

《数字电路与数字系统实验》实验报告

第 2 次实验： 二位四选一选择器

姓名： 张逸凯

学号： 171840708

院系： 物理 学院

邮箱：

电话：

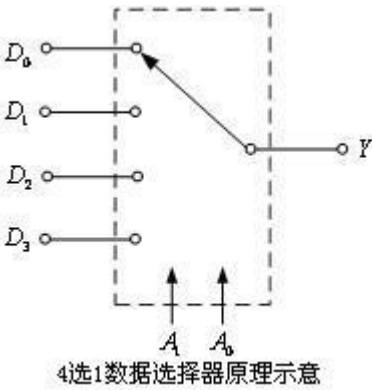
一.实验目的

复习数电相关知识以及选择器的实现原理,学会在 verilog 语言下设计一个二位的四选一选择器

二.实验原理（ 知识背景, 结合理论课总结 ）

基本定义：

在多路[数据传送](#)过程中,能够根据需要将其中的任意一路选出来的电路,叫做数据选择器,也称[多路选择器](#)或[多路开关](#).



基本工作原理：

选择器常用于信号的切换,四选一选择器常用于信号的切换,四选一选择器可以用于 4 路信号的切换, 四选一选择器有四个输入端 $x[0] \sim x[3]$ (图上的 $D0 \sim D3$),两个信号选择端 $A0$ 和 $A1$ 及一个信号输出端 Y .

当给 $A1$, $A0$ 一组信号 比如 1 0 那么就相当于给了它一个二进制数字 2, 也就相当于选了 $D2$ 这个输入端.

这个时候 输出 Y 输出的就是 $D2$ 的信号, $D2$ 是啥 Y 就输出啥.

在本次实验中只是将输入 $D0$ 到 $D3$ 改为了两位数的输入.

输出表：

控制		输出
A1	A0	Y
	0	0 $D0$
	0	1 $D1$
	1	0 $D2$
	1	1 $D3$

三.实验设备环境

硬件器材：FPGA 开发板.

软件平台：Qaartus 开发平台.

四.实验步骤 / 过程（设计思路、设计代码、测试代码、仿真结果和硬件实现等的截图代码等）

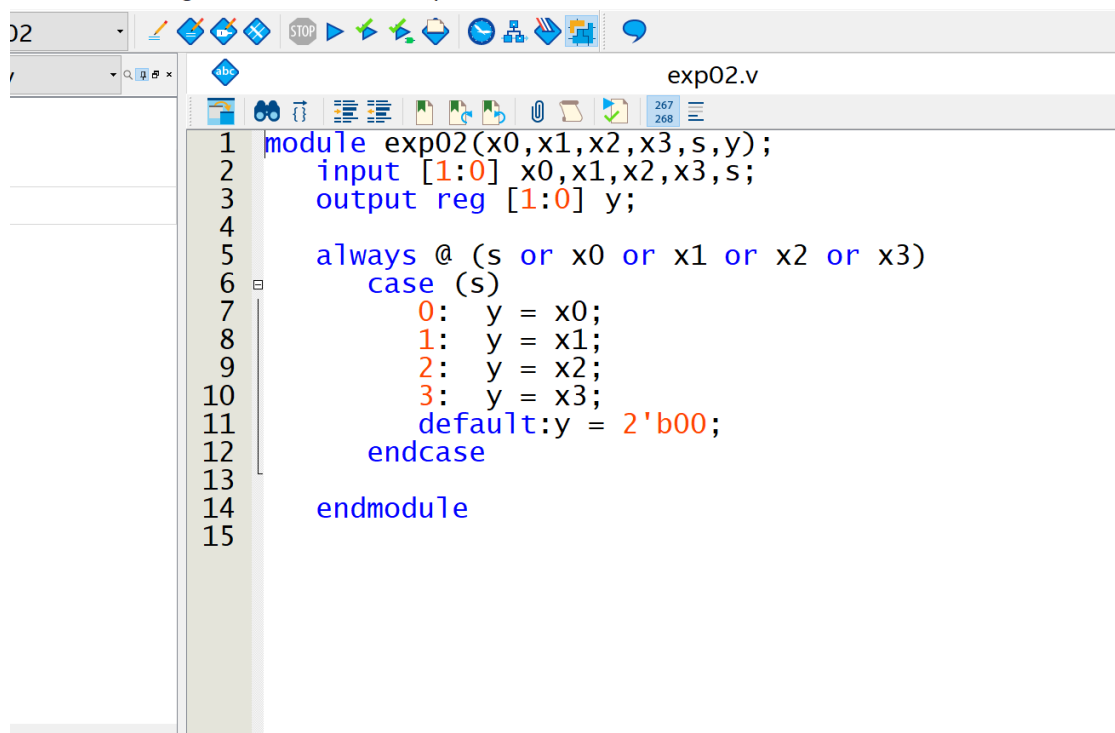
设计思路:

