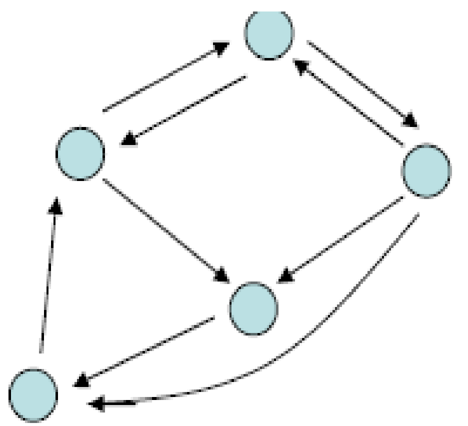
Synchronous network model

- A Synchronous Network System consists of a collection of computing elements (“processors” of “Processes”) located at the nodes of a directed network graph.

- The computing elements are connected via a connected network (i.e., there exists one channel between any pair of processes).



Synchronous network model

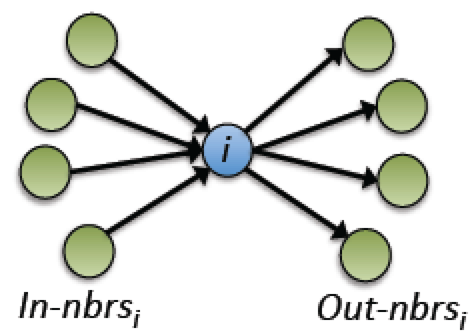
- Formally,

  Directed graph $G=(V,E)$

  – $n = |V|$, number of nodes in the graph

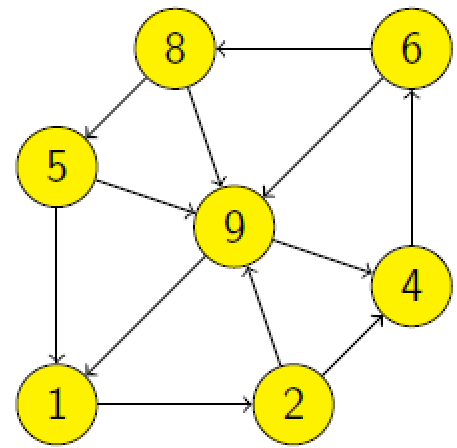
  – For $i \in V$

    - $Out\text{-}nbrs_i$ set of nodes j s.t. $(i,j) \in E$
    - $In\text{-}nbrs_i$ set of nodes j s.t. $(j,i) \in E$



Neighboring Processes

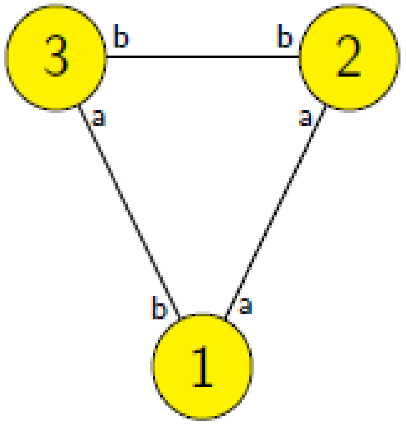
- We say vertex v is outgoing neighbor of vertex u if
  - the edge uv is included in G.
- We say vertex u is incoming neighbor of vertex v if
  - the edge uv is included in G.
- We define $nbrs_u^{out} = \{v | (u, v) \in E\}$ all the vertices that are *outgoing neighbors* of vertex u.
- We define $nbrs_u^{in} = \{v | (v, u) \in E\}$ all the vertices that are *incoming neighbors* of vertex u.



5 is outgoing neighbor of 8
8 is incoming neighbor of 5
$nbrs_9^{out} = \{1, 4\}$
$nbrs_9^{in} = \{2, 5, 6, 8\}$

Modeling Communication Channels

- Channels are the edges of the graph.
  - The edges may be directed – to represent unidirectional communication.
  - or undirected – to represent bidirectional communication.
- Processes can distinguish each communication channel and select a specific one to use.



Network Properties

`distance`(u,v)

Let `distance`(u,v) denote the length of the shortest directed path from u to j in G, if any exists; otherwise `distance`$(u,v)=\infty$.

`diam`(G)

Let `diam`(G) denote the diameter of the graph G, the maximum distance `distance`(u,v), taken over all paths (u,v).

