EECS388 Embedded Systems Fall 2022

HW 2 – Due date: Friday Dec 9th **11:59PM**

Canvas submission.

****** Important: There is no late policy. It means that if you submit after the deadline you'll receive a zero! ******

The reason is that we release the solution right after the deadline

Part 1 – True/False (1 point each)

Indicate true or false for each of the following statements:

T 1. Function (subroutine) implementation requires Control instructions. (T/F)

T 2. We have one register file in LC-3 and all the instructions in all subroutines read and write from that one register file.

T 3. We use regular load and store instructions to access the status and data registers of a Memory Mapped IO (MMIO) device.

T 4. We use a handshaking protocol to control the exchange of data between a microcontroller and an asynchronous I/O device.

F 5. In a system with an OS, a programmer can directly access I/O registers.

T 6. The TRAP instruction in LC3 requests service from the OS running in the privilege mode.

F 7. TRAP Vector Table is a component inside the Processing Unit in the Von Neumann computer model.

F 8. Interrupt and Trap are the same thing.

F 9. An external interrupt can always preempt the execution of the currently running program on the CPU.

F 10. The Interrupt enable-bit ensures that the highest priority interrupt is always selected to be sent to the CPU.

T 11. In case several devices request to send an interrupt to the CPU simultaneously, the highest priority device is selected to send the Interrupt.

T 12. An interrupt is sent to the processor whenever a device needs attention.
you can interprete this as "false" as well if you think about pri

T 13. In direct mode interrupt, we have a common interrupt handler that finds the cause of the received interrupts.

T 14. An I/O interface is considered synchronous if there is a clock signal between the sender and receiver.

F 15. We cannot have a parallel and synchronous interface.

F 16. RS232 is a synchronous serial protocol.

F 17. I2C uses a shift register to shift out bits to the output port and shift in bits from the input pin for communication.

T 18. PWM is an effective way of controlling DC motors' speed, because the speed of a DC motor changes with the average input voltage.

F 19. Brushed motors have a permanent-magnet rotor.

T 20. PWM is a simple way to generate signals with various average voltages.

T 21. We often use PWM to control the angle and position of servo motors.

F 22. Stepper motors use PWM.

F 23. Stepper motors are equipped with a positional feedback device.

T 24. Baremetal execution is faster than OS-assisted execution (assuming similar hardware config).

T 25. Process Control Block (PCB) data structure is used to support multiple active processes in a system.

T 26. Context switches have overhead.

F 27. A real-time scheduler should always prioritize more critical tasks.

T 28. The priority of jobs dynamically changes in an EDF scheduler.

Question #2: Regarding option "d", you can argue that a fast device that does not receive frequent events does not benefit from polling. Think about a high bandwidth network adapter that does not receive any packet ….
If you connect speed to the frequency of events, then option "d" is correct.

Part 2 – Multiple Choice

Select the appropriate options for the following questions:

1. Why do we need to save *PC* before a subroutine call? [2 points]
   a. To know the return address
   b. To be able to perform operations on the PC register in the subroutine
   c. Not to miss any of the register values saved in the register file
   d. To use it and jump to the starting address of the subroutine

2. Select all correct statements about polling and Interrupt (chose all the correct statements) [+1 point for each correct selection]
   a. Polling or Interrupt are two mechanisms that are used to identify whether an I/O device has data (or need attention)
   b. A device with bursty behavior can adaptively use polling and interrupt to reduce Interrupt's context switches and Polling's CPU cycle waste.
   c. Polling save CPU resources
   d. Polling is better for fast devices and Interrupt is better for slow devices.
   e. We used the Interrupt mechanism when implementing blinky LED in the lab. We used a timer and interrupt
   f. We can implement Polling by checking the value of a volatile MMIO register in a while loop.
   g. An Interrupt can result in the preemption of the currently running program.

3. What is the benefit of TRAP (system call)? [2 points]
   a. It implements protection for shared device registers
   b. Improves programmers productivity
   c. Hides low-level details of I/O device communication from the programmer.
   d. All above

4. What is a *TRAP service routine*? [2 points]
   a. A function that executes in the privileged mode on behalf of the user
   b. A function that executes by the OS and provides a service to the user
   c. A function that its instructions are placed in the user memory space.
   d. All above.
   e. A & B

5. What is the functionality of the *TRAP Vector Table*? [2 points]
   a. Store the starting address of TRAP service routines
   b. Store the starting address of interrupt service routines
   c. Store the first instruction of the TRAP service routines
   d. Store the return address to the next instruction after the TRAP instruction

6. The advantage of serial interfaces is that: [2 points]
   a. They use fewer pin on the processor and I/O device
   b. They consume lesser power
   c. They deliver higher bandwidth (bit rate) compared with a parallel interface.
   d. All above
   e. A & B

7. What is stored inside an *interrupt vector table*? [2 points]
   a. Interrupt vector table offset
   b. Address of an interrupt service routine
   c. Interrupt service routine.
   d. Interrupt vector number.

