Digital Circuits and Systems* Homework IV LaTeX

* Teacher: Song Zhou. TA: Yuan Xu

1st 张逸凯 171840708 (转专业到计科, 非重修)

Department of Computer Science and Technology

Nanjing University

zykhelloha@gmail.com

6.9

解:

当`IN == 0`, `IN`为低态输入时, 其中3个与非门从低态到高态，而另外3个与非门从高态到低态。(由`IN`输入和真值表列举可知)

$$t_p = 3 \times (t_{pLH} + t_{pHL}) = 3 * 15 + 3 * 15 = 90(ns)$$

同理, 当`IN == 1`, `IN`为高态输入时, $t_p = 90ns$

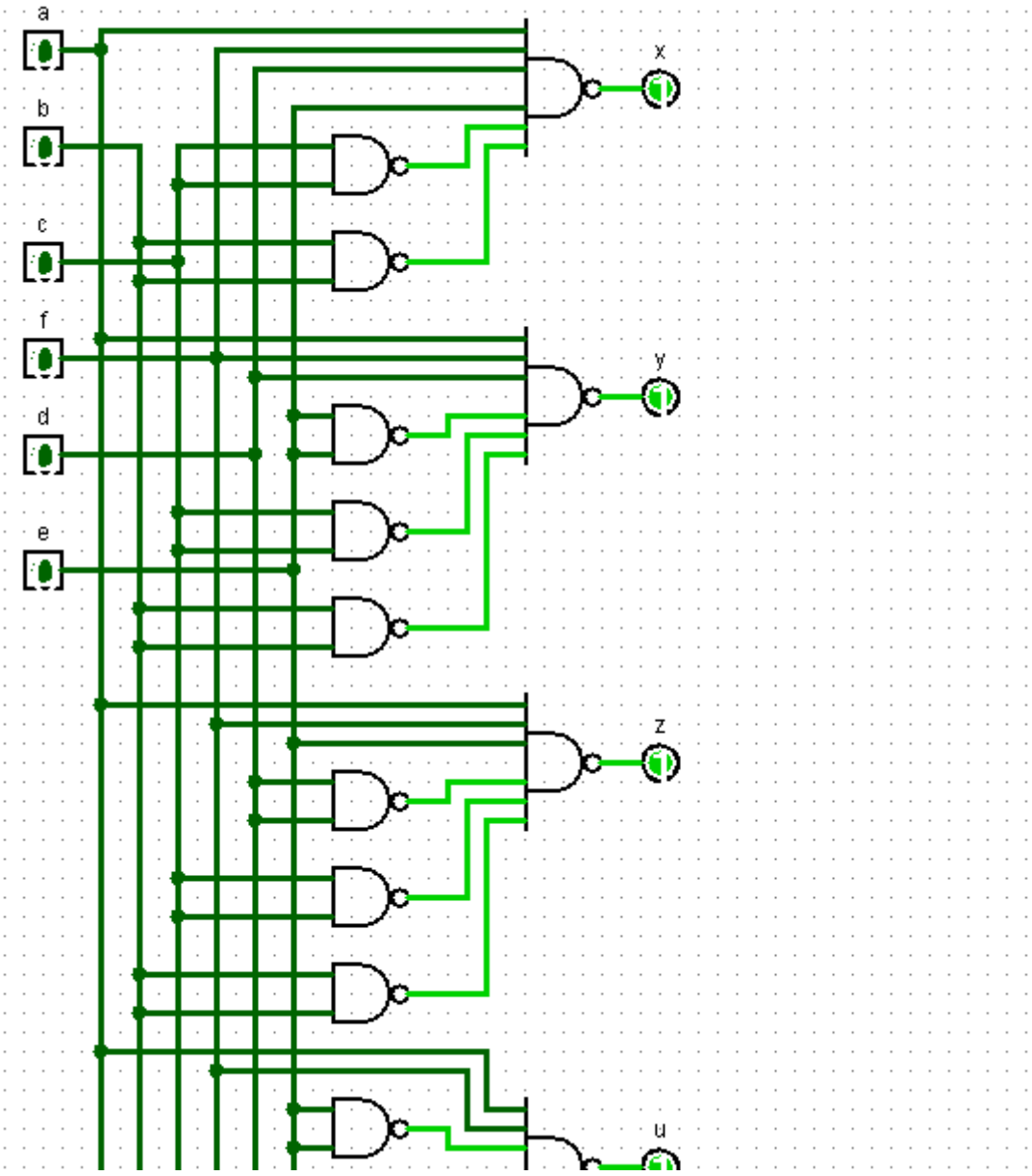
因为这里`74LS`的芯片传播延迟最大值t_{pLH}, t_{pHL}相等, 并且在电路图中低态到高态的与非门数量一样, 所以对单个与非门采用最坏情况的延迟量计算时对结果并没影响.

6.16

解:

低电平有效输出的译码器更快。因为这种译码器使用与非门, 而高电平有效输出的译码器使用与门，与门大多是与非门后加一个反相器实现的, 反相门快于非反相门，因此低电平有效输出的译码器更快。这正是为什么要用"圈到圈的设计".

下面是74x138的部分电路图, 在其中也可以容易发现为什么与非门更快.

