CH1

1.5 将下列 16 进制数转换为 10 进制数，将 10 进制数转换成 16 进制数

789AH，0CEFH，1234D，7890D

[解析] 789AH=30874D 0CEFH=3311D 1234D=4D2H 7890D=1ED2H

1.6 将下列 10 进制数转换成 8 位二进制补码：

19，63，－1，－44，127，－127

[解析] 19=00010011 原=00010011 补

63=00111111 原=00111111 补

-1=10000001 原=11111111 补

-44=10101100 原=11010100 补

127=01111111 原=01111111 补

-127=11111111 原=10000001 补

1.8 已知：

1）$[x_1]_{补}$=0000 0001，$[y_1]_{补}$=1111 1111

2）$[x_2]_{补}$=0101 1110，$[y_2]_{补}$=0011 0111

3）$[x_3]_{补}$=1001 1110，$[y_3]_{补}$=1100 0101

4）$[x_4]_{补}$=0110 1110，$[y_4]_{补}$=1000 0100

请计算$[x_i]_{补}+[y_i]_{补}$，($i=1\sim4$)，并判断结果是否出现溢出?

[解析] 1）0000 0000 未溢出

2）1001 0101 溢出

3）0110 0011 溢出

4）1111 0010 未溢出

注：若记符号位向前进位为 CP，次高位向前进位为 CF，当且仅当 CP^CF=1 时，结果发生溢出

[补充 1.16] 冯•诺依曼结构计算机中数据采用二进制编码表示，其主要原因是（D）

I. 二进制的运算规则简单

II. 制造两个稳态的物理器件较容易

III. 便于用逻辑门电路实现算术运算

A) 仅 I、II

B) 仅 I、III

C) 仅 II、III

D) I、II、III

[解析] 常识，略。

[补充 1.17] 假定带符号整数采用补码表示，若 int 型变量 *x* 和 *y* 的机器数分别是 FFFF FFDFH 和 0000 0041H ，则 *x*、*y* 的值以及 *x-y* 的机器数分别是（C）

A) *x* = -65，*y* = 41，*x-y* 的机器数溢出

B) *x* =-33，*y* = 65，*x-y* 的机器数为 FFFF FF9DH

C) x = -33，y = 65，x-y 的机器数为 FFFF FF9EH

D) x=-65，y = 41，x-y 的机器数为 FFFF FF96H

[解析]

$\because 4\times16+1=65,\ \therefore 0000\ 0041H=65D$

可知只有 B 和 C 可能正确

$x=-33D,\ y=65D \Longrightarrow x-y=-98D$

方法 1：依据补码定义计算，

$2^{32}-98$

$=(2^{32}-1)-97$

$=[(2^{32}-1)-(2^8-1)]+[(2^8-1)-97]$

$=FFFFFF00H+158$

$=FFFFFF00H+(128+16+8+4+2)$

$=FFFFFF00H+1001\ 1110B$

$=FFFFFF00H+9EH$

$=FFFF\ FF9EH$

方法 2：真值取反+1 得到补码

$98D=96D+2D=6\times16+2=62H=0110\ 0010B$

低 7 位取反得， 001 1101B

+1 得，001 1110B

故 98D 的补码是，1001 1110B = 9EH

[补充 1.18] 已知带符号整数用补码表示，float 型数据用 IEEE754 标准表示。假定变量 x 的类型只可能是 int 或 float，当 x 的机器数为 C800 0000H 时，x 的值可能是（A）

A) -7×2^{27}

B) -2^{16}

C) 2^{17}

D) 25×2^{27}

[解析]

（1）若 x 是 int 类型

$C800\ 0000H=1100\ 1000\ 0000\ldots0000B$

方法 1：补码定义

因为 x 是补码表示的负数，故 $1100\ 1000\ 0000\ldots0000B=2^{31}+2^{30}+2^{27}=2^{32}-|x|$

$|x|=2^{32}-(2^{31}+2^{30}+2^{27})=2^{27}\times(2^5-2^4-2^3-2^0)=2^{27}\times7$

$\Longrightarrow x=-7\times2^{27}$

方法 2：真值取反+1 得到补码

除符号位外取反，

$1100\ 1000\ 0000\ldots0000B \Longrightarrow 1011\ 0111\ 1111\ldots1111B$

除符号位外部分-1，

$1011\ 0111\ 1111\ldots1B\Longrightarrow 1011\ 1000\ 0000\ldots0000B$

$x=-(2^{29}+2^{28}+2^{27})=-(2^2+2^1+2^0)\times2^{27}=-7\times2^{27}$

（2）若 x 是 float 类型

依 FP32 定义，$(-1)^S \times 2^{e-127} \times (1+M)$

$C800\ 0000H = 1100\ 1000\ 0000 \ldots 0000B \Longrightarrow S=1,\ e=2^7+2^4=128+16,\ M=0$

故 $x=(-1)^1 \times 2^{64-127} \times (1+0) = -2^{17}$

[补充 1.19] 已知带符号整数用补码表示。变量 X，Y，Z 的机器数分别为 FFFDH，FFDFH，7FFCH，下列结论中，正确的是（D）

A) 若 X，Y，Z 为无符号整数，则 Z<X<Y

B) 若 X，Y，Z 为无符号整数，则 X<Y<Z

C) 若 X，Y，Z 为带符号整数，则 X<Y<Z

D) 若 X，Y，Z 为带符号整数，则 Y<X<Z

[解析]

若 X，Y，Z 为无符号整数，FFFDH > FFDFH > 7FFCH $\Longrightarrow X>Y>Z \Longrightarrow$ A 和 B 错误

若 X，Y，Z 为有符号整数，FFFDH <0, FFDFH<0, 7FFCH>0 $\Longrightarrow$ Z 最大

$FFFDH = 2^{16} - |X| \Longrightarrow |X| = 2^{16} - FFFDH = 2^{16} - (FFFFH - 2H)$

$FFDFH = 2^{16} - |Y| \Longrightarrow |Y| = 2^{16} - FFDFH = 2^{16} - (FFFFH - 20H)$

