

# 东南大学电子科学与工程学院

# 实验报告

课程名称： 集成电路CAD

实验名称： 基于Tanner Pro 的 3-8 译码器设计

姓 名：孙寒石

学 号：06219109

实验地点：东南大学无锡国际校区

实验时间：2022-4-28

评定成绩：

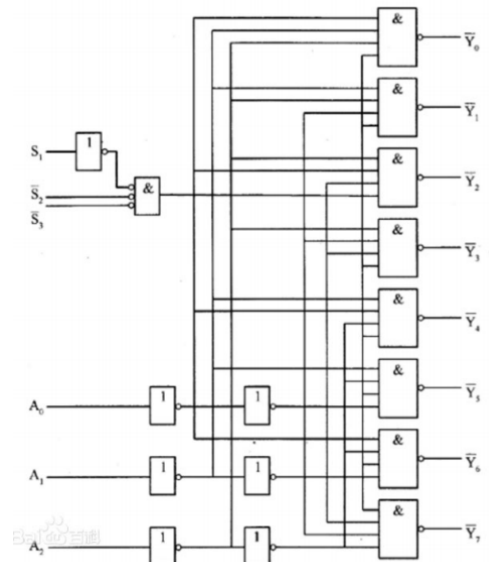
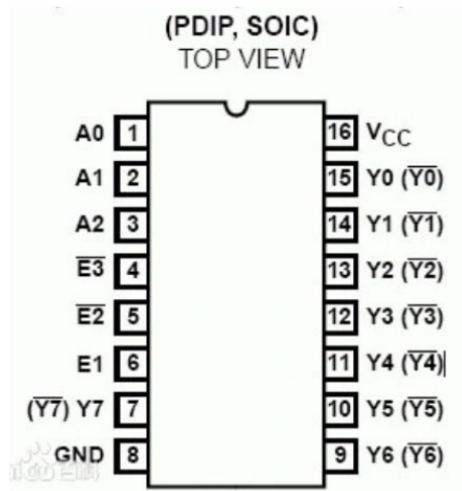
审阅教师：

# 基于 Tanner Pro 的 3-8 译码器设计

## 1 选题背景

3-8译码器是我们熟知的，早在大二的数电实验中，我们就用到过，当时用的是 74LS138。3-8 译码器的有 3 个 Input port，8 个 Output port。3-8 译码器的功能就是把输入的3位二进制数译码成为 10 进制的输出。举个例子，如果我们输入“110”，那么 Output 的第六个port就是“0”（低电平），其表示的二进制数字为 6（110）。

下面是 74LS138 的 Top view 和具体内部逻辑电路图。



## 2 实验内容

- 对 3-8 译码器的基本功能进行理论与逻辑仿真（这一步或许会用到 Quartus 或者 Multisim 等软件）
- 利用 Tanner Pro 的 S-Edit 进行电路设计和仿真
- 利用 Tanner Pro 的 L-Edit 进行布局图的设计

## 3 3-8 译码器功能分析及逻辑设计

### 3.1 功能分析

3-8 译码器的输入是 3 个口，输出是 8 个口。3-8 译码器的功能就是把输入的 3 位 2 进制数翻译成 10 进制的输出。3-8 译码器具有 3 个数据输入端  $A_2, A_1, A_0$ ，1 个片选输入端  $C_S$ ，8 个数据输出端  $Y_0 \sim Y_7$ ，实现数据输入端到输出端的译码逻辑功能，片选端  $C_S$  低电平有效。

