

1. 假定计算机 M1 和 M2 具有相同的指令集体系结构 (ISA)，主频分别为 1.5GHz 和 1.2GHz。在 M1 和 M2 上运行某基准程序 P，平均 CPI 分别为 2 和 1，则程序 P 在 M1 和 M2 上运行时间的比值是

解答：

CPU 时间=指令数 × CPI × 时钟周期时间

运行一个基准程序，所以指令数相同，运行时间之比为：
$$\frac{2}{\frac{1.5}{1}} = 1.6$$

2. 假定主存地址为 32 位，按字节编址，主存和 Cache 之间采用直接映射方式，主存块大小为 4 个字，每字 32 位，采用写回(Write Back)方式，则能存放 4K 字数据的 Cache 的总容量的位数至少是()。

A.146K B. 147K C.148K D.158K

解答：Cache 和主存直接映射方式的规则为：主存储器分为若干区，每个区与缓存容量相同；每个区分为若干数据块，每个块和缓存块容量相同；主存中某块只能映射到 Cache 的一个特定的块中。本题中，Cache 总共存放 4K 字数据，块大小为 4 个字，因此 Cache 被分为 $4K/4=1K$ 个块，由 10 位表示。块内共 16 字节，所以由 4 位表示，于是标记位为 $32-10-4=18$ 位。所以，Cache 的每一行需要包含所存的数据 4 个字，每个字 32 位，18 位标记位、一个有效位和一个一致性维护位(写回方式)，因此总容量为： $(4 \times 32 + 18 + 1 + 1) \times 1K = 148K$ 。

3. 某计算机按字节编址，指令字长固定且只有两种指令格式，其中三地址指令 29 条，二地址指令 107 条，每个地址字段为 6 位，则指令字长至少应该是

A. 24 位 B. 26 位 C. 28 位 D. 32 位

解答：三地址指令有 29 条，所以它的操作码至少为 5 位。以 5 位进行计算，它剩余 $32-29=3$ 种操作码给二地址。而二地址另外多了 6 位给操作码，因此它数量最大达 $3 \times 64=192$ 。所以指令字长最少为 23 位 ($6+6+6+5=23$)，因为计算机按字节编址，需要是 8 的倍数，所以指令字长至少应该是 24 位，选 A。

4. I/O 指令实现的数据传送通常发生在 。

- A. I/O 设备和 I/O 端口之间 B. 通用寄存器和 I/O 设备之间
C. I/O 端口和 I/O 端口之间 D. 通用寄存器和 I/O 端口之间

解答：

在执行一条指令时，CPU 使用地址总线选择所请求的 I/O 端口，使用数据总线在 CPU 寄存器和端口之间传输数据。所以选 D。

5. 若磁盘转速为 7200 转/分，平均寻道时间为 8ms，每个磁道包含 1000 个扇区，存取时间大约为（12.2ms）

解答：磁盘的平均寻址时间包括平均寻道时间和平均等待时间。平均寻道时间为 8ms，平均等待时间为 $60/7200 \times 0.5 = 4.165\text{ms}$ ，总计为 $8 + 4.165 = 12.165\text{ms}$

6. 某计算机的 CPU 主频为 500MHz，CPI 为 5（即执行每条指令平均需 5 个时钟周期）。假定某外设的数据传输率为 0.5MB/s，采用中断方式与主机进行数据传送，以 32 位为传输单位，对应的中断服务程序包含 18 条指令，中断服务的其他开销相当于 2 条指令的执行时间。请回答下列问题，要求给出计算过程。

1) 在中断方式下，CPU 用于该外设 I/O 的时间占整个 CPU 时间的百分比是多少？

2) 当该外设的数据传输率达到 5MB/s 时，改用 DMA 方式传送数据。假定每次 DMA 传送块大小为 5000B，且 DMA 预处理和后处理的总开销为 500 个时钟周期，则 CPU 用于该外设 I/O 的时间占整个 CPU 时间的百分比是多少（假设 DMA 与 CPU 之间没有访存冲突）？

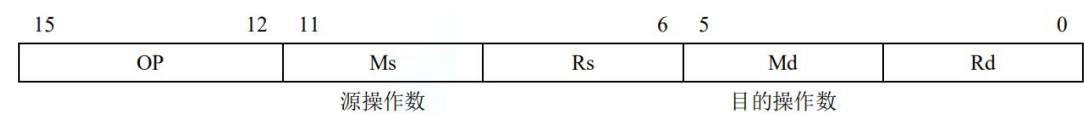
解答：

1) 按题意，外设每秒传送 0.5MB，中断时每次传送 4B。中断方式下，CPU 每次用于数据传送的时钟周期为 $5 \times 18 + 5 \times 2 = 100$ 。为达到外设 0.5MB/s 的数据传输率，外设每秒申请的中断次数为 $0.5\text{MB}/4\text{B} = 125\,000$ 。1s 内用于中断的开销为 $100 \times 125\,000 = 12\,500\,000 = 12.5\text{M}$ 个时钟周期。CPU 用于外设 I/O 的时间占整个 CPU 时间的百分比为 $12.5\text{M}/500\text{M} = 2.5\%$ 。

