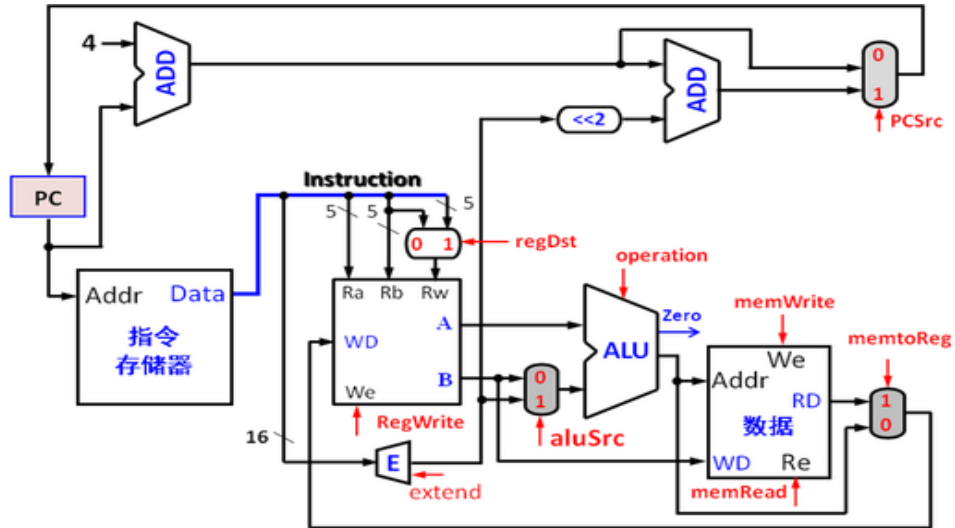


| | | | | | | | | | | | |
|----|---|---|----|-----|---|-----|---|-----|----|------|----|
| 71 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| B | A | D | CD | BCD | A | BCD | B | ACD | D | ABCD | C |

- 1、 加法器采用先行进位的目的是（ ）
A.优化加法器结构 **B.提高进位产生速度**
C.保证运算结果正确 D.正确传递进位值
- 2、 十进制数 178.125 表示成 IEEE 754 单精度浮点数的结果是（ ）
A.43322000H B.43591000H C.B3591000H D.B3322000H
- 3、 指令所需要的操作数不会来自（ ）
A.指令寄存器 IR B.主存
C.通用寄存器 **D.变址寄存器**
- 4、 **[多选]**下列计算机系统性能评价的描述中正确的是
A.程序 MIPS 值越高，计算机的性能越高
B.程序的 CPI 值越低，计算机的性能越高
C.主频高的机器性能不一定高
D.同一程序在不同机器的 MIPS 不一定相同
- 5、 **[多选]**下列关于取指令阶段指令流的描述中，正确的是()
A.取指流程中只有一条数据通路；
B.取指流程包含取指令和 PC 增量流程
C.不同 PC 增量方式影响取指流程；
D.CPU 内总线结构影响取指流程；
- 6、 某机有 5 级中断 L4~L0，中断响应优先级从高到低的次序是 L4>L3>L2>L1>L0，设中断屏蔽字为 M4M3M2M1M0， $M_i=1$ ($0 \leq i \leq 4$) 表示对 Li 级中断进行屏蔽。若要实现中断完成（处理）优先级从高到低的顺序 L4>L1>L3>L2>L0，则在 L3 的中断服务程序中，应把中断屏蔽字设置为()

A.01101 B.10010 C.11000 D. 00111

- 7、 **[多选]**下列关于 Cache 的说法中，错误的是
- A.采用直接映像时，Cache 无需考虑替换问题
 - B.如果选用最优替换算法，则 Cache 的命中率可达到 100%
 - C.提高 Cache 本身速度比提高 Cache 命中率更有利于提升存储器的等效访问速度
 - D.Cache 的容量与主存的容量差别越大，存储系统的等效访问速度越高
- 8、 在补码定点加减法运算的溢出判别中，下列说法错误的是()
- A.对于减法，符号不同的两个数相减可能发生溢出
 - B.对于加法，符号相同的两个数相加一定发生溢出
 - C.对于加法，符号不同的两个数相加永不会发生溢出
 - D.对于减法，符号相同的两个数相减永不会发生溢出
- 9、 **[多选]**单周期 MIPS 在一个时钟周期中可能完成的操作 ()
- A. ALU 运算和向寄存器堆写数据
 - B.从数据存储器读数据和向数据存储器写数据；
 - C.更新 PC 内容和向数据存储器写数据
 - D.寄存器堆读数据，ALU 运算和数据存储器写数据
- 10、 某型 MIPS32 指令架构的单周期 CPU，其数据通路结构如下图：



执行指令 **sw rt, offset(rs)** 时，应由控制器产生的控制信号 PCSrc、aluSrc、memRead/memWrite、regWrite 和 extend 分别是（ ）

- A. 1、0、1/0、1、0;
- B. 1、1、0/0、0、1;
- C. 0、1、1/0、1、0;
- D. 0、1、0/1、0、1;

11、[多选]下列寄存器中，对系统硬件工程师不透明的是

- A. 存储器地址寄存器(MAR)
- B. 程序计数器(PC)
- C. 存储器数据寄存器(MDR)
- D. 指令寄存器(IR)

12、已知一个虚拟页式存储系统，其物理内存内存容量为 512MB，虚存容量为 128GB，均按字节寻址,假定页面大小为 4KB，则该页式虚拟存储系统的虚页号为（ ）位。

- A. 37
- B. 29
- C. 25
- D. 12

| | |
|----|-----|
| 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 11011111。

- 1) 试根据 CRC 校验码的编码规则给出该编码的 CRC 余数。

$$111000000 \bmod 11011111 = 010011$$

- 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。

$$111000011 \bmod 11011111 = 010000 \quad \text{发生错误}$$

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 |
| | |

三、分析计算题（12 分）

计算分析题（10 分）已知 $[x]_{\text{补}}=11111011$ ， $[y]_{\text{补}}=01010111$ ，用补码一位乘法计算 $[x \times y]_{\text{补}}=?$ （单符号位）将答案填写在下面，并将计算过程填写在表格中。

$$[x \times y]_{\text{补}} = \underline{7E4D} \quad (\text{十六进制}) \quad x \times y = \underline{-435} \quad (\text{10 进制})$$

