

EECS 388
Homework #2
Due: October 1, 2007

Problem #1.

Consider the following setup for memory and registers. Assume the same starting setup for each part of the question. Show the values of the registers and memory locations after each series of instruction(s) are executed. If an unaligned access is encountered, write exception next to the instruction and also show the contents of the memory and address registers. (note* on an illegal access the instruction that caused the unaligned access will not be executed).

Register Set		Main Memory	
R0	0000	0000	12
R1	0000	0001	34
R2	0002	0002	56
R3	0004	0003	78
R4	0002	0004	9A
R5	0003	0005	9B
R6	0005	0006	9C
R7	0001	0007	9D

- a. lw r2, r3,r0
 shi r2,r3,0
- b. lw r0, r3,r1
 addi r5,r5, 1
 sw r5,r5,r0
- c. lbi r1,r6, 0
 or r1,r1,r3
 swi r1,r2,2
- d. lw r1,r5,r3
 sw r1, r1, r0

Problem #2.

Translate the following C code into Microblaze assembler. To keep things easy to grade, assume that the starting address of the A array is in memory location 0x0010 and should be loaded into register R1, and the address of the B array is in memory location 0x0040 and should be loaded into R2.

```
Int A[100], B[100]; /*determines the size of the operands */
```

```
For(I = 0; I < 100, I += 2){
```

```
    If (A[I] == B[I])
```

```
        A[I] = 0;
```

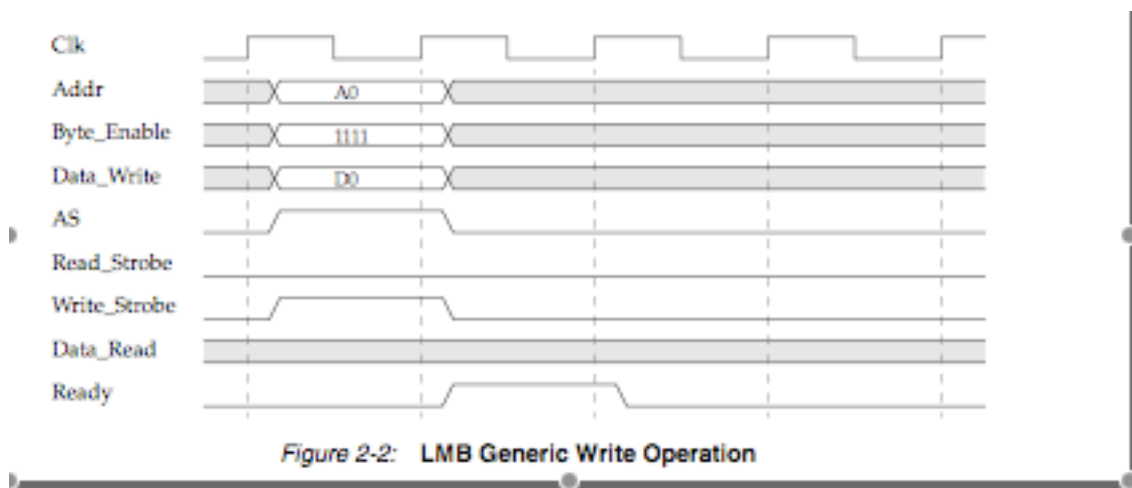
```
    Else
```

```
        A[I] = B[I];
```

```
}
```

Problem #3

Redraw the following waveform for the following instructions.



a. shi, r1, r0, 2

b. lwi r1, r0, 4