1. (a) Give the basic definition of state equivalency in a completely specified FSM (this definition involves only the equality of outputs).

(b) For the FSM shown below, without using any minimization method (e.g., partitioning method, implication table method), determine which state pairs are equivalent and give the proof of equivalency of one of these state pairs (the proofs for the other equivalent pairs, if any, will follow by symmetry for this case) from first principles using the basic definition of state equivalency given above. Give your proof clearly, methodically and convincingly. Note that we have done similar proofs “informally” when discussing the implication table method.

![ FSM Diagram ]
2. The next-state (NS) combinational logic for an edge-triggered synchronous sequential circuit is shown below.

![Next-State Logic: External inputs: A, B, C; Current state bits: Q0, Q1; FF excitation inputs: x, y]

The maximum and minimum delay for each gate are \( k \) and \( 3k \) nsecs (ns), respectively, where \( k \) is the number of inputs of a gate. The sequential circuit uses edge-triggered D-FFs. After clock routing, the maximum clock skew is 7 ns. The delay and other timing data for a D-FF are as follows:

- \( T_{\text{setup}} = 12 \text{ ns} \)
- \( T_{\text{hold}} = 5 \text{ ns} \)
- Typical \( T_{\text{ph}} = 10 \text{ ns} \)
- \( \min T_{\text{ph}} = 7 \text{ ns} \)
- Typical \( T_{\text{ph}} = 14 \text{ ns} \)
- \( \min T_{\text{ph}} = 9 \text{ ns} \)

(a) Is the skew at a safe level? Why or why not? If not, then how would you change the circuit structure (note that you cannot change its functionality) in an incremental way so that the given clock skew is safe (note that you cannot decrease the clock skew amount, as we assume here that a minimized clock skew has been obtained).

(b) Draw the complete sequential circuit by augmenting the above NS combinational logic as (and if) required for the skew to be safe, and connecting the needed D-FFs to the inputs and outputs of the combinational logic.

(c) Keeping a 10% safety margin, determine the minimum clock period for operating the sequential circuit correctly.

3. 12.3
4. 12.6
5. 12.8 plus determine a minimal set of test vectors for covering all faults in the fault table. As mentioned in class, this is done in a similar manner as the PIT stage of the single-function QM method—think of the test vectors as PIs in QM, faults as MTs in QM, and the fault table as the PI table in QM. In other words, this is a classical min-cost covering problem, as is the PIT part of QM. The minimal test vector set problem is actually slightly simpler than the PIT part of the QM method for gate-based digital circuits, as all test vectors have cost = 1.