想到的第一设计思路就是使用 **case** 语句, 而不是使用 **if**、**else** 语句, 因为这样最简洁, 所以可以模仿 ppt 给出的例子写出代码

设计代码:

E:/My_design/exp02/exp02 - exp02

ents Processing Tools Window Help



```
1 module exp02(x0,x1,x2,x3,s,y);
2     input [1:0] x0,x1,x2,x3,s;
3     output reg [1:0] y;
4
5     always @ (s or x0 or x1 or x2 or x3)
6     case (s)
7         0: y = x0;
8         1: y = x1;
9         2: y = x2;
10        3: y = x3;
11        default:y = 2'b00;
12    endcase
13
14 endmodule
15
```

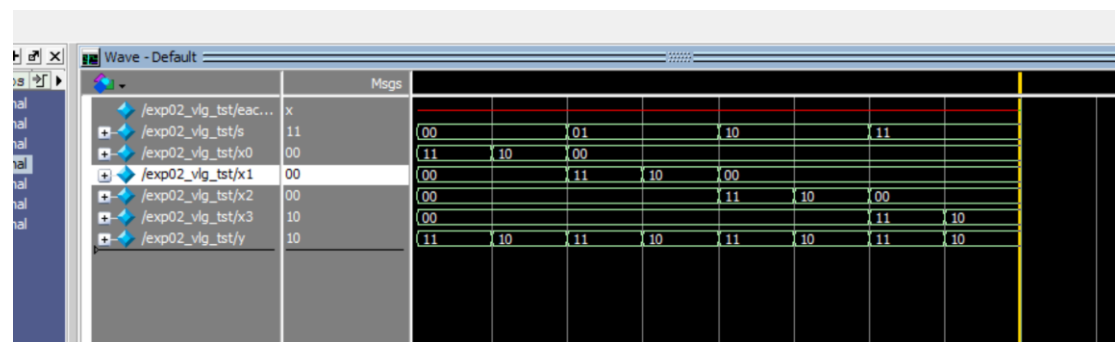
图1 设计代码

激励代码:

```
exp02.v
Compilation Report - exp02
exp02.v

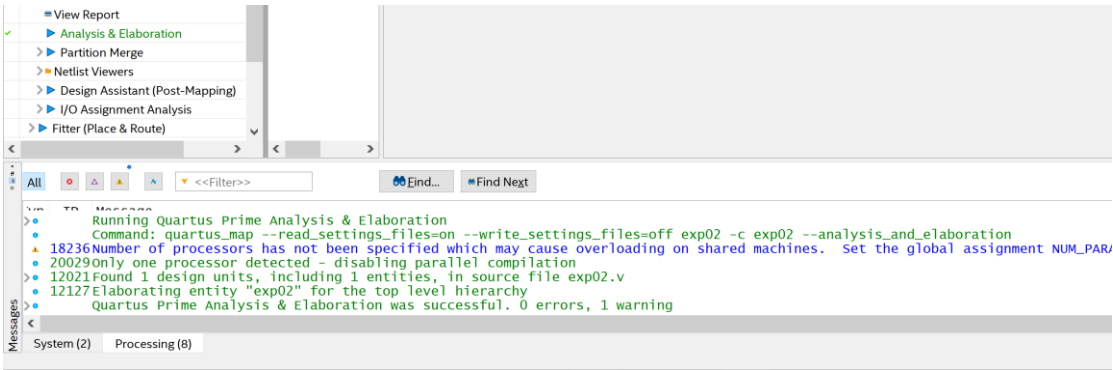
35 reg [1:0] x1;
36 reg [1:0] x2;
37 reg [1:0] x3;
38 // wires
39 wire [1:0] y;
40
41 // assign statements (if any)
42 exp02 i1 (
43 // port map - connection between master ports and signals/registers
44 .s(s),
45 .x0(x0),
46 .x1(x1),
47 .x2(x2),
48 .x3(x3),
49 .y(y)
50 );
51 initial
52 begin
53 // code that executes only once
54 // insert code here --> begin
55 s = 2'b00; x0 = 2'b11; x1 = 2'b00; x2 = 2'b00; x3 = 2'b00; #10;
56 s = 2'b01; x0 = 2'b10; x1 = 2'b00; x2 = 2'b00; x3 = 2'b00; #10;
57 s = 2'b00; x0 = 2'b00; x1 = 2'b11; x2 = 2'b00; x3 = 2'b00; #10;
58 s = 2'b00; x0 = 2'b00; x1 = 2'b10; x2 = 2'b00; x3 = 2'b00; #10;
59 s = 2'b10; x0 = 2'b00; x1 = 2'b00; x2 = 2'b11; x3 = 2'b00; #10;
60 s = 2'b00; x0 = 2'b00; x1 = 2'b00; x2 = 2'b10; x3 = 2'b00; #10;
61 s = 2'b11; x0 = 2'b00; x1 = 2'b00; x2 = 2'b00; x3 = 2'b11; #10;
62 s = 2'b00; x0 = 2'b00; x1 = 2'b00; x2 = 2'b00; x3 = 2'b10; #10;
63
64 // --> end
65 // $display("Running testbench");
66 end
67 // always
68 // optional sensitivity list
69 // @(event1 or event2 or .... eventn)
70 // begin
71 // code executes for every event on sensitivity list
72 // insert code here --> begin
73
74
75
```

