

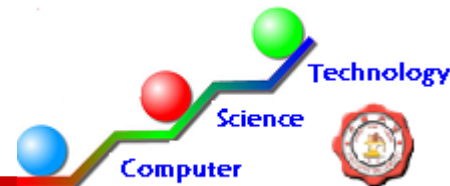
# 计算机组成原理第七章题解

# 第七章 7.1



- ❑ 7.1 请分析CPU内部采用分散互连结构和单总线以及多总线结构的优缺点。
- ❑ 答：
- ❑ 分散互连结构是在需要进行数据传输的部件间设置专用通路。该结构的优点是部件间不存在传输通路的竞争问题，所以指令执行速度快。
- ❑ 单总线结构是将各部件都连接在单一的总线上。其优势是CPU结构紧凑，但由于部件间争用总线造成指令执行速度慢。
- ❑ 双总线结构是在单总线结构的基础上增加了一条总线，使得ALU的两个输入可以分别来自两条线总线。双总线结构相对于单总线结构来说，通过增加一条总线来分担数据传输流量，使得指令执行速度得以提高。
- ❑ 三总线结构是在双总线结构的基础上再增加一条总线，使得ALU的两个输入可以分别来自两条线总线，且ALU的输出连接到第三条总线上。总之，多总线结构通过增加硬件开销，换取指令执行速度。

# 第七章 7.2



7.2、设数据总线上接有A,B,C,D四个寄存器，要求选用合适的74系列芯片，完成下列逻辑设计：

(1) 设计一个电路，在**同一时间**实现 $D \rightarrow A$ ,  $D \rightarrow B$ 和 $D \rightarrow C$ 寄存器间的传送；

(2) 设计一个电路，实现下列操作：

**T0时间**完成 $D \rightarrow$ 总线；

**T1时间**完成总线 $\rightarrow A$ ；

**T2时间**完成 $A \rightarrow$ 总线；

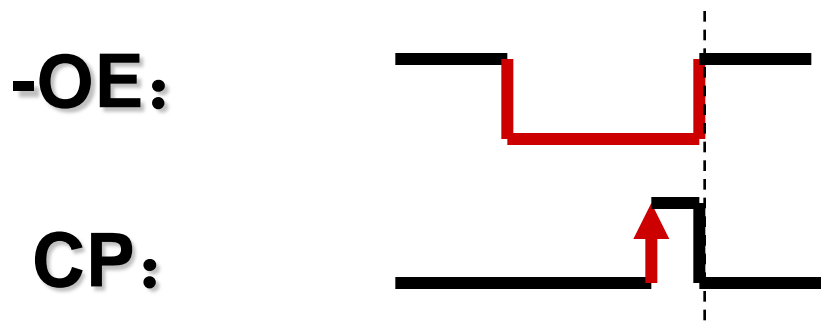
**T3时间**完成总线 $\rightarrow B$ 。

## 第七章 7.2



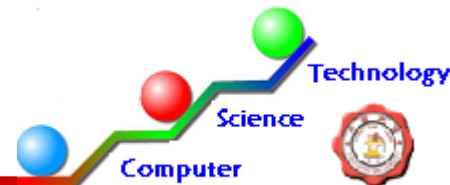
解：

(1) 采用三态输出的D型寄存器74LS374做A、B、C、D四个寄存器，其输出可直接挂总线。A、B、C三个寄存器的输入采用同一脉冲打入。注意-OE为电平控制，与打入脉冲间的时间配合关系为：



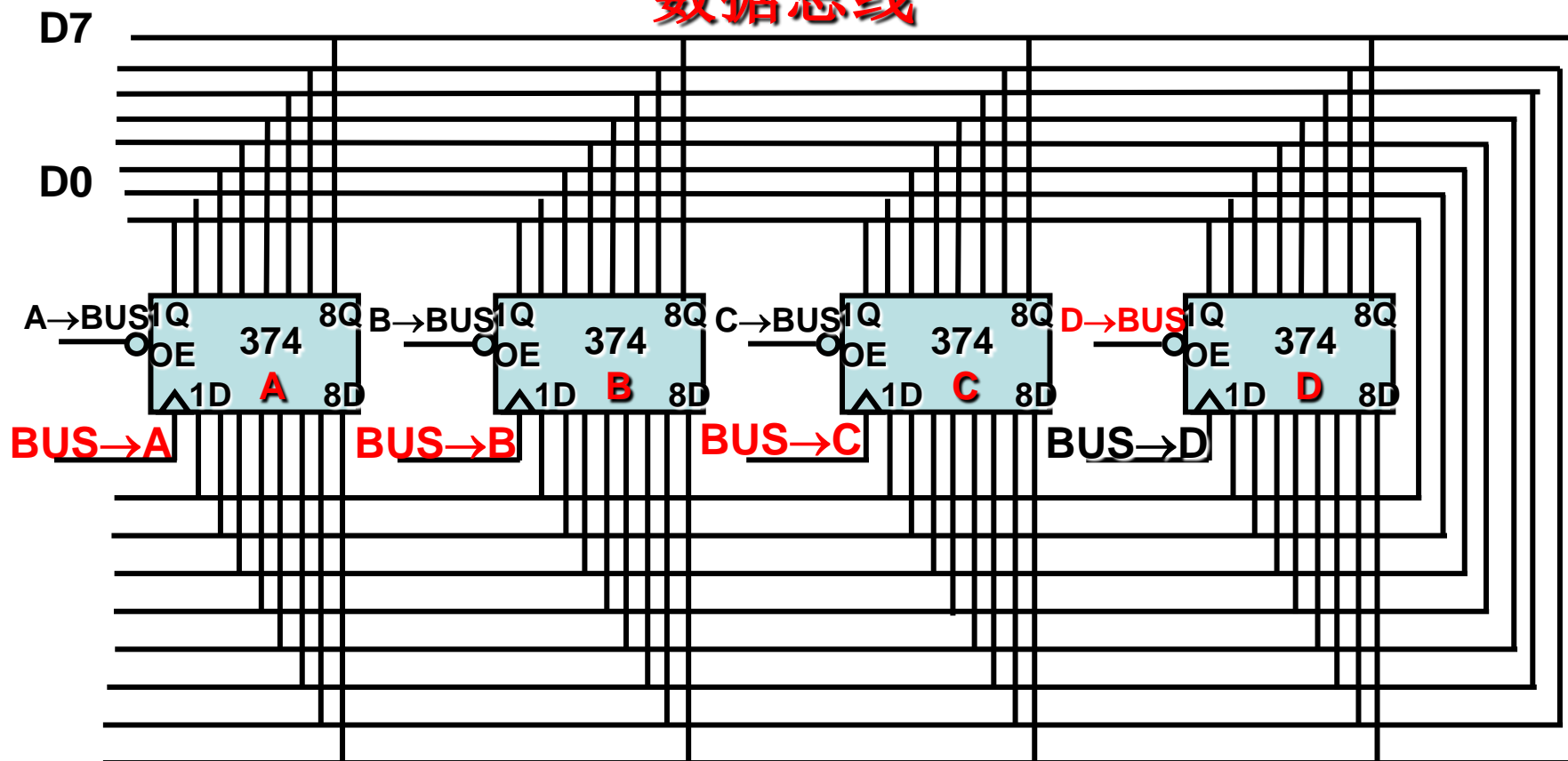
令：  $BUS \rightarrow A = BUS \rightarrow B = BUS \rightarrow C = CP$ ;  
 $D \rightarrow BUS = -OE$ ;  
当CP前沿到来时，将  $D \rightarrow A、B、C$ 。

