Université d'Ottawa
Faculté de genie

École d'ingénierie et de technologies de l'information

University of Ottawa
Faculty of Engineering

School of Information Technology and Engineering

uOttawa

L'Université canadienne
Canada's university

COURSE: CEG3136/CEG3536
SEMESTER: Fall 2009

PROFESSOR: Gilbert Arbez
DATE: Dec 18, 2009
TIME: 14h00 – 17h00

Computer Architecture II

FINAL EXAM

`

NAME and STUDENT NUMBER:_______________________________/__________________

Instructions:

- Answer ALL questions on the examination paper.
- This is a close-book examination**.**
- Use the provided space to answer the following questions. If more space is needed, use the back of the page.
- Show all your calculations to obtain full marks.
- Calculators are allowed.
- Read all the questions carefully before you start.

There are three (3) parts in this exam.

Part 1	**Short Answer (Theory)**	**20 marks**	
Part 2	**Assembler Programming**	**25 marks**	
Part 3	**An Alarm System**	**45 marks**	
Total		**90 marks**	

Total Pages: 11

Part 1 Short Answer Questions (Theory) (total 20 points)

Circle the correct answer or give a short answer to the following questions.

1. (2 points) Interrupts are masked by default during the execution of an ISR (interrupt service routine).

a) True b) False

2. (2 points) Which of the following indicates to the HCS12 CPU where a specific ISR (interrupt service routine) is stored in memory?

a) The $\overline{\text{IRQ}}$ input pin b) The Reset vector

c) The Program Counter (PC) d) The interrupt vector

3. (2 points) What is the difference between the inputs $\overline{\text{IRQ}}$ and $\overline{\text{XIRQ}}$?

4. (4 points) The following list contains certain operations performed by the CPU to service an interrupt. Determine the correct order of the operations and fill out the table with their corresponding letters to show this order (e.g., a b c d e f???).

a) The return address is stored on the stack.
b) The address of the ISR (interrupt service routine) is assigned to the program counter (PC).
c) The CPU continues the execution of the program interrupted by the interrupt.
d) The ISR is executed by the CPU.
e) The CPU tests the value of the interrupt mask bit I.
f) The return address is loaded into the Program Counter (PC).

1	
2	
3	
4	
5	
6	

5. (2 points) Select the instruction that clears the flag bit C7F (the most significant bit of the timer status register TFLG1):

a) bclr TFLG1,$80	b) movb 127, TFLG1
c) bclr TFLG1,127	d) movb %10000000,TFLG1

6. (4 points) Consider a signal, containing frequences from 10 kHz to 23 kHz (10000 to 23000 Hz), applied to a pin on the HCS12 ATD port. Assume that this signal is converted to an 8-bit binary value using the SCAN mode and that 8 cycles is used for the 2nd phase of the sampling time. Determine and select an ATD clock frequency available in the HCS12 ATD module to digitize this signal. Assume a 24 MHz system clock.

7. (2 points) What is largest value that can be reached by the timer's TCNT register?

8. (2 points) Is is possible to read the contents of the SCI Transmit Data Register? Explain your answer.

Part 2 – Assembler Programming (total 25 points)

a) (18 points) The following .lst file gives a sequence of CPU instructions for an HCS12 microcontroller. Give the content of the indicated registers (in hexadecimal) after the execution of the corresponding instruction (on the same line). (2 points per value for a total of 18 points).

```
     =00002000                     ORG $2000
2000                       main:
2000 CF 1500                       lds #$1500
2003 96 7F                         ldaa $7f
2005 CE 2500                       ldx #BCDCODES
2008 16 200C                       jsr convertBCD   PC = __________
200B 3F                            swi

   ; Subroutine: convertBCD
   ; Parameters: hexValue - in accumulator A
   ;             bcdAddr - in register Y
   ; Converts the hexadecimal value, hexValue, into 3 binary
   ; coded decimal values, to store at address provided by
   ; bcdAddr.  Registers used in the subroutine are preserved.
200C                       convertBCD:
200C 34                            PSHX
200D 3B                            PSHD             SP = __________
200E 180E                          TAB
2010 87                            CLRA
2011 CE 000A                       LDX #10         X = __________
2014 1810                          IDIV
2016 6B 42                         STAB 2,Y        Y = __________

2018 B7C5                          XGDX     D = _________ X = _________
201A CE 000A                       LDX #10
201D 1810                          IDIV
201F 6B 41                         STAB 1,Y        B = __________
2021 B7C5                          XGDX
2023 6B 40                         STAB 0,Y
2025 3A                            PULD             SP = __________
2026 30                            PULX
2027 3D                            RTS              PC = __________

     =00002500                     ORG $2500
2500 +0003                 BCDCODES DS.B 3   ; for storing BCD digits
```

b) (7 points) Give the contents (hexadecimal values) of the following memory locations **after** the execution of the above code:

$2500: __________

$2501: __________

$2502: __________

Part 3 – An Alarm System (45 points)

This part of the exam develops software to emulate an alarm system using the Dragon-12 card. The bank of 8 DIP switches is used to represent door and window contacts. LEDs and an additional DIP switch represent the alarm panel. Finally a speaker represents the alarm siren. The developpement of the alarm system is completed using three separate questions which reflect a modular design. The development is completed using C programming.

