

6.1 名词解释

机器周期：机器周期是计算机执行一条指令所需的时间单位。它包括取指周期、译码周期、执行周期和存储周期，通常以时钟脉冲为基准

同步控制：同步控制是一种计算机系统中的指令执行方式，所有操作都是根据同一个时钟信号来同步进行的。在同步控制中，各个部件按照预定的时序和时钟信号进行操作，保证了系统的稳定性和可靠性。

异步控制：异步控制是一种计算机系统中的指令执行方式，各个部件的操作不依赖于统一的时钟信号，而是根据信号的到达和状态的改变来进行控制。

微命令：微命令是指微程序中的最小执行单位，它是对硬件操作的微细控制指令，用于控制计算机中的各个功能模块的动作。

微操作：微操作是计算机系统中的最小操作单位，通常是对寄存器、存储器或其他部件进行的基本操作，如读、写、清零等。

时序发生器：时序发生器是计算机系统中的一个模块，用于产生各种时序信号，控制计算机各个部件的操作顺序和时序，确保指令的正确执行。

微程序控制器：微程序控制器是计算机中的一个控制单元，用于执行微程序，控制计算机中各个部件的操作。它根据指令译码的结果产生微命令，并将其送到各个功能模块，实现指令的执行。

控制存储器：控制存储器是计算机中用于存储微程序的一种存储器，它存储了微程序的控制信息，包括各个微命令的执行顺序、条件跳转等信息，用于控制计算机各个部件的操作。

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)

分析：可见寄存器可以通过汇编指令进行访问，汇编语言程序员可以通过转移指令等指令修改程序计数器 (PC) 的值，选 B

(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

分析：题目要求分析在取数阶段和执行阶段，在这两个阶段没有译码操作，所以不需要用到指令译码器（ID），排除 C、D。由于使用了寄存器、寄存器间接寻址方式，因此在取数阶段需要用到通用寄存器组和存储器，选 B

（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

分析：PC 位宽与主存地址总线相同，而 $4\text{GB} = 2^{32}\text{B}$ ，且计算机按字节编址，所以 PC 至少为 32 位。但指令按字边界对齐存放，所以 PC 可以按字编址，只需要 30 位。IR 存放当前正在执行的指令，位宽和指令字长相同，为 32 位，选 B

（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

分析：CPU 从内存中取出并执行一条指令所需的全部时间为指令周期，指令周期由若干机器周期组成，一个机器周期又包含若干时钟周期，D 错误

（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

分析：由于单周期处理器只能在一个时钟周期内完成取指令执行过程，指令执行过程中任何资源都不能被重复使用，只能使用专用数据通路结构，不能用单总线结构，选 A

（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

分析：控制存储器在 CPU 内部，用来存放由微指令组成的微程序，按微指令地址进行访问，并不是按内容访问，选 B

(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

分析：硬布线控制器采用专门的逻辑电路实现，速度取决于电路的延迟，速度比微程序控制器更快，但修改和扩展困难，灵活性差，选 D

(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

分析：字段字节编码法中相容性微指令分在不同字段中，但互斥性微指令应该分在相同字段，每个字段还要留出一个空状态，表示该字段不发出任何微命令。5 个互斥类分别包含 7、3、12、5 和 6 个微指令，操作控制字段分别需要 3、2、4、3、3 位，共 15 位，选 C

(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

分析：32 条机器指令对应微指令为 $32 \times 24 = 128$ 条，公共取指令微程序还包括两条微指令，所以微指令条数共 $128 + 2 = 130$ 条，采用下址字段法，需要 $\log_2(130) = 8$ 位，选 C

(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

分析：空操作指令会使 PC 寄存器+1，指向下一条指令，因此 PC 寄存器会被修改，选 C

6.3 回答问题

(6) 比较单周期 MIPS 处理器与多周期 MIPS 处理器的差异。

答：主要差异是指令周期长度和数据通路的区别。单周期处理器中所有指令在一个时钟周期内完成，比如指令的取出和执行操作，指令执行过程中数据通路的任何资源都不能被重复使用，应该是专用数据通路，对于需要被多次使用的资源都需要设置多个，为避免访问冲突，指令存储器和数据存储器要单独设置。

多周期处理器指令周期包括多个时钟周期，一条指令的执行过程细分为若干个更小的步骤，每个时钟周期执行其中一部分操作，并将操作结果暂存在相关寄存器中供下一个时钟周期进行处理，直至指令执行完毕。多周期数据通路中的功能部件可在一条指令执行过程的不同时钟周期中被多次使用，这种共享复用方式能提高硬件实现效率，所以多周期指令寄存器和数据存储器不需要分开设置