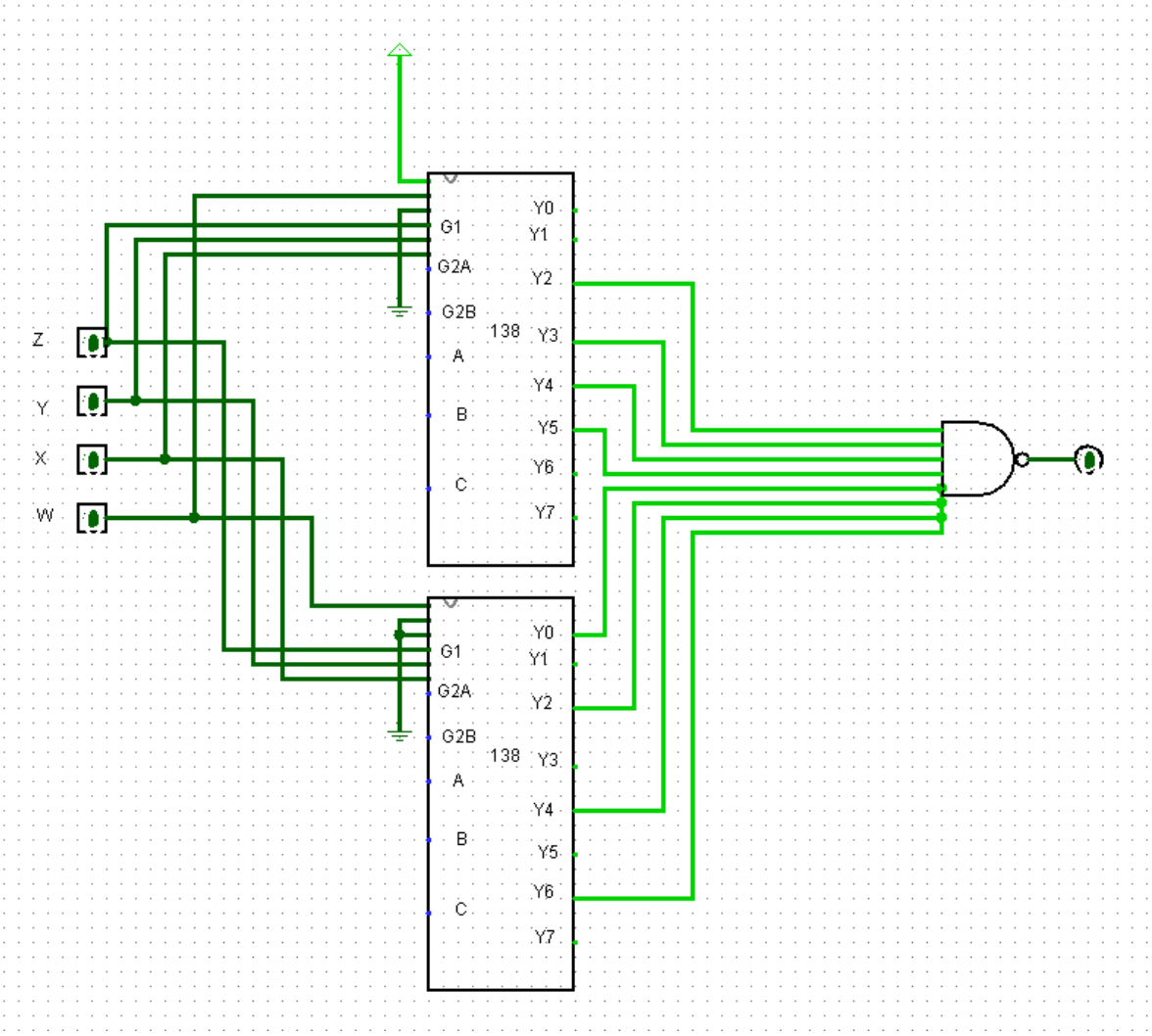


6.20

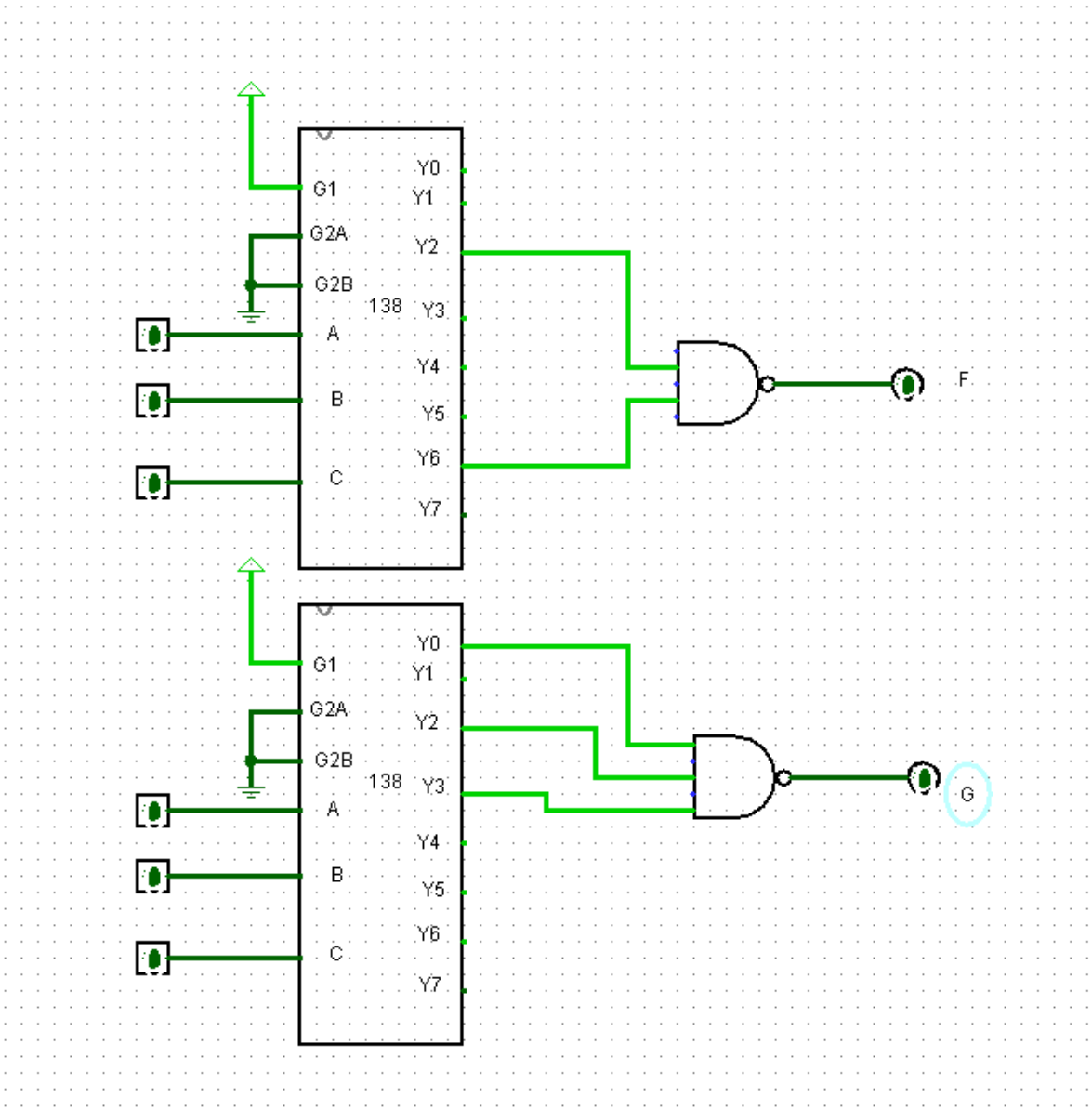
解:

(d)

这里不需要用卡诺图化简, 因为根据138译码器我们可以知道只要将输入Z作为某些使能端输入, 可以级联构成较大的译码器, 由此可以通过选择子选择相应的输出,


Z
Y
X
W
G1
G2A
G2B
A
B
C
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
138
G1
G2A
G2B
A
B
C
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
138

(f) 同理可得:



6.21

解: 错误： `74x139` 两个 `2-4` 译码器同时使能，2个3态门同时导通, 使能端会同时有效, 译码器接出后会导致存在2个三态门同时导通，可能出现输出逻辑电平冲突。

