EECS388 Embedded Systems: Sample Question

Question 1: How much should you increment the PC after fetch stage in a processor that is a word addressable (1 word=32 bit) and the width of each instruction is 4 Bytes? (4 points)

Answer:

a. 1 (__)
b. 2 (__)
c. 8 (__)
d. 4 (__)

Question 2: Let's assume that a 32 bit variable with value of AABBCCDD in hexadecimal has to be stored in the memory. Assume that each memory address can store **8 bits** and the processor uses **Little Endian** method to store data. Write down the hex digits to be stored in different addresses under data column of the table below. (4 points)

Answer:

Address	data
0x00	DD
0x01	CC
0x02	BB
0x03	AA

Question 3: What are the main components of a single DRAM cell? (4 points)

Answer:

a. Transistor (__)
b. flip-flops (__)
c. inverters (__)
d. capacitor (__)

Question 4: What are the advantages of on-chip SRAM memory over DRAM? (4 points)

Answer:

a. Read/write speed (latency) (__)
b. Ease of fabrication with the processor (__)
c. Cost per bit (__)
d. non-volatile (__)

Question 5: Consider the following C code. Let us assume that the function sum() is currently executing. What is the address stored in the return address register 'ra'? (4 points)

```
int sum(int a, int b){
  return a+b;
}
void main() {
  int x=20;
  int y=10;
  int z= sum(x,y);
  z++;
  y=z+1;

}
```

Answer:

a. Memory address that stores line "z++;" (__)
b. Memory address that stores line "y=z+1;" (__)
c. Memory address that stores line "return a+b;" (__)
d. Memory address that stores line "int z= sum(x,y);" (__)

Question 6: Why embedded systems primarily use the C programming language? (4 points)

Answer:

a. Supports bitwise operations (__)
b. C language is architecture-dependent (__)
c. Secure against buffer overflow attack (__)
d. Lower code overhead for execution (__)

Question 7: For the following assembly program, how many times the Addi instruction inside the loop will execute? All instructions are based on 32 bit MIPS ISA. (4 points)

```
addi r1, r0, 4
Loop:
addi r1, r1, -1
bne r1, r0, Loop
sub r1, r0, r0
```

Answer:

a. 0 (__)
b. 1 (__)
c. 4 (__)
d. 5 (__)

Question 8: What should the MIPS assembly for the following C code?

```
If (a<b) {
 goto Label1; }
```

Assumptions: a is in $r1, b is in $r2. Use two instructions.

Answer:

```
slt   $r3, $r1, r2          # if $r1<$r2, make #r3=1
bne $r3, $r0 , Label1       # goto Label1, if $r3 not equal to 0
```

Question 9: Mark the cases inside the brackets for which you cannot execute the same assembly program compiled for one microprocessor in another one. (4 points)

Answer:

a. If both microprocessors have the same ISA (___)
b. If both microprocessors have the different ISA and different microarchitecture (___)
c. If both microprocessors have the same ISA but different microarchitecture (___)
d. If both microprocessors have different ISA (___)

Question 10: Mark the multiplications that can be performed using logical left shift? Assume Z=00000111 in binary. (4 points)

Answer:

a. Z x 9 (___)
b. Z x 16 (___)
c. Z x 32 (___)
d. Z x 64 (___)

Question 11: Which of the following statements (if any) are true for the code used in the lab below. (4 points)

```
1  #include <stdint.h>
2  #include "eecs388_lib.h"
3  int main()
4  {
5      int gpio = GREEN_LED;
6
7      gpio_mode(gpio, OUTPUT);
8
9      while(1)
10     {
11         gpio_write(gpio, ON);
12         delay(1000);
13         gpio_write(gpio, OFF);
14         delay(300);
15     }
16 }
```

a. The while(1){…} loop iterates just once (___)
b. The gpio_write() in line 11 is a function declaration (___)
c. We may not need such while loop if an operating system is present (___)
d. The eecs388_lib.h contains the definition of function gpio_write() (___)

Question 12: which of the following memory technologies are volatile? (4 points)
Answer:

a. DRAM (___)
b. Hard drive (___)
c. Flash memory (___)
d. Registers (___)

Question 13: Which of the following statements are true for port mapped and memory mapped I/O? (4 points)
Answer:

a. Port mapped I/O uses only load and store instructions to communicate with I/O devices (___)
b. In memory mapped I/O, the I/O memory is mapped into the CPU address space (___)
c. In port mapped I/O, the I/O memory is mapped into the CPU address space (___)
d. The Hifive platform used in your lab uses memory mapped I/O (___)

Question 14: For the C code presented below, indicate the section in memory layout each variable is stored or makes use of. Circle the correct option for each question. (BSS is Uninitialized Data Segment and Data is Initialized Data Segment). (8 points)

```
int globA=10;
int globB=0;
int main () {
   int varA;
   int varB = 10;
   static int varC = 1;
   static int varE = 0;
   char *varD;
   varD = (char*)malloc(8);
   varA = varB + varC;
   return varA;
}
```

Answer:

a. int globB=0; (Stack—Heap—*BSS*—Data)
b. int varA; (*Stack*—Heap—BSS—Data)
c. int varB = 10; (*Stack*—Heap—BSS—Data)
d. static int varC = 1; (Stack—Heap—BSS—*Data*)
e. char *varD; (*Stack*—Heap—BSS—Data)
f. varD = (char*)malloc(8); (Stack—*Heap*—BSS—Data)

Question 15: What information is **not stored** in program counter register? (4 points)

Answer:

a. Current instruction (__)
b. Next instruction (__)
c. Address of currently executing inst (__)
d. Address of next instruction (__)

Question 16 For the following assembly program, what is the content in memory address 100 after executing the following code? All numbers are decimal. Please note that register r0 contains zero. All instructions are based on MIPS ISA.

```
addi r1,r0, 54
addi r2,r0, 58
addi r3,r0, 100
sw r1, 0(r3)
sw r2, 4(r3)
```

Answer:

a. 54 (__)
b. 58 (__)
c. 100 (__)
d. 4 (__)