6.1 解释下列名词

机器周期：通常将从主存取出一个存储字所需的最短时间定义为机器周期

同步控制：又称固定时序方式，其基本思想是选取各部件中最长的操作时间作为统一的时间间隔标准，使所有部件都在这个时间间隔内启动并完成操作。通常利用同步时序发生器产生固定的周而复始的状态周期电位、节拍电位，再用这些统一的时序信号对各种操作定时，实现同步控制。同步控制时序关系比较简单，控制器设计方便,但存在慢速部件时 CPU 效率较低的问题。

异步控制：异步控制方式没有统一时钟信号，没有固定的机器周期和节拍，若采用异步时序进行控制，执行每条指令需要多长时间就占用多少时间。各功能部件及操作的时序采用应答机制实现，控制部件发出操作控制信号给功能部件后，必须等到功能部件发出应答信号才能开始下一步的操作。例如 CPU 访问主存时，可以采用主存的“READY”应答信号作为读写周期的结束信号。异步控制方式的优点是每个部件都可按各自实际需要的时间工作，没有快者等待慢者的过程，从而提高了系统的速度，但是异步控制方式结构更加复杂。

时序发生器：传统三级时序中时序发生器主要根据时钟脉冲信号持续不断地产生状态周期电位和节拍电位，操作控制器利用这些周期、节拍电位信号对操作控制信号进行时序的调制，生成控制信号序列。通常将状态周期电位信号、节拍电位信号与指令译码信号、反馈信号等进行适当的逻辑与操作，从而限定操作控制信号的开始时间和持续时间。

微命令：控制部件向执行部件发出的各种控制命令。

微操作：执行部件收到微命令后所进行的操作。

微程序控制器：采用微程序控制方式的控制器。

控制存储器：存放仿照程序设计的基本方法，将实现指令系统中所有指令功能所需要的所有控制信号按照一定的规则编码成微指令，若干条实现同一条指令功能的微指令构成一段微程序，实现所有指令的微程序的只读存储器。

6.2

(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]←- 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

指令译码器在指令分析时使用，在取数和执行过程中不使用。

(3)[2016]某计算机主存空间为 4GB，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

因为是 32 位定长指令，所以指令寄存器(IR)至少 32 位；32 位是 4B，4GB/4B = 2^{30}，又因为指令按字边界对齐存放，所以至少 30 位

(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
每个机器周期指令一条指令，而不是时钟脉冲

(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
单周期处理器在一个时钟周期内完成一条指令需要专用的数据通路，单总线结构数据通路一个时钟周期只能传递

(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
CS 按地址访问

(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
硬布线控制器采用硬连线逻辑，故一旦构成，除非在物理上进行重新布线，否则指令功能无法修改和扩展。所以指令功能的修改和扩展难。
微程序控制器采用存储逻辑，当需要对指令功能进行修改和扩展时，只要重新设计微代码的码点，并将其注入控制存储器中即可；但是由于采用存储逻辑，相比硬布线控制器多了从控制存储器中读出码点的过程, 故其执行速度较慢。

(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
7 个微命令+1 个 nop 命令，需要 3 位
3 个微命令+1 个 nop 命令，需要 2 位
12 个微命令+1 个 nop 命令，需要 4 位
5 个微命令+1 个 nop 命令，需要 3 位
6 个微命令+1 个 nop 命令，需要 3 位
故需要 3+2+4+3+3 = 15 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

32*4 = 128,128+2=130 条，$2^7 < 130 < 2^8$

(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

空操作指令执行完后，程序计数器 PC 会自加一，会改变

6.3

(6) 比较单周期 MIPS 处理器与多周期 MIPS 处理器的差异。

1. 时钟周期

单周期处理器：每个指令在一个时钟周期内完成执行，因此时钟周期固定。

多周期处理器：指令的执行被划分为多个时钟周期，不同类型的指令可能需要不同数量的时钟周期来完成。

2. 性能

单周期处理器：性能较低，因为每个指令都需要一个时钟周期。

多周期处理器：性能较高，因为它可以充分利用硬件资源，允许多个指令在不同阶段同时执行。

3. 复杂性

单周期处理器：相对简单，因为每个指令的执行过程是固定的。

多周期处理器：复杂性较高，因为需要设计和管理多个时钟周期的执行。

4. 资源利用

单周期处理器：资源利用不高，因为一些硬件单元可能在某些时钟周期内处于空闲状态。

多周期处理器：更有效地利用了硬件资源，因为不同的指令可以在不同的时钟周期内使用相同的硬件单元。

(7) 组合逻辑控制器与微程序控制器各有什么特点?