如果 $[x]_{\text{补}}=1000000$ 时， $[y]_{\text{补}}=00000001$ ，运算结果等于多少？结果是否正常，为什么？

=80，结果不正常，因为 $-x$ 溢出了，无法进行运算。

$$[-x]_{\text{补}}=00000101$$

| # | 运算 | 部分积 | 移出位 | 判断位 $y_n y_{n+1}$ |
|----|---------------------|----------|---------|-------------------|
| 1 | | 00000000 | | <u>010101110</u> |
| 2 | $+ [-x]_{\text{补}}$ | 00000101 | | |
| 3 | = | 00000101 | | |
| 4 | → | 00000010 | 1 | <u>01010111</u> |
| 5 | +0 | 0 | | |
| 6 | = | 00000010 | 1 | |
| 7 | → | 00000001 | 01 | <u>0101011</u> |
| 8 | +0 | 0 | | |
| 9 | = | 00000001 | 01 | |
| 10 | → | 00000000 | 101 | <u>010101</u> |
| 11 | $+ [x]_{\text{补}}$ | 11111011 | | |
| 12 | = | 11111011 | 101 | |
| 13 | → | 11111101 | 1101 | <u>01010</u> |
| 14 | $+ [-x]_{\text{补}}$ | 00000101 | | |
| 15 | = | 00000010 | 1101 | |
| 16 | → | 00000001 | 01101 | <u>0101</u> |
| 17 | $+ [x]_{\text{补}}$ | 11111011 | | |
| 18 | = | 11111100 | 01101 | |
| 19 | → | 11111110 | 001101 | <u>010</u> |
| 20 | $+ [-x]_{\text{补}}$ | 00000101 | | |
| 21 | = | 00000011 | 001101 | |
| 22 | → | 00000001 | 1001101 | <u>01</u> |
| 23 | $+ [x]_{\text{补}}$ | 11111011 | | |
| 24 | = | 11111100 | 1001101 | |
| 25 | | | | |

| | |
|----|-----|
| 得分 | 评卷人 |
| | |

四、综合分析题（12 分）

一个包含 10 个元素的一维数组顺序存放在主存连续单元，主存每个存储单元存放一个数组元素，机器的 Cache 分为指令 Cache 和数据 Cache，其中数据 Cache 有 4 行，每行可存放两个数组元素，Cache 的初始状态为空，采用 LRU 替换算法，某程序的伪代码如下：

```
SUM :=0
for j:=0 to 9 do
  SUM:= SUM +A(j)
end
Ave:= SUM/8
for i:=9 to 0 do
  A(i) := A(i)/Ave
end
```

- 1) 将 Cache 分为指令 Cache 和数据 Cache 的架构称为哈佛结构，这种结构有什么好处？
- 2) 计算全相联映射方式下 Cache 读操作的命中率。
- 3) 假定主存的存取时间为 200ns，Cache 的存取时间为 20 ns，求该存储系统的平均存取时间。

参考答案：

- 1) 将 Cache 分为指令 Cache 和数据 Cache 时,CPU 可并行访问指令和数据，缩短程序的执行时间。（2 分）
- 2) 全相联系方式下 Cache 中的内容变化如下图所示
a)第一轮循环读命中情况

| | j=0-7 | 命中 | j=8,9 | 命中 |
|----|-----------|------|-----------|------|
| C0 | a[0],a[1] | a[1] | a[8],a[9] | a[9] |
| C1 | a[2],a[3] | a[3] | a[2],a[3] | |
| C2 | a[4],a[5] | a[5] | a[4],a[5] | |
| C3 | a[6],a[7] | a[7] | a[6],a[7] | |

b)第二轮循环命中的情况

| 循环前 | j=9-2 | 命中 | j=1, 0 | 命中 |
|-----------|-----------|-----------|-----------|------|
| a[8],a[9] | a[8],a[9] | a[8],a[9] | a[1],a[0] | a[0] |
| a[2],a[3] | a[2],a[3] | a[3],a[2] | a[2],a[3] | |
| a[4],a[5] | a[4],a[5] | a[5],a[4] | a[4],a[5] | |
| a[6],a[7] | a[6],a[7] | a[7],a[6] | a[6],a[7] | |

两轮循环中数组中的元素共被访问 20 次，两轮循环中从数据 Cache 中共命中 14 次，命中率为 $14/20 = 70\%$

3) $T = 20 \text{ ns} \times 0.7 + 200 \text{ ns} \times (1-0.7) = 74 \text{ ns}$ （4 分）

| | |
|----|-----|
| 得分 | 评卷人 |
| | |

五、设计分析题（12 分）

某计算机数据线 32 位，地址线 18 位，按字节编址。主存 字 地址空间分配如下：0000H - 3FFFH 为 ROM 区，4000H - FFFFH 为 RAM 区，现要求用 8K×8 位的 RAM 和 ROM 芯片构成上述 32 位的存储器。完成下列问题：

（1） 计算所需要的 RAM 和 ROM 芯片数量；

（1） ROM 有 16K 地址，需要 $\frac{16}{8} \times \frac{32}{8} = 8$ 片 8K×8 位的 ROM

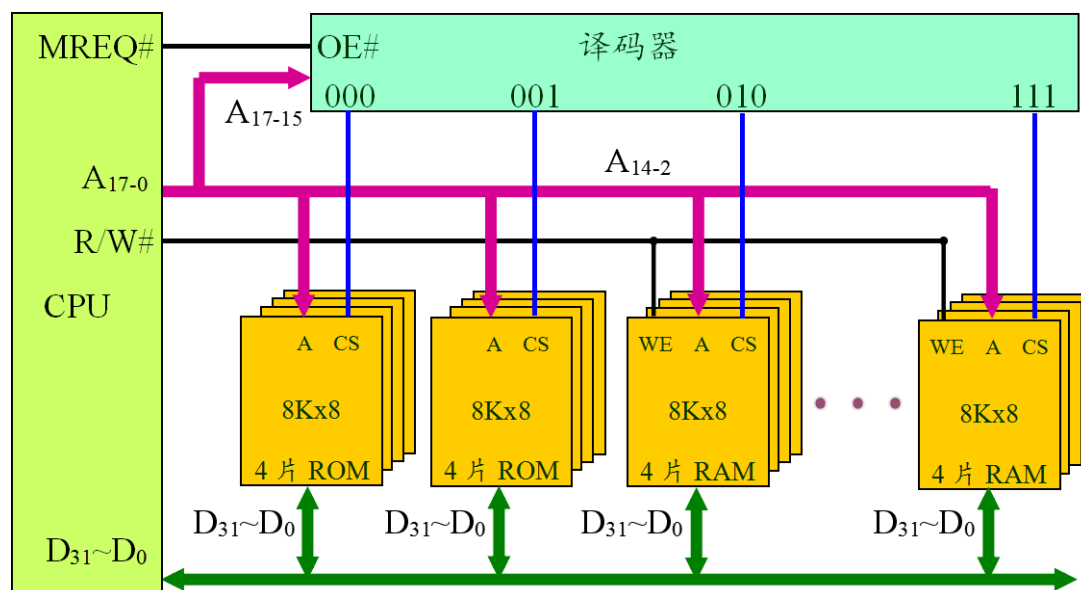
RAM 有 48K 地址，需要 $\frac{48}{8} \times \frac{32}{8} = 24$ 片 8K×8 位的 RAM

（2） 如何保证存储器按字边界对齐？

该 32 位存储器总共有 64K 地址，需要 16 根地址线访问，为了保证存储器按字边界对齐可以采用以下方法：即计算机的 18 位地址线高 16 位连接存储器，低 2 位地址则不连接存储器，而是用于字节访问。

