King Fahd University of Petroleum and Minerals
College of Computer Science and Engineering
Computer Engineering Department

COE 306: INTRODUCTION TO EMBEDDED SYSTEMS

Term 171 (Fall 2017-2018)
Final Exam
Tuesday Jan. 2, 2018

Time: 150 minutes, Total Pages: 11

Name:__KEY_____ ID:_____ Section: _____

Notes:

- Do not open the exam book until instructed
- Answer all questions
- All steps must be shown
- Any assumptions made must be clearly stated

Question	Max Points	Score
Q1	41	
Q2	19	
Q3	9	
Q4	10	
Q5	11	
Total	90	

Dr. Aiman El-Maleh

[41 Points]

(Q1) Fill in the blank in each of the following questions:

(1) The analog to digital conversion process involves the following two main steps sampling and holding and quantization and encoding.

(2) In ADC, aliasing occurs when the input signal is changing much faster than the sample rate or when the sampling frequency is less than twice the maximum frequency of a signal.

(3) Adding 1 extra bit to an n-bit ADC increases the SNR by 6 dB.

(4) Given an 8-bit ADC with conversion time equal to 100 ns, the maximum frequency that the signal A sin(2πf) can have to guarantee conversion accuracy is Rate of Change x $t_c \leq$ resolution => $f \leq 1/2^n \pi t_c$ => $f \leq 1/256 \pi$ 100ns => $f \leq$12.44 KHZ

(5) Given an analog signal that has a range from -5v to +5v, the analog value 2.5v will be converted to the following hex value C0 using 8-bits successive approximation ADC.

(6) Given an analog signal that has a range from -5v to +5v, complete the table given below showing the generated 8-bits for the analog value 2.5v using delta-sigma modulator:

Input	δ	∑	Out	DAC
2.5	2.5	2.5	1	+5v
2.5	-2.5	0.0	1	+5v
2.5	-2.5	-2.5	0	-5v
2.5	7.5	5.0	1	+5v
2.5	-2.5	2.5	1	+5v
2.5	-2.5	0.0	1	+5v
2.5	-2.5	-2.5	0	-5v
2.5	7.5	5.0	1	+5v

(7) The fastest ADC technique is Flash ADC, while the ADC technique with highest resolution is delta-sigma ADC.

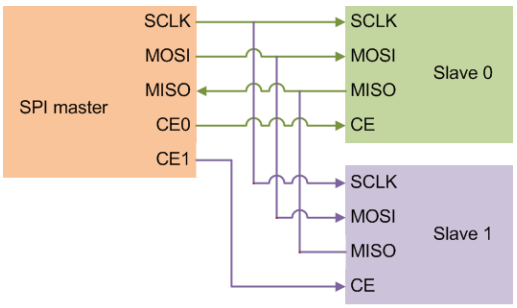
(8) Parallel communication is faster than serial communication for short distances. However, bandwidth of parallel wires is much lower than bandwidth of serial wires for long distances because Inter-symbol interference (ISI) and noise cause corruption over long distances and data may arrive at different times at the receiver (skew).

(9) To double the signal to noise ratio in serial transmission for long distances, differential signals are used.

(10) The advantages of synchronous serial transmission over asynchronous serial transmission are amount of overhead information restricted to few characters for each block and it can be used at higher speeds while the disadvantages are if error were to occur, whole block of data is lost and it requires storage as user cannot transmit characters instantaneously.

(11) Given UART with 3 bits of protocol (start, stop and parity), 7 bits of data, 14400 baud rate, and 2 bits per symbol, the information rate is 14400 * 2 * 7/10 = 20160 bps.

(12) Using SPI interface, a single master is connected to two slaves as follows:



(13) Two advantages of SPI interface are high throughput, simple hardware interface and not limited to any maximum clock speed and two disadvantages are requires more pins, no hardware flow control by the slave, typically supports only one master device and only handles short distances.

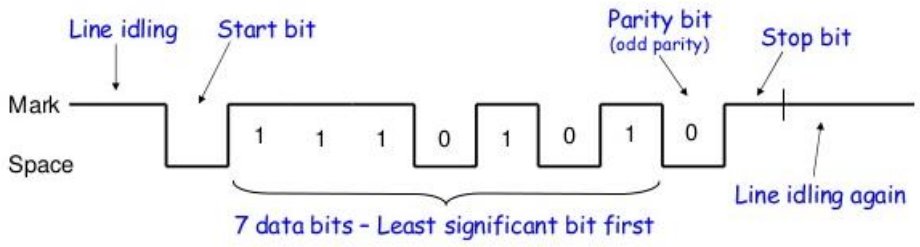
(14) In I2C interface, data line can only change state when the clock is low.