Part 3 – Short Answer

Provide a short answer to the following questions:

1. Assume that we have a keyboard that is connected to a microcontroller. When is operating the keyboard in interrupt-driven mode beneficial? Explain each selection. (multi-choice) [4 points]
   a. The inter-arrival time of keystrokes is fixed.

   b. The inter-arrival time of keystrokes is unknown.

   we don't waste CPU Cycles

   c. The average inter-arrival time of keystrokes is in the same order as the micro controller's frequency

   d. The average inter-arrival time of keystrokes is orders of magnitudes larger than the micro controller's frequency

The interrupt overhead would not be significant since we have few interrupts and we don't waste CPU cycles for polling

2. Why should we save registers in the memory before calling a subroutine or initiating an interrupt? [2 points]

To protect them from being over written by the subroutine and lose data.

3. Why do we need to have a privileged mode of execution in processors? [2 points]

protect share data and resources

4. Why do we set/reset the privileged bit while executing TRAP/RTI instructions? [2 points]

because trap executes in the privilege mode

5. Why do processors handle interrupts in the instruction boundaries? [2 points]

simplicity

6. Why do we save PC and PSR before initiating an interrupt? [2 points]

know where to return after finishing the interrupt service routine

7. Where do we save PC and PSR before initiating an interrupt? [2 points]

Memory (system stack)

8. What is baudrate in UART? [2 points]

The agreed upon speed if the transmission of one bit between sender and receiver

9. Write an advantage and a disadvantage for bus compared with a point-to-point network. [2 points]

Bus:
+ less resources
-Need arbitration

Point to point:
+ faster
- More expensive

10. Your classmate wrote this code and claimed that this would set the content of memory address Value1 to zero. Do you agree with your classmate? What is your proposed solution? [3 points]

NO! R1 is overwritten in FUNC

save R1 & restore it.

```
            .ORIG x3000
(1)         AND     R1, R0, #0 // R1 <- R0 & 0
            JSR     FUNC
(2)         ST      R1, Value1 // Mem[Value1] <- R1
            HALT

FUNC        Add     R1, R1, #1 // R1 <- R1 + 1
            JMP     R7

Value1  (3) .BLKW   1
            .END
```

(1) (2) ST R1, SAVE
LD R1, SAVE

(3) SAVE .BLKW 1

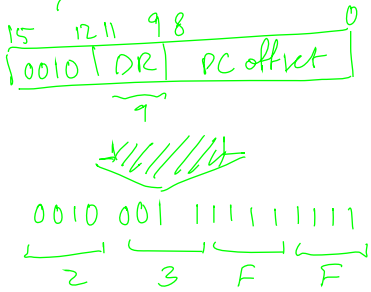
11. Consider the following address space and the following assembly program and answer the questions. (X, Y, Z)

