

电子技术基础实验

实验8 数字设计1

祝尔乐

2020013020

2021/12/12

一.实验目的

- 熟悉 Quartus Prime 开发环境。
- 熟悉用硬件描述语言开发数字系统的方法。
- 熟悉 FPGA 硬件平台。

二.实验内容

1、实现 2 输入 与门/或门。仿真验证，FPGA 板验证。

这部分我们在第2部分一并实现了。

2、实现 2 输入 与门、或门、异或门、与非门和或非门。仿真验证，FPGA 板验证。

设计文件代码如下：

```
module simple_logic(  
    input x1,  
    input x2,  
    output and2,  
    output or2,  
    output xor2,  
    output nand2,  
    output nor2  
);  
    assign and2 = x1 & x2;  
    assign or2 = x1 | x2;  
    assign xor2 = x1 ^ x2;  
    assign nand2 = ~(x1 & x2);  
    assign nor2 = ~(x1 | x2);  
  
endmodule
```

设计仿真文件代码如下：

```
`timescale 1ns/1ps  
module simple_logic_tb;  
  
    reg x1;  
    reg x2;  
  
    wire and2;
```

```

wire or2;
wire xor2;
wire nand2;
wire nor2;

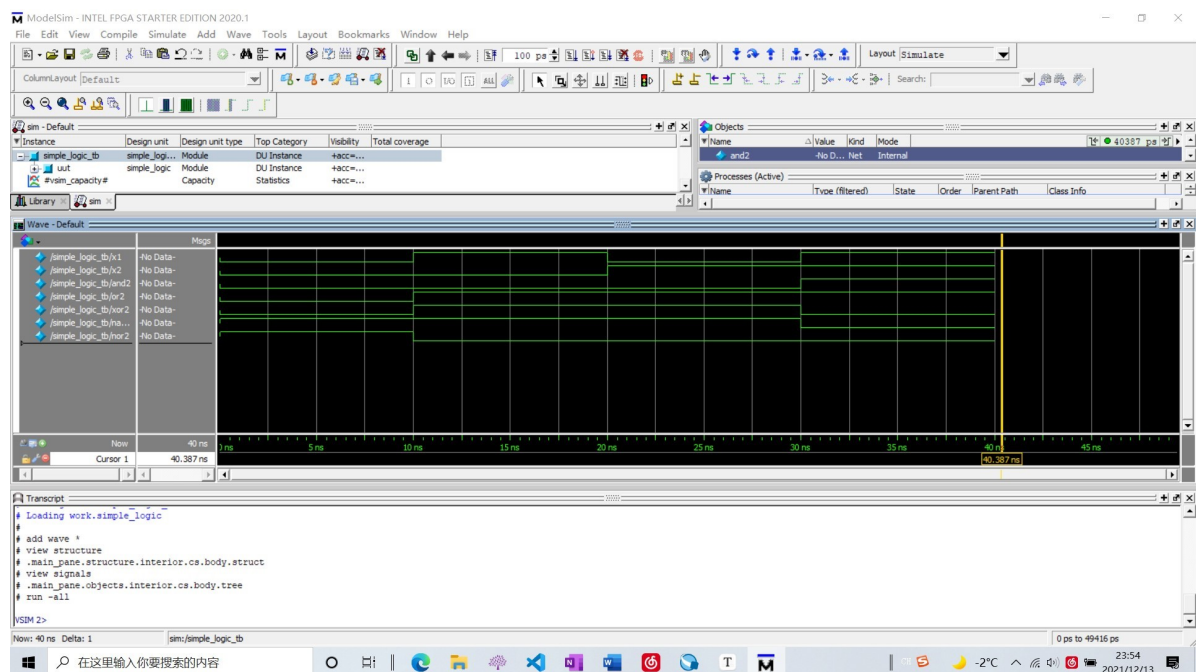
simple_logic uut(
    .x1(x1),
    .x2(x2),
    .and2(and2),
    .or2(or2),
    .xor2(xor2),
    .nand2(nand2),
    .nor2(nor2)
);

initial begin
    x1 = 0; x2 = 0;
    #10;
    x1 = 1; x2 = 0;
    #10;
    x1 = 0; x2 = 1;
    #10;
    x1 = 1; x2 = 1;
    #10;
end

endmodule

