Program Modeling Concepts:

Lesson-3: State Machine Programming Models For Event Controlled Programs

State machine model

- Assumed that program is a states producing machine
- Program flow from one stage to another models as changes in the state
- State transition function produce the states.
- A state transition function is a function, which changes a state to its next state

Examples of States

- Three states in a washing machine: ‘Washing, ‘Rinsing and Drying,
- A telephone process five states: Idle, Receiving a ring, Dialing, Connected and Exchanging messages.

Examples of States

- States of a process (function or interrupt service routine (ISR) or task: Idle, Running, Blocked, Finished

A process modeled using the states

- When there are inputs to a process that change the state of the process to a new state, and generate outputs which may also be the inputs for the next state.

Programming Model for an Event Controlled or Response Controlled Programs

- A programming model is to assume that the running process (program flow) can be considered as running of a machine in a particular state, which will lead to other states. or thread on an event(s) resulting into state transition

State machine model

A program flow in system can be modeled simply by an inter-state transition (from one state to another).

Steps in Modeling as State machine

The steps to model and represent the states and inter-state transitions are as follows:

1) A transition to a new state occurs from the previous state on an event (input). The event may be setting a value of certain parameter or the result of the execution of certain codes.

Steps in Modeling as State machine

2) A transition may be also be interrupt flag driven (after a flag sets) or semaphore driven or interrupt-source servicing need driven.

Steps in Modeling as State machine

3) A state is identified by a flag condition or set of codes being executed or set of values of certain parameters.

Steps in Modeling as State machine

4) A state can receive multiple tokens (inputs, messages, flags interrupts or semaphores) from another state (s). [A token (event) is a general term that means either an input or event-input.]

Steps in Modeling as State machine

- An event-input characteristic is that it is asynchronous (one never knows the instance when an event may happen). An event-input may be the setting or resetting of a flag.

Steps in Modeling as State machine

5) A state can generate multiple tokens (outputs, messages, flags interrupts or semaphores)

Finite State Machine (FSM) Model for Event Controlled or Response Controlled Programs

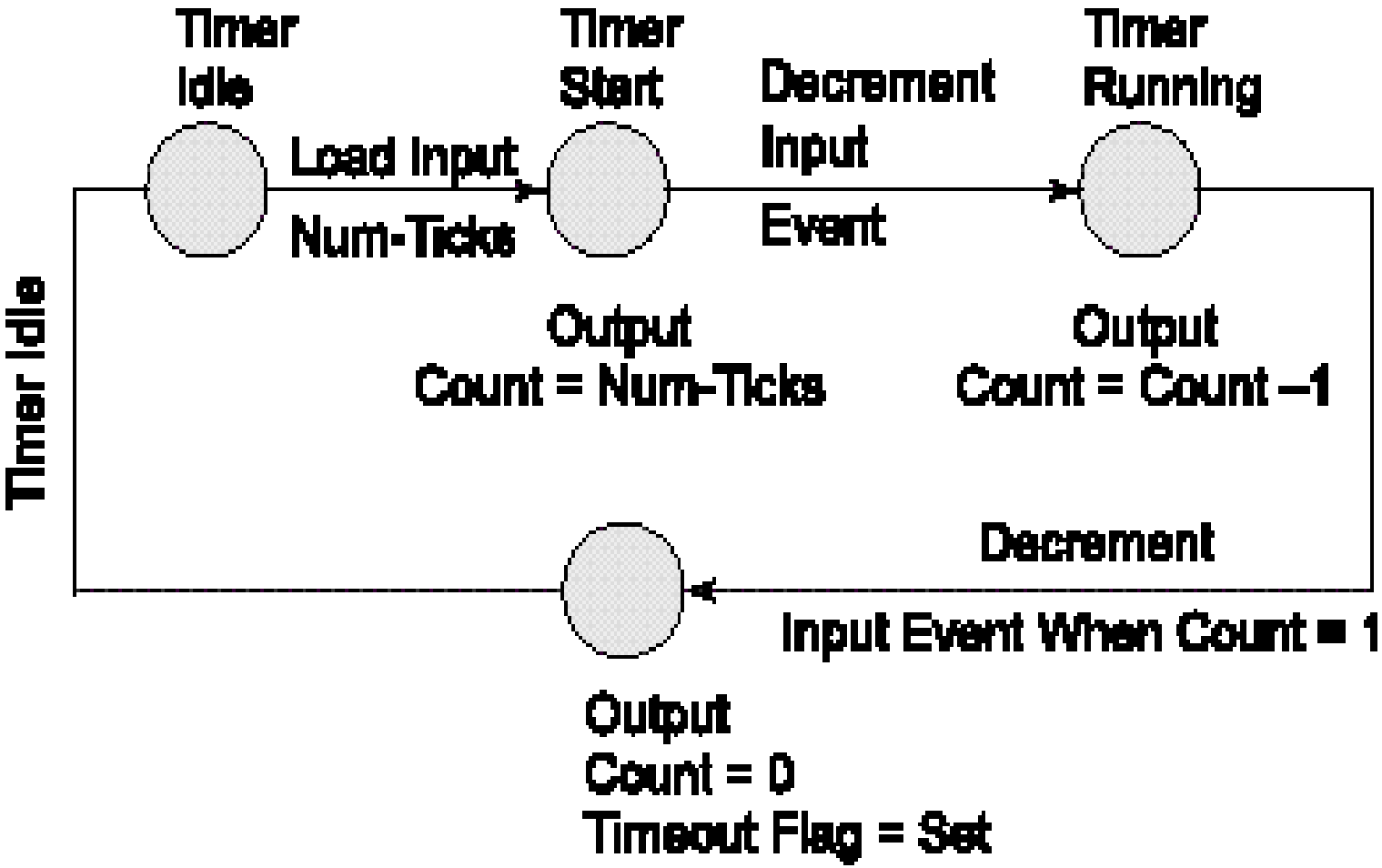
In FSM model:

- A circle represents a state of a program
- A directed arc (or an arrow) represent the program flow from a state to another.