（3） 画出 CPU 与存储器的连接，注意片选信号和不同芯片的顺序。

CPU 与存储器的连接如下图所示：



| | |
|----|-----|
| 得分 | 评卷人 |
| | |

六、设计分析题（12 分）

现针对云计算开发了一款64位的MIPS指令集的变体，为了更适应大数据场景，操作码Opcode字段为7位，通用寄存器的数目由32个变成了128个，寄存器数据位宽64位，假设R型指令中64位字长的指令字多余位均扩展到最右侧的Funct字段，I型指令中多余位均扩展到Immediate字段。

（1）请给出R型指令和I型指令字段划分。

| Opcode | Rs | Rt | Rd | Shamt | Funct |
|--------|----|----|----|-------|-------|
| 7 | 7 | 7 | 7 | 6 | 30 |

| Opcode | Rs | Rt | Imm |
|--------|----|----|-----|
| 7 | 7 | 7 | 43 |

（2）该MIPS指令系统最多可以支持多少条指令，给出计算表达式。

$$2^7 - 1 \text{ 非R型号指令} + 2^{30} \text{ R型指令}$$

（3）执行一条R型指令后，PC的增量是多少，给出计算表达式；

$$\$pc = \$pc + 8;$$

（4）请问有条件分支指令Beq最多能将PC寄存器的值增加多少？给出计算表达式。

$$\begin{aligned} &\text{Imm字段位宽 } 43\text{位, 补码最大整数为 } 2^{42} - 1, \text{ 跳转指令数最大为 } (2^{42} - 1) \ll 3 \\ &\$pc = \$pc + 8 + (2^{42} - 1) \ll 3 = \$pc + 2^{45} \end{aligned}$$

| | |
|----|-----|
| 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 | Mem[PC]→IR PC+4→PC | IRwrite=1 IorD=0 PcWrite=1 MemRead ALU_OP ALU_srcA ALU_SrcB |
| T2 | Reg→A、B PC+4+IMM16<<2→C | ALU_OP ALU_srcA ALU_SrcB |
| T3 | | |
| T4 | | |

2) 执行指令阶段

| 时钟 | 数据通路 | 控制信号 |
|----|-----------------|---------------------------------|
| T1 | A+B→C | ALU_OP ALU_srcA ALU_SrcB |
| T2 | B→Mem[C] | IorD=1 MemWrite |
| 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

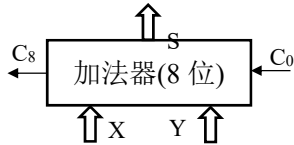
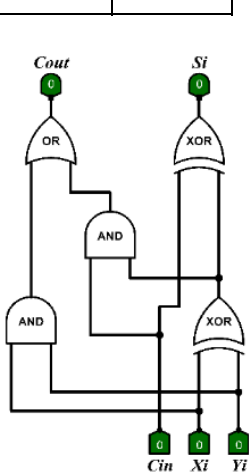
下址字段 4 位

- 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。

| | |
|-----|--|
| 分 数 | |
| 评卷人 | |

一、现有 8 位加法器, 输入分别为被加数 $X = X_8 \sim X_1$, 加数 $Y = Y_8 \sim Y_1$, 进位输入 C_0 , 输出 $S = S_8 \sim S_1$, 第 8 位为最高位, C_8 为最高位进位, 完成下列各题。(14 分)



1) 若该 8 位加法器是由 8 个一位全加器 FA 串联构成, FA 内部结构如图所示, 假设所有门电路时间延迟均为 $1T$, 则全加器 FA 的关键时延为 $3T$ 。8 位串行加法器关键时延为 $17T$ 。(4 分)

2) 结合组合逻辑电路的特性, 简要分析怎样才能保证 8 位串行加法器运算结果是正确的?(2 分)

组合逻辑电路的特点是输出随输入的改变而改变, 从工程上看, 该加法器输入端数据一定要持续 8 位串行加法器的延迟, 只有这样才能保证每次运算结果的有效性。

- 3) 若 X 、 Y 、 Z 采用补码, 则 S 最大值为 $7F$ (16 进制), 最小数为 80 (16 进制)。(2 分)
- 4) 该加法器是否可以无符号数的运算, 简要说明理由。(2 分)
- 5) 该 8 位加法器具有溢出检测功能吗? 如果有请说明原因; 如果没有, 请给出一种增加溢出检测功能的方法。从软硬协同的角度分析运算器提供硬件溢出检测机制的意义(5 分)

只能对无符号数加法具有溢出检测功能。

对有符号数的溢出没有检测功能。对于本运算器而言, 只能通过额外设计电路, 通过参加运算数据的符号位和结果的符号位是否相同的方式来判断溢出。

硬件提供溢出检测后, 程序员可通过该溢出位判断溢出与否, 而不再需要通过专门程序判断溢出, 简化了程序设计, 且有利于流水线 CPU 的高效运行。

| | |
|-----|--|
| 分 数 | |
| 评卷人 | |

二、某原始 16 位数据为 0x0181, 采用 CRC 校验码进行传输和校验, 若采用的生成多项式为 $G(x)=x^6+x^5+x^3+x^2+x+1$, 完成下列各题。(14 分)

1) 该原始数据对应的编码可能是某汉字 GB2312 编码吗, 为什么? (2 分)

不是, 因为汉字 GB2312 编码两个字节的最高位应该是 1。

2) 若采用 CRC 校 1 验码对该数据进行编码传输, 求对应的 CRC 编码 (最终结果以 16 进制表示, 要求写出计算过程) (4 分)

0000 0001 1000 0001 000000 mod 1101111 = 001011

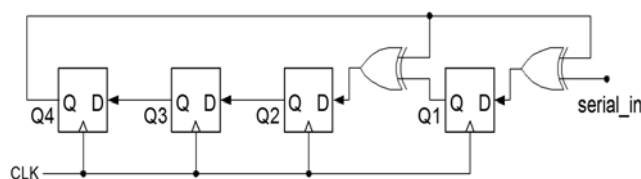
最终编码为 0000 0001 1000 0001 001011

16 进制为 604B

3) 若接收方进行 CRC 解码时得到 CRC 余数是 010000, 在什么条件下可通过 CRC 校验的原理判断出错的位置, 并给出此条件下由上述余数计算出的出错位。(4 分)

在能判断只有一位出错的情况下, 可利用接收方解码时得到的不为 0 余数判断出错位。这里应该是传输编码的第 5 位出错。

4) 下图为某串行 CRC 电路



(1) 根据该图给出此 CRC 校验使用的生成多项式, 并简要说明理由。(2 分)

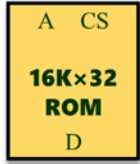
10011, 根据 CRC 校验码的原理, 被除数最高位为 1 时才进行减法运算, 图中最低两个 D 触发器的输入端有 2 个异或门

(2) 简要对比并行 CRC 电路与串行 CRC 电路的特点。(2 分)