ModelSim 仿真波形:



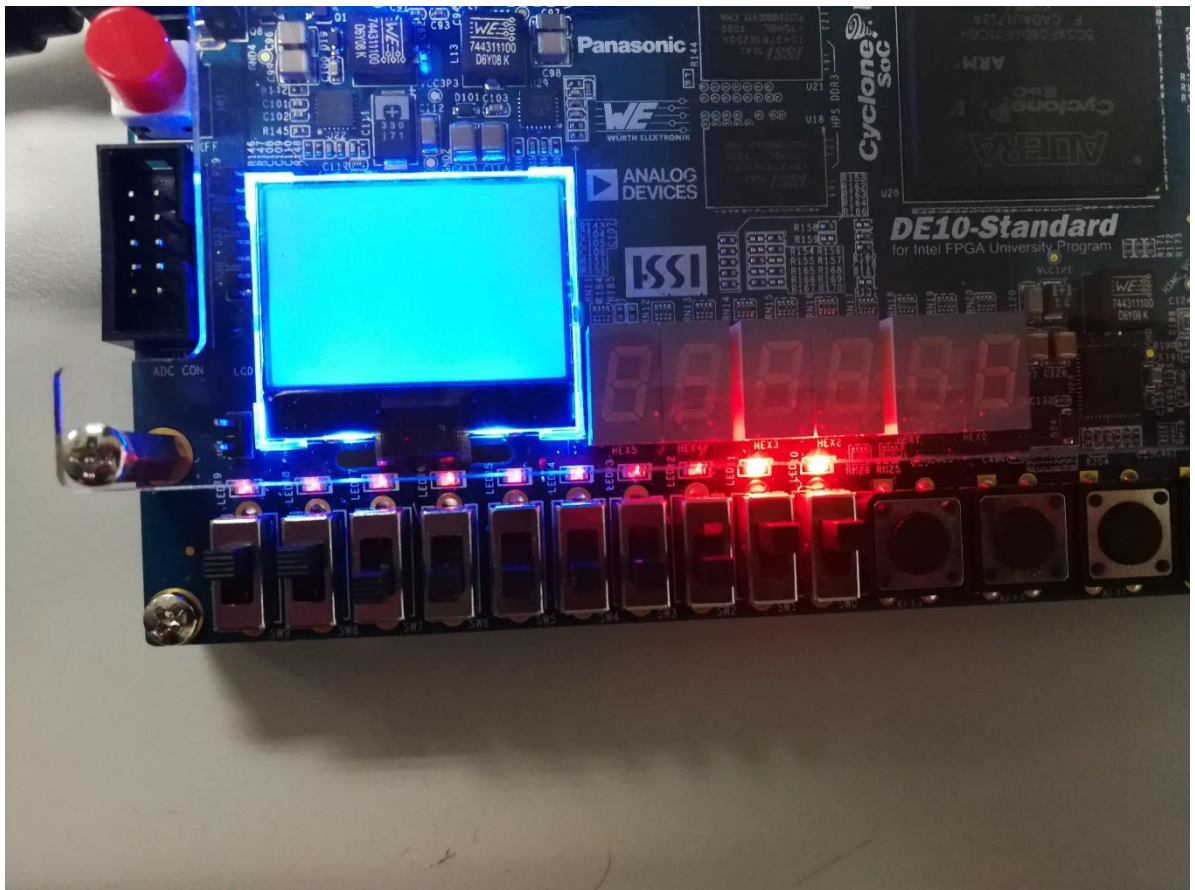
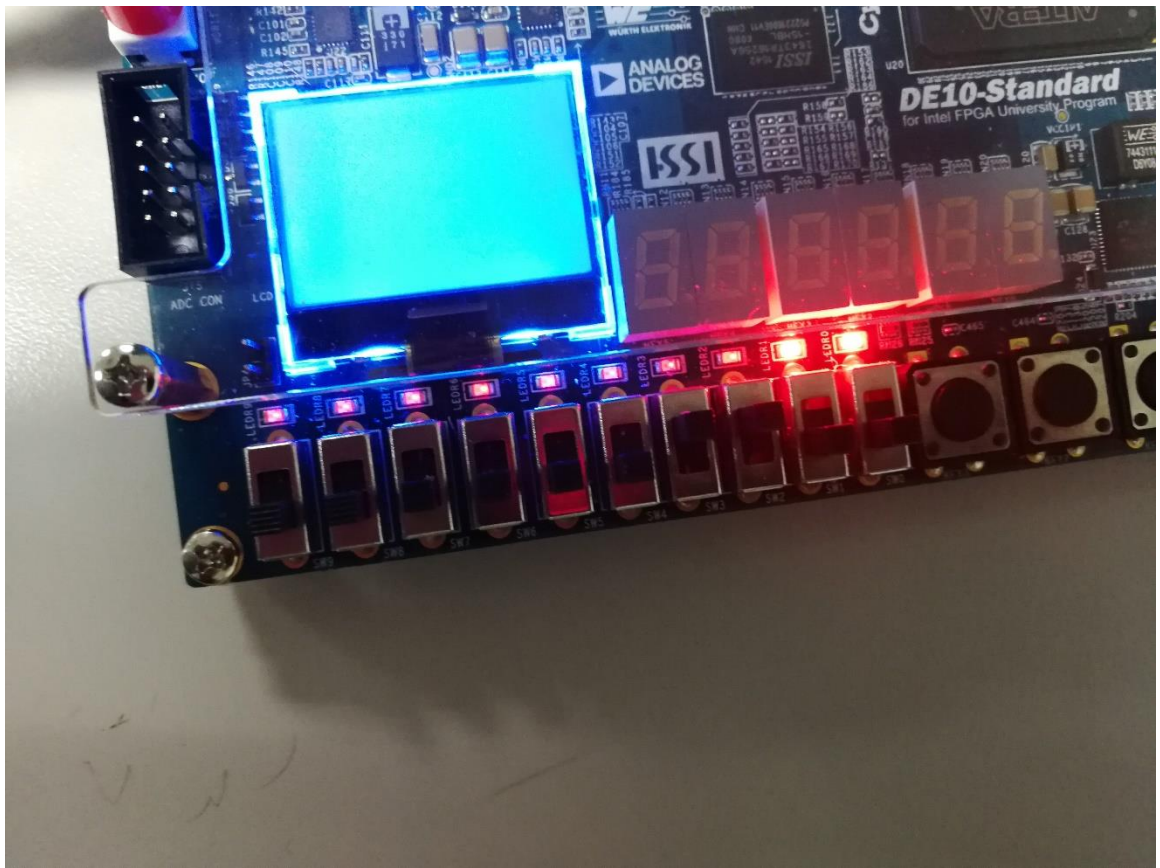
硬件实现:

