CSE211s Introduction to Embedded Systems

Spring 2023

Sheet 3

Introduction to Embedded Systems

For these questions remember to consult to the ARMv6 Architecture Reference Manual when in doubt.

1. How does the word 0xdec0ded1 appear in memory in a little-endian memory system and a big-endian memory system.

Address	Little-Endian	Big-Endian
n	0xd1	0xde
n+1	0xde	0xc0
n+2	0xc0	0xde
n+3	0xde	0xd1

2. Is the Freescale Kinetis series of ARM Cortex processors little-endian or big-endian?

   Little-endian

3. Does the stack grow toward larger or smaller addresses?

   Smaller addresses.

4. Assuming that SP is 0x0000_2220 initially, what is its value after executing the instruction PUSH {r0,r2}?

   2 registers * 4 bytes each = 8 bytes (0x08)
   SP=0x0000_2220 – 0x08 = 0x0000_2218

5. Assuming that SP is 0x0000_2010 initially, what is its value after executing the instruction POP {r0-r7,PC}?

   9 registers * 4 bytes each = 36 bytes (0x24)
   SP=0x0000_2220 + 0x24 = 0x0000_2244

6. Write the Thumb code (16-bits instruction) to add five to the contents of register r6.

   ADDS r6, #5 (5 is directly encoded in the instruction itself, 5 bits max)

7. Write the Thumb code to subtract 1000 from the contents of register r6, using r3 as a temporary register.

   LDR r3, =1000
   SUBS r6, r3

8. Write the Thumb code to multiply the two 32-bit in memory at addresses 0x1234_5678 (ex: 5) and 0x7894_5612, storing the result in address 0x2000_0010.

   LDR r0, =0x12345678 ;this instruction loads 0x12345678 in r0.

```
LDR r1, [r0] ;r1=5
LDR r0, =0x78945612
LDR r2, [r0]
MULS r2, r1, r2
LDR r0, =0x20000010
STR r2, [r0] ;this instruction stores the value of r2 in the location pointed by r0.
```

9. Write the Thumb code to load register r0 with the letter ‘E’ if the number in r12 is even, or else the letter ‘O’ if it is odd.

```
    MOV     r12, r7
    LDR     r0, =1
    ANDS    r0, r7
    BNE     Odd

    LDR     r0,='E'
    B       Done

Odd
    LDR     r0,='O'
Done
    … next instruction
```

10. Why would you use a BLX instruction rather than a BX instruction?

The BLX instruction saves the return (link) address, so it is used as a subroutine call. The BX instruction doesn’t save the return address, so it is used as a subroutine return or computed jump instead.

11. Why would you use a BL instruction rather than a BLX instruction?

Both instructions can be used to call subroutines. BLX requires the target address to already be stored in a register. BL uses an address which is computed as the current PC address plus an offset which is stored in the instruction.

12. Write the thumb code to call a subroutine at address 0x6555_8888. Where is the return address located?

```
LDR r0,=0x65558888
BLX r0
```

The return address is stored in the link register (lr, r14)