```

最终得到仿真波形图：



由仿真结果可以看出，端口and2, or2, xor2, nand2, nor2分别实现了2输入与门、或门、异或门、与非门和或非门。将逻辑程序烧录进入FPGA板子，通过按钮和LED灯再次验证了电路设计的正确性。

3、用 Verilog 实现布尔表达式： $y = \overline{a}\overline{b}\overline{c} + \overline{a}b\overline{c} + a\overline{b}c$ ，仿真实验。

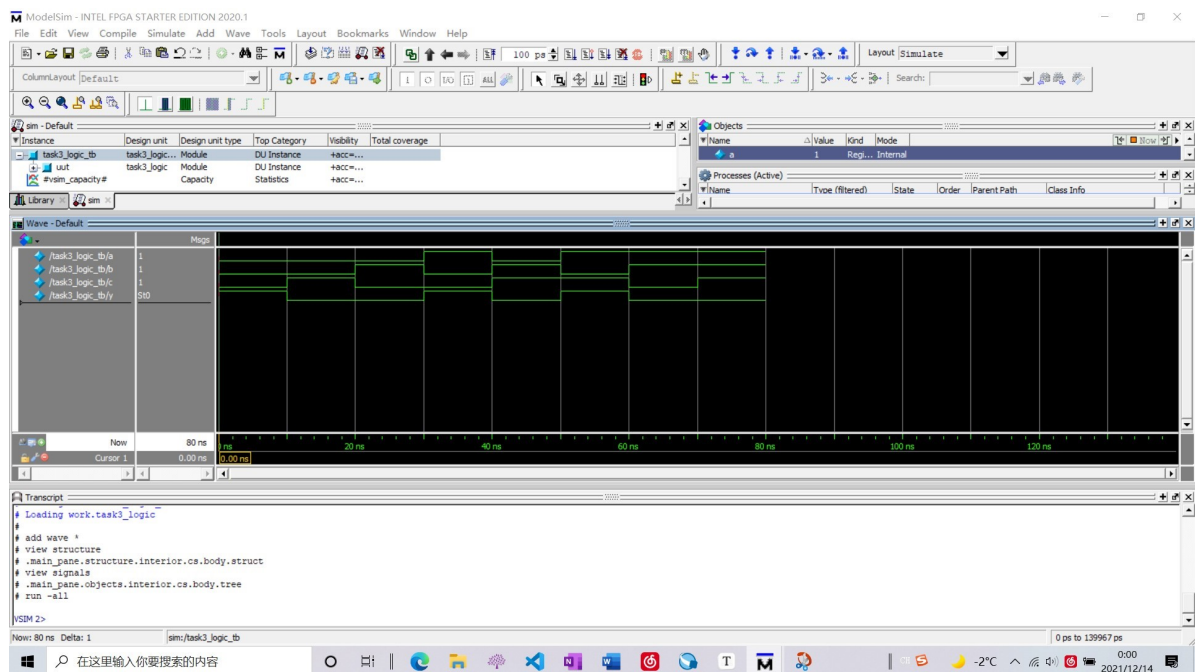
设计文件的代码为：

```
module task3_logic(  
    input a,  
    input b,  
    input c,  
    output y  
);  
  
assign y = (~a & ~b & ~c) | (a & ~b & ~c) | (a & ~b & c);  
  
endmodule
```

测试文件的代码为：

```
`timescale 1ns/1ps  
module task3_logic_tb;  
  
    reg a;  
    reg b;  
    reg c;  
  
    wire y;  
  
    task3_logic uut(  
        .a(a),  
        .b(b),  
        .c(c),  
        .y(y)  
    );  
  
    initial begin  
        a = 0; b = 0; c = 0;  
        #10;  
        a = 0; b = 0; c = 1;  
        #10;  
        a = 0; b = 1; c = 0;  
        #10;  
        a = 1; b = 0; c = 0;  
        #10;  
        a = 0; b = 1; c = 1;  
        #10;  
        a = 1; b = 0; c = 1;  
        #10;  
        a = 1; b = 1; c = 0;  
        #10;  
        a = 1; b = 1; c = 1;  
        #10;  
    end  
  
endmodule
```

