

西安电子科技大学

考试时间 120 分钟

试 题

| 题号 | 一 | 二 | 三 | 四 | 五 | 六 | 总分 |
|----|---|---|---|---|---|---|----|
| 分数 | | | | | | | |

1. 考试形式：闭卷 ☒ 开卷 ☐ ； 2. 本试卷共六大题，满分 100 分；
3. 考试日期：2020 年 月 日；（答题内容请写在装订线外）

一、VHDL 程序分析（本题共 20 分）

根据 VHDL 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 | use IEEE.STD_LOGIC_ARITH.ALL; |
| 4 | use IEEE.STD_LOGIC_UNSIGNED.ALL; |
| 5 | entity BoothDivider is |
| 6 | port(|
| 7 | clk,rst : in STD_LOGIC; |
| 8 | A, B : in STD_LOGIC_VECTOR(7 downto 0); |
| 9 | D :out STD_LOGIC_VECTOR(15 downto 0); |
| 10 | Done: out STD_LOGIC); |
| 11 | end BoothDivider; |
| 12 | architecture Behavioral of BoothDivider is |
| 13 | signal cnt : STD_LOGIC_VECTOR(2 downto 0):= "000"; |
| 14 | signal DD : STD_LOGIC_VECTOR(15 downto 0); |
| 15 | signal tempdone: STD_LOGIC; |
| 16 | Begin |
| 17 | D<=DD; |
| 18 | Done<=tempdone; |
| 19 | process(clk, rst, A, B) |
| 20 | variable A1 : STD_LOGIC_VECTOR(1 downto 0); |
| 21 | begin |
| 22 | If (rst = '1') then |
| 23 | cnt <= "0000"; |
| 24 | tempdone<='0'; |
| 25 | elsif (clk'event AND clk = '1') then |
| 26 | if (cnt <>"111") then |
| 27 | ① cnt <= cnt+1 |
| 28 | tempdone<='0'; |
| 29 | if cnt = "0000" then |
| 30 | A1(0) := '0'; |
| 31 | A1(1) := A(0); |
| 32 | DD(15 downto 8) <= "00000000"; |
| 33 | DD(7 downto 0) <= ② A; |

| | |
|----|--|
| 34 | else |
| 35 | A1(1 downto 0):=DD (1 downto 0); |
| 36 | DD(15)<= DD(14); |
| 37 | DD(14 downto 0) <= ③, DD(15 downto 1) |
| 38 | end if; |
| 39 | if A1 = "10" then |
| 40 | DD(15 downto 8) <= DD(15 downto 8) - B(7 downto 0) ; |
| 41 | elsif A1 = ④ then |
| 42 | DD(15 downto 8) <= DD(15 downto 8) + B(7 downto 0) ; |
| 43 | end if; |
| 44 | else |
| 45 | <u>tempdone</u> <= '1' |
| 46 | ⑤ |
| 47 | end if; |
| 48 | end if; |
| 49 | end procese; |
| 50 | end Behavioral; |

1. (12 分) 请根据所给出的程序代码, 说明上述程序实现了什么功能? 并补充完成①②③④⑤处所缺少的代码。

~~原码加减交替法~~

2. (8 分) 请问 VHDL 常用的数据对象包括哪些? 上述程序采用的是哪几种数据对象, 它们之间的区别是什么?

常量
 ✓ 变量
 ✓ 信号
 文件

- 1、Constant (常量) 在程序中不可以被赋值
- 2、Variable (变量) 在程序中可以被赋值(用 “:=”) , 赋值后立即变化为新值。★
- 3、Signal (信号) 在程序中可以被赋值(用 “<=”) , 但不立即更新, 当进程挂起后, 才开始更新。
- 4、File (文件) 在程序中实现对文件的写入和读出操作。