时序/组合逻辑 多个时钟周期/单周期 慢/快 硬件少/硬件多

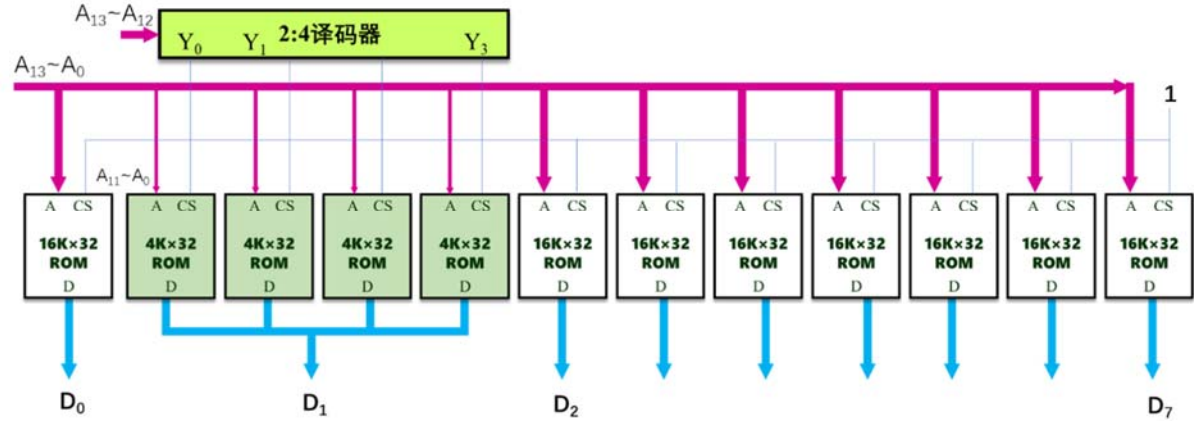
| | |
|-----|--|
| 分 数 | |
| 评卷人 | |

三、汉字显示可以采用字形码进行点阵显示，现需要构建一个 16*16 点阵的汉字字库，用于显示 16 位的 GB2312 汉字编码，假设需要一次读出一个汉字字形码的全部点阵信息。现在有 7 块 16K*32 位的 ROM 芯片，7 块 4K*32 位的 ROM 芯片，规格如下图。（14 分）



1) 选用其中的部分芯片设计汉字字库存储系统，则该字库存储系统地址线至少是 14 根，数据线至少 256 根，需要 16K*32 位 ROM 7 片，4K*32 位 ROM 4 片。（4 分）

2) 给出该汉字字库的逻辑示意图，假设输入为地址线 $A_1 \cdots A_0$ ，要求标注各芯片的详细地址输入，片选 CS 输入（高电平有效），数据输出 $D_k \cdots D_0$ 。（5 分）



3) 假设 GB2312 汉字点阵信息全部存放在一个文件中，每个汉字的点阵数据在文件中顺序存放，请问汉字数据如何分割载入到你设计的存储系统的每个芯片中。（3 分）

首先需要将字库文件均分为 8 个文件，其中 7 个文件载入 16K*32 位的然后将其中一个文件再均分成 4 份，

4) 这个汉字字库如何才能显示英文字符和数字字符？（2 分）

可以使用全角字符代替英文字符和数字字符

| | |
|-----|--|
| 分 数 | |
| 评卷人 | |

四、某计算机系统中主存地址 16 位，全部主存空间都被使用；该机 Cache 采用四路组相联，共有 8 个 Cache 行；主存数据块大小为 4 个字节，完成下列各题。（16 分）

1) 该 Cache 分为几组？每个 Cache 行应该包含哪些信息？这些信息各为多少位？（4 分）

Cache 分为 2 组，每组 4 行

valid 位 1 位，标记字段 13 位，淘汰计数位（尽可能大）

2) 下表中依次给出了 10 个字节访问地址(10 进制)，假设 Cache 内容初始状态为空，从左到右依次访问这 10 个地址，请给出每次访问对应的主存块地址，组号，行号，以及载入、命中、替换情况（在对应位置打钩即可）。（6 分）

| 访问地址（10 进制） | 0 | 8 | 17 | 33 | 2 | 7 | 37 | 25 | 1 | 9 |
|-------------|---|---|----|----|---|---|----|----|---|---|
| 主存块地址(10) | 0 | 2 | 4 | 8 | 0 | 1 | 9 | 6 | 0 | 2 |
| Cache 组号 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| Cache 组内行号 | 0 | 1 | 2 | 3 | 0 | 4 | 5 | 1 | 0 | 2 |
| 载入 | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | | | |
| 命中 | | | | | ✓ | | | | ✓ | |
| 替换 | | | | | | | | ✓ | | ✓ |

3) 如果 2) 中没有给出“Cache 内容初始状态为空”的假设, 会对 2) 中的命中率产生影响? 分析产生影响的原因。（2 分）

如果开始工作时 Cache 内容不空，此时，如果 Cache 中保存的内容与 CPU 将要访问的内容无关，则对结果没有任何影响，如果 Cache 中保存的内容与 CPU 将要访问的内容有关，则对上表中的载入、替换和命中率等都可能产生影响。

4) 计算 2) 中 Cache 命中率。结合存储体系的工作原理和设置 Cache 的目标，对该 Cache 的命中率进行分析，并从软硬协同的角度简要分析能实现设置 Cache 目标的可能方法。（4 分）

命中率= 2/10

存储体系中设置 Cache 的目的是解决主存速度不快的问题，为此要求 Cache 有比较高的命中率。本题中 Cache 命中率低，因此，难以达到设置 Cache 的目的，从软硬协同的角度看，可分别从硬件和软件两个角度提高 Cache 命中率，可增加 Cache 的容量，也可提高程序访问的局部性。

| | |
|-----|--|
| 分 数 | |
| 评卷人 | |

五、现针对云计算开发了一款64位的MIPS指令集的变体，为了更适应大数据场景，操作码Opcode字段为7位，通用寄存器的数目由32个变成了128个，寄存器数据位宽64位，假设R型指令中64位字长的指令字多余位均扩展到最右侧的Funct字段，I型指令中多余位均扩展到Immediate字段。（14分）

(1) 下图为该MIPS变体R型和I型指令格式，根据题目条件，给出下列指令中空字段的位数。（4分）

| Opcode | Rs | Rt | Rd | Shamt | Funct |
|--------|----|----|----|-------|-------|
| 7 | 7 | 7 | 7 | 6 | 30 |

| Opcode | Rs | Rt | Immediate |
|--------|----|----|-----------|
| 7 | 7 | 7 | 43 |

(2) 该MIPS指令系统最多可以支持多少条R型指令和非R型指令。（3分）

2^7-1 非R型号指令 + 2^{30} R型指令

(3) 该指令体系下，执行一条R型指令后，PC的增量是多少，给出计算表达式。（2分）

$SpC=SpC+8;$

(4) 简要分析X86系统中支持多种寻址方式的意义。简要分析MIPS指令没有设置寻址方式特征位字段以支持多种寻址方式的原因（5分）

X86 系统支持多种寻址方式的意义包括：访问更大的存储空间、访问更大的数据、方便程序设计、提高指令的执行速度等。MIPS 将指令分成三类，只有 I 型指令访问存储器，且寄存器数量多，每类指令需要的寻址方式有限，所以不需要设置寻址方式特征位字段。

| | |
|-----|--|
| 分 数 | |
| 评卷人 | |

六、假定计算机主频为 500MHZ，平均 CPI 为 4，现有设备 A 和 B，其数据传输率分别为 $2 \times 10^6 \text{Byte/s}$ 和 $40 \times 10^6 \text{Byte/s}$ ，对应 I/O 接口中各有一个 32 位的数据缓冲寄存器。回答如下问题，并给出计算过程。（12 分）

