

西安电子科技大学

考试时间 120 分钟

试 题

| 题号 | 一 | 二 | 三 | 四 | 五 | 总分 |
|----|---|---|---|---|---|----|
| 分数 | | | | | | |

1. 考试形式：闭卷 ☒ 开卷 ☐ ； 2. 本试卷共五大题，满分 100 分；
3. 考试日期：2022 年 月 日；（答题内容请写在装订线外）

一、 单项选择题（在每小题的四个备选答案中选出一个正确的答案，将其序号填写在下面表格中，未填入表格无分。每小题 2 分，共 20 分）

| | | | | |
|----|----|----|----|-----|
| 1. | 2. | 3. | 4. | 5. |
| 6. | 7. | 8. | 9. | 10. |

1. 原码加减交替除法又称为不恢复余数法，因此_____。

[A]

- A. 不存在恢复余数的操作
B. 当某一步运算不够减时，做恢复余数的操作
C. 仅当最后一步余数为负时，做恢复余数的操作
D. 当某一步余数为负时，做恢复余数的操作

2. 在微程序控制器设计中，通过_____发来的指令操作码确定机器指令在控制存储器的入口地址。

[D]

- A. 程序指针 B. 后续地址
C. 程序存储器 D. 指令寄存器

3. 指令周期是指_____。

C

- A. CPU 从主存取出一条指令的时间
B. CPU 执行一条指令的时间
C. CPU 从主存取出一条指令加上执行这条指令的时间
D. 时钟周期时间

4. 在 VHDL 中，PROCESS 结构内部是由_____语句组成的。

[A]

- A. 顺序 B. 顺序和并行
C. 并行 D. 任何

5. 根据运算器内部总线与构成运算器的基本部件的连接情况运算器分为 3 种结构，以下不属于 ALU 结构的是_____。 [C]

- A. 单总线 B. 双总线
C. 多总线 D. 三总线

6. 本课程实验中所要设计的 SoC 程序存储器容量是_____。 [A]

- A. 4KB B. 256B
C. 2KB D. 128KB

7. 加法器采用先行进位的目的是_____。 [C]

- A. 优化加法器的结构 B. 节省器材
C. 加速传递进位信号 D. 增强加法器

8. 下面不属于 SoC 类型的是_____。 [A]

- A. 嵌入式型 B. 通信网络型
C. 计算控制型 D. 信号处理型

9. 不属于顺序语句的是_____。 [C]

- A. IF 语句 B. LOOP 语句
C. PROCESS 语句 D. CASE 语句

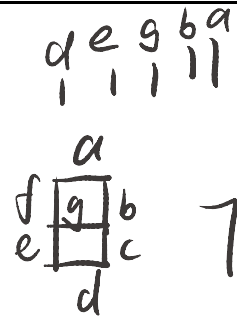
10. 在加减交替除法器设计过程中，当余数大于零时，商为_____，并且_____除数。 [A]

- A. 1，减 B. 1，加
C. 0，加 D. 0，减

二、VHDL 程序分析题（每小题 10 分，共 20 分）

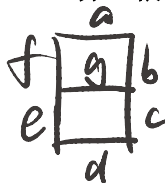
1. 根据以下程序的分析，回答下列问题。

| 行号 | 程序 1 代码 |
|----|---|
| 1 | library IEEE; |
| 2 | use IEEE.STD_LOGIC_1164.ALL; |
| 3 | entity bcd_7seg is |
| 4 | port(bcd_led : in std_logic_vector(3 downto 0); |
| 5 | ledseg : out std_logic_vector(6 downto 0)); |
| 6 | end bcd_7seg; |
| 7 | architecture behavior of bcd_7seg is |
| 8 | begin |
| 9 | process(bcd_led) |
| 10 | <u>① begin</u> ; |
| 11 | if bcd_led="0000" then ledseg<="0111111"; |
| 12 | elsif bcd_led="0001" then ledseg<="0000110"; |
| 13 | elsif bcd_led="0010" then ledseg<= <u>② "1011011"</u> ; |
| 14 | elsif bcd_led="0011" then ledseg<="1001111"; |



| | |
|----|---|
| 15 | elsif bcd_led="0100" then ledseg<= ③ "1100110"; |
| 16 | elsif bcd_led="0101" then ledseg<="1101101"; |
| 17 | elsif bcd_led="0110" then ledseg<="1111101"; |
| 18 | elsif bcd_led="0111" then ledseg<= ④ "1000111"; |
| 19 | elsif bcd_led="1000" then ledseg<="1111111"; |
| 20 | elsif bcd_led="1001" then ledseg<="1101111"; |
| 21 | else ledseg<= ⑤ "1111111"; |
| 22 | end if; |
| 23 | end process; |
| 24 | end behavior; |
| 25 | |

(1) (2分) 根据程序 1 说明所实现的 7 段数码管编码 abcdefg 的顺序是什么?



ledseg [6:0]

