EE2004: Microcomputer Systems
Test 2 Review

Topics

III. PIC18 Assembly Language Programming

3.1

(a) Terminology
- Machine vs. assembly language
- 2 types of assembly language statements: instructions and directives
- 4 elements of an assembly language statement:
    - label
    - mnemonics
    - operands
    - comments

(b) Know functions of common directives: org, set, equ, cblock

(c) Fetching and execution in PIC18
- Understand each step of my animation

3.2

(a) Know instructions used for subtracting unsigned number

(b) Know microcontroller does not subtract; it adds a negative number; negative number is represented by the 2's complement format

(c) Be able to determine all status flags in an addition operation
- OV: would not occur when +ve added by a –ve number
- If +ve added by a +ve results in an –ve number *or* –ve added by a –ve results in a +ve number, then OV must have occurred.

(d) BCD addition
- Under two conditions, you would get an incorrect BCD addition result
- Use daw to adjust

(e) Compare instruction: Make branching decision based on the value in a file register

3.3

(f) Looping
- Initialization
- Statements to repeat
- monitor number of iterations/repetitions

(g) Conditional jumps
- make branching decision based on status flag
- encode destination by relative address
- relative address is 8-bit, ranging from -128 to 127

(h) goto
- encode destination by absolute address
- last bit of address not encoded; only most significant 20 bits are encoded.

(i) bra
- encode destination by relative address
- relative address is 11-bit, ranging from -1024 to 1023

(j) Know how absolute/relative addresses are encoded in machine code.

(k) Branch instruction timing
- Conditional jumping: 2 instruction cycles if jump, 1 if not jump
- Unconditional jumping always takes 2 instruction cycles.
- Conditional skipping: 1 instruction cycle if not jump, 2 or 3 if jump.

(l) Application of nested loop in generating time delay
- Calculate the total time delay generated by the loop with different values of [DELAY_H] and [DELAY_L]. See Week 6 tutorial.

3.4

(m) Know difference between branching and subroutine calling
- After calling a subroutine, program counter needs to get back to the main program.
- Return address must be stored in <u>hardware stack</u> before executing the subroutine

(n) List the events that occur when <u>calling</u> and <u>returning from</u> a subroutine

(o) Know how the instruction call encodes absolute address of destination and the instruction rcall encodes relative address of destination.

(p) Know how to determine the contents of the hardware stack, TOS and STKPTR at all points of a program, and in particular immediately after the execution of a call/rcall or return instruction.

- Be prepared to answer a question similar to the tutorial question.

3.5
(q) Understand why we need register indirect addressing mode.
(r) Know the auto-increment options

3.6
(s) Know how to implement lookup table using the computed goto/retlw method
(t) Know the disadvantage of the computed goto/retlw method
(u) Know how to implement lookup table using the table read operation (Know the 4 steps)

IV. I/O Programming

(a) Code assembly language to use the ports for input or output

(b) Code I/O bit manipulation programs for the PIC

(c) Check the state of an I/O port and make branching decision based on it

(d) Know algorithm to interface with 1-digit/4-digit 7-segment LED and keypad matrix

V. Timer

(a) Know timers' function in generating time delay (i.e., counting internal instruction cycles). Know time delay equations for each timer:
- Time delays below are expressed in instruction cycles.
- Timer0/1 16-bit mode
  - Convert the initial value of TMRx into its decimal equivalence NNNNN first.
  - Time Delay = (65536 – NNNNN) x Prescaler
- Timer0 8-bit mode
  - Time Delay = (256 – NNN) x Prescaler
- Timer 2
  - Time Delay = (PR2+1) x Prescaler x Postscaler

(b) Know how to set control registers in order to manipulate Prescaler and Postscaler. Data sheet will be provided in the exam.

(c) Know timers' function in counting rising/falling edges of external signal.

(d) Know how to set control registers in order to use timers for counting rising/falling edges of external signal. Data sheet will be provided in the exam.

(e) Know properties of Timers 0, 1, 2:

<u>**Timer0**</u>	<u>**Timer1**</u>	<u>**Timer2**</u>
8-bit or 16-bit timer or counter.	16-bit timer or counter.	8-bit timer. Cannot be used as counter
Supports prescaling factors of 1, 2, 4, 8, 16, 32, 64, 128, 256.	Supports prescaling factors of 1, 2, 4 and 8.	Supports prescaling and postscaling factors.
Timer0 counts clock pulses fed into PORTA.4	Timer1 counts clock pulses fed into PORTC.0	Cannot be used as counter
When counter rolls over, TMR0IF in INTCON SFR is raised	When counter rolls over, TMR1IF in PIR1 SFR is raised	When counter rolls over, TMR2IF in PIR1 SFR is raised

VI. <u>Interrupt Programming</u>

(a) Know advantages of use of interrupt over polling.

(b) Explain each component of the block diagrams for the two scenerios: (i) interrupt priority scheme is used, (ii) interrupt priority scheme is NOT used.

(c) Know what would occur if Interrupt B is activated when the ISR of Interrupt A is being executed.

(d) Know how to write an interrupt-based program. Decide what should be included in the Main program and ISR. Understand the example interrupt-based programs thoroughly.

(e) Know where the interrupt flags, enable and priority bits of each interrupt source are located by looking at the data sheet.

VII. Serial Communication

(a) Know the disadvantages of parallel I/O and how serial I/O addresses these disadvantages.

(b) Define simplex, half duplex and full duplex communications.

(c) Know the difference between synchronous and asynchronous serial communications. Know how synchronous and asynchronous communications determine sampling time. Know which one is suitable for long-distance communication.

7.1 UART

(d) Baud rate

- Know how to control the baud rate in the UART module implemented in PIC18.
- Actual baud rate almost always deviates from desired baud rate. Why?
- What is the error tolerance?

(e) Be prepared to describe the workflow of UART transmission in PIC18. Which registers are involved? Which bits are required to be set to enable transmission? (Information about UART registers will be provided in the exam.) Which flags are involved in the transmission? How does a programmer know a complete byte has been shifted out?

(f) Be prepared to describe the workflow of UART reception in PIC18. Which registers are involved? Which bits are required to be set to enable reception? How does a programmer know a complete byte has been received?

(g) What is the framing error? When would it occur?

(h) Be prepared to describe the overflow error. How does it happen? Which registers are involved? After the overflow error has occurred, how does a programmer re-enable reception?

7.2 I^2C

(i) Know that I^2C classifies devices involved into two roles: Master and Slave. Know what the master is responsible for.

(j) The I^2C protocol accommodates multiple masters. Describe why Start and Stop conditions are required in this setting.

(k) Describe the role of the control byte.

(l) Describe the motivation of introducing the repeated start condition (i.e., Why Start and Stop conditions are not enough?).

(m) Be prepared to describe the workflow of I^2C transmission in PIC18. How does a programmer initiate transmission? (Information about I^2C registers will be provided in the exam.) Which flags are involved in the transmission? How does a programmer know a complete byte has been shifted out?

(n) Answer the same set of questions for I^2C reception.

(o) Given the timing diagrams, prepare to describe the sequence of events that occur during the Byte Write, Page Write, Byte Read and Sequential Read operations.