# 第七章 7.2

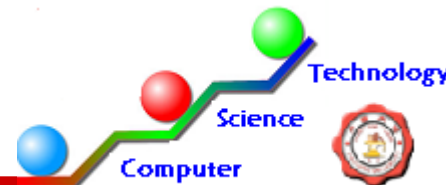


现以8位总线为例，设计此电路，如下图示：

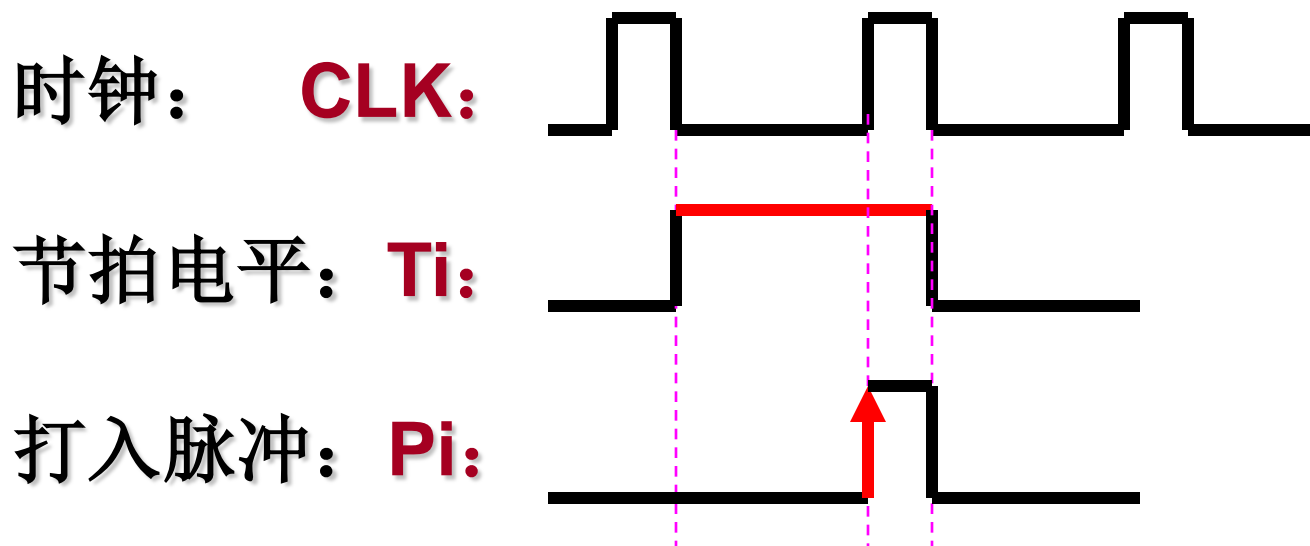
## 数据总线



## 第七章 7.2



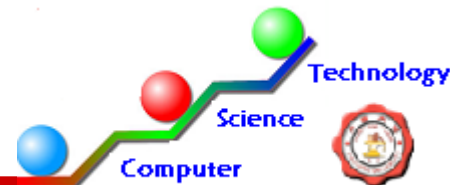
(2) 寄存器设置同(1)，由于本题中发送、接收不在同一节拍，因此总线需设**锁存器缓冲**，锁存器采用**74LS373**（电平使能输入）。节拍、脉冲配合关系如下：