解决方法：将使能端分开，或其中一个接反相器, 反相连接各自使能。

6.26

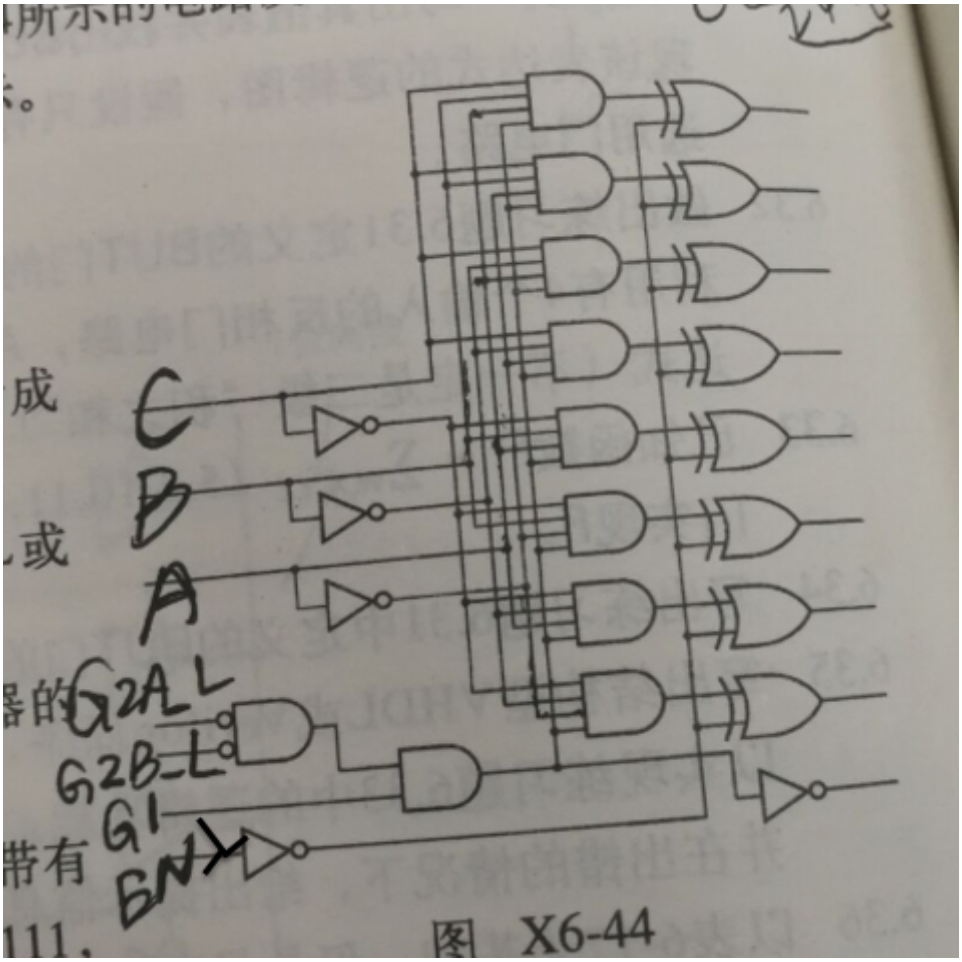
通过280芯片的延迟, 138以及86的某些总和就是整个电路的延迟

$$10 + 10 + 13 = 33(ns)$$

6.44(a,b)

解:

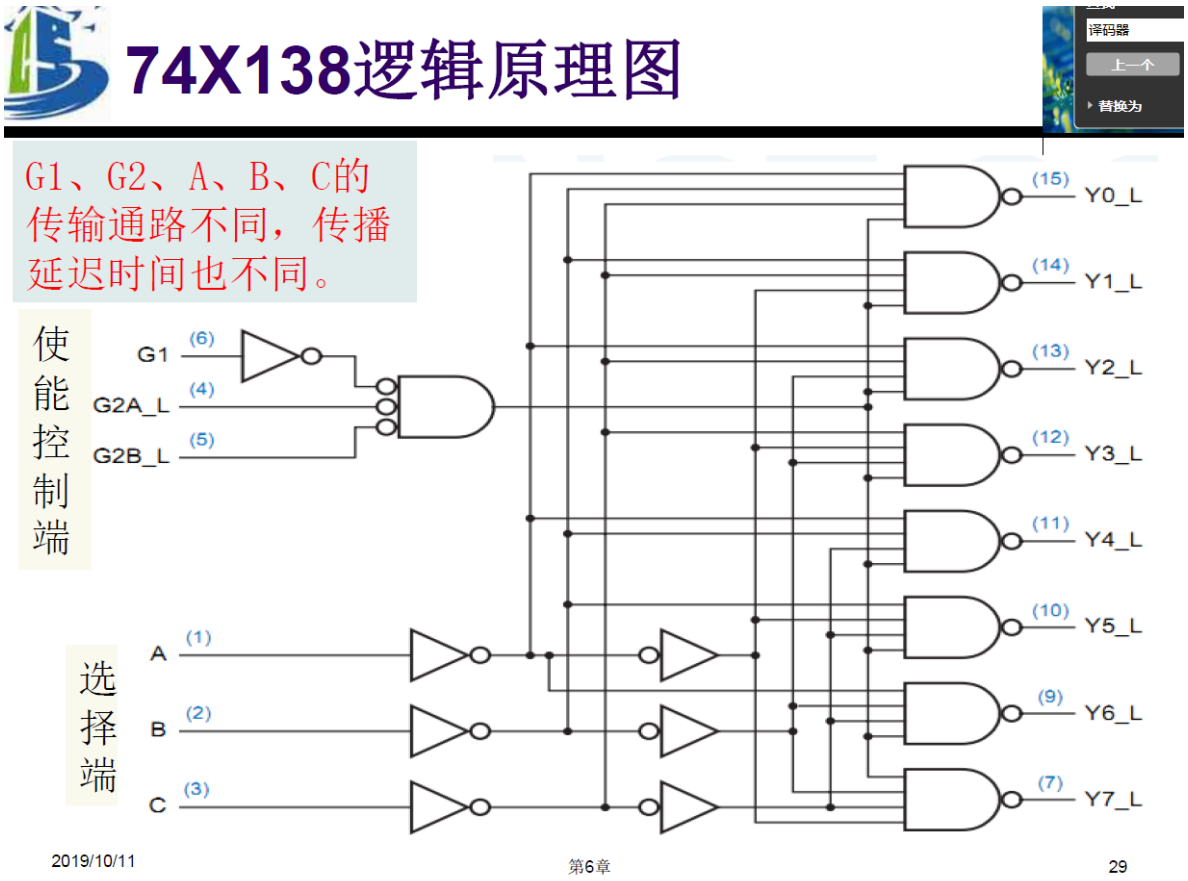
(a)



(b)

仔细观察(题目画的好密) 发现八个与门从上到下编号为1~8, C 输入 1, 2, 3, 4, 其余为 C', B 输入1256, 其余为B', A输入1357, 其余为A'. 所有与门后接输出的异或门;

毕竟之前手写了74x138内部结构, 很显然只看A, B, C 和八个与门的输出, 会意识到这即是一个: 当 `ABC` 表示什么值, 与门就把什么值筛选出来的部分.



