

计算机组成原理课程作业

——第六次

黄勖 22920212204392



(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)

CPU中的寄存器，用户可见的是通用寄存器组、程序状态字寄存器、程序计数器；用户不可见的是存储器地址寄存器、存储器数据寄存器、指令寄存器、暂存寄存器。

(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) V. 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

取数过程：寄存器寻址用到通用寄存器组，寄存器间接寻址用到存储器。

执行过程：用到算术逻辑单元。

(指令译码器在分析指令过程中用到，在取数及执行过程中未用到。)

6.2

(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

PC每次定位一条指令，并且指令按字边界对齐存放，因此PC无需定位到每个字节，只需定位字即可。字长为32位，即4B。 $4\text{GB}/4\text{B}=2^{30}$ ，即主存中一共可存储 2^{30} 个字，所以PC至少需要30位才能将每个字都定位到。

采用32位定长指令字格式，所以IR的位数至少是32位。

6.2

(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

指令流水线中，理想情况下每个机器周期完成一条指令并开始执行一条新的指令，而非一个时钟脉冲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

单周期处理器一个时钟内处理完取指令、寻址等操作，有多种类型数据要传输，需要专用数据通路，而单总线结构数据通路一个时钟只能传输一类数据，所以单周期处理器不能采用单总线。

6.2

6.2

(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 (TLB快表按内容访问)

(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。

6.2

(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

第1个微命令互斥类包含7个微命令，再加上一个NOP微命令(离散状态没命令)，共8个微命令，需要3位二进制编码；

第2个微命令互斥类包含3个微命令，再加上一个NOP微命令；共4个微命令，需要2位二进制编码；

第3个微命令互斥类包含12个微命令，再加上一个NOP微命令，共13个微命令，需要4位二进制编码；

第4个微命令互斥类包含5个微命令，再加上一个NOP微命令，共6个微命令,需要3位二进制编码；

第5个微命令互斥类包含6个微命令，再加上一个NOP微命令，共7个微命令，需要3位二进制编码。

共需要 $3+2+4+3+3=15$ 位二进制编码。

(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条指令（微程序）对应的微指令： $32 \times 4 = 128$ 条 公共微指令2条 $128 + 2 = 130$

$2^7 < 130 < 2^8$ 8位

(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

- A.因为没有cache，所以CPU要直接访问内存才能完成取指，否则无法执行完一个指令周期。所以每个指令周期中CPU至少访问主存一次
- B.指令周期是指CPU取出并执行一条指令的时间，至少与CPU时钟周期相等。即指令周期大于等于一个CPU时钟周期。
- C.即使是空操作，在完成操作后，程序寄存器PC也会自动加1
- D.开中断状态下可以响应外部中断。

6.2

插入一段文本插入

(1)CPU的基本功能是什么?从实现其功能的角度分析,它应由哪些部件组成?

答: (1) 基本功能:

- ①指令控制: 完成取指令、分析指令和执行指令的操作,即产生指令执行过程中需要的操作控制信号。
- ②操作控制: 一条指令的功能往往是由若干操作信号的组合来实现的。CPU管理并产生由内存取出的每条指令的操作信号,把各种操作信号送往相应的部件,从而控制这些部件按指令的要求进行动作。
- ③时间控制: 对各种操作加以时间上的控制。时间控制要为每条指令按时间顺序提供应有的控制信号
- ④数据加工: 对数据进行算数和逻辑运算。
- ⑤中断处理: 对计算机运行过程中出现的异常情况和特殊请求进行处理。

(2) 组成: CPU主要由控制器和运算器两部分构成。控制器的主要功能包括: 取指令、计算下一条指令的地址、对指令译码、产生相应的操作控制信号、控制指令执行的步骤和数据流动的方向。运算器是执行部件,由算术逻辑单元和各种寄存器组成。

(2)CPU内部有哪些寄存器?功能分别是什么?哪些是程序员可见的?哪些是必需的?

答：CPU内部的寄存器有：

1. 程序计数器（PC）保存将要执行的指令的字节地址；
2. 存储器地址寄存器（AR）用于保存CPU访问主存的单元地址；
3. 存储器数据寄存器（DR）用于存放从主存中读取的数据或准备写入主存的数据；
4. 指令寄存器（IR）用于保存当前正在执行的指令；
5. 通用寄存器（GR）是指运算器内部的若干寄存器，通常是指这些寄存器具有多种功能，可作为ALU的累加器、变址寄存器等；
6. 程序状态字寄存器（PSW），用于保存由算术运算指令、逻辑运算指令，测试指令等建立的各种条件标志。

用户可见的寄存器：通用寄存器、程序状态寄存器、程序计数器PC

CPU内部不必需的寄存器有AR、DR、IR，剩余的为必需寄存器。

(3)什么是取指周期?取指周期内应完成哪些操作?

