
EECE 276

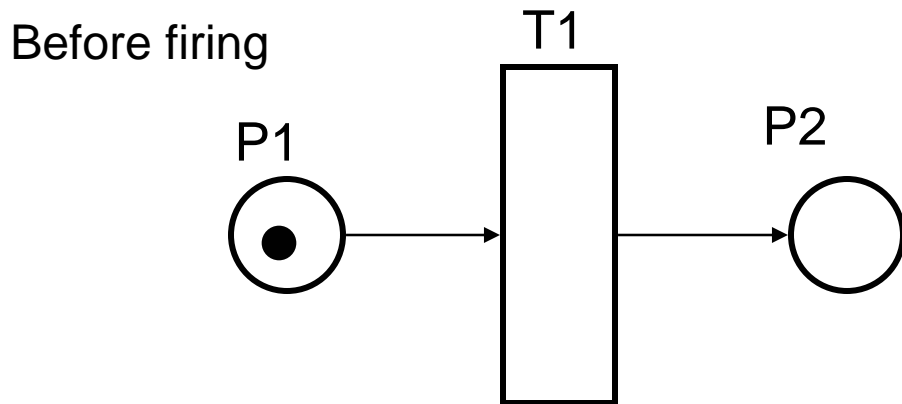
Embedded Systems

Techniques:
Petri nets

Petri nets

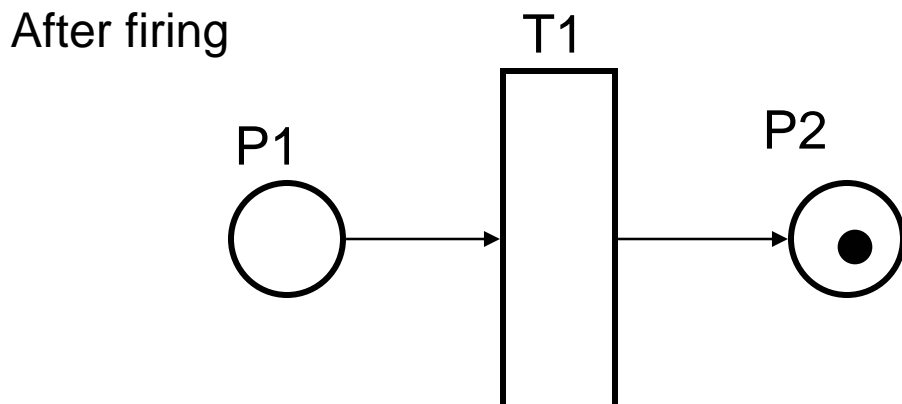
- Graphical representation for concurrent systems
- Ingredients:
 - » “Places”: process states, data stores, etc.
 - » “Transitions” : changes, operations, etc.
 - » “Marking”: initial “data count” for places
- Firing: a transition fires if it has sufficient input marks. Marks from the input places are removed and moved to the output places of the transition.