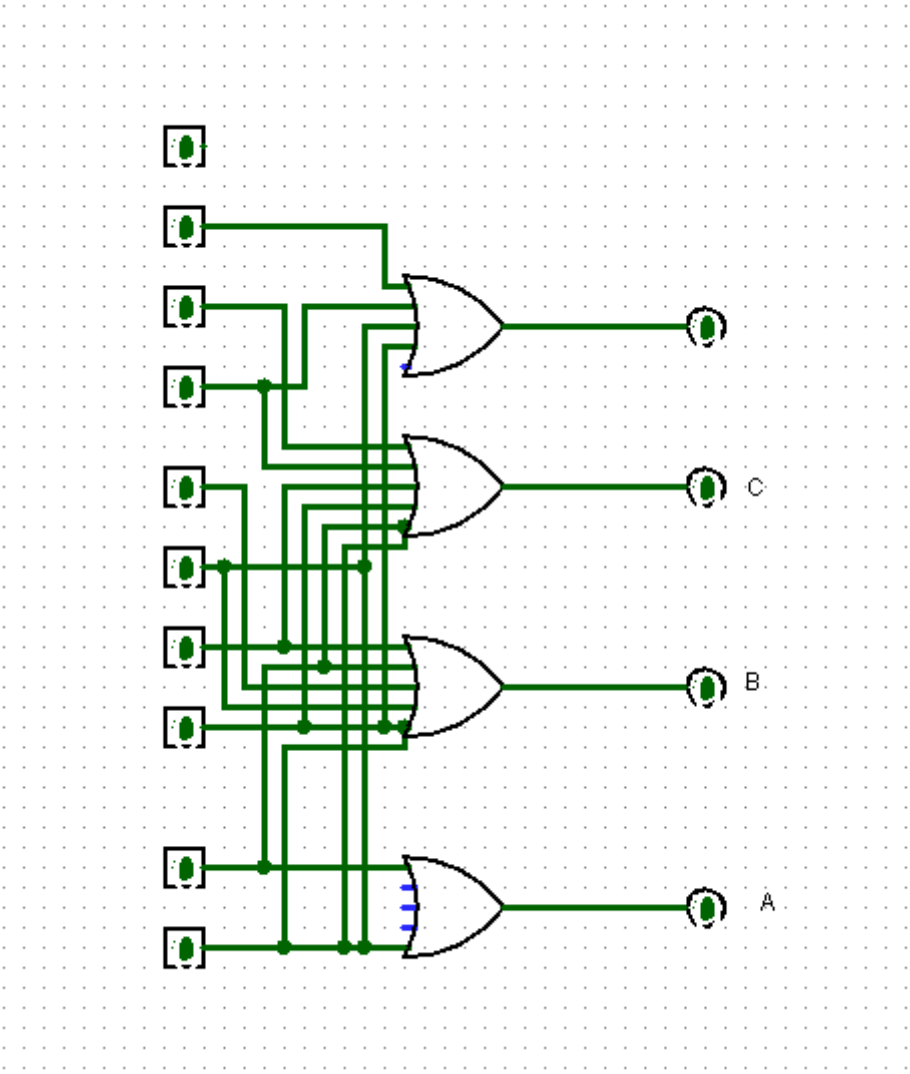
还有其他的异或门就是为了使能设置的, 可以发现多了一个全局**反相端**, 就是控制结果输出 `00000001` 还是 `11111110` 的.

电路的功能就是多一个反相端的 `3-8` 译码器.

6.50

解:

十个输入变量, 不可能用卡诺图方法解, `Q-M` 算法也太麻烦, 不妨从编码器的内部结构下手, 看真值表规律;