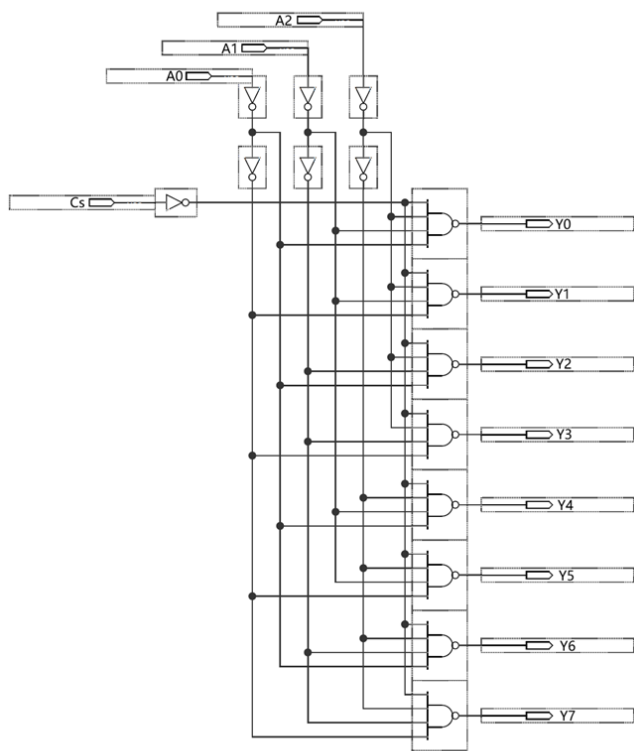
### 3.2 逻辑设计

各个输出口的逻辑表达式如下：

真值表如下：

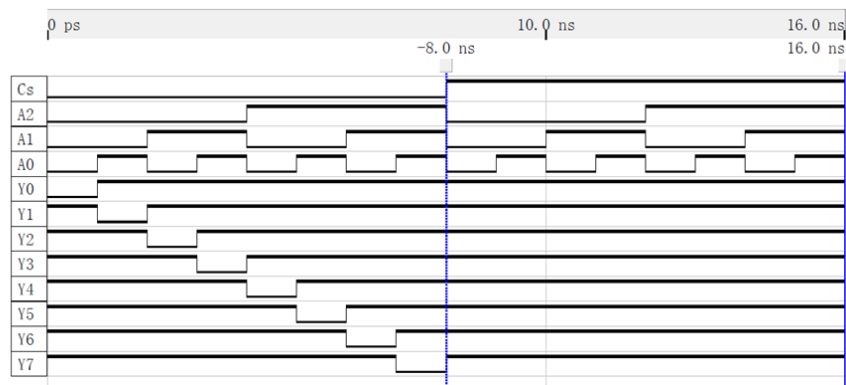
| 片选    | 数据输入  |       |       | 数据输出  |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| $C_s$ | $A_2$ | $A_1$ | $A_0$ | $Y_0$ | $Y_1$ | $Y_2$ | $Y_3$ | $Y_4$ | $Y_5$ | $Y_6$ | $Y_7$ |
| 0     | 0     | 0     | 0     | 0     | 1     | 1     | 1     | 1     | 1     | 1     | 1     |
| 0     | 0     | 0     | 1     | 1     | 0     | 1     | 1     | 1     | 1     | 1     | 1     |
| 0     | 0     | 1     | 0     | 1     | 1     | 0     | 1     | 1     | 1     | 1     | 1     |
| 0     | 0     | 1     | 1     | 1     | 1     | 1     | 0     | 1     | 1     | 1     | 1     |
| 0     | 1     | 0     | 0     | 1     | 1     | 1     | 1     | 0     | 1     | 1     | 1     |
| 0     | 1     | 0     | 1     | 1     | 1     | 1     | 1     | 1     | 0     | 1     | 1     |
| 0     | 1     | 1     | 0     | 1     | 1     | 1     | 1     | 1     | 1     | 0     | 1     |
| 0     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 0     |
| 1     | ×     | ×     | ×     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     |

根据 3-8 译码器的逻辑表达式及真值表，设计出相应的电路原理图，如图示。



3.3 逻辑仿真

在 Quartus (或者别的仿真软件都可以) 中对刚刚画出的 3-8 译码器进行仿真，设置仿真时长为 16ns， $C_s$  周期为 16ns， $A_2$  周期为 8ns， $A_1$  周期为 4ns， $A_0$  周期为 2ns。得到仿真波形图如下：



**结果：**在 0~1ns，输出  $Y_0$  为低电平；在 1~2ns，输出  $Y_1$  为低电平；在 2~3ns，输出  $Y_2$  为低电平；在 3~4ns，输出  $Y_3$  为低电平；在 4~5ns，输出  $Y_4$  为低电平；在 5~6ns，输出  $Y_5$  为低电平；在 6~7ns，输出  $Y_6$  为低电平；在 7~8ns，输出  $Y_7$  为低电平；在 8~16ns，使能端  $C_S$  为高电平，输出均为高电平；

