# CS-417
# COMPUTER SYSTEMS MODELING

## Spring Semester 2020

### Batch: 2016-17

### (LECTURE # 28)

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

# Recap of Lecture # 27

Petri Nets - Introduction

Graphical & Set Notation Representation

Dynamic behavior of Petri-Nets

Dual of a Petri-Net

## Chapter # 8 (Cont’d)

# PETRI NET-BASED PERFORMANCE MODELING

Prepared by: Ms. Fakhra Aftab (Lecturer, CISD, NEDUET)

# Inverse of Petri Net

The *inverse* of a Petri net keeps all places and transitions the same and switches input functions with output functions.

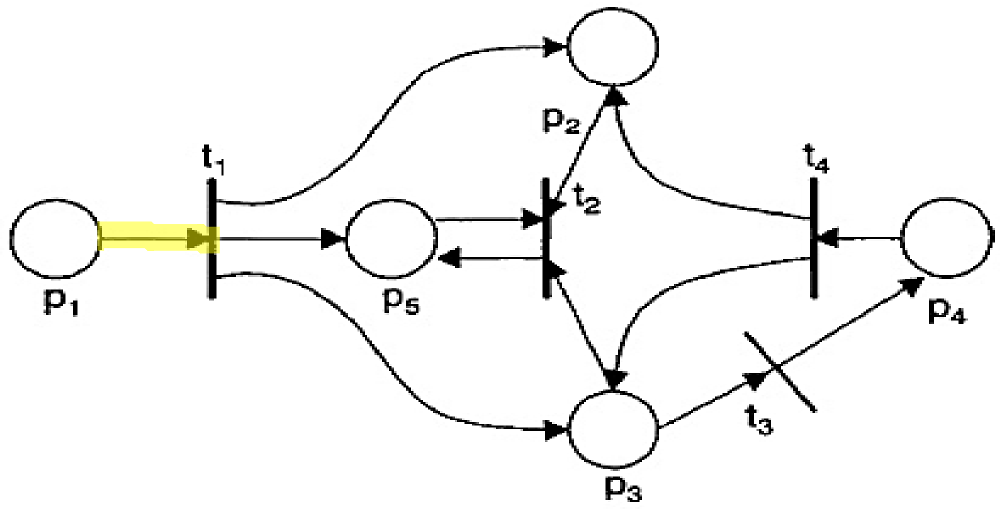


**Fig 3:** Petri net example

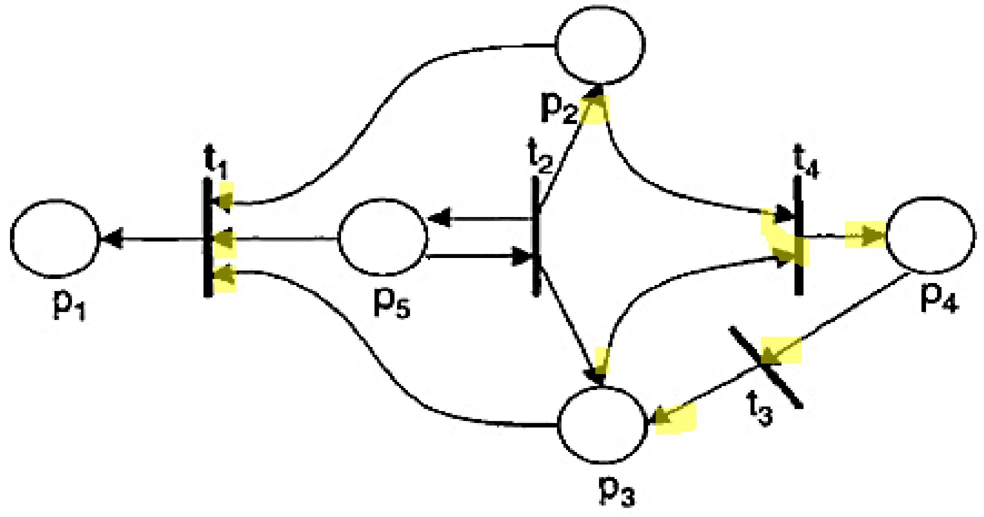


**Fig 6:** Inverse of Petri Net from Fig 3

# Petri Nets as Multi-graph

Petri nets are defined also as *multi-graphs*, since a place can represent multiple inputs and/or outputs from or to a transition.

