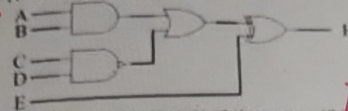


9. 下列哪些 Verilog 的基本门级元件是多输出 ()
 A. nand B. not C. and D. nor
10. 在下列 Verilog HDL 运算符中, 属于三目运算符的是 ()
 A. && B. ! C. ? D. ==

二、填空题 (每空 2 分, 共 20 分)

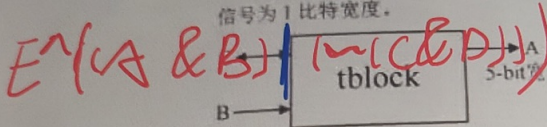
得分

1. 写出表达式以实现对应电路的逻辑功能。



assign F =

2. 根据图中输入输出关系将 Verilog 模块定义补充完整, 其中信号 A 为 5 比特宽度, 信号为 1 比特宽度。



module tblock (A, B, C);

output [4:0] A;
 input B;
 input C;

endmodule

3. IEEE 标准的硬件描述语言是 Verilog 和 VHDL

4. 假定某 4 比特位宽的变量 a 的值为 4'b1011, 计算下列运算表达式的结果

&a = 160

{3{a}} = 12'b101110111011

(a < 4'd3) || (a >= a) = 1

5. 下面两段代码中信号 in, q1, q2 和 q3 的初值分别为 0, 1, 2 和 3, 那么经过 1 个钟周期后, 左侧程序中 q3 的值变成 0, 右侧程序中 q3 的值变成 0

always @(posedge clk)

always @(posedge clk)

begin

q1 <= in;

q2 <= q1;

q3 <= q2;

end

begin

q1 = in;

q2 = q1;

q3 = q2;

end

(4) wire -> 综合时是线

reg 变量在 always 块中

带边沿 带边沿 带边沿

1. reg 型与 wire 型变量的差别是什么，并试说明在什么情况下如何选择两种类型。

• wire 和 reg 的区别

- 1、wire 型数表示的网线类型，相当于物理连线。只能在 assign 左侧赋值，默认初始值是 z。
- 2、reg 型表示的寄存器类型，相当于触发器，存储单元。在 initial 和 always 过程内被赋值的信号。默认初始值是 x。
- 3、reg 寄存器型数据保持最后一次的赋值，而 wire 需要持续的驱动；wire 表示直通，即只要输入有变化，输出马上无条件地反映；reg 表示一定要有触发，输出才会反映输入。
- 4、wire 使用在连续赋值语句中，而 reg 使用在过程赋值语句中。wire 只能被 assign 连续赋值，reg 只能在 initial 和 always 中赋值。模块的输入输出端口类型都默认为 wire 型。
- 5、wire 型的变量综合出来一般是一根导线；reg 变量在 always 块中有两种情况：always 后不带时钟触发的，综合出来是组合逻辑；always 后的敏感表中是带时钟边沿触发的，综合出来一般是时序逻辑。

2. 请详细解析阻塞赋值与非阻塞赋值的区别。

① = , <=

② 在一个 begin end 中，先执行当前，再执行下一行
非：同时赋值

3. FPGA 流程

四、综合题 (共 46 分)

1. 试设计一个 3/8 译码器。规定模块定义为 module Decoder(Out, In, En), 其中 Out 为译码器输出, In 为译码器输入, En 为译码使能输入。要求: 写出 3/8 译码器 Verilog HDL 设计程序。(10 分)

$A_2, A_1, A_0 \quad Y_7 \dots Y_0$
 $0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0$
 module Decoder(Out, In, En);
 input [2:0] In;
 input En;
 output reg [7:0] Out;
 always @(In) begin
 if (!En) begin
 out = 8'b11111111;

end else begin

case (In)

3'b000 : Out = 8'b11111110;

3'b011 : Out = 8'b01111111;

end case

end

endmodule

写出下列表达式运行结果。(11 分)

wire [3:0] A = 4'b0101;
 wire [3:0] B = 4'b1001;
 wire [3:0] C = 4'b01z;
 wire signed [3:0] D = 4'b1001;

(1) ~A 运行结果为 (1 分)

移位运算符

- 算术右移将操作数右移n位, 并且在左边空出来的位置补n位操作数的符号;
- 逻辑右移将操作数右移n位, 并且在左边空出来的位置补n位0, 忽略操作数的符号。
- 算术左移和逻辑左移相同, 都是在以为后右边空出的位置补0, 忽略操作数的符号

操作符	功能
<<	逻辑左移
>>	逻辑右移
<<<	算数左移
>>>	算数右移

(2) A & B 运行结果为 (1 分)

(3) A && B 运行结果为 (1 分)

(4) A ^ B 运行结果为 (1 分)

(5) B & C 运行结果为 (1 分)

(6) ~C 运行结果为 (1 分)

(7) B && C 运行结果为 (1 分)

(8) A || C 运行结果为 (1 分)

(9) B >> 2 运行结果为 (1 分)

(10) D >> 2 运行结果为 (1 分)