可知 |X|<|Y| $\Longrightarrow$ X>Y，故选 D

[补充 1.20] 下列数值中，不能用 IEEE~~E~~-754 浮点精确表示的（A）

A) 1.2

B) 1.25

C) 2.0

D) 2.5

[解析]

IEEE-754 浮点数，规格化意味着 23bits 的尾数（定点小数表示的尾数）中小数点在最高比特左边，且隐含了小数点左边的 1。

而$bit[22]=2^{-1}, bit[21]=2^{-2} \ldots\ldots bit[0]=2^{-23}$。

若一个数可以被表示为类似 $\sum_{i\in\mathbb{Z}} 2^i$ 形式，就可以被 IEEE-754 精确表示。

用排除法，B、C、D 的小数部分都是2^{-n}形式，一定可以被表示。故选 A。

进一步解读，令集合$\Phi=\{-1,-2,\cdots,-23\}$，令集合$\Psi=\{-127,-126,\cdots,127,128\}$

若一个数可以被表示为类似 $\sum_{i=\phi+\psi,\phi\in\Phi,\psi\in\Psi} 2^i$ 形式，就可以被 IEEE-754 精确表示。

[补充 1.21] DeepSeek-V3 使用 FP8（8 位浮点数）来提高计算速度并减少训练期间的显存使用量。FP8 是一种 8 位浮点数表示法，FP8 的详细介绍可以参考 https://developer.nvidia.com/zh-cn/blog/fp8-precision-performance/ 。FP8 采取 E4M3 和 E5M2 两种表示方式，其中 E 代表指数位（Exponent），M 代表尾数位（Mantissa），具体信息如下表所示。

Table 1: Details of FP8 Binary Formats

	E4M3	E5M2
Exponent bias	7	15
Infinities	N/A	$S.11111.00_2$
NaN	$S.1111.111_2$	$S.11111.\{01,10,11\}_2$
Zeros	$S.0000.000_2$	$S.00000.00_2$
Max normal	$S.1111.110_2 = 1.75 * 2^8 = 448$	$S.11110.11_2 = 1.75 * 2^{15} = 57,344$
Min normal	$S.0001.000_2 = 2^{-6}$	$S.00001.00_2 = 2^{-14}$
Max subnorm	$S.0000.111_2 = 0.875 * 2^{-6}$	$S.00000.11_2 = 0.75 * 2^{-14}$
Min subnorm	$S.0000.001_2 = 2^{-9}$	$S.00000.01_2 = 2^{-16}$

某 8bits 二进制机器数为 1100 0011B，若其表示的是 E4M3 的 FP8 浮点数，其真值为（　　）；若其表示的是 E5M2 的 FP8 浮点数，其真值为（　　）。提示：（1）无需计算最后数值，写出表达式即可。（2）按照规格化浮点数格式（上表中 normal 情形）计算。

[解析]

E4M3，bias=7，1100 0011B， (-1)*(8-7)*(1+0.25+0.125)

E5M2，bias=15，1100 0011B， (-1)*(16-15)*(1+0.5+0.25)

CH2

2.10 什么是微指令？什么是微程序？控制 ROM 的作用是什么？

[解析] 抄书，合理即可

- 微指令：在机器的一个 CPU 周期中，一组实现一定操作功能的微命令的组合，构成一条微指令；微指令=微操作码+执行顺序位。
- 微程序：一条机器指令的功能是由许多条微指令组成的序列来实现的。这个微指令序列通常叫做微程序
- 控制 ROM：指令集中所有指令都对应一段微程序，他们存放在控制 ROM 中，每段微程序的每条微指令都有唯一的地址，当一条指令被执行时，指令译码器对指令操作码进行译码，找到对应微程序的存放地址，开始执行。

[2.13 修改版] 请参照例 2.1，分步骤写出第 3 条加法指令“ADD R1, R0, R1”的执行过程。

[解析] 过程解析可以不严谨，但是取指令和指令执行 2 而阶段必要要有。

1) 程序计数器 PC 的内容 0x20000008 送到地址生成部件，寻址内存单元。
2) 操作控制器发读信号，将 0x20000008 单元的内容“0x???? ????”读出至指令寄存器 IR。
3) 程序计数器 PC 自动加 4，指向下一条指令的存放地址 0x20000000C。
4) 指令译码器 ID 对指令操作码进行译码，操作控制器 OC 按照操作时序发出相应的控制信号。
5) 指令的地址码部分指示操作数在哪。本条指令中的两个源操作数存放在 R0 和 R1，运算结果存放到 R1。操作控制器 OC 控制实现 R0＋R1 → R1。

2.14 假设 A 和 B 是同一条总线所连接的两个存储器单元，总线位宽大于或等于存储单元的

位数。现在需要将 A 单元的内容传送到 B 单元中，能否在一个总线周期内完成传送任务？为什么？

[解析]

不能。数据传送过程必须分为两步，第一步先将源操作数从内存单元中加载到 CPU 内某个寄存器暂存，第二步将暂存的内容写入目的存储单元。

2.15 假设 I_j 和 I_{j+1} 是前后相继的两条指令，请举例说明指令流水线的“WAR”和“WAW”两种数据相关问题。

[解析]

WAR 示例：I_{j+1} 试图在指令 I_j 读一个数据之前写该数据，此时指令 I_j 读到的是被 I_{j+1} “篡改后的数据”。

WAW 示例：I_{j+1} 试图在指令 Ij 写数据之前写数据，这样最终结果将由 I_j 决定，而程序的本意是保留 I_{j+1} 的结果。

[补充 2.26] 在按字节编址，采用小端方式的 32 位计算机中，按边界对齐方式为以下 C 语言结构型变量 a 分配存储空间。

```
struct record{
    short x1;
    int x2;
} a;
```

若 a 的首地址为 2024 FE00H，a 的成员变量 x2 的机器数为 1234 0000H，则其中 34H 所在存储单元的地址是（　　）

A) 2024 FE03H
B) 2024 FE04H
C) 2024 FE05H
D) 2024 FE06H

[解析] D。

（1）小端模式下低字节对应低地址；

（2）边界对齐即字对准存放。

C 结构体第一个成员存放的地址为结构体变量偏移量为 0 的地址。其他结构体成员自身对齐时，存放的地址为 min{有效对齐值为自身对齐值，指定对齐值} 的最小整数倍的地址