答：取指周期就是从开始取指令到取指令完成所需要的时间。

取值周期应完成的操作有：

- 1.通过PC的内容找到主存中对应地址的指令，将找到的指令存入我们的数据寄存器中，然后再传入指令寄存器中，以实现取值；
- 2.在向主存中取指令的过程中，也应该对PC值的内容进行处理，以计算出下一次取址的地址。

(4)计算机为什么要设置时序系统?说明指令周期、机器周期和时钟周期的含义。

答：指令执行过程中的所有操作必须按照一定的次序完成，而且这些操作持续的时间也有严格的限制，因此，在计算机系统中需要设置时序系统，对指令执行过程中的所有控制信号进行时间控制，以保证指令功能的正确实现。

指令周期通常是将一条指令从取出到执行完成所需要的时间，包括取指周期和执行周期，指令周期通过若干和机器周期组成，所包含的机器周期的数量随指令功能和寻址方式的不同而不同。

机器周期也称为CPU周期，是为了便于管理，常把一条指令的执行过程划分为若干个阶段（如取指、译码、执行等），每一个阶段完成一个基本操作，而完成一个基本操作所需要的时间称为机器周期。

时钟周期又称为振荡周期，常定义为时钟脉冲频率的倒数，是时序中最小的时间单位。

三者的关系为：指令周期由若干个机器周期组成，而机器周期又包含若干个时钟周期。

(5)简述传统三级时序和现代时序的差异。

答：（1）传统三级时序：早期的计算机采用状态周期、节拍电位和节拍脉冲三级时序体制来对操作控制信号进行定时控制。其中状态周期用电位来表示当前处于指令执行的哪个机器周期，节拍电位用电位表示当前处于机器周期的第几个节拍。

（2）现代时序：现代计算机中，已经不再使用多级时序体制，指令执行过程中的定时信号就是基本时钟信号，一个时钟周期就是一个节拍，不再设置节拍脉冲，称为现代时序系统。

(6)比较单周期MIPS处理器与多周期MIPS处理器的差异。

答：二者的差异主要是指指令周期长度、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，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，都应该是专用数据通路，而需要被多次使用的资源都需要设置多个，为避免访存冲突，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，所以多周期指令存储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：组合逻辑控制器又称硬布线控制器，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：（1）程序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。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，对用户的透明性不强，存放在CPU内的控制存储器中；

（2）指令是指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。

(9)微命令有哪几种编码方法？它们是如何实现的？

答：微指令的微命令有三种编码方法，分别是直接表示方法、字段直接译码法和混合控制法。

①直接表示法的基本思想是：将微指令操作控制字段的每个二进制位定义为一个微命令，用“1”或“0”表示相应的微命令的“有”或“无”。

②字段直接译码法的基本思想是：将微指令格式中的操作控制字段分成若干组，每组中包含若干个互斥性微命令，将相容性的微命令安排在不同组。

③混合控制法：将直接表示法与字段直接译码法混合使用，以便在微指令字长、并行性及执行速度和灵活性等方面进行折衷，发挥它们的共同优点。

(10)简述微程序控制器和硬布线控制器的设计方法。

答：（1）微程序控制器设计方法：

- ①分析指令执行的数据通路，列出每条指令在所有寻址方式下的执行操作流程和每一步所需要的控制信号；
- ②对指令的操作流程进行细化，将每条指令的每个微操作分配到具体的机器周期的各个时间节拍信号上；
- ③设计微指令格式、微命令编码方法和程序组织方式；
- ④编制每条指令的微程序；并按照所设计的微程序组织方式存放到控存中；
- ⑤对微命令进行同步控制，并送数据通路的相关控制点。

（2）硬布线控制器设计方法：

- ①分析指令执行的数据通路，列出每条指令在所有寻址方式下的执行操作流程和每一步所需要的控制信号；
- ②对指令的操作流程进行细化，将每条指令的每个微操作分配到具体的机器周期的各个时间节拍信号上，即对操作控制信号进行同步控制。
- ③对每一个控制信号进行逻辑综合，得到每个控制信号的逻辑表达式。
- ④最后采用逻辑门或PLA或ROM实现逻辑表达式的功能，各控制信号送数据通路的相关控制点。

(11)简述CPU中内部异常与外部中断的区别。

答：（1）异常（Exception）通常是指CPU内部引起的异常事件，也称为内部中断或软件中断，可分为故障（Fault）、自陷（Trap）、终止（Abort）3种。异常与指令或硬件有关，产生异常的指令可能需要重新执行，所以异常的断点是当前指令，而不是下一条指令。

（2）外部中断（Interrupt）是指由外部设备向CPU发出的中断请求（如鼠标点击、按键动作等）要求CPU暂停当前正在执行的程序，转去执行为某个外部设备事件服务的中断服务程序，处理完毕后再返回断点继续执行。注意外部设备中断的时机是一条指令结束后，指令结束时需要查询是否有外部中断请求。外部中断来自CPU外部，与具体指令无关，是随机事件。

（3）区别：广义的中断包括中断和异常，统一称为中断。狭义的中断和异常的区别在于是否与正在执行的指令有关，中断可以被屏蔽，异常不能被屏蔽，一旦出现应立即处理。

(12)简述异常与中断处理的一般流程。

答：异常与中断的处理方式基本一致，不同的计算机体系结构和不同教材对异常和中断的定义也不尽相同，为方便描述，后文统一称为中断（Interrupt）。当发生中断事件时，CPU 接收到中断请求，在指令执行结束时 CPU 要进入中断响应周期进行响应处理。当然也有例外，例如产生故障异常的指令并没有执行完毕，但必须立即进行中断响应。中断响应周期内的主要任务是关中断、保存断点和中断识别。

(1) 关中断的目的是临时禁止中断请求，是为了在中断响应周期以及中断服务程序中保护现场操作的完整性，只有这样才能保证中断服务程序执行完成后能返回断点正确执行。关中断类似操作系统的 PV 操作中的 P 操作；而开中断类似 V 操作。在单级中断中只有中断返回时才需要开中断；而在多级嵌套中断中完成现场保护后就可以通过指令开中断，方便中断嵌套。