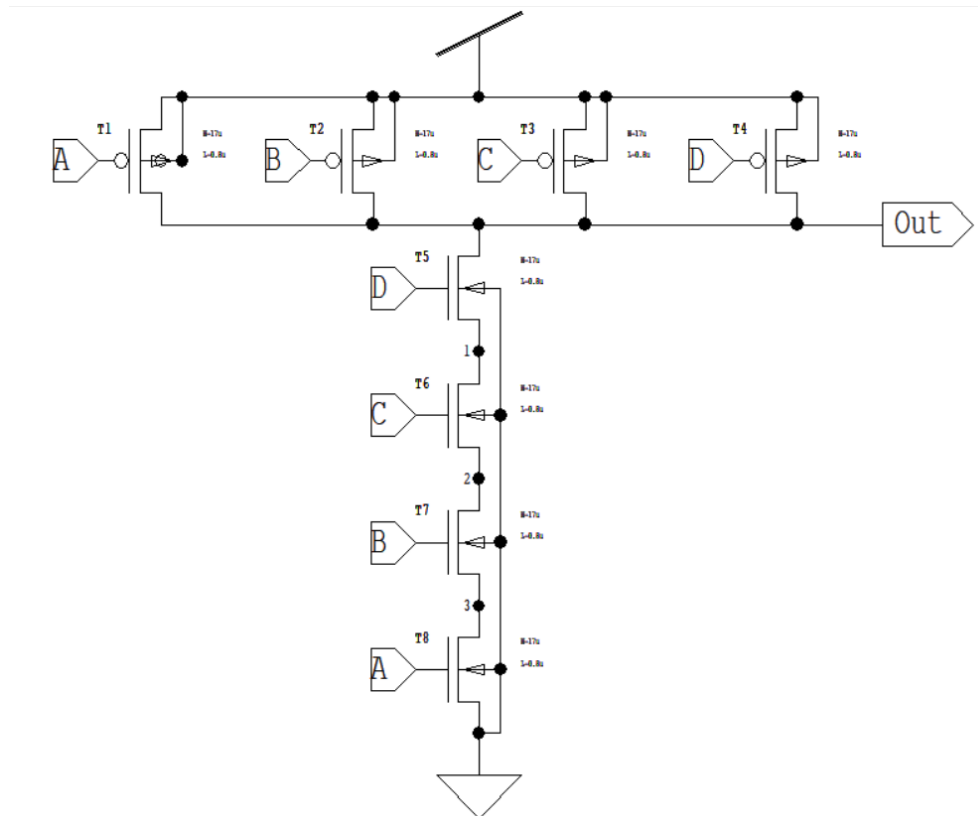
上述情况与真值表所有情况一一对应，因此该逻辑原理图实现了3-8译码器的功能。

## 4 S-Edit 电路设计

在本次设计中，按级顺序，3-8 译码器由 4 个 INV1、3 个 INV2、8 个 NAND4、8 个 INV3、8 个 INV4 模块构成，并在每个输出端接 10pF 负载电容。以下将对组成模块的电路原理图以及仿真情况做详细分析，以及对 3-8 译码器的总原理图和仿真作出分析。

### 4.1 NAND4 电路及仿真

在 S-Edit 的电路设计模块中画出 4 输入与非门的晶体管级电路图，如下图所示，其中 A、B、C、D 为输入端信号，OUT 为输出信号。



利用 T-Spice 对原理图输出的 SPICE 文件进行仿真，T-Spice 代码如下：

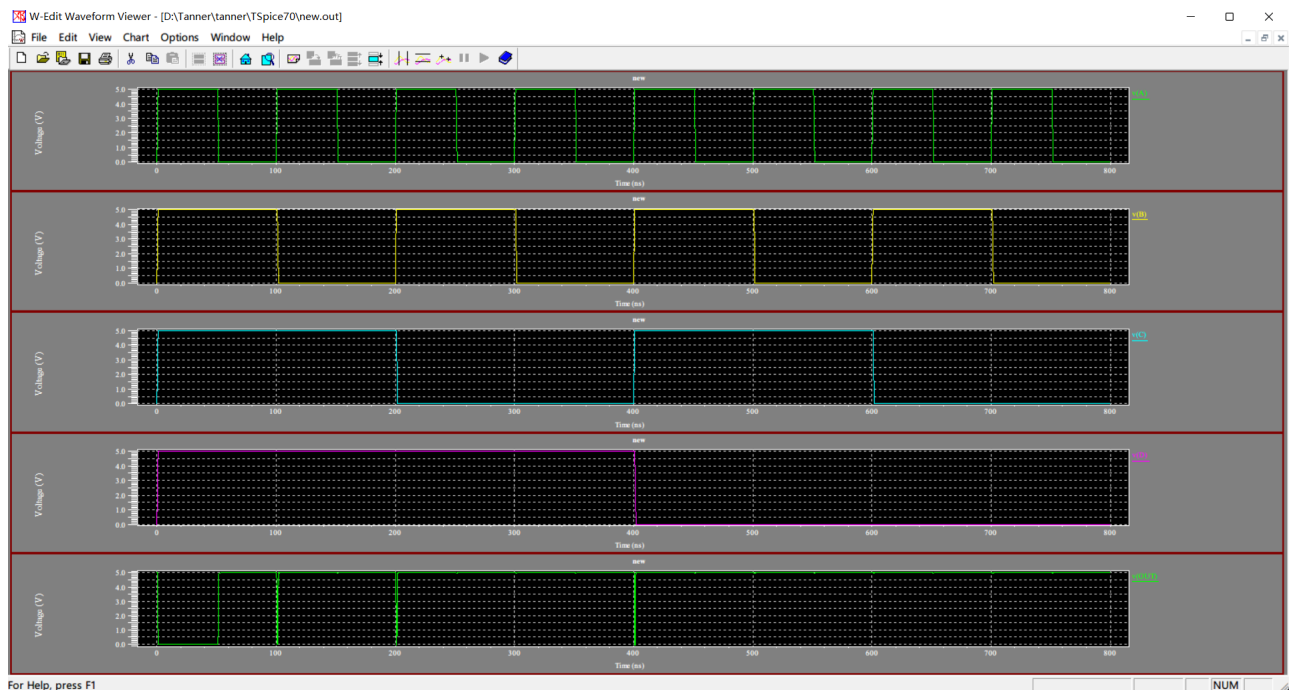
```
1 * SPICE netlist written by S-Edit Win32 7.03
2 * Written on Apr 28, 2022 at 11:29:59
3
4 * Waveform probing commands
5 .probe
6 .options probefilename="decoder3-8.dat"
7 + probesdbfile="D:\Tanner\exp_digital\decoder3-8.sdb"
8 + probetopmodule="NAND_TEST"
9
10 .SUBCKT NAND4 A B C D Out Gnd Vdd
```

```

11 M5 Out D 1 Gnd NMOS W=17u L=0.8u M=1
12 M6 1 C 2 Gnd NMOS W=17u L=0.8u M=1
13 M7 2 B 3 Gnd NMOS W=17u L=0.8u M=1
14 M8 3 A Gnd Gnd NMOS W=17u L=0.8u M=1
15 M1 Out A Vdd Vdd PMOS W=17u L=0.8u M=1
16 M2 Out B Vdd Vdd PMOS W=17u L=0.8u M=1
17 M3 Out C Vdd Vdd PMOS W=17u L=0.8u M=1
18 M4 Out D Vdd Vdd PMOS W=17u L=0.8u M=1
19 .ENDS
20
21 .include "D:\Tanner\tanner\TSpice70\models\m12_125.md"
22 * Main circuit: NAND_TEST
23 vvdd Vdd GND 5
24 va A GND PULSE (0 5 0 0n 0n 50n 100n)
25 vb B GND PULSE (0 5 0 0n 0n 100n 200n)
26 vc C GND PULSE (0 5 0 0n 0n 200n 400n)
27 vd D GND PULSE (0 5 0 0n 0n 400n 800n)
28 XNAND4_1 A B C D OUT Gnd Vdd NAND4
29 * End of main circuit: NAND_TEST
30 .tran 1n 800n
31 .print tran v(A) v(B) v(C) v(D) v(OUT)