二、SoC 结构设计分析（本题共 20 分）

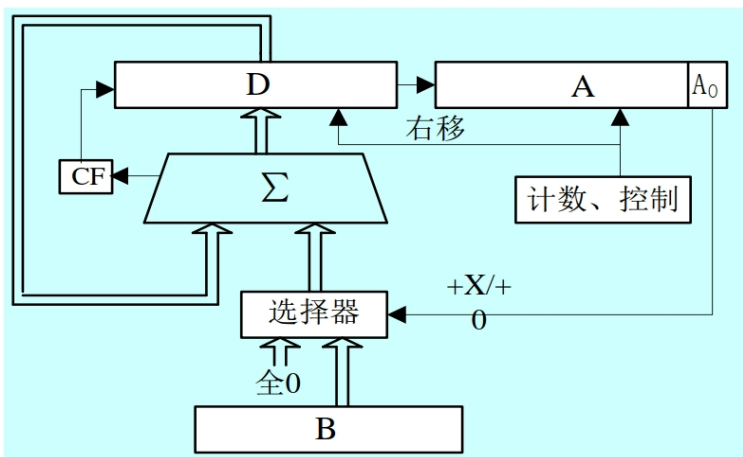
片上系统 SoC 在单个芯片上实现了系统的功能，是计算机与微电子学科交叉的新兴方向。

1.（6 分）调试测试是 SoC 设计开发的重要环节，在测试过程使用的 JTAG 接口表示什么含义？具体包括哪几个信号？

2.（4 分）SoC 片内总线是片上处理器与片内功能 IP 核连接的重要数据通路。典型的 SoC 片上总线包括哪几种？

AMBA, Core Connect, Avalon
Wishbone, OPC

3.（10 分）乘法器是 SoC 重要的运算模块。在实现过程中，若采用硬件设计思路，请画出构成原码一位乘法器的基本结构，并说明每个功能模块的作用。



三、有限状态机设计（本题共 15 分）

按键是人机交互的重要输入设备。按键开关通常为机械弹性开关，在开关闭合及断开的瞬间均伴随有一连串的抖动。抖动时间的长短由按键的机械特性决定。按键动作发生时，按键输出会出现不稳定的逻辑‘0’和逻辑‘1’的跳变，如图 1 所示。

为了获得稳定的按键输入信号，需要进行按键消抖处理。请采用有限状态机的方法进行按键消抖电路设计。

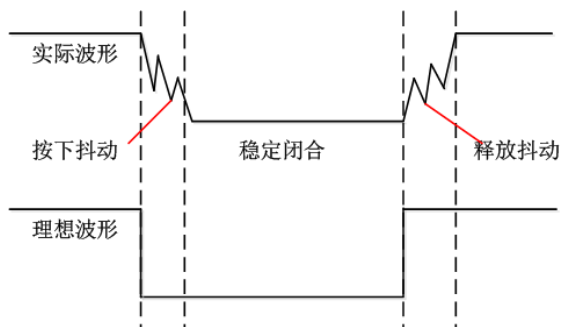
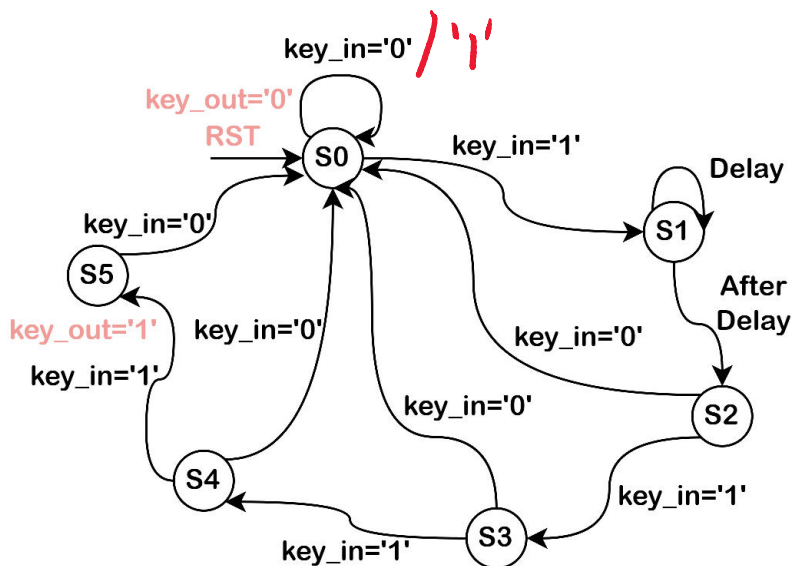