C 语言中如果没有使用“#pragma pack(n)”指定对齐宽度，默认“指定对齐值”规则是按照结构体中最宽的成员所需要的字节数（本题中 x2 是 4 字节类型，即按照 n=4 对齐存放成员）。

地址	变量	内容
2024 FE00H	x1	
2024 FE01H		
2024 FE02H		
2024 FE03H		
2024 FE04H	x2	00H
2024 FE05H		00H
2024 FE06H		34H

2024 FE07H		12H

[补充 2.27] 2016 年 6 月，由国家并行计算机工程技术研究中心研制的“神威·太湖之光”在全球超级计算机 TOP500 榜单上排名第一。其浮点运算速度为 93.0146 PFLOPS，这说明该计算机每秒完成的浮点数操作次数为（　　）

A) 9.3×10^{13} 次

B) 9.3×10^{15} 次

C) 9.3 千万亿次

D) 9.3 亿亿次

[解析] 计算机常用计数单位，实际上这是第 3 章中会讲解的，在第 1 章超级计算机示例中出现过。列为第 2 章的作业有承上启下的意图。

中文单位	中文简称	英文单位	英文简称	十进制（Byte=1）
位	比特	bit	b	0.125
字节	字节	Byte	B	1
开字节	开	KiloByte	KB	10^3
兆字节	兆	MegaByte	MB	10^6
吉字节	吉	GigaByte	GB	10^9
太字节	太	TeraByte	TB	10^{12}
拍字节	拍	PetaByte	PB	10^{15}

[补充 2.28] 下列给出的处理器类型中，理想情况下，CPI(Cycle Per Instruction)为 1 的是（　　）

I. 单周期 CPU（指令周期等于 T 周期）

II. 多周期 CPU（指令周期对应多个 T 周期）

III. 基本流水线 CPU

IV. 超标量流水线 CPU

A) 仅 I、II

B) 仅 I、III

C) 仅 II、IV

D) 仅 III、IV

[解析] B。

教材中并未定义“单周期 CPU”和“多周期 CPU”，但是在其他一些资料里有这些术语。

多周期 CPU 中 CPI>1；超标量因为有多条流水线，CPI<1。

[补充 2.29] 若某计算机最复杂指令的执行需要完成 5 个子功能，分别由功能部件 A~E 实现，各功能部件所需时间分别为 80ps、50ps、50ps、70ps 和 50ps，采用流水线方式执行指令，流水段寄存器延时为 20ps，则 CPU 时钟周期至少为（　　）

A) 60ps

B) 70ps

C) 80ps

D) 100ps

[解析] A。
每个流水段时间应取最大的部件时间 80ps，
加上寄存器延时 20ps，
故 CPU 时钟周期是 100ps。选 D。

[补充 2.30] 下列给出的指令系统特点中，有利于实现指令流水线的是（ ）

I. 指令格式规整且长度一致
II. 指令和数据按边界对齐存放
III. 只有 Load/Store 指令才能对操作数进行存储访问

A) 仅 I、II
B) 仅 I、III
C) 仅 II、III
D) I、II、III

[解析] D。
II，如果不是字对齐，需要 2 个总线周期来取指令或数据
III，非 Load/Store 体系中，以加法指令为例，操作数全部在寄存器需要而时间少于有操作数在存储器的情形，即指令执行所需的 T 周期不同，不利于实现流水线

[补充 2.31] 在采用“取指、译码/取数、执行，访存、写回”5 段流水线的处理器中，执行如下指令序列，其中 s0、s1,s2、s3 和 t2 表示寄存器编号。下列指令对中，不存在数据冒险的是（ ）

I. I1: add s2, s1, s0 //R[s2]←R[s1]+R[s0]
II. I2: load s3, [t2] //R[s3]←Memory[R[t2]]，加载地址为 R[t2]的存储单元
III. I3: add s2, s2, s3 //R[s2]←R[s2]+R[s3]
IV. I4: store s2, [t2] //Memory[R[t2]]←R[s2]，保存至地址为 R[t2]的存储单元

A) I1 和 I3
B) I2 和 I3
C) I2 和 I4
D) I3 和 I4

[解析] C。
数据冒险（数据相关），指后一条指令需要用前一条指令的结果
A 选项，I1 中 s2 是结果，但 I3 的源操作数有 s2
B 选项，I2 中 s3 是结果，I2 的源操作数有 s3
C 选项，I2 和 I4 都操作同一个存储单元，资源冲突
D 选项，I4 中 s2 是结果，I3 中源操作数有 s2

[补充 2.32] 改错题。

（1） 标量处理器中可以利用单条流水线实现整数运算和浮点数运算的并行。
（2） 流水线机制能够有效提高 CPI (Cycles Per Instruction)和 IPC(Instructions Per Cycle)。
（3） 相比 CISC，RISC 处理器完成相同功能需要更多的指令，故采用同一个 C 程序在 RISC 上的执行时间长于 CISC 处理器。
（4） 由于不是标量，向量处理器不能像标量处理器那样使用多条流水线。

[解析]
（1） 整数运算和浮点数运算规则不同，无法用单一的流水线实现并行。
（2） 流水线机制能够有效提高 IPC，理想情况下 CPI=1。
（3） 相比 CISC，RISC 处理器完成相同功能需要更多的机器指令，但 C 程序经编译后的机器代码执行效率可能更高。
（4） 向量处理器也可以设计多条流水线。

CH3

3.20 某计算机按字节编址，其主存容量为 1MB，Cache 容量为 16KB，Cache 和主存之间交换的块大小为 64B，采用直接相联映射方式。
（1）Cache 共有多少个字块（Cache line）？
（2）主存地址为 02021H 的单元装入 Cache 后对应的 Cache 地址是?
（3）主存地址为 02021H 的单元装入 Cache 后存放在 Cache 中的第几字块中（Cache 起始字块为第 0 字块）？

[解析]
(1) 字块数=Cache 容量/块大小=$2^{14}/2^6=2^8$
(2) 直接相联映射方式的主存地址由三部分组成：页号T，块号C，块内地址W
其中，T=主存容量/Cache 容量=$2^{20}/2^{14}=2^6$，故T 有6 位； 由(1)知，C 有8 位；
因为块大小=64B=2^6，故W 有6 位。 由此可对主存地址02021H 进行分析:
000000100000000100001(T:蓝色，C:绿色，W:红色)
由直接相联映射，该单元装入cache 的第128(10000000)块，块内地址为100001。 因此得cache 地址10000000100001，即2021H
(3)由(2)知该单元装入cache 的第128(10000000)块。

3.21 某计算机按字节编址，其主存容量为 1MB，Cache 容量为 16KB，Cache 和主存之间交换的块大小为 64B，采用 8 路组相联映射方式。

（1）主存地址中页号 s、页内块号 u、块内地址 W 各占多少位?

