**OREGON INSTITUTE OF TECHNOLOGY**

**Computer Systems Engineering Technology Department**

**CST 204 – Introduction to Microcontrollers**

**Final Exam**

Name: \_\_\_\_\_\_\_\_Cari Blaker\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1) What do the serial communication acronyms **DTE** and **DCE** stand for?

**DTE:\_\_\_\_Data Terminal Equipment \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (2 points)**

**DCE:\_\_\_\_\_\_\_\_\_\_\_\_\_\_Data Circuit-terminating Equipment \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (2 points)**

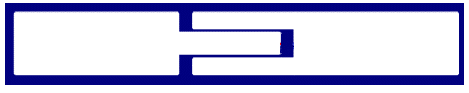
2) Circle correct answer - A terminal is a DTE / DCE device. **(2 points)**

3) Circle correct answer - A computer system with a RS-232 port is a DTE / DCE device. **(2 points)**

4) Circle correct answer - A telephone computer modem is a DTE / DCE device. **(2 points)**

5) Indicate the correct DTE/DCE sense for the serial communications interface shown below (PIN, Socket) by circling the appropriate choice (DTE or DCE). **(5 points)**

PIN Socket



DCE / DTE DCE

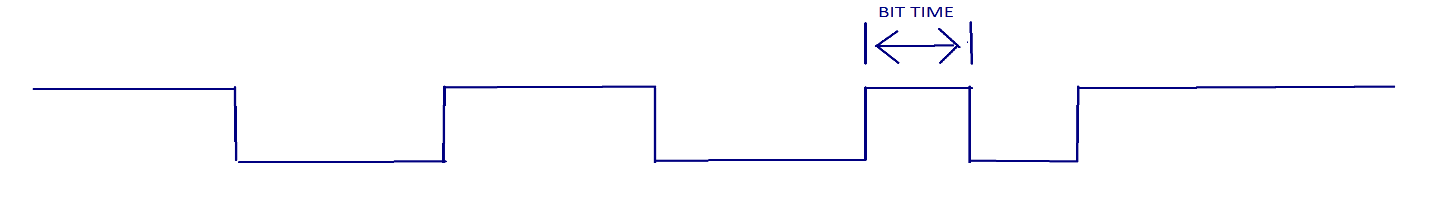
6) Show the direction of the TXD (transmit data/data send) signal with an arrow on the pin/socket interface shown above in question 6. **(5 points)**

7) Assume the logic level signal below (8N1) is received by a UART serial port (time axis goes from left to right).

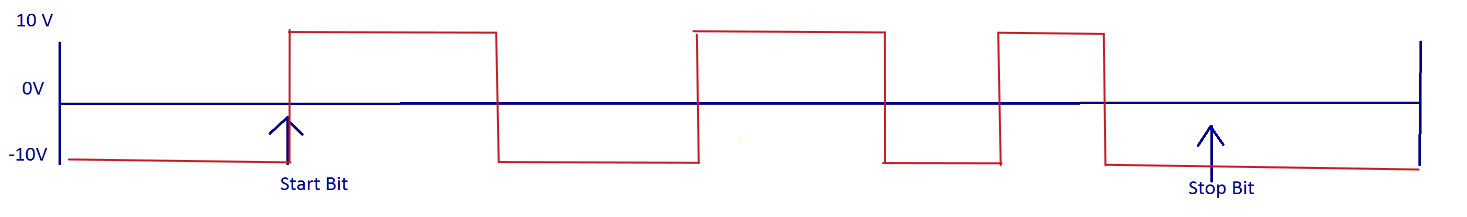
a) Highlight the **start** bit, and the **stop** bit (assume no parity bit) on the drawing below. **(6 points)**

b) Write the binary form of the 8 data bits received (most significant bit first). **(6 points) 01100101**

start 0 1 1 0 0 1 0 1 stop

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (binary)

8) Assume the UART signal in problem 7 is sent over a RS-232 interface with drivers operating at +/- 10V. Sketch on the figure below how the signal would appear in the interface cable. The position of the start and stop bits must be aligned to the indicated positions.  **(6 points)**



9) We want to transmit the serial information in problems 7 and 8 at 4800 baud. What would be the values of BRGH, UxBRG, and the calculated baudrate if the PBCLK is running at 500 KHz? Pick the value of BRGH (0 or 1) that gives the calculated baudrate closest to the desired baudrate.