(2) (10分) 请补充完成①②③④⑤处所缺少的代码。

2. (8分) 下面程序完成的是 60 进制减计数器功能，请补充完整缺少的代码。

| 行号 | 程序 2 代码 |
|----|--|
| 1 | library ieee; |
| 2 | use ieee.std_logic_1164.all; |
| 3 | use ieee.std_logic_unsigned.all; |
| 4 | entity count is |
| 5 | port(clk: in std_logic; |
| 6 | high,low: out std_logic_vector(3 downto 0)); |
| 7 | end count; |
| 8 | architecture bhv of count is |
| 9 | begin |
| 10 | process(clk) |
| 11 | variable hhigh,tlow: std_logic_vector(3 downto 0); |
| 12 | begin |
| 13 | if clk'event and clk='1' then |
| 14 | if tlow=0 and hhigh=0 then |
| 15 | hhigh:="0101"; |
| 16 | tlow := ⑥ "1001"; |
| 17 | elsif tlow=0 then |
| 18 | tlow:="1001"; |
| 19 | hhigh:= ⑦ hhigh-1; |
| 20 | else |
| 21 | tlow:= ⑧ tlow-1; |
| 22 | end if; |
| 23 | |

| | |
|----|---------------------|
| 24 | end if; |
| 25 | high<=hhigh; |
| 26 | low<= ⑨ tlow _____; |
| 27 | end process; |
| 28 | end bhv; |

装

三、设计分析题（本题共 20 分）

1. (10 分) 信号的时序关系对硬件逻辑电路的功能具有重要的影响。有限状态机 (FSM) 是一种控制时序的有效方法。

(1) (5 分) FSM 一般由哪三部分组成？它们各自的功能是什么？

- ① 主控时序逻辑部分
- ② 主控组合逻辑部分
- ③ 辅助逻辑部分

- 主控时序逻辑部分：任务是负责状态机运转和在外时钟驱动下实现内部状态转换的过程。
- 主控组合逻辑部分：任务是根据状态机外部输入的状态控制信号（包括来自外部的和状态机内部的非进程的信号）和当前的状态值 Current state 来确定下一状态 next state 的取值内容，以及对外部或对内部其他进程输出控制信号的内容。
- 辅助逻辑部分：辅助逻辑部分主要是用于配合状态机的主控组合逻辑和主控时序逻辑进行工作，以完善和提高系统的性能。

订

(2) (5 分) FSM 按照输出方式分为哪两种？它们有哪些区别？

Mealy 状态机的输出是现在状态和所有输入的函数，输出随输入变化而随时发生变化，属于异步输出的状态机。

输出不依赖系统时钟，也不存在摩尔状态机中输出滞后一个时钟周期来反映输入变化问题。

摩尔型状态机输出只与当前状态有关，而与输入信号的当前值无关，是严格的现态函数。

线

2. 控制器是 CPU 重要部件，负责产生 CPU 各个功能部件的控制信号。根据不同的设计实现方法，CPU 控制器可以分为哪两种基本类型，两者有哪些区别？请画出微程序控制器的基本结构。