Petri nets



Transition table

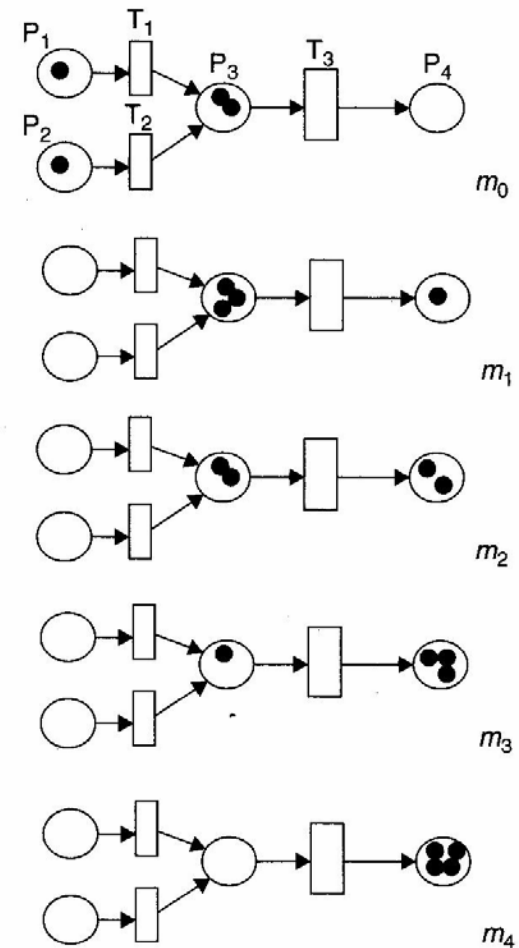
	P1	P2
Before	1	0
After	0	1



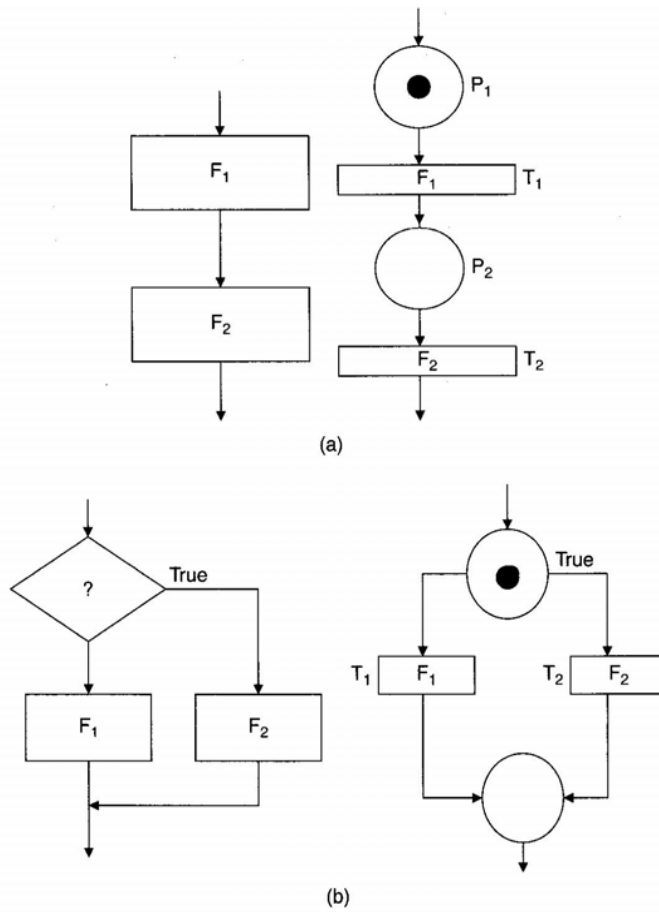
Petri net example

A more complex sequence and its firing table

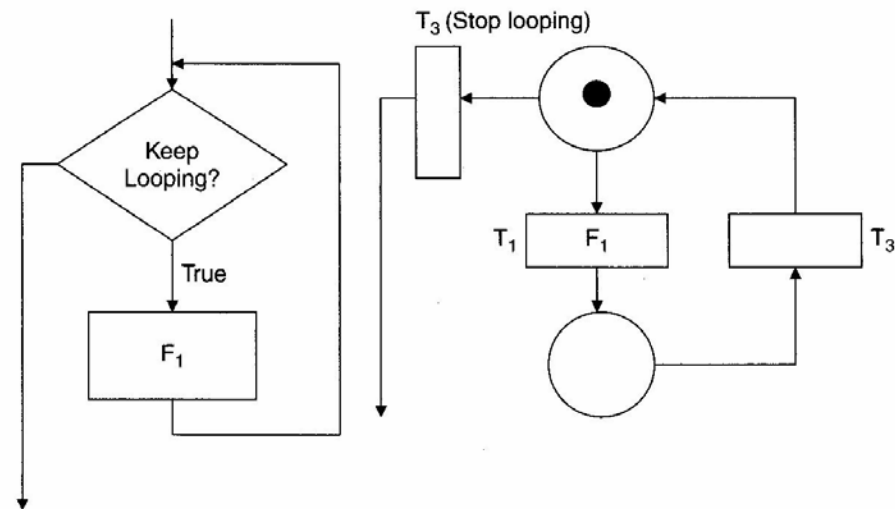
	P_1	P_2	P_3	P_4
m_0	1	1	2	0
m_1	0	0	3	1
m_2	0	0	2	2
m_3	0	0	1	3
m_4	0	0	0	4