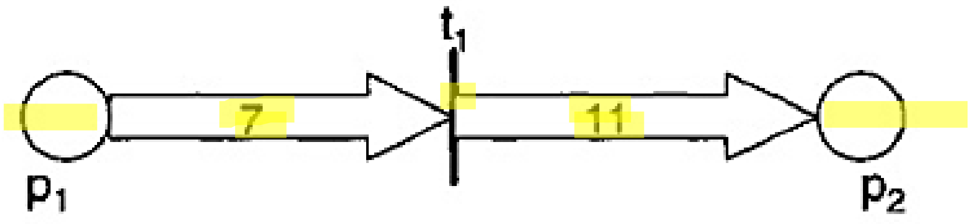


**Fig 7:** Multipath arc

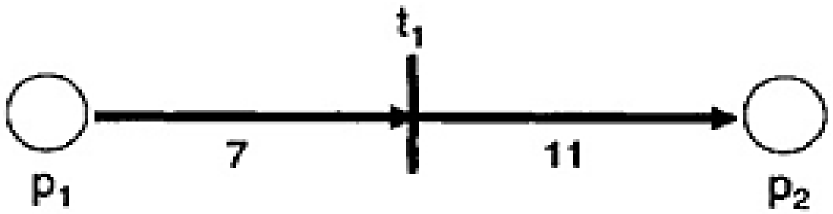


Fig 8: Multipath arc as bold line

# State of a Petri Net

- Petri nets have a *state* defined by the cardinality of tokens and their distribution throughout the places in the Petri net.
- Marking represented as a function, $\mu$ (or MP), as follows:

$$\mu: p \rightarrow Z^+$$

- The marking, $\mu$, can also be defined as an *n* vector.

$$\mu = (\mu_1, \mu_2, \mu_3, \ldots, \mu_n)$$

Where $n = |P|$ and each $\mu_i \in Z^+$ , $i = 0, \ldots, n$ and $\mu(p_i) = \mu_i$.

- Therefore, the true representation of a marked Petri net is:

$$M = (P, T, I, O, \mu_t)$$

where $\mu_t$ represents state of Petri net at time *t*, where $t \in Z^+$.

- Set of all possible markings for a Petri net with n places
  - the set of all n vectors, $N^n$,
  - N represents all possible states and *n* the no. of places.
- The number of tokens that may be assigned to a place is unbounded.

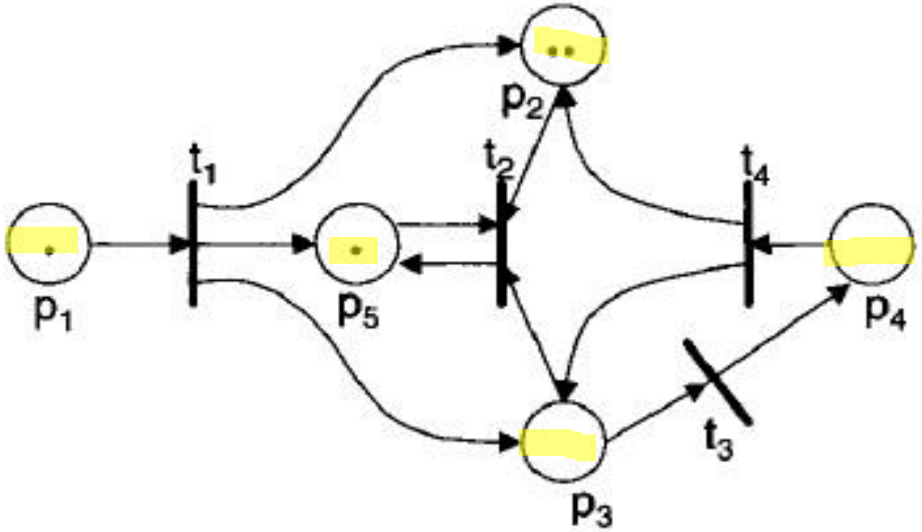


**Fig 9:** Marked Petri net

- The marking for the Petri net shown in Fig 9 represented as a vector would be $\mu_t = (1, 2, 0, 0, 1)$.

# Classical Petri Net

- The classical PNs do not convey any notion of time.
- The exact moment of firing can be pictured as occurring as a clock signal in a computer system.

