计组手写作业6

6.2 选择题
1-5.BBBDA 6-10.BDCCC

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，指令的功能是将主存** `addr` **单元的内容送入AC中。**

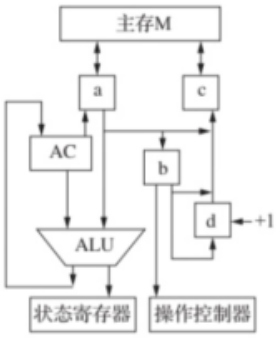


图 6.69　某 CPU 的结构框图

(1)d寄存器进行+1操作，所以`d为PC`，a、c连接主存和CPU，所以证明是AR和DR，c与d相连证明`c是AR`，`a是DR`，剩下`b是IR`
(2)取指数据通路：PC->AR->主存M->DR->IR,PC+1->PC
　执行数据通路：IR->AR->主存M->DR->ALU->AC

6.6假设图6.25所示的单周期MIPS处理器中，操作控制器输出某个控制信号时发生了恒0故障,表6.2中的哪些指令会发生错误呢?为什么?如果是恒1故障呢?
(1) RegWrite;(2) RegDst;(3) MemWrite。

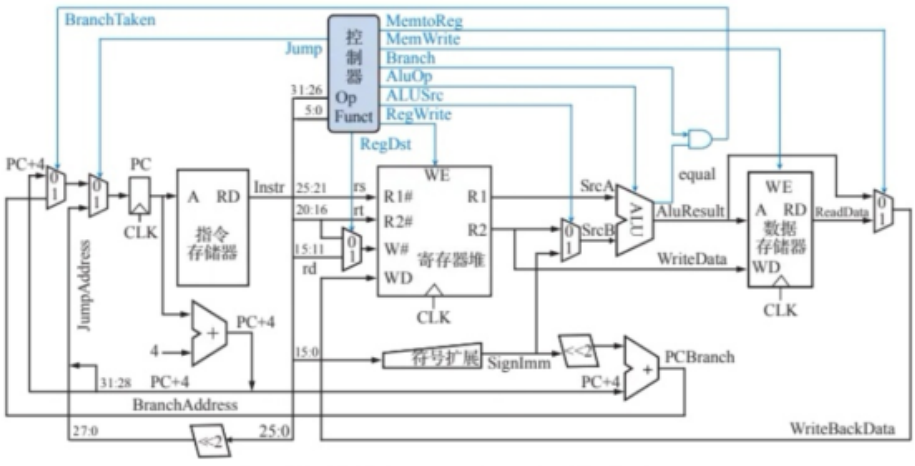


图 6.25　单周期 MIPS 处理器的数据通路高层视图

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] ← M[R[rs] + SignExt(imm)]
2	sw	sw rt,imm(rs)	I 型	M[R[rs] + SignExt(imm)] ← R[rt]
3	beq	beq rs,rt,imm	I 型	if(R[rs] ==R[rt])　　PC ← PC + 4 + SignExt(imm) << 2
4	addi	addi rt,rs,imm	I 型	R[rt] ← R[rs] + SignExt(imm)
5	add	add rd,rs,rt	R 型	R[rd] ← R[rs] + R[rt]
6	slt	slt rd,rs,rt	R 型	R[rd] ← (R[rs] < R[rt])?1:0
7	j	j imm26	J 型	PC ← {(PC+4)$_{31:28}$,imm26<<2}

(1)RegWrite恒为0时，寄存器无法写入，所以需要写回寄存器的指令无法实现，即lw、addi、add、slt出问题;RegWrite恒为1时，所有最后不写回通用寄存器的指令都会出错，即beq、sw和j指令，因为它们可能向寄存器堆中对应的寄存器写入错误的数据
(2)RegDst恒为0时，会一直将写入的目的寄存器设为rt，所以add、slt出现问题；恒为1时，写入的目的寄存器一直为rd，　因此addi、lw出现问题
(3)MemWrite恒为0时，无法将数据写入存储器，因此指令sw出现问题;恒为1时，没有更新存储器的指令都会出错，即lw、beq、addi、add、alt、j指令

6.9修改图6.25所示的单周期MIPS处理器，使其能够支持如下MIPS指令，具体指令功能请查阅MIPS32指令手册。试描述需要增加修改哪些数据通路和控制信号，尝试给出各指令的执行流程和每一步的操作控制信号。
(2) lui;

(2)`lui rt,imm`指令实现`R[rt]=imm<<16`,因此需要增加一条新的数据通路在指令译码阶段获取指令中的立即数，之后再进行符号扩展，之后利用移位器左移16位，接回寄存器堆的`WD`引脚，同时在接入`WD`引脚前加一个多路选择器来选择`WriteBackData`和运算得到的立即数，同时增加一个控制信号`WDdate`来控制该多路选择器的输出

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、beq、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

单总线：lw、sw、beq的CPI是9，R型指令和I型指令CPI是7，因此平均
$CPI = 9 \times (0.1 + 0.1 + 0.1) + 7 \times (0.5 + 0.2) = 7.6$
$T_{min\_clk} = T_{clk\_to\_q} + max(T_{ALU}, T_{mem}) + T_{setup} = 20 + max(150, 90) + 10 = 180ps$

$T_{CPU} = CPI \times T_{min\_clk} \times IC = 7.6 \times 10^{12} \times 180 \times 10^{-12}s = 1368s$

