Department of Electrical and Computer Engineering
University of Alabama in Huntsville

CPE 323 – Introduction to Embedded Computer Systems
Final Exam

Instructor: Dr. Aleksandar Milenkovic

Date: May 02, 2012

Place: EB 207

Time: 3:00 PM – 4:45 PM

Note: Work should be performed systematically and neatly. This exam is closed books and closed neighbor(s). Allowable items include exam, pencils, straight edge, calculator, and materials distributed by the instructor. The bonus question is not mandatory. Good luck!

Question	Points	Score
1	20	
2	20	
3	20	
4	20	
5	20	
6	10 (bonus)	
Sum	100+10(b)	

Please print in capitals:

Last name:_____

First name: _____

1. (20 points) MSP430 System Architecture/Miscellaneous

Answer the questions or circle the correct answers when appropriate.

A. (6 points) The MSP430FG4618 device has an address space of 128 KB (0x0000-0x1FFFF). Fill in the table below if we know the following. The first 16 bytes of the address space (starting at the address 0x00000) is reserved for special function registers (IE1, IE2, IFG1, IFG2, etc.), the next 240 bytes is reserved for 8-bit peripheral devices, and the next 256 bytes is reserved for 16-bit peripheral devices. The RAM memory of 8 Kbytes starts at the address 0x01100. Finally, right after the RAM memory starts the flash memory that occupying the rest of address space (the last byte address is 0x1FFFF).

$128 = 2^7 \cdot 2^{10} = 2^{17}$ bytes => 0x0_0000 - 0x1 - FFFF

What	Start Address	End Address
Special Function Registers (16 bytes) 0-16	0x00000	0x0-000F
8-bit peripheral devices (240 bytes) 0-256	0x0-0010	0x0_00FF
16-bit peripheral devices (256 bytes) 0-512	0x0-0100	0x0-01FF
RAM memory (8 Kbytes)	0x01100	0x0 - 30FF
Flash Memory	0x 0_3100	0x1-FFFF

$8kB = 2^{13}$ 10.0000.0000.0000. 1.1111.1111.1111 => 0x1FFF Ram: 0x0-1100 -> 0x0_1FFF

B. (2 points) What is the size of the flash memory (in Kilobytes) in the example above?

128 kB => 115.75 kB 1_FFFF - 0-3100

C. (2 points) (True | False) The MOV.W #34, &0x1000 will actually write the constant 34 into a flash memory location at the address 0x1000.

Cannot write into flash memory using instruction

D. (2 points) (True | False) The program downloaded into the flash memory through a JTAG port will remain in the flash memory only until the power is turned on (i.e., will be lost when the device is turned off).

flash memory is non-volatile

E. (2 points) (True | False) The content of the RAM memory is lost when a system goes into a low-power mode.

Preserves content of memory

F. (2 points) (True | False) The content of the general-purpose and special-purpose registers is preserved while the processor is in a low-power mode.

G. (2 points) (True | False) An instruction is executed to set certain bits in the status register to cause a transition from the active mode into a low-power mode.

H. (2 points) (True | False) An instruction is executed to clear certain bits in the status register to cause a transition from a low-power mode into the active mode.

Low power, no IE in CPU.

2. (20 points) Interrupts Asynchronous, hardware. Priorities fixed, IVT, depends on hw.

Answer the questions or circle the correct answers when appropriate.

A. (2 points) (True | (False)) When a new interrupt request is received, the processor immediately stops processing the currently executing instruction (before finishing it) and starts the exception processing.

Always finish exe.

B. (2 points) ((True) | False) The RETI (return from interrupt) instruction retrieves both the status register and the program counter from the stack.

RetI is unique, vs. RET is emulated.
MOV @SP+, PC

C. (2 points) (True | False) If multiple interrupt requests are pending, the processor selects the one that arrived first.

-> Priority

D. (2 points) (True | False) The GIE (Global Interrupt Enable) bit in the status register can be set inside an interrupt service routine explicitly by the programmer to enable nesting of interrupts.

E. (2 points) (True | False) In case that multiple interrupt request sources share a single entry in the interrupt vector table, programmers need to explicitly clear interrupt flag bits in the corresponding service routine.

Your task is to design an MSP430-based system that can interface 2 external devices (ID1, ID2), each capable of generating an interrupt request. The external devices place a request by setting the request line (a transition from a logic zero to a logic one). This line is kept active as long as the interrupt request is pending, until the request is serviced. Answer the following questions.

F. (2 points) How would you connect the interrupt request lines coming from the devices to the MSP430 platform?

ID1 request => P2.BIT0
ID2 request => P2.BIT1
P2IE => 0000.0011 (P2 interruptenable)
P2IFG initially cleared
P2IES (0->1)
GIE Bit should be set.