1) 若设备 A 采用定时查询 I/O 方式，每次 I/O 都至少执行 10 条指令。设备 A 最多间隔多长时间查询一次才能不丢失数据？此时 CPU 用于设备 A 的输入/输出时间占 CPU 总时间的百分比是多少？（12 分）

方法 1:

根据数据传输率，A 准备一个数据的时间为 $4\text{B}/2 \times 10^6 = 2 \mu\text{s}$ ，所以每隔 $2 \mu\text{s}$ 必须查询一次。
一次传输需要的 CPU 时间为 $10 \times 4 / (500 \times 10^6) = 0.08 \mu\text{s}$
百分比 $= 0.08 / 2 = 4\%$

方法 2

每秒查询次数至少 $1\text{s} / 2 \mu\text{s} = 5 \times 10^5$ ，CPU 占用时间 $5 \times 10^5 \times 10 \times 4$ 个时钟周期
占比 $5 \times 10^5 \times 10 \times 4 / 500\text{M} = 4\%$

2) 在中断 I/O 方式下，若每次中断响应时钟周期数为 100，中断处理时钟周期数为 300，则设备 B 能否采用中断 I/O 方式进行传输？为什么？

每次中断过程需要时间 $(100+300) / (500 \times 10^6) = 0.8 \mu\text{s}$
每次中断只能传输 32 位 4 个字节
中断方式最大速率 $4 / ((100+300) / (500 \times 10^6)) = 5\text{MB/s}$
中断方式不能满足要求

方法 2:

每次中断过程需要时间 $(100+300) / (500 \times 10^6) = 0.8 \mu\text{s}$
设备 B 准备好一个数据的时间 $4\text{B} / 40 \times 10^6 = 0.1 \mu\text{s}$
中断方式来不及处理

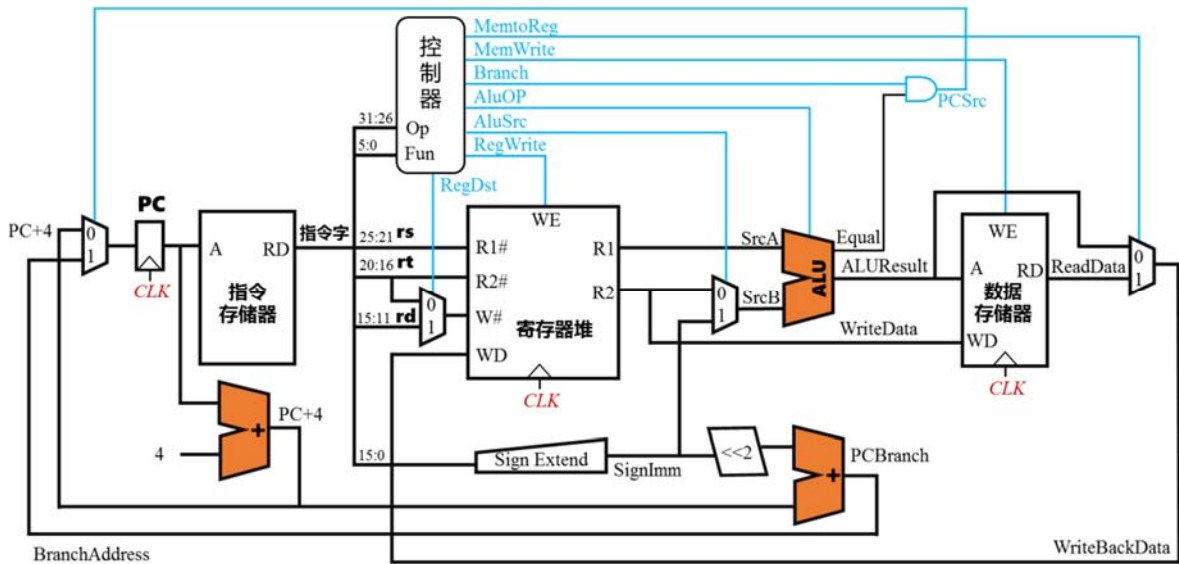
3) 若设备 B 采用 DMA 方式进行数据传输，DMA 数据块大小为 1000Byte，DMA 预处理时间为 250 个时钟周期，后处理时间为 250 个时钟周期，则 CPU 用于设备 B 输入/输出的时间占 CPU 总时间的百分比最多是多少？

每传输 1000 字节 CPU 占用时间为 $(250+250) / 500\text{M} = 1 \mu\text{s}$ 中断方式来不及处理
实际传输时间为 $1000 / 40 \times 10^6 = 25 \mu\text{s}$
比例 $= 1 / 25 = 4\%$

| | |
|-----|--|
| 分 数 | |
| 评卷人 | |

七、下图为某 32 位单周期 MIPS CPU 的逻辑框图,完成下列各问。(16 分)

1) $lw \$rt, imm(\$rs)$ 的功能为: $R[\$rt] \leftarrow Mem_{4B}(R[\$rs] + SignExt16b(imm))$ 。分别给出该指令取指和执行阶段需要使用的主要功能部件及相关控制信号的值, 确保该指令能正确执行。(5 分)



取指令阶段部件: PC 寄存器, 指令寄存器, 加法器

执行阶段部件: 寄存器堆 ALU 符号扩展器 数据存储器

| 控制信号 | MemtoReg | MemWrite | Branch | AluOP | AluSrc | RegWrite | RegDst |
|------|----------|----------|--------|-------|--------|----------|--------|
| 值 | 1 | 0 | 0 | 5 | 1 | 1 | 0 |

2) 假设 lw 指令中 imm 字段的值为 $8AF2H$, 则经过图中扩展电路后的值为_____ (16 进制示)? 为什么要对指令 imm 字段的值进行符号扩展? (3 分)

FFFF8AF2

因为 ALU 为 32 位, imm 字段本身只有 16 位, 当与另一个 32 位数据进行运算时, 必须将符号位扩展为 32 位数才能参与运算

3) 如果要修改 CPU 设计方案, 变成多周期 CPU, 则需要对上述结构做哪些变化? (4 分)

可以将指令存储器和数据存储器合二为一; 在每个功能部件后增加一个缓冲器, 如主存、寄存器堆、ALU 等部件后都要增加。

4) 如果多周期 MIPS CPU 采用微程序控制器, 若要加入中断逻辑, 应该进行哪些扩展? (4 分)