(2) 保存断点就是保存将来返回被中断程序的位置，对于已经执行完毕的指令，其断点是下一条指令的位置（注意有可能不是顺序指令地址）；对于缺页故障、段错等执行指令引起的故障异常，由于指令并没有执行，因此断点应该是异常指令的 PC 值。为了支持多级嵌套中断，通常会将断点放置在内存堆栈中进行保护，如 x86 计算机系统；而 MIPS 中是先将断点存放到异常指令地址寄存器 EPC 中，再在中断服务程序中将其作为现场进行压栈保护。

(3) 中断识别的主要任务就是根据当前的中断请求识别出中断来源，也就是识别出发生了什么中断，并将对应中断的中断服务程序入口地址送入程序计数器 PC，中断识别的方法将在最后一章具体介绍。中断响应周期内的操作都是由硬件实现的，整个响应周期是不可被打断的。中断响应周期结束后 CPU 就开始从当前 PC 中取出中断服务程序的第一条指令开始执行，直至中断返回；这部分任务是由 CPU 通过执行中断服务程序完成的，是由软件实现的，整个中断处理过程是软、硬件协同实现的。

(13)要支持异常与中断处理，CPU需要对硬、软件进行哪些扩展？

答：（1）数据通路升级，需在数据通路上添加相关的中断功能和数据通路；

（2）操作控制器升级

1.三级时序硬布线控制器升级，需增加中断相应周期，用于处理中断异常情况；

2.现代时序硬布线控制器升级，增加eret指令的支持，该指令需要一个时钟周期；

3.微程序控制器升级，增加一条微指令实现eret指令的功能；对于新增的中断相应周期的两个状态可以增加两条实现中断相应操作的微指令。并同时增加Pend用来标记当前微指令为最后一条微指令。

（3）中断服务程序拓展

中断服务程序主要包括4个步骤：保护现场、终端服务、恢复现场、中断返回。软件方面需要增加开关中断相关、中断返回指令，另外需要设置保存现场的堆栈，对MIPS结构需要增加EPC寄存器，程序中需要设置好堆栈指针sp。

6.4

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个寄存器的名标。

答: d是PC会进行自加,因为在顺序取指时会有该步骤; 由于b会对d进行影晌,则b应是IR,指令中会判断程序执行的顺序,从而影晌PC的操作; d中数据会传入c中,则c是AR,用来保存即将访问的指令地址; a是DR,用来存储指令和需要用到的操作数据;

(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→AR→主存M→DR→IR;
PC→PC+1

执行阶段: IR(A)→AR→主存M→DR→AC

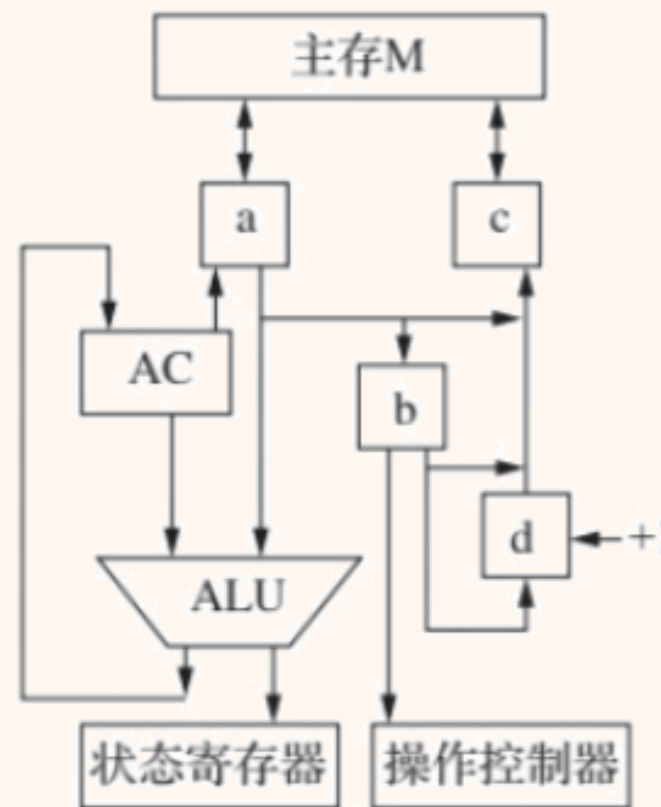


图 6.69 某 CPU 的结构框图

6.5

6.5修改图6.8所示的单总线结构处理器,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。

(1) sll ;

答: 需要新增IR[SHAMT]out信号来输出指令中的SHAMT用于移位运算、增加ALU的移位控制信号SLL, sll rd, rt, shamt; 将rt寄存器逻辑左移shamt后, 存入rd寄存器中;

取指周期:

PC->AR, PC->X

PCout=ARin=Xin=1

X+4->Z

+4=1

Z->PC, M[AR]->DR

Zout=PCin=1; Read=DREin=1

执行周期:

R(rt)->X

Rout=Xin=1

X<<IR(shamt)->Z

Zin=IR(shamt)out=SLL=1

Z->R(rd)