System Initialization

- Initially
  - all processes are set to an initial state,
  - all channels are empty.
- Algorithms groups processes in two sets
  1. Initiators – a process is initiator if it activates the execution of the algorithm in the local neighborhood.
  2. Non-initiators – a non-initiating process is activated when a message is received from a neighboring process.

Centralized vs Decentralized

An algorithm is classified as centralized if the exists one and only one initiator in each execution and decentralized if the algorithm may be initialized with an arbitrary subset of processes.

- Usually centralized algorithms achieve low message complexity.
- Usually decentralized algorithms achieve improved performance in the presence of failures.

Uniformity

An algorithm is uniform if its description is independent of the network size n.

- A property that holds for a small network size, also holds for large network sizes.
- We only have to examine the behavior of a protocol (for a given property) in small network sizes.

Algorithm execution: Steps and Rounds

- All processes, repeat in a "synchronized" manner the following steps:

  1^{st} Step

  1. Apply the message generator function.
  2. Generate messages for each outgoing neighbor.
  3. Transmit messages over the corresponding channels.

  2^{nd} Step

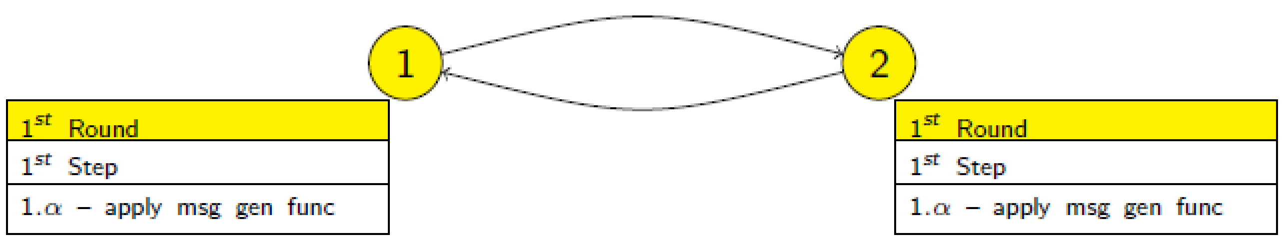
  1. Apply the state transition function.
  2. Remove all incoming messages from all channels.

- The combination of these two steps is called a round (of execution).

Example of execution of a Synchronous System

- Initially
  - all processes are set to an initial state,
  - all channels are empty.
- the processes execute in a "synchronized" manner the protocol.

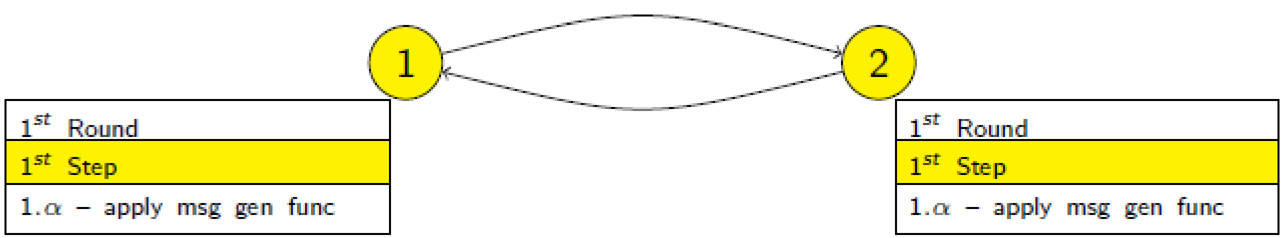
Execution of Synchronous System



Example of execution of a Synchronous System

- Initially
  - all processes are set to an initial state,
  - all channels are empty.
- the processes execute in a “synchronized” manner the protocol.

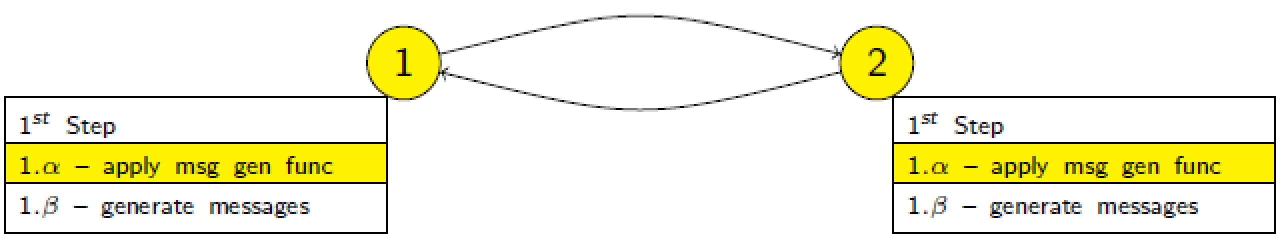
Execution of Synchronous System



Example of execution of a Synchronous System

- Initially
  - all processes are set to an initial state,
  - all channels are empty.
- the processes execute in a "synchronized" manner the protocol.