(7) 组合逻辑控制器与微程序控制器各有什么特点？

答：硬布线控制器又称组合逻辑控制器，控制器由各种类型的逻辑门电路和触发器等构成。与微程序控制器相比，组合逻辑控制器具有结构复杂但速度快的特点，但其指令功能修改和扩展较为困难。微程序控制器的设计采用了存储技术和程序设计技术，使复杂的控制逻辑得到简化。计算机通过读出存放在微程序控制器中微指令产生指令执行过程中所需要的控制信号，与硬布线控制器相比，微程序控制器的速度较慢

(8) 说明程序与微程序、指令与微指令的异同。

答：微程序是多条微指令的集合，用于实现指令的功能，属于机器指令级别，对用户透明，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，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、判别测试字段和下地址字段等组成。

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。

(1) 根据 CPU 的功能和结构标明图中 4 个寄存器的名称。

答：a: MDR; c: MAR; b: IR; d: PC

(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。

答：取指阶段：PC→MAR→主存 M→MDR→IR; PC→PC+1

执行阶段：IR(A)→MAR→主存 M→MDR→AC

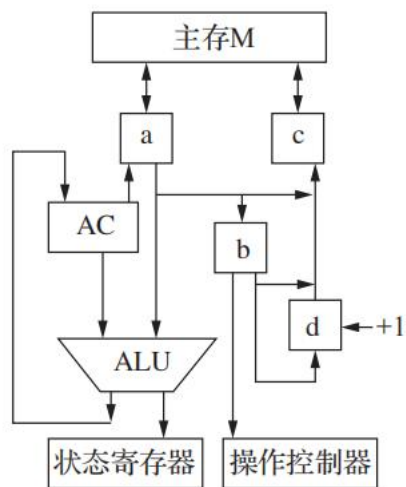


图 6.69 某 CPU 的结构框图

图 6.25 单周期 MIPS 处理器的数据通路高层视图

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。

(4) jal

答：jal 为子程序调用指令，执行 jal 指令时，jal 会保存下一条指令的地址，再无条件跳转到下一个目标地址，执行某一段程序。执行完毕后，根据之前保存的指令地址，又跳回到原先的位置。

jal address	$R[31] \leftarrow PC+4$ $PC \leftarrow \{(PC+4)_{31:28}, address \ll 2\}$	子程序调用指令
-------------	---	---------

首先，jal 指令一定会进行分支跳转，目标地址是将 PC+4 的高 4 位作为高地址部分与指令字中的 26 位立即数 Address 左移两位得到 28 位数据进行拼接，生成一个 32 位的分支目标地址。在图中存在这样的数据通路，所以只需实现将 PC 中的地址存放到寄存器堆中。

增加一条从 PC 指向寄存器堆的线路，并控制 PC 计数器的写使能端，使其能够向寄存器中写入地址。由于 PC 还要从寄存器堆中读取之前存的地址，所以还需要一条从寄存器堆到 PC 的线路，并控制 PC 能够读取寄存器堆中的数据。

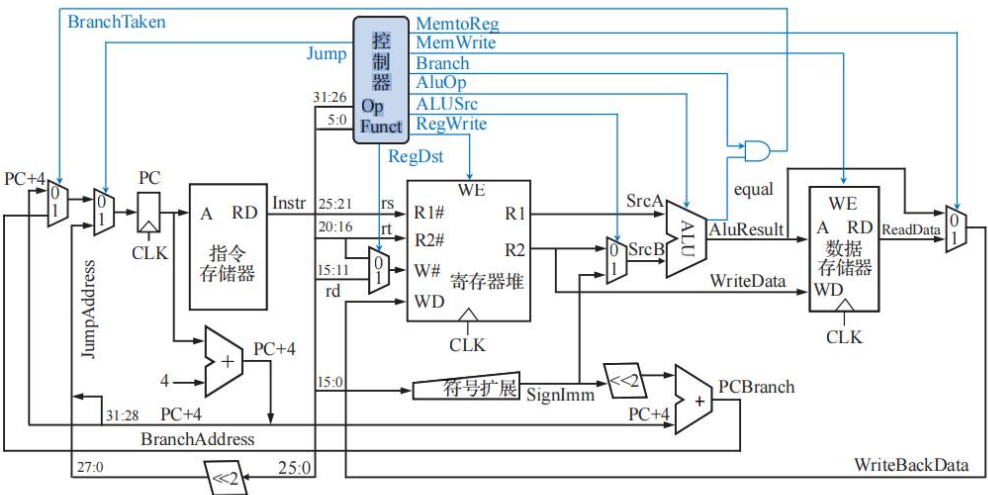


图 6.25 单周期 MIPS 处理器的数据通路高层视图

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。

(2) lui

答：lui 指令作用是将一个 16 位立即数左移 16 位后加载到一个寄存器中。为实现这条指令，需要先在译码阶段识别出 lui 指令；在执行阶段，从指令中提取出 16 位的立即数，进行左移 16 位操作，然后将其存放到目标寄存器中。在一下电路，从指令寄存器 IR 中取出 lui 指令并进行译码，将指令中的立即数左移 16 位后传送到寄存器堆中的目标寄存器或其他寄存器中。