组合逻辑优点：产生微命令速度快。缺点：设计不规整效率低，不易修改，扩展困难。

微程序优点：设计规整，效率高，易于修改和扩展，可靠性高，性价比高。缺点：速度慢，执行效率不高，没充分利用数据通路。

(8)说明程序与微程序、指令与微指令的异同。

微程序是多条微指令系列的集合，用于实现指令的功能，属于机器指令级别，对用于透明性不强，存放在 CPU 内的控制存储器中；程序则是为了完成某一应用功能所编写的指令（包括机器语言指令或高级语言指令）集合，属于高级语言级别，对用户的透明性好，运行时存放在计算机的主存中。指令是指挥计算机执行某种功能的命令，是构成程序的基本单位，由操作码和地址字段构成;而微指令则用于微程序控制器中产生指令执行过程中所需要的微命令，是构成微程序的基本单位，由操作控制字段，判别测试字段和下址字段等组成。

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。

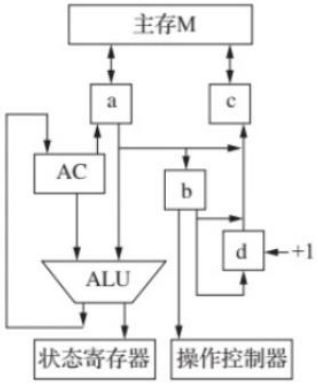


图 6.69　某 CPU 的结构框图

(1) 根据 CPU 的功能和结构标明图中 4 个寄存器的名标。

因为 d 是自加一，所以 d 是程序计数器 PC，由于 b 会影响程序计数器 d，所以 b 是指令寄存器 IR，d 的值会传入 c，c 由于 M 互相连接，所以 c 是地址寄存器 AR，a 与 ALU 相连且与主存相连，所以 a 是数据寄存器 DR

(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。

取指阶段：PC→AR→主存 M→DR→IR；PC→PC+1
执行阶段：IR(A)→AR→主存 M→DR→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?

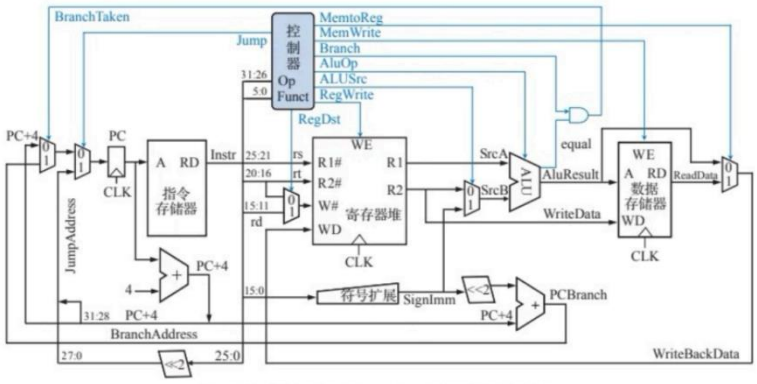


图 6.25　单周期 MIPS 处理器的数据通路高层视图

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 0：

lw,addi,add,slt 都需要写入寄存器堆，RegWrite 发生错误就无法正确写入
恒 1：
sw,beq,j 都需要访问寄存器堆，而 RegWrite 恒 1 代表写入则无法正确读出
(2) RegDst;
恒 0：
add,slt 都需要将 regdst 置 1，若其发生恒 0 故障则无法正确执行
恒 1：
lw,addi 都需要将 regdst 置 0，发生恒 1 故障，则无法正确运行
(3) MemWrite.
恒 0：
sw 需要写入 Mem，MemWrite 恒 0 错误则无法写入 Mem
恒 1：
lw,beq,addi,add,slt,j 都需要将 MemWrite 置 0，而其恒 1 则表示 Mem 正在写入，无法正确访问

