

**This examination is closed book, closed notes, closed calculator**

**Show all your work on these pages!**

**Excerpts from the UC Davis *Code of Academic Conduct*:**

1. Each student should act with personal honesty at all times.
2. Each student should act with fairness to others in the class. This means, for example, that when taking an examination, students should not seek an unfair advantage over classmates through cheating or other dishonest behavior.
3. Students should take group as well as individual responsibility for honorable behavior.

**I understand the honor code and agree to be bound by it.**

**Signature** \_\_\_\_\_

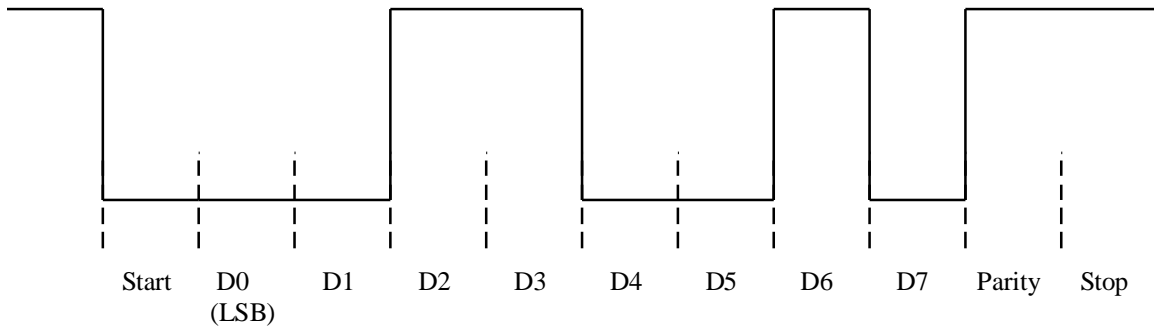
**NAME (print)** \_\_\_\_\_ **SOLUTIONS** \_\_\_\_\_

**Student ID** \_\_\_\_\_

PROBLEM	MAXIMUM (Points)	SCORE
1	30	
2	20	
TOTAL	50	

## I. UART (30)

1. Draw the UART transmission waveform for the character 'L' (ASCII code 0x4C) with 8-bit data, 1 stop bit, start bit and even parity. Label the bits in the waveform and mark the divisions between the bits. (5)



2. Explain how each of the following UART interrupt sources are triggered when the Tiva is in its default configuration. Be specific. (6)

UART\_INT\_RT

**The receive timeout interrupt is triggered when the receive FIFO is not empty, and no further data is received over a (32-bit or 64-bit) timeout period.**

UART\_INT\_TX

**The transmit interrupt is triggered when the transmit FIFO progresses through the programmed trigger level (from above). The default FIFO level is 1/2 full.**

UART\_INT\_RX

**The receive interrupt is triggered when the receive FIFO reaches the programmed trigger level (from below). The default FIFO level is 1/2 full.**

3. How is an ISR address loaded into the Tiva processor's interrupt vector table using our Keil development tools? (2)

**The ISR address is entered into the vector array in the Startup.s file.**

What is the default location of the Tiva processor's interrupt vector table in its address space? Is the vector table in flash memory or SRAM by default? (2)

**By default, the vector table is fixed at 0x0000 0000 on system reset. This is in flash memory.**

4. The TivaWare Peripheral Driver Library contains the following function.

```
bool ROM_UARTCharPutNonBlocking (uint32_t ui32Base, uint8_t ui8Data)
```

- a) Explain how the function works. (2)

**The function puts a character into the transmit FIFO if there is room. If there is no room, it returns immediately with a return value of false.**

- b) What kind of problem can you run into if you use this function without checking the return value? (3)

**If the FIFO is full, the character will not be stored in the FIFO. If you don't check the return value, the character will never be sent.**

5. Write function that uses polling (i.e. *not interrupts!*) to receive a string of characters from UART0 and store them into a static character array. When '\r' is received, store it in the character array and also store '\n' and a null termination ('\0') and exit the loop. Then toggle the red LED (PF1) while leaving the blue and green LEDs (PF2 and PF3) unchanged and exit the function.

Assume that all the necessary initializations have been done for you. Show the declaration of the character array as well as any other variables that you use. (You can make the character large (~200 bytes) and you do not need to check for an array overflow condition.) Show how your function is called from main. (10)

```
char buffer[200];           // buffer (can be global or local to main)
```

```
void In_str (char *ptr) {
    char in_char;
    unsigned char i = 0, leds;           // i = index for char array

    do {
        in_char = ROM_UARTCharGet(UART0_BASE);
        ptr[i++] = in_char;
        if (in_char == '\r') {
            ptr[i++] = '\n';
            ptr[i++] = '\0';
        }
    } while (in_char != '\r');
    leds = ROM_GPIOPinRead(GPIO_PORTF_BASE, GPIO_PIN_1);
    ROM_GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1, ~leds);
}

In_str(buffer);                 // call from main
```

## II. XBees (20)

1. An XBee has the following configuration:

ID (PAN ID): 0x3332  
CH (Channel): 0x0B  
MY: 0x1234  
DH: 0x0  
DL: 0x2222

Show how to configure a second XBee so that it can communicate with the XBee described above. The two XBees should be able to send and receive messages specifically to or from each other (without using broadcast messages). (5)

ID: **0x3332**

CH: **0x0B**

MY: **0x2222**

DH: **0x0**

DL: **0x1234**

2. An API mode 1 UART data frame has the following components: Start byte, Length, Frame Data, Checksum. The Frame Data for a UART data frame structure is shown below. Build the complete data frame structure by adding the Start byte, Length, and Checksum in the correct locations. Show your calculations for the checksum. (5)

			0x08	0x43	0x4d	0x31	0x02	0x00	
--	--	--	------	------	------	------	------	------	--

<b>0x7E</b>	<b>0x00</b>	<b>0x06</b>	0x08	0x43	0x4d	0x31	0x02	0x00	<b>0x34</b>
-------------	-------------	-------------	------	------	------	------	------	------	-------------

08	ff
43	- <u>cb</u>
4d	34
31	
02	
<u>00</u>	
cb	

3. What does it mean for communication to be full-duplex? Can the XBees communicate with each other in full-duplex? (3)

**Full-duplex communication means that a device can transmit and receive at the same time. The XBees cannot communicate with each other in full duplex. They use half-duplex communication. (Transmit and receive use the same RF channel, so full-duplex communication is not possible since the messages would interfere with each other.)**

4. What are two advantages of using an XBee in API mode (AP mode 1) compared to transparent mode? (2)

1. **Can send to multiple specific destination addresses without going into AT command mode in order to change DH:DL.**
2. **Can receive status (success/failure) of each RF packet sent**

Also,

3. **Can identify the source (Sending Node) of each received packet**
  4. **Can send AT commands to remote XBees.**
5. If an XBee is in API mode 1 and has DOUT tied to DIN, what do you expect to see on its DOUT when it receives an RF message “ABC” from an XBee in transparent mode? What do you expect it to echo back by RF? Explain your answer. (5)

**The XBee in API mode 1 will send out an RX Packet frame on DOUT. It will be either RX Packet: 16-bit address or RX Packet: 64-bit address, depending on the address mode of the sending XBee. This RX Packet frame will go into DIN, but since it is not a TX Request packet frame, nothing will be sent by RF. Thus, we don't expect to see the XBee in API mode 1 transmit anything by RF based on the message received.**