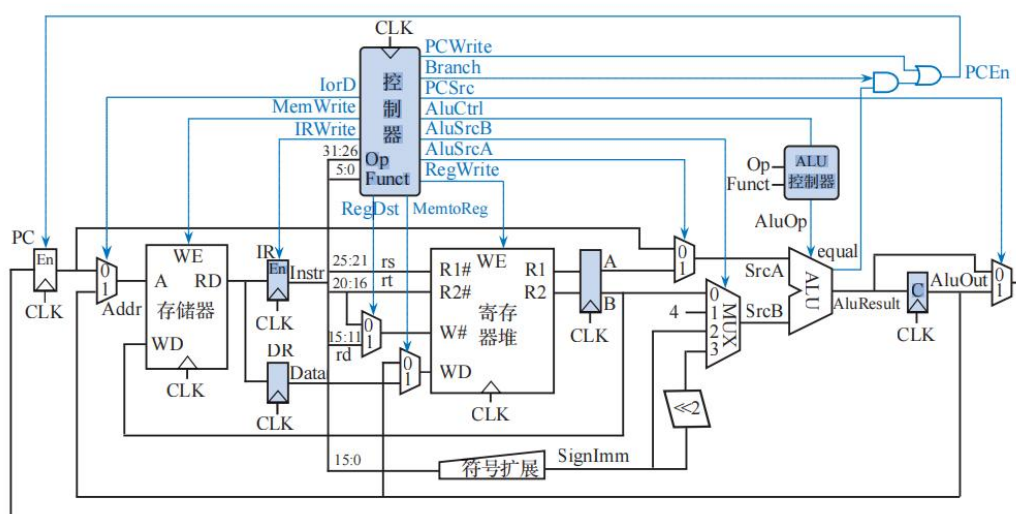


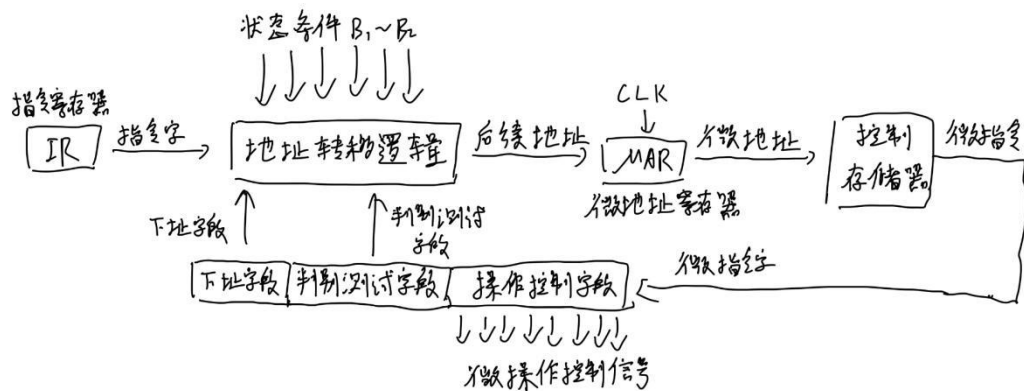
图 6.27 多周期 MIPS 处理器的数据通路高层视图

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？

答：下址字段为 7 位，判别测试条件 3 位，操作控制字段 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。