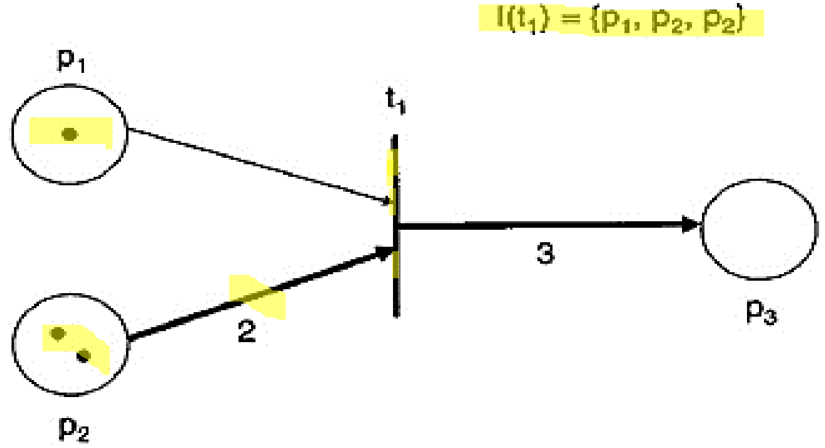


**Fig 10:** Enabled transition

Marking $\mu_o = (1,2,0)$

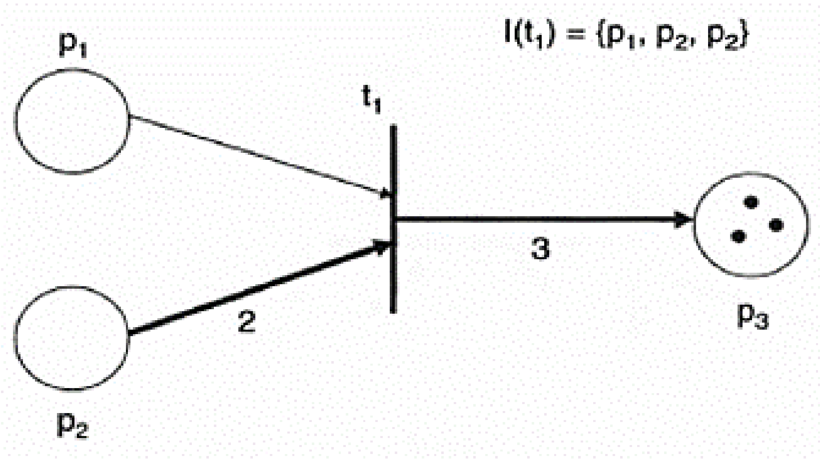


**Fig 11:** New Petri net state

Marking $\mu_1 = (0,0,3)$

- Input function $I(t_1) = \{P_1, P_2, P_2\}$ and Output function $O(t_1) = \{P_3, P_3, P_3\}$

# State Space

- The collection of all possible states of a Petri net.

- Next-state function, δ applied to a Petri net state as follows:

$$\delta\ (\mu_i\{t\}) = \mu_{i+1}$$

- The set *{t}* represents the set of all enabled transitions within this Petri net.

- If a transition not enabled, then this function is undefined.

# Petri Nets and the Modeling of Computer Systems

- PN used for modeling real systems sometimes referred to as *Condition/Events* nets.

- Places identify conditions of parts (working, idle, queuing, failed), and transitions describe the passage from one condition to another (end of a task, failure, repair ...).

- An event occurs (a transition fires) when all conditions satisfied.

- The number of tokens in a place used to identify the number of resources lying in the condition denoted by that place.

# Concurrency (Parallelism)

- In reliability modeling, the PN of Fig 12 can represent two components $C_1$ and $C_2$ in parallel redundancy.
- $p_1$ & $p_3$ represent working condition, $p_2$ & $p_4$ the failed condition and $t_1$ & $t_2$ the event of failure of $C_1$ & $C_2$ respectively.