2) 当外设数据传输率提高到 5MB/s 时，改用 DMA 方式传送，每次 DMA 传

送 5 000B，1s 内需产生的 DMA 次数为 5MB/5 000B=1 000。

7. 某计算机字长为 16 位，主存地址空间大小为 128KB，按字编址。采用单字长指令格式，指令各字段定义如图所示。



转移指令采用相对寻址方式，相对偏移量用补码表示，寻址方式定义见表：

Ms/Md	寻址方式	助记符	含义
000B	寄存器直接	Rn	操作数=(Rn)
001B	寄存器间接	(Rn)	操作数=((Rn))
010B	寄存器间接、自增	(Rn)+	操作数=((Rn)), (Rn)+1→Rn
011B	相对	D(Rn)	转移目标地址=(PC)+(Rn)

请回答下列问题：

- 1) 该指令系统最多可有多少条指令？该计算机最多有多少个通用寄存器？存储器地址寄存器（MAR）和存储器数据寄存器（MDR）至少各需要多少位？
- 2) 转移指令的目标地址范围是多少？
- 3) 若操作码 0010B 表示加法操作（助记符为 add），寄存器 R4 和 R5 的编号分别为 100B 和 101B，R4 的内容为 1234H，R5 的内容为 5678H，地址 1234H 中的内容为 5678H，地址 5678H 中的内容为 1234H，则汇编语言为 “add(R4), (R5)+”（逗号前为源操作数，逗号后为目的操作数）对应的机器码是什么（用十六进制表示）？该指令执行后，哪些寄存器和存储单元中的内容会改变？改变后的内容是什么？

解答：

- 1) 操作码占 4 位，则该指令系统最多可有 $2^4=16$ 条指令。操作数占 6 位，其中寻址方式占 3 位、寄存器编号占 3 位，因此该机最多有 $2^3=8$ 个通用寄存器。主存地址空间大小为 128KB，按字编址，字长为 16 位，共有 $128KB/2B=216$ 个存储单元，因此 MAR 至少为 16 位；因为字长为 16 位，故 MDR 至少为 16 位。
- 2) 寄存器字长为 16 位，PC 和 Rn 可表示的地址范围均为 $0\sim2^{16}-1$ ，而主存地址空间为 2^{16} ，故转移指令的目标地址范围为 $0000H\sim FFFFH (0\sim2^{16}-1)$ 。将对应的机器码写成十六进制形式为 $0010\ 0011\ 0001\ 0101B=2315H$ 。

3) 汇编语言 `add(R4), (R5)` 对应的机器码为

字段	OP	MS	Rs	Md	Rd
内容	0010	001	100	010	101
说明	add	寄存器间接	R4	寄存器间接、自增	R5

该指令的功能是将 R4 的内容所指存储单元的数据与 R5 的内容所指存储单元的数据相加，并将结果送入 R5 的内容所指存储单元中。

`(R4)=1234H, (1234H)=5678H;`

`(R5)=5678H, (5678H)=1234H;`

执行加法操作 `5678H+1234H=68ACH`，之后 R5 的内容从 5678H 变为 5679H，存储单元 5678H 中的内容变为该指令的计算结果 68ACH。

8. 某计算机的主存地址空间大小为 256MB，按字节编址。指令 Cache 和数据 Cache 分离，均有 8 个 Cache 行，每个 Cache 行大小为 64B，数据 Cache 采用直接映射方式。现有两个 功能相同的程序 A 和 B，其伪代码如下：

<pre> 程序 A: int a[256][256] int sum_array1() { int i,j,sum=0; for(i=0;i<256;i++) for(j=0;j<256;j++) sum+=a[i][j]; return sum; } </pre>	<pre> 程序 B: int a[256][256] int sum_array2() { int i,j,sum=0; for(j=0;j<256;j++) for(i=0;i<256;i++) sum+=a[i][j]; return sum; } </pre>
--	--

假定 `int` 类型数据用 32 位补码表示，程序编译时 `i`、`j`、`sum` 均分配在寄存器中，数组 `a` 按行优先方式存放，其首地址为 320（十进制数）。请回答下列问题，要求说明理由或给出计算过程。

- 1) 若不考虑用于 Cache 一致性维护和替换算法的控制位，则数据 Cache 的总容量为多少？
- 2) 数组元素 `a[0][31]` 和 `a[1][1]` 各自所在的主存块对应的 Cache 行号分别是多少

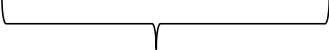
(Cache 行号从 0 开始) ?