G. (2 points) In a section of the program you would like to service requests coming from the device ID1 but not the requests coming from the device ID2. What would you do to support this option?

Use selective masking; P2IE = 0000.0001 (enable ID1, disable ID2)

H. (2 points) Outline the main steps that need to be taken in the main program to initialize the system for accepting the interrupts from the devices ID1 and ID2.

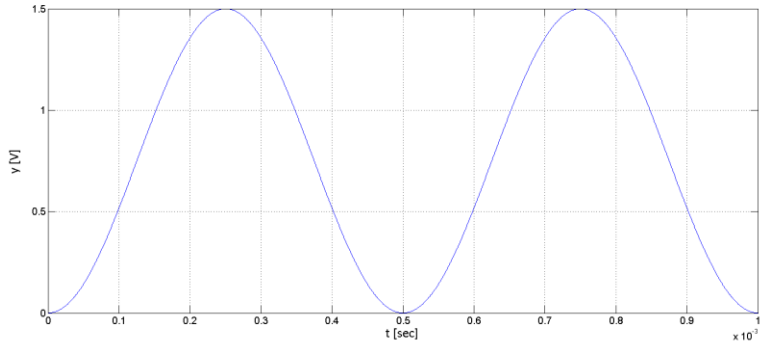
P2IE => 0000.0011 (P2 interruptenable)
P2IFG initially cleared
P2IES (0->1)
GIE Bit should be set.

I. (4 points) Outline the main steps that need to be taken inside the ISR for the interrupts that come from the devices. If both requests are pending concurrently, ID1 is processed first. Only one request can be processed at a time in the ISR. Outline the key steps in the main program.

```
P2_ISR:        // one ISR for P2
IF (ID1 pending) then { Call ID1 service();
                        Clear P2IFG[0]; }
else if (ID2 pending) then { Call ID2 service();
                             Clear P2IFG[0]; }
```

3. (20 points) Embedded Software Design (Digital to Analog Conversion).

To drive a sensor you need to provide the following wave signal y, $y = 1.5*\sin^2(x)$, as shown below (x=2*pi*f*t, f=1000 Hz, t is time in seconds). To accomplish this task we will use the MSP430's digital-to-analog converter (DAC12) peripheral and a lookup table prepared in memory, similarly to the laboratory exercise.



A. (2 points) What is the maximum and minimum output voltage at the analog output? What is the reference voltage you would use to generate the required signal?

$V_{R+} = 1.5V$ $\quad y_{max} = 1.5V$

$V_{R-} = -1.5V$ $\quad y_{min} = 0V$

B. (2 points) What is duration of one period of the wave signal y in miliseconds?

$T = 0.5ms$

C. (4 points) Let us assume that we provide a lookup table with 10 samples for a single period of the signal y? Fill in the following table with the first 5 samples in the table (sample #1 has been provided in the table).

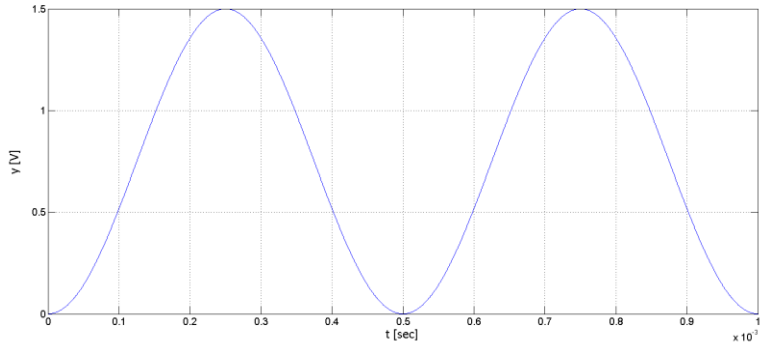
sin(2·Pi·1000·.005E-3) $\quad$ int(4095x(.143/1.5))

Sample	*t=? [sec]*	*sin(2*pi*f*t)*	*y=? [Volts]*	*Lookup table*	
				decimal	*hex*
1	**0**	**0**	**0**	**0**	**0x0000**
2	0.05ms	.309	.143	390	
3	0.10ms	.588	.818	1404	
4	0.15ms	.809	.982	2680	
5	0.2 ms	.951	1.358	3707	

Radians

$T = 0.5ms \cdot 10SPS \Rightarrow dt = T/10 = 0.05ms$

D. (2 points) Sketch the output analog signal for the first 0.5 ms (as you would see it on the oscilloscope).



E. (2 points) If we use a TimerA ISR to control sending the next value to the DAC12, how many interrupts per second TimerA should generate?

