CS-417
COMPUTER SYSTEMS MODELING

Spring Semester 2020

Batch: 2016-17

(LECTURE # 30)

FAKHRA AFTAB
LECTURER
DEPARTMENT OF COMPUTER & INFORMATION SYSTEMS ENGINEERING
NED UNIVERSITY OF ENGINEERING & TECHNOLOGY

Recap of Lecture # 29

Modeling of CS via Petri-Nets (Mutual Exclusion, Logical Conditions, Conflict & Concurrency)
Reachability & Reversibility in Petri-Nets
Deadlocked Petri-Net
Properties of Petri-Net

Chapter # 8 (Cont'd)

PETRI NET-BASED PERFORMANCE MODELING

Example Problem 1

Consider the Petri-Network in Fig 25.

a) Provide the sets of places, transitions & initial marking.
b) Provide the sets of all the input and output functions w.r.t the transitions given in the figure.
c) Suppose at time T_1, transition t_1 is able to fire. What would be the marking M_1?
d) Identify M_2, M_3, M_4 and M_5 at times T_2, T_3 and T_4.

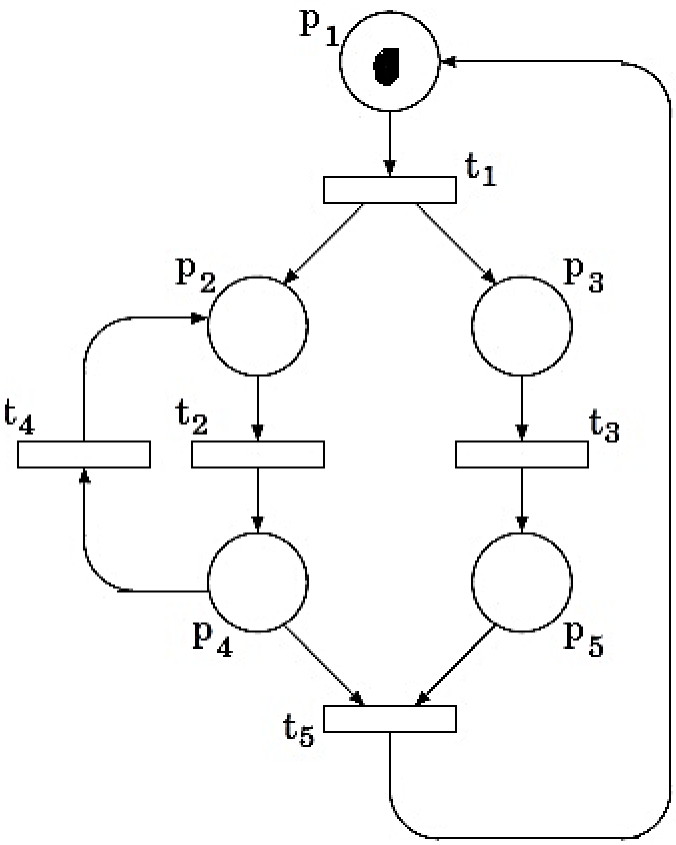
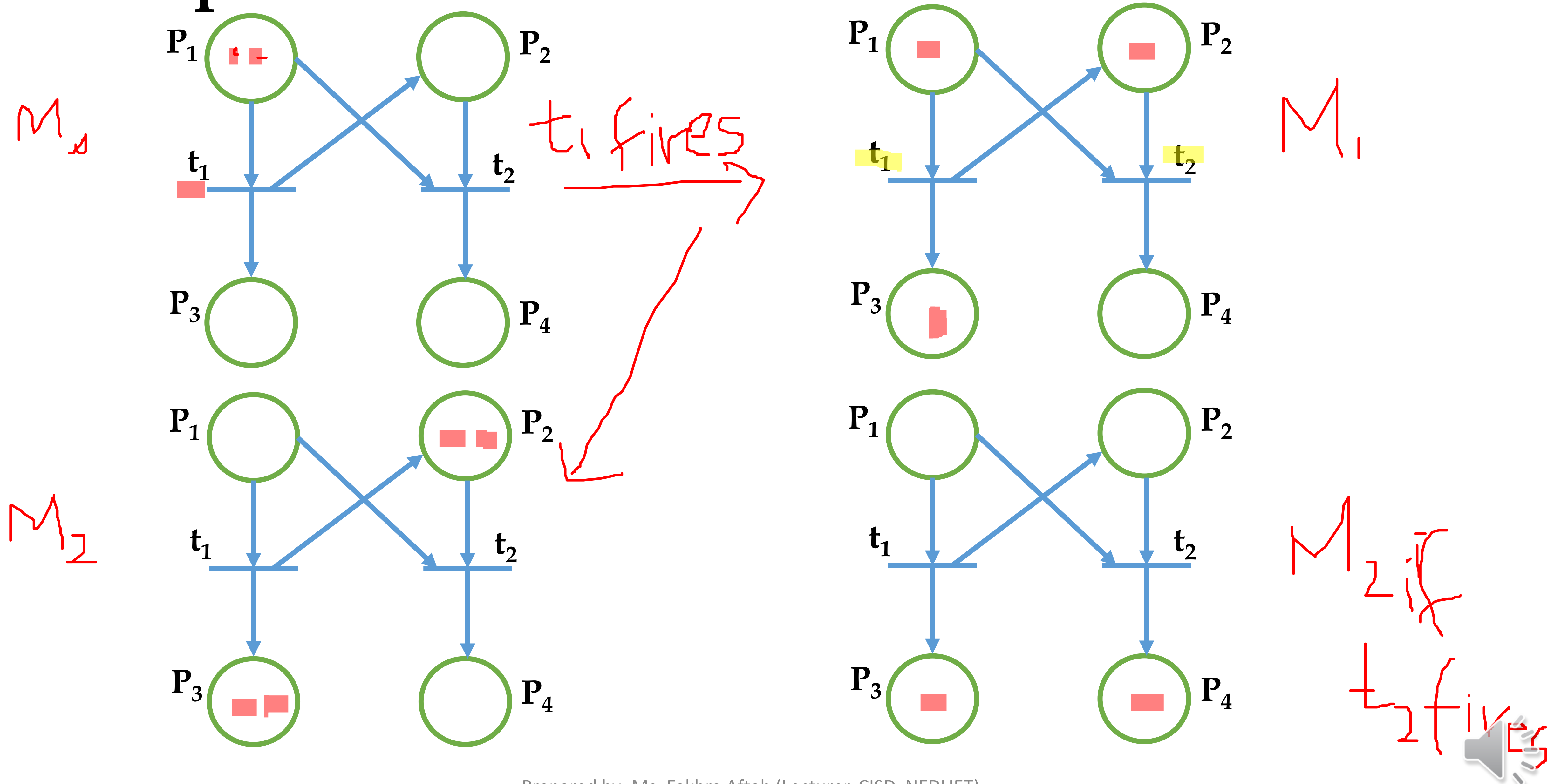