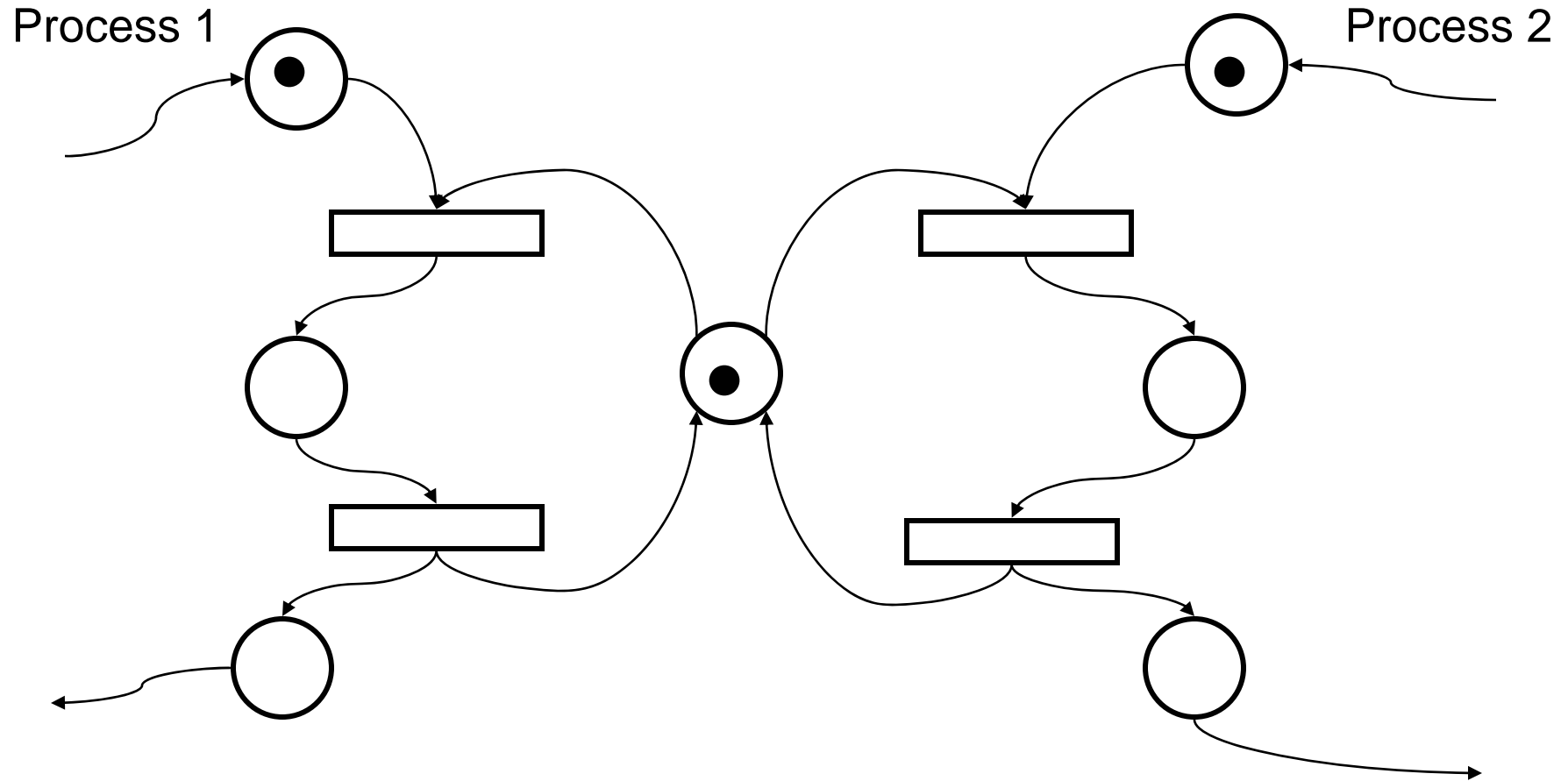
Petri nets represent control flow



Sequencing
Branching
Looping

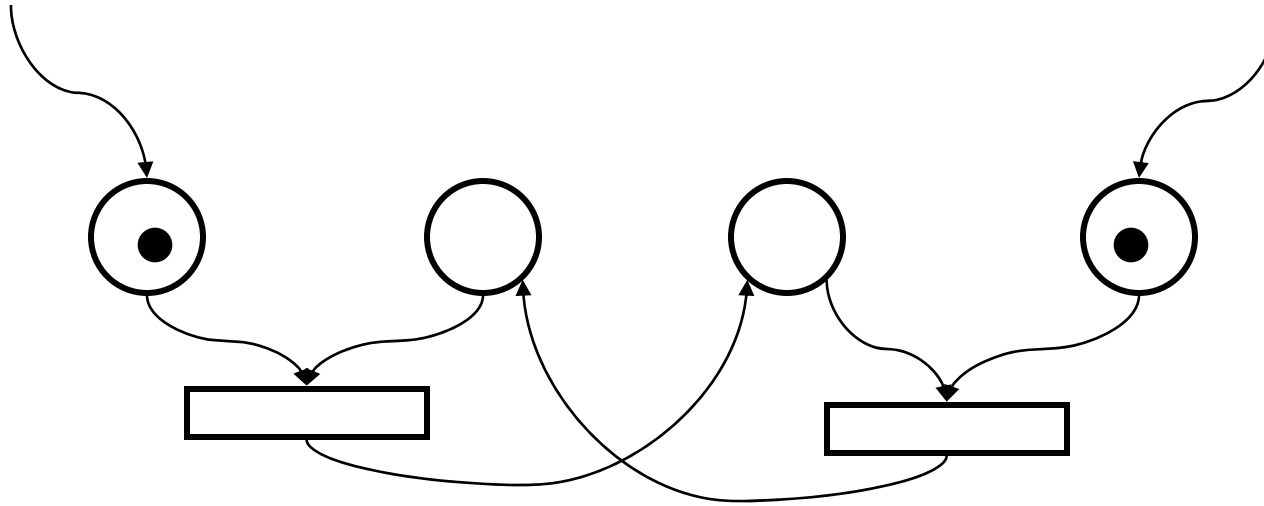


Petri nets represent concurrency



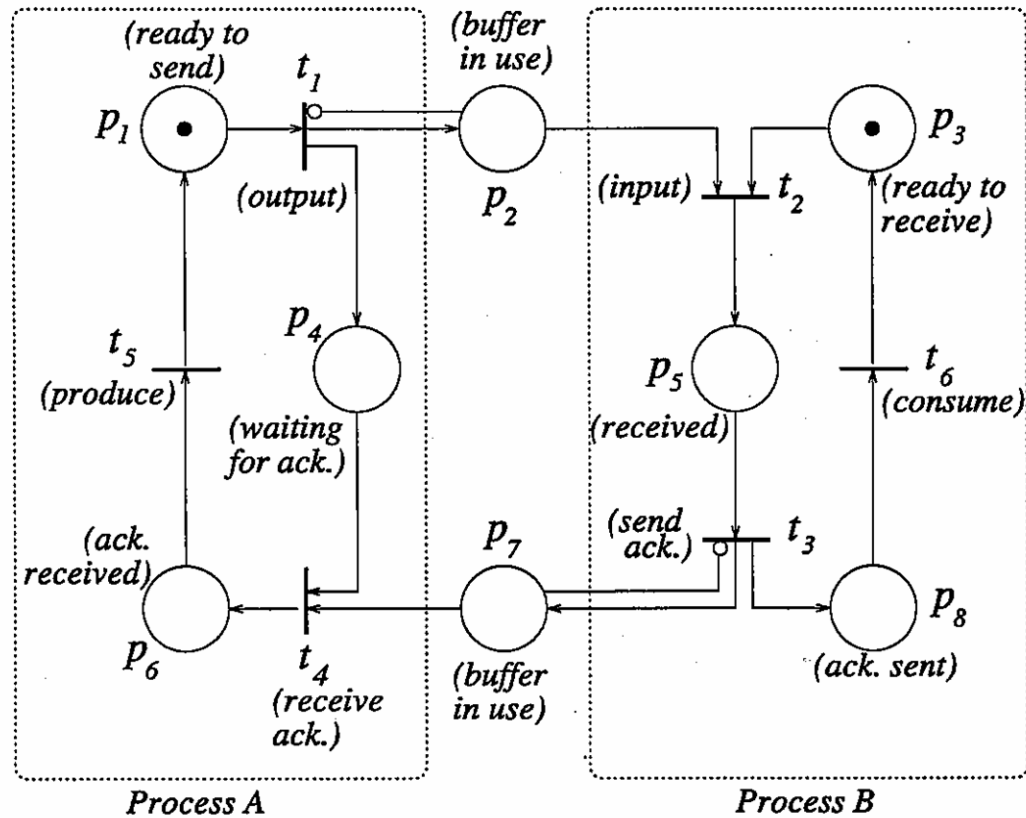
Two processes + one semaphore

Petri nets and deadlocks



Typical deadlock situation

Petri Net Example



Shortcomings of Petri nets

- Very simple building blocks lead to complex nets
- Extensions:
 - » Enabling/firing separated
 - » Tokens not only for control but also as data
 - » Attributes for transitions
 - Mean/variance of execution time
 - Transition probability