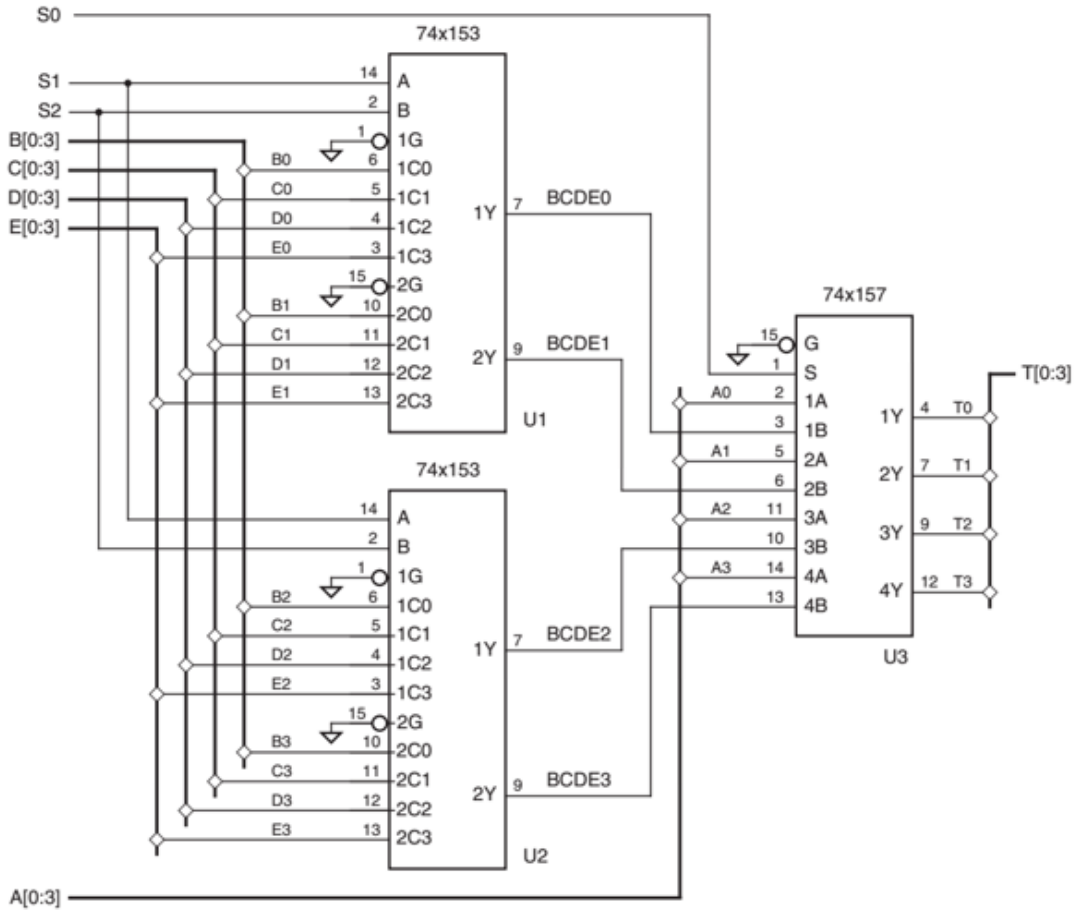
观察真值表知, 是 10-4 编码器内部实现稍微改一点, 改动的地方是: 对于 0000000010 和 0000000001 输出为 1110 (E)和 1111 (0xF), 上图输入端从上到下为1~10, 这里是输入10位取一码, 输出为 ABCD 表示的 BCD 码.

观察输出, 输出中有那个十位取一码为1, 就把这个数值对应的输入用或门连接起来就可以了. 电路很简单, 因为在真值表中有很多无关项可以利用.

真值表如下:

IN0	IN1	IN2	IN3	IN4	IN5	IN6	IN7	IN8	IN9	D	C	B	A
0	0	0	0	0	0	0	0	0	1	1	1	1	1
0	0	0	0	0	0	0	0	1	0	0	1	1	1
0	0	0	0	0	0	0	1	0	0	1	1	1	0
0	0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	1	0	0	0	0	1	0	1	0
0	0	0	0	1	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	1	1	0	0
0	0	1	0	0	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0

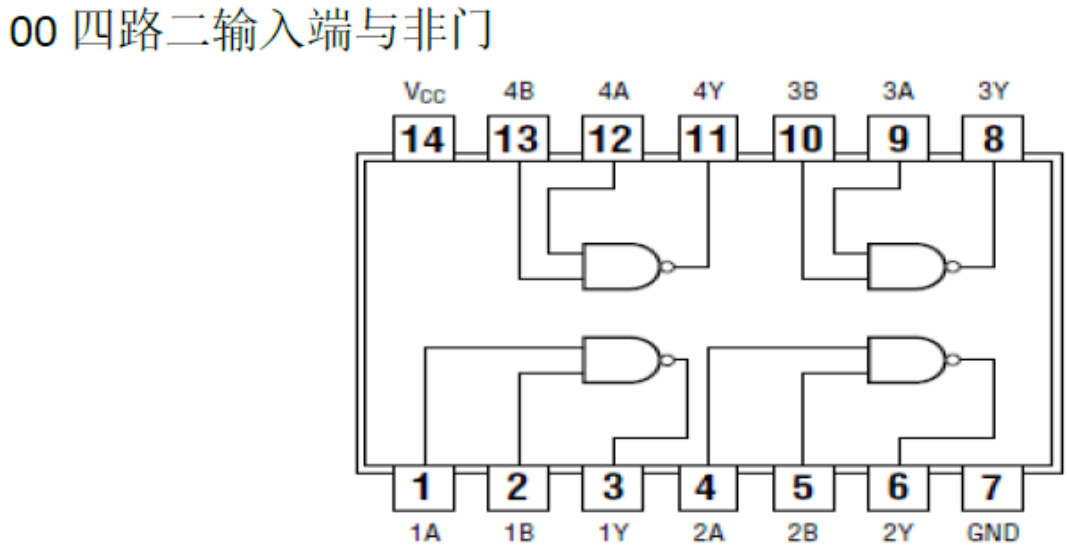
6.77



多路复用器的级联题, 具体思路为把每一层选择相应的位, 选择出来的再综合下一层继续选.

6.80

解:



注意到`74x00`引脚和`x08`的引脚是一样的(输入输出), 所以电路不会出问题.

但是`74x00`是与非门, `74x08`是与门, 注意到**除了读操作发生时**, 即RD == 0的时候, `I`都会被置为1, 这样的影响不管在读还是写操作的时候都是等效于输出 ODD 取反, 在存储芯片的 POUT 输出也有相同的效果, 所以后面的541接受的数据就抵消了, 而 ERROR 输出端的: 正好是两个输入端都取反:

在`74x00`下和`74x08`下的真值表对应为:

A	B	A	B
0	0	0	0
1	0	0	1
0	1	1	0
1	1	1	1

其中`A`, `B`是两个芯片的输入.