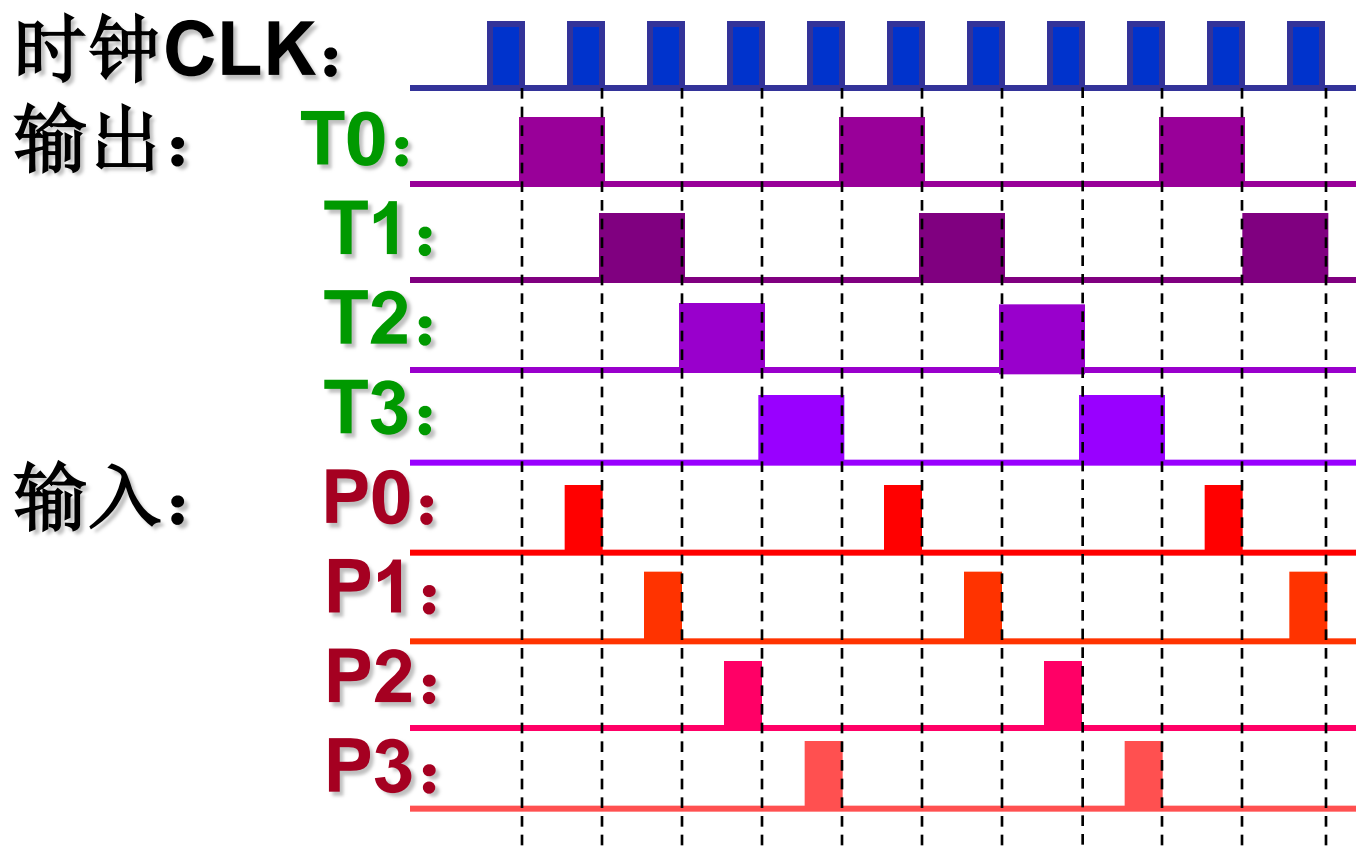
图中，脉冲**包**在电平中，为了**留有较多的**传送时间，脉冲设置在靠近电平**后沿处**。

[illegible]