Zout=Rin=1

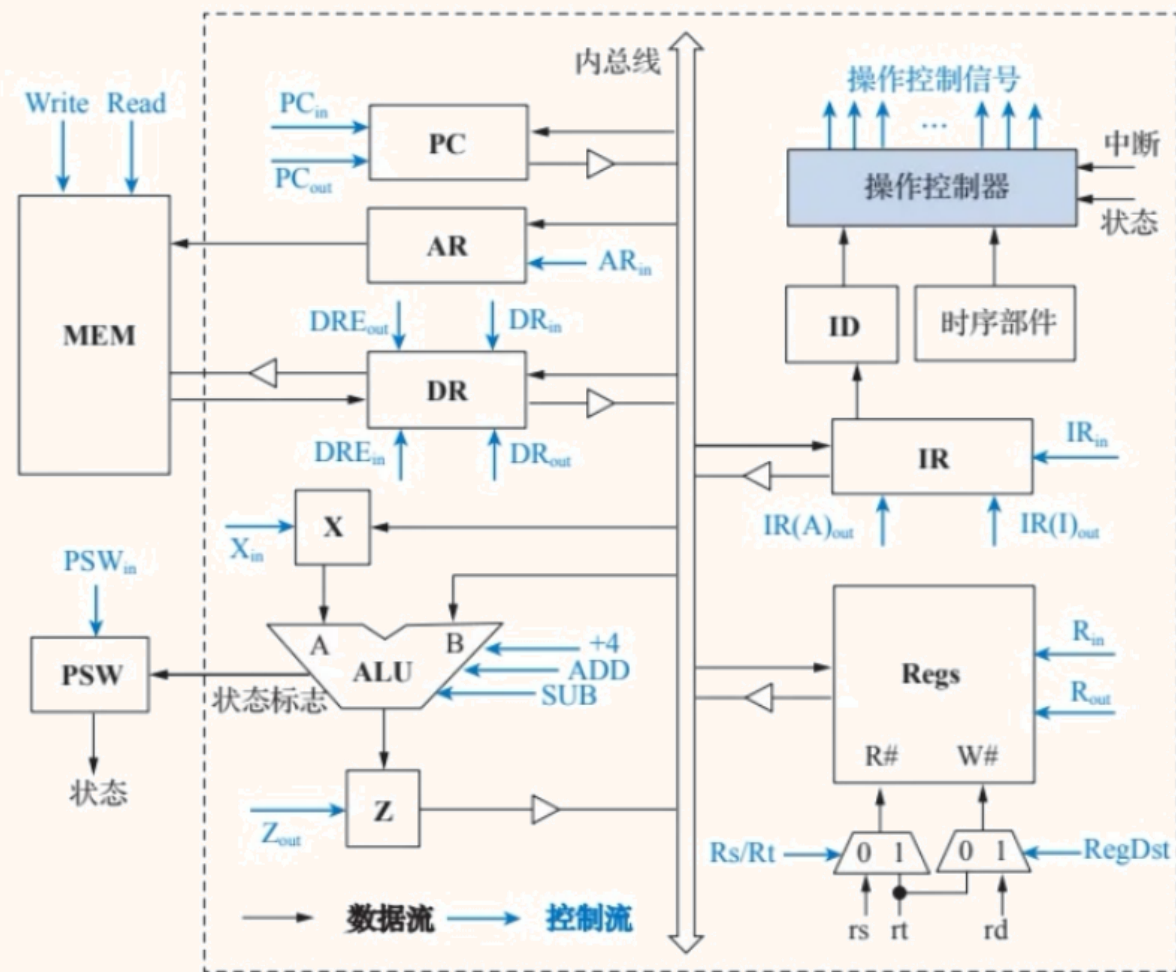


图 6.8 单总线结构的计算机框图

6.7

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。

(2) lui ;

答: 需要增加一条立即数左移16位的数据通路连接到数据存储器后的多路选择器。

lui rt,imm; 将16位立即数送入符号扩展,将指令中的16bit立即数保存到地址为rt的通用寄存器的高16位。
指令中的目的寄存器字段送入寄存器堆的写寄存器编号端 W#(RegDst =0), 地址为rt的通用寄存器的低16位使用0填充, 令ALUSRC = 1, 得到 ALURESULT, 再令MEMTOREG = 0 将运算结果写回运算器, 将选择将此数据输出到寄存器堆写数据端口 WD (设置RegWrite=1), 将数据写到rt编号的寄存器中。