（2）主存地址为 02021H 的单元装入 Cache 后存放在 Cache 中的第几组（起始组为第 0 组）?

（3）Cache line 对应的 Tag 字段占用多少位?

[解析]

(1)组相联映射方式的主存地址由三部分组成：页号s，块号u，块内地址W

页内块号u=cache 中组号=cache 容量/(组中数据块数目*数据块大小)=$2^{14}/2^6*2^3=2^5$，u 有5位。 因为块大小=64B=2^6，故W 有6 位。

页号s 位数=20-6-5=9.

(2)对主存地址02021H 进行分析:

00000010000000100001(s:蓝色，u:绿色，W:红色)

故存放在第0(00000)组

(3)Tag 字段位数=页数位数=9

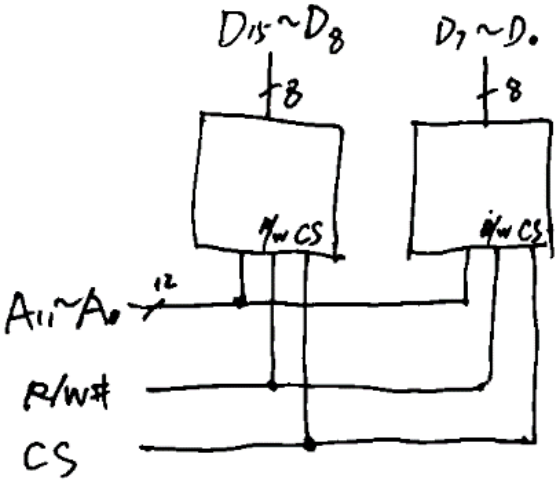
3.22 某按字节编址的计算机系统，使用了 40 位地址线，16 位数据线，请问该计算机存储器空间的最大寻址范围是什么?

[解析]

0x00 0000 0000 ~ 0xFF FFFF FFFF

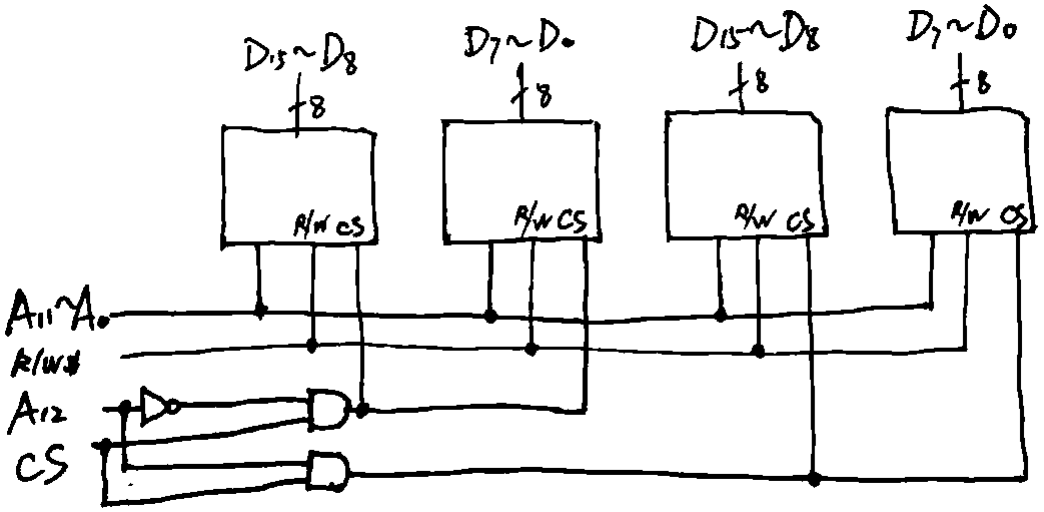
3.24 试用 4K×8 位的芯片构成 4K×16 位的存储器。

[解析]