# 第七章 7.2

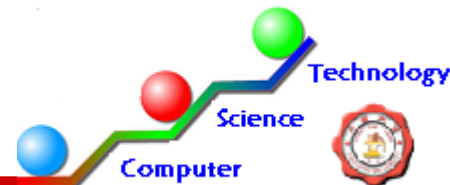


节拍、脉冲时序图如下：

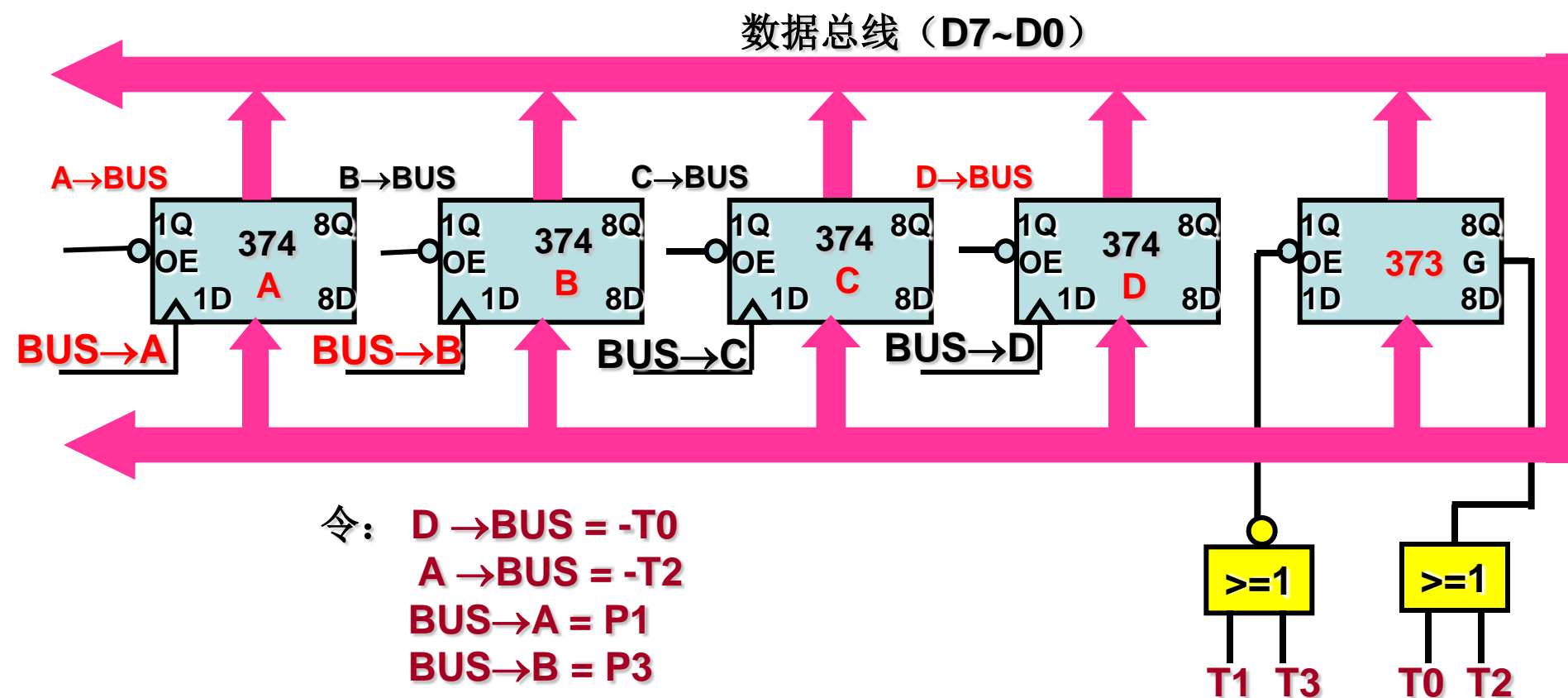




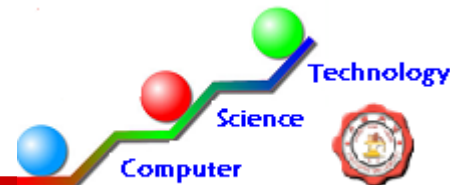
# 第七章 7.2



以8位总线为例，电路设计如下：（图中，A、B、C、D四个寄存器与数据总线的连接方法同上。）

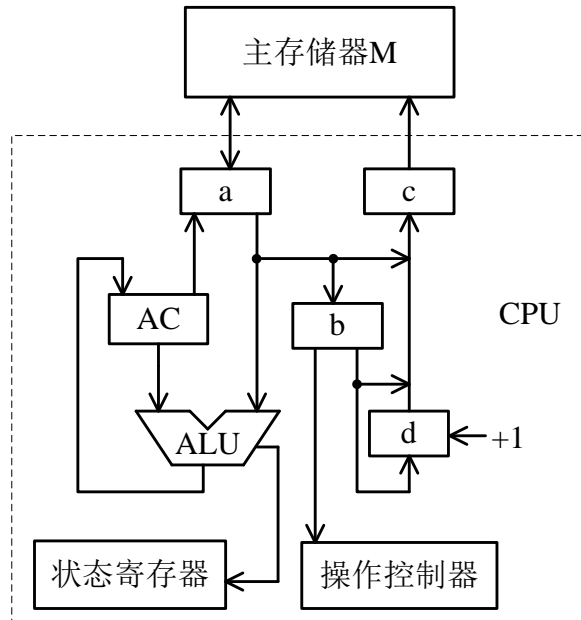


# 第七章 7.3

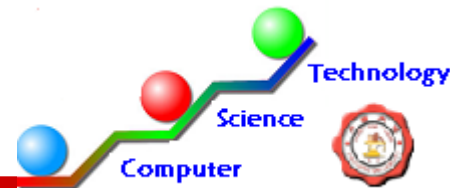


□ 7.3 若某CPU的数据通路结构如下图所示，其中有一个累加寄存器AC，一个状态条件寄存器和其它四个寄存器，各部分之间连接线的箭头表示信息传送方向。要求：

- (1) 写出图中a、b、c、d四个寄存器的名称；
- (2) 用寄存器传输语言描述指令从主存取到控制器的操作过程；
- (3) 设计一条加法指令，并用寄存器传输语言描述加法指令执行阶段的操作过程。



## 第七章 7.3



□ 题解:

□ (1) a —— 存储器数据缓冲寄存器MDR;

b —— 指令寄存器IR;

c —— 存储器地址寄存器MAR;

d —— 程序计数器PC;

□ (2)  $MAR \leftarrow (PC);$

$MDR \leftarrow M[MAR];$  MemRd

$IR \leftarrow (MDR);$

$PC \leftarrow (PC)+1,$  操作控制器  $\leftarrow (IR);$  +1

## 第七章 7.3



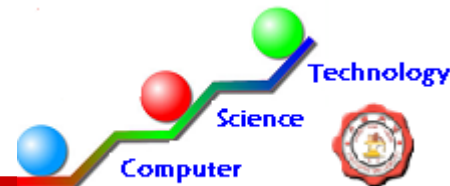
- (3) 假设加法指令为：add x，其中，x为主存单元地址。  
该指令的功能是将累加器AC的值与存储单元内容相加结果保存在累加器AC。

$MAR \leftarrow x;$

$MDR \leftarrow M[x] ; \text{MemRd}$

$ALU \leftarrow (MDR), ALU \leftarrow (AC) , AC \leftarrow ALU; +$

# 第七章 7.4



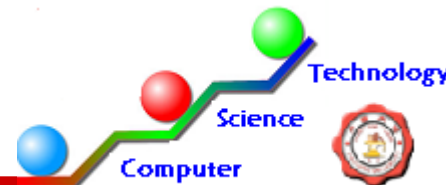
□ 7.4 欲在7.3.4给出的目标指令集中增加一条立即数加法指令`addi rt,rs,imm16`，若CPU采用单周期数据通路设计方案，请问在7.5.1中给出的图7-19数据通路能否支持该指令的执行？若不能，请问如何修改？并用指令周期流程图描述该指令的完整执行过程。

□ 题解：

○ 支持。

○ 指令周期流程见下页。

## 第七章 7.4



用RTL描述指令周期流程如下：

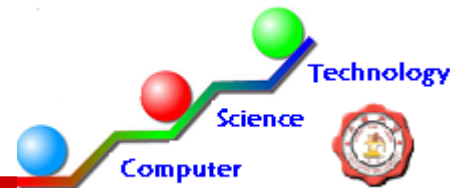
取指阶段：

- (1)  $\text{addr(IM)} \leftarrow (\text{PC})$ ;
- (2)  $\text{read(IM)}$ ;
- (3)  $\text{Add1\_B} \leftarrow (\text{PC}), (\text{PC})+4$ ;