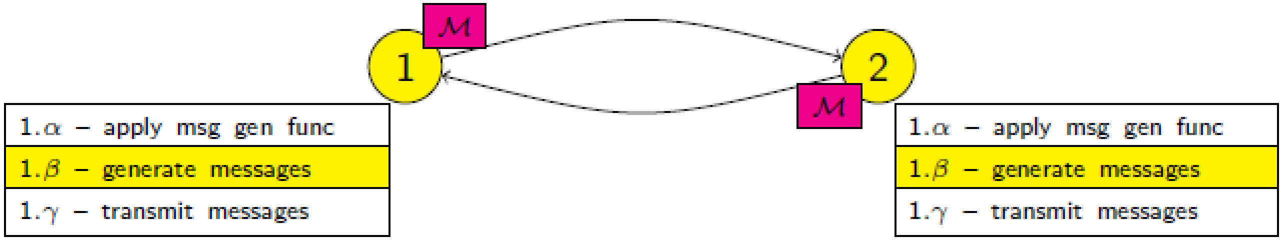
Execution of Synchronous System



Example of execution of a Synchronous System

- Initially
  - all processes are set to an initial state,
  - all channels are empty.
- the processes execute in a "synchronized" manner the protocol.

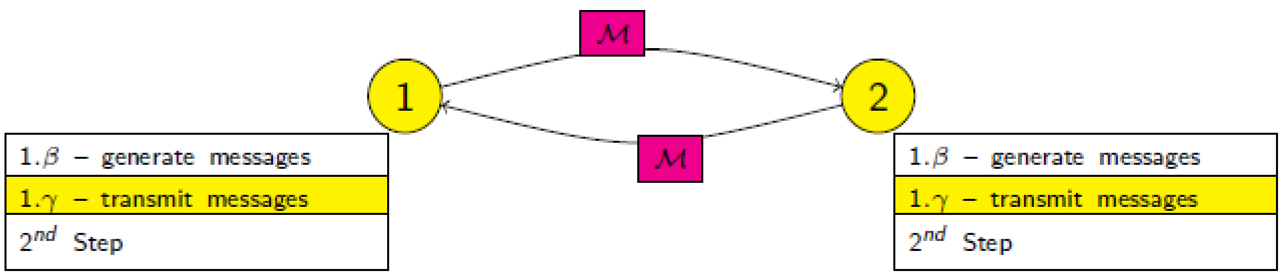
Execution of Synchronous System



Example of execution of a Synchronous System

- Initially
  - all processes are set to an initial state,
  - all channels are empty.
- the processes execute in a "synchronized" manner the protocol.

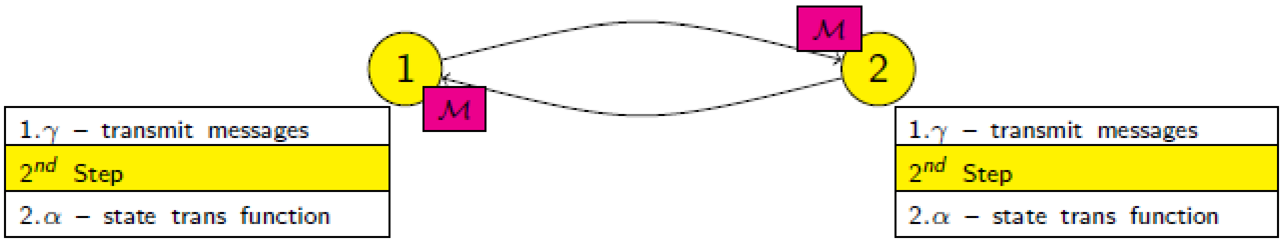
Execution of Synchronous System



Example of execution of a Synchronous System

- Initially
  - all processes are set to an initial state,
  - all channels are empty.
- the processes execute in a "synchronized" manner the protocol.