10 samples X2,000 = 20,000

F. (2 points) If $f_{MCLK}=f_{SMCLK}$= 1 MHz, how would you configure TimerA?

dT = .05ms

1,000,000 / 20,000 = 50

TA is configured in UP mode TACCR0 = 49

G. (2 points) You wrote the TimerA ISR that reads an entry in the lookup table, increments the table pointer and checks its range, and sends the value to the DAC12. By profiling you determined that the ISR requires 20 processor clock cycles? Is it possible to implement the sine wave signal generator under these conditions? Elaborate your answer?

Budget: ISR is 50cc assuming that CPU mclk is also 1 MHz.)

Required 20cc for ISR

H. (4 points) How would you improve this design? Elaborate your answer. What needs to be changed in the code?

1. Ref voltages are fine 0 -> 1.5v => no room for improvement

2. Increase number of samples in look up table dT = 0.025ms.

4. (20 points) Time, Timers

Answer the questions or circle the correct answers when appropriate.

A. (2 points) (True | False) The MSP430's clock module can internally generate clocks used by the processor and peripherals or use external crystal oscillators.

B. (2 points) (True | False) MCLK, SMCLK, and ACLK are fixed and cannot be changed in software.

C. (2 points) (True | False) To prevent a controlled reset of the system, the watchdog timer in the watchdog mode requires a hardware signal to clear its control bit before the time period expires.

D. (2 points) (True | False) TimerA can be configured to generate pulse-width modulated signals without intervention from software (excluding initialization stage).

E. (2 points) What does it involve capturing an external event in TimerA (what happens inside TimerA)?

Let us consider a program that displays time on the LCD of the TI's Experimenter platform. The time is displayed with resolution of one tenth of a second (100 ms, deciseconds). The processor wakes up periodically, updates local time variables, performs necessary format conversions, and displays the time on an LCD in the following format (hh.mm.ss.d). These steps take N=2,000 processor clock cycles during which the processor is in the active mode; after that the processor goes back into a low-power mode. The MCLK in the active mode is 1 MHz. The platform draws 1 mA when the MSP430 is in the active mode and 0.1 mA when in the low-power mode.

F. (3 points) What is the total active and sleep time during one application cycle? Calculate the average current drawn by the platform.

G. (3 points) Calculate the total power P consumed (in milliWatts) if we power the platform by two AA batteries (V=3 V). Determine the system operating time in days if we know that the battery capacity is 2500 mAh.

H. (4 points) Outline briefly organization of the program that displays time (list peripherals needed, initialization, main loop, interrupt service routines).

5. (20 points) Communication

Answer the questions or circle the correct answers when appropriate.

A. (2 points) Two devices A and B are communicating using a bidirectional asynchronous serial link. How many signals need to be routed between the devices excluding the common ground?

B. (2 points) What is the total number of bits transferred when a single character is sent from device A to device B using the asynchronous serial link? Assume the following: 8-bit character, parity is on, 2 stop bits.

C. (2 points) How long does it take in seconds to transfer the character from above, assuming that the devices are configured for 57,600 bits per second?

D. (2 points) What happens when a new character is received in the receive buffer before the previous one is read by the software?

E. (2 points) What does this interrupt service routine do?

```
#pragma vector=USCIAB0RX_VECTOR
__interrupt void USCIA0RX_ISR (void)
{
  while(!(IFG2&UCA0TXIFG));  //
  UCA0TXBUF = UCA0RXBUF;     //
}
```

F. (4 points) Two devices C and D are communicating using a serial peripheral interface (SPI) link. How many signals need to be routed between the devices excluding the common ground? Sketch the connection between the devices, including signal names.

G. (4 points) Two devices E and F are communicating using a parallel bidirectional link with 8 data lines (half-duplex link, i.e., E and F cannot send data at the same time). In addition to data lines, what other control signals are needed to implement proper data communication? Sketch the connections.

H. (4 points) What needs to be done in software during initialization and what during data communication assuming the following protocol: E starts exchange by sending one byte and F responds back by sending one byte.

Device E	*Device F*

6. (10 points) BONUS QUESTION, DMA, LCD

Answer the questions or circle the correct answers when appropriate.

A. (2 points) (True | False) DMA controller can be initialized to perform a transfer of a block of data from one place in memory to another place in memory.

B. (3 points) List DMA controller registers that enable transfer of a block of data and explain their function.

C. (3 points) Could you use a DMA controller to transfer 128 samples coming from the ADC12 controller to a buffer in RAM memory? If yes, explain what steps would you take to configure the DMA properly? Use illustrations. How would you know when the task has been completed?

D. (2 points, bonus) What segments need to be ON to display a digit 5 on an LCD display?