微程序中需要增加中断隐指令的微程序, 该微程序的功能是保存断点, 修改 PC 地址为中断程序入口地址, 微指令 P 字段需要增加一位用于进行中断判断, 每条指令对应微程序最后一条指令的中断判断位为 1, 如果当前有中断请求信号, 要进行分支跳转中断隐指令对应的微程序。

| | |
|-----|--|
| 分 数 | |
| 评卷人 | |

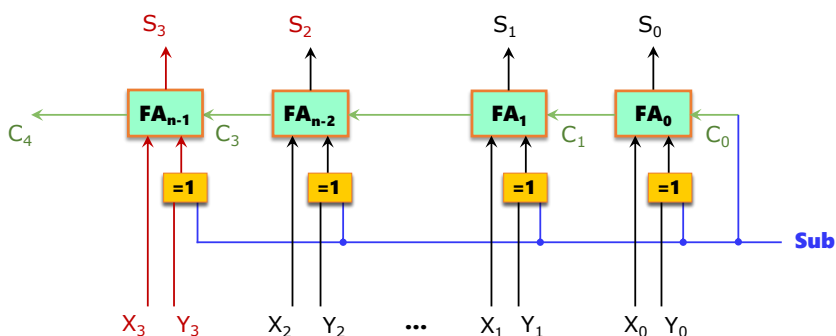
一、要基于一位全加器 FA 串联设计一个 4 位无符号补码可控加减法器，两个运算操作数分别为 $X = X_3X_2X_1X_0$, $Y = Y_3Y_2Y_1Y_0$ ，进位输入信号为 C_0 ，运算结果输出为 $S = S_3S_2S_1S_0$ ， C_4 为进位输出，运算控制信号为 Sub ，试完成下列各题。(12 分)

- 1) 请设计一位全加器 FA 的电路，运算操作数为 X , Y ，进位信号为 C_{in} ，输出为运算结果 S ，进位输出 C_{out} ，给出所有输出信号逻辑表达式。

$$S_i = X \oplus Y \oplus C_{in} \quad (2 \text{ 分})$$

$$C_{out} = XY + (X \oplus Y)C_{in} \quad \text{或} \quad C_{out} = XY + (X + Y)C_{in} \quad (2 \text{ 分})$$

- 2) 以一位全加器 FA 为基础，设计一个 4 位串行无符号补码加减法器，请给出其电路图，并说明其工作原理。(5 分)



- 3) 假设所有门电路时间延迟均为 $1T$ ，则一位全加器 FA 的时延为 $3T$ ，(1 分) 该 4 位可控加减法器的关键路径延迟为 $13T$ 。(13T 给 1 分，10T 给 2 分)

| | |
|-----|--|
| 分 数 | |
| 评卷人 | |

二、某校验码编码长度 15 位，采用了海明码进行校验，编码左到右依次为 $H_{15}H_{14}H_{13} \dots H_1$ ，海明校验组采用偶校验，试完成下列各问。(14 分)

- 1) 根据海明校验的原理，请用打钩的方式在下表中标记出 15 位海明码中的校验位。

| H_{15} | H_{14} | H_{13} | H_{12} | H_{11} | H_{10} | H_9 | H_8 | H_7 | H_6 | H_5 | H_4 | H_3 | H_2 | H_1 |
|----------|----------|----------|----------|----------|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | | | | | | | ✓ | | | | ✓ | | ✓ | ✓ |

(2 分)

- 2) 根据海明码定义，该编码应该分为四组，请给出每组中校验位的逻辑表达式。

| | H_{15} | H_{14} | H_{13} | H_{12} | H_{11} | H_{10} | H_9 | H_8 | H_7 | H_6 | H_5 | H_4 | H_3 | H_2 | H_1 |
|-------|----------|----------|----------|----------|----------|----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| G_4 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | | |
| G_3 | ✓ | ✓ | ✓ | ✓ | | | | | ✓ | ✓ | ✓ | ✓ | | | |
| G_2 | ✓ | ✓ | | | ✓ | ✓ | | | ✓ | ✓ | | | ✓ | ✓ | |
| G_1 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |

(上表为编码设计辅助表格, 可以自行使用, 不做判分依据) (4分)

$$H1 = H3 \oplus H5 \oplus H7 \oplus H9 \oplus H11 \oplus H13 \oplus H15$$

$$H2 = H3 \oplus H6 \oplus H7 \oplus H10 \oplus H11 \oplus H14 \oplus H15$$

$$H4 = H5 \oplus H6 \oplus H7 \oplus H12 \oplus H13 \oplus H14 \oplus H15$$

$$H8 = H9 \oplus H10 \oplus H11 \oplus H12 \oplus H13 \oplus H14 \oplus H15$$

3) 假设指错字为 $G_4G_3G_2G_1$, 如果校验码最多只有一位错, 如何判断错误并纠正错误, 如校验码为 010101101101010, 请进行出错情况判断, 给出计算过程。

指错字值=0 表示没有错误, 否则表示出错位的位置, 只需将对应为取反即可纠错 (2分)

$$G1 = 0+0+0+1+1+0+0+0 = 0$$

$$G2 = 1+0+1+1+1+0+1+0 = 1$$

$$G3 = 1+0+1+1+1+0+1+0 = 1$$

$$G4 = 0+1+1+0+1+0+1+0 = 0$$

$$G_4G_3G_2G_1=6 \text{ 因此 } H6 \text{ 出错} \quad (2分)$$

4) 该编码纠错的前提是什么, 假设没有三位错, 如何识别一位错, 两位错?

假设只有一位错才能纠错 (1分)

可以引入总校验位, (1分)

指错字=1, 总检错位=1 时, 表示一位错, 否则表示两位错。 (2分)

| | |
|-----|--|
| 分 数 | |
| 评卷人 | |

三、某计算机系统主存容量为 256B, 按字节编址, 其高速缓冲存储器 (Cache) 数据存储体的容量为 32B, 假定主存和 Cache 每个数据块大小均为 4B。
(16 分)

1) 若 Cache 采用 2 路组相联, 请给出主存地址向 Cache 地址映射时主存地址划分图, 分别给出标记字段 (Tag)、索引字段 (Index) 和块偏移字段 (Offset) 的位数。 (3 分)

| | | |
|-------------|---------------|----------------|
| Tag (4bits) | Index (2bits) | Offset (2bits) |
|-------------|---------------|----------------|

2) 假定 Cache 采用 LRU 替换策略, 且 Cache 的初始内容为空; 画出 N=10 时, 执行下列代码后 Cache 各组各行中保存的数组数据情况 (按映射方法直接将 v[i] 写在 Cache 特定组的特定行, i 要用 0-9 中具体的值代替, 如 v[1] 等)。(注意: int 类型为 4 个字节, 假定代码执行时数组 V 被加载到主存地址 0 开始的连续存储器地址中, 变量 i, sum 编译时分配到寄存器中)

```
int sumv(int v[N])
{
    int i, sum=0;
    for (i=0; i<N; i++)
        sum += v[i];
    return sum;
}
```

(8 分)

| 组号 | 组内行号 | 内容 |
|----|------|------|
| 0 | 0 | V[8] |
| | 1 | V[4] |
| 1 | 0 | V[9] |
| | 1 | V[5] |
| 2 | 0 | V[2] |
| | 1 | V[6] |
| 3 | 0 | V[3] |
| | 1 | V[7] |

3) 结合 Cache 工作原理和存储体系构建的基本原理, 简要说明存储体系中设置 Cache 的目的是什么? 分析上述代码执行过程中 Cache 作用是否得到了发挥? 给出你认为能提高上述代码执行过程中 Cache 作用有效发挥的办法。