(11) B >>> 2 运行结果为 (1 分)

3. 设计一个带复位端且对输入时钟 clk 进行二分频模块, 并画出仿真波形。(12 分)

设计示例

例：（分频器）

通过变换得到所需的各种频率成分。
常用的偶数分频、奇数分频。

代码、图象

带复位的二分频电路：



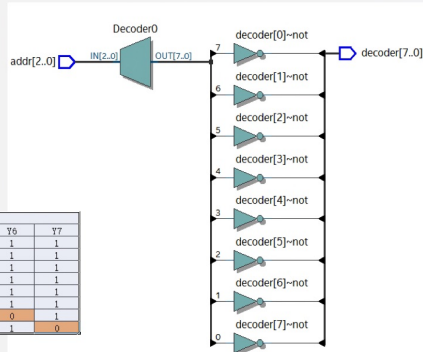
```
module div_2(clk,rst,out);
input clk,rst;
output out;
reg q;
always@(posedge clk or negedge
rst)
if(!rst)
q<=1'b0;
else
q<=~q;
assign out=q;
endmodule
```

设计示例

例：（三八译码电路）

有效电平可以是高电平(称为高电平译码),也可以是低电平(称为低电平译码)。以低电平译码为例:

输入			输出							
A2	A1	A0	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	0	1	1
1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	0



```
module top(addr, decoder) ;
input [2:0] addr ;
output reg [7:0] decoder ;
```

```
always @(addr) begin
```

```
case(addr)
```

```
3'b000 : decoder =8'b1111_1110 ;
3'b001 : decoder =8'b1111_1101 ;
3'b010 : decoder =8'b1111_1011 ;
3'b011 : decoder =8'b1111_0111 ;
3'b100 : decoder =8'b1110_1111 ;
3'b101 : decoder =8'b1101_1111 ;
3'b110 : decoder =8'b1011_1111 ;
3'b111 : decoder =8'b0111_1111 ;
```

```
endcase
```

```
end
```

```
endmodule
```



如何带复位？

安徽大学 2022-2023 学年第 2 学期
《FPGA 数字系统设计》期末考试试卷(A 卷)

年级_____专业_____姓名_____学号_____座位号_____

大项	一	二	三	四	总分	阅卷人
得分						

一、单项选择 (在备选答案中选出一个正确答案, 并将其号码填在题干后的括号内。每题 2 分, 共 20 分)

- 下列标识符中, () 是不合法的标识符。
A. system1 B. _reg C. Slatch D. exec\$
- 下面运算符, 优先级最高的是 ()
A. << B. * C. ~ D. +
- 在 verilog 语言中整型数据 integer 与 () 位寄存器数据在实际意义上是相同的。
A. 8 B. 16 C. 32 D. 64
- 关于 Verilog 中操作符描述错误的是 ()
A. << 的优先级大于 <
B. 条件操作符优先于拼接操作符
C. >>> 逻辑右移, 左边补 0
D. 使用“=”比较二值逻辑, 如果出现 X 或者 Z, 则结果为 X
- 下列关于阻塞和非阻塞赋值说法不正确的是 ()
A. 阻塞赋值属于顺序执行, 即下一条语句执行前, 当前语句一定会执行完毕。
B. 非阻塞赋值属于并行执行语句, 即下一条语句的执行和当前语句的执行是同时进行的。
C. 赋值操作符是“=”的过程是阻塞性过程赋值
D. 一个过程结构中混合使用阻塞赋值与非阻塞赋值
- 时间尺度定义为 timescale 10ns/100ps, 下列说法正确的是 ()
A. 时间精度 10ns B. 时间单位 100ps
C. 时间精度 100ps D. 时间精度不确定
- 下列语句中不属于并行语句的是 ()。
A. 过程语句 B. case 语句 C. assign 语句 D. 元件例化语句
- 在 verilog 语言中, a=4b'1011, 那么 &a=()
A. 4b'1011 B. 4b'1111 C. 1b'1 D. 1b'0

9、reg[3:0] A; 执行 A=2'h0F; 那么最后变量 A 的值是 ()
 A 4'b0011 B 4'h0F C 4'b1111 D 4'b0000

10、在 Verilog HDL 的逻辑运算中, 设 A=8'b11010001, B=8'b00011001, 则表达式“A&B”的结果为 ()

A.8'b00010001 B.8'b11011001 C.8'b11001000 D.8'b00110111

二、填空题: (请在横线上填写答案。每空 2 分, 共 30 分)

1、变量是在程序运行过程中其值可以改变的量。变量分为两种, 一种类型为_____类型, 一般指示硬件电路的物理连接, 另一种是_____对应的是具有状态保持作用的存储元件。

2、Verilog 的端口具有以下三种类型: input , _____, _____。

3、IEEE 标准的硬件描述语言有_____和_____两种。

4、下面两段代码中信号 in, q1, q2 和 q3 的初值分别为 0, 1, 2 和 3, 那么经过 1 个时钟周期后, 左侧程序中 q3 的值变成_____, 右侧程序中 q3 的值变成_____。

```
always @(posedge clk)
begin
```

```
q1 = in;
```

```
q2 = q1;
```

```
q3 = q2;
```

```
end
```

```
always @(posedge clk)
```

```
begin
```

```
q1 <= in;
```

```
q2 <= q1;
```

```
q3 <= q2;
```

```
end
```