3) 程序 A 和 B 的数据访问命中率各是多少? 哪个程序的执行时间更短?

解答:

1) 不考虑用于 Cache 一致性维护和替换算法的控制位。地址总长度为 28 位 ($2^{28}=256\text{M}$), 块内地址 6 位 ($2^6=64$), Cache 块号 3 位 ($2^3=8$), 故 Tag 的位数为 $28-6-3=19$ 位, 还需使用一个有效位, 题中数据 Cache 行的结构如下图所示:

1	19 bit	64B=512bit
有效位	标记位	每行存储的数据



该行对应的标记项

数据 Cache 共有 8 行, 数据 Cache 的总容量为 $8*(64+20/8)\text{B}=532\text{B}$ 。

2) 数组按行优先方式存放, 首地址为 320, 数组元素占 4 个字节。a[0][31]所在的主存块对应的 Cache 行号为 $(320+31*4)/64=6$; a[1][1]所在的主存块对应的 Cache 行号为 $(320+256*4+1*4)/64\%8=5$ 。

3) 数组 a 按行存放, 程序 A 按行存取。每个字块中存放 16 个 int 型数据, 除访问的第一个不命中, 随后 15 个全都命中, 访问全部字块都符合这一规律, 且数组大小为字块大小的整数倍, 故程序 A 的命中率为 $15/16=93.75\%$ 。

程序 B 逐列访问数组 a, Cache 总容量为 $64\text{B}*8=512\text{B}$, 数组 a 一行的大小为 1KB, 正好是 Cache 容量的 2 倍, 可知不同行的同一列数组元素使用的是同一个 Cache 单元, 故逐列访问每个数据时, 都会将之前的字块置换出, 也即每次访问都不会命中, 命中率为 0。

由于从 Cache 读数据比从主存读数据快很多, 所以程序 A 的执行比程序 B 快得多。

9. 假定在一个 8 位字长的计算机中运行如下 C 程序段:

```

unsigned int  x=134;
unsigned int  y=246;
int  m=x;
int  n=y;
unsigned int  z1=x-y;
unsigned int  z2=x+y;
int  k1=m-n;
int  k2=m+n;

```

若编译器编译时将 8 个 8 位寄存器 R1~R8 分别分配给变量 x、y、m、n、z1、z2、k1 和 k2。

请回答下列问题。（提示：带符号整数用补码表示。）

- （1）执行上述程序段后，寄存器 R1、R5 和 R6 的内容分别是什么（用十六进制表示）？
- （2）执行上述程序段后，变量 m 和 k1 的值分别是多少（用十进制表示）？
- （3）上述程序段涉及带符号整数加/减、无符号整数加/减运算，这四种运算能否利用同一个加法器辅助电路实现？简述理由。
- （4）计算机内部如何判断带符号整数加/减运算的结果是否发生溢出？上述程序段中，哪些带符号整数运算语句的执行结果会发生溢出？

解答：

1) $134=128+6=10000110B$ ，所以 x 的机器数为 10000110B，故 R1 的内容为 86H。
 $246=255-9=11110110B$ ，所以 y 的机器数为 11110110B。

$x-y$: $10000110+00001010=(0)10010000$ ，加法器的进位为 0，故 R5 的内容为 90H。

$x+y$: $10000110+11110110=(1)01111100$ ，加法器的进位为 1，故 R6 的内容为 7CH。

2) m 的机器数与 x 的机器数相同，皆为 86H=1000 0110B。因为带符号整数 m（用补码表示）时，其值为-1111010B=-122。

m-n 的机器数与 x-y 的机器数相同，皆为 90H=1001 0000B，因为带符号整数 k1（用补码表示）时，其值为-111 0000B=-112。

3) 能。

n 位加法器实现的是模 2^n 无符号整数加法运算。对于无符号整数 a 和 b，a+b 可以直接用加法器实现，而 a-b 可用 a 加 b 的补数实现，即 $a-b=a+[-b]_{\text{补}} \pmod{2^n}$ ，

