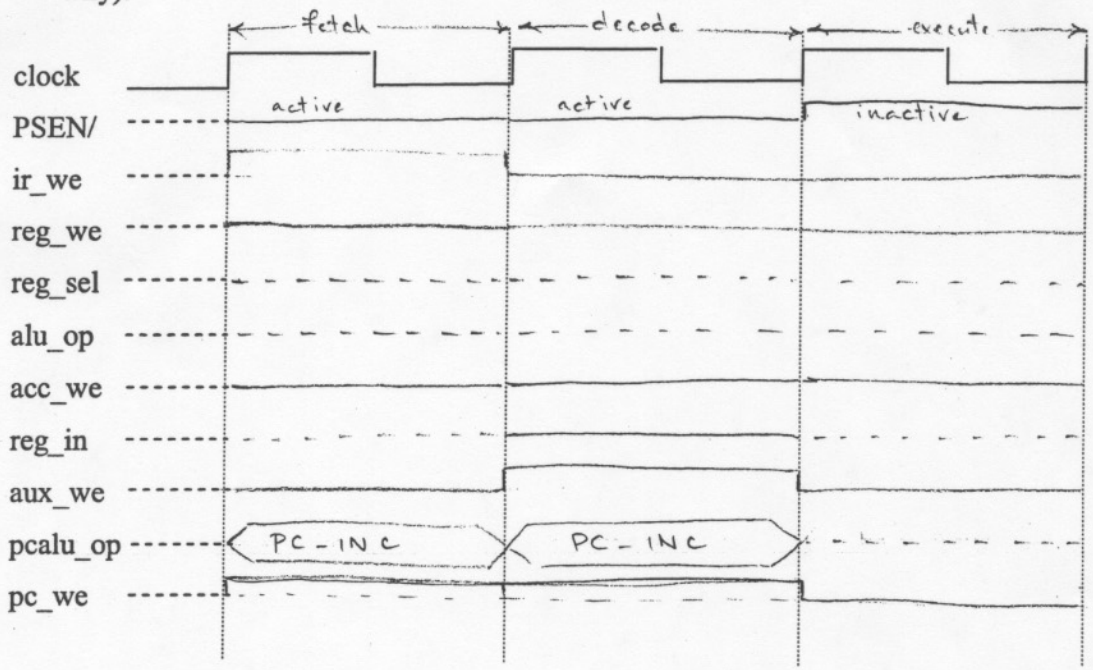
CpE213 Assignment 3 (due Thursday Oct. 2 at 11 am)

Name Solution

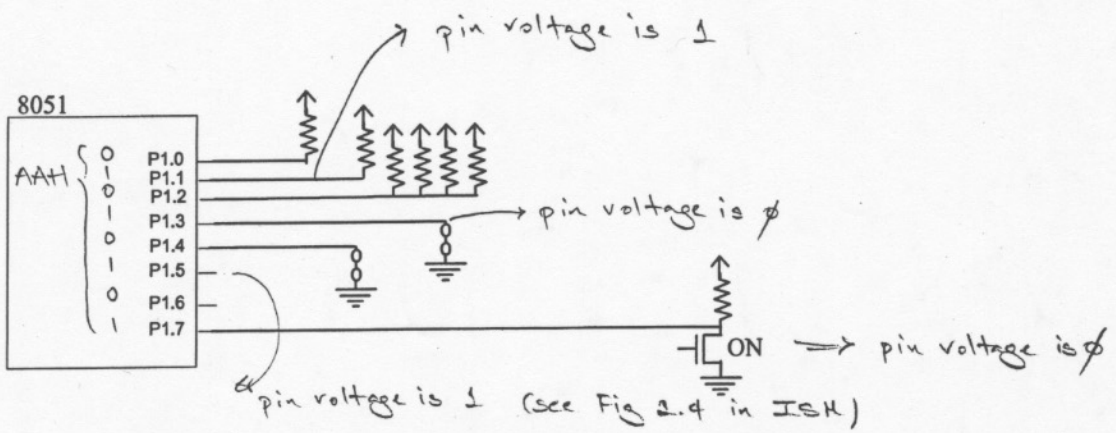
Show all your work in the space provided. Answers with a simple "yes", "no", or a single number are typically incomplete and will not be given full credit. Answers in non-reduced form, like (a+sqrt(b))/c, are fine where appropriate. Good English on essay/short answer questions is required. ON MULTIPLE CHOICE QUESTIONS, IF YOU'RE NOT SURE DON'T GUESS – you will get points off for wrong answers. If you know part of an answer, write what you know for partial credit.