得到的仿真波形图为：



做出逻辑表达式的真值表：

a	b	c	y
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	1
0	1	1	0
1	1	0	1
1	0	1	0
1	1	1	0

将仿真结果与真值表进行比较，两者逻辑值一致，验证了设计的正确性。

将逻辑程序烧录进入 FPGA 板子，通过按钮和LED灯再次验证了电路设计的正确性。

4、选做：设计 8 位反相器。仿真验证。

设计文件的代码为：

```
module inv8(input [0:7]x, output [0:7]y);
    assign y = ~x;
endmodule
```

测试文件的代码为：

```

`timescale 1ns/1ps
module inv8_tb;
reg [0:7]x;
wire [0:7]y;

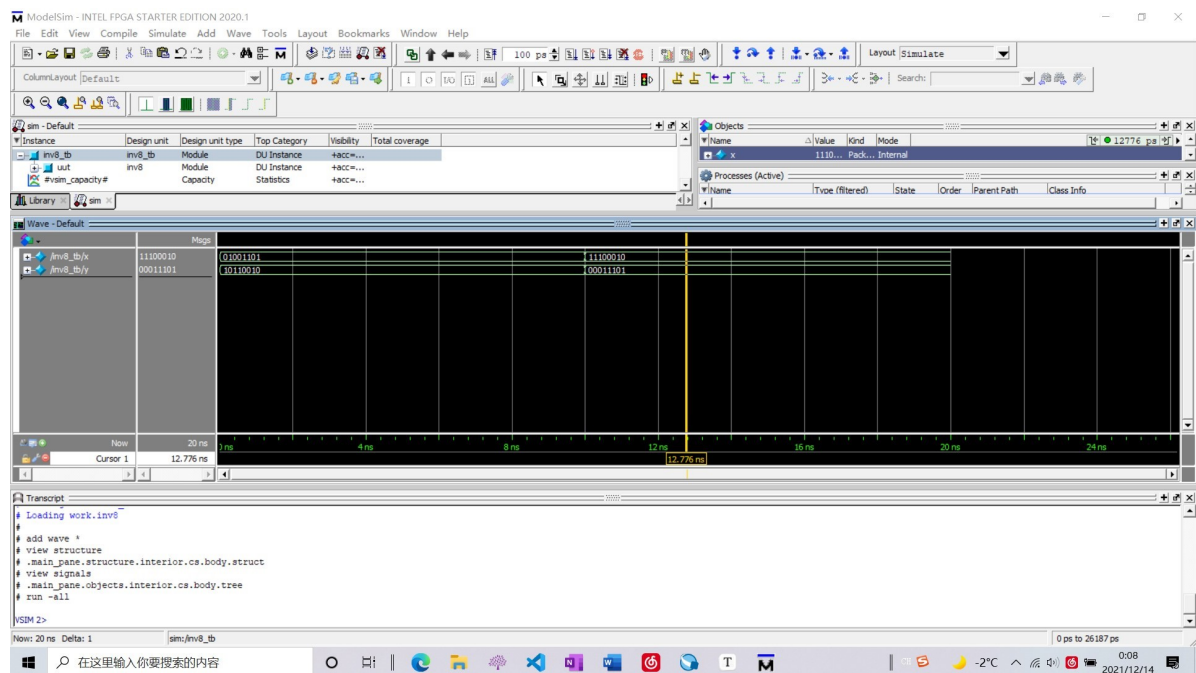
inv8 uut(.x(x),.y(y));

initial begin
x = 8'b01001101;
#10;
end

endmodule

```

得到的仿真波形图为：



这里我们只验证了两个数据点，但由于选取数据的随机性，以及代码逻辑的简单性，我们已经足以验证电路设计的正确性。将逻辑程序烧录进入 FPGA 板子，通过按钮和LED灯再次验证了电路设计的正确性。

