

6.2 选择题

(1) [2010] 下列寄存器中， 汇编语言程序员可见的是(B) 。

- 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)

(2) [2019] 某指令功能为 $R[r2] \leftarrow R[r1] + M[R[r0]]$ ， 其两个源操作数分别采用寄存器、 寄存器间接寻址方式。 对于下列给定部件， 该指令在取数及执行过程中需要用到的是(B) 。

I . 通用寄存器组 (GPRs) II . 算术逻辑单元 (ALU) III . 存储器 (Memory) IV . 指令译码器 (ID)

- A . 仅 I 、 II B . 仅 I、 II 、 III
- C . 仅 II 、 III、 IV D . 仅 I 、 II 、 IV

(3) [2016] 某计算机主存空间为 4 GB， 字长为 32 位， 按字节编址， 采用 32 位定长指令字格式。 若指令按字边界对齐存放， 则程序计数器 (PC) 和指令寄存器 (IR) 的位数至少分别是(B) 。

- A . 30、 30 B . 30、 32 C . 32、 30 D . 32、 32

(4) [2019] 下列有关处理器时钟脉冲信号的叙述中， 错误的是(D) 。

- A . 时钟脉冲信号由机器脉冲源发出的脉冲信号经整形和分频后形成
- B . 时钟脉冲信号的宽度称为时钟周期， 时钟周期的倒数为机器主频
- C . 时钟周期以相邻状态单元间组合逻辑电路的最大延迟为基准确定
- D . 处理器总是在每来一个时钟脉冲信号时就开始执行一条新的指令

(5) [2016] 单周期处理器中所有指令的指令周期为一个时钟周期。 下列关于单周期处理器的叙述中， 错误的是(A) 。

- A . 可以采用单总线结构数据通路 B . 处理器时钟频率较低
- C . 在指令执行过程中控制信号不变 D . 每条指令的 CPI 为 1

(6) [2017] 下列关于主存 (MM) 和控制存储器 (CS) 的叙述中， 错误的是(B) 。

- A . MM 在 CPU 外, CS 在 CPU 内
- B . MM 按地址访问, CS 按内容访问
- C . MM 存储指令和数据, CS 存储微指令
- D . MM 用 RAM 和 ROM 实现, CS 用 ROM 实现

(7) [2009] 相对于微程序控制器, 硬布线控制器的特点是(D) 。

- A . 指令执行速度慢, 指令功能的修改和扩展容易
- B . 指令执行速度慢, 指令功能的修改和扩展难
- C . 指令执行速度快, 指令功能的修改和扩展容易
- D . 指令执行速度快, 指令功能的修改和扩展难

(8) [2012] 某计算机的控制器采用微程序控制方式, 微指令中的操作控制字段采用字段直接编码法, 共有 33 个微命令, 构成 5 个互斥类, 分别包含 7、3、12、5 和 6 个微命令, 则操作控制字段至少有(C) 。

- A . 5 位 B . 6 位 C . 15 位 D . 33 位

(9) [2014] 某计算机采用微程序控制器, 共有 32 条指令, 公共的取指令微程序包含两条微指令, 各指令对应的微程序平均由 4 条微指令组成, 采用断定法(下址字段法) 确定下条微指令地址, 则微指令中下址字段的位数至少是(C) 。

- A . 5 B . 6 C . 8 D . 9

(10) [2011] 假定不采用 cache 和指令预取技术, 且计算机处于“开中断”状态, 则在下列有关指令执行的叙述中, 错误的是(C) 。

- A . 每个指令周期中 CPU 都至少访问内存一次
- B . 每个指令周期一定大于或等于一个 CPU 时钟周期
- C . 空操作指令的指令周期中任何寄存器的内容都不会被改变
- D . 当前程序在每条指令执行结束时都可能被外部中断打断

6.4 某 CPU 的结构如图 6.69 所示, 其中 AC 为累加器, 条件状态寄存器保存指令执行过程中的状态。a、b、c、d 为 4 个寄存器。图中箭头表示信息传送的方向, 试完成下列各题。