Fig 25

Example Problem 2



MATRIX ANALYSIS

- The input and output functions of a PN can be equivalently defined using a matrix notation.

- Let D^- denotes the input matrix. $\boldsymbol{D^-}$ is a (n_t x n_p) matrix, whose generic element $\boldsymbol{d_{ij}^-}$ is equal to the number of arcs connecting place $\boldsymbol{p_j}$ with transition $\boldsymbol{t_i}$.

- Similarly the output matrix $\boldsymbol{D^+}$ is a (n_t x n_p) matrix, whose generic element $\boldsymbol{d_{ij}^+}$ is equal to the number of arcs connecting transition $\boldsymbol{t_i}$ with place $\boldsymbol{p_j}$.

- The *incidence matrix D* is defined by the following relation:

$$\boldsymbol{D = D^+ - D^-}$$

Example

Following are the matrices D^-, $D+$ and D for the PN of Fig 26:

$$D^- = \begin{array}{c|ccccccc|} & p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & p_7 \\ \hline t_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ t_2 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ t_3 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ t_4 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ t_5 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ t_6 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}$$

$$D^+ = \begin{array}{c|ccccccc|} & p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & p_7 \\ \hline t_1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ t_2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ t_3 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ t_4 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ t_5 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ t_6 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{array}$$

$$D = \begin{array}{c|ccccccc|} & p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & p_7 \\ \hline t_1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ t_2 & 0 & -1 & 1 & -1 & 0 & 0 & 0 \\ t_3 & 1 & 0 & -1 & 1 & 0 & 0 & 0 \\ t_4 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ t_5 & 0 & 0 & 0 & -1 & 0 & -1 & 1 \\ t_6 & 0 & 0 & 0 & 1 & 1 & 0 & -1 \end{array}$$

