第一章

第二章

Q2-1：

高序：最低位字节存储在字的最低位；

低序：最低位字节存储在字的最高位；

Q2-2：

冯.诺依曼结构：数据和指令都存储在同一存储器中；

哈佛结构：数据和程序存储在各自独立的存储器中。

Q2-3：

a) ARM 有 16 个通用寄存器，r0-r15，其中 r15 还被用作程序计数器
b) CPSR 是程序状态寄存器，包含了条件码标识、中断禁止位、当前处理器模式和其他状态\控制信息，其中条件码标识在进行算术、逻辑或移位运算的过程中被设置。
c) 如果运算结果的每一位都为 0，则 CPSR 的 Z 位置 1
d) 程序计数器保存在 r15 寄存器

Q2-4；

a) 2-3：0x00000002-0x00000003=0xffffffff， NZCV=1000;
b) 题目改成$-2^{31}+1-1$：0x80000001-0x00000001=0x80000000，NZCV=1010
c) -4+5：0xfffffffc+0x00000005=0x00000001，N=0010

Q2-5：

a) EQ：等于零
b) NE：不等于零
c) MI：负数
d) VS：溢出
e) GE：有符号大于或等于
f) LT：有符号小于

Q2-6：

BL 指令引导处理器转移到子程序处开始执行，在子程序跳转之前，会把下一条指令的地址存储到 R14（LR）中，然后将目标地址存储到 R15 中。

Q2-7：

MOV r15, r14

Q2-8： 不作要求。

Q2-9：

NEON 指令集是一组适用于 ARM 处理器的单指令多数据（SIMD）指令，用于支持向量和多媒体处理 。

Jazelle 指令集允许直接执行 8 位 Java 字节码，因此执行 Java 程序时不需要使用字节码解释器。

第三章：

Q3-1：内存映射 I/O 的优点如下：

a) 省略了 I/O 操作的复杂逻辑，易实现，耗费低；
b) 可以利用丰富的内存寻址模式实现灵活的 I/O 操作。

Q3-2：

忙等 I/O 效率低，当 I/O 操作未完成时，CPU 除了反复测试设备状态什么都不能做。而 CPU 需要很多操作与 I/O 事务并行执行。

Q3-3：

假设存储单元 ds1 处寄存器地址为 0x2000，代码如下

```
#define ds1 0x2000

while(*ds1 == 0);
```

Q3-4：

假设设备（dev1）中有两个寄存器 ds1 和 dd1，dev1 的地址为 0x1000，ds1 的偏移量（offset）为 0，则 ds1 地址为 0x1000，同理 dd1 的地址为 0x1004，代码如下

```
#define based_addr 0x1000
#define ds1 (based_addr + 0)
#define dd1 (based_addr + 4)
int data_dd1 ;
while ((peek (ds1) & 0x0001) == 0) ;
        data_dd1 = peek (dd1) ;
```

Q3-9：

CPU 没有需要并行的操作事务，仅有一项 I/O 事务。

Q3-10：

选择中断优先级。中断优先级能容许多台处理设备连接到中断线路上，并且允许 CPU 在处理重要请求时忽略不重要的中断请求。而中断向量只是提供了灵活性，使中断设备可以指定为它服务的中断服务程序。

Q3-21：

0 做除数、未定义的 Resets 指令和非法的内存访问

Q3-22：

陷阱，也就是软件中断，是一种显式产生异常状态的指令，最通用的用法是进入管态。

Q3-24：

a) 强制性未命中 compulsory miss: 发生在单元第一次被访问时；
b) 容量未命中 capacity miss: 工作集大于高速缓存容量；
c) 冲突未命中 conflict miss: 两个地址映射到高速缓存的同一个单元。

Q3-25：

$t_{av} = ht_{cache} + (1-h)t_{main}$　$t_{av} = 6.08$ ns

Q3-26：

$t_{av} = ht_{cache} + (1-h)t_{main}$　$h = 98\%$

Q3-27：

将地址的最后一位作为组的索引（index）

存取 001 后

组	块 0 标记	块 0 数据	块 1 标记	块 1 数据
0	-	-	-	-
1	00	1111	-	-

存取 010 后

组	块 0 标记	块 0 数据	块 1 标记	块 1 数据

0	01	0000	-	-
1	00	1111	-	-

存取 011 后

组	块 0 标记	块 0 数据	块 1 标记	块 1 数据
0	01	0000	-	-
1	00	1111	01	0110

存取 100 后

组	块 0 标记	块 0 数据	块 1 标记	块 1 数据
0	01	0000	10	1000
1	00	1111	01	0110

存取 101 后

组	块 0 标记	块 0 数据	块 1 标记	块 1 数据
0	01	0000	10	1000
1	10	0001	01	0110

存取 111 后

组	块 0 标记	块 0 数据	块 1 标记	块 1 数据
0	01	0000	10	1000
1	10	0001	11	0100

Q3-28：

a．每条指令执行 1 次，到 B loop，高速缓存的状态

块	标记	指令
0	110	B loop
1	101	ADD r0, r0, #1

循环执行完，高速缓存的状态

块	标记	指令
0	011	BEG loopend
1	010	CMP r0, r1

b．每条指令执行 1 次，到 B loop，高速缓存的状态

块	标记	指令
00	11	B loop
01	10	MUL r4, r4, r6
10	10	ADD r2, r2, r4
11	10	ADD r0, r0, #1

循环执行完，高速缓存的状态

块	标记	指令
00	11	B loop
01	01	CMP r0, r1
10	01	BEG loopend
11	10	ADD r0, r0, #1

c.（采用 LRU 替换原则）

每条指令执行 1 次，到 B loop，高速缓存的状态

组	块 0 标记	块 0 指令	块 1 标记	块 1 指令
0	110	B loop	101	ADD r2, r2, r4
1	100	MUL r4, r4, r6	101	ADD r0, r0, #1

循环执行完，高速缓存的状态

组	块 0 标记	块 0 指令	块 1 标记	块 1 指令
0	100	B loop	011	BGE loopend
1	010	CMP r0, r1	011	ADD r0, r0, #1

Q3-31：取指、译码、执行

Q3-33：

指令延迟：指令从开始执行到结束的时间

指令吞吐量：单位时间执行的指令数

Q3-36：电压降、切换、泄漏

Q3-37：a．节电模式

b．当指令运行时，CPU 会关掉部分不需要运行的指令，从而减少功耗

第四章

Q4-1：微处理器、I/O 设备和存储器组建是计算平台的三大组件。

Q4-2：HAL 是硬件抽象层，提供硬件的基本层次抽象。

Q4-4：

a) R/W：当总线读时为 1；当总线写时为 0。
b) Data ready：当数据束上的值有效时发信号。
c) Clock：提供总线组件的同步。

Q4-24：

存储器控制器是 CPU 和存储器间的接口，控制器管理 CPU 对所有存储器组件的访问，进行调度。

Q4-28：

Tcpu <= 1/44.1khz = 2.3 x 10-5 s

20MHZ = 5 x 10-8 s 执行指令条数 n = 2.3 x 10-5 / 5 x 10-8 - 100 =360

Q4-29：

如果下次发生的中断优先级低于或等于前次，则正在执行的中断服务子程序继续执行；若下次发生的中断优先级高于前次，则正在执行的中断服务子程序被打断，中断服务程序优先处理后发生中断之后，然后再处理前次中断。

Q4-36/Q4-37：

题意不清楚，不再作要求。