(1) 根据 CPU 的功能和结构标明图中 4 个寄存器的名称。

(2) 简述指令 LDA addr 的数据通路, 其中 addr 为主存地址, 指令的功能是

表 6.2 典型 MIPS32 指令

序号	指令	汇编代码	指令类型	RTL 功能说明
1	lw	lw rt,imm(rs)	I 型	$R[rt] \leftarrow M[R[rs] + \text{SignExt}(imm)]$
2	sw	sw rt,imm(rs)	I 型	$M[R[rs] + \text{SignExt}(imm)] \leftarrow R[rt]$
3	beq	beq rs,rt,imm	I 型	if($R[rs] == R[rt]$) $PC \leftarrow PC + 4 + \text{SignExt}(imm) \ll 2$
4	addi	addi rt,rs,imm	I 型	$R[rt] \leftarrow R[rs] + \text{SignExt}(imm)$
5	add	add rd,rs,rt	R 型	$R[rd] \leftarrow R[rs] + R[rt]$
6	slt	slt rd,rs,rt	R 型	$R[rd] \leftarrow (R[rs] < R[rt]) ? 1 : 0$
7	j	j imm26	J 型	$PC \leftarrow \{(PC+4)_{31:28}, imm26 \ll 2\}$

答：(1) 恒 0 故障：指令 lw、addi、add、slt 出错。因为这些指令需要向寄存器堆写入数据 (RegWrite = 1)，若 RegWrite 恒 0，无法执行写操作，寄存器内容无法更新。

恒 1 故障：指令 sw、beq、j 出错。这些指令本不需要写寄存器 (RegWrite = 0)，但 RegWrite 恒 1 会强制写寄存器，破坏寄存器原有内容。

(3) 恒 0 故障：指令 sw 出错。sw 需写内存 (MemWrite = 1)，若 MemWrite 恒 0，无法执行写操作，内存数据无法更新。

恒 1 故障：指令 lw、addi、add、slt、beq、j 出错。这些指令本不需要写内存 (MemWrite = 0)，但 MemWrite 恒 1 会强制写内存，破坏内存数据。例如，lw 本为读内存，却错误执行写操作。

6.10 假设构成 CPU 的各功能部件的时间延迟如表 6.21 所示，试分别计算单周期、多周期 MIPS 处理器的最小时钟周期和最大时钟频率。假设某 MIPS 程序包含 1000 亿条指令，其中 lw、sw、beg、R 型算术逻辑运算、I 型算术逻辑运算指令比例分别为 10%、10%、10%、50%、20%，试分别计算该程序在单总线结构处理器、单周期 MIPS、多周期 MIPS 处理器上的 CPI 值及执行时间。

表 6.21 各功能部件的时间延迟

功能部件	参数	延迟	功能部件	参数	延迟
寄存器延迟	$T_{clk_to_q}$	20 ps	运算器 ALU	T_{alu}	90 ps
存储器读	T_{mem}	150 ps	多路选择器	T_{mux}	20 ps
寄存器堆读	T_{RF_read}	90 ps	寄存器建立时间	T_{setup}	10 ps

答：(1)

6.10
单总线MIPS处理器: $T_{min_clk} = T_{clk_to_q} + \max(T_{alu}, T_{mem}) + T_{setup}$
 $= 20 + \max(40, 150) + 10 = 180 \text{ ps}$

$$f_{max_freq} = \frac{1}{180 \times 10^{-12}} = 5.56 \text{ GHz}$$

单周期MIPS处理器: $T_{min_clk} = T_{clk_to_q} + T_{alu} + T_{mem} + T_{rf_read} + T_{mux} + T_{setup}$
 $= 20 + 90 + 2 \times 150 + 90 + 20 + 10 = 530 \text{ ps}$

$$f_{max_freq} = \frac{1}{530 \times 10^{-12}} = 1.89 \text{ GHz}$$

多周期MIPS处理器: $T_{min_clk} = T_{clk_to_q} + T_{mux} + \max(T_{alu} + T_{mux}, T_{mem}) + T_{setup}$
 $= 20 + 20 + \max(40 + 20, 150) + 10 = 200 \text{ ps}$

$$f_{max_freq} = \frac{1}{200 \times 10^{-12}} = 5 \text{ GHz}$$

(2) 对于单总线 MIPS 处理器, lw、sw、beq、R 型运算、I 型运算指令的 CPI 分别为 9、9、9、7、7, 因此 $CPI = 9 \times 0.1 + 9 \times 0.1 + 9 \times 0.1 + 7 \times 0.5 + 7 \times 0.2 = 7.6$, $T_{cpu} = 1000 \times 10^8 \times 7.6 \times 180 \times 10^{-12} = 136.8 \text{ s}$ 。

对于单周期 MIPS 处理器, $CPI = 1$, $T_{cpu} = 1000 \times 10^8 \times 1 \times 530 \times 10^{-12} = 53 \text{ s}$ 。

对于多周期 MIPS 处理器, lw、sw、beq、R 型运算、I 型运算指令的 CPI 分别为 5、4、3、4、4, 因此 $CPI = 5 \times 0.1 + 4 \times 0.1 + 3 \times 0.1 + 4 \times 0.5 + 4 \times 0.2 = 4$, $T_{cpu} = 1000 \times 10^8 \times 4 \times 200 \times 10^{-12} = 80 \text{ s}$ 。

