**EECE 237 Spring 2016** Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Quiz #1** (Total 50 points) Section # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Convert 0x0ABC to the equivalent decimal number. Show your work process. (5 points)

10x16\*\*2+11x16+12=2748 Answer 274810

2. What is the assembly language instruction to **return** from a function? Circle your answer.(5 pt)

a. BL b. BEQ c. BL return d. BX return e. BX LR

4. Number (1,2, 3, 4) each event by the order of instruction execution. (8 points)

|  |  |
| --- | --- |
| Sequential  Order | Events |
| 3 | Instruction decoder decodes the instruction code into micro-codes. |
| 1 | PC fetches an Instruction code from the code ROM. |
| 4 | Registers and ALU execute a series of operations such as read /write a register, shift ALU, transfer data. |
| 2 | Instruction Register loads the instruction code. |

5. Number (1,2, 3....) each event by the order of interrupt execution. (12 points)

|  |  |
| --- | --- |
| Sequential  Order | Events |
| 3 | The processor halts the currently running program. All active data in the registers & memory are pushed to the stack. The stack pointer is adjusted. |
| 6 | The processor resumes the previous thread. |
| 1 | The processor is running its own routine in thread. |
| 2 | An I/O source initiates an IRQ (interrupt request). |
| 4 | The processor runs the Interrupt Service Routine (ISR). |
| 5 | Once ISR is served (completed), the processor pops the previous data from the stack back to the registers & memory. |

6. Check if each statement is true or false.(20 points)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Statement | True | False |
| A | Overflow occurs when data operation generates a carry bit. |  | x |
| B | Flushing is an operation that clears the content of ALU. |  | x |
| C | PC (program counter) holds the next instruction code to execute. |  | x |
| D | Positive is one of the flags in PSR (program status register). |  | x |
| E | The flow in a pipeline can be broken when an unconditional branch instruction is executed. |  | x |
| F | In ROM, the first instruction to start at reset is located at 0x0000 0000. |  | x |
| G | The stack pointer (SP) is located in the data (SRAM) memory. |  | x |
| H | The difference between exception and interrupt is in the way of handling the interrupt service. |  | x |
| I | Vector represents a order (sequence) of multiple interrupts. |  | x |
| J | MOV instruction moves data from one memory location to another memory location. |  | x |

|  |  |
| --- | --- |
| Address | Data |
| 0x2000 0FF0 | 0x0000 AAAA |
| 0x2000 0FF4 | 0x0000 0102 |
| 0x2000 0FF8 | 0x0000 2000 |
| 0x2000 0FFC | 0x0000 0010 |
| 0x2000 1000 | 0x0000 0020 |
| 0x2000 1004 | 0x0000 0030 |
| 0x2000 1008 | 0x0000 0800 |
| 0x2000 100C | 0x0000 0400 |
| 0x2000 1010 | 0x0000 0001 |
| 0x2000 1014 | 0x0011 0000 |
| 0x2000 1018 | 0x0200 0FFF |
| 0x2000 101C | 0x0000 0000 |
| 0x2000 1020 | 0x0002 0010 |
|  |  |

6. Write all values in hexadecimal. Use the memory chart shown at the next page. The results from each question are self-contained and do not carry over to any other questions. (Assume leading zeroes if not specified – 0x02 = 0x0000 0002) In ARM, the address in SP decreases as data are pushed to the stack. The address in SP increases as data are popped out. (10 points)

|  |  |  |  |
| --- | --- | --- | --- |
| R0 = 0x02 | R3 =0x10 | R6 = 0xFFFF FF00 | R9 = 0x2000 1020 |
| R1 = 0x04 | R4 = 0x06 | R7 = 0x2000 1008 | R10 = 0x2000 0FF0 |
| R2 = 0x07 | R5 = 0x20 | R8 = 0x2000 100C | R11 = 0x2000 1000 |

|  |
| --- |
| a. After the instruction MVN R3, R4, LSR, 2  What are the contents of R3? \_\_0xFFFF FFFE\_\_\_\_  b. After the instruction ADD R0, R5, #10  What are the contents of R0? \_\_0x02A\_\_\_\_\_\_  c. After the instruction LDR R2, [R7]  What are the contents of R2? \_\_\_\_0x0000 0800\_\_  d. After the instruction LDR R4, [R8, R1]  What are the contents of R4? \_\_\_\_0x0000 0001\_\_\_\_  e. After the instruction STR R6, [R10]  What memory location is modified? \_0x2000 0FF0\_\_\_\_ |