单周期：CPI固定为1，
$T_{min\_clk} = T_{clk\_to\_q} + 2T_{mem} + T_{RF\_read} + T_{ALU} + T_{MUX} + T_{setup} = 20 + 300 + 90 + 90 + 20 + 10 =$
$T_{CPU} = CPI \times T_{min\_clk} \times IC = 1 \times 10^{12} \times 530 \times 10^{-12}s = 530s$

多周期：lw的CPI是5，sw、R型指令、I型指令的CPI是4，beq的CPI是3，所以平均

$CPI = 5 \times 0.1 + 4 \times (0.1 + 0.5 + 0.2) + 3 \times 0.1 = 4$

$T_{min\_clk} = T_{clk\_to\_q} + T_{MUX} + max(T_{ALU} + T_{MUX}, T_{mem}) + T_{setup} = 20 + 20 + max(110, 150) + 1($

$T_{CPU} = CPI \times T_{min\_clk} \times IC = 4 \times 10^{12} \times 200 \times 10^{-12} s = 800s$

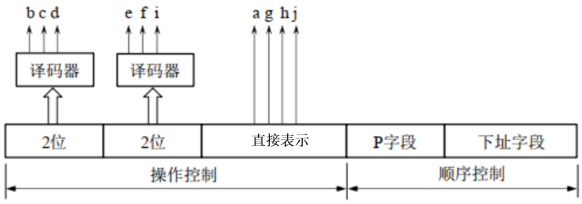
6.21某微程序包含**5**条微指令，每条微指令发出的操作控制信号如表**6.22**所示，试对这些微指令进行编码，要求微指令的控制字段最短且能保持微指令应有的并行性。

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		

为了使控制字段最短，因此先使用编码表示,先计算互斥组

根据互斥组定义可知(b,c,d),(f,g,i)是两组互斥组，它们组内最多只有一个控制信号为1，之后剩下的四个4微指令直接表示，因为它们都不满足互斥性，所以最后的编址方案为



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$MDR_{in}E$为**1**表示允许数据从**DB**送入**MDR**中，MDR_{in}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(R1),R0`的功能为`(R0)+((R1))→(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。

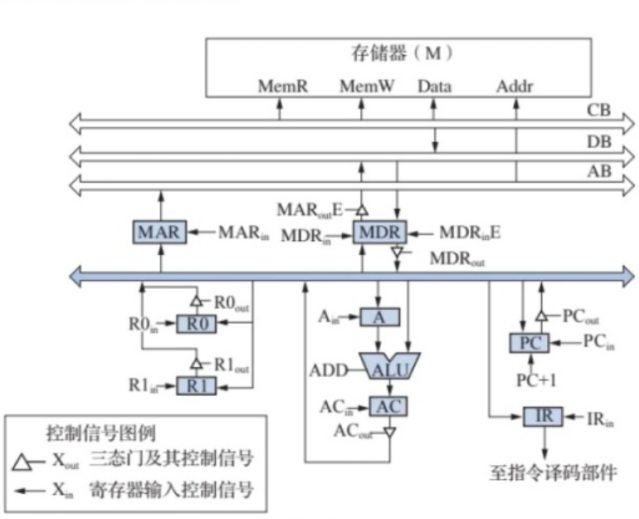


图 6.70　某计算机数据通路图

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 ← (PC)	PC_{out}, MAR_{in}
C2	MDR ← M(MDR) PC ← (PC)+1	MemR, $MDR_{in}E$, PC+1
C3	IR ← (MDR)	MDR_{out}, IR_{in}
C4	指令译码	无

时钟	功能	有效控制信号
C5	MAR←(R1)	$R1_{out}, MAR_{in}$
C6	MDR←M(MAR),A←(R0)	$R0_{out}, A_{in}, MemR, MDR_{in}E$
C7	ALU←(MDR)	MDR_{out}, ADD, AC_{in}
C8	MDR←AC	AC_{out}, MDR_{in}
C9	M(MAR)←MDR	$MemW, MAR_{out}E$

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SR_{out}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SR_{out}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?**

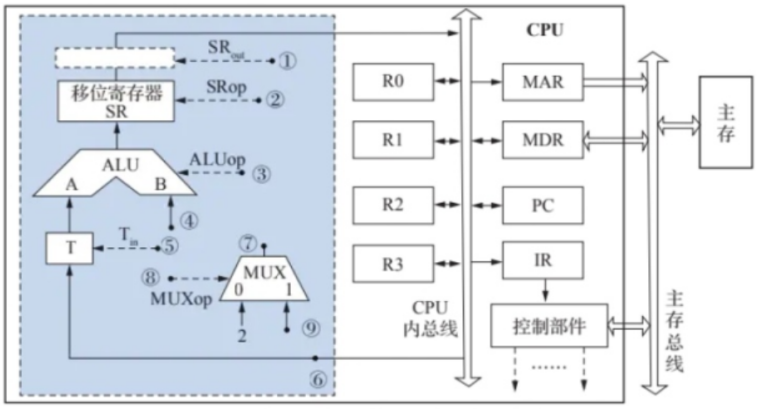


图 6.71　某 16 位计算机的部分数据通路

(1)R0,R1,R2,R3和PC程序员可见，因为采用了单总线结构，ALU的两个端口无法同时获取两个不同的输入数据，因此必须将先到达的数据进行锁存

(2)ALIop至少需要3位，SRop至少需要2位

(3)三态门，用于控制移位寄存器的输出是否流向内总线

(4)①②③⑤⑧

(5)⑥和⑨相连，④和⑦相连

(6)该计算机按照字节编址，因此一个存储单元存储8位，又采用16位定长指令，所以存储一

条指令需要两个存储单元，所以每执行完一条指令需要进行PC←(PC)+2,所以MUX的一个输入端是2