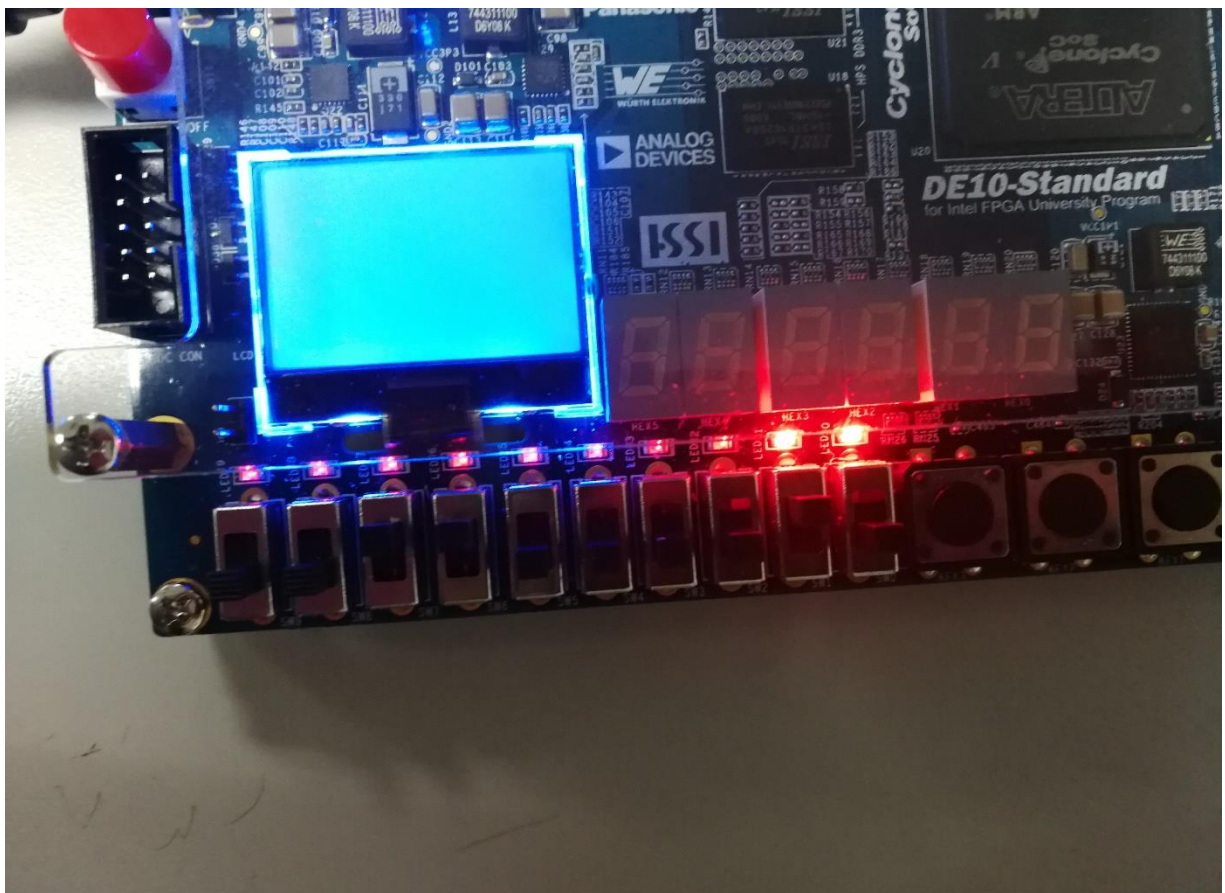