```

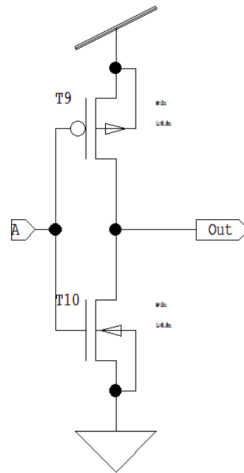
NAND4 的仿真输出模拟波形如图所示。



从上到下的信号依次为 A、B、C、D、OUT，由 NAND4 仿真波形图可知，当输入端信号 A、B、C、D 中含有低电平时，输出 OUT 为高电平。只有当 A、B、C 都为高电平时，输出 Y 才为低电平。符合 4 输入与非门的逻辑功能，从而验证了 NAND4 电路设计的正确性。

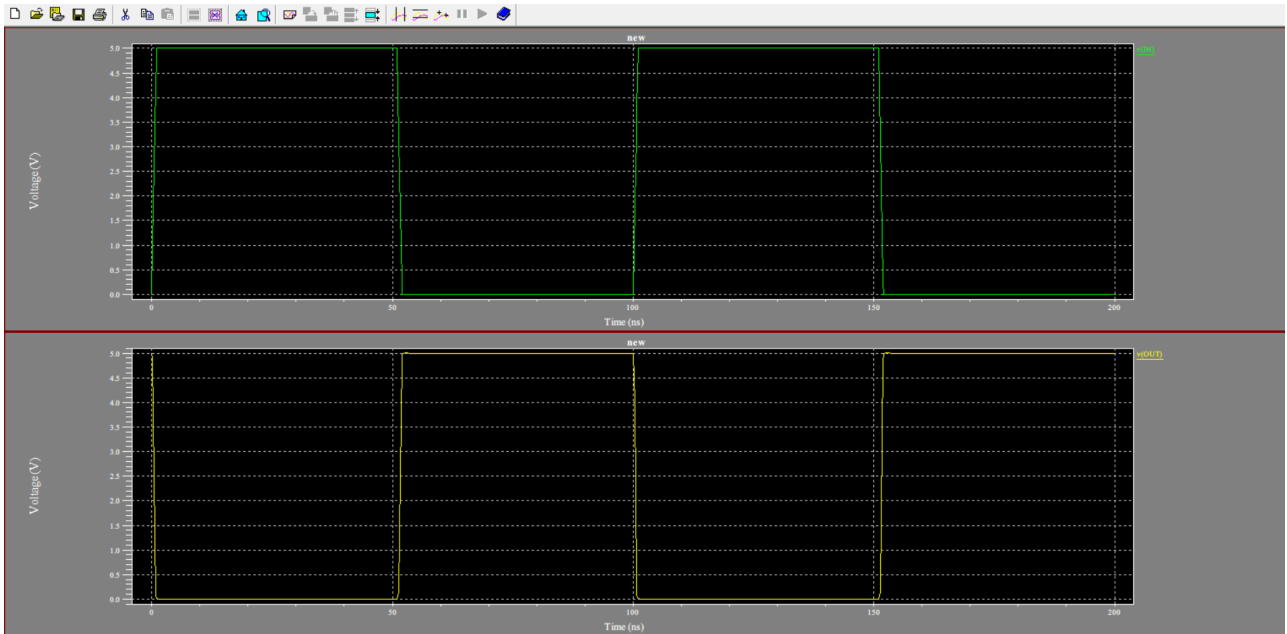
## 4.2 INV 电路及仿真

在 S-Edit 中的电路设计模块中画出第一级反相器 (INV1) 的电路图, 如图所示, 其中 A 为输入端信号, OUT 为输出端信号。



利用 T-Spice 对原理图输出的 SPICE 文件进行仿真, T-Spice 代码如下:

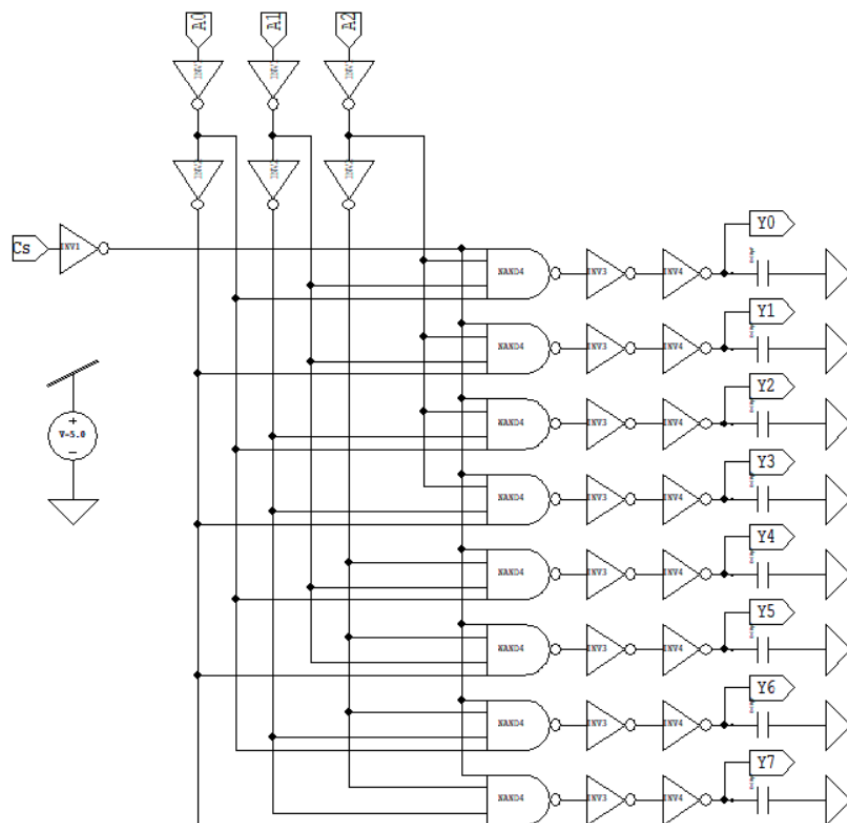
```
1  * SPICE netlist written by S-Edit Win32 7.03
2  * Written on Apr 28, 2022 at 12:39:11
3
4  * Waveform probing commands
5  .probe
6  .options probefilename="decoder3-8.dat"
7  + probesdbfile="D:\Tanner\exp_digital\decoder3-8.sdb"
8  + probetopmodule="INV1_TEST"
9
10 .SUBCKT INV1 A Out Gnd Vdd
11 M10 Out A Gnd Gnd NMOS W=2u L=0.8u M=1
12 M9 Out A Vdd Vdd PMOS W=2u L=0.8u M=1
13 .ENDS
14
15 .include "D:\Tanner\tanner\TSpice70\models\m12_125.md"
16 * Main circuit: INV1_TEST
17 vvdd Vdd GND 5
18 vin IN GND PULSE (0 5 0 0n 0n 50n 100n)
19 XINV1_1 IN OUT Gnd Vdd INV1
20 * End of main circuit: INV1_TEST
21 .tran 1n 200n
22 .print tran v(OUT) v(IN)
23
```



从上到下的信号依次为输入信号 A、输出信号 OUT，当输入信号 A 为高电平时，输出信号 OUT 为低电平。当输入信号 A 为低电平时，输出信号 OUT 为高电平。起到逻辑非的功能，从而实现反相器的功能。根据仿真可以看出 INV 电路设计的正确性。

### 4.3 译码器的电路图及仿真

在 S-Edit 中的电路设计模块下层次化组合出 3-8 译码器的电路图，如图所示。3 个数据输入端  $A_2, A_1, A_0$ ，1 个片选输入端  $C_S$ ，8 个数据输出端  $Y_0 \sim Y_7$ 。



由 3-8 译码器的电路图生成 SPICE 文件，用 T-Spice 打开生成的该电路生成的 SPICE 文件，对 SPICE 文件进行设置，包括加载包含文件、设定电源电压、设定输入信号、分析设定以及输出设定。设置的 SPICE 代码如下：

```

1  * SPICE netlist written by S-Edit Win32 7.03
2  * Written on Apr 28, 2022 at 13:48:54
3
4  * Waveform probing commands
5  .probe
6  .options probefilename="decoder3-8.dat"
7  + probesdbfile="D:\Tanner\exp_digital\decoder3-8.sdb"
8  + probetopmodule="decode3-8"
9
10 .SUBCKT INV1 A Out Gnd Vdd
11 M10 Out A Gnd Gnd NMOS W=2u L=0.8u M=1
12 M9 Out A Vdd Vdd PMOS W=2u L=0.8u M=1
13 .ENDS
14
15 .SUBCKT INV2 A Out Gnd Vdd
16 M12 Out A Gnd Gnd NMOS W=11u L=0.8u M=1
17 M11 Out A Vdd Vdd PMOS W=11u L=0.8u M=1
18 .ENDS
19
20 .SUBCKT INV3 A Out Gnd Vdd
21 M14 Out A Gnd Gnd NMOS W=50u L=0.8u M=1
22 M13 Out A Vdd Vdd PMOS W=50u L=0.8u M=1
23 .ENDS
24
25 .SUBCKT INV4 A Out Gnd Vdd
26 M16 Out A Gnd Gnd NMOS W=300u L=0.8u M=1
27 M15 Out A Vdd Vdd PMOS W=300u L=0.8u M=1
28 .ENDS
29
30 .SUBCKT NAND4 A B C D Out Gnd Vdd
31 M5 Out D 1 Gnd NMOS W=17u L=0.8u M=1
32 M6 1 C 2 Gnd NMOS W=17u L=0.8u M=1
33 M7 2 B 3 Gnd NMOS W=17u L=0.8u M=1
34 M8 3 A Gnd Gnd NMOS W=17u L=0.8u M=1
35 M1 Out A Vdd Vdd PMOS W=17u L=0.8u M=1
36 M2 Out B Vdd Vdd PMOS W=17u L=0.8u M=1
37 M3 Out C Vdd Vdd PMOS W=17u L=0.8u M=1
38 M4 Out D Vdd Vdd PMOS W=17u L=0.8u M=1
39 .ENDS
40
41 .include "D:\Tanner\tanner\TSpice70\models\m12_125.md"
42 * Main circuit: decoder3-8
43 va0 A0 GND PULSE (0 5 0 0n 0n 50n 100n)
44 va1 A1 GND PULSE (0 5 0 0n 0n 100n 200n)
45 va2 A2 GND PULSE (0 5 0 0n 0n 200n 400n)
46 vcs Cs GND PULSE (0 5 0 0n 0n 400n 800n)
47