(15) In I2C interface, when a master is reading multiple bytes from a slave, a master informs the slave that a read byte is the last to be read by sending NACK signal.

(16) In I2C interface connecting multiple masters, bus arbitration occurs as follows: each master transmitter checks the level of the data line (SDA) and compares it with the levels it expects; if they do not match, that transmitter has lost arbitration, and drops out of this protocol interaction.

(17) Two advantages of I2C interface are uses only 2 wires interconnect, multi-master capability, incorporates ACK/NACK functionality for improved error handling and two disadvantages are each slave must have an address, imposes protocol overhead that reduces throughput, supports a limited range of speeds, any device having a fault could hang the entire bus and requires pull up resistors.

(18) Using UART, transmitting character 'W'=1010111 with odd parity, 7 data bits and one stop bit, the following data waveform is generated:



(19) In UART, when the stop bit is 0, this causes a framing error.

(20) In LPC176x UART 0 interface, given that bit 0 in line state register (`LPC_UART0->LSR`) is set when the receiver buffer register (`LPC_UART0->RBR`) holds an unused character, the following code is used to receive a character in variable ch:

```
while (!(LPC_UART0->LSR & 1))
ch = LPC_UART0->RBR;
```

(21) In LPC176x UART 0 interface, given that bit 5 in line state register (`LPC_UART0->LSR`) is set when the transmitter holding register (`LPC_UART0->THR`) is empty, the following code is used to transmit a character in variable ch:

```
while (!(LPC_UART0->LSR & (1 << 5)))
LPC_UART0->THR = ch;
```

(22) Real time operating systems solve the main problems of a cooperative multitasking system based on two basic concepts preemption, context switching.

(23) In FreeRTOS, a task can be in one of the following states ready, running, blocked, and suspended.

[19 Points]

(Q2) Assume that the following periodic processes are ready at time 0. We would like to schedule these processes using RMS and EDF scheduling techniques. In each case, compute the schedule for an interval equal to the least-common multiple of the periods of the processes. If processes have the same priority, schedule processes to minimize preemption. If processes have the same priority under all these criteria then give priority in the following order P1, P2, P3.

Process	Execution Time	Period (Deadline)
P1	1.0	3
P2	1.0	4
P3	2.0	6

(a) **(2 points)** Compute CPU utilization.

CPU utilization = 1/3 + 1/4 + 2/6 = 91.67%

(b) **(3 points)** Compute the response time for P3 under RMS. Is there a feasible schedule under RMS? Justify your answer.