BRGH = \_\_\_\_\_\_1\_\_\_\_\_\_; UxBRG = \_\_25\_\_\_\_\_\_; calculated baudrate = 4807.69 BAUD \_\_\_\_\_\_\_\_\_\_ **(10 points)**

10) Find the even parity for the binary string **10101011** **(2 points) it’s a 1, so the string becomes 101010111**

11) PIC32 interrupt IRQ number and the CPU vector number are always the same. True / False **(2 points)**

12) The CP0 register that determines where the ISR will “return” to is:

Circle the correct answer - EBASE / EPC / Cause / Status / Intctl **(2 points)**

13) When an ISR is entered, all registers need to be saved. True / False **(2 points) well, all registers being used that may be modified. So, not necessarily every register that exists every time.**

14) If the OSCCON SFR bits COSC<2:0> = <1, 1, 1>, and FRCDIV<2:0> bits = <0, 0, 1> what is the frequency of the PIC32’s SYSCLK? \_\_\_\_\_\_\_\_4 MHz\_\_\_\_\_\_\_ **(5 points)**

15) If the PFMWS<2:0> bits of the CHECON SFR = <1, 1, 1> how many wait states have been selected? 7 wait states have been selected\_\_\_\_\_\_\_ **(5 points)**

16) Recall that CPU instruction execution time not only depends on the SYSCLK but also on the number of wait states programmed. Instruction execution time is (1 + #wait states) \* TSYSCLK. The Cache Control Register (CHECON) configures the number of CPU wait states with the PFMWS<2:0> bits (Programmable Flash Memory Wait States). Determine the **instruction execution time** given the data in problems 14 and 15 above. \_\_\_2,000 ns\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **(6 points)**

**clock period: 1/4\*10^6 = 250 nanoseconds**

**(7 wait states + 1) \* 250 ns = 2,000 ns**

17) If Timer1 and Timer2 have the same interrupt priority and sub-interrupt priority, but they occur simultaneously, which ISR will run first? \_\_\_\_\_timer1 because it has a lower vector address.\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **(5 points)**

18) Analyze the ISR code given below for Timer1. The Prologue and Epilogue have been omitted for clarity. The code’s purpose is to generate a periodic waveform on RA0. The timer’s clock is 1 MHz. Draw the approximate waveform after the system has been running for a while. Indicate the time durations of all of the high and low pulses. **(10 points)**

// Prologue here – omitted |

|

// Timer1 ISR Handler code here |

la t0,T1CONCLR //stop timer |

li t1,0x8000 |

sw t1,0(t0) |

|

la t0,IFS0CLR //clear flag |

li t1, 0x10 |

sw t1, 0(t0) |

la t0, TMR1 |

sw zero,0(t0) |

|

la t0,PORTA |

lw t1, 0(t0) |

ori t1, t1, 0x0000 |

beq t1,zero,next |

nop |

|

la t0,PR1 |

li t1, 1000 //decimal 1000

sw t1, 0(t0)

j finish

nop

next:

la t0, PR1

li t1,2500 //decimal 2500

finish:

la t0,T1CONSET

li t1,0x8000

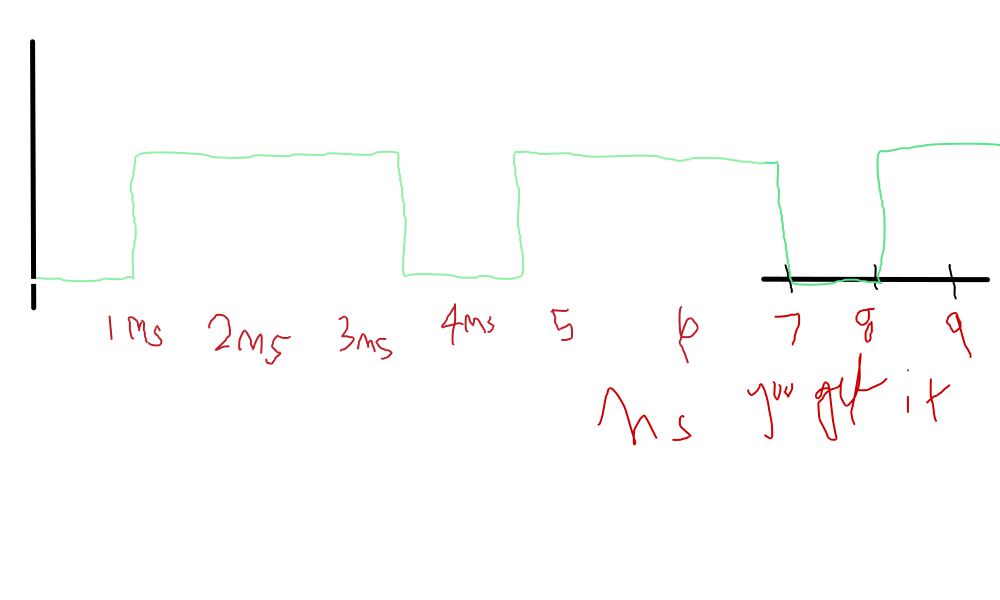
sw t1, 0(t0)

la t0,PORTAINV

li t1,0x0001

sw t1, 0(t0)

//Epilogue here – omitted



The above drawing is assuming that the code snippet given isn’t just run the single time that the snippet shows—it should be in a loop called back to back to get a waveform like that. You know, ish.

19) Determine how long the instructions below will take to execute from **top:** to **bottom:** using the instruction cycle time from question 16 above. **(10 points)**

**top**:

li t0, 1

looping:

addiu t0, t0, -1 Execution Time = \_8000 nanoseconds, or 8 microseconds.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

nop

beq t0, zero, looping

nop

**bottom**:

//continue from here

20) What does an analog to digital converter (A/D) do? **(3 points)**

it converts analog signals into digital representations. Mainly, to enable the conversion of analog signals such as voltage or current into a digital format that can be understood and processed by digital systems.

**CHECON: Cache Control Register**

bit 2-0 PFMWS<2:0>: PFM Access Time Defined in terms of SYSLK Wait states bits

111 = Seven Wait states

110 = Six Wait states

101 = Five Wait state

100 = Four Wait states

011 = Three Wait states

010 = Two Wait states

001 = One Wait state

000 = Zero Wait states

**OSCCON Control Register**

bit 26-24 FRCDIV<2:0>: Fast Internal RC Clock Divider bits

111 = FRC divided by 256

110 = FRC divided by 64

101 = FRC divided by 32

100 = FRC divided by 16

011 = FRC divided by 8

010 = FRC divided by 4

001 = FRC divided by 2 (default setting)

000 = FRC divided by 1

bit 20-19 PBDIV<1:0>: Peripheral Bus Clock Divisor

11 = PBCLK is SYSCLK divided by 8(default)

10 = PBCLK is SYSCLK divided by 4

01 = PBCLK is SYSCLK divided by 2

00 = PBCLK is SYSCLK divided by 1

Note: On Reset these bits are set to the value of the Configuration bits (DEVCFG1<13:12>)

bit 14-12 COSC<2:0>: Current Oscillator Selection bits

111 = Fast Internal RC Oscillator divided by OSCCON<FRCDIV> bits

110 = Fast Internal RC Oscillator divided by 16

101 = Low-Power Internal RC Oscillator (LPRC)

100 = Secondary Oscillator (SOSC)

011 = Primary Oscillator with PLL module (XTPLL, HSPLL or ECPLL)

010 = Primary Oscillator (XT, HS or EC)

001 = Fast RC Oscillator with PLL module via Postscaler (FRCPLL)

000 = Fast RC Oscillator (FRC)

Note: On Reset these bits are set to the value of the FNOSC Configuration bits (DEVCFG1<2:0>)

