

# 大 连 理 工 大 学

学号: \_\_\_\_\_

姓名: \_\_\_\_\_

院系: 微电子学院

专业: \_\_\_\_\_

2017 级 \_\_\_\_\_ 班

注: 所有代码仅限用

Verilog HDL 编写

装

每张答题纸都写上

专业、

学号、姓名

订

课程名称: FPGA 设计及应用 试卷: A 考试形式: 闭卷 试卷共 4 页

授课院 (系): 微电子学院 考试日期: 2020 年 8 月 24 日

题号	一	二	三	四	五	六	七	八	总分
满分	10	10	5	5	10	15	25	20	100
得分									

一. (10 分) Verilog HDL 描述 4bit 全加器 (5 分)。设计对应的 Test Bench 模块, 用随机数函数产生激励, 将期望结果与被测模块实际输出做比对, 若不相等, 用 \$display 函数输出错误信息 (包括输入值、期望结果、实际结果, 以及必要的文字信息)。(5 分)

二. (10 分) 设计 4 选 1 数据选择器, 数据输入端口为 in0~in3, 数据输出为 out。选择信号 sel 值为 0~3 对应选择 in0~in3。输出使能端 oe 高电平时, out 输出所选的信号; oe 为低电平时, out 输出高阻。补齐下方框架内正确的 module。

```
module mux4ch(out, in0, in1, in2, in3, sel, oe);  
input in0, in1, in2, in3;   input [1:0]sel;   input oe;  
output out;
```

//补齐此处的代码 (答题纸上仅写缺失的代码段即可)

```
endmodule
```

三. (5 分) 设计一个分频器, 输入时钟 sysclk 为 50MHz, 分频输出端 lfclk 频率 1Hz。在下面框架内补全正确的代码。

```
module divider(lfclk, sysclk, rst);  
output lfclk;   input sysclk, rst;
```

//补齐此处的代码 (答题纸上仅写缺失的代码段即可)

```
endmodule
```

四. (5 分) 设计 **4bit** 电平敏感锁存器。锁存输入 **le** 为 1, 输出 **out** 随输入 **in** 变化; **le** 为 0 则输出不变。使能端 **oe** 为 1 输出数据, **oe** 为 0 输出高阻, **oe** 优先级高。

五. (10 分) 设计边沿检测器, 检测信号 **rx** 的上升沿和下降沿, 采样时钟 **sysclk** 为 10MHz, **rx** 的频率低于 1MHz。检测到上升沿, **re** 输出 1, **fe** 输出 0; 检测到下降沿, **re** 输出 0, **fe** 输出 1; 未检测到边沿时, **re**、**fe** 均输出 0。**re**、**fe** 输出高电平宽度为 1 个 **sysclk** 周期。**rst** 是复位信号。在下方的框架内补齐正确代码。

```
module edgedetect(re, fe, rx, sysclk, rst);  
output re, fe;    input rx, sysclk, rst;
```

//补齐此处的代码 (答题纸上仅写缺失的代码段即可)

```
endmodule
```

## 六. 简答题 (共 15 分)

1. (3 分) 阻塞赋值、非阻塞赋值有什么区别? 设计时序逻辑电路时是否可以用阻塞赋值?
2. (3 分) 用门控 D 触发器时钟信号的方式实现带有使能端的 D 触发器有什么缺点? 正确的实现方式应该是什么?
3. (3 分) Xilinx 公司 ZYNQ 系列可编程器件与此前传统 FPGA 的架构有什么主要异同?
4. (3 分) FPGA 上的软核、硬核嵌入式处理器在实现方式和性能上有何区别? FPGA 上实现的 NiosII、Microblaze、PowerPC、ARM 处理器分别属于软核还是硬核?
5. (3 分) “自上而下”与“自下而上”的设计方法各有什么特征? 复杂数字系统设计适合采用哪一种?

七. 填空题 (25 分) 在答题纸上写清题号和正确输出结果

1. module ArithTest;  
reg [3:0] a, b, c, d;  
initial begin  
a=4'b1001; b=4'b1101; c=5; d=4'hx;  
\$display(7 % -3); // \_\_\_\_\_ (1)  
\$display(a < b); // \_\_\_\_\_ (2)  
\$display(d <= 4'b1111); // \_\_\_\_\_ (3)  
\$display**b**(d); // \_\_\_\_\_ (4)  
\$display(a + b); // \_\_\_\_\_ (5)  
\$display(a - b); // \_\_\_\_\_ (6)  
\$display(a \* b); // \_\_\_\_\_ (7)  
\$display(a / b); // \_\_\_\_\_ (8)  
end  
endmodule
2. module BitTest;  
reg [3:0] a, b, c, d, e;  
initial begin  
a=4'b1101; b=4'b0101; c=4'b1001; d = 4'b1xx0; e = 4'b0;  
\$displayb(~a); // \_\_\_\_\_ (9)  
\$displayb(a&c); // \_\_\_\_\_ (10)  
\$displayb(b^c); // \_\_\_\_\_ (11)  
\$displayb(a~^c); // \_\_\_\_\_ (12)  
\$displayb(b<<1); // \_\_\_\_\_ (13)  
\$displayb(b>>2); // \_\_\_\_\_ (14)  
\$displayb(4'bx<<2); // \_\_\_\_\_ (15)  
\$displayb(4'b1101<<2); // \_\_\_\_\_ (16)  
\$displayb({a, b}); // \_\_\_\_\_ (17)  
\$displayb({c[3:1], a}); // \_\_\_\_\_ (18)  
\$displayb(4'b1x); // \_\_\_\_\_ (19)  
\$displayb(&a); // \_\_\_\_\_ (20)  
\$displayb(|b); // \_\_\_\_\_ (21)  
\$displayb(^b); // \_\_\_\_\_ (22)  
\$displayb(&c); // \_\_\_\_\_ (23)  
\$displayb(4'b0x1z); // \_\_\_\_\_ (24)  
\$displayb(d&e); // \_\_\_\_\_ (25)  
end  
endmodule

八. (20 分) 用 Verilog HDL 设计一个序列检测器有限状态机，检测到输入序列中存在“100”序列，输出 1 个时钟长度的 1，否则输出 0。检测到存在“010”序列则停止检测，直到再次复位（高电平同步复位）。(1) 在答题纸上画完整的状态图 (10 分)；  
(2) 在答题纸上写出对应的完整 module (10 分)。

```
module seq_detector(clk, reset, din, out);  
  input clk, reset, din;    output out;  
  reg [2:0] state, nextstate;  
  parameter ST0=0, ST1=1, ST2=2, ST3=3,  
            ST4=4, ST5=5, ST6=6;
```