6.7 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。试描述需要增加或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。

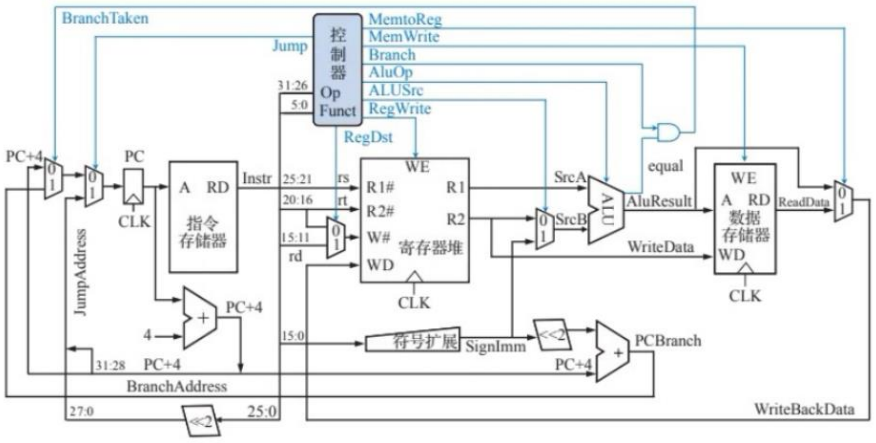


图 6.25 单周期 MIPS 处理器的数据通路高层视图

(4) Jal
取值周期：
1. $PC_{out} = AR_{in} = X_{in} = 1$
2. +4 = 1
3. $Z_{out} = PC_{in} = 1$, Read = $DRE_{in} = 1$
4. $DR_{out} = IR_{in} = 1$
执行周期：
1. $PC_{out} = X_{in} = AR_{in} = 1$
2. $IR(A)_{out} = J = 1$
3. $Z_{out} = PC_{in} = 1$
取值周期与其他指令相似，执行周期时，将 PC+4 的值先送入 AR，再送入 X，IR 和 J 此时控制 ALU 将 X 与 imm26 进行拼接，最后将拼接后的地址送入 PC，而 AR 则存储了 jal 下一条指令的地址，便于子程序调用后的返回