5、设计 4 位全加器。仿真验证，FPGA 板验证。

全加器有多种实现方法，以下设计文件代码是最简洁的一种设计方法：

```

module fulladder4(
    input [3:0] x,
    input [3:0] y,
    output [3:0] s,
    output cout
);

assign {cout , s} = x + y;

endmodule

```

测试文件的代码如下：

```

`timescale 1ns/1ps

```

```

module fulladder4_tb;
    reg [3:0]x;
    reg [3:0]y;

    wire [3:0]s;
    wire cout;

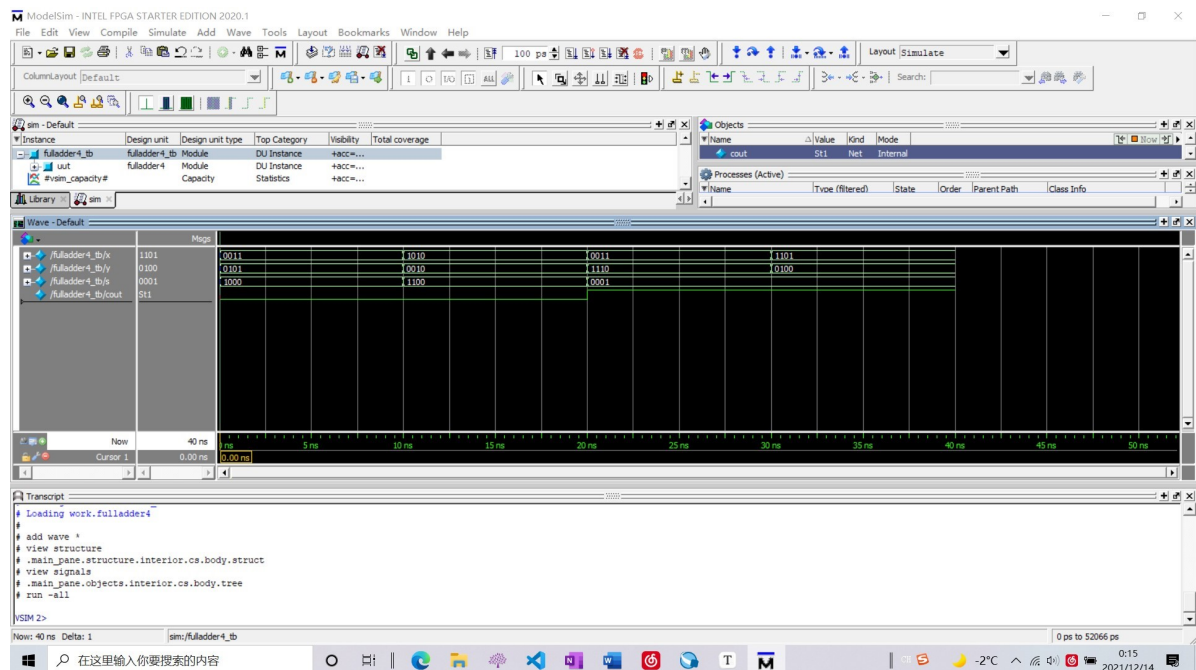
    fulladder4 uut(
        .x(x),
        .y(y),
        .s(s),
        .cout(cout)
    );

    initial begin
        x = 4'b0011;
        y = 4'b0101;
        #10;
        x = 4'b1010;
        y = 4'b0010;
        #10;
        x = 4'b0011;
        y = 4'b1110;
        #10;
        x = 4'b1101;
        y = 4'b0100;
        #10;
    end

endmodule