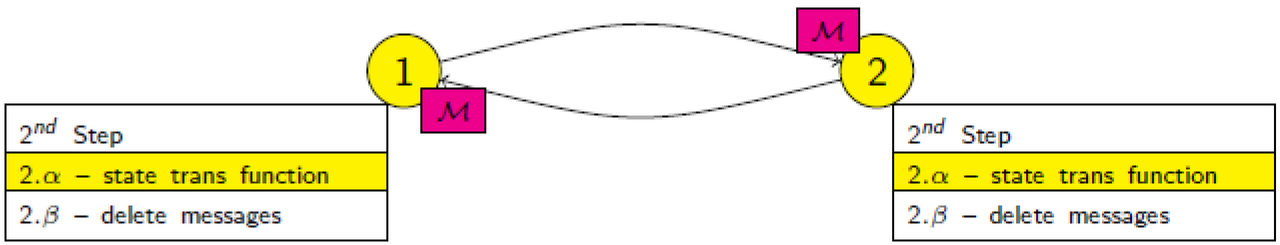
Execution of Synchronous System



Example of execution of a Synchronous System

- Initially
  - all processes are set to an initial state,
  - all channels are empty.
- the processes execute in a "synchronized" manner the protocol.

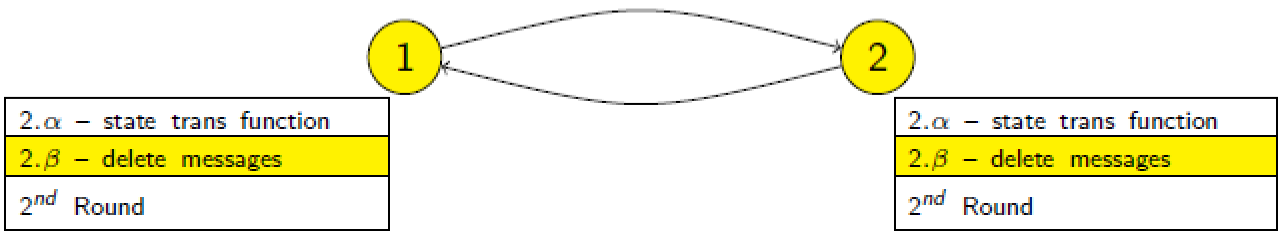
Execution of Synchronous System



Example of execution of a Synchronous System

- Initially
  - all processes are set to an initial state,
  - all channels are empty.
- the processes execute in a "synchronized" manner the protocol.

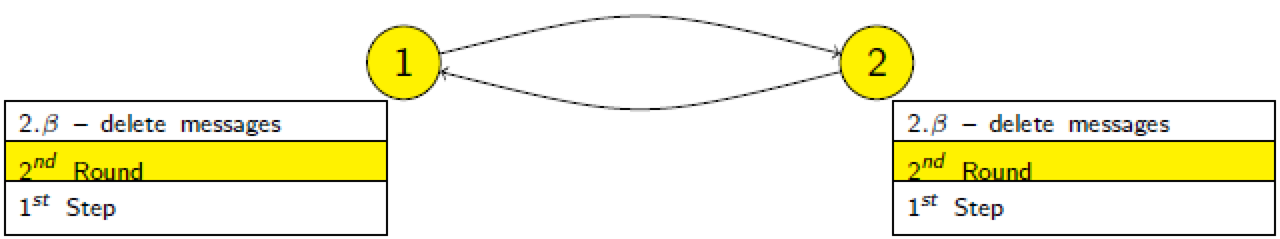
Execution of Synchronous System



Example of execution of a Synchronous System

- Initially
  - all processes are set to an initial state,
  - all channels are empty.
- the processes execute in a "synchronized" manner the protocol.