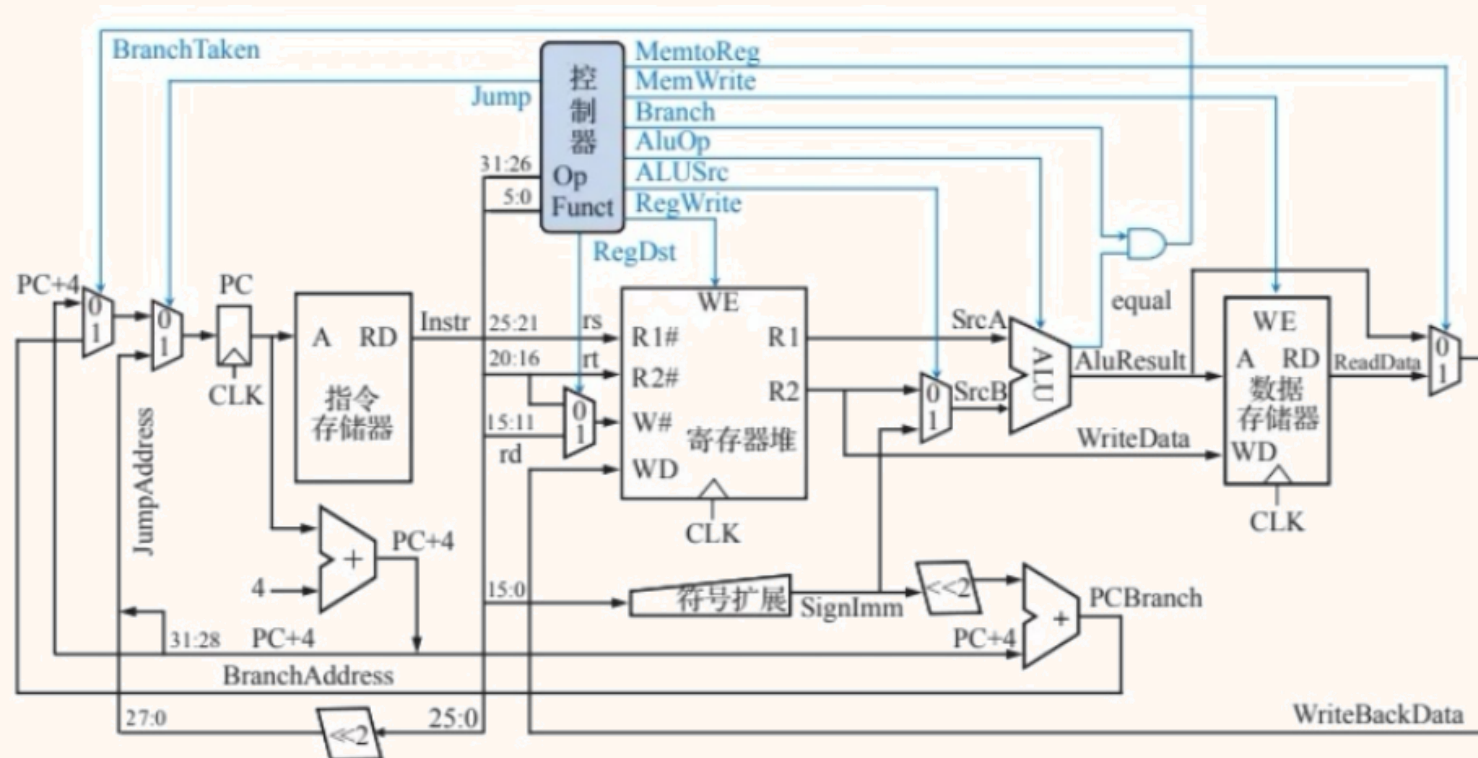


图 6.25 单周期 MIPS 处理器的数据通路高层视图

6.9

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。

(3) bgtz ;

答: bgtz rs, offset, 如果地址为rs的通用寄存器的值大于零, 那么发生转移

T1:

- ① $M[PC] \rightarrow IR$ $lorD=0, MemWrite=0, IRWrite=1$
- ② $PC+4 \rightarrow PC$ $AluCtrl=0, AluSrcA=0, AluSrcB=1, PCSrc=0, PCWrite=1$

T2:

- ① 指令译码
- ② $R[rs] \rightarrow A$
- ③ $PC+Imm \ll 2 \rightarrow C$ $AluCtrl=0, AluSrcA=0, AluSrcB=3$

T3:

if ($R[rs]>0$) $C \rightarrow PC$ 将分支目标地址送入PC输入端, 比较寄存器产生的大于0标志信号送入简单组合逻辑生成程序计数器 PC 的写使能信号, 决定是否写入分支目标地址

$AluSrcA=PCSrc=Branch=1$

需要增加的数据通路和控制信号:

Alu需要增加一个判断数据是否大于0的greater信号, 即AluOp需要增加一个新的运算功能, 增加新的数据通路: greater信号与Branch信号进行逻辑与再与 PCwrite信号进行逻辑或后生成PC写使能控制信号PCEn

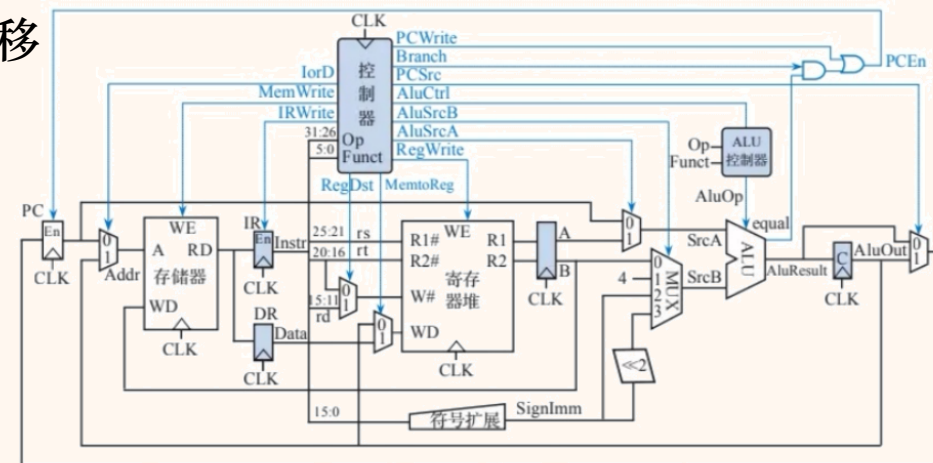


图 6.27 多周期 MIPS 处理器的数据通路高层视图

6.11

6.11 基于加快经常性事件的原理,显然优化 R 型指令数据通路可以提高程序执行效率。尝试优化图6.25所示的多周期 MIPS 处理器的 R 型算术逻辑运算指令的数据通路,以缩短 R 型算术逻辑运算指令执行周期,给出优化理由以及优化后的最小时钟周期,并结合上题中的参数计算测试程序执行的时间。

答: R型运算指令中计算的结果需要先缓存在C寄存器中,再送寄存器堆写回;直接将ALU运算结果送寄存器写回数据端,这样有望减少1个时钟周期。

修改后lw、sw、beq、R型运算、I型运算指令的CPI分别为5、4、3、3、4,因

$CPI = 5 \times 0.1 + 4 \times 0.1 + 3 \times 0.1 + 3 \times 0.5 + 4 \times 0.2 = 3.5$, $T_{cpu} = 700s$