3.27 设计一个用 4 片 4K×8 位的芯片构成 8K×16 位的存储器。

[解析] 答案不唯一，合理即可



[补充 3.31] 以下关于存储器原理描述正确的是（　　）

I. 硬盘、软盘等磁介质存储器往往依靠磁颗粒的磁化方向来代表了“0”和“1”

II. HDD 容量计算公式为：容量=盘片数×磁头数 × 磁道(柱面)数 × 每道扇区数 × 每扇区字节数

III. CD-DA、DVD 等光盘上有凹陷和凸起，光照射到凹陷和凸起区域后反射强度不同，代表了“0”和“1”

IV. 因为晶体管只有开和关 2 种状态，故而 1 个晶体管只能保存 1 比特信息

A) I、II

B) II、III

C) III、IV

D) A~C 均不符合题意

[解析] C

III 错误，激光照射到平坦区和凹陷/凸起变化区反射强度不同。

IV 错误，一个晶体管也可以保存多个比特

[补充 3.32] 以下关于典型 ROM 说法正确的是（　　）

V. Mask ROM（掩模 ROM）中的数据用户不能修改

VI. ROM 芯片中字线用来选中存储单元，而位线用于数据的输入和输出

VII. PROM (Programmable ROM)可以利用外围电路实现数据的多次写入

VIII. EPROM (Erasable PROM)可以利用外围电路实现数据的多次修改

IX. EEPROM (Electrically EPROM)可以利用外围电路实现数据的多次修改

X. Flash 和 E^2PROM 都基于浮置栅中是否存有电荷来实现“0”或“1”的逻辑状态存储

XI. NAND Flash 比 NOR Flash 存储密度更高是因为少一个选通管

A) I、III

B) II、III

C) III、IV

D) V、VI

[解析] D

III 错误，PROM (Programmable ROM)只能一次写入

IV 错误，EPROM (Erasable PROM)擦除需要使用紫外线照射

VII 错误，NAND Flash 比 NOR Flash 存储密度更高主要是因为位线采用串联而非并联的方式

[补充 3.33] 以下关于计算机存储器说法正确的是（ ）

I. 从物理机制上看，任何一种存储器都要能表示“0”和“1”两种状态
II. 从物理机制上看，无论是磁存储、光存储还是半导体存储，都只能表示表示“0”和“1”两种状态
III. 硬磁盘（HDD）的容量≈盘片数×磁头数 × 磁道(柱面)数 × 每道扇区数 × 每扇区字节数
IV. 光盘的容量往往受限于激光头组件的波长
V. 半导体存储器往往受限于芯片面积而难以实现超大容量
VI. 半导体存储与磁存储、光存储相比，具有体积小、重量轻的优点
VII. 半导体存储被广泛用于芯片内存储、计算机主存和外存

A) I、II、III、IV

B) II、III、IV、V

C) I、II、III、V

D) IV、V、VI、VII

[解析] D

II 错误，磁存储、光存储或者半导体存储都有可能存储多个状态，只是存储 2 个状态技术实现较为容易。

[补充 3.36] 某按字节编址的计算机系统，有 14 根地址线，8 根数据线，用 4 片 4K×8 位的 SRAM 芯片构成 16K×8 位的存储器。请①画出 4 片 SRAM 芯片与 CPU 的连接图；并②列出 4 片 SRAM 芯片的地址范围。

[解析]

等于 CPU 而言，可以产生 0~16K 的地址范围。

4 片 SRAM 芯片的数据线与 CPU 数据线连接（并联）；低 12 根地址线与 CPU 地址线低 12 位连接（并联）。

芯片 1 地址线 A[13]、A[12]均经非门与 CPU 的 A[13]、A[12]连接

芯片 2 地址线 A[13]经非门与 CPU 的 A[13]连接，芯片 2 的 A[12]直接连接 CPU 的 A[12]

芯片 3 地址线 A[13]直接连接 CPU 的 A[13]，芯片 3 的 A[12]经非门连接 CPU 的 A[12]

芯片 4 高 2 位地址线 A[13]、A[12]均直接连接 CPU 的 A[13]、A[12]

芯片 1 地址范围：0x0000 ~ 0x0FFF

芯片 2 地址范围：0x1000 ~ 0x1FFF

芯片 3 地址范围：0x2000 ~ 0x2FFF

芯片 4 地址范围：0x3000 ~ 0x3FFF

[补充 3.37] 以下关于 Cache 说法正确的是（　）

A) 贯穿读出（Look Through）方式下，CPU 得到的所有数据都是从 Cache 中读出的

B) 旁路读出（Look Aside）方式下，CPU 得到的所有数据都是从 Cache 中读出的

C) 写通方式（Write Through）方式下，CPU 写入存储器的数据可能不经过 Cache

D) 写回方式（Write Back）方式下，CPU 写入存储器的数据可能不经过 Cache

[解析] C

A、B，如果数据不在 Cache 中，CPU 访问主存，同时将数据（所在的主存数据块）装入 Cache，并设置 Cache 块有效位字段。

D 写回方式（Write Back）方式下，更新数据只写到 Cache

[补充 3.38] 假定主存地址为 32 位，按字节编址，指令 Cache 和数据 Cache 与主存之间均采用 8 路组相联映射方式、直写(Write Through)写策略和 LRU 替换算法，主存块大小为 64B，指令 Cache 和数据 Cache 容量均为 32KB，开始时 Cache 均为空。请回答下列向题。

（1）Cache 每一行中标记(Tag)、LRU 位各占几位？是否有修改位？

（2）有如下 C 语言程序段：

```
for(k=0; k<1024; k++)  s[k]=2*s[k];
```

若数组 s 及变量 k 均为 int 型，int 型数据占 4B，变量 k 分配在寄存器中，数组 s 在主存中的起始地址为 0080 00C0H，则该程序段执行过程中，访问数组 s 的数据 Cache 缺失次数为多少？

[解析]

(1)主存块大小为 64B=2^6字节，故主存地址低 6 位为块内地址，Cache 组数为 32KB/(64B×8)=64=2^6，故主存地址中间 6 位为 Cache 组号，主存地址中高 32-6-6=20 位为标记，采用 8 路组相联映射，故每行中 LRU 位占 3 位，采用直写方式，故没有修改位。

(2)因为数组 s 的起始地址最后 6 位全为 0，故 s 位于一个主存块开始处，占 1024×4B/64B=64 个主存块：执行程序段过程中，每个主存块中的 64B/4B=16 个数组元素依次读、写 1 次，因而对于每个主存块，总是第一次访问缺失，以后每次命中。综上，数组 s 的数据 Cache 访问缺失次数为 64 次。

【拓展思考】对于追求高性能的程序，程序中所需处理的数据如果能全部载入 Cache，代码运行效率将有极大的提升。

[补充 3.39] 某计算机主存地址为 24 位，采用分页虚拟存储管理方式，虚存空间大小为 4GB，页大小为 4KB，按字节编址。某进程的页表部分内容如下表所示。

虚页号	实页号（页框号）	存在位
82	024H	0
…	…	…

129	180H	1
130	018H	1

当 CPU 访问虚拟地址 0008 2840H 时，虚实地址转换的结果是。（　　）

A) 得到主存地址 02 4840H
B) 得到主存地址 18 0840H
C) 得到主存地址 01 8840H
D) 检测到缺页异常

[解析] C

页大小为 4KB → 地址地 12 位是页内偏移，即 840H；

虚页号是高 12 位，082H → 130D；

虚页号 130D 在页表中存在 → 查表得到 018H → 实地址是 018840H。

此题有个细节易被忽略，页表中虚页号是十进制表示，实页号是 16 进制表示。

CH4

4.15　周期分裂式总线操作时序有哪些特点？适用于什么样的场景？

[解析]

