University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Science

EECS 150 R. H. Katz
Fall 2000

Solution Set # 6

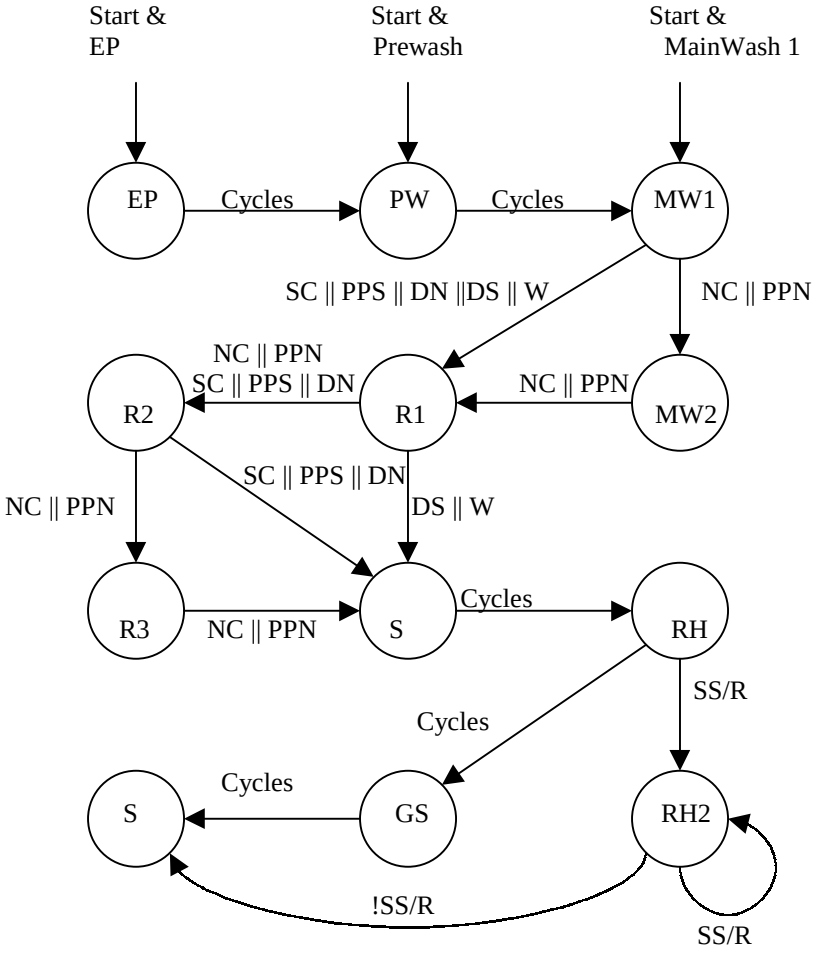
4. (a) Identify your inputs, outputs, and name and describe your states.

Inputs: Start, Extra Prewash, Prewash, Main Wash 1, Cotton Normal, Permanent Press Normal, Permanent Press Short, Delicates Normal, Delicates Short, Woolens, Short Spin/Rinse

Outputs: Same as states

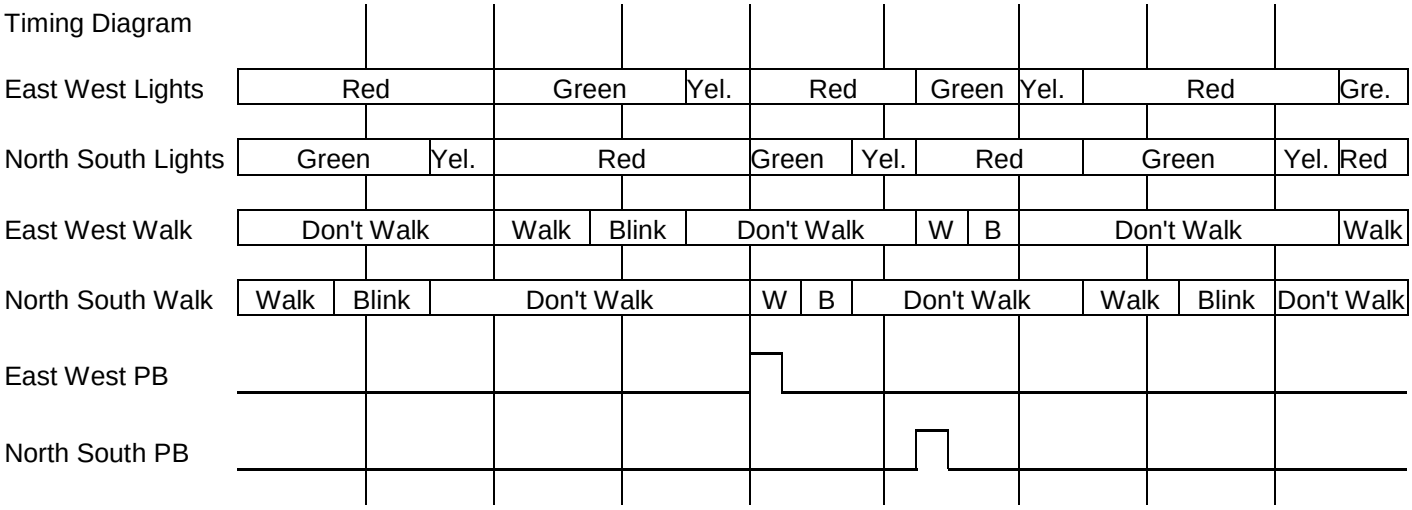
States: Extra Prewash, Prewash, Main Wash 1, Main Wash 2, Rinse 1, Rinse 2, Rinse 3, Starch, Rinse Hold, Graduated Spin, Spin

(b) Draw a symbolic state diagram for your design, labeling all state transitions. Write down any additional assumptions you are making.