执行阶段：

- (1)  $\text{R\_Reg1(RF)} \leftarrow \text{Inst [25-21]}, \text{ALU\_A} \leftarrow (\text{R\_data1})$ ;
- (2)  $\text{W\_Reg(RF)} \leftarrow \text{Inst [20-16]}$ ; **RegDst=0**
- (3)  $\text{SigExt16/32} \leftarrow \text{Inst [15-0]}, \text{ALU\_B} \leftarrow \text{SigExt16/32}$ ; **ALUSrc=1**
- (4) **ALU操作 (add)** ; **ALUOp=00, ALUCtrl=100**
- (5)  $\text{W\_data(RF)} \leftarrow \text{ALU\_C}, \text{PC} \leftarrow (\text{PC})+4$ ;  
**Branch=0, Jump=0, MemtoReg=0, RegWr=1, CLK  $\uparrow$**

# 第七章 7.6

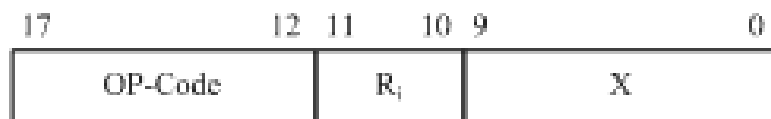


□ 7.6下图所示的CPU逻辑框图中，有两条独立的总线和两个独立的存储器。已知指令存储器IM最大容量为16384字（字长18位），数据存储器DM最大容量是65536字（字长16位）。各寄存器均有“打入”（Rin）和“送出”（Rout）控制命令，但图中未标出。

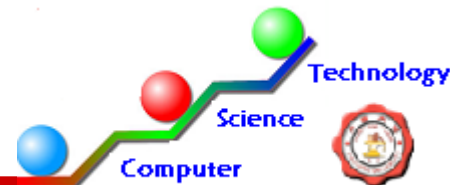
（1）指出下列各寄存器的位数：

程序计数器PC，指令寄存器IR，累加器 $AC_0$ 和 $AC_1$ ，通用寄存器 $R_0 \sim R_3$ ，指令存储器地址寄存器IAR，指令存储器数据寄存器IDR，数据存储器地址寄存器DAR，数据存储器数据寄存器DDR；

（2）设该CPU的指令格式为：

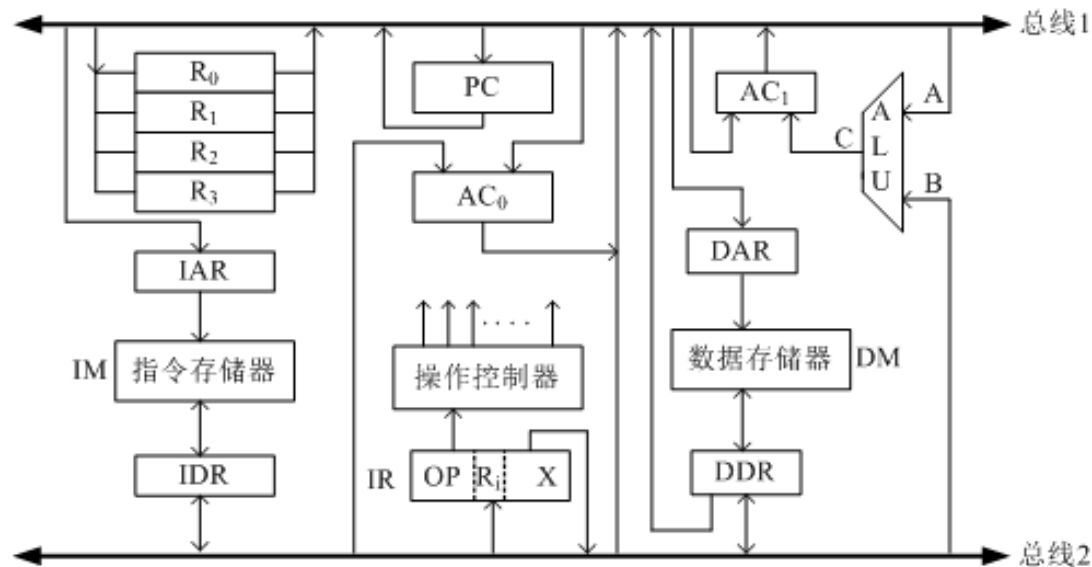


# 第七章 7.6



加法指令可写为“**ADD X (R<sub>i</sub>)**”，其功能是：

**$AC_1 \leftarrow (AC_0) + ((R_i) + X)$** ，其中  **$((R_i) + X)$**  部分通过寻址方式指向数据存储器，用指令周期流程描述**ADD**指令从取指令开始到执行结束的操作过程，并标明完成具体操作所需要的微操作控制信号。





# 第七章 7.6



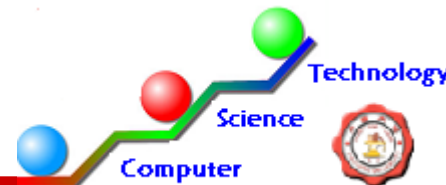
□ 题解:

(1)  $PC = IAR = 14\text{位}$ ;  $IR = IDR = 18\text{位}$ ;

$AC_0 = AC_1 = R_0 \sim R_3 = DDR = DAR = 16\text{位}$

(2) 接下页

# 第七章 7.6



取指令阶段：

- (1)  $IAR \leftarrow (PC)$ ;  $PC_{out}, IAR_{in}$
- (2)  $IDR \leftarrow IM[IAR]$ ;  $IM_{读}, IDR_{in}$   
 $PC \leftarrow (PC) + 1$ ;  $PC+1, PC_{in}$  (若PC具有自增功能)
- (3)  $IR \leftarrow (IDR)$ ;  $IDR_{out}, IR_{in}$