6.82

解:

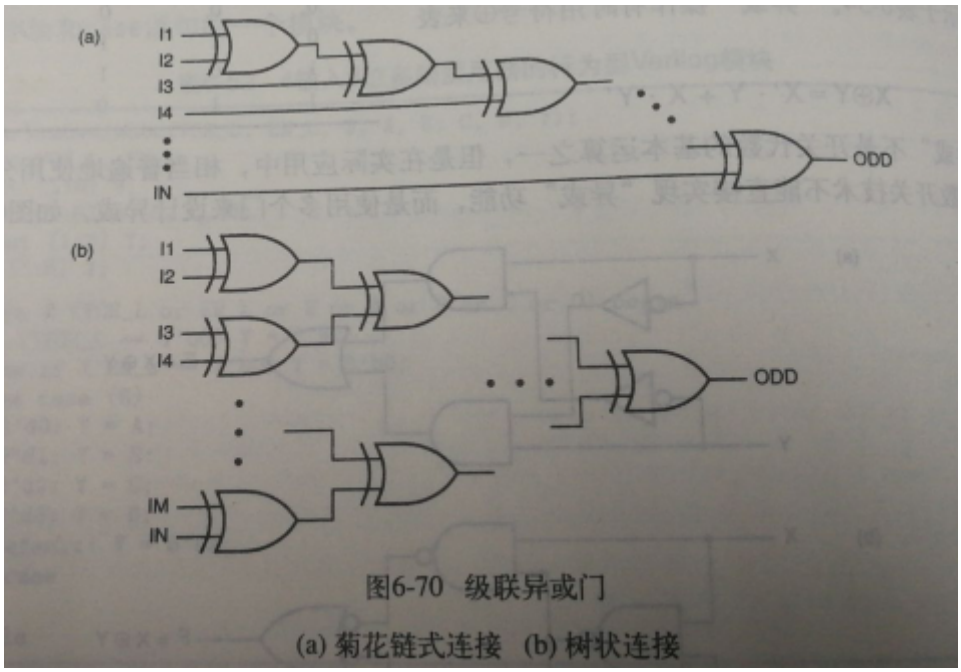


图6-70 级联异或门

(a) 菊花链式连接 (b) 树状连接

这里假设第一种为级联, 第二种为菊花式.

菊花式最大传播延迟: (考虑CMOS一路导通的情况) 假设有`n`个异或门，则该延迟时间为`2n-1`个异或门延迟时间的和。

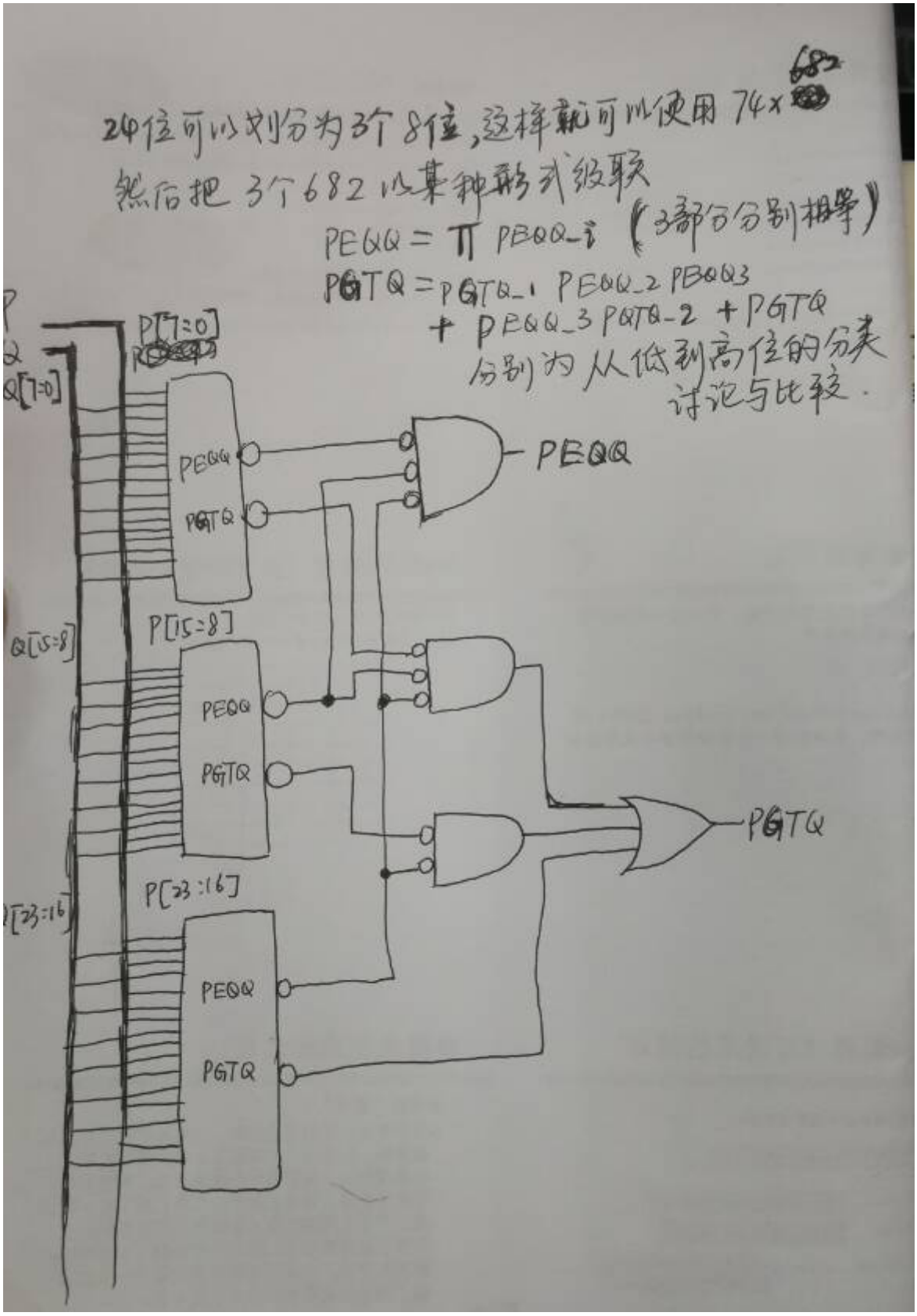
级联式最坏情况最小传播延迟: 为单个异或门延迟时间相加的和.

