

2019级《数字系统设计》测验3 (2021年6月24日)

专业_____ 学号_____ 姓名_____ 得分_____

一、(40分) Verilog HDL

(1) 仔细阅读下列 Verilog 代码描述的模块, 指出这段代码存在的错误, 并进行改正。

(注: 假设所描述的行为是正确的。)(5分)

```
module divide2 (clk, clk_o, reset)
input clk, reset;
output clk_o;
wire in;
wire out;
always (posedge clk or posedge reset)
if (reset)
out <= 0;
else
out <= in;
assign in <= ~out;
assign clk_o = out;
```

①第13行尾加';'
②第5行'wire'改为'reg'
③第6行always后加'@'
④第11行'<='改为'='
⑤第12行后第13行加'endmodule'

错1扣1分, 扣完

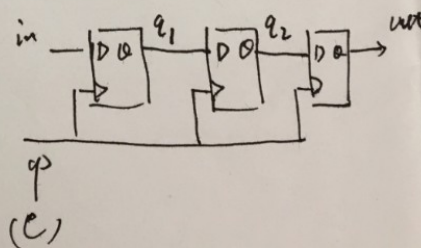
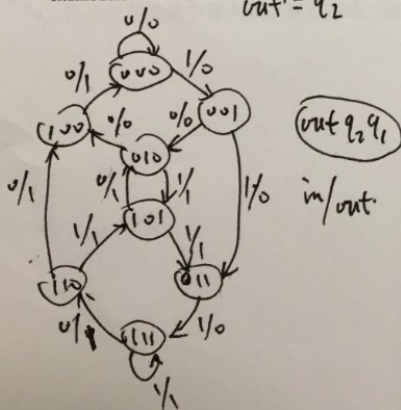
(2) 以下是 Verilog 描述的一个电路, 请使用适当的门电路和 D 触发器, 画出对应的逻辑电路图, 要求电路最简化。并画出该电路的状态转换图。(10分)

```
module nonblocking(in, c, out);
input in, c;
output out;
reg q1, q2, out;
always @ (posedge c)
begin
q1 <= in;
q2 <= q1;
out <= q2;
end
endmodule
```

5+5
2分. 状态表
in q2 q1 q1' q2' out'(q2)

0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	1
1	0	0	1	0	1
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

q1' = in
q2' = q1
out' = q2



二 (60分) 一个简单的 ADC 控制算法分为以下四步: (1) 启动 AD 设备; (2) 触发 AD 数据; (3) 获取 AD 的状态, 若未结束则继续转换; (4) 停止 AD 设备。请设计实现 ADC 的控制器。

- 1) 写出控制器的状态表, 画出控制器的 ASM 流程图;
- 2) 请用 D 触发器及门电路设计控制器, 写出触发器激励输入函数及输出函数;
- 3) 请设计 MUX 型控制器, 画出电路图;
- 4) 如采用微程序型控制器, 请设计微指令格式, 并加以说明;
- 5) 请写出对应这种微指令格式的控制器微程序代码。

二. 设计

设启动AD设备的控制信号为 $A=1$, 停止时 $A=0$

触发AD数据采样的控制信号为 B .

获取AD状态的控制信号为 C .

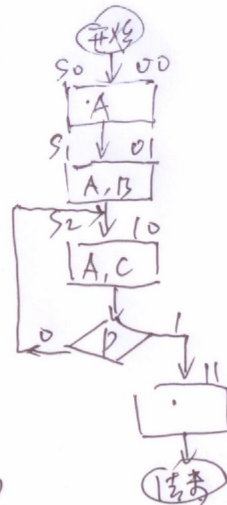
得到的状态信息 P , $P = \begin{cases} 0 & \text{未结束} \\ 1 & \text{结束} \end{cases}$

四丁状态(步骤)用二进制触发器 Q_1, Q_0 . 并设 $Q_1 Q_0 = \begin{cases} 00 & \text{启动AD} \\ 01 & \text{能在AD} \\ 10 & \text{获取地址} \\ 11 & \text{结束} \end{cases}$

1) 状态表

$Q_1 Q_0$	$A \neq 0$		标志 $A B C$	
	$P=0$	$P=1$	$P=0$	$P=1$
00	01	01	100	100
01	10	10	110	110
10	10	11	101	101
11	11	11	000	000

ASM 流程图



注: 如果 A 在四状态下才有, 也并时。

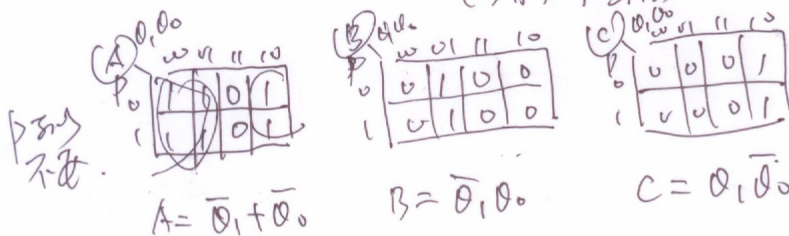
2)



$$Q_1' = D_1 = Q_1 + Q_0$$

$$Q_0' = D_0 = \bar{Q}_1 \bar{Q}_0 + Q_1 Q_0 + Q_1 P$$

(万有不同图法)



$$A = \bar{Q}_1 + \bar{Q}_0$$

$$B = \bar{Q}_1 Q_0$$

$$C = Q_1 \bar{Q}_0$$

3) 地址 (见上面 D_i 表) 5) 微程序线 20.

4) 微指令格式

$A B C$	$P(1)$	$UA(1) UA(0)$
100	0	01
110	0	10
101	1	10
000	0	11

← 微命令 → 判判 → T地址 →

6 bit

地址	微指令		
	$A B C$	$P(1)$	$UA(1) UA(0)$
00	100	0	01
01	110	0	10
10	101	1	10
11	000	0	11

S_2 时, 缺省下地址:

$$UA(1) UA(0) = 10,$$

当 $P(1) = 1$ 时,

仍设地址:

$$UA(0) = P(1) \cdot P \cdot T_2$$

缺省地址仍设在

地址同数在 T_2 时刻