图 1 按键输入信号

1. (5 分) 有限状态机按照输出方式分为哪两种类型？两者的区别是什么？

米里型 输入, 当前状态
摩尔型 当前状态

2. (10 分) 根据按键输入信号特点，请画出按键消抖电路的状态转移图。



四、SoC 优化设计（本题共 15 分）

在同步系统中,为了保证时钟质量应使时钟偏移最小化。若要设计一个计数器系统,该系统包含两个计数器,其中一个计数器以系统时钟 `clk` 频率计数;另一个计数器以 `clk` 的 $1/4$ 频率计数。两个计数器同时复位。假设仅要求两个计数器速度满足 4 倍关系,对相位无任何要求。请画出派生时钟和派生使能两种时钟产生方案的结构图和波形图。

装

订

线

五、程序计数器设计（本题共 15 分）

程序计数器是 SoC 系统的重要功能部件，指示当前指令所在位置。若程序存储器的容量为 $8K \times 8\text{bit}$ 。程序计数器的功能包括以下几点：

a) 全局异步复位功能

当复位信号为低电平时， $\text{ADDR} \leftarrow X \text{ "1FF0"}$ ；数据总线高阻态；

b) 输出地址加 1 功能

clk_PC 上升沿有效； PC_Plus 高电平有效，原 PC 地址+1 $\rightarrow\text{ADDR}$ ；

c) 地址更新功能

clk_PC 上升沿有效， nLD_PC 低电平有效，新的 PC $\rightarrow\text{ADDR}$ 。

1. (5 分) 在执行什么指令的时候，程序计数器会进行地址更新？更新的地址存放在哪个功能模块？若系统增加了函数调用指令，程序计数器应当增加什么功能及操作？

跳转指令 PC

堆栈功能

压入返回地址与弹出返回地址

2. (10 分) 请根据 PC 程序计数器的功能，采用硬件描述语言编写 PC 完整的实体及结构体实现部分代码。具体的输入输出端口要求如下：

输入信号：时钟信号 clk_PC ；复位信号 rst ；地址加 1 控制信号 PC_Plus ；

地址更新控制信号 nLD_PC ；

地址输入信号 $\text{PC}(12 \text{ downto } 0)$ 。

输出信号：地址总线信号 $\text{ADDR}(12 \text{ downto } 0)$ ；数据总线 $\text{data}(7 \text{ downto } 0)$ 。

六、堆栈指针设计（本题共 15 分）

堆栈模块是 SoC 系统存储数据的重要功能模块，用于保护寄存器数据或者程序执行的位置及状态信息。堆栈指针的功能包括堆栈初始化，堆栈指针加 1 和减 1 操作。具体功能如下：

a) 栈顶初始化功能

clk_SP 上升沿有效，SP_CS 低电平，data→SP。

b) 加 1 功能

clk_SP 上升沿有效，nSP_en 低电平有效，SP_CS 高电平，SP_UP 高电平， $SP+1 \rightarrow SP$, $SP \rightarrow Addr$ 。

c) 减 1 功能

clk_SP 上升沿有效，nSP_EN 低电平有效，SP_CS 高电平，SP_DN 高电平， $SP-1 \rightarrow SP$, $SP \rightarrow Addr$ 。

1. （5 分）若要执行 PUSH R0 寄存器压栈指令，请描述该指令执行时涉及的功能模块及微操作。

PC → ADDR;
PC: M[ADDR] → Bus;
Bus → IR;
PC = PC + 1;
IR: IR → Microcontrol;
{ SP + 1 → SP;
SP → ADDR;
R0 → M[ADDR];

2. （10 分）请根据堆栈指针的功能要求，采用硬件描述语言编写堆栈指针的实体及结构体实现部分代码。具体的输入输出端口要求如下：

输入信号：时钟信号 clk_SP；异步复位信号 rst；堆栈选择信号 SP_CS；

指针加 1 控制信号 SP_UP；指针减 1 控制信号 SP_DN；

堆栈输出使能信号 nSP_EN；

输出信号：RAM 地址信号 AR（7 downto 0）；

双向信号：数据信号 data（7 downto 0）。

装

订

线