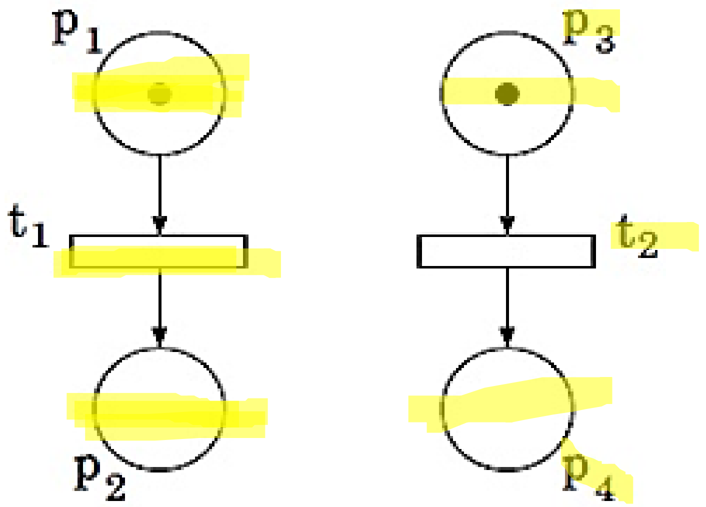


**Fig 12:** PN modeling two parallel activities

# Synchronization

- Both the routines of a parallel program should be terminated before the program execution can proceed.
- The synchronization activity modeled in Fig 13 by means of $t_3$ whose firing requires a token both in $p_2$ and $p_4$.

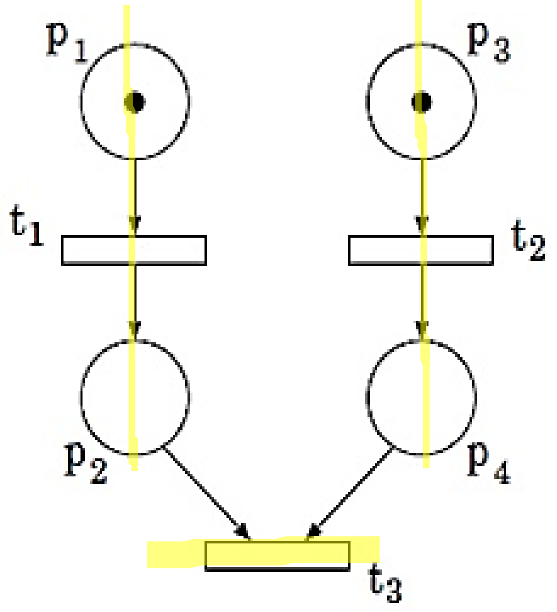


**Fig 13**: PN modeling two parallel activities with synchronization

# Limited Resources

- This is a typical factor influencing the performance of computer systems.
- A PN representation of a buffer with limited size.