1. (10 Points) Choose between the underlined words. Circle the correct answer. Write an explanation if you think you need to.
   a. For an embedded system using a microcontroller, code is typically stored in: RAM (ROM) BOTH.
   b. Data is typically stored in: (RAM) ROM BOTH.
   c. Registers are relatively ~~CHEAP~~ (EXPENSIVE). Chip designers use them LIBERALLY (CONSERVATIVELY).
   d. Pins coming off the chip are relatively CHEAP (EXPENSIVE). Chip designers use them LIBERALLY (CONSERVATIVELY).
   e. There are potentially two values that can be read from the 8051's internal memory location 90H, depending on how you read it. (TRUE) FALSE.

2. (20 Points) Draw the timing diagram for the following control signals when the WIMP51 is executing the instruction "JZ 42H", assuming ACC=2AH. If you don't think a control signal is doing anything important, indicate so with a "don't care" (a dotted line down the middle). Give "command" values for pcalu_up and alu_op rather than numeric values (e.g. give value as "pcalu_up=PC_INC" instead of "pcalu_up=42"). If you're not sure, make your best guess (but tell me it's a guess and why).



ACC = 2AH => Z = 0 => JZ does not branch.



3. (20 Points) The schematic above shows some external hardware connected to the 8051 microcontroller (P1.0 is bit 0 of port 1). Say we performed the following instructions:

   MOV P1,#AAH AAH = 10101010

   MOV A, P1

   a) (8 Points) What value would end up in the ACC (Assume resistors are large).

   22H

   b) (12 Points) Explain your answer for a)… please explain carefully, as though you were trying to teach someone who knew nothing about the 8051 microcontroller how this thing worked (you will be graded by how well they might understand).

MOV P1, #AAH writes AAH to the port.

MOV A, P1 reads the contents of the port into ACC.

To be able to read the contents of any pin on the port, we need to configure that pin as an input pin by writing a '1' to the pin.

Writing AAH to the port configures pins P1.7, P1.5, P1.3, and P1.1 as input pins.

This means we will be able to read the pin voltage (0 or V_{cc}) from these pins. The other 4 pins, which have had ø written to them, will yield ø when read, regardless of the pin voltage.

Reading from P1, we will get:

P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.ø
ø	ø	1	ø	ø	ø	1	ø

Hence, ACC will have the value 22H after both instructions are executed!

4. (25 Points) For the following WIMP51 code:
   a. (10 Points) Give the opcode for the JZ instruction (You will need to fill in some addresses below to do this… but not many).

60 02 H 9 + rel = B => rel = 02

   b. (5 Points) Give the opcode for the XRL instruction.

01101000 = 68 H

   c. (10 Points) Fill in the values of the WIMP51 registers listed below as it executes this code. I have already filled in some of the values for you.

addr.	ASM code	
0		MOV A,#42H
2		MOV R0,A
3		MOV A, #2AH
5	top:	XRL A, R0
6		MOV R0,A
7		JZ stop
9		SJMP top
B	stop:	SJMP stop

registers:

PC	ACC	Z	R0	R1	
0	X	X	X	X	
2	42	X	X	X	
3	42	X	42	X	
5	2A	X	42	X	
6	68	X	42	X	← Ready to do MOV instruction

PC	ACC	Z	R0	R1
0	X	X	X	X
2	42	0	X	X
3	42	0	42	X
5	2A	0	42	X
6	68	0	42	X
7	68	0	68	X
9	68	0	68	X
5	68	0	68	X
6	00	1	68	X
7	00	1	--	-

2A = 0010 1010
42 = 0100 0010
XOR = 01101000 = 68

PC	ACC	Z	R0	R1
B	00	1	00	X

5. (25 Points) For the components in the following diagram

   a. (10 Points) Give the address space of RAM0.