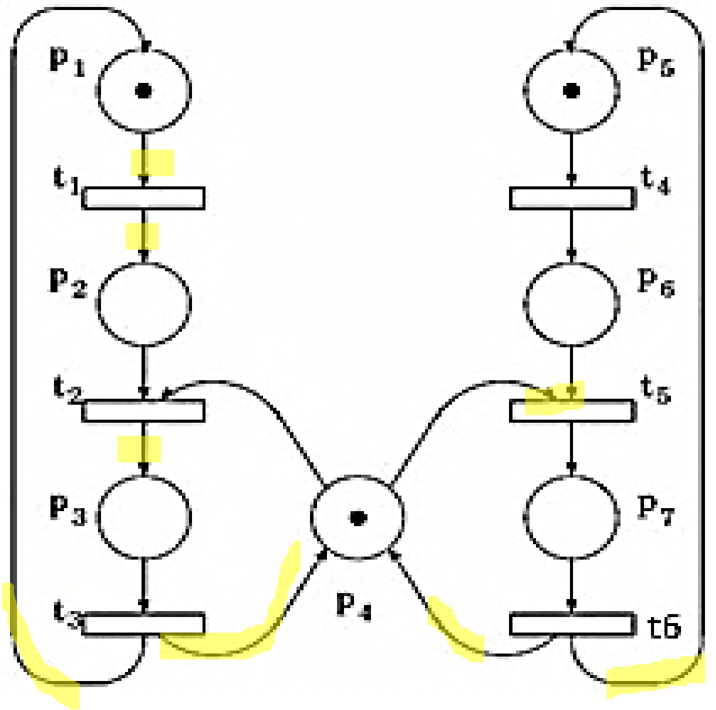


Fig 26

- Introducing the vector $\underline{e}_j$ which is a n_t-dimensional row vector with all the entries equal to 0 except entry j equal to 1.

- With this notation the execution rules of a PN becomes:
  - a transition t_j is enabled in marking M iff $\boldsymbol{M \geq \underline{e}_j D^-}$ (note that $\underline{e}_j D^-$ is the *j-th* row of D^-);
  - firing of t_j in M produces a marking M' given by:

$$\boldsymbol{M' = M - e_j D^- + e_j D^+ = M + e_j D}$$

- Given a PN with initial marking $\boldsymbol{M_1}$ and a firing sequence $t_i \to t_j \to t_k \to t_j \to t_i$, the marking obtained at the end of the sequence is given by the following matrix equation:

$$\boldsymbol{M_{fin} = M_1 + (\underline{e}_i + \underline{e}_j + \underline{e}_k + \underline{e}_j + \underline{e}_i)D}$$