<u>Question 3.1 (15 points) – Door and Window Monitor Module</u>

Figure 1 shows how 8 DIP switches, used to emulate the door and window contacts, are connected to the HCS12 microcontroller Port H on the Dragon-12 card. The figure also shows how the 8 LEDs are connected to Port B.

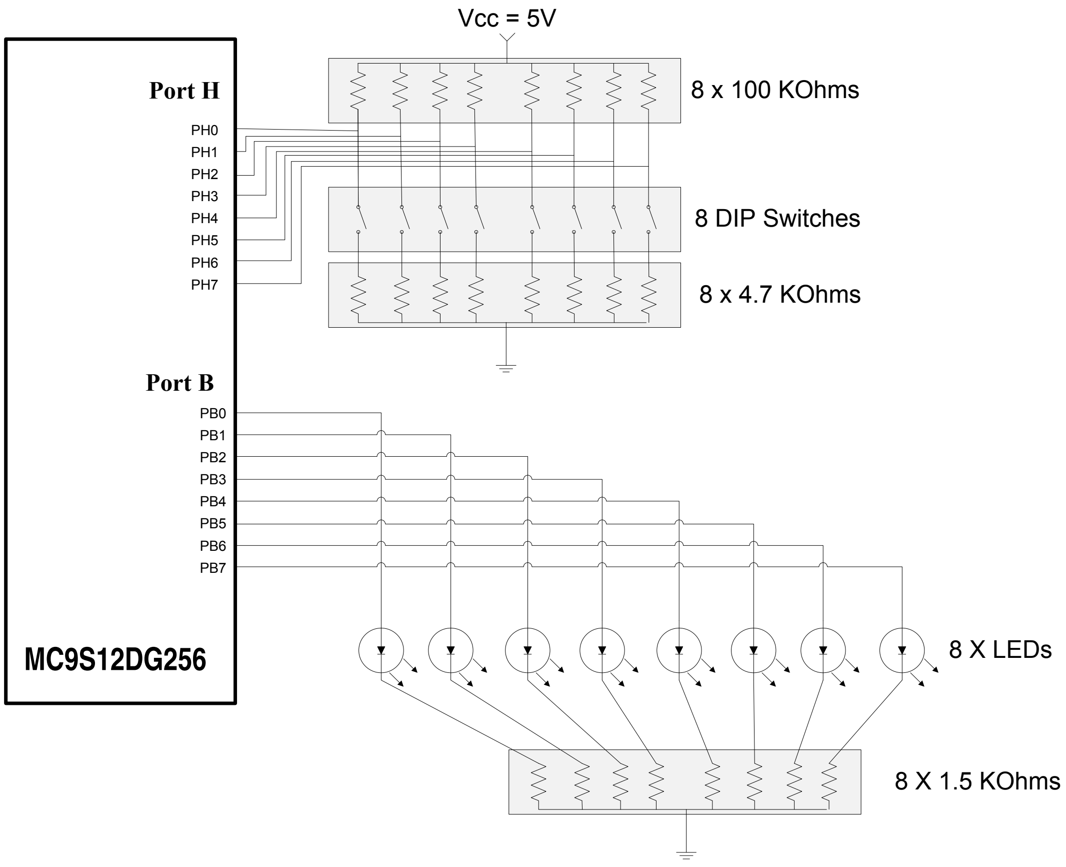


Figure 1

A. Briefly explain the purpose of the resistors in the circuit used to connect the DIP switches.

B. Briefly explain the purpose of the resistors in the circuit used to connect the LEDs.

C. Software Design: Door/Window Monitor Module

The purpose of the module is to

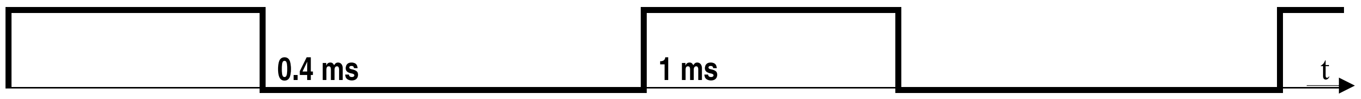
- Reflect the status of the door/window contacts on the LEDs; when a switch is open, a corresponding LED is turned ON. The LED connected to PB0 reflects the status of the switch connected to PH0, LED connect to PB1 reflects the status of the switch connected to PH1, etc.
- Set a global variable, `alarmStatus`, to TRUE (value 1) when one of the switches is open (i.e. a door or window is open) and FALSE (value 0) when all switches are closed.
- Monitor (see the above two points) doors and windows (i.e. DIP switches) using an ISR triggered by timer channel 0 every 100 ms.

