 当 $\overline{CE} = 0$ 时, EPROM处于工作状态, 其功耗较大; 而 $\overline{CE} = 1$ 时, EPROM处于备用状态, 功耗较前者小得多。以2732A为例, 前者 V_{cc} 电流为100mA, 后者仅35mA。因此, 在用EPROM设计脉冲输出的数字电路时, 总是控制 $\overline{CE}$, 而 $\overline{OE}$ 则可接地, 既达到输出脉冲的目的, 又可大大减少功耗。

用EPROM设计数字电路, 可以大大减少芯片数量, 缩小电路板尺寸, 使得电路规整。用EPROM设计数字电路的方法与传统的数字电路设计方法有很大差别, 它不用进行繁琐的逻辑函数化简工作, 而是罗列出电路的各种可能的输入输出状态, 形成真值表, 把真值表写入EPROM。不但设计方便, 而且提高了设计的可靠性。此外, 即使需要改变设计, 也只要重新确定真值表, 改写EPROM即可。目前, 电擦除只读存储器E²PROM已经投入使用, 因此如能对电路进行巧妙设计, 甚至可以在线修改, 改变电路功能。

限于篇幅, 文中所举事例, 都是输入输出变量较少的电路, 其优越性似乎不大, 但作为一种数字电路设计方法实际上是很有用的, 它特别适合于设计多输入多输出的复杂的数字电路。

参 考 资 料

- (1) 西安交通大学 蒋大宗, 余秉钧, 金德庆: “数字逻辑”, 国防工业出版社, 1979。
- (2) Intel Corporation: Memory Components Handbook, 1983.

大连理工大学电子系319室为您提供 温度-数字转换微机接口板和装置

我室经多年研究和实验, 开发出全系列热电阻和热电偶STD总线、S-100总线温度-数字转换接口板和装置。集变送器、A/D转换器和数据处理软件为一体。输出数据是已经线性化了的温度的数字量。有普通接口板, 更有带单片机的智能接口板。放大器校正、通道转换、冷端补偿(热电偶)、引线电阻自动补偿(热电阻)、线性化、采集数据和数字滤波等均由单片机完成, 接口板自动工作。主机选通接口板, 用极短时间就可以把八个通道十六个字节的数据读入, 显著提高了主机效率(见本刊今年第3期“温度-数字转换微机接口电路”一文)。

用户遍及全国, 深受欢迎。备有资料, 函索即寄, 长期服务, 保证质量。

电话: 471511(总机)转电子系 电报挂号: 7108 邮政编码: 116024

开户银行: 大连工商银行沙办莱金分理处 帐号: 144367192

户名: 大连理工大学科技开发总公司

负责人: 大连理工大学电子系319室 邵军哲(副教授) 陈育斌(工程师)