特点：将一个总线读周期或写周期分解成两个分离的子周期：寻址子周期，数据传送子周期。

适用场景：多主模块系统，从模块不能随时及时响应。

4.21　简述 AHB 总线的流水线机制。

[解析]

本次数据传输操作的地址阶段和上次操作的数据阶段重复。

[修改版] 4.22　定性分析 AHB 中 SPLIT 操作的优点。

[解析]

从机不能响应时，可以把总线的使用权让给其他主机。这样做支持多个主机同时访问 AHB 总线，降低总线冲突的发生。

4.35　异步串行通信中的起始位和停止位有什么作用？

[解析]

标识传输的开始和结束。

4.58　异步串行通信系统中，采样数据时为什么要在数据位的中间？

[解析]

提高采样的准确率，避免因为干扰而误采样，靠近传送波形的上升沿或者下降沿都有可能采样到相邻位的信号。

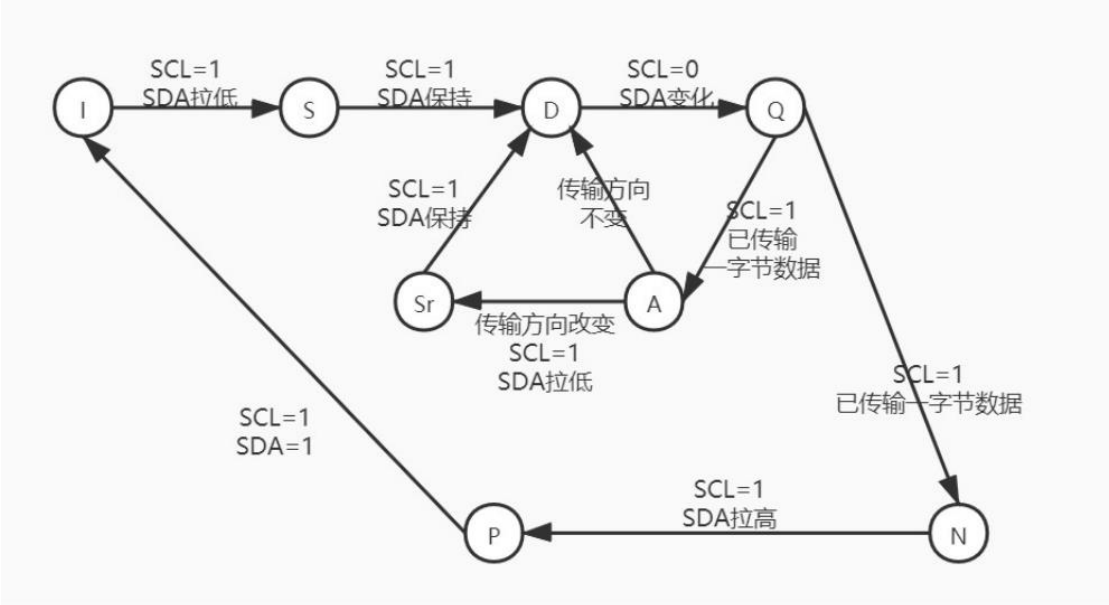
4.61　描述 I^2C 总线协议中的状态，并画出状态转移图。

[解析]

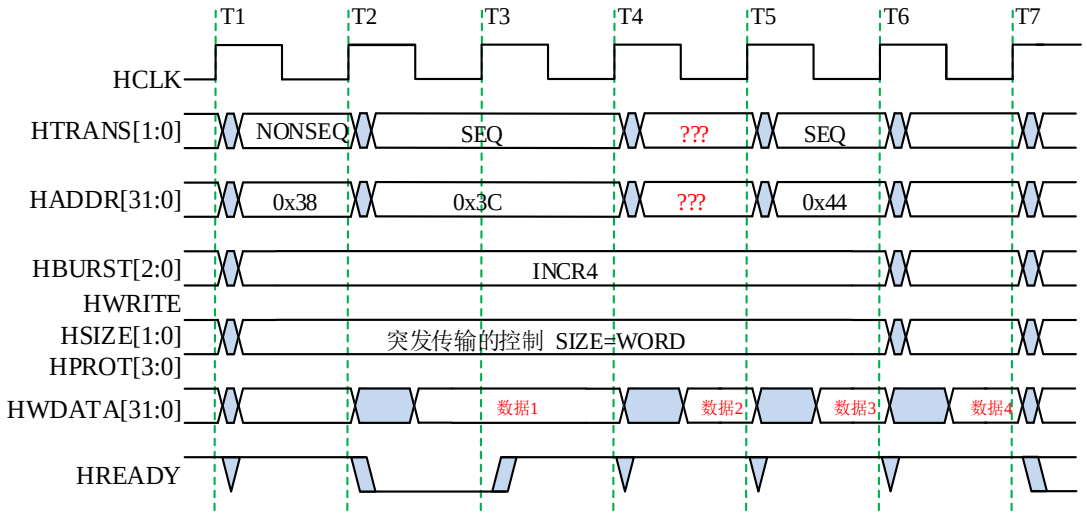
总线空闲 I：SDA 与 SCL 均为高电平

启动数据传输 S：SCL 为高电平，SDA 负跳变

停止数据传输 P：SCL 为高电平，SDA 正跳变
重新启动 Sr：SCL 为高电平，SDA 负跳变
数据有效 D：SCL 为高电平期间，SDA 电平稳定
等待/ 数据无效 Q：SCL 保持低电平，此时可修改线上数据
应答 A /不应答 N：接收方将 SDA 拉低发出 ACK 或释放 SDA 发出 NACK



[补充 4.62] 下图中 HTRANS[1:0]信号“???”应该为什么取值？HADDR[31:0]信号“???”应该为什么取值？HWDATA[31:0]中数据 1、数据 2、数据 3、数据 4 对应的地址分别是什么？



[解析]

