CSE/EE 5/7385 Microcontroller Architecture and Interfacing

HOMEWORK 2

1. An embedded systems engineer is examining the machine code for a store instruction with a pre-indexed operand. The hexadecimal code is:

0xE583706C

The general form of the store instruction is:

STR <Rd>, [<Rn>, #<immed_5>*4]

a) Give the actual destination register: _____

b) Give the actual base address register: _____

c) Give the actual address offset value (in hex): _____

2. Assume that the memory of an ARM processor has the following content and that the ARM processor is operating with little endian access:

ADDRESS	DATA
0x4010	20
0x4011	40
0x4012	00
0x4013	00
0x4014	DA
0x4015	14
0x4016	40
0x4017	00
0x4018	00
0x4019	00
0x401A	40
0x401B	1C
0x401C	1C
0x401D	1C
0x401E	40
0x401F	00
0x4020	00

Give the content of the destination register after the following instructions execute.

a)
```
ldr   r0, =0x4018       ;r0 = _____________
```

b)
```
ldr   r1, =0x4018
ldr   r3, [r1]          ;r3 = _____________
```

c)
```
ldr   r1, =0x4018
ldr   r3, [r1, #4]      ;r3 = _____________
```

d)
```
ldr   r1, =0x4014
ldrsh r3, [r1]          ;r3 = _____________
```

e)
```
ldr   r1, =0x4014
ldrsb r3, [r1]          ;r3 = _____________
```

g)
```
ldr   r1, =0x4014
ldrsb r3, [r1, #4]      ;r3 = _____________, r1 =_____________
```

h)
```
ldr   r1, =0x4014
ldrsb r3, [r1, #4]!     ;r3 = _____________, r1 =_____________
```

i)
```
ldr   r1, =0x4010
ldr   r2, =0x3
ldr   r3, [r1, r2, lsl #2]!   ;r3 = _____________, r1 =_____________
```

3. An embedded systems designer is implementing a stack. She must decide which of the following multiple register load/store instructions should be used for the "pop" and "push" instructions. There are four possible types of stacks listed below. Give the appropriate form of **`ldm`***xx* and **`stm`***xx* instructions to use for each stack type where *xx* may be **`ia`**, **`ib`**, **`da`**, or **`db`**.

STACK TYPES

Fully Ascending Stack: "pop" _____ "push" _____

Fully Descending Stack: "pop" _____ "push" _____

Empty Ascending Stack: "pop" _____ "push" _____

Empty Descending Stack: "pop" _____ "push" _____

4. Consider the following ARM program segment and assume that the 32-bit data bus requires 2 clock cycles for the transfer of a single word. The microcontroller system clock is running at 100 MHz. Give the data transfer bandwidth in Mb/s (Megabits per second). NOTE: Mb is MegaBITS NOT MegaBYTES.

```
            ;r0 points to the start of the source data
            ;r3 points to the start of the destination data

            mov    r2, #0x1
            mov    r4, #0x1, 30
looplab     ldmia  r0!,   {r5-r7}
            stmia  r3!,   (r5-r7}
            cmp    r4, r2
            mov    r4, r4, 1
            bne    looplab
stop        b      stop
```

Bandwidth in Mb/s = _____

5. Give the value of **r0** (in hex) after each instruction executes.

a) `mov r0, 0x1 ; r0 = _____`

b) `mvn r0, 0x1 ; r0 = _____`

c) `mov r0, 0x1, 11 ; r0 = _____`

d) `mvn r0, 0x1, 6 ; r0 = _____`

6. Draw the logic diagram of a 4-bit barrel shifter that receives a 4-bit data value as input, **D[3:0]**, and a 2-bit control value, **C[1:0]**, as input and produces a 4-bit value output, **F[3:0]**. The output value is a left-shifted version of the input value and is shifted by the number of bits specified in **C[1:0]**. So, **F[3:0]**, can be left-shifted by 3, 2, 1, or 0 bits. The only logic parts you can use are 2:1 multiplexers with 4-bit data inputs and outputs. You can also use concatenation for the inputs. For example, if **D[3:0]=$d_3d_2d_1d_0$**, the concatenation indicated by **{D[1:0],00}** would be the bit string **$d_1d_0$00**. HINT: the posted class notes give an example of a 32-bit barrel shifter, so check them if you need help and re-familiarize yourself with the functionality of a multiplexer if you have forgotten how they operate.