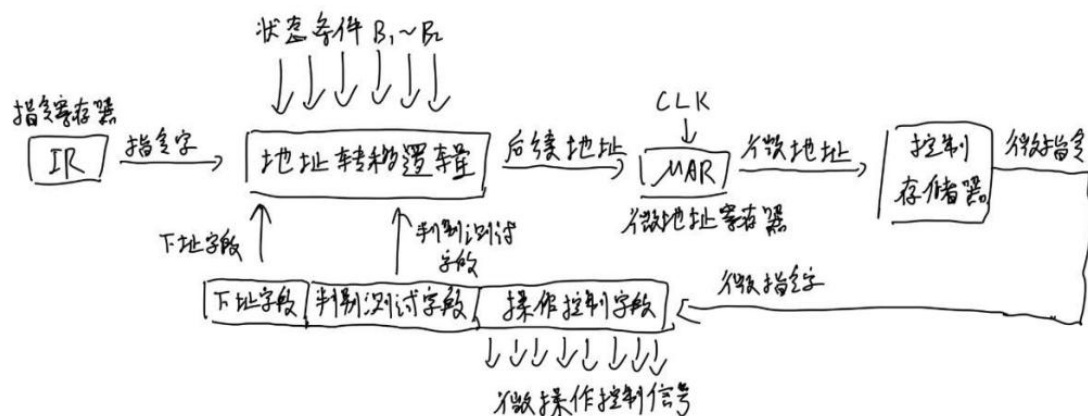
6.20 已知某计算机采用微程序控制方式, 控制存储器容量为 128×32 位。微程序可在整个控制存储器中实现分支跳转, 控制微程序判别测试条件共 3 个, 微指令采用水平型格式, 后续微指令地址采用下址字段法。回答下列问题。

(1) 微指令的 3 个字段分别应为多少位?

(2) 画出对应这种微指令格式的微程序控制器逻辑框图。

答: (1) 控制存储器容量 128×32 位 $= 2^7 \times 32$ 位, 因此下址字段为 7 位, 判别测试条件为 3 位, 所以操作控制字段为 $32 - 7 - 3 = 22$ 位。

(2) 跟课本 6.49 图一样



6.21 某微程序包含 5 条微指令，每条微指令发出的操作控制信号如表 6.22 所示，试对这些微指令进行编码，要求微指令的控制字段最短且能保持微指令应有的并行性。

表 6.22 微指令及其对应的微操作控制信号

微指令	微操作控制信号	微指令	微操作控制信号	微指令	微操作控制信号
μI_1	a, c, e, g	μI_3	a, d, e	μI_5	a, d, f, j
μI_2	a, d, f, h, j	μI_4	a, b, i		

答：为使微指令控制字段最短且保持并行性，采用字段直接编码，将互斥信号分组：

组 1 (c、g)：2 位，01 表示同时有效（仅 μI_1 ）。

组 2 (b、i)：2 位，10 表示同时有效（仅 μI_4 ）。

组 3 (e、h)：1 位，1 表示 e 有效 (μI_1 、 μI_3)，0 表示 h 有效 (μI_2)。

d：1 位， μI_2 、 μI_3 、 μI_5 时有效。

f：1 位， μI_2 、 μI_5 时有效。

j：1 位， μI_2 、 μI_5 时有效。

最终编码如下：

微指令	控制字段编码 (组 1 + 组 2 + e/h + d + f + j)
μI_1	01 00 1 0 0 0
μI_2	00 00 0 1 1 1
μI_3	00 00 1 1 0 0

微指令	控制字段编码 (组 1 + 组 2 + e/h + d + f + j)
μl_4	00 10 0 0 0 0
μl_5	00 00 0 1 1 1

6.23 某计算机字长为 16 位，采用 16 位定长指令字结构，部分数据通路结构如图 6.70 所示，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 MDRE 为 1 表示允许数据从 DB 送入 MDR 中，MDR 为 1 表示允许数据从内总线送入 MDR 中。假设 MAR 的输出一直处于使能状态。加法指令“ADD(RI),RO”的功能为 $(RO) + ((R1)) \rightarrow (R1)$ ，即将 R0 中的数据与 R1 内容所指主存单元的数据相加，并将结果送入 R1 内容所指的主存单元中保存。