```

仿真得到的波形为：



通过随机选取的四组数据：

- 0011B+0101B=01000B
- 1010B+0010B=01100B
- 0011B+1110B=10001B

- $1101B + 0100B = 10001B$

我们验证了全加器电路设计电路的正确性。

将逻辑程序烧录进入 FPGA 板子，通过按钮和LED灯再次验证了电路设计的正确性。

6、选做：设计 8 位 4 选 1 多路选择器。仿真验证。

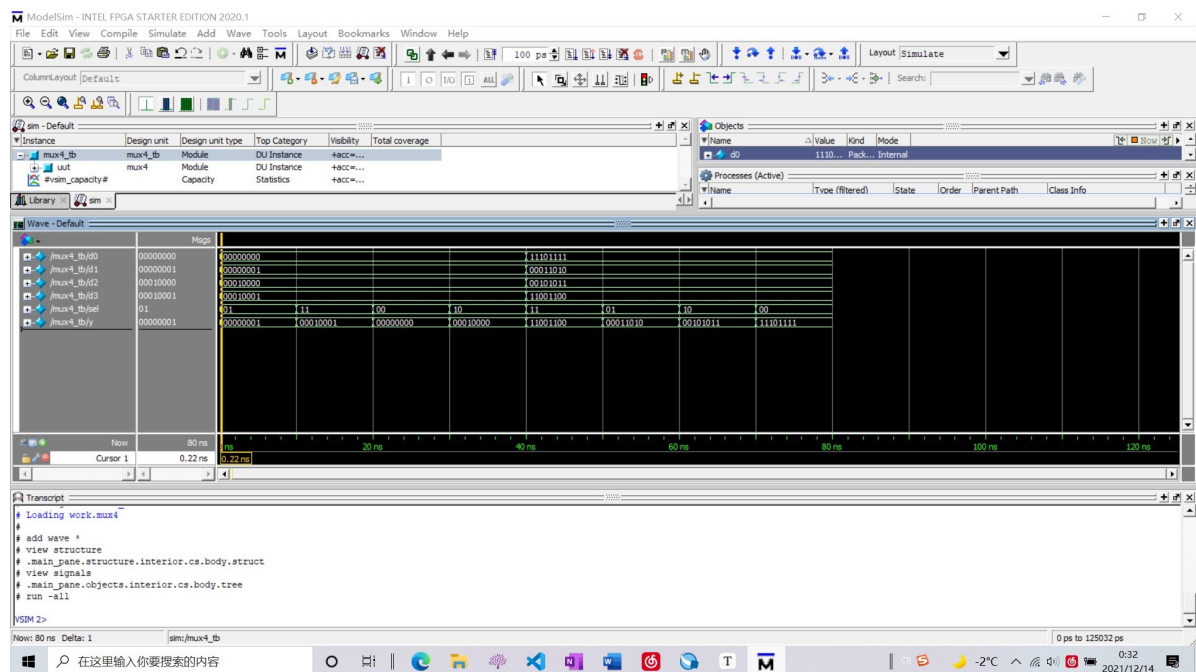
设计文件的代码为：

```
module mux4(  
    input[7:0] d0,d1,d2,d3, //4个4位的输入  
    input[1:0] sel, //2位输入，表示选择的输入  
    output reg [7:0] y //4位输出，和选择的输入一致  
);  
  
always @(*) //敏感信号列表，当d0,d1,d2,d3,select任意一个改变时执行  
    begin  
        case(sel)  
            0: y=d0;  
            1: y=d1;  
            2: y=d2;  
            3: y=d3;  
            default: y=0;  
        endcase  
    end  
endmodule
```

测试文件的代码为：

```
`timescale 1ns / 100ps  
module mux4_tb;  
    reg[7:0] d0;  
    reg[7:0] d1;  
    reg[7:0] d2;  
    reg[7:0] d3;  
    reg[1:0] sel;  
    wire[7:0] y;  
    mux4 uut(.d0(d0),.d1(d1),.d2(d2),.d3(d3),.y(y),.sel(sel));  
  
    initial begin  
        d0 = 8'h00; d1 = 8'h01; d2 = 8'h10; d3 = 8'h11;  
        sel = 1;  
        #10;  
    end  
endmodule
```

仿真得到的波形图为：



可以看到，当`sel`设定为`i`时，输出`y`会设定为通道`i`的值，验证了8位4选1选择器。

由于开关有限，不易用 FPGA 检验，我们主要通过仿真波形检验电路设计的正确性。