指令执行阶段：

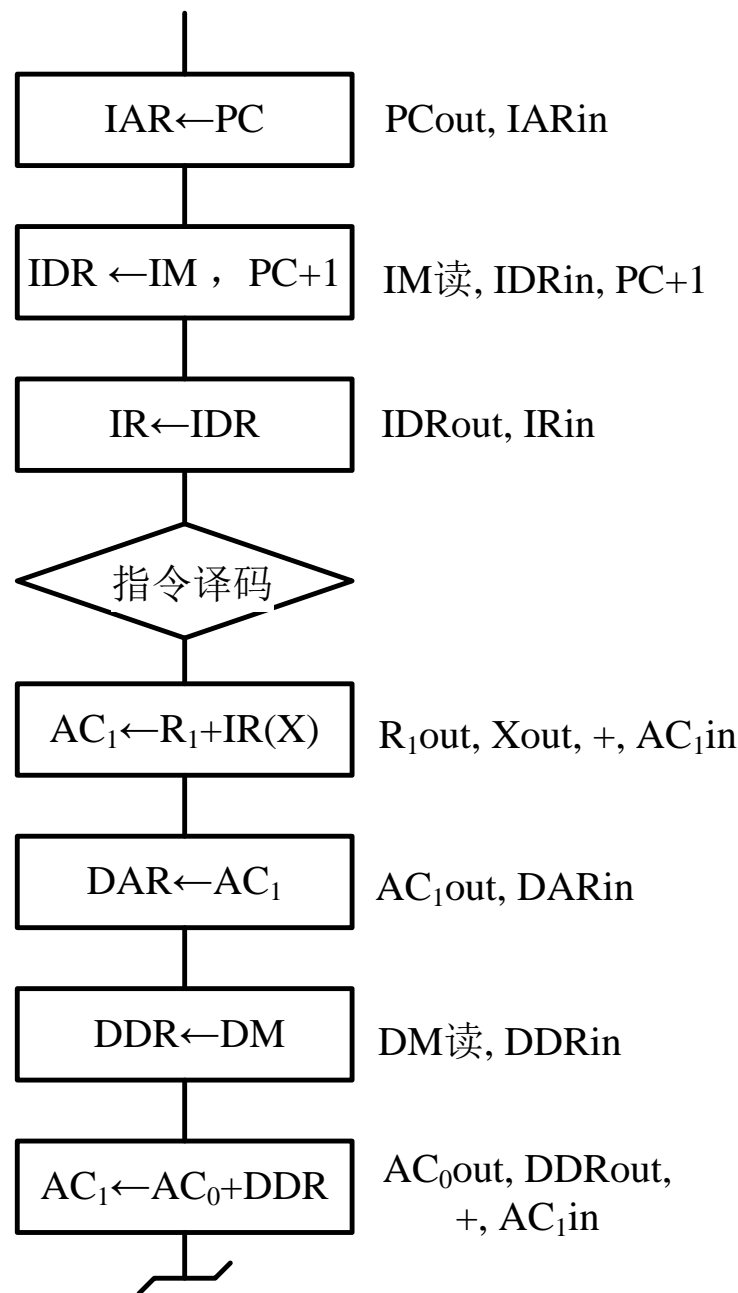
- (1)  $ALU\_A \leftarrow (R_i), ALU\_B \leftarrow (IR)X$ ;  $Ri_{out}, (IR)X_{out}, Add$   
 $AC1 \leftarrow ALU\_C$ ;  $AC_1_{in}$
- (2)  $DAR \leftarrow (AC1)$ ;  $AC_1_{out}, DAR_{in}$
- (3)  $DDR \leftarrow DM[DAR]$ ;  $DM_{读}, DDR_{in}$
- (4)  $ALU\_A \leftarrow (AC0), ALU\_B \leftarrow (DDR)$ ;  $AC_0_{out}, DDR_{out}, Add$   
 $AC1 \leftarrow (ALU\_C)$ ;  $AC_1_{in}$

注：分号左边为微操作，分号右边为所需微命令（微操作控制信号）

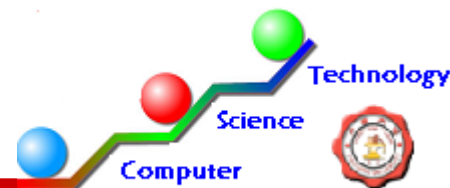
# 第七章 7.6

## 解法2:

用指令周期流程图描述。  
左边为指令周期流程图，  
右边为所需微命令信号序列。



# 第七章 7.7

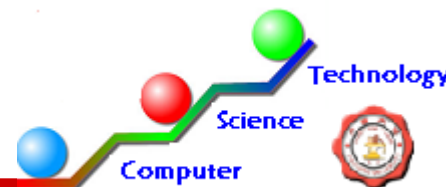


□ 7.7 某CPU的主频为4MHz，各类指令的平均执行时间和使用频度如下表所示。

指令类别↕	存取↕	加、减、比较、转移↕	乘除↕	其它↕
平均指令执行时间↕	0.6 $\mu$ s↕	0.8 $\mu$ s↕	10 $\mu$ s↕	1.4 $\mu$ s↕
使用频度↕	35%↕	50%↕	5%↕	10%↕

- (1) 试计算该CPU的速度（单位用MIPS表示）；
- (2) 若上述CPU主频提高为6MHz，则该CPU的速度又为多少？

# 第七章 7.7



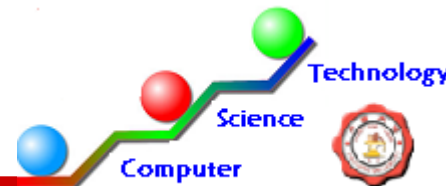
□ 题解:

$$(1) \text{ 平均指令执行时间} = 0.6 \times 35\% + 0.8 \times 50\% + 10 \times 5\% + 1.4 \times 10\% = 1.25 \mu\text{s}$$

$$\text{CPU速度} = 1/1.25\mu\text{s} = 0.8\text{MIPS}$$

$$(2) 0.8 \times (6/4) = 1.2\text{MIPS}$$

## 第七章 7.16



□ 7.16 某计算机CPU中有如下部件：**ALU**（具有+、-、直送V等功能），**移位器**（具有左移L、右移R、直送V等功能），主存储体**M**，主存数据寄存器**MDR**，主存地址寄存器**MAR**，指令寄存器**IR**，程序计数器**PC**（具有自增+1功能），通用寄存器**R0~R3**，暂存器**C**（连ALU左输入端）、**D**（连ALU右输入端）。