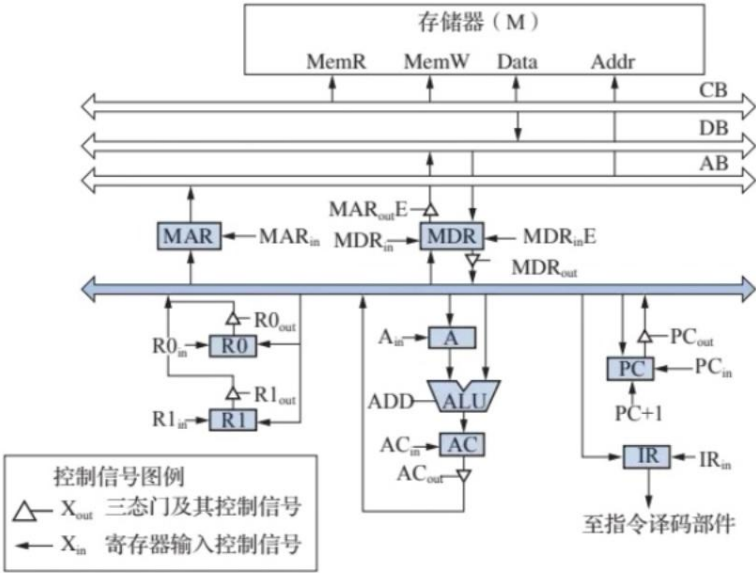


图 6.70 某计算机数据通路图

表 6.23 所示为上述指令取指和译码阶段每个节拍（时钟周期）的功能和有效控制信号，请按表中描述的方式用表格列出指令执行阶段每个节拍的功能和有效控制信号。

表 6.23 取指周期的功能与信号

时钟	功能	有效控制信号
C1	$MAR \leftarrow (PC)$	PC_{out} , MAR_{in}
C2	$MDR \leftarrow M(MDR)$ $PC \leftarrow (PC)+1$	MemR, MDR_{inE} , $PC+1$
C3	$IR \leftarrow (MDR)$	MDR_{out} , IR_{in}
C4	指令译码	无

答：如下表所示

时钟	功能	有效控制信号
C5	MAR(R1)	R1_out,MAR_in
C6	MDR←M(MAR)	MemR,MDR_inE
C6	A←(R0)	R0_out,A_in
C7	AC←(MDR)+(A)	MDR_out,Add,AC_in
C8	MDR←(AC)	AC_out,MDR_in
C9	M(MAR)←(MDR)	MDR_outE,MemW

6.24 某 16 位计算机的主存按字节编址，存取单位为 16 位；采用 16 位定长指令字格式；CPU 采用

单总线结构，主要部分如图 6.71 所示。图中 R0 ~ R3 为通用寄存器；T 为暂存器；SR 为移位寄存器，可实现直送 (mov)、左移一位 (left) 和右移一位 (right) 3 种操作，控制信号为 SROP，SR 的输出由信号 SRout 控制；ALU 可实现直送 A (mova)、A 加 B (add)、A 减 B (sub)、A 与 B (and)、A 或 B (or)、非 A (not)、A 加 1 (inc) 7 种操作，控制信号为 ALUop。请回答下列问题。

- (1) 图中哪些寄存器是程序员可见的？为何要设置暂存器 T？
- (2) 控制信号 ALUop 和 SROP 的位数至少各是多少？
- (3) 控制信号 SRout 控制部件的名称或作用是什么？
- (4) 端点①~⑨中，哪些端点须连接到控制部件的输出端？
- (5) 为完善单总线数据通路，需要在端点①~⑨中相应的端点之间添加必要的连线。写出连线的起点和终点，以正确表示数据的流动方向。
- (6) 为什么二路选择器 MUX 的一个输入端是 2？

答：(1) 通用寄存器(R0~R3)和 PC 程序员可见。因为采用了单总线结构，若无暂存器 T，则 ALU 的 A、B 端口会同时获得两个相同的数据，数据通路不能正常工作。增加暂存器后可以使 A、B 端口输入不同的数据

(2) ALU 共有 7 种操作，故操作控制信号 ALUop 至少需要 3 位；移位寄存器有 3 种操作，其操作控制信号 SROP 至少需要 2 位

(3) 控制的部件是三态门，用于控制移位器和总线之间数据通路的连接与断开

(4) 端口①、②、③、⑤、⑧都是控制信号，需要连接到控制部件输出端，其他都是数据通路

(5) 首先内总线数据需要送 MUX 的 1 号引脚，因此⑥连接⑨；另外多路选择器输出应该连接到运算器 B 端，所以④连接⑦

(6) 因为每条指令的字节长度是 2，顺序寻址时应该实现(PC)+2 的逻辑，所以这里 MUX 的一个输入端为 2，便于执行(PC)+2 操作