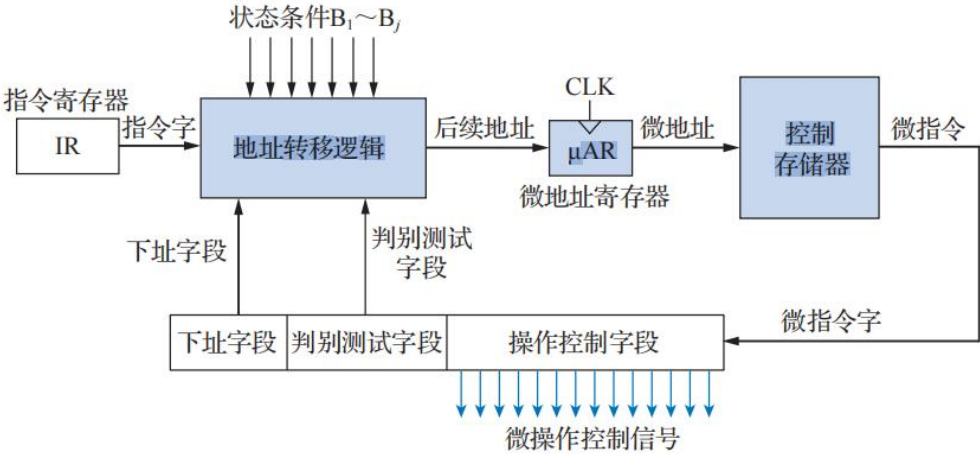


图 6.49 微程序控制器组成框图

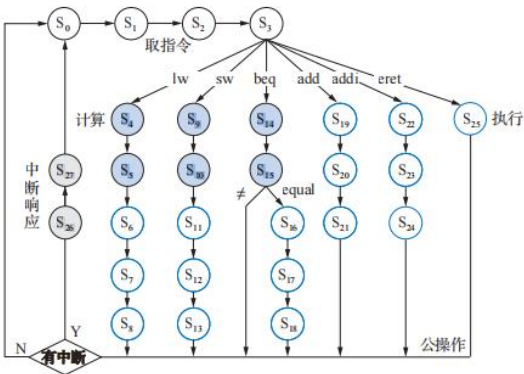


图 6.66 支持中断的现代时序状态机

答：eret 微程序：

S0: 检查中断返回地址是否有效。

S1: 将中断返回地址加载到程序计数器 (PC)。

S2: 结束，返回到主程序流程。

中断响应周期：

S0: 检测到中断信号。

S1: 保存当前 PC 到中断向量。

S2: 跳转到中断处理程序入口。

微程序地址转移逻辑

对于 `eret` 和中断响应周期，我们需要设计逻辑来确定微指令的跳转条件。假设指令译码信号分别为 `lw`、`sw`、`beq`、`add`、`addi`，微程序入口地址可由状态机的当前状态决定。如果当前状态为 `eret` 的 `S0` 且条件满足 (比如检测到中断返回标志位)，则跳转到 `S1`。如果在 `S1` 结束处理，则跳转到 `S2` 或返回到主流程。

微程序入口地址 S4 ~ S0 的逻辑表达式如下

$S4 = 0; S3 = 0; S2 = \text{add} \vee \text{addi}; S1 = \text{sw} \vee \text{beq} \vee \text{addi}; S0 = \text{lw} \vee \text{sw} \vee \text{beq}$

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 为暂寄存器；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？为何要设置暂寄存器 T？

答：通用寄存器（R0~R3）和 PC 程序员可见。因为采用了单总线结构，若无暂寄存器 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。增加暂寄存器后可以使 A、B 端口输入不同的数据

（2）控制信号 ALUop 和 SROP 的位数至少各是多少？

答：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）控制信号 SRout 控制部件的名称或作用是什么？

答：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) 端点①~⑨中，哪些端点须连接到控制部件的输出端？

答：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) 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。

答：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) 为什么二路选择器 MUX 的一个输入端是 2？

答：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

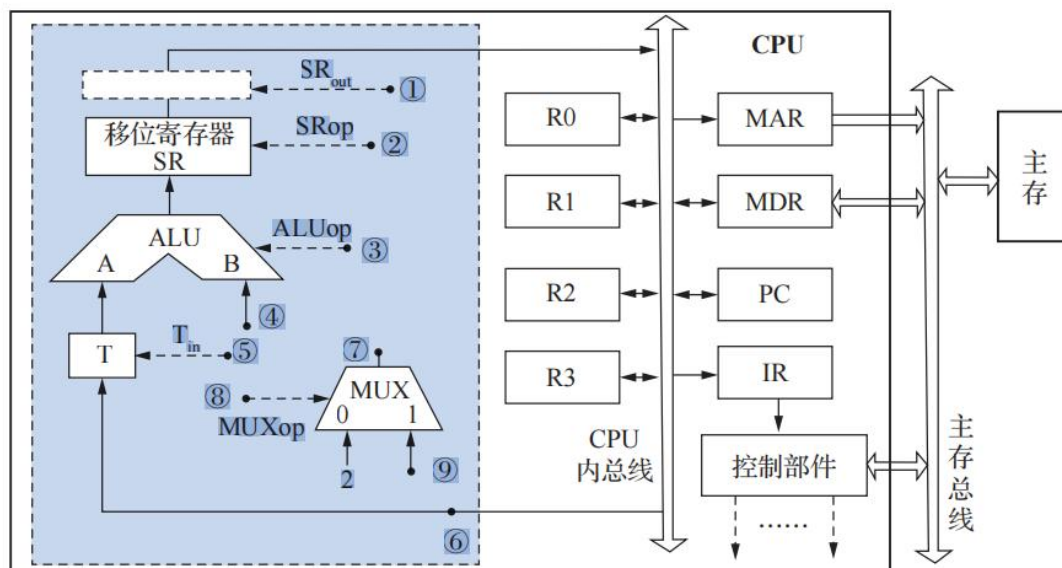


图 6.71 某 16 位计算机的部分数据通路