

第三章习题（p65） 1, 2。

1. 请写出对应 C 代码的 ARM 指令

If(a>b)

a++;

else

b++;

答：

ADR R0, a;

LDR R1, [R0];

ADR R2, b;

LDR R3, [R2];

CMP R1, R3;

ADDGT R1, R1, #1;

ADDLE R3, R3, #1;

STRGT R1, [R0];

STRLE R3, [R2];

2. 请写出下列 ARM 指令的功能

答：参考教材第三章。

第四章：请简述 S5PV210 处理器 GPIO 端口操作步骤？

答：S5PV210 处理器 GPIO 端口操作步骤如下：

首先，确定所使用的 GPIO 端口的功能，如作为输入/输出引脚使用时，是否需要设置上拉/下拉电阻；作为其它功能使用时，对应 S5PV210 处理器的芯片手册进行设置。

其次，确定 GPIO 端口的输入/输出方向，通过端口设置寄存器完成端口的输入/输出功能或其它功能设置。

最后，对数据寄存器操作。如果设置为输入引脚，读取数据寄存器对应位值，实现引脚状态的读取；如果设置为输出引脚，通过写数据寄存器对应位值，实现引脚状态的设置。

补充：

1. 什么是寻址方式？请写出 ARM 指令系统 6 种以上寻址方式？

答：

所谓寻址方式，就是根据指令中操作数的信息寻找操作数实际物理地址的方式。

- a) 立即数寻址
- b) 寄存器寻址
- c) 寄存器移位寻址
- d) 寄存器间接寻址
- e) 基址变址寻址
- f) 相对寻址

2. 假设 R0 的内容为 0x8000，寄存器 R1、R2 的内容分别为 0x01 与 0x10，存

存储器内容为 0。连续执行下述指令后，说明每条指令执行后 PC 如何变化？存储器以及寄存器内容怎么变化？

STMIB R0!,{R1,R2}

LDMIA R0!,{R1,R2}

答：执行第一条指令后，存储器中地址为 0x8004 保存的内容为 R1 的内容，即 0x01，存储器中地址为 0x8008 保存的内容为 R2 的内容，即 0x10，寄存器 R1,R2 的内容不变，仍分别为 0x01 与 0x10。R0 的内容为 0x8008。PC=PC+4。

执行第二条指令后，存储器内容不变，寄存器 R1 保存的是存储器地址为 0x8008 的内容：0x10,R2 保存的是存储器的地址为 0x800c 的内容，即 0。R0 的内容为 0x8010。PC=PC+4

2. 写出下述内存变量 x,y,z 的地址。

begin equ 0x20000

map begin

x field 4

y filed 8

z filed 16

答：x 长度为 4 字节，位置在：0x20000

y 长度为 8 字节，位置在：0x20004

z 长度为 16 字节，位置在：0x2000c

3. 写出执行以下计算的指令序列，其中 X,Y,Z,R,W 均为 32 位无符号数，两数乘积不超过 32 位数范围。

(1) $Z \leftarrow W - (X+6) - (R+9)$

(2) $Z \leftarrow (W*X)/16$

答：(1) ADR R4, W; 变量 W 的地址装入 R4 中;

LDR R0, [R4] ;(W)->R0

ADR R4, X; LDR R4,=X

LDR R1, [R4] ;(X)->R1

ADD R1, R1,#6;

SUB R0, R0,R1;

ADR R4, R;

LDR R1, [R4] ;(R)->R1

ADD R1, R1,#9;

SUB R0, R0,R1;

ADR R4, Z;

STR R0, [R4];

另一版本答案：

LDR R4,=X

LDR R0,[R4]

ADD R1,R0,#6

LDR R4,=R

LDR R0,[R4]

ADD R2,R0,#9

```

LDR R4,=W
LDR R0,[R4]
SUB R3,R0,R1
SUB R3,R3,R2
LDR R4,=Z
STR R3,[R4]

```

```

(2) LDR R4,=W
    LDR R0,[R4]
    LDR R4,=X
    LDR R1,[R4]
    MUL R3,R0,R1
    MOV R3,R3,LSR #4
    LDR R4,=Z
    STR R3,[R4]

```

5. 有一字符串 “**Hello World!\n**”，编写汇编语言程序，将其所有的小写字母转换为大写字母，并输出。

答：

```

.data
str:
    .asciz "Hello World !\n"

.text
.globl main
main:
    stmfd sp!, {r1,r2,lr}
    ldr r1, =str
    ldrb r2, [r1]
    b    test
loop:
    cmp r2, #97      @#'a'
    blt next
    cmp r2, #122     @#'z'
    bgt next
    sub r2, r2, #32
    strb r2, [r1]
next:
    ldrb r2, [r1, #1]!
test:
    cmp r2, #0
    bne loop
    ldr r0, =str

```

```

bl printf
mov r0, #0
ldmfd sp!, {r1,r2, pc}
.end

```

6. 先使用 ARM 汇编语言指令编写一个实现排序功能的程序段，再编写一个调用该程序段实现数组排序的 C 语言程序，输出排序前和排序后数组元素。

答：

bubble.s:

```

.text
.globl bubble
bubble:
    mov r7,#25

```

```

loop0:
    mov r2,r0
    mov r6,r7

```

```

loop1:
    ldrb r3,[r2]
    add r4,r2,#1
    ldrb r5,[r4]
    cmp r3,r5
    stcbsb r3,[r4]
    stcbsb r5,[r2]
    add r2,r2,#1
    sub r6,r6,#1
    cmp r6,#1
    bne loop1
    sub r7,r7,#1
    cmp r7,#1
    bne loop0

```

```

    mov pc,lr
.end

```

csort.c:

```

#include<stdio.h>
#include<stdlib.h>
#include<time.h>
extern char bubble(char *num);
char num[25];
void output()

```

```
{
    int i;
    for(i=0;i<25;i++)
    {
        printf("%d ",num[i]);
    }
    printf("\n");
}
int main()
{
    int i;
    srand((unsigned)time(NULL));
    for(i=0;i<25;i++)
    {
        num[i]=rand()%100;
    }
    printf("Before sorting:\n");
    output();
    bubble((char *)num);
    printf("After sorting:\n");
    output();
    return 0;
}
```