$r_3 = 2$

$r_3 = 2 + 1 + 1 = 4$

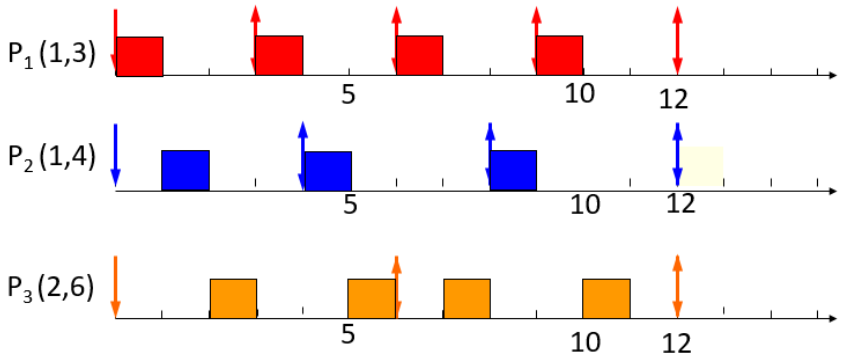
$r_3 = 2 + 2 + 1 = 5$

$r_3 = 2 + 2 + 2 = 6$

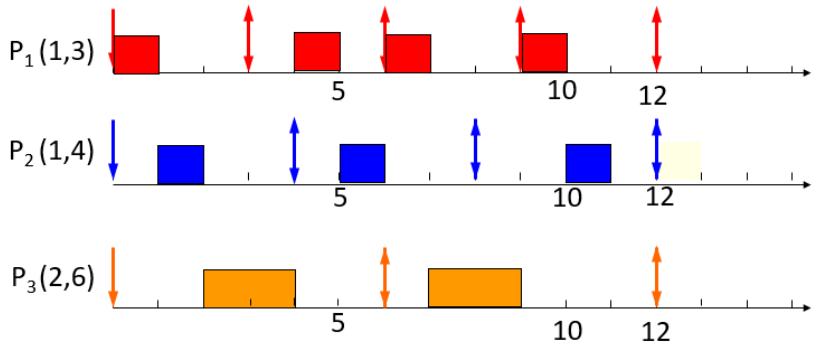
$r_3 = 2 + 2 + 2 = 6$

Thus, the response time for P2 is 6 which means that it will meet its deadline and hence a schedule is feasible under RMS.

(c) **(5 points)** Schedule the processes using an RMS policy. If a schedule does not meet the required deadlines for all processes, complete the schedule and indicate this in your solution.



(d) **(5 points)** Schedule the processes using an EDF policy. If a schedule does not meet the required deadlines for all processes, complete the schedule and indicate this in your solution.



(e) **(4 points)** Is it possible to find a feasible schedule where all deadlines for all processes are met using both RMS and EDF if the execution time of P1 is increased to 1.25? Justify your answer.

CPU utilization = 1.25/3 + 1/4 + 2/6 = 100%

Thus, it is possible to find a feasible schedule under EDF.

Next, we compute the response time for P2.

$r_3 = 2$

$r_3 = 2 + 1.25 + 1 = 4.25$

$r_3 = 2 + 2.5 + 2 = 6.5$

$r_3 = 2 + 3.75 + 2 = 7.75$

$r_3 = 2 + 3.75 + 2 = 7.75$

Thus, the response time for P3 is 7.75>6, which means that it will not meet its deadline and hence a schedule is not feasible under RMS.

[9 Points]

(Q3) Three periodic tasks T1, T2 and T3 are to be executed on a system with task periods 50ms for T1, 200ms for T2 and 1s for T3.

(a) **(6 marks)** It is required to execute the three tasks using a single timer that is configured to expire every 50ms Write a timer interrupt handler (ISR) in C that runs the three tasks according to their periods using a single counter. Assume that each task is run by calling a function of its name, e.g. T1().

```
static int counter = 0;

void timer_handler() {


  counter++;

  T1();

  if (counter %4= 0) {
        T2();
  }

  if (counter == 20) {
        T3();
        counter = 0;
  }

}
```

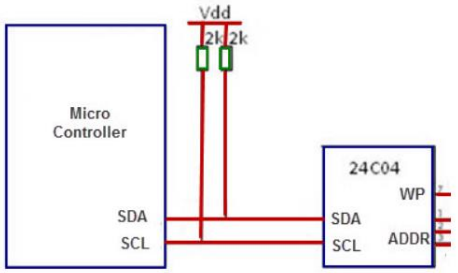
(b) **(3 marks)** Briefly explain how such three periodic tasks could execute using FreeRTOS.

Each task is created with xTaskCreate() API. Each task runs forever within an infinite loop. At the end of each loop iteration, vTaskDelayUntil() API Function is used which moves the task into the Blocked state and specifies the exact tick count value at which the calling task should be moved from the Blocked state into the Ready state relative to the time at which the task left the Blocked state obtained by xTaskGetTickCount().

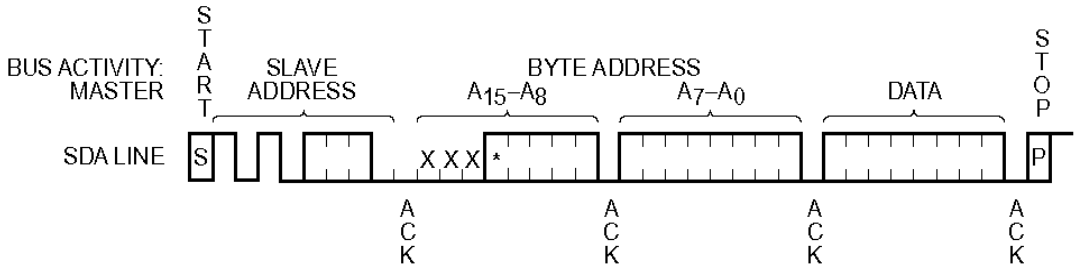
[10 Points]

(Q4) Given 24WC32 EEPROM that is 400 KHz I2C Bus Compatible. The EEPROM is 32KBit memory organised as 4K x 8bit with 12 address bits. The EEPROM device address is 1010111. It supports byte write and page write with a 32-byte write buffer. It also supports immediate/current address reading, selective/random read, and sequential read. Use S and P to indicate a start and stop conditions on the data line.

