98/100 ⇒ 29/30 97/100

UNCA CSCI 255
Exam ~~2~~ Fall 2011

This is a closed book and closed notes exam. Laptops, cell phones, and any other electronic storage or communication devices may not be used during this exam.

Name: Joshua Lancaster, Joshua Lancaster

If you want partial credit for imperfect answers, explain the reason for your answer!

Problem 1 (20 points) True/False Questions

1. (**true** or false) The PIC24 data memory is byte addressable. but program memory isn't
2. (true or **false**) The PIC24 has 24 working registers, W0-W24. W0-W15, only 16
3. (**true** or false) A PIC24 instruction can be stored in 24 bits. — even though some take longer than 1 clock, the instruction to do this only takes 24-bits.
4. (**true** or false) The PIC24 data bus is 16 bits wide.
5. (**true** or false) The *program counter* stores an instruction. stores all of the instructions doesn't it? −2
6. (true or **false**) The PIC24 instruction cycle is the same as its clock cycle---1 instruction is executed each clock cycle. Some instructions take more than one clock to comp.
7. (true or **false**) The *ALU* is the PIC24 *controller*. ALU does bit-wise ops., arithmetic, comparison upon request. it doesn't "control" the PIC24.
8. (**true** or false) The *stack pointer* stores a memory address.
9. (**true** or false) All PIC24 working registers are 16 bits wide. 16 total regs, 16-bits wide, 16x16
10. (**true** or false) Working register 15 (W15) is the PIC24 *stack pointer*.

Problem 2 (25 points) Short Answer Questions

1. Briefly, contrast machine code and assembly code.

   Machine code is in binary and assembly code is a set of mnemonics, or word-like, representations of these binary sequences.

2. How many bits are required to address 8M bytes of memory if a byte is the smallest addressable unit?

   $M = 2^{20}$, $8 = 2^3$

   $2^3 \cdot 2^{20} = 2^{23}$ $2^x = 2^3 \cdot 2^{20}$ $2^x = 2^{23}$, $x = 23$

   / this seems right to me

   23 bits

3. If a memory has 100 words, and each word contains 16 bits, how many bytes are contained in this memory?

   $\frac{100 \cdot 16}{8} = 200$ bytes

−7

4. Explain why the stack is needed to implement subroutines.

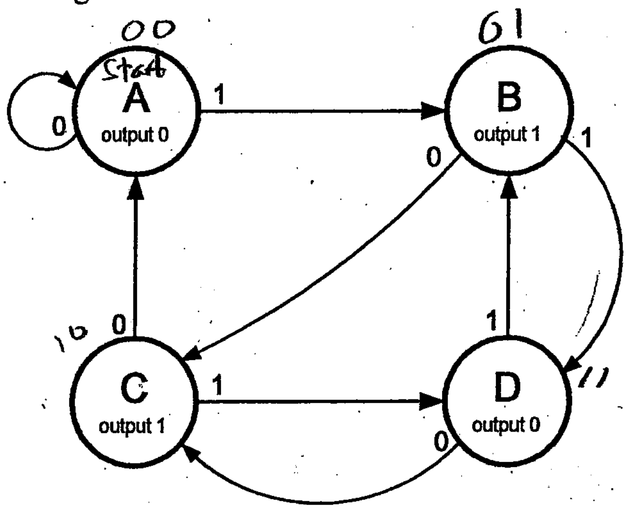
The stack is needed to implement a stack frame which is used to hold each part of a subroutine. Namely the memory address for the return, so it goes back to the right place.

5. Explain the difference between `MOV [W1++],W0` and `MOV [++W1],W0`.

The first increments AFTER the move is executed. The second does the increment BEFOR. This inc is done on the data located in the mem. address, stored in W1, not what is stored in W1.

Problem 3 (10 points) Finite State Machines

A finite state diagram is given below. This FSM has four states, labeled A to D. At each state transition, it receives a single input bit and produces a single output bit (the output is shown inside the state). The starting state for the FSM is A.



1. Suppose the FSM is started (in state A) and then receives three successive inputs 0 1 1. After the 3[rd] input bit is received, which state is the FSM in and what is its output?

   0 @ D or 11 — 1

2. In the table below give a two-bit state encoding for the four states of the FSM. In other words, represent each state with a unique 2-bit pattern. There are many possible correct answers for this subproblem.

State as letter	State as two bits	
A	0	0
B	0	1
C	1	0
D	1	1

3. Complete the following table to show the state transitions of the FSM at the top of the page using your state encodings from the previous subproblem.

Joshua Lancaster
Joshua Lancaster

present state		input	next state	
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	0
1	0	1	1	1
1	1	0	1	0
1	1	1	0	1

✓

Problem 4 (20 points) Translating C to Assembly
Using the PIC24 assembly language instruction summary attached to this test, write a PIC24 instruction sequence that accomplishes each of the C statements below. Assume that *i, j*, and *k* are 16-bit variables that have already been allocated in memory. In other words, assume that the following assembly code has already been written:

```
        .bss            ;data section
i:      .space 2        ;Allocating space (in bytes) to variable.
j:      .space 2        ;Allocating space (in bytes) to variable.
k:      .space 2        ;Allocating space (in bytes) to variable.
```

1. k = (i – j) >> 2

```
MOV j, WREG
sub i, WREG
LSR W0, #2, W0
MOV W0, k
```

✓ good

2. if (i == j) {
   if-body statements
   }

```
if_top:
MOV j, WREG
CP i
BRA NZ, end_if
;stuff
end_if:
```

✓ good

3. while (i != 0) {
   loop-body statements
   }

```
while_top:
MOV i
BRA Z, while_end
;Stuff
BRA while_top
```

✓ good

0

Problem 5 (15 points) Register and Memory Contents
Assume the memory/register contents shown below at the start of the instruction in each subproblem. For each subproblem, state the memory location or register that is modified along with its new contents.

Location	Value	Location	Value
W0	0x080A	0x0800	0x382A
W1	0x0808	0x0802	0xFB80
W2	0x0804	0x0804	0x4D19
W3	0x0800	0x0806	0xE7C0
W4	0x0806	0x0808	0xFF00
W15	0x0802	0x080A	0x0808

1. MOV 0x0802, WREG

W0 = 0xFB80 ✓

2. MOV #0x0802, W0

W0 = 0x0802 ✓

3. MOV [W3], [W4]

0x0806 = 0x382A ✓ good

4. MOV W3, W4

W4 = 0x8000 ✓

5. PUSH W1

MOV W1, [W15++] ...
W15 = 0x0804 ✓
0x0802 = 0x0808
great!

Problem 6 (10 points) C Expressions
Evaluate each of the following expressions assuming *i*, and *j* have been declared as follows:
uint16_t i=1, j=0;

Expression	Evaluation
!i	0, false —
j & 0x0001	0x0000 = 0 —
j && 0x0001	0, false —
j | 0x0001	0x0001 = 1 —
j || 0x0001	0x0001 = 1, or true —