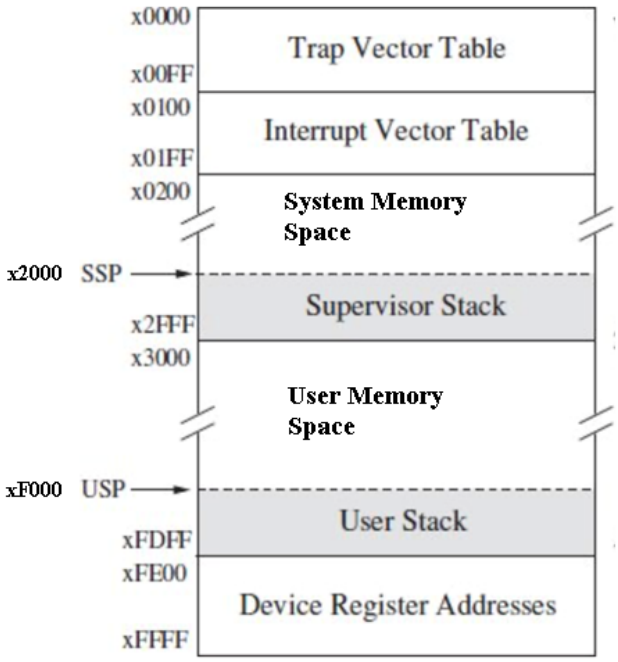
memory address

		Condition Code (N,P,Z)			
x3000	1		.ORIG	x3000	
x3001	2		LDI	R3, L1	// R3 = Mem[Mem[L1]] = Mem[xFE04]
x3002	3		LDI	R4, L2	// R4 = Mem[x2100]
x3003	4		LDI	R5, L3	// R5 = Mem[x0001]
x3004	5		LDI	R6, L4	// R6 = Mem[x02FF]
x3005	6	(0,1,0)	LD	R7, L3	// R7 = Mem[L3] = Mem[x3014] = x0001
x3006	7	(0,1,0)	ADD	R7, R7, #1	// R7 = R7+1 = 0x0002
x3007	8	unchanged	ST	R7, L5	// Mem[L5] = R7 => Mem[x3013] = x0002
x3008	9	"	BRz → not taken	FINISH	
x3009	10	L0	LD	R1, L0	// R7 = instruction "LD R7, L0" = 0x23F
x300A	11		AND	R2, R2, x0000	
x300B	12		ST	R2, L0	
x300C	13	STR	.STRINGZ	"Hi!"	// 'H' // 'i' // '!' // 'NULL'
x300D					
x300E					
x300F	14	L1	.FILL	xFE04	
x3010	15	L2	.FILL	x2100	
x3011	16	L3	.FILL	x0001	
x3012	17	L4	.FILL	X02FF	
x3013	18	L5	.BLKW	2	
x3014	19	FINISH	.END		

lecture 8 slide 8

"LD"





a. Which lines of the code access a Memory-Mapped I/O device? What are the addresses of the IO registers? [2 points]

line 2 0xFE04

b. In which lines do we access a privileged memory space? [2 points]

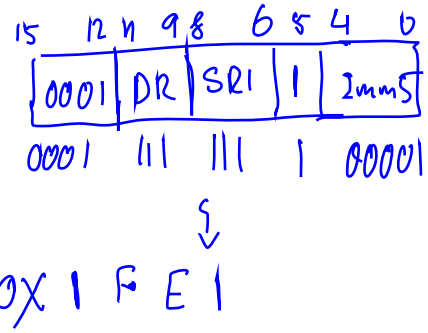
lines 2, 3, 4, 5

c. Do we access Trap Vector Table in the code snippet? [2 points]

Yes, in line 4

d. After the code execution, what is the content of the following memory addresses? Write it in Hex. [4 points]

1. L5: 0x0002
2. L5+1: Unknown !
3. x3005: "ADD R7, R7, #1" ⇒



4. x3008: 0x 0000

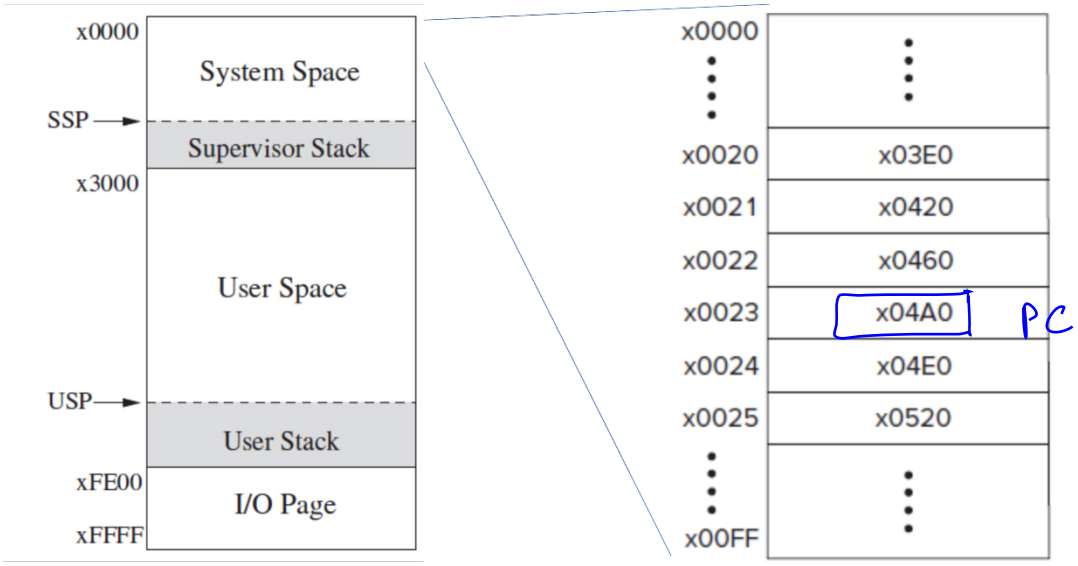
The STORE instruction at line 12 updates address 0x3008 to 0x0000