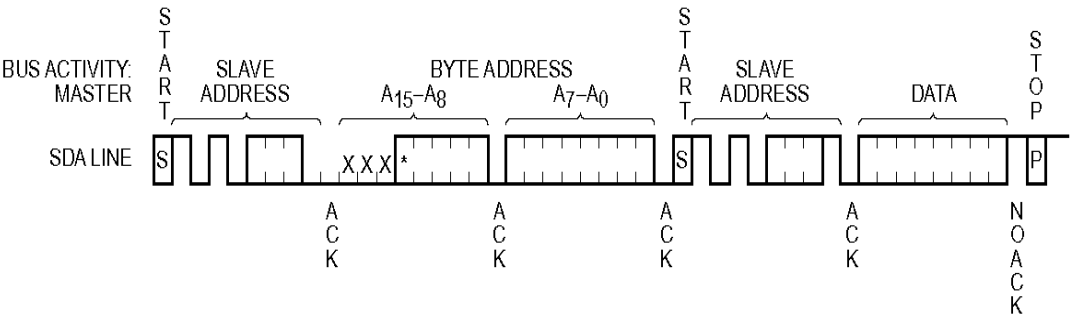
(a) **(2 points)** Show the block diagram interface of the microcontroller with the 24WC32 EEPROM using I2C Bus.



(b) **(4 points)** Show the data waveform for writing a single data byte to a selected address.



(c) **(4 points)** Show the data waveform for reading a single data byte from a specified address location.



[11 Points]

(Q5) Given an analog signal in the range of 0v to +5v, it is required to convert the analog signal value **4.6v** into a digital signal with 4-bit resolution.

(a) **(3 points)** Show the 4-bit converted signal using Flash ADC. Clearly explain your answer.

The Flash ADC will have 15 comparators with the first comparator having a reference value of 15/16*5=4.6875, the second comparator having the reference value of 14/16*5=4.375, and the 15^{th} comparator having the reference value of 1/16*5=0.3125. The input will be compared with all comparators and will be fed to a priority encoder. Since the input value 4.6 is greater than the reference values of the second comparator, the priority encoder will generate the value 1110.

(b) **(4 points)** Show the 4-bit converted signal using Half-Flash ADC with each Flash ADC generating two bits. Clearly explain your answer.

The first Flash ADC will have 3 comparators with reference values 12/16*5=3.75, 8/16*5=2.5, and 4/16*5=1.25. Since the input value 4.6 is greater than the first comparator reference value, the Flash ADC will generate the code 11. Then using a DAC, this value will be subtracted from the analog signal value and will result in the signal 4.6-3.75=0.85. The 2^{nd} Flash ADC will have 3 comparators with reference values 3/16*5=0.9375, 2/16*5=0.625 and 1/16*5=0.3125. The value 0.85 will be compared with the 3 reference values and since it is greater than the 2^{nd} ref. values, the code 10 will be generated. The two codes will be concatenated and the 4-bit value will be 1110.

(c) **(4 points)** Show the 4-bit converted signal using Successive Approximation ADC. Clearly explain your answer.

The 4-bits in the SAR register will have the following weights: 5/2=2.5, 5/4=1.25, 5/8=0.625, 5/16=0.3125. The SAR will be set to the value 1000 and this value will be converted to the analog value 2.5 and will be compared with the input. Since the input value 4.6 is greater than 2.5, this bit will remain set. Then the next bit in the SAR will be set and SAR=1100. This will be converted to the analog value 2.5+1.25=3.75. Since the input value 4.6 is greater than 2.5, this bit will remain set. Then the next bit in the SAR will be set and SAR=1110. This will be converted to the analog value 2.5+1.25+0.625=4.375. Since the input value 4.6 is greater than 4.375, this bit will remain set. Finally, the last bit in the SAR will be set and SAR=1111. This will be converted to the analog value 2.5+1.25+0.625+0.3125=4.6875. Since the input value is less than this value, the last bit in SAR will reset. Thus, the converted value is 1110.

Empty Page