Example Problem 3

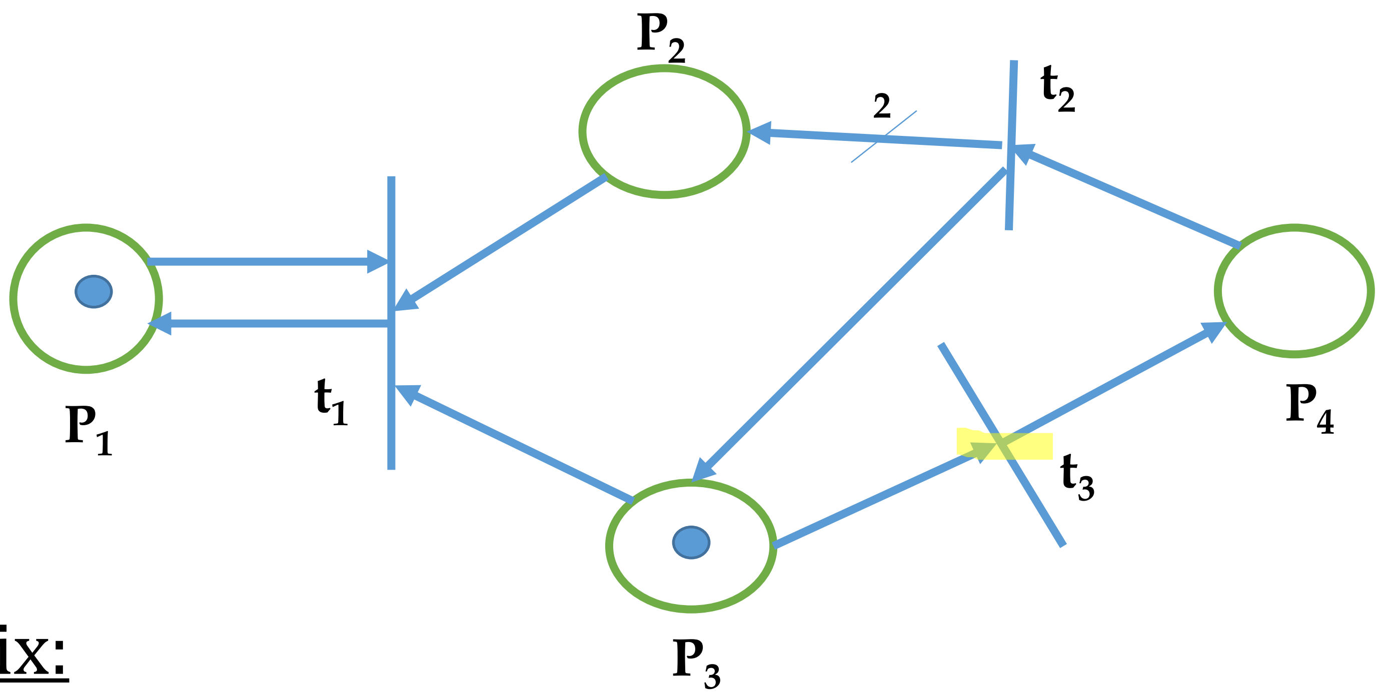
Q) Consider the figure, find M' if t_3 is enabled to fire.

Input or Pre-Incidence Matrix:

$$D^- = \begin{array}{c} t_1 \\ \\ \\ \end{array} \begin{vmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{vmatrix}$$

Output or Post-Incidence Matrix:

$$D^+ = \begin{vmatrix} 1 & 0 & 0 & 0 \\ \underline{0} & 2 & 1 & \underline{0} \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{array}{c} \\ \underline{t_1} \\ \\ \end{array}$$