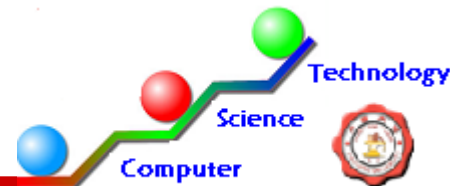
(1) 若上述逻辑部件按单总线结构组成CPU数据通路，请画出指令“**SUB (R0), R3**”的**指令周期流程图**，该指令的含义为：源操作数的**有效地址**在寄存器**R0**中，目的**操作数**在**R3**中，目的操作数减去源操作数后结果送到目的操作数的有效地址中；（**注**：减法时，要求**被减数**送ALU的左输入端，**减数**送ALU的右输入端）

## 第七章 7.16

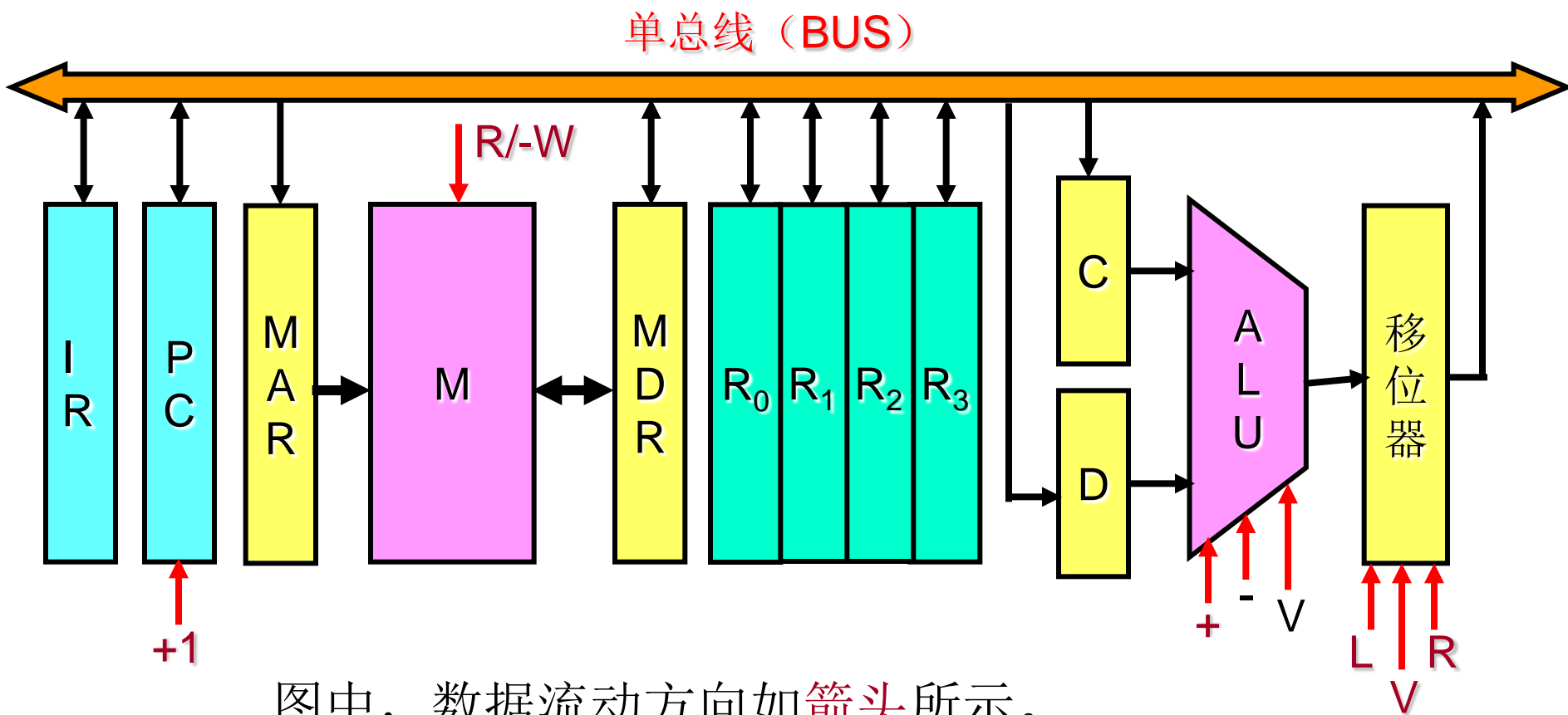


- (2) 若用类似**MDRo**的方式表示寄存器输出类微命令，类似**MDRi**的方式表示寄存器输入类微命令，**1→R**表示主存读命令，**1→W**表示主存写命令，MDR、MAR与主存储体M之间，暂存器C、D与ALU输入端之间是**直通**的，**不需要微命令控制**。请写出对应该指令周期流程图所需的**全部微操作命令序列**。