5. x300B: 0x 0048 → ZEXT (0x 48)

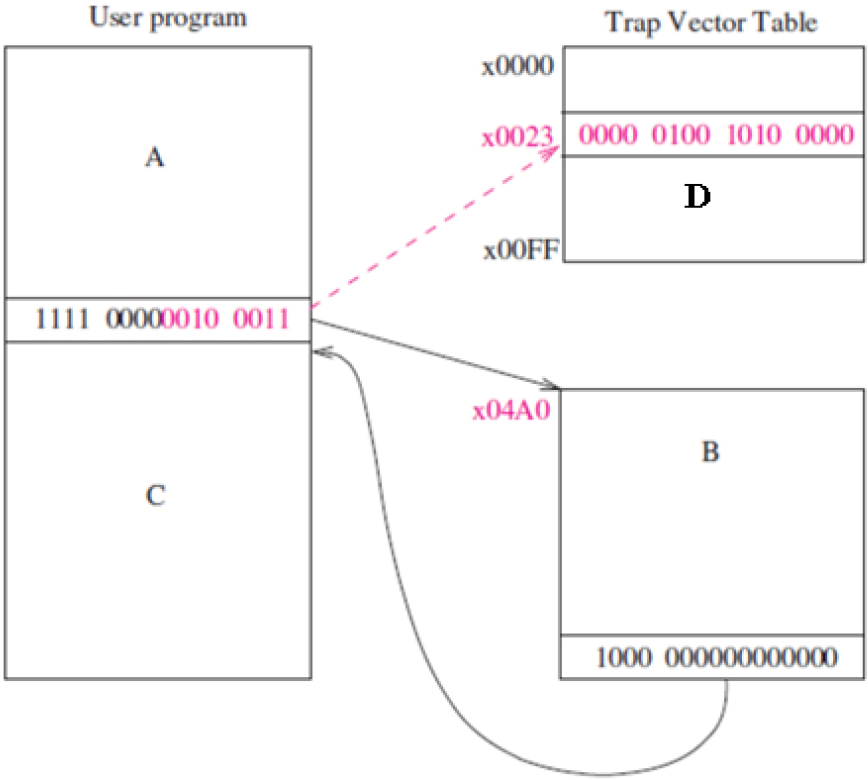
Ascii of "H"

12. Consider the following TRAP vector table in LC3.
What is the PC value after executing the following TRAP instruction? [3 points]

```
.ORIG x3000
TRAP x23
```



13. Answer the following questions based on the figure.



a. What is the trap vector number? [1 points] 0X23

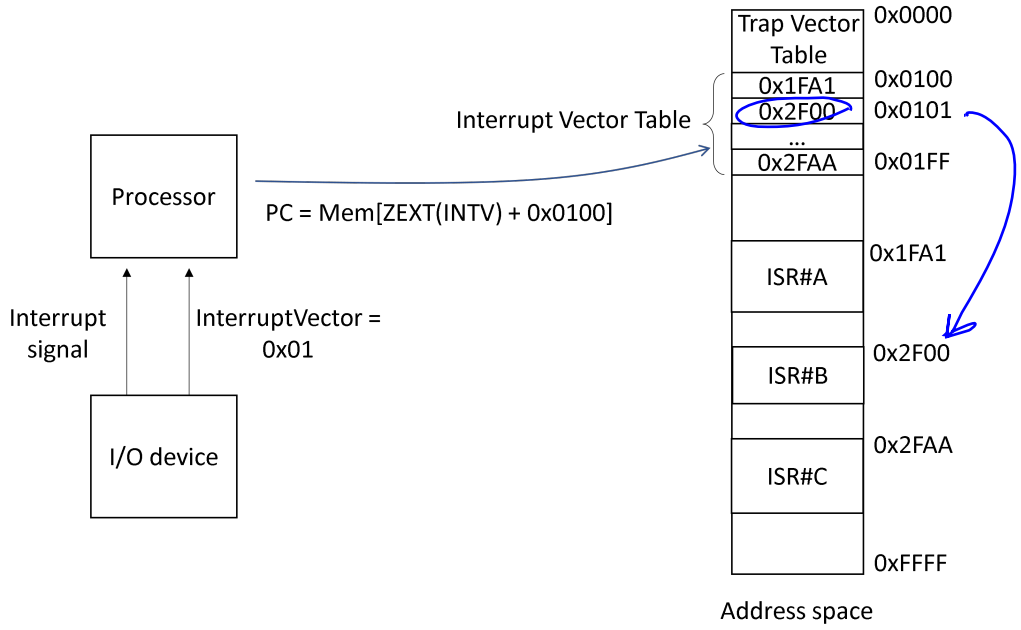
b. What is the size of the Trap Vector Table? [1 points]