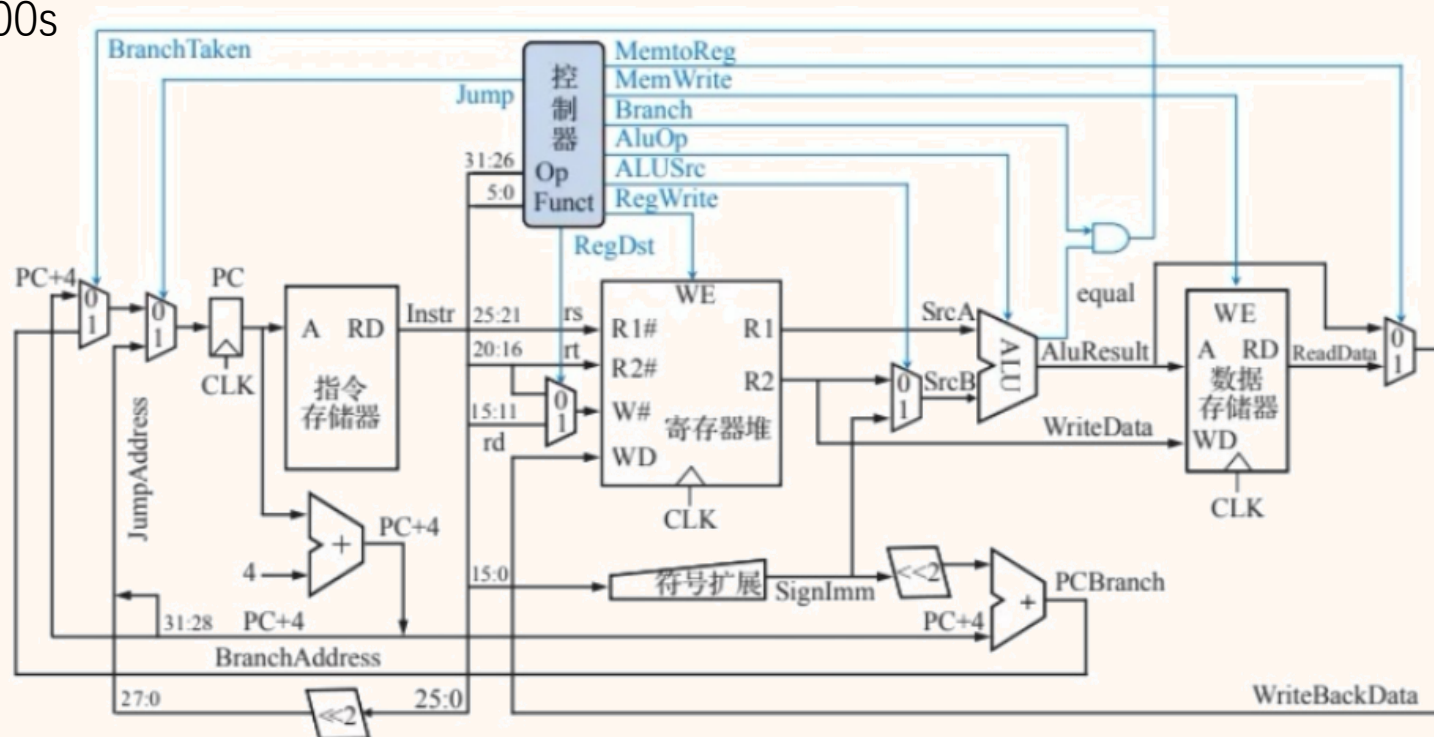


图 6.25 单周期 MIPS 处理器的数据通路高层视图

6.12

6.12对于例6.5中的多周期处理器,其各功能部件使用表6.21的时间延迟,如果可以优化其中一个功能部件的关键延以提升处理器整体性能,应该选择哪个部件进行优化? 如果这种优化与成本是线性关系,如何优化才能使得处理器性能达到最优,且成本最低?

答: 应当优化存储器性能, 在该表中是判断 $\max(T_{alu}+T_{mux}, T_{mem})$, 则当 T_{mem} 等于 $T_{alu}+T_{mux}=110\text{ps}$ 后, 存储器的性能达到最优, 此时成本最低, 性能最优。