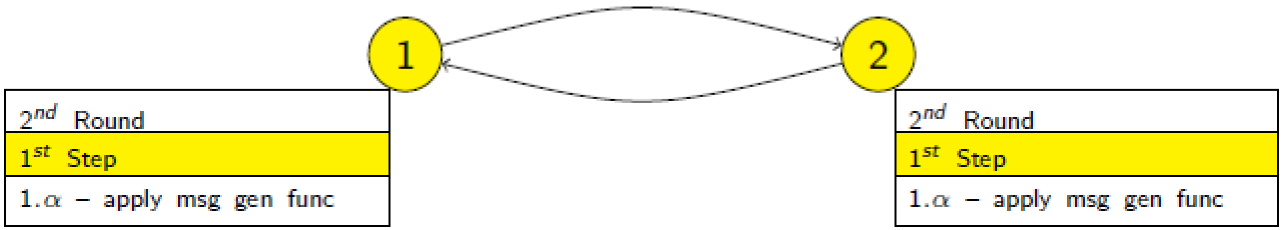
Execution of Synchronous System



Example of execution of a Synchronous System

- Initially
  - all processes are set to an initial state,
  - all channels are empty.
- the processes execute in a "synchronized" manner the protocol.

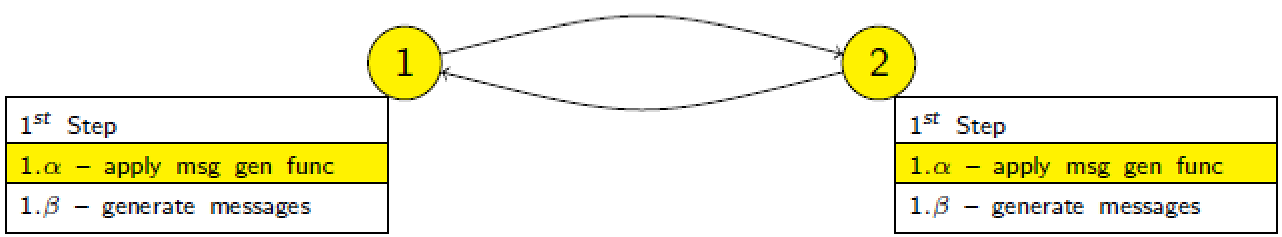
Execution of Synchronous System



Example of execution of a Synchronous System

- Initially
  - all processes are set to an initial state,
  - all channels are empty.
- the processes execute in a "synchronized" manner the protocol.

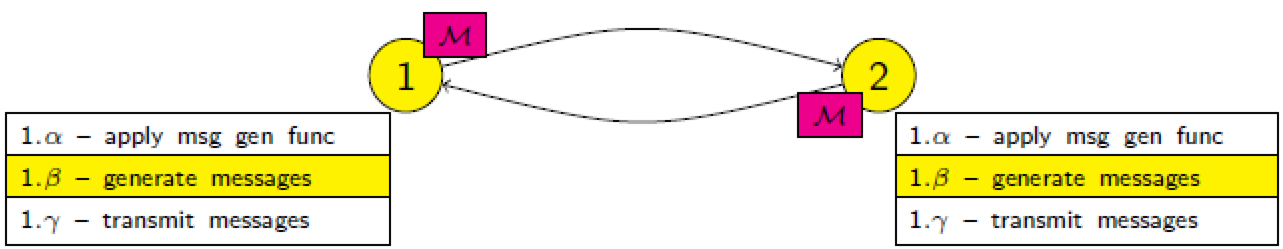
Execution of Synchronous System



Example of execution of a Synchronous System

- Initially
  - all processes are set to an initial state,
  - all channels are empty.
- the processes execute in a “synchronized” manner the protocol.

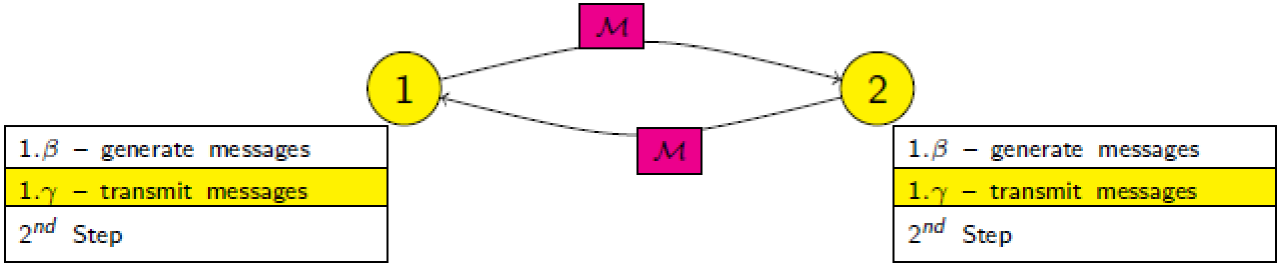
Execution of Synchronous System



Example of execution of a Synchronous System

- Initially
  - all processes are set to an initial state,
  - all channels are empty.
- the processes execute in a "synchronized" manner the protocol.