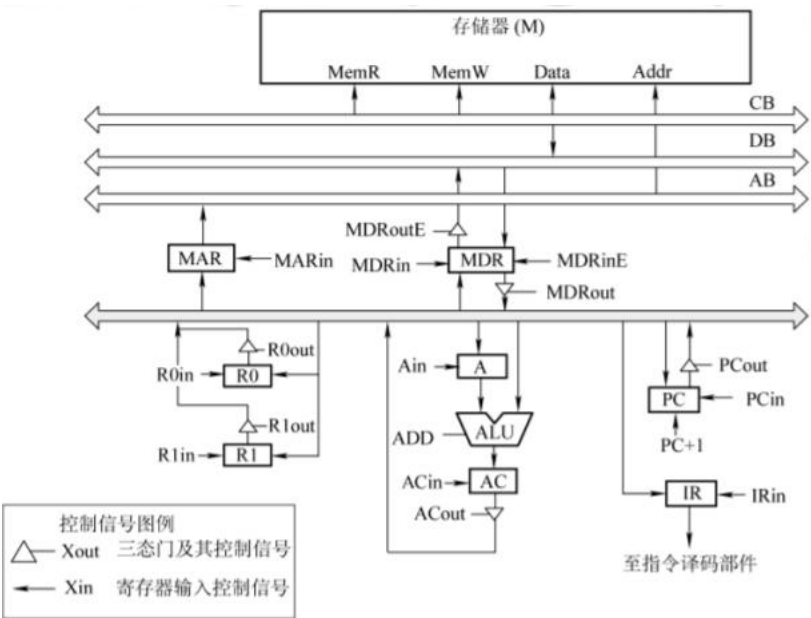
所以 n 位无符号整数加/减运算都可在 n 位加法器中实现。

由于带符号整数用补码表示，补码加/减运算公式为： $[a+b]_{补}=[a]_{补}+[b]_{补} \pmod{2^n}$ ， $[a-b]_{补}=[a]_{补}+[-b]_{补} \pmod{2^n}$ ，所以 n 位带符号整数加/减运算都可 n 位加法器中实现。

4) 带符号整数加/减运算的溢出判断规则为：若加法器的两个输入端（加法）的符号相同，且不同于输出端（和）的符号，则结果溢出，或加法器完成加法操作时，若次高位的进位和最高位的进位不同，则结果溢出。

最后一条语句执行时会发生溢出。因为 $1000\ 0110+1111\ 0110=(1)0111\ 1100$ ，加法器的进位为 1，根据上述溢出判断规则，可知结果溢出。

10. 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 A-2 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 打入 MDR，MDRin 为 1 表示允许数据从内总线打入 MDR。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 的内容所指主存单元的数据相加，并将结果送入 R1 的内容所指主存单元中保存。



表给出了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。

解答：

首先分析题目中的示例：

取指令的功能是根据 PC 的内容所指主存地址，取出指令代码，经过 MDR，最终送至 IR。

C1: (PC)→MAR

在读写存储器前，必须先将地址（这里为(PC)）送至 MAR。

C2: M(MAR)→MDR, (PC)+1→PC

读写的数据必须经过 MDR，指令取出后 PC 自增 1。

C3: (MDR)→IR

然后将读到 MDR 中指令代码送至 IR 进行后续操作。

指令“ADD (R1),R0”的操作数一个在主存中，一个在寄存器中，运算结果在主存中。根据指令功能，要读出 R1 的内容所指的主存单元，必须先将 R1 的内容送至 MAR，即(R1)→MAR。而读出的数据必须经过 MDR，即 M(MAR)→MDR。因此，将 R1 的内容所指主存单元的数据读出到 MDR 的节拍安排如下：

C5: (R1)→MAR C6: M(MAR)→MDR

ALU 一端是寄存器 A，MDR 或 R0 中必须有一个先写入 A 中，如 MDR。

C7: (MDR)→A

然后执行加法操作，并将结果送入寄存器 AC。

C8: (A)+(R0)→AC

之后将加法结果写回到 R1 的内容所指主存单元，MAR 中的内容没有改变。

C9: (AC)→MDR C10: (MDR)→M(MAR)

有效控制信号的安排，只需看数据是流入还是流出，如流入寄存器 X 就是 Xin，流出寄存器 X 就是 Xout。还需注意其他特殊控制信号 PC+1、Add 等。

所以答案如下：

时钟	功能	有效控制信号
C5	MAR←(R1)	R1out,MARin
C6	MDR←M(MAR)	MemR,MDRinE
C7	A←(MDR)	MDRout,Ain
C8	AC←(A)+(R0)	R0out,Add,ACin

C9	$MDR \leftarrow (AC)$	ACout, MDRin
C10	$M(MAR) \leftarrow (MDR)$	MDRoutE, MemW