- 单次传输，每次传输单个数据
- HTRANS[1:0]信号“???”是 SEQ
- HADDR[31:0]信号“???”是 0x40
- HWDATA[31:0]中数据 1 数据 2、数据 3、数据 4 对应的地址分别是：0x38、0x3C、0x40、0x44

[补充 4.63] QPI 总线是一种点对点全工同步串行总线，总线上的设备可同时接收和发送信息，每个方向可同时传输 20 位信息（16 位数据+4 位校验位），每个 QPI 数据包有 80 位信息，分 2 个时钟周期传送，每个时钟周期传递 2 次，因此，QPI 总线带宽为：每秒传送次数×2B×2.若 QPI 时钟频率为 2.4GHz,则总线带宽为（　）

A) 4.8 GB/s

B) 9.6 GB/s

C) 19.2 GB/s

D) 38.4 GB/s

[解析] C

每个时钟周期传递 2 次→每秒传输次数 2.4GHz ×2 = 4.8G

根据题意，带宽=每秒传送次数×2B×2=4.8G×2B×2=19.2 GB/s

[补充 4.64] 对比“I^2C 总线规范中定义的地址”、“访问存储器的地址”、“Cache 地址”、“I/O 端口地址”有何不同？

[解析] 大致描述出以下信息即可，

- “I^2C 总线规范中定义的地址”是 I^2C 器件地址，即器件的 ID，用于指示被访问的从器件。
- “访问存储器的地址”是存储器内存储单元的 ID，用于指示拟访问的存储单元。
- “Cache 地址”是 Cache 内存储单元的 ID。
- “I/O 端口地址”是 I/O 接口中（控制、数据、状态）寄存器的 ID，用于指示被访问的寄存器。

[补充 4.65] 以下关于“I/O 端口”说法正确的是（　）

XII. 一个 I/O 接口只能有一个 I/O 端口

XIII. I/O 端口本质上是具备存储能力的电路单元

XIV. 一般的 I/O 接口电路中常包括控制端口、数据端口、状态端口

XV. I/O 端口的地址可以和主存中存储单元地址统一编址

XVI. I/O 端口的地址必须位于独立于主存的地址空间

XVII. I/O 端口统一编址情形，I/O 端口的地址必须位于高位地址段

E) 仅 I、II、III

F) 仅 II、III、IV

G) 仅 III、IV、V

H) 仅 IV、V、VI

I) 以上答案均不正确

[解析] B。

I，错误，计算机系统中可能有多个 I/O 接口，每个 I/O 接口可以有多个 I/O 端口。

V，错误， I/O 端口统一编址时，I/O 端口地址和内存存储单元在同一个地址空间

VI，错误，I/O 端口地址取决于地址译码电路的设计，可以在任意地址段

CH5

5.2 请简述哈佛结构的主要优缺点。

[解析]

μ 优点： 取指和数据存取可以通过两套独立的总线同时进行， 从而减小指令流水线发生资

源冲突的概率

μ 缺点： 哈佛结构较为复杂， 与外设以及扩展存储器的连接难度较大

5.10 Cortex-M3 与 Cortex-M4 使用两个堆栈的目的是什么？ 在中断响应时， 程序断点和程序

状态寄存器的内容保存在哪个堆栈中？

[解析]

μ 使用两个堆栈是为了服务于不同的操作模式和特权访问等级， 处理模式总是使用 MSP，

线程模式可以使用 MSP 或 PSP。

μ 程序断点和程序状态寄存器的内容保存在 MSP 中。

5.17 在 Cortex-M3/M4 中， 寄存器 R0~R12 有何异同？ 如果这些寄存器都是空闲的， 你觉得

首先使用哪些？ 为什么？

[解析]

μ 相同点： 都是通用寄存器

μ 不同点： 为了能够实现汇编程序与 C 语言程序的相互调用， ARM 公司制定了 AAPCS

（ARM Architecture Procedure Call Standard） 规范： R0~R3 用于子程序之间的参数传递；

R4~R11 用于保存子程序的局部变量； R12 作为子程序调用中间寄存器。

μ 优先使用低位寄存器

ρ R0~R7 低位寄存器(许多 16 位的 thumb 指令只能访问低位寄存器)

ρ R8~R12 高位寄存器(可用于 32 位指令和少数几个 16 位指令)

5.22 某基于 Cortex-M4 的 SOC 芯片共有 64 级外部中断， BASEPRI 寄存器的宽度共有几位？

如果想屏蔽所有优先级大于 16 的中断， 请写出对 BASEPRI 寄存器进行设置的汇编指令。

如果想屏蔽所有优先级大于 0 的中断， 又该如何设置？

[解析]

由题知芯片配置了 $64=2_6$ 级外部中断， 故 BASEPRI:7:2 共 6 位屏蔽所有优先级大于 16 的中断：

```
MOV R0, #0x0000 0044 或者 MOV R0, #0b 010001(00) MSR BASEPRI, R0
```

屏蔽所有优先级大于 0 的中断：

MOV R0,#0x 0000 0004 或者 MOV R0, #0b 000001(00) MSR BASEPRI,R0

5.29 Cortex-M3 存储空间的哪些区域支持位段（ bit-band） 操作？
[解析]
SRAM 区域的最低 1MB（ 0x20000000 ~ 0x200FFFFF)外设区域的最低 1MB（ 0x40000000 ~ 0x400FFFFF）

5.31 写出利用位段操作读取 0x4000 1000 的第 3 位的代码。
[解析] 此处第 3 位的理解可能有歧义， 一种是理解为 bit[2]、 一种是理解为 bit[3]， 以下按照 bit[3]给出参考代码
汇编实现：
LDR R0， =0x4202 000C LDR R1， [R0]
C 实现（volatile 关键字非必须）：

```
int x;
x = *(( volatile unsigned long*) (0x4202 000C));
```

5.36 处理器进入异常处理子程序之前保护现场需要把哪些寄存器的值保护起来？
[解析] PSR， PC， LR， R0~R3， R12。

5.39 解释向量表重定位机制。
[解析]
向量表重定位使用 VTOR 指示向量表的位置， 保存向量表相对于存储器的偏移量，处理器通过修改 VTOR 值修改向量表的起始位置， 从而实现向量表重定位。