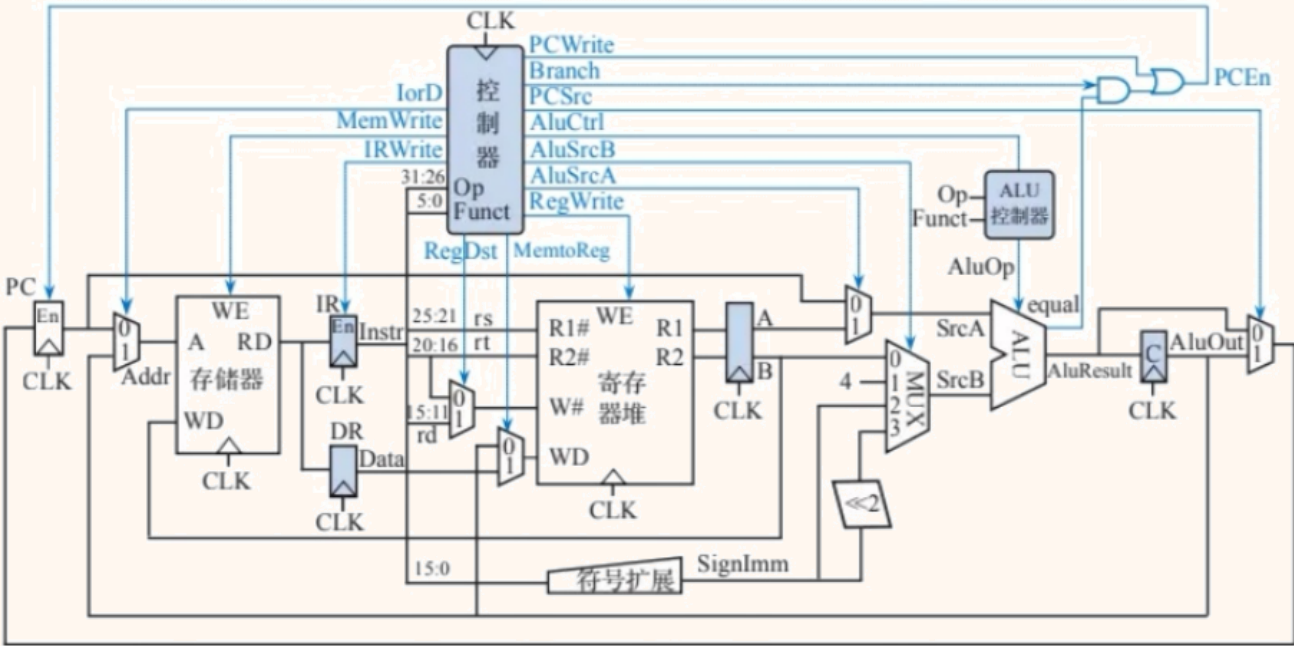


图 6.27 多周期 MIPS 处理器的数据通路高层视图

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

6.20

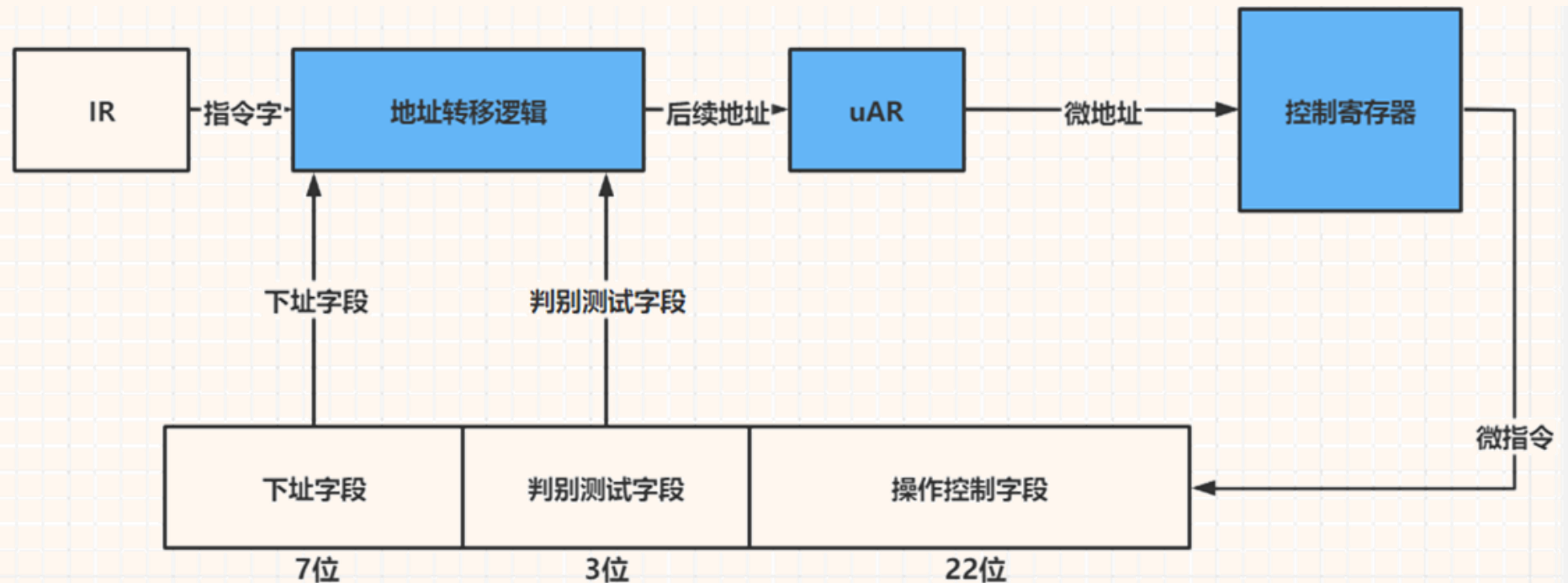
6.20 已知某计算机采用微程序控制方式,控制存储器容量为 128×32 位。微程序可在整个控制存储器中实现分支跳转,控制微程序判别测试条件共3个,微指令采用水平型格式,后续微指令地址采用下址字段法。回答下列问题。

(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) 由于存储器容量为 128×32 位,则微指令总字长为32位,其中其总数目为128,则下址字段为7位,由于判别测试条件为3位,则P字段为3位,剩余的全为操作控制字段共有 $32 - 3 - 7 = 22$ 位。

(2)