Cycles = NC, PPN, SC, PPS, DN, DS, W

4. Consider the following variation on the classical traffic light controller problem. A North-South road intersects an East-West road. In addition to the Red/Yellow/Green traffic lights, each of the four corners of the intersection have *Walk*/*Don't Walk* signs facing N-S and E-W. In addition, there are push buttons for impatient pedestrians to force a quicker cycle of the lights so they may cross the street (separate buttons for N-S and E-W crossings). The *Walk*/*Don't Walk* signs cycle for the N-S road as follows. When the traffic lights turn green in the N-S direction, *Walk* is illuminated. Half way through the green light time, the *Don't Walk* sign becomes illuminated and starts to blink (*Walk* is no longer illuminated). When the traffic light turns yellow, the *Don't Walk* sign remains solidly illuminated (i.e., no blinking). It stays this way throughout the red light cycle. The behavior is the same in the E-W direction. The Pedestrian Button works as follows. If the facing light is red, and the button is pushed, the remaining green light time of the cross street is reduced by half (e.g., shift the time left by one bit to the right). Note that despite popular myth, it doesn't matter how many times this button is pushed! Further note that the Yellow light time on the cross street must never reduced (otherwise expect serious accidents!).
   (a) Assume that the standard Green light time is 56 seconds, the Yellow time is 8 seconds, and the Red time is 64 seconds. Draw a simple timing chart that shows the behavior of the N-S and E-W traffic lights and *Walk*/*Don't Walk* lights. Show a complete cycle with and without a pedestrian button press.



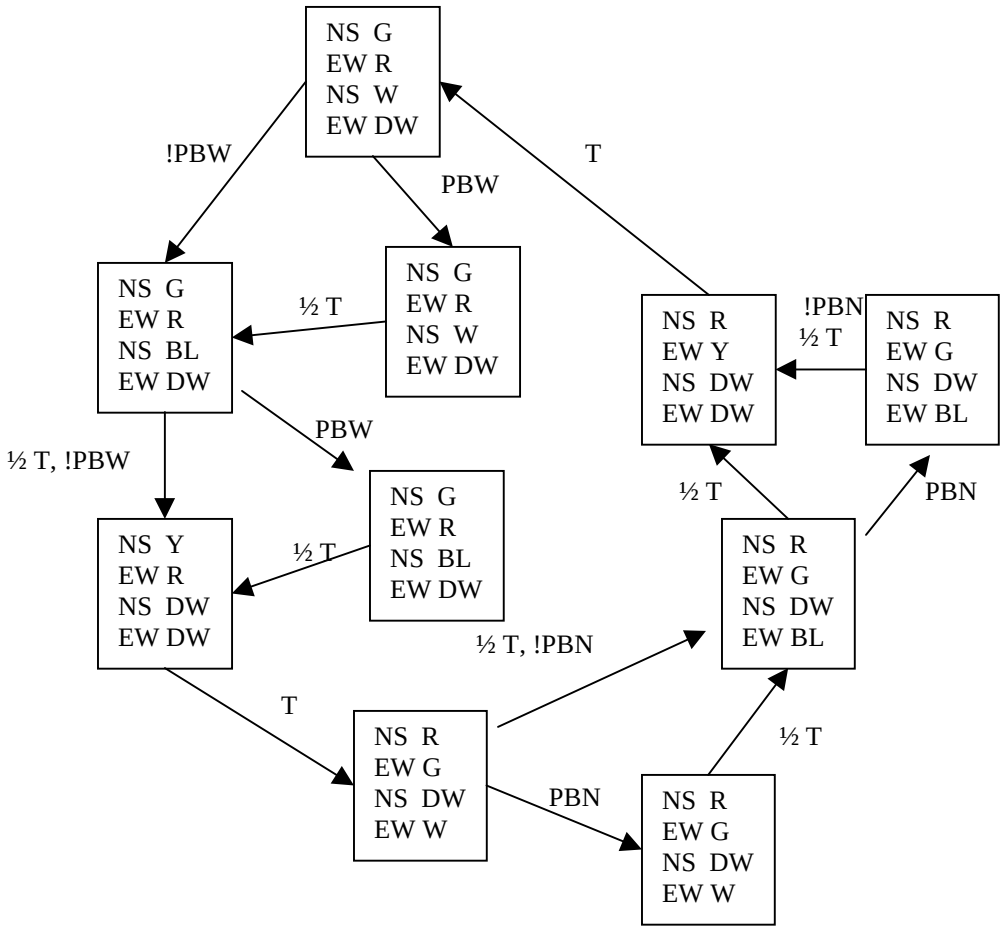
(b) Identify your inputs and outputs. Think about issues like whether the state machine needs to see a normal time interval or a half time interval, as well as the need to reset the push button inputs once they have affected the state machine. What additional circuitry, like timers and flipflops, do you need outside of the state machine?

Inputs: Pedestrian Button NS, Pedestrian Button EW, timers, shifter for PB push

Outputs: NS Light – Green , Red, Yellow
EW Light – Green, Red, Yellow
NS Walk – Walk, Blink, Don't Walk
EW Walk – Walk, Blink, Don't Walk

Need timers to keep track of duration of lights
flip flop to keep state of pedestrian button so that only one push will affect green light time,
shifter to divide green light time in half

(c) a symbolic state diagram. Make clear all of your assumptions about the problem consistent with the specification above.



(d) Choose a state encoding. What is the rationale for your choice?

I'd choose one- hot encoding because it would be simpler to implement. Binary encoding would reduce the number of flip flops from 9 to 3, but it is more complex than one – hot.