5、请写出下列表达式运行结果 (请用二进制数表示)。

```
wire [3:0] A = 4'b0101;
```

```
wire [3:0] B = 4'b1001;
```

```
wire [3:0] C = 4'bx010;
```

```
wire signed[3:0] D = 4'b1001;
```

(1) A & B 运行结果为_____;

(2) A && B 运行结果为_____;

(3) B & C 运行结果为_____;

(4) B&&C 运行结果为_____;

(5) B>>2 运行结果为_____;

(6) D>>2 运行结果为_____;

(7) B>>>2 运行结果为_____。

三、简答题（请在答题卡填写答案。共 20 分）

- 1、简述有限状态机 FSM 分为哪两类？有何区别？有限状态机的状态编码主要有哪三种？
- 2、简述 Verilog 电路模型三种抽象级别的特点：结构描述，行为描述，数据流描述。

四、程序设计题（请在答题卡填写答案。共 30 分）

- 1、编写四选一数据选择器程序，并进行仿真测试。（10 分）

（1）通过选择信号，选择不同的输入信号输出到输出端，如下图真值表，sel[1:0]为选择信号，a,b,c,d 为输入信号，Mux 为输出信号。

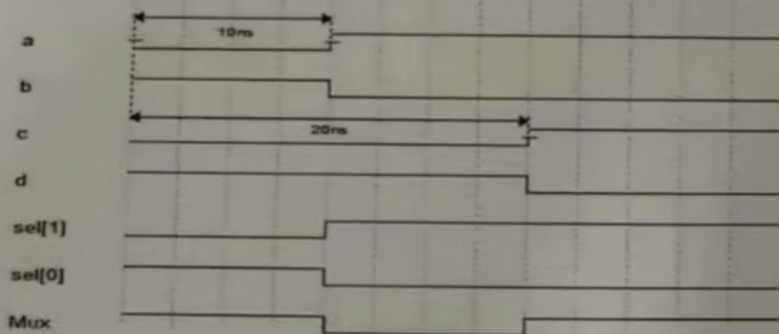
选择信号		输入				输出
sel[1]	sel[0]	a	b	c	d	Mux
0	0	x	x	x	x	a
0	1	x	x	x	x	b
1	0	x	x	x	x	c
1	1	x	x	x	x	d

```
module top_mux(a, b, c, d, sel, Mux);
```

（请补充完整）

```
endmodule
```

- （2）根据如下时序仿真图，完成仿真文件的编写。



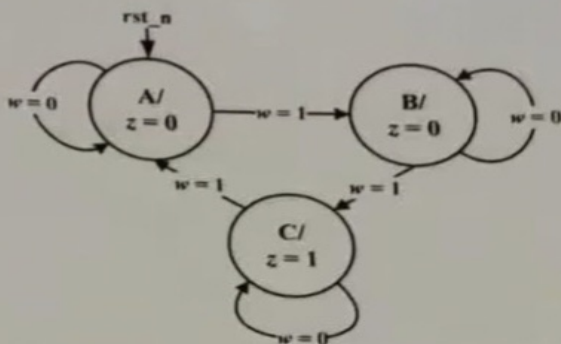
```
'timescale 1ns/1ns
```

```
module tb_top_mux;
```

（请补充完整）

```
endmodule
```


2、根据下图的状态转移关系图，编写状态机程序。其中 A、B、C 是状态，z 是输出，w 是状态转移条件，rst_n 是复位。（10 分）



```

module my_fsm(
    input clk,
    input rst_n,
    input w,
    output reg z
);

```

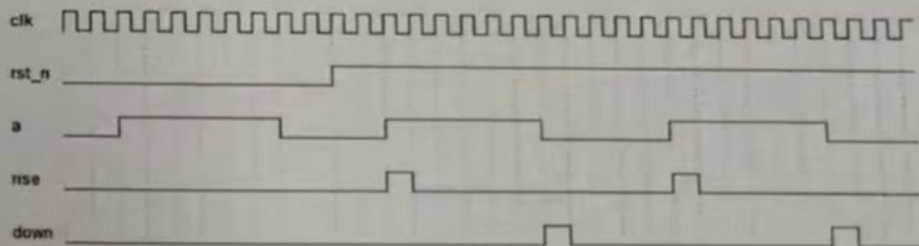
（请补充完整）

```

endmodule

```

3、如下图所示，有一个周期脉冲信号 a，编写一个程序检测 a 信号的上升沿，给出指示信号 rise，当 a 信号出现下降沿时给出指示信号 down；复位时不工作；rise,down 应为单脉冲信号，在相应边沿出现时输出为高，之后恢复到 0，一直到再一次出现相应的边沿。请将程序补充完整。（10 分）



```

module edge_detect(input clk, input rst_n, input a, output reg rise, output reg down);

```

（请补充完整）

```

endmodule

```