6.21

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。

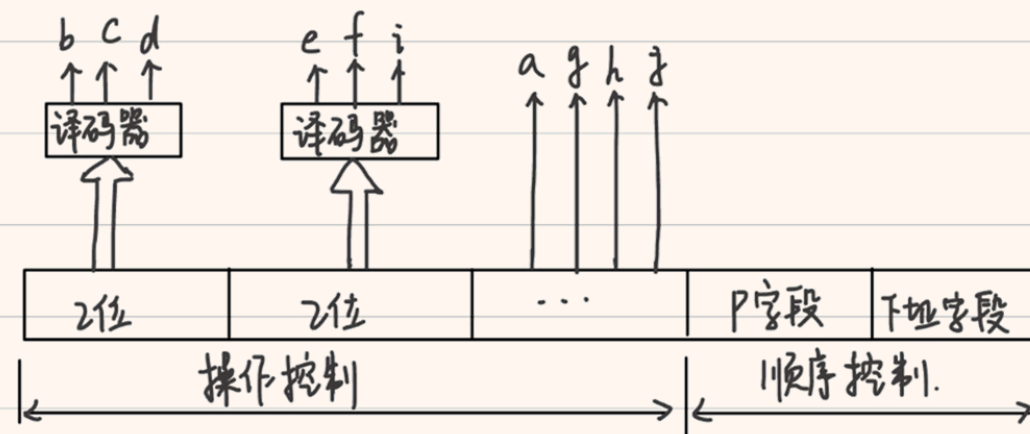
表 6.22 微指令及其对应的微操作控制信号

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		

可以发现两个互斥组 (b, c, d) , (e, f, i) 采用字段译码法直接进行编码, 其余的 a,g,h,j 等四个微命令采用直接表示法

由题可得下表:

微指令\控制信号	a	b	c	d	e	f	g	h	i	j
μI_1	T		T		T		T			
μI_2	T			T		T		T		T
μI_3	T			T	T					
μI_4	T	T							T	
μI_5	T			T		T				T



6.23

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 MDRinE 为1表示允许数据从 DB 送入 MDR 中, MDRin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 $(R0)+((R1))\rightarrow(R1)$,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。

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。

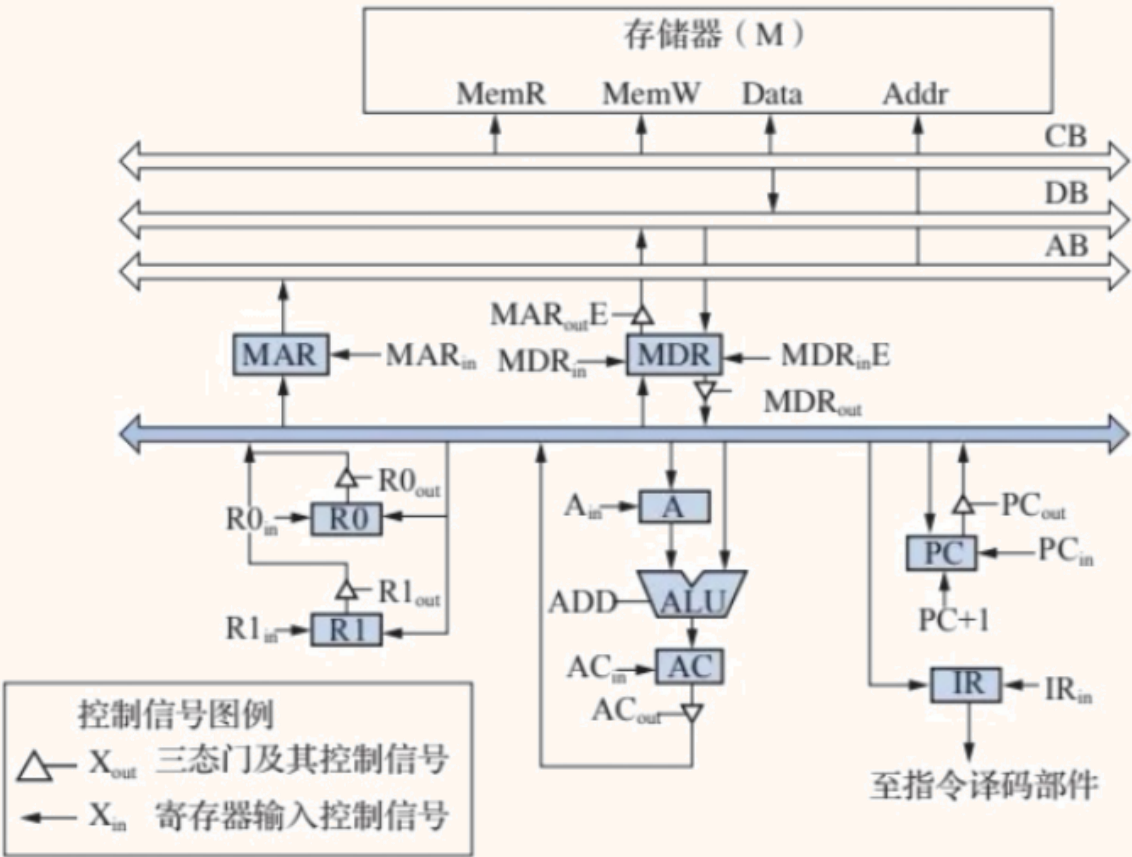


图 6.70 某计算机数据通路图

时钟	功能	有效控制信号
C1	MAR←(PC)	PCout,MARin
C2	MDR←M(MAR) PC←(PC)+1	MemR,MDRinE,PC+1
C3	IR←(MDR)	MDRout,IRin
C4	指令译码	无
C5	MAR ← (R1)	R1out,MARin
C6	MDR← M(MAR) A← R0	MEMR,MDRinE,R0out,Ain
C7	AC ← (MDR) + (A)	MDRout,Add,ACin
C8	MDR← (AC)	ACout,MDRin
C9	M(MAR) ← (MDR)	MDRoutE,MemW

ABOUT

感谢观看

THANKS FOR WATCHING

---汇报人：黄勖