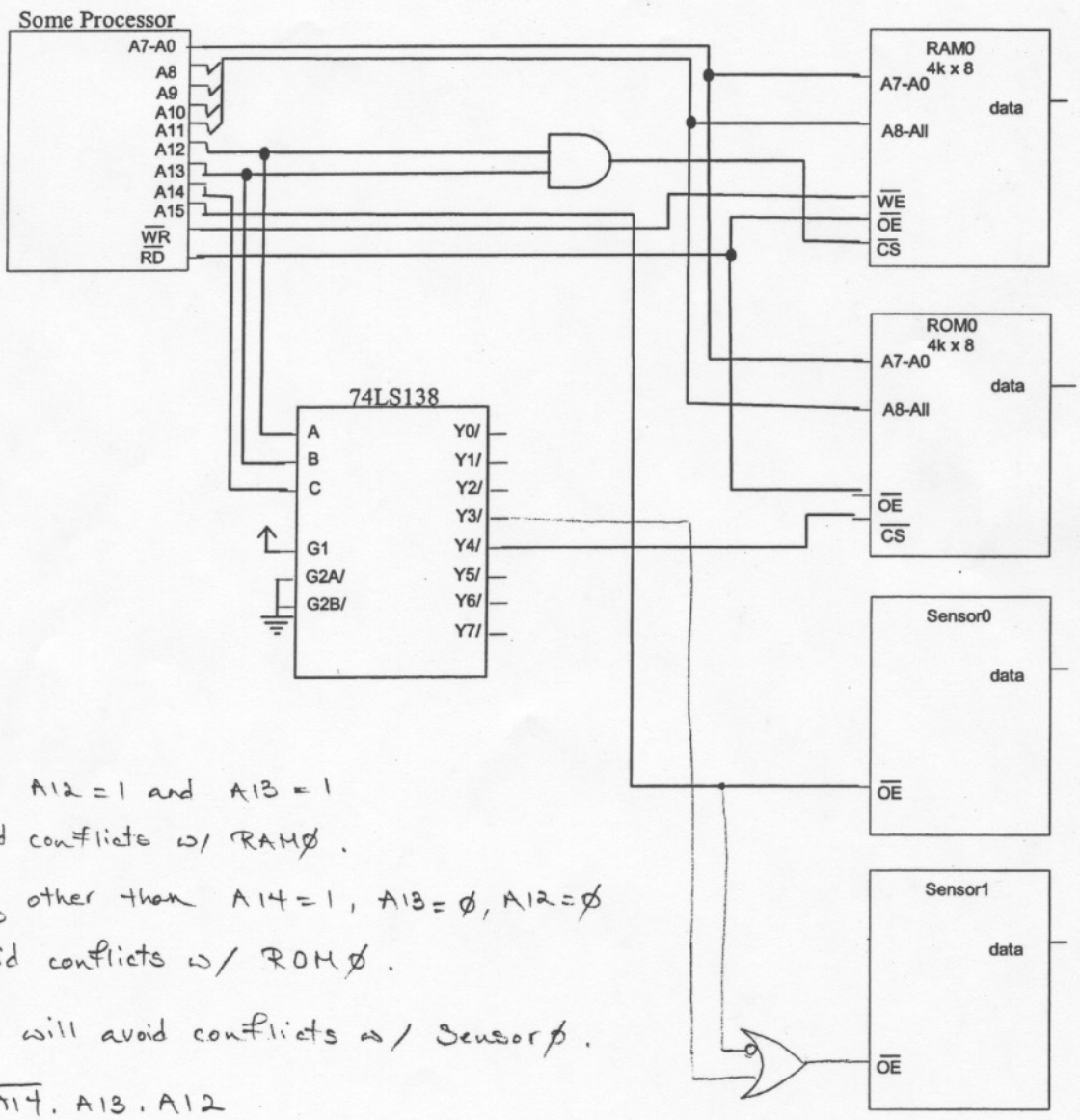
A15	A14	A13	A12	A11 ... A0
x	x	0	0	000 - FFF
x	x	0	1	
x	x	1	0	

   b. (5 Points) Name 2 address locations the processor could read from to access Sensor0.

A15 = 0, A14 ... A0 = anything

Any address with A15 as the first bit is a valid answer.

   c. (10 Points) Connect register Sensor1 to the processor, such that its address space does not conflict with the address space of any of the other devices. Feel free to add additional logic.



c)

We need A12 = 1 and A13 = 1 to avoid conflicts w/ RAM0.

Anything other than A14 = 1, A13 = 0, A12 = 0 will avoid conflicts w/ ROM0.

A15 = 1 will avoid conflicts w/ Sensor0.

$\overline{y3} = \overline{A14} \cdot A13 \cdot A12$

$\overline{OE} = \overline{y3} + \overline{A15}$