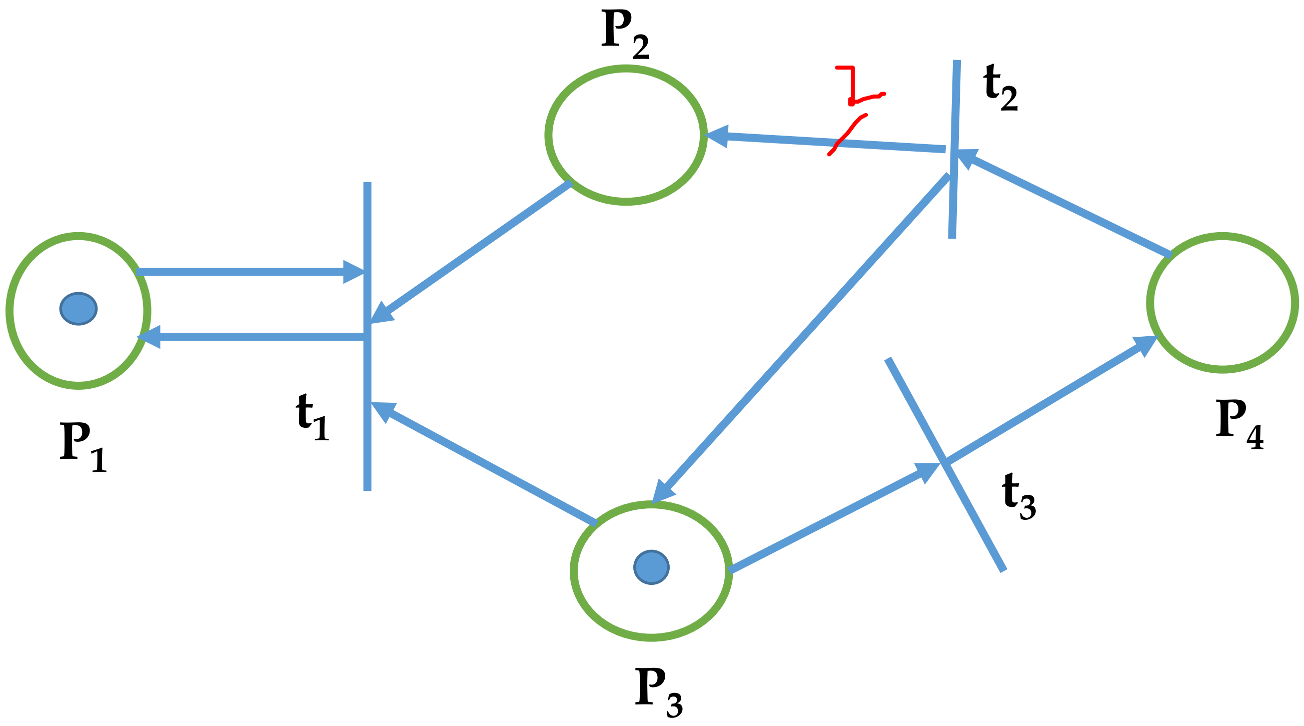
Fig

Incidence Matrix:

$$D = \begin{vmatrix} 0 & -1 & -1 & 0 \\ 0 & 2 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{vmatrix}$$

Initial Marking:

$M_0 = [1\ 0\ 1\ 0]$



Fig

From the Fig, t_3 is enabled:

$\boldsymbol{e_j = (0, 0, 1)}$

$$M' = M + e_j D$$

$$M' = [1\ 0\ 1\ 0] + [0\ 0\ 1] \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 2 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$M' = [1\ 0\ 0\ 1]$$

Q) Let the firing sequence be: $t_3 t_2 t_3 t_2 t_1$

$\Longrightarrow\ e_j = [1\ 2\ 2]$

If $M = [1\ 0\ 1\ 0]$,

$$M' = [1\ 0\ 1\ 0] + [1\ 2\ 2] \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & 2 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$M' = [1\ 3\ 0\ 0]$$

Timed Petri Nets

- An activity in a real system takes finite time to perform its operation.
- e.g., to read a file from a disk, to execute a program, or to communicate with some other machine.
- Adding time provides Petri net modeler with another *powerful tool* to study the performance of computer systems.
- Time can be associated with transitions, selection of paths, waiting in places, inhibitors, and with any other component of the Petri net.
- The most typical way that time used in Petri nets is with transitions.
  - the firing of a transition can be viewed as the execution of an event being modeled – e.g., a CPU execution cycle.
- These timed transitions are represented graphically as a *rectangle or thick bars*.

The Semantics of the Firing

- When a transition becomes enabled, its clock timer is set and begins to count down.
- Once the timer reaches 0, the transition fires.
- In Fig 27, when token arrives at p_l, the timer for t_1 set to τ_1 and begins to count down.
- The decrement of the timer must be at a constant fixed speed for all transitions in the Petri net model.

