Lec-5-HW-1-digitalLogic

Reading: PP, Chapter 3:

3.1 (transistors)
3.2 (OR, NOR, AND, NAND, and DeMorgan's Law)
3.3 (DEC, MUX, FA, PLA)
3.4 (R-S latch, register)
3.5 (memory)
3.6 (sequential machines, FSM)

Problems:

3.1 (n-type vs. p-type transistors)
3.2 (cmos inverter)
3.6 (as 3.5, but tricky ckt.)
3.9 (expression to truth-table),
3.10 (NOR truth-table),
3.17a (4-bit input, 7 1's as output)

Problem 1: Implement the function from 3.17a as a minterm AND-OR circuit. Use only 2-input devices and NOT.

3.12 (3-Dec, show minterm exp.)
3.13 (5-dec, how many output lines?)
3.14 (16X1 mux, how many select lines?)
3.22 (build 4X1 mux from 2x1 MUXes
3.25c (how many gate delays for ADD?)
3.29 (d-latch transparency)
3.31 (#word X word_size = mem. Size)
3.32 (addressability vs. address)
3.33a (memory addressing: find 4-th word)
3.33b (how many address bits for 60 words?)
3.33c (how many words max if PC has _ bits?)
3.43a (FSM truth table)
3.43b (state-transition diagram for (a))

Problem 2: Design a soda machine's FSM controller. It has a coin slot that accepts nickels, dimes, and quarters. To start the machine, you push the "go" button, and each time you put in a coin, you push the button again. The controller changes state each time the button is pushed: The button serves as the FSM's clock signal. When the controller reaches any state such that at least 35 cents has been put in, it dispenses a soda and change and halts in that state. (There is no choice about what soda you get.) Show the controller's FSM state-transition diagram. Indicate which are halting states by labeling them with an "H". Assume the machine is always in one of its halting states before the next person comes to buy a soda, and it enters its start state when that person pushes the "go" button for the first time.