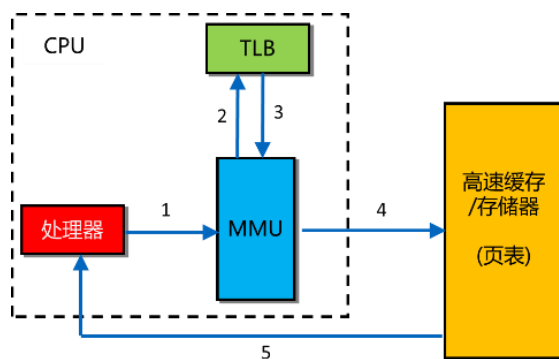
基于局部性原理, 提高 CPU 在 Cache 中访问数据的命中率来缓解 CPU 与主存间的速度差异, 从而提高存储系统的访问速率。 (2 分)

代码在执行过程, Cache 的作用没有发挥出来, 因为数据块大小刚好就是一个整数, 所构建的存储体系没有体现局部性的思想, 导致 CPU 对数据的访问没有一次能在 Cache 中命中。(1 分)

可行的办法: 提高数据块的大小 (2 分)

| | |
|-----|--|
| 分 数 | |
| 评卷人 | |

四、下图为虚拟存储器的工作原理图。(14 分)



- 1) 页式虚拟存储器工作过程中涉及到 VA (虚拟地址)、PA (物理地址)、VPN (虚拟页号)、PPN (物理页号) 等概念。根据页式虚拟存储器的工作原理, 给出 VA、PA、VPN、PPN 在图中的编号。(4 分)

| | | | |
|----|---|-----|---|
| VA | 1 | VPN | 2 |
| PA | 4 | PPN | 3 |

- 2) 如果不使用 TLB 会导致什么问题, 简要说明原因?

如果不使用 TLB, 会降低存储系统的访问速率 (或增加存储系统的访问时间), 因为实现虚拟地址与物理地址的转换需要增加一次访问主存/高速缓冲存储器。(2 分)

- 3) 假定某虚拟页式存储器页大小为 1024B, 物理空间为 64KB。结合下表求对应于十进制虚拟地址 2050 和 3080 的主存物理地址 (十进制)。(第一列为有效位, 1 表示有效) (4 分)

| | |
|---|--------|
| 1 | 000010 |
| 1 | 000110 |
| 1 | 000111 |
| 0 | 000100 |

页表

| VA (10 进制) | PA (10 进制) |
|------------|------------|
| 2050 | 7170 |
| 3080 | 缺页 |

根据虚拟页式存储器页面大小 1024B, 可知页内偏移地址为 10 位;

$(2050)_{10} = (10\ 000000010)_B$, 对应的虚页号为 2, 查页表得到物理页号为 000111, 且有效位为 1, 因此可得到物理地址为: $(000111\ 0000000010)_B = 7170$

$(3080)_{10} = (11\ 0000001000)_B$, 对应的虚页号为 3, 查页表得该页有效位为 0, 因此本次访问实效, 本次访问不能获得与虚拟十进制地址 3080 对应的物理地址, 将发生缺页异常。

- 4) 访问 TLB 不命中时一定会发生缺页异常吗? 如果 TLB 命中, cache 是否一定命中? 简单分析原因。

TLB 不命中, 只是表示对应页表项不在 TLB 中, 只有访问主存系统中的页表项提示缺页时才会发生缺页异常; TLB 命中, 只能说明要访问的页在主存, 页载入主存和数据块载入 cache 并不同步, 所以 cache 有可能命中, 也有可能缺失。 4 分

| | |
|-----|--|
| 分 数 | |
| 评卷人 | |

五、某计算机采用 16 位定长指令字格式,其 CPU 中有一个标志寄存器,其中包含进位/借位标志 CF、零标志 ZF 和符号标志 NF。假定为该机设计了条件转移指令,其格式如下:

| | | | | |
|-------|----|---|---|--------|
| 15~11 | 10 | 9 | 8 | 07~00 |
| 00000 | C | Z | N | OFFSET |

其中, 00000 为操作码 OP; C、Z 和 N 分别为 CF、ZF 和 NF 的对应检测位, 某检测位为 1 时表示需检测对应标志, 需检测的标志位中只要有一个为 1 就转移, 否则不转移, 例如, 若 C=1, Z=0, N=1, 则需检测 CF 和 NF 的值, 当 CF=1 或 NF=1 时发生转移; OFFSET 是相对偏移量, 用补码表示。转移执行时, 转移目标地址为 (PC)+2+OFFSET×2; 顺序执行时, 下条指令地址为 (PC)+2。请回答下列问题。(11 分)

1) 该计算机存储器按字节编址还是按字编址? 该条件转移指令向前和向后最多可跳转多少条指令?

按字节编址 (2 分)

Offset 采用 8 位补码表示, 所以表示范围位-128~127, 可以向前跳跃 128, 向后跳跃 127 条指令 (1 分)

2) 某条件转移指令的地址为 200CH, 指令内容如下图所示, 若该指令执行时 CF=0, ZF=0, NF=1, 则该指令执行后 PC 的值是多少? 若该指令执行时 CF=1, ZF=0, NF=0, 则该指令执行后 PC 的值又是多少? 请给出计算过程。

| | | | | |
|-------|----|---|---|-----------------|
| 15~11 | 10 | 9 | 8 | 07~00 |
| 00000 | 0 | 1 | 1 | 1 1 1 0 0 0 1 1 |

指令中 C = 0, Z = 1, N = 1, 故应根据 ZF 和 NF 的值来判断是否转移。当 CF=0, ZF=0, NF=1 时, 需转移。(1 分)

已知指令中偏移量为 1110 0011B=E3H, 符号扩展后为 FFE3 H, 左移一位 (乘 2) 后为 FFC6 H, 故 PC 的值 (即转移目标地址) 为 200CH+2+FFC6H=1FD4H。(2 分)

当 CF = 1, ZF = 0, NF = 0 时不转移。(1 分)

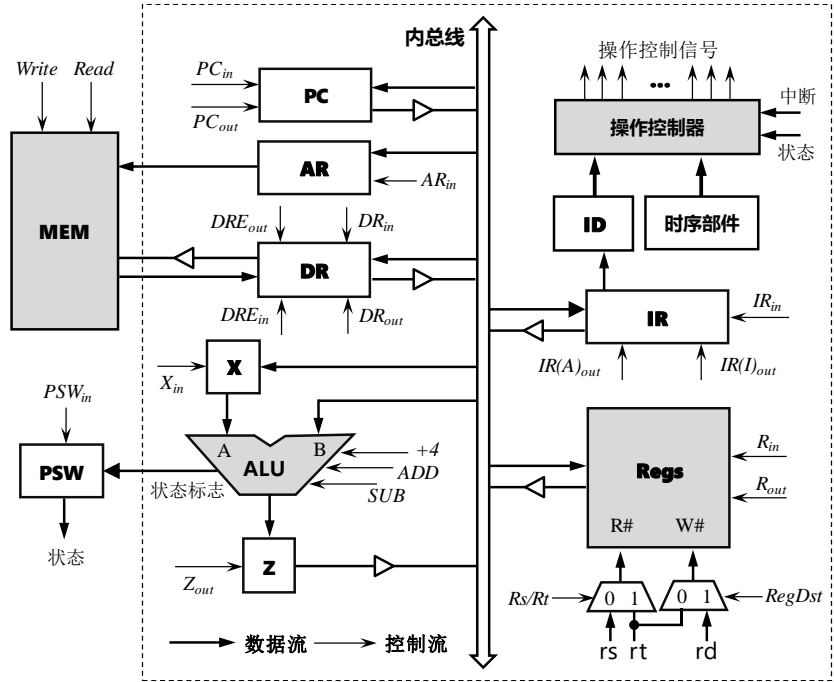
PC 的值为: 200CH+2=200EH。(1 分)

3) 实现“无符号数比较小于等于时转移”功能的指令中, C、Z 和 N 应各是什么?