6.9 修改图 6.27 所示的多周期 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。

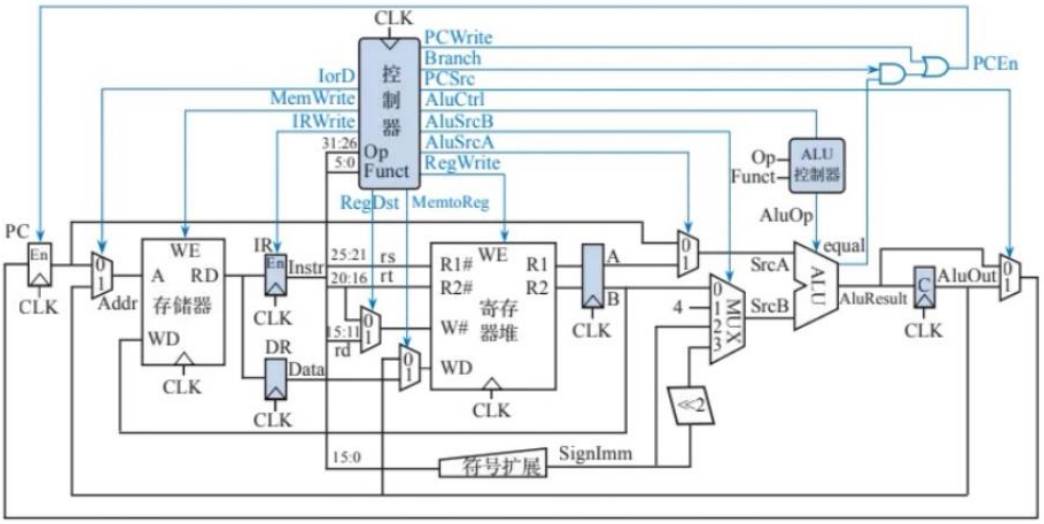


图 6.27　多周期 MIPS 处理器的数据通路高层视图

(4) lui

lui rd,imm

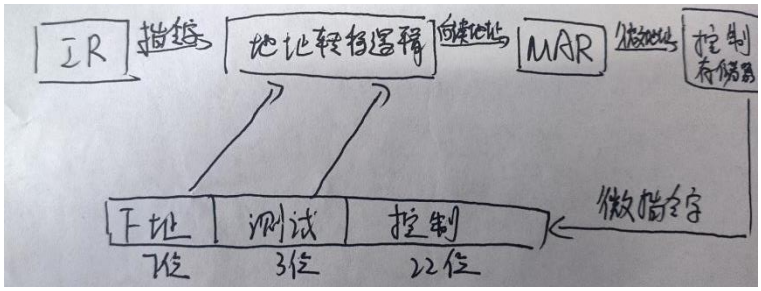
是将 imm 的内容左移 16 位后写入 rd，故需要一个可以将立即数左移 16 位的移位器连入多路选择器，通过 AluCtrl 的值得到移位后的 AluResult，然后将 RegDst 置 1，RegWrite 置 1，将处理后的值写入 rd 寄存器

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?

32 表示微指令共 32 位，128 = 2^7，所以下址字段 7 位，因为测试条件是 3 个，所以测试字段是 3 位，因此控制字段是 32-7-3 = 22 位

(2) 画出对应这种微指令格式的微程序控制器逻辑框图。



6.22 依照图 6.49 所示的微程序控制器原理，结合图 6.66 所示的支持中断的现代时序状态机，重新设计微指令，设计 eret 指令微程序和中断响应周期微程序，利用数字逻辑的方法设计微程序地址转移逻辑。假设指令译码信号分别为 lw、sw、beq、add、addi，给出微程序入口地址 S4~S0,的逻辑表达式，给出地址转移逻辑中多路选择器选择控制信号的逻辑表达式。

机器指令译码信号							微程序入口地址					
LW	SW	BEQ	SLT	ADDI	ERET	ADD	入口地址 10进制	S4	S3	S2	S1	S0
1							4	0	0	1	0	0
	1						9	0	1	0	0	1
		1					14	0	1	1	1	0
			1				19	1	0	0	1	1
				1			22	1	0	1	1	0
					1		25	1	1	0	0	1
						1	28	1	1	1	0	0