```

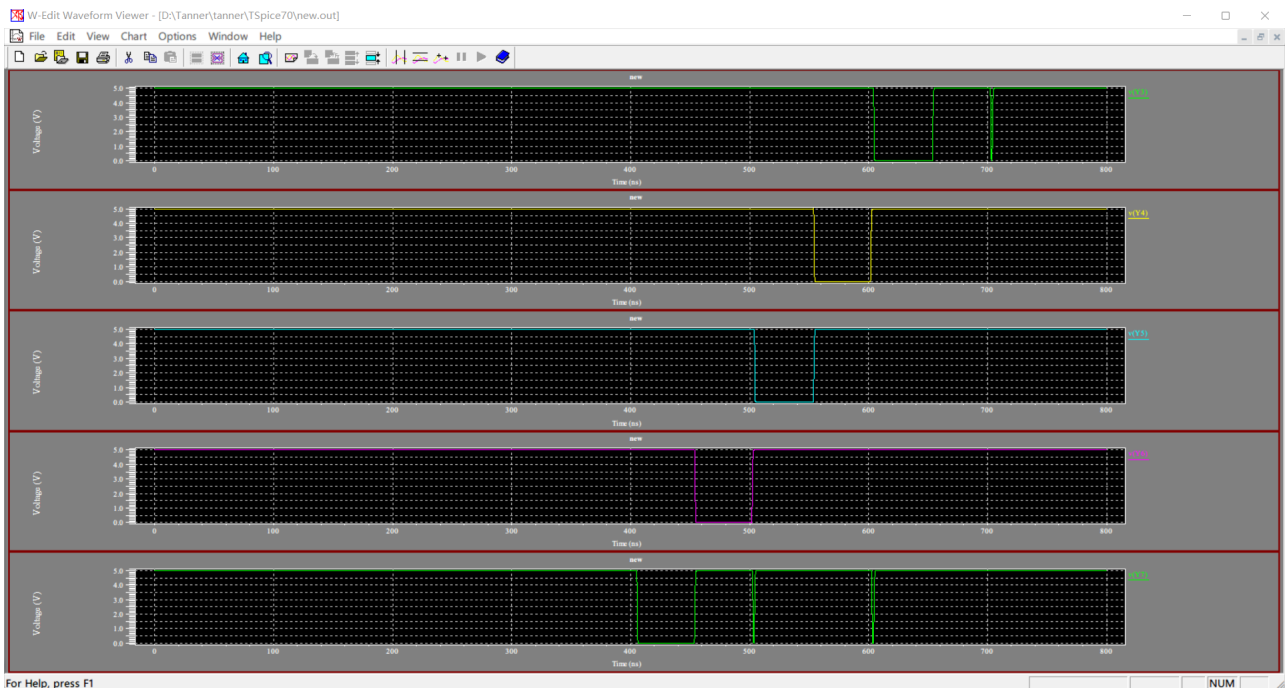


```

48 C1 Y0 Gnd 10pF
49 C2 Y1 Gnd 10pF
50 C3 Y2 Gnd 10pF
51 C4 Y3 Gnd 10pF
52 C5 Y4 Gnd 10pF
53 C6 Y5 Gnd 10pF
54 C7 Y6 Gnd 10pF
55 C8 Y7 Gnd 10pF
56 XINV1_1 A2 N6 Gnd Vdd INV1
57 XINV1_2 A1 N4 Gnd Vdd INV1
58 XINV1_3 A0 N2 Gnd Vdd INV1
59 XINV1_4 Cs N7 Gnd Vdd INV1
60 XINV2_1 N6 N5 Gnd Vdd INV2
61 XINV2_2 N4 N3 Gnd Vdd INV2
62 XINV2_3 N2 N1 Gnd Vdd INV2
63 XINV3_1 N13 N9 Gnd Vdd INV3
64 XINV3_2 N15 N14 Gnd Vdd INV3
65 XINV3_3 N17 N16 Gnd Vdd INV3
66 XINV3_4 N19 N18 Gnd Vdd INV3
67 XINV3_5 N21 N20 Gnd Vdd INV3
68 XINV3_6 N23 N22 Gnd Vdd INV3
69 XINV3_7 N25 N24 Gnd Vdd INV3
70 XINV3_8 N27 N26 Gnd Vdd INV3
71 XINV4_1 N9 Y0 Gnd Vdd INV4
72 XINV4_2 N14 Y1 Gnd Vdd INV4
73 XINV4_3 N16 Y2 Gnd Vdd INV4
74 XINV4_4 N18 Y3 Gnd Vdd INV4
75 XINV4_5 N20 Y4 Gnd Vdd INV4
76 XINV4_6 N22 Y5 Gnd Vdd INV4
77 XINV4_7 N24 Y6 Gnd Vdd INV4
78 XINV4_8 N26 Y7 Gnd Vdd INV4
79 XNAND4_1 N7 N6 N4 N2 N13 Gnd Vdd NAND4
80 XNAND4_2 N7 N6 N4 N1 N15 Gnd Vdd NAND4
81 XNAND4_3 N7 N6 N3 N2 N17 Gnd Vdd NAND4
82 XNAND4_4 N7 N6 N3 N1 N19 Gnd Vdd NAND4
83 XNAND4_5 N7 N5 N4 N2 N21 Gnd Vdd NAND4
84 XNAND4_6 N7 N5 N4 N1 N23 Gnd Vdd NAND4
85 XNAND4_7 N7 N5 N3 N2 N25 Gnd Vdd NAND4
86 XNAND4_8 N7 N5 N3 N1 N27 Gnd Vdd NAND4
87 v9 Vdd Gnd 5.0
88 * End of main circuit: decoder3-8
89 .tran 1n 800n
90 .print tran v(Y7) v(Y6) v(Y5) v(Y4) v(Y3) v(Y2) v(Y1) v(Y0) v(A0) v(A1) v(A2) v(Cs)