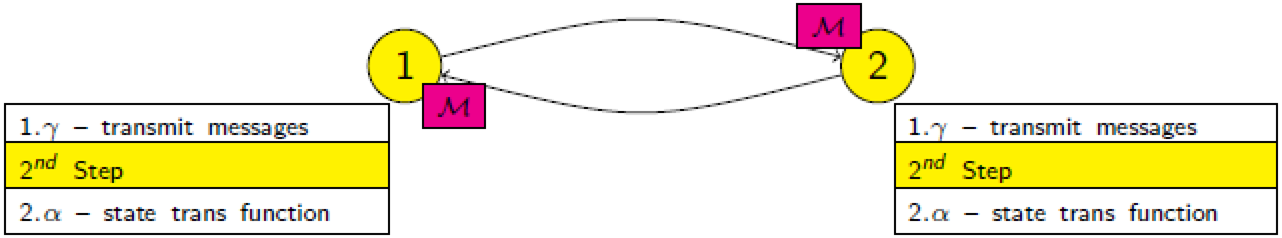
Execution of Synchronous System



Example of execution of a Synchronous System

- Initially
  - all processes are set to an initial state,
  - all channels are empty.
- the processes execute in a "synchronized" manner the protocol.

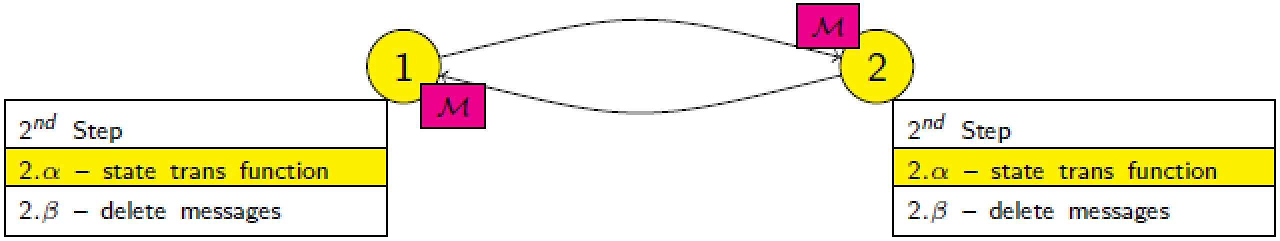
Execution of Synchronous System



Example of execution of a Synchronous System

- Initially
  - all processes are set to an initial state,
  - all channels are empty.
- the processes execute in a "synchronized" manner the protocol.

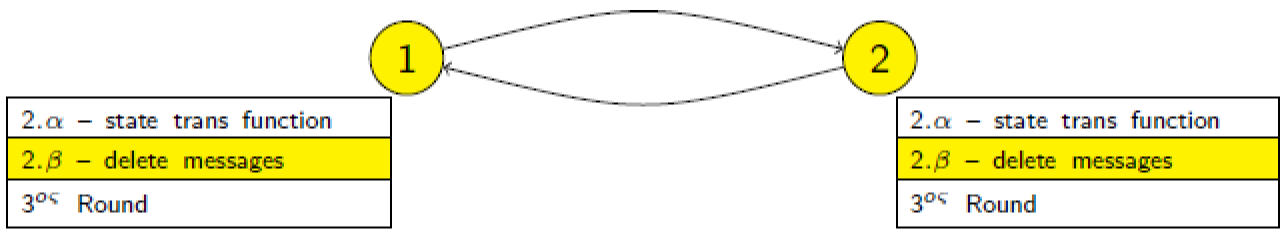
Execution of Synchronous System



Example of execution of a Synchronous System

- Initially
  - all processes are set to an initial state,
  - all channels are empty.
- the processes execute in a "synchronized" manner the protocol.

Execution of Synchronous System



Example of execution of a Synchronous System

- Initially
  - all processes are set to an initial state,
  - all channels are empty.
- the processes execute in a "synchronized" manner the protocol.

Execution of Synchronous System