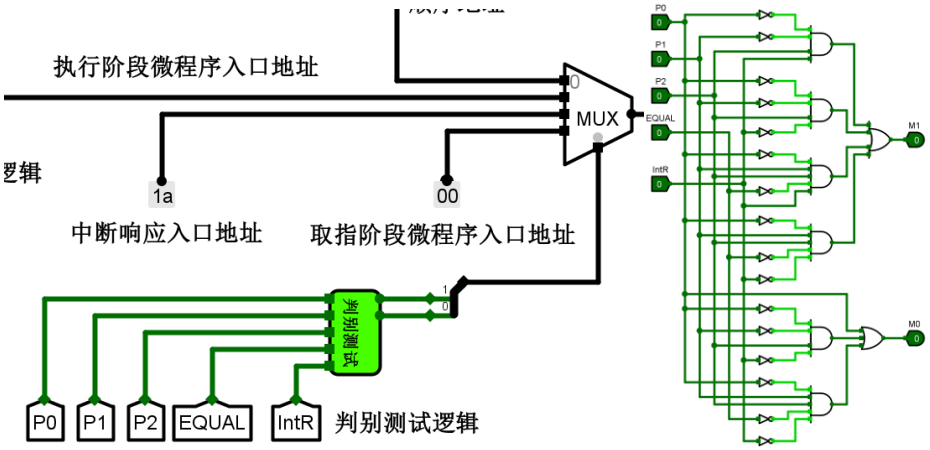
S4 =SLT+ADDI+ERET+ADD

S3= SW+BEQ+ERET+ADD

S2 = LW+BEQ+ADDI+ADD

S1=BEQ+SLT+ADDI

S0=SW+SLT+ERET



M1=~P0&~P1&P2&IntR+~P0&~P1&P2&~IntR+~P0&P1&P2&~EQUAL&IntR+~P0&P1&P2&~EQUAL&~IntR

M0 = P0+~P0&~P1&P2&~IntR+~P0&P1&P2&~EQUAL&~IntR

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 l(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?

R0,R1,R2,R3 以及 PC 是程序员可见的。因为是单总线结构，如果没有暂存器 T，那个到达 ALU 的两个操作数始终相同，无法正常运算。如果有暂存器 T，则可以第一个时钟周期使第一次操作数到达 T 暂存，第二个时钟周期到达 ALU 即可完成正常运算

(2) 控制信号 ALUop 和 SRop 的位数至少各是多少?

Alu 有 7 种操作，所以 ALUop 至少要有 3 位，即 2^3=8>7，移位寄存器有 3 种操作，所以 SRop 至少要有 2 位，即 2^2 = 4>3。

(3) 控制信号 SR_{out} 控制部件的名称或作用是什么?

SR_{out} 控制 SR 是否输出，所以 SR_{out} 控制部件应该是三态门

(4) 端点①~⑨中，哪些端点须连接到控制部件的输出端?

只有控制信号需要连接到控制部件输出端，所以应该是 ①②③⑤⑧

(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->⑨，⑦->④

(6)为什么二路选择器 MUX 的一个输入端是 2？

因为指令共 16 位，16bit = 2B，所以每执行完一条指令需要 PC<--PC+2，所以 MUX 的一个输入端是 2 可以完成 PC+2 的操作。

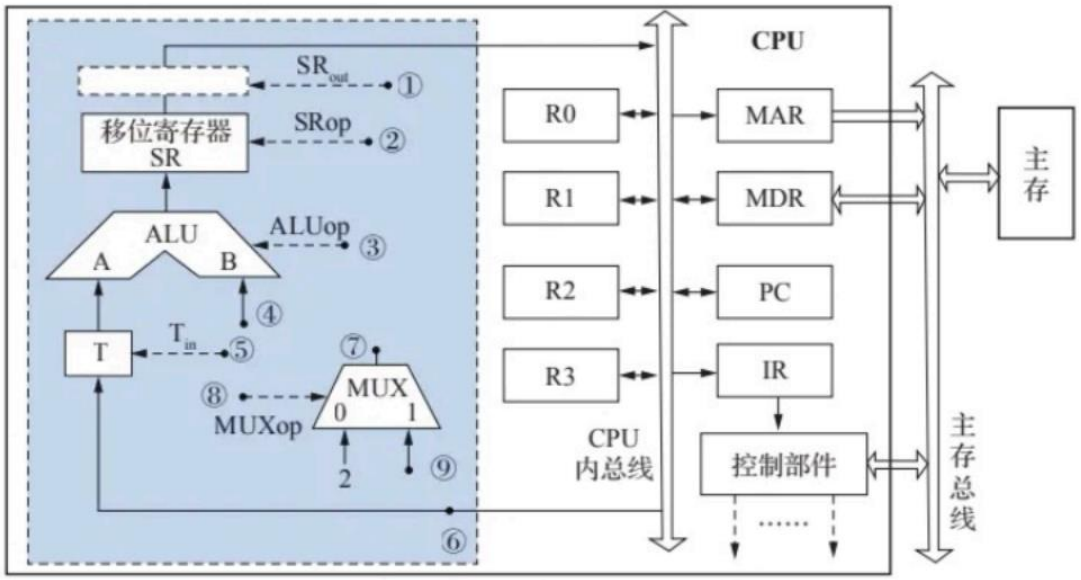


图 6.71　某 16 位计算机的部分数据通路