CmpE213 – Digital Systems Design

Homework 2

The 8051 - internal and external hardware

1. In problem 6 of the last homework, you drew a timing diagram of control signals in the Gnome when executing a ”sta Rn” instruction, but did not draw the values on the address or data lines or within internal registers. Draw a timing diagram of the fetch, decode, and execution of the instruction ”sta 5”, showing the binary or hex values on/in the: PC, IR, ACC, addr bus, data bus, clk, OE/, and WR/ lines. Assume that before beginning the fetch, the PC contains the value 42 (where the ”sta 5” instruction is located) and the ACC contains the value 0x0F.

2. Ch2, question 3.

3. Ch2, question 8. In a few sentences (and maybe a small sketch), explain why. (PS. I love to ask this question on tests).

4. Explain, briefly, the technique the Gnome uses to separate code and data space. Is it possible for the two spaces to overlap – that you might accidently overwrite your code when writing data? Why not? How does the Gnomes technique differ from the 8051's technique of separating code and data? Will the 8051's technique also prevent overlapping of the two spaces?

5. Ch2, question 10. Draw a schematic to show it.

6. Why is it possible that after the following code sequence:
   ```
   SETB P1.0
   MOV C,P1.0
   ```
   that C may contain a zero? (SETB P1.0 sets bit zero of port 1. MOV C,P1.0 moves the value at port 1,bit 0 into C, the carry flag). Explain your answer. If we had written a zero to P1.0 instead (CLR P1.0), could we ever read a one into C with the MOV C,P1.0 instruction?