(a) Develop an initialisation function, `initDoorWindowMonitor()`, that setups up PORT B, PORT H and Timer channel 0 (note that for Timer Channel 0, configure only the channel, **not** the general functions of the Timer – this shall be done later; you may assume a Timer clock frequence of 187.5 kHz, i.e. a clock cycle of 5 1/3 μsec).

(b) Develop the ISR, `doorWindowMonitorISR()`.

Question 3.2 (15 points) – Siren Module

A speaker is connected to pin 5 (PT5) of the Timer port (controlled by timer channel 5). Generating a 1 kHz signal rectangular wave-shape with a 40% duty cycle (40% of the period, the signal is at 5V as shown below) on PT5 have the speaker produce a sound emulates an alarm siren.



The siren module provides the means to turn on and off the siren (i.e. control the 1 kHz signal to the speaker). Interrupts are used to generate the waveform continuously when the siren is turned on.

A. Develop an initialisation function, `initSiren()`, to setup timer channel 5. (configure only the channel, not the general functions of the Timer – this shall be done later; you may assume a Timer clock frequence of 187.5 KHz, i.e. a clock cycle of 5 1/3 μsec).

B. Develop an the function, `turnOnSiren()`, to turn on the siren (i.e. apply a signal to the speaker).

C. Develop an the function, `turnOffSiren()`, to turn on the siren (i.e. remove a signal from the speaker).

D. Develop the ISR, `sirenISR()`.

<u>Question 3.3 (15 points) Alarm Module</u>

Figure 2 below shows the complete "Alarm System". Note that an additional DIP switch has been added to pin 0 of PORT A. The switch is debounced in hardware. See Figure 1 of Question 3.1 for details on how the DIP switches are connected to PORT H and how the LEDs are connected to PORT B.

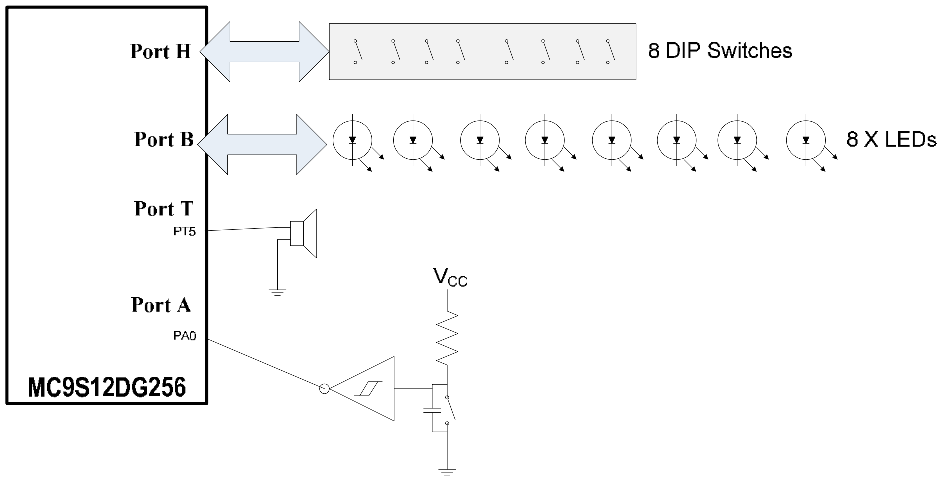


Figure 2

The Alarm Module is the main module that controls the overall operation of the alarm system. Initially the system is disarmed; the LEDs still show the status of the 8 DIP switches and the siren is off. The alarm system should function as follows:

- The 8 LEDs always reflect the status of the 8 DIP switches connected to port H, as specified in question 3.1.
- To arm the system, the switch connected to PORT A is closed.
- When the system is armed, if one of the switches connected to PORT H (i.e. a window or door) is opened, the siren (i.e. speaker) should be turned on (used the Siren module developped in question 3.2). The siren should be maintained even if the switch is closed, that is, the only way to turn off the siren once turned on is to disarm the system.
- To disarm the system, open the switch connected to PORT A.

A. Develop the main intialisation function, `initMain()`, that intialises the global functions in the Timer (not the specific channels) and PORT A, and that calls the initialisation functions developed in questions 3.1 and 3.2. Include any additional configuration that you feel may be necessary.

B. Develop the function `main()` that implements the logic of the alarm system as described above.