```

根据加载后的 SPICE 文件，生成仿真波形图如图：



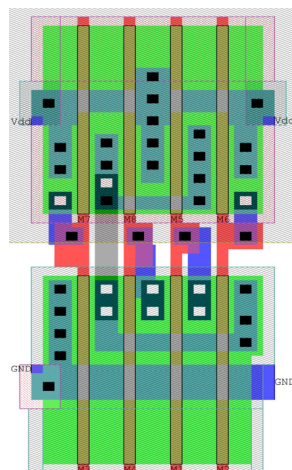
虽然有些地方有毛刺，但是从图中可以直观看出上述情况与真值表所有情况一一对应，与做的逻辑仿真结果类似，因此该原理图实现了 3-8 译码器的功能，设计很成功！

## 5 L-Edit 电路设计

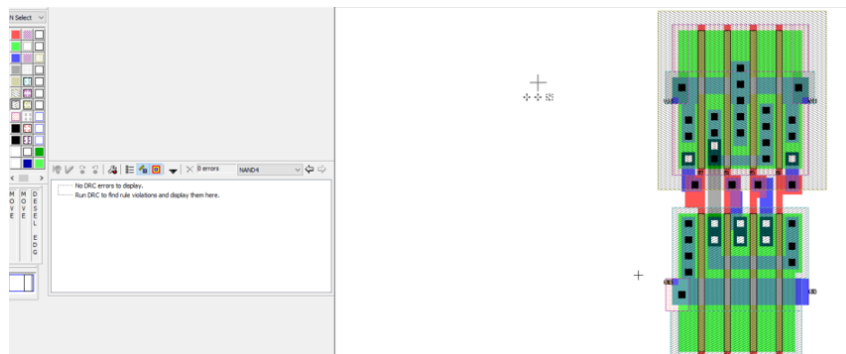
在 L-Edit 中将总版图分为 NAND4、INV1、INV2、INV3、INV4 不同的 cell，再组合成总版图。下面将对各个 cell 和总版图进行设计以及 DRC。

