# MAJOR

## EEL324 Digital Hardware Design

**Time: 2 Hour** **Max. Marks: 40**

**N. B.:-** This is **open notes examination. Transfer** of notes to each other is strictly **prohibited**. Write in the answer script.

**Q 1:-** For the following state assignment and transition table (Of synchronous sequential machine) construct excitation and output maps for **D flip-flop**. Draw the diagrm (using D flip-flop) of resulting circuit.

**(4)**

| $y_1y_2$ | $(y_1y_2)^+$,z<br>x=0 x=1 |
|---|---|
| A→00 | 01,0 00,0 |
| B→01 | 01,0 10,0 |
| D→11 | 01,0 10,1 |
| C→10 | 11,0 00,0 |

**Q 2:-** Draw flow table for the asynchronous circuit that has two inputs (**x** and **y**) and an output (**z**). If **x** and **y** agree (**x = y**), then **z = x = y**. Otherwise **z** retains the last value at which **x** and **y** agreed.

**(3)**

**Q3 (a):-** Why and how hardware-software co-design implemented? In your words (Less than **60** words **strictly**) highlight the main points.

**(b):-** In your words discuss **issues** and their **remedies** in energy efficient design methodology. (**Preferably use tabular form with less than 60 words**) **(3+3)**

**Q4:** If there are two functions '**f**' and '**g**' as given below:

$f = x + y.z'$ $g = y + z.r$ (where $z'$ is complement of **z**)

The variable ordering is **x < y < z < r**. Compute the function **f. g** using **ITE** operator. Show all steps and ROBDD's of original and resultant function. **(5)**

**Q5 (a):** Sketch schematics for the following constructs. Unless specified other wise, assume signals of the type **std._logic** or **std_logic_vector** of appropriate size.

```
IF  a OR b  THEN
        Z <= X;
ELSEIF  a AND b  THEN
        Z <= NOT X;
ELSE
        Z <= Y;
ENDIF;
```

**(b):** Write a short VHDL description of a positive edge-triggered 'D' flip-flop using a VHDL process.

**(2+3)**

**Q6:** A Moore machine has two inputs $x$ and $y$, and an output $z$. The machine starts in an *idle* state in which z = 0. When in the idle state, it processes its inputs as follows:

| *xy* | action |
|---|---|
| 00 | remain idle |
| 01 | assert $z$ for 1 clock cycle, and then return idle |
| 10 | assert $z$ for 2 clock cycles, and then return idle |
| 11 | assert $z$ for 3 clock cycles, and then return idle |

When the machine is not in the **idle state**, it disregards the value of the inputs. Draw the state transition diagram for this machine using the least number of states, and implement using 1-hot encoding.

**(6)**

**Q7:** If the collision vector of nonlinear pipeline is as:

**Collision vector: C = {1 0 1 1 0 0 0 1}**

Find greedy cycles, cycles and MAL after drawing the state diagram. (**Right most is smallest element of vector)**

**(4)**

**Q8 (a):-** For the following Petri net (as given in Fig. 1.3.5), construct the coverability-tree. Also if possible draw reachibility-tree for the same Petri net. **(4)**

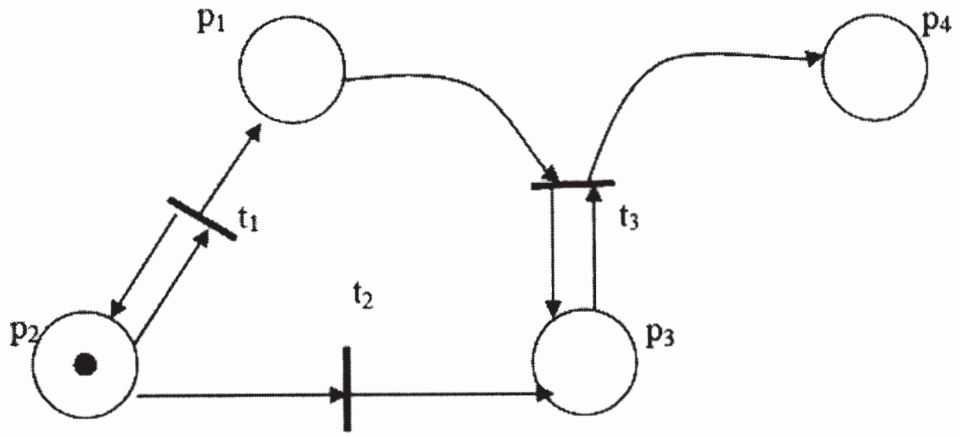


Fig.1.3.5 Petri net with initial setting <0,1,0,0>.

**(b):-** High light main features of Petri net as compared to others. It's strong and week points in **less than 30 words only**.

**(3)**