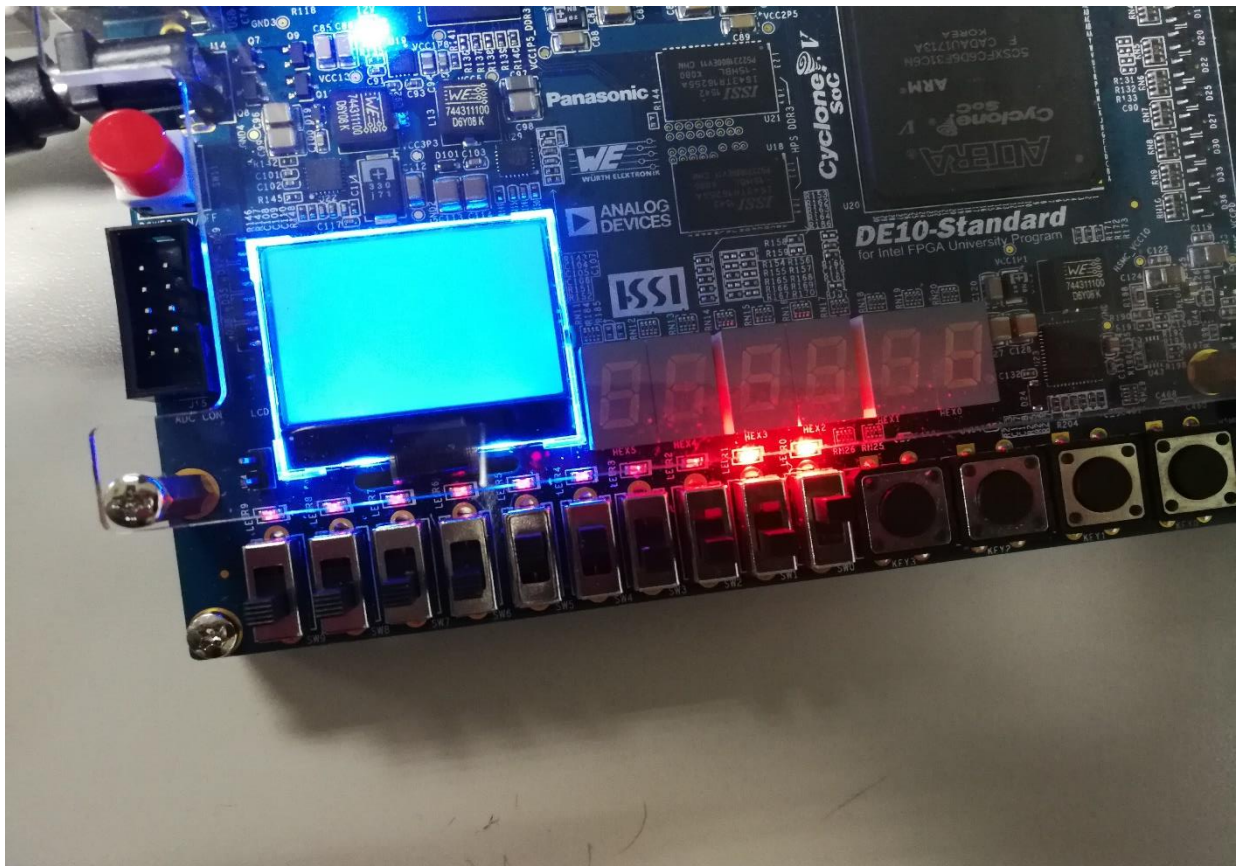
分配前:



引脚分配:

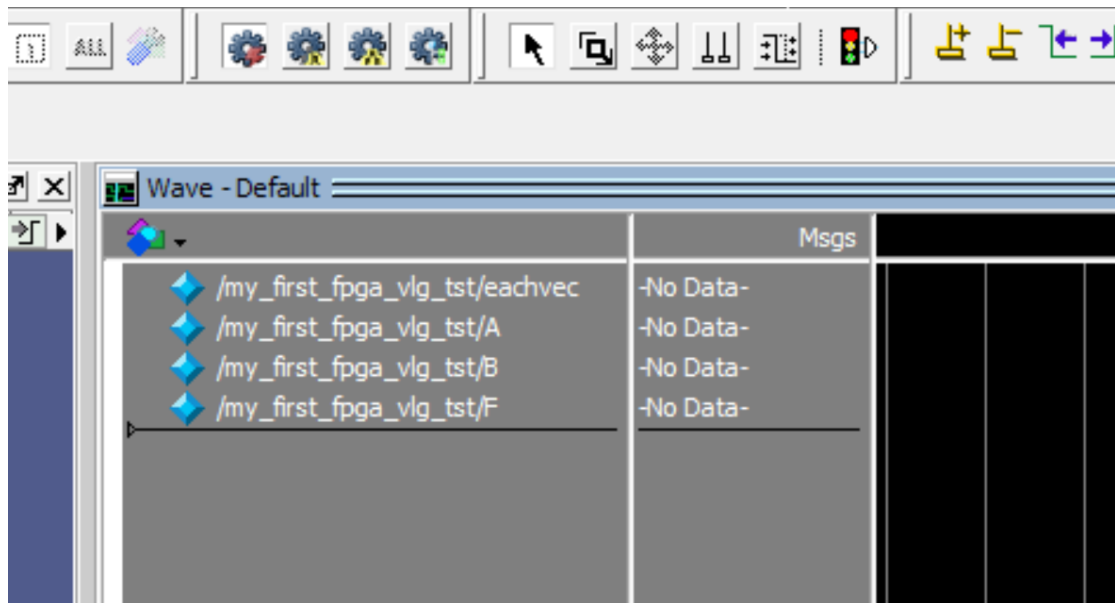
Named: * Edit: x PIN_Y27					
Node Name	Direction	Location	I/O Bank		
s[1]	Input	PIN_Y27	5B	B5B_NO	
s[0]	Input	PIN_AB30	5B	B5B_NO	
x0[1]	Input	PIN_AC30	5B	B5B_NO	
x0[0]	Input	PIN_AB28	5B	B5B_NO	
x1[1]	Input	PIN_V25	5B	B5B_NO	
x1[0]	Input	PIN_W25	5B	B5B_NO	
x2[1]	Input	PIN_AD30	5B	B5B_NO	
x2[0]	Input	PIN_AC28	5B	B5B_NO	
x3[1]	Input	PIN_AA30	5B	B5B_NO	
x3[0]	Input	PIN_AC29	5B	B5B_NO	
y[1]	Output	PIN_AB23	5A	B5A_NO	
y[0]	Output	PIN_AA24	5A	B5A_NO	
<<new node>>					