256

c. Which section of the memory stores the Trap service routine? [2 points]

a. A
b. B
c. C
d. D

14. Consider the following illustration of a system implementing vectored interrupts. Which ISR will be executed assuming that the interrupt vector number provided by the I/O device is 0x01? [3 points]
    a. ISR#A
    b. ISR#B
    c. ISR#C



15. What are the data that the following bit stream is transmitting? Assume RS-232 serial protocol.Is there any error in the transmission? [4 points]

11111110111111111111111111110000000000011111111

Start Data Stop P

Start data Stop P

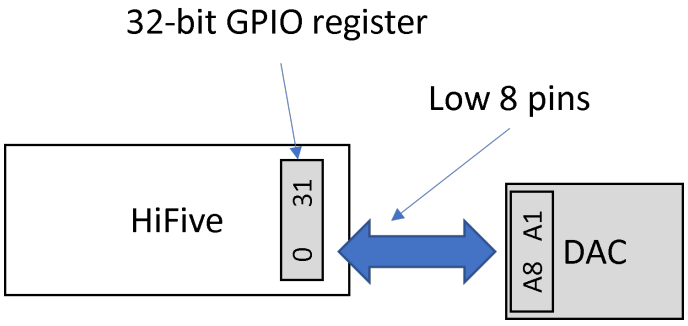
0XFF

has Error because parity should be 0

0x00

No Error.

16. Complete the following code to set the output voltage to 40/256 volts. Assume that the DAC is a +10V output DAC. [4 points]




```
void set_dac()

{

   uint32_t val = *(volatile uint32_t *) (GPIO_CTRL_ADDR +
GPIO_OUTPUT_VAL);

      // YOUR CODE HERE

   *(volatile uint32_t *) (GPIO_CTRL_ADDR + GPIO_OUTPUT_VAL) = val;

}
```

$$V_0 = 10 * \left(\frac{A_1}{2} + \frac{A_2}{4} + \frac{A_3}{8} + \frac{A_4}{16} + \frac{A_5}{32} + \frac{A_6}{64} + \frac{A_7}{128} + \frac{A_8}{256} \right)$$

$$V_0 = \frac{40}{256}$$

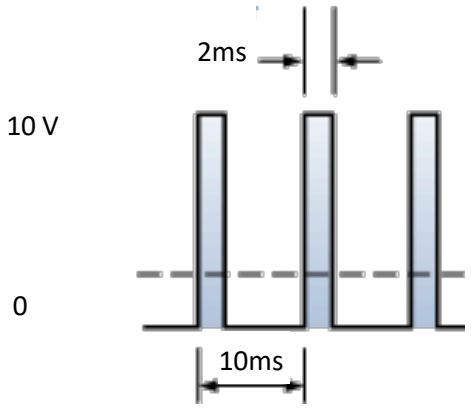
$$\Rightarrow \left(\frac{A_1}{2} + \frac{A_2}{4} + \frac{A_3}{8} + \frac{A_4}{16} + \frac{A_5}{32} + \frac{A_6}{64} + \frac{A_7}{128} + \frac{A_8}{256} \right) = \frac{1}{64}$$

$$\Rightarrow \boxed{A_6 = 1}$$

Code: val &= 0XFF FF FF 00;

val |= 0X 00 00 00 04;

17. What is the average voltage of the following PWM signal? [2 points]



Duty cycle = $\frac{2ms}{10ms} = 0.2$

Avg voltage = $10 * 0.2 = 2V$