# Term 051

## COE 561 Digital System Design and Synthesis

## HW# 6

**Q.1.** Consider the incompletely-specified FSM represented by the following state table:

| Input | Present State | Next State | Output |
|---|---|---|---|
| 0 | S1 | S4 | 0 |
| 1 | S1 | S5 | 1 |
| 0 | S2 | S2 | 0 |
| 1 | S2 | S1 | 1 |
| 0 | S3 | S3 | 0 |
| 1 | S3 | S4 | * |
| 0 | S4 | S6 | 0 |
| 1 | S4 | S5 | 1 |
| 0 | S5 | S6 | 1 |
| 1 | S5 | S3 | 0 |
| 0 | S6 | S1 | * |
| 1 | S6 | S5 | 1 |

**(i)** Replace the don't-care conditions by 0's and minimize the obtained state stable.

**(ii)** Replace the don't-care conditions by 1's and minimize the obtained state stable.

**(iii)** Minimize the state table of the incompletely specified FSM.

**Q.2.** Consider the incompletely-specified FSM represented by the following state table:

| Input | Present State | Next State | Output |
|---|---|---|---|
| 0 | S1 | S3 | 1 |
| 1 | S1 | S5 | * |
| 0 | S2 | S3 | * |
| 1 | S2 | S5 | 1 |
| 0 | S3 | S2 | 0 |
| 1 | S3 | S1 | 1 |
| 0 | S4 | S4 | 0 |
| 1 | S4 | S5 | 1 |
| 0 | S5 | S1 | 0 |
| 1 | S5 | S4 | 1 |

**(i)** Perform symbolic minimization of the state table using implicant merging, covering and disjunctive relations. Represent the encoding constraints resulting from symbolic minimization in matrices.

**(ii)** Show all the seed dichotomies and prime dichotomies satisfying the constraints.

**(iii)** Find a minimum cover of seeds by primes. Derives the states codes based on the solution obtained.

**(iv)** Using K-MAP, obtain the equations for the output and flip-flops. Compare your solution to the solution obtained by running the SIS command ***stg_to_network*** using the state codes obtained in (iii).

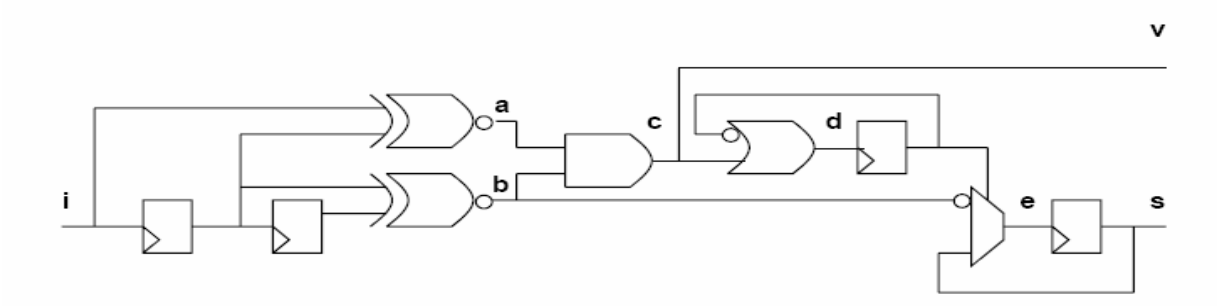
**(v)** Perform state assignment using the program nova by running the SIS command ***state_assign nova***. Compare the obtained solution to your solution in (iv) in terms of number of literals.

**(vi)** Perform state assignment using the one-hot encoding by running the SIS command ***state_assign jedi –e h***. Compare the obtained solution to your solutions in (iv)-(v) in terms of number of literals.

**(vii)** Derive the weighted complete state graph based on the **fanout oriented** algorithm. Then, assign the state codes to minimize the cost function $\sum_{i=1}^{N_s}\sum_{j=i+1}^{N_s} Weight(v_i,v_j) * Dist(v_i,v_j)$. Perform state assignment using the SIS command ***state_assign jedi –e o***. Compare the obtained solution to the solution you obtained.

**(viii)** Derive the weighted complete state graph based on the **fanin oriented** algorithm. Then, assign the state codes to minimize the cost function $\sum_{i=1}^{N_s}\sum_{j=i+1}^{N_s} Weight(v_i,v_j) * Dist(v_i,v_j)$. Perform state assignment using the SIS command ***state_assign jedi –e i***. Compare the obtained solution to the solution you obtained.

**Q.3.** Consider the synchronous sequential circuit given below-specified FSM represented by the following state table:




**(i)** Draw the synchronous network graph corresponding to the given circuit. In the graph, vertices represent inputs, outputs, gates, and fanout stems.

**(ii)** Assume that delay of each gate is 1. Retime the graph to reduce the cycle time. Draw the resulting circuit.

**(iii)** Read the library **synch.genlib** using the command **read_library synch.genlib**. Then, map your design to the library using the command **map –s**. Then, retime the circuit using the command **retime**. Compare the maximum arrival time before and after retiming. Compare the obtained solution to the solution you obtained in (ii).