

得分	评卷人

## 二、分析问答题（12 分）

在 CRC 编码实验中，待传输 16 位数据位 = 0000 0000 0000 0111，采用 CRC 循环冗余校验码进行数据校验，生成多项式为 1101111。

- 1) 试根据 CRC 校验码的编码规则给出该编码的 CRC 余数。

$$111000000 \bmod 1101111 = 010011$$

- 2) 假设接收方接收到的最终编码为 0000 0000 0000 0111 0000 11，假设最多发生一位错，最低位为第 1 位，结合课程实验中所用到的方法说明 CRC 编码如何定位错误并纠正错误。

$$111000011 \bmod 1101111 = 010000 \quad \text{发生错误}$$

根据余数应该右起第 5 位错

- 3) 课程实验中并行 CRC 编码电路的基本思路是什么？

提前计算若干编码的余数，然后将编码数据拆解成若干编码的排列组合，将余数异或加即可得到余数。

- 4) 在 CRC 编码流水传输中是如何区分一位错还是两位错的？对于两位错，流水线是如何处理的？

1、可以引入偶校验位，也可以直接利用 CRC 编码余数进行判断。。

2、对于两位错，要求清除两个接口的数据，并在发送端重新回滚。

得分	评卷人

## 七、工程设计题（16 分）

在 CPU 设计实验中我们实现了如下 8 条 MIPS 指令的多周期 CPU，指令功能描述如下表所示，后页给出了主机数据通路图。

#	MIPS 指令	RTL 功能描述
1	add \$rd,\$rs,\$rt	$R[\$rd] \leftarrow R[\$rs] + R[\$rt]$
2	slt \$rd,\$rs,\$rt	$R[\$rd] \leftarrow R[\$rs] < R[\$rt]$ 小于置 1，有符号比较
3	addi \$rt,\$rs,imm	$R[\$rt] \leftarrow R[\$rs] + \text{SignExt}_{16b}(\text{imm})$
4	lw \$rt,imm(\$rs)	$R[\$rt] \leftarrow \text{Mem}_{4B}(R[\$rs] + \text{SignExt}_{16b}(\text{imm}))$
5	sw \$rt,imm(\$rs)	$\text{Mem}_{4B}(R[\$rs] + \text{SignExt}_{16b}(\text{imm})) \leftarrow R[\$rt]$
6	beq \$rs,\$rt,imm	if( $R[\$rs] = R[\$rt]$ ) $PC \leftarrow PC + \text{SignExt}_{18b}(\{\text{imm}, 00\})$
7	bne \$rs,\$rt,imm	if( $R[\$rs] \neq R[\$rt]$ ) $PC \leftarrow PC + \text{SignExt}_{18b}(\{\text{imm}, 00\})$
8	syscall	系统调用，这里用于停机

- 1) 根据主机数据通路图的信息请给出 sw 指令在取指令阶段和执行指令阶段的数据通路和控制信号。

### 1) 取指令阶段

时钟	数据通路	控制信号
T1	<b>Mem[PC]→IR</b> <b>PC+4→PC</b>	<b>IRwrite=1 IorD=0 PcWrite=1 MemRead</b> <b>ALU_OP ALU_srcA ALU_SrcB</b>
T2	<b>Reg→A、B</b> <b>PC+4+IMM16&lt;&lt;2→C</b>	<b>ALU_OP ALU_srcA ALU_SrcB</b>
T3		
T4		

### 2) 执行指令阶段

时钟	数据通路	控制信号
T1	<b>A+B→C</b>	<b>ALU_OP ALU_srcA ALU_SrcB</b>
T2	<b>B→Mem[C]</b>	<b>IorD=1 MemWrite</b>
T3		
T4		

- 2) 在 Logisim 环境中支持 Ctrl+R 进行系统总复位，为什么电路中还要增加一个 Rst 复位信号？

Ctrl+R 复位会清空 RAM 存储器，导致程序丢失，增加的 Rst 信号可避免 RAM 数据丢失

- 3) 如果采用微程序构造控制器，采用水平型直接表示法，微指令分为哪几个字段，各字段长度多少，各字段包括哪些信息，

操作控制字段 17 位

P 字段 1 位

下址字段 4 位

- 4) 微程序通常是串行执行的，简要叙述取指微程序执行完毕后系统是如何跳转到当前指令对应的微程序入口地址的？

P 字段为 0 时，下址字段提供下条微指令地址，P 字段为 1 时，由地址转移逻辑根据指令功能给出下条微指令地址。