在不同情况下, 比如数据**传输进入输入端的一致性**来看两种方式的优势所在: 当所有信号同时到达输入端是适合采用级联结构，反之采用菊花结构。

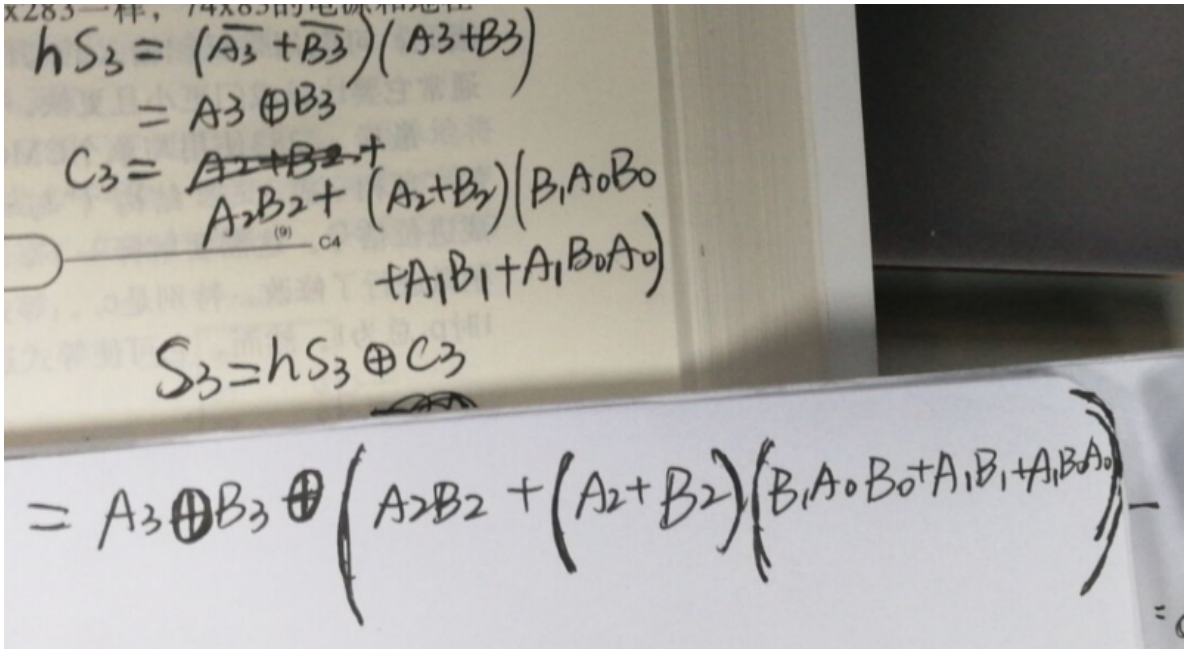
或者可以看异或门之间传输速度的差异, 如果差异较大而且分组有差异, 则使用级联可能更好一点.

> 就像图6-70的那种宽门电路, 通过将门电路类似树形组织起来, 可以获得较快的电路 (书$P325$)

6.96



6.100



这里是没有经过处理的表达式, 但是可以注意到一位全加器C_{out}的表达式在式子的后部, 加上第二位再这样依次进位上来, 进位和A_3, B_3(本位)之间的关系是异或(如上表达式), 从而可以知道这个表达式就是本来第三位的结果.

6.101

解:

32位二进制加法器有65个输出, 我们得到一个很松的上界: 2^{65} 个最小项, 但是显然没有用.

不妨依次看进位:

$$C_0 is\ 1\ productItem$$
$$c_1 = c_0(A_0 + B_0) + A_0B_0$$
$$c_2 = c_1(A_1 + B_1) + A_1B_1 = (A_1 + B_1)(c_0(A_0 + B_0) + A_0B_0) + A_1B_1$$

上面就是 $2 \times 3 + 1$ 项.

易得递推公式:

$$ans_i = 2 * ans_i + 1$$

所以共有

$$2^{33} - 1 = 8589934591$$