① 硬布线控制器

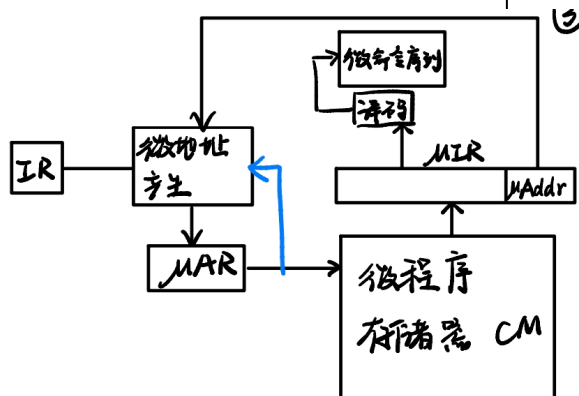
② 微程序控制器

硬布线控制器

- 将控制部件看做产生专门固定时序控制信号的逻辑电路，以使用最少的元件和取得最高操作速度作为设计目标。
- 缺点：设计不规整，并且不易修改或扩展。

微程序控制器

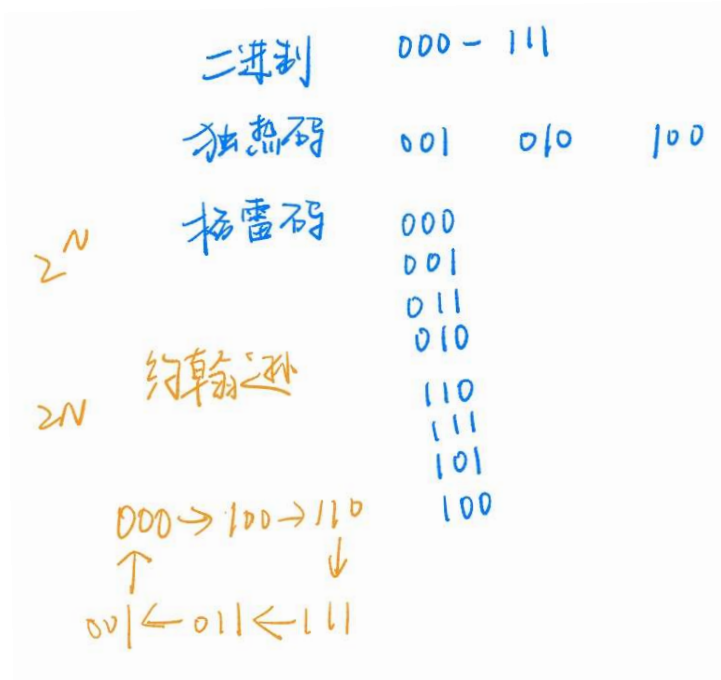
- 将机器指令的操作（从取指令到执行）分解为若干更基本的微操作序列，并将有关的控制信息（微命令）以微码形式编成微指令输入控制存储器中。
- 优点：设计规则，方便修改及功能扩展。



四、时钟模块设计（共 20 分）

时钟模块是 CPU 设计的重要组成部分。计数器是时钟分频设计的主要模块。

1. （10 分）请问计数器可采用哪几种编码方式进行计数？并以 3 位二进制数的为例，写出不同编码方式对应的编码。



2. （10 分）请使用硬件描述语言设计一个时钟模块，该模块的复位方式为异步复位，并且输出时钟信号频率与输入时钟信号频率满足如下关系：clk_out 频率为 clk 频率的五分之一；并且 clk_out 的输出占空比为 50%。

```
entity clk_div3 is
    Port ( clk_in : in  STD_LOGIC;
           clk_out : out STD_LOGIC);
end clk_div3;
architecture Behavioral of clk_div3 is
    signal cnt1,cnt2:integer range 0 to 6;
    signal clk_1,clk_2:std_logic;
begin
    process(clk_in)
    begin
        if(rising_edge(clk_in))then
            if (cnt1<6)then
                cnt1<=cnt1+1;
            else cnt1<=0;
            end if;
            if(cnt1 < 3)then
                clk_1<='1';
            else
                clk_1<='0';
            end if;
        end if;
    end process;

    process(clk_in)
    begin
        if(falling_edge(clk_in))then
            if (cnt2<6)then
                cnt2<=cnt2+1;
            else cnt2<=0;
            end if;
            if(cnt2<3)
                then clk_2<='1';
            else
                clk_2<='0';
            end if;
        end if;
    end process;
    clk_out<=clk_1 or clk_2;
end Behavioral;
```


五、通用寄存器设计（共 20 分）

1. (5 分) 处理器通过寄存器完成数据传送及运算操作。请问寄存器和数据存储器在数据访问和存储上有什么区别？



访问速度 快, 慢

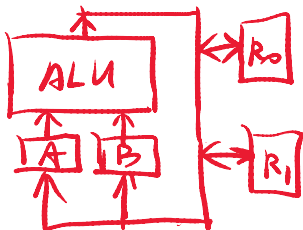
存储容量 小, 大

存临时数据、指令

存长期数据和程序

2. (3 分) 在单总线结构 ALU 中, 完成 ADD R0,R1 指令, 需要对寄存器分别进行多少次读操作和写操作？

3 3



读: R0, R1, A, B 4

写: A, B, R0 3

3. (12 分) 请采用硬件描述语言设计一个具有 8 个通用寄存器的 RN, 它的输入输出接口和功能如下。编程要求写出完整的实体及结构体实现部分。

输入信号: 时钟信号 clk; 异步复位信号 rst; 片选信号 RN_CS;

写信号 WRRN; 读信号 RDRN;

源寄存器选择信号 RS(2 downto 0);

目的寄存器选择信号 RD(2 downto 0);

输入输出信号: 数据总线 data(7 downto 0)。

(1) 读寄存器功能

clk 上升沿有效, RN_CS 低电平有效, 读信号 RDRN 高电平有效, 根据源寄存器选择信号 RS, 将相应寄存器数据输出到 data[7..0]。

(2) 写寄存器功能

clk 上升沿有效, RN_CS 低电平有效, 写信号 WRRN 高电平有效, 根据目的寄存器选择信号 RD, 将 data[7..0]数据写入到相应的寄存器。

entity RN is

port (clk, rst, RN_CS: in std_logic;

WRRN, RDRN: in std_logic;

RS: in std_logic_vector(2 downto 0);

RD: in std_logic_vector(2 downto 0);

data: inout std_logic_vector(7 downto 0);

end RN;

architecture behavioral of RN is

type RData is array (7 downto 0) of std_logic_vector
(7 downto 0);

Signal RDD : RData;

begin

process (clk)

begin

if (rising-edge(clk) and RN_CS = '0')

then

if RDN = '1' then

data ← RDD(conv_integer(RS));

elsif WRRN = '1' then

RDD(conv_integer(RD)) ← data;

end if;

end if;

end process;

end behavioral;

装

订

线