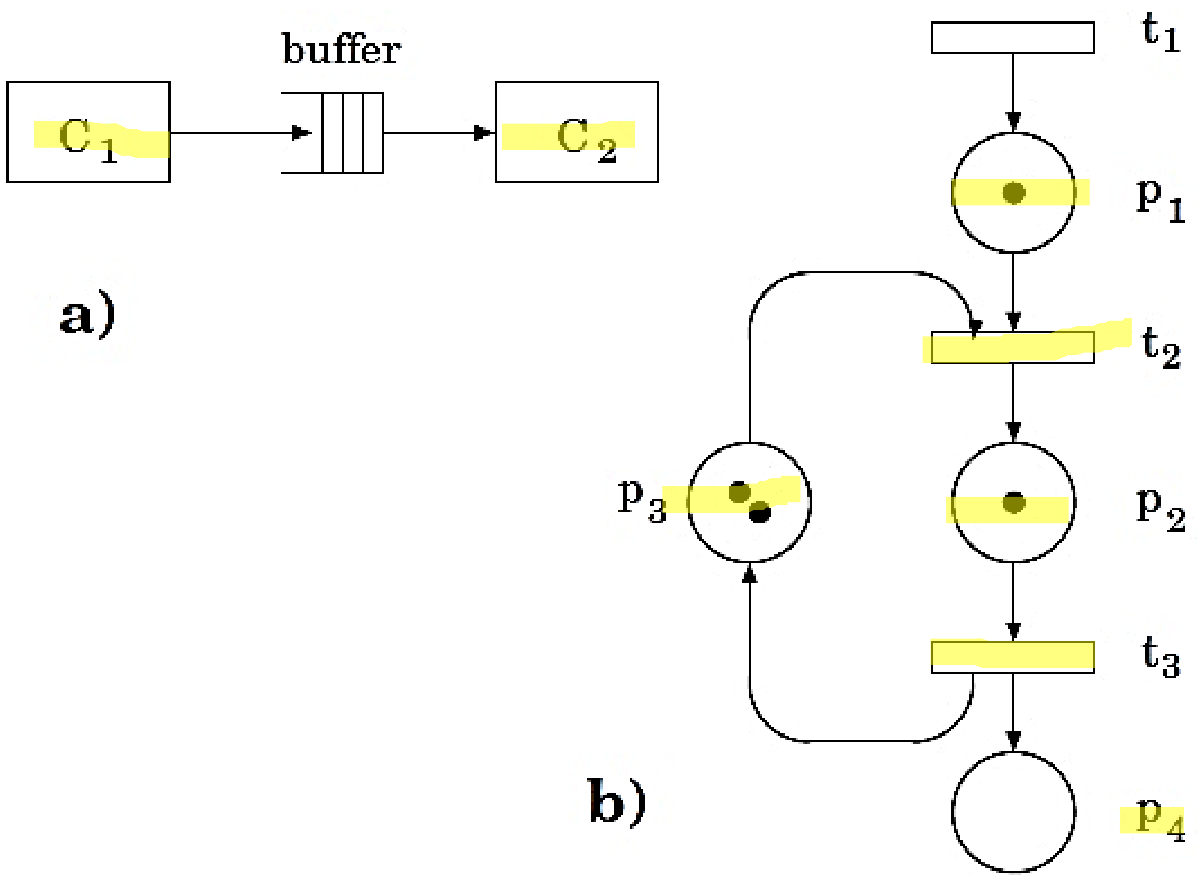


**Fig 14:** Block diagram and PN of a buffer with finite size.

# The Bounded Buffer Producer/Consumer Problem

- A realistic situation is obtained by considering a buffer of *limited capacity* (Fig 15).

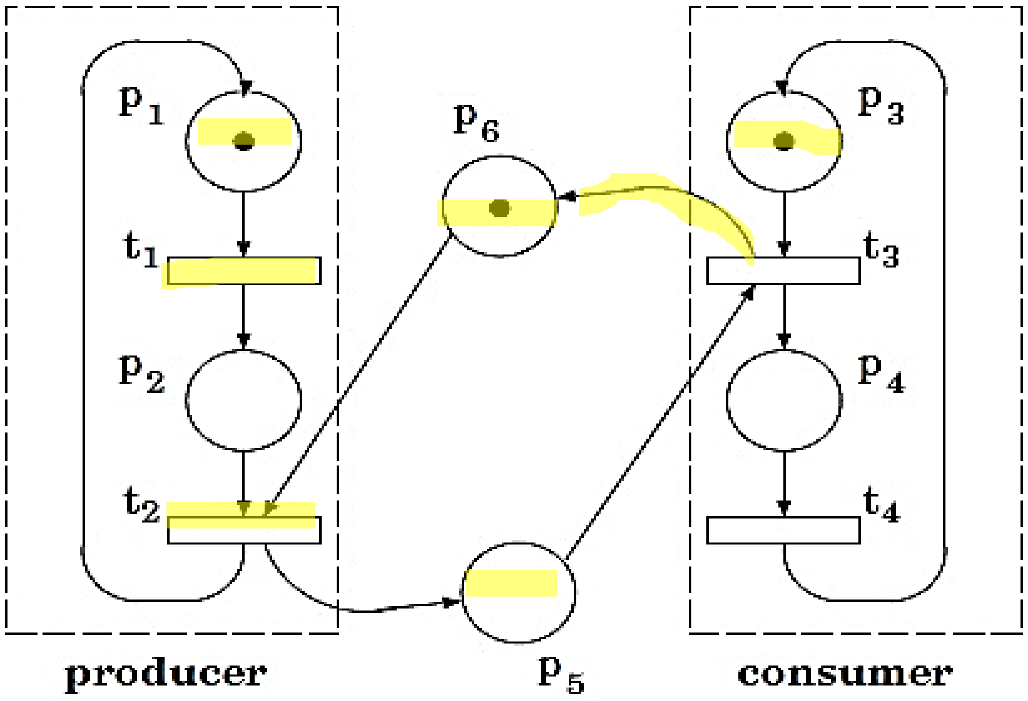


**Fig 15**: The producer/consumer problem with finite buffer