五.实验中遇到的问题及解决方案（请具体的描述问题和解决方法）

在模拟波形时，出现了模拟文件为空的情况

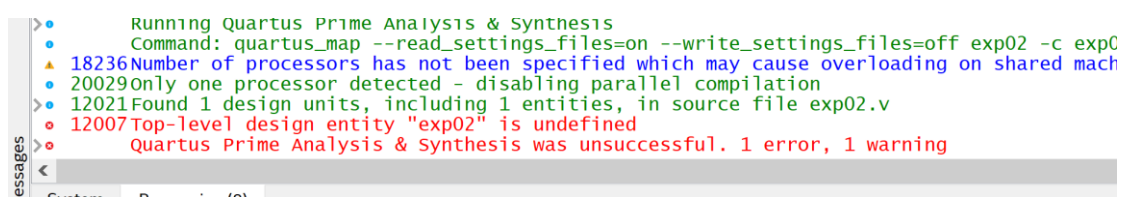


可能情况排查:

1. test bench 没有成功加入，导致读不到测试文件。
2. 测试代码有问题，测试代码没有起到测试的作用。
3. 测试文件没有保存

反向排查之后发现是测试文件改过之后没有保存

在仿真的过程中我还出现了代码写错的情况,根据仿真测试结果的提示发现是自己在赋值 2 位二进制数的时候将 `a=2'b10` 错写成了 `a=2b' 10`, 需要更加仔细认真.



程序中用到了数值字串 (literal) "n'Bdd::d", 其中 n 是字串的位数, 用十进制表示, 这里字串的位数是"dd...d" 这个值存放在机器中(二进制)所用的位数, 而不是其用"B" 进制表示的位数。"B" 是指定基数的单个字母, 可以是 b (二进制), o (八进制), d (十进制, 可省略) 和 h (十六进制)。这是此数值字串, 用 "B" 进制表示的值在第一次写完激励代码之后还出现了这样的错误, 后来排查发现是激励文件的 vlg 层次名没有写对, 写成了 exp02, 不符合, 所以就 Analyze 不通过了。

六.实验得到的启示（积极思考）

进行每一步操作的时候需要细心,这样能减少程序的错误,节省很多不必要的时间,同时出错以后要细心检查反向排查,这样才能快速定位错误.同时还要能逐渐熟练 Quartus 软件的使用,比如某个地方出错以后能够利用此软件的提示信息较快的找出错误源,只有这样做实验和查错才会更加迅速.

七.意见和建议等

老实说这次实验还是花了很大功夫, 上网找资料之类, 然后对于之前的知识掌握得不是很牢固, 也是花了很多时间在复习上. 不过老师有给几个例子, 慢慢学还是挺好的.

做实验时出现错误不要慌,要慢慢去寻找,这样能节约很多的时间,同时做实验要看清实验要求,这样也会少花一些功夫.