C=Z=1, N=0 (3 分)

| | |
|-----|--|
| 分 数 | |
| 评卷人 | |

六、某计算机字长 32 位，支持下表中的五条 MIPS32 指令，CPU 内部采用单总线结构，具体数据通路如图所示。除多路选择器选择控制信号外，图中所有控制信号为 1 时表示有效、为 0 时表示无效。例如，控制信号 PC_{in} 为 1 表示允许数据从内总线输入 PC， PC_{out} 为 1 时表示允许数据从 PC 寄存器输出到内总线。假设 Z 寄存器的输入一直处于使能状态。（16 分）



| # | MIPS 指令 | RTL 功能描述 |
|---|----------------|---|
| 1 | lw rt,imm(rs) | $R[rt] \leftarrow M[R[rs] + \text{SignExt}(imm)]$ |
| 2 | sw rt,imm(rs) | $M[R[rs] + \text{SignExt}(imm)] \leftarrow R[rt]$ |
| 3 | beq rs,rt,imm | if($R[rs] == R[rt]$) $PC \leftarrow PC + 4 + \text{SignExt}(imm) \ll 2$ |
| 4 | addi rt,rs,imm | $R[rt] \leftarrow R[rs] + \text{SignExt}(imm)$ |
| 5 | add rd,rs,rt | $R[rd] \leftarrow R[rs] + R[rt]$ |

1) 根据主机数据通路图的信息请给出 *beq* 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（仅给出为 1 的信号）。

1) 取指令阶段 (5 分)

| 时钟 | 数据通路 | 控制信号 |
|----|--|------------------------------------|
| T1 | $PC \rightarrow AR, PC \rightarrow X$ | $PC_{out}, AR_{in}, X_{in}$ |
| T2 | $X + 4 \rightarrow Z$ | $+4$ |
| T3 | $Z \rightarrow PC, M[AR] \rightarrow DR$ | $Z_{out}, PC_{in}, DRE_{in}, Read$ |
| T4 | $DR \rightarrow IR$ | DR_{out}, IR_{in} |

2) 执行指令阶段(5 分)

| 时钟 | 数据通路 | 控制信号 |
|----|--------------------------------------|---------------------------------|
| T1 | $R[rs] \rightarrow X$ | R_{out}, X_{in} |
| T2 | $X - R[rt] \rightarrow PSW$ | $R_{out}, Rs/Rt, SUB, PSW_{in}$ |
| T3 | $PC \rightarrow X$ | PC_{out}, X_{in} |
| T4 | $IR(A) + X \rightarrow Z$ | $IR(A)_{out}, ADD$ |
| T5 | If (PSW.equal) $Z \rightarrow PC$ | $Z_{out}, PC_{in} = PSW.equal$ |

- 2) 常见 MIPS 指令实现中是没有程序状态寄存器 PSW 的，为什么在本题中的 CPU 架构中需要设计程序状态字 PSW？

beq 指令需要分别计算比较结果和分支地址，本题的主机架构中只有一个运算器，所以需要两次使用运算器，因此必须暂存比较结果于 PSW 中。 (3 分)

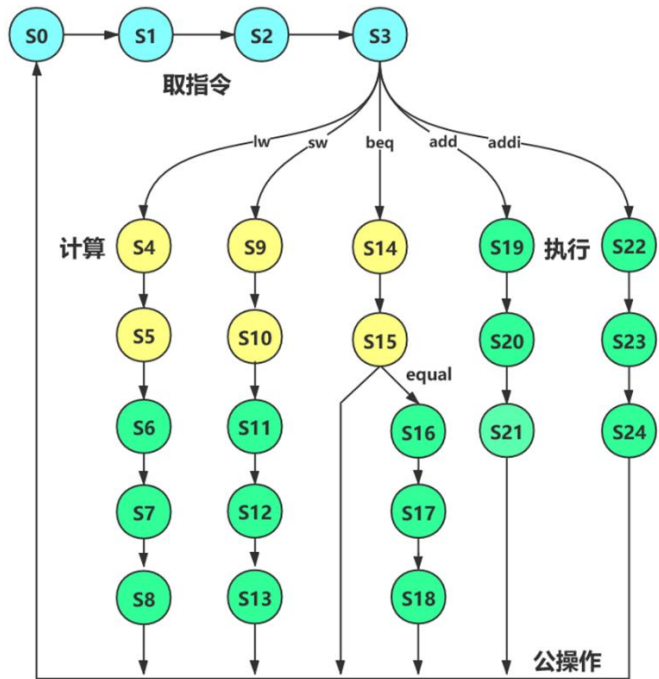
- 3) 在定长指令周期三级时序单总线 CPU 实验中，测试程序预期功能是在 0x80 的内存数据单元进行排序，请问这个排序是降序还是升序，是有符号比较还是无符号比较？为什么实际 Educoder 平台上通关的结果是在 0x00 处进行内存单元数据排序的，而且代码区部分代码会被覆盖？

降序排序，有符号比较 (2 分)

Z 寄存器没有锁存控制，才用定长周期时，刚刚计算完地址应该直接送 AR，但由于计算周期和执行周期之间插入了空周期，所以导致送入 AR 中的地址错误。 (1 分)

| | |
|-----|--|
| 分 数 | |
| 评卷人 | |

七、对于上题中的 CPU 数据通路，如果采用现代时序方式，其指令执行状态图如下所示，如果采用微程序方式实现控制器，尝试回答如下问题。(17 分)



1) 如果采用下址字段法，直接表示的水平微指令，则微指令操作控制字段长度为 21/22 位，判别测试位至少 3 位，下址字段至少为 5 位，实现上题中五条机器指令共需要多少条 25 微指令。如果改用计数器法，则判别测试位至少 3 位。如果采用对微指令字中的操作控制字段采用编码方式缩短字长，则该字段最短为 18 位。

(最后一空 2 分，其他各 1 分，共 7 分)

2) 如果要为该 CPU 增加单级中断处理机制，需要增加哪些硬件单元，简要叙述增加的硬件单元的功能。

- EPC 保存断点
- 中断使能寄存器 IE 开关中断
- 中断识别控制逻辑 中断识别 (6 分)