[补充 5.47] 以下关于 Cortex-M3/M4 操作状态与操作模式法正确的是（ ）

I. 特权线程模式下可以通过置位 CONTROL 寄存器 nPRIV 位进入非特权模式
II. 非特权线程模式下可以通过置零 CONTROL 寄存器 nPRIV 位进入特权模式
III. 处理模式下可以通过置零 CONTROL 寄存器 nPRIV 位进入特权线程模式
IV. 非特权模式下访问 CONTROL 寄存器会触发 Usage Fault 异常
V. 非特权线程模式下访问存储器可能会触发 MemManage Fault 异常
VI. 处理模式下可以通过置零 CONTROL 寄存器 nPRIV 位进入特权线程模式
VII. 处理模式下只能使用 MSP 而不能使用 PSP
VIII. 线程模式下只能使用 PSP 而不能使用 MSP

A) I、II、III、IV、V

B) II、III、IV、V、VI
C) III、IV、V、VI、VII
D) IV、V、VI、VII、VIII
E) 以上选项均不正确

[解析] C
I. 特权线程模式下可以通过置位 CONTROL 寄存器 nPRIV 位进入非特权模式
II. 非特权线程模式下不能访问 CONTROL 寄存器
III. 处理模式下可以通过置零 CONTROL 寄存器 nPRIV 位进入特权线程模式
IV. 非特权模式下访问 CONTROL 寄存器会触发 Usage Fault 异常
V. 非特权线程模式下访问存储器可能会触发 MemManage Fault 异常
VI. 处理模式下可以通过置零 CONTROL 寄存器 nPRIV 位进入特权线程模式
VII. 处理模式下只能使用 MSP 而不能使用 PSP
VIII. 线程模式下既可以使用 PSP 也可以使用 MSP

[补充 5.53] 以下关于 Cortex-M3/M4 堆栈说法正确的是（ ）
I. 只支持满递减类型的栈
II. 往栈内放置一个新的数据后，栈指针数值会增加
III. 具有双栈的结构，有 MSP 和 PSP 两个堆栈指针
IV. 非特权线程模式只能使用 PSP 而不能使用 MSP
V. MSP 的值在上电前就需要存储在片内的 Flash 中
VI. PSP 的值需要在初始化程序中进行设置
VII. 栈空间只能放置在 4GB 空间的 SRAM 区
VIII. 按照 AAPCS 要求，栈需要双字对齐
A) I、II、III、IV
B) II、III、IV、V
C) III、IV、V、VI
D) IV、V、VI、VII
E) V、VI、VII、VIII
F) 以上选项均不正确

[解析] C
I. 只支持满递减类型的栈
II. 往栈内放置一个新的数据后，栈指针数值会递减
III. 具有双栈的结构，有 MSP 和 PSP 两个堆栈指针
IV. 非特权线程模式只能使用 PSP 而不能使用 MSP
V. MSP 的值在上电前就需要存储在片内的 Flash 中
VI. PSP 的值需要在初始化程序中进行设置
VII. 栈空间可以放置在 4GB 空间的 SRAM 区的 SRAM 或者 CODE 区的 SRAM 中
VIII. 按照 AAPCS 要求，栈需要双字对齐

[补充 5.54] 以下关于 Cortex-M3/M4 存储器子系统说法正确的是（　　）

I. 4GB 的存储器映射关系是固定不变的
II. 既可以配置为大端模式，也可以配置为小端模式
III. 所有 Cortex-M3/M4 的存储器访问指令都支持非对齐的数据传输
IV. 4GB 空间所有区域均可以采用位段（Bit-Band）操作方式来访问
V. 位段操作能保障“读→修改→写”的流程不被其他操作打断
VI. 处理器读存储器时，存储器系统会检查所访问区域的存储器访问属性
VII. 非特权线程访问内核私有区域可能会触发 MemManage Fault 异常
VIII. 处理器不会改变代码的执行顺序，因而不需要存储器屏障指令

A) I、II、III、IV
B) II、III、IV、V
C) III、IV、V、VI
D) IV、V、VI、VII
E) V、VI、VII、VIII
F) A~E 均不满足题意

[解析] F

I. 4GB 的存储器映射关系是固定不变的
II. 既可以配置为大端模式，也可以配置为小端模式
III. 并非所有 Cortex-M3/M4 的存储器访问指令都支持非对齐的数据传输
IV. 4GB 空间中特定的区域可以采用位段（Bit-Band）操作方式来访问
V. 位段操作能保障“读→修改→写”的流程不被其他操作打断
VI. 处理器读存储器时，存储器系统会检查所访问区域的存储器访问属性
VII. 非特权线程访问内核私有区域可能会触发 MemManage Fault 异常
VIII. 处理器不会改变代码的执行顺序，但指令中的存储器访问操作完成的时间不一定和指令的顺序一致，仍可能用到存储器屏障指令

[补充 5.55] 以下关于 Cortex-M3/M4 中断/异常机制说法正确的是（　　）

I. 特殊寄存器 PRIMASK 和 FAULTMASK 均用于中断屏蔽
II. 与 PRIMASK 不同的是，FAULTMASK 无需清理，当负责错误处理的异常处理程序返回时，会自动复位 FAULTMASK
III. 特殊寄存器 BASEPRI 采用了可伸缩的设计
IV. 每个异常都会处于激活、非激活、挂起、激活并挂起之一的某一个状态
V. 每个异常都有一个独立的优先级寄存器
VI. 异常的优先级寄存器只能由特权访问等级代码访问
VII. 中断优先级寄存器中数值约小，表示对应中断源的优先级约高
VIII. 多个相同优先级的异常同时发生时，处理器会先处理异常类型号低的异常
IX. 中断使能、中断禁止采用了 2 个独立的寄存器
X. 特定中断源的中断使能或禁止用寄存器中的 1 位来实现

A) I、II、III、IV、V
B) II、III、IV、V、VI
C) III、IV、V、VI、VII
D) IV、V、VI、VII、VIII
E) V、VI、VII、VIII、IX
F) VI、VII、VIII、IX、X
G) A~F 均不满足题意

[解析] G
只有 X 选项不正确