# 第七章 7.16



解：(1) 采用单总线结构的CPU硬件框图如下：



图中，数据流动方向如箭头所示。

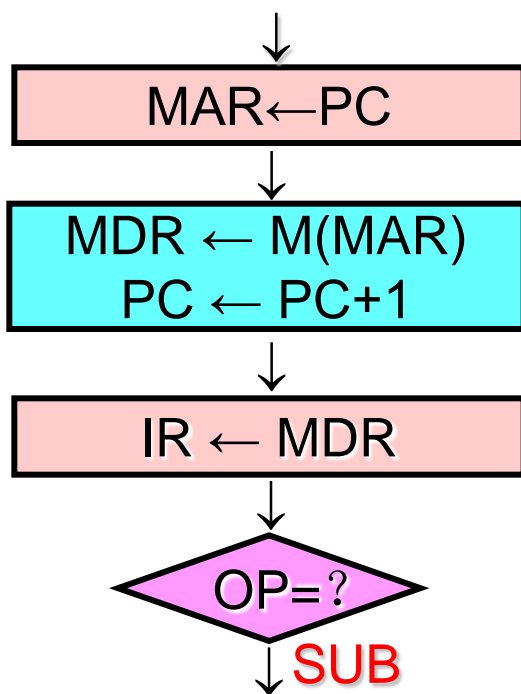


# 第七章 7.16



(1) SUB(R0), R3指令流程图

(2) 对应的全部微操作命令

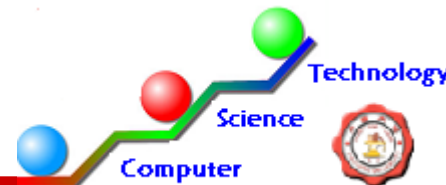


$PC_0, MAR_i$

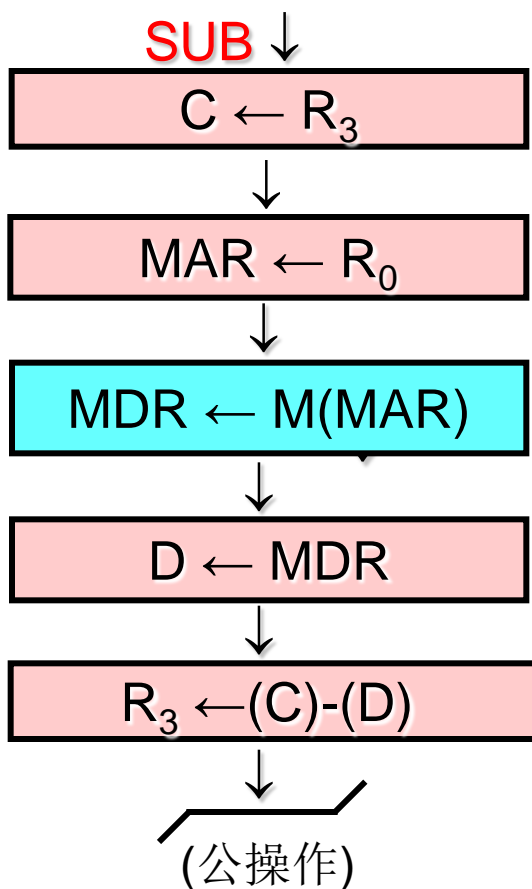
$1 \rightarrow R, +1$

$MDR_0, IR_i$

# 第七章 7.16



(1)



(2)

$R3_0, C_i$

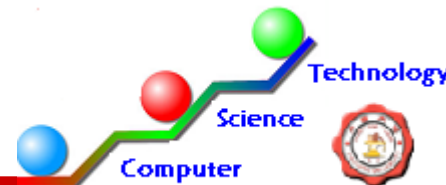
$R0_0, MAR_i$

$1 \rightarrow R$

$MDR_0, D_i$

$-, V, R3_i$

# 第七章 7.20



□ 7.20 由于指令周期= $\text{CPI} \times \text{时钟周期}$ ，所以主频和CPI是影响CPU性能的两个重要参数。但是，同时降低这两个参数往往是不可能的，在设计过程中始终存在这两个参数的权衡。具体方法如下：

(1) 以提高CPI为代价，提高处理器的主频；

(2) 降低主频换取较小的CPI。

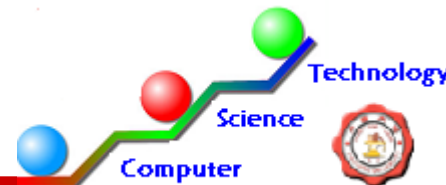
请分析MIPS多周期数据通路，分别说明这两种方法实现的具体措施。

# 第七章 7.20



- 解答：
- MIPS多周期CPU数据通路的设计思想是：将指令执行过程分成时间大致相同的多个阶段，每个阶段完成少数几个微操作，各阶段的最长时间设置为一个时钟周期。
- （1）提高CPI，即增加时钟周期数。这种方法的具体措施是把指令执行过程划分成较多的阶段，每个阶段完成较少的微操作。
- （2）降低主频，即延长时钟周期。这种方法的具体措施是把指令执行过程划分成较少的阶段，每个阶段完成较多的微操作。

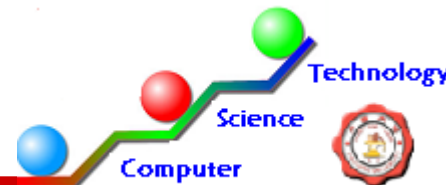
# 第七章 7.21



- 7.21 有三台计算机M1、M2和M3，CPU主频分别为：3.2GHz、2.8GHz、1GHz。假设这三台机器可以运行同一个指令系统，该指令系统由三类指令构成，各类指令在程序中的使用频度以及在三个CPU中实现的CPI如下表所示。则每台计算机的运行速度分别是多少？

指令类型	使用频度	CPI		
		M1	M2	M3
存数/取数	50%	5	4	3
运算	35%	4	3	3
其它	15%	3	3	3

# 第七章 7.21



□ 解答:

□ M1:

$$\text{平均CPI} = 50\% * 5 + 35\% * 4 + 15\% * 3 = 4.35$$

$$\text{平均速度} = 1 / (4.35 * 1 / 3.2\text{G}) = 736\text{MIPS}$$

□ M2:

$$\text{平均CPI} = 50\% * 4 + 35\% * 3 + 15\% * 3 = 3.5$$

$$\text{平均速度} = 1 / (3.5 * 1 / 2.8\text{G}) = 800\text{MIPS}$$

□ M3:

$$\text{平均CPI} = 50\% * 3 + 35\% * 3 + 15\% * 3 = 3$$

$$\text{平均速度} = 1 / (3 * 1 / 1\text{G}) = 333\text{MIPS}$$