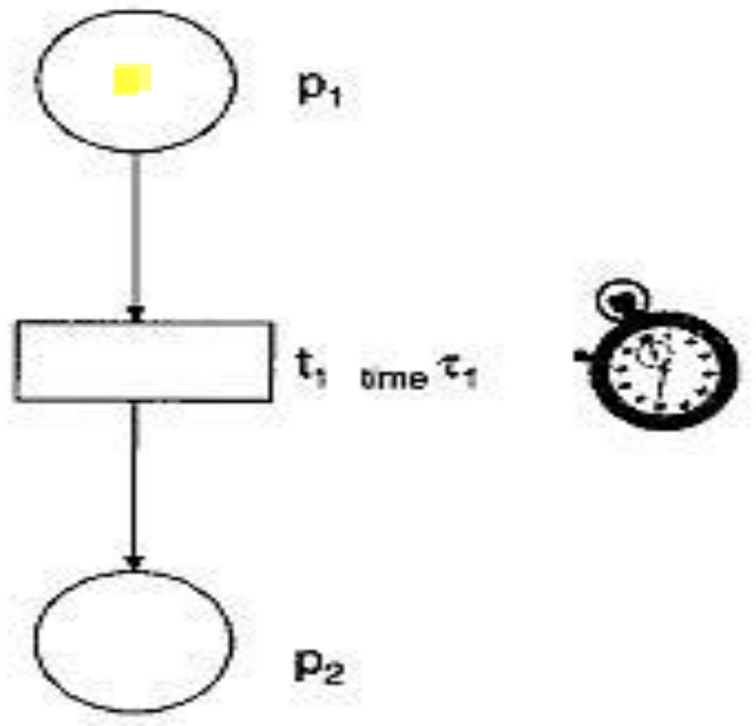


Fig 27: Timed Petri net

- A consideration to think about
  - what occurs when a transition becomes non-enabled due to the initial enabling token being used to ultimately fire a competing transition.
- This condition shown in Fig 28.
- If we assume the time for $\mathbf{t_1}$ is less than that for $\mathbf{t_2}$, then, when $\mathbf{p_1}$ receives a token, the two timers would begin counting down.
- At some time ($\boldsymbol{\tau_1}$) in the future, the timer for $\mathbf{t_1}$ would reach its zero value, resulting in the firing of $\mathbf{t_1}$.
- Since the token enabling $\mathbf{t_2}$ is now gone, $\mathbf{t_2}$ is no longer enabled and, therefore, its timer ($\boldsymbol{\tau_2}$) would stop ticking down.
- The question now is what to do with transition $\mathbf{t_2}$'s timer.

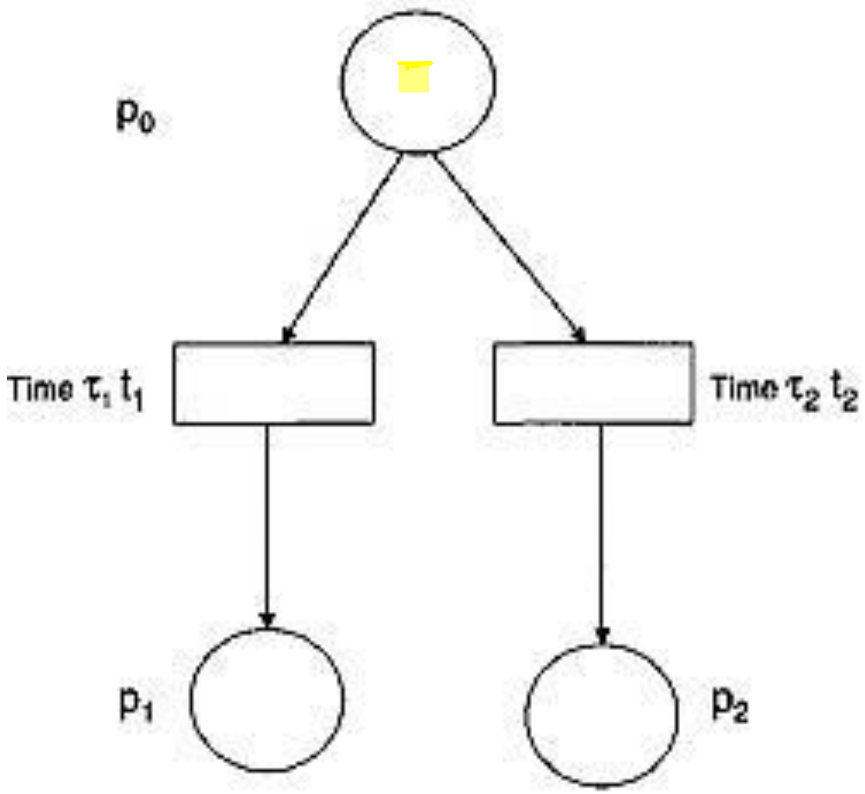


Fig 28: Timed Petri net with conflict

Two possibilities

1) The first is to simply reset the timer on the next cycle.

- In this case, unless place $\mathbf{p_1}$ has a state where it has more than one token present, transition $\mathbf{t_2}$ will never fire.

2) Allow transition t_2's timer to maintain the present clock timer value $(\tau_2 - \tau_1)$.

- When the next token is received at p_1, if the remaining time in t_2's timer is less than t_1's timer, then t_2 will fire, leaving t_1 with the remaining time $(\tau_1 - (\tau_2 - \tau_1))$.

The choice of which protocol to use will depend on the system one wishes to model.