### 5.1 NAND4 的版图设计与 DRC

根据 NAND4 电路图，在 L-Edit 中画出 NAND4 的版图如图所示。



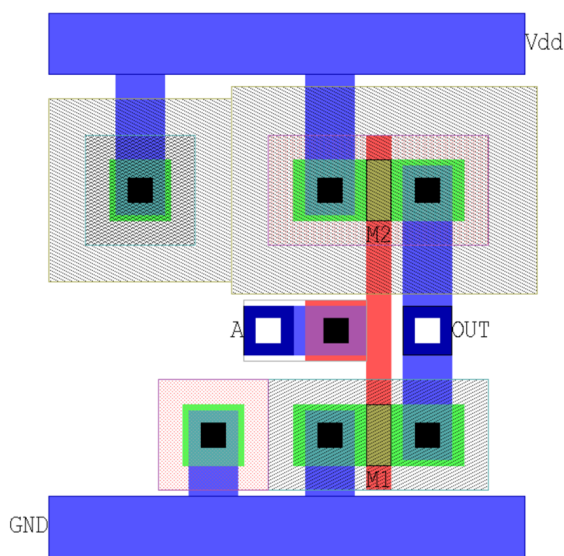
进行 DRC Check，得到如下图结果：



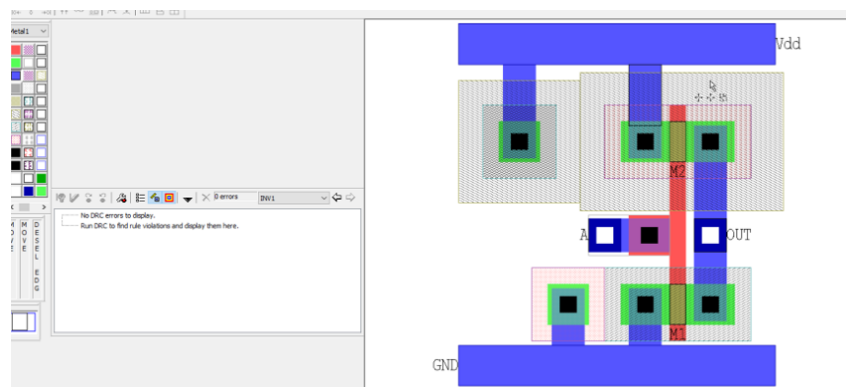
可知所设计的 NAND4 的版图符合规则。

## 5.2 INV 的版图设计与 DRC

根据反相器的电路图在 L-Edit 中画出版图，如图所示。其中 A 为输入端信号，OUT 为输出端信号。



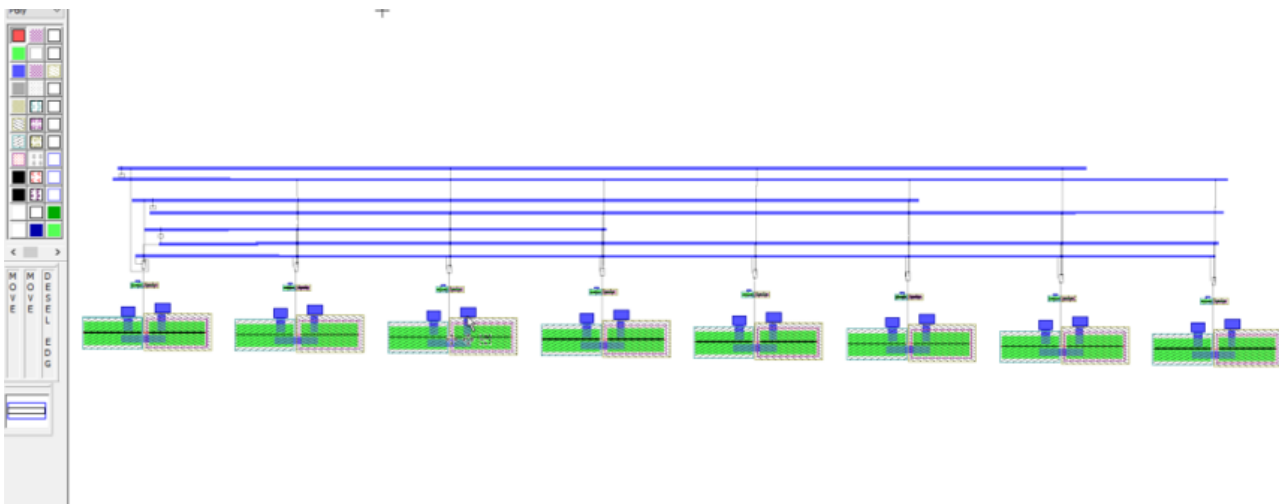
进行 DRC Check，得到如下图结果：



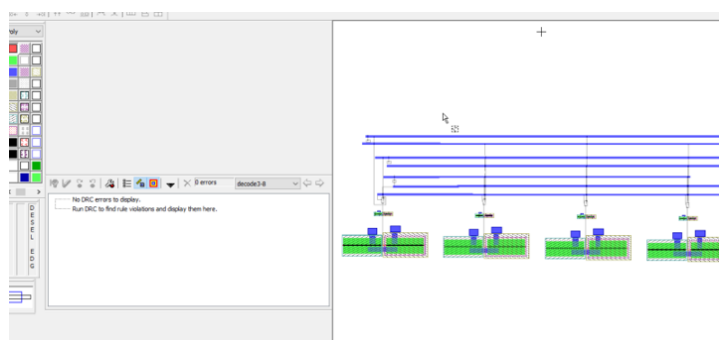
可知所设计的 INV 的版图符合规则。

## 5.3 译码器的版图设计及 DRC

根据分别设计出的 INV1、INV2、INV3、INV4 并通过 DRC 验证后，由设计的译码器电路图和已有的版图模块在 L-Edit 中编辑 3-8 译码器的版图，如图所示。



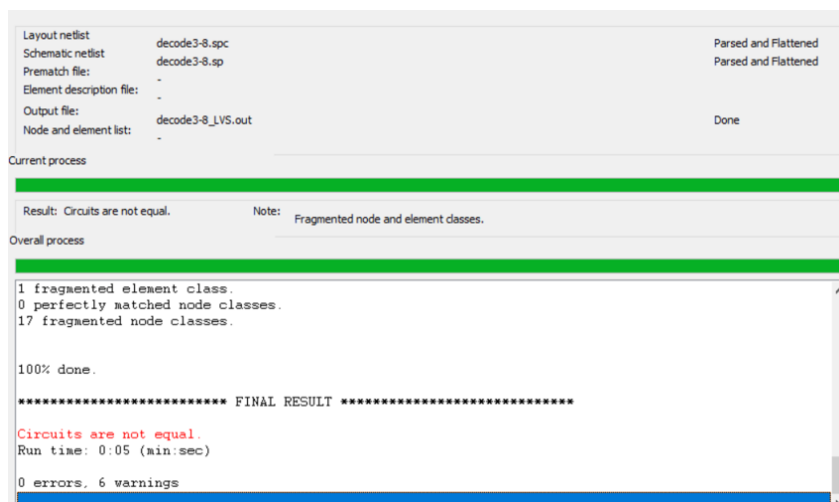
对该版图进行 DRC 检查，其结果报告如图所示。



从 译码器 的 DRC 结果中可以看出译码器版图在设计规则方面的正确性。

## 6 LVS

版图绘制完后，除需要通过设计规则检查外，还要与原理图进行对比，以检查在版图中实际形成的电路的与原理图中的电路是否一致。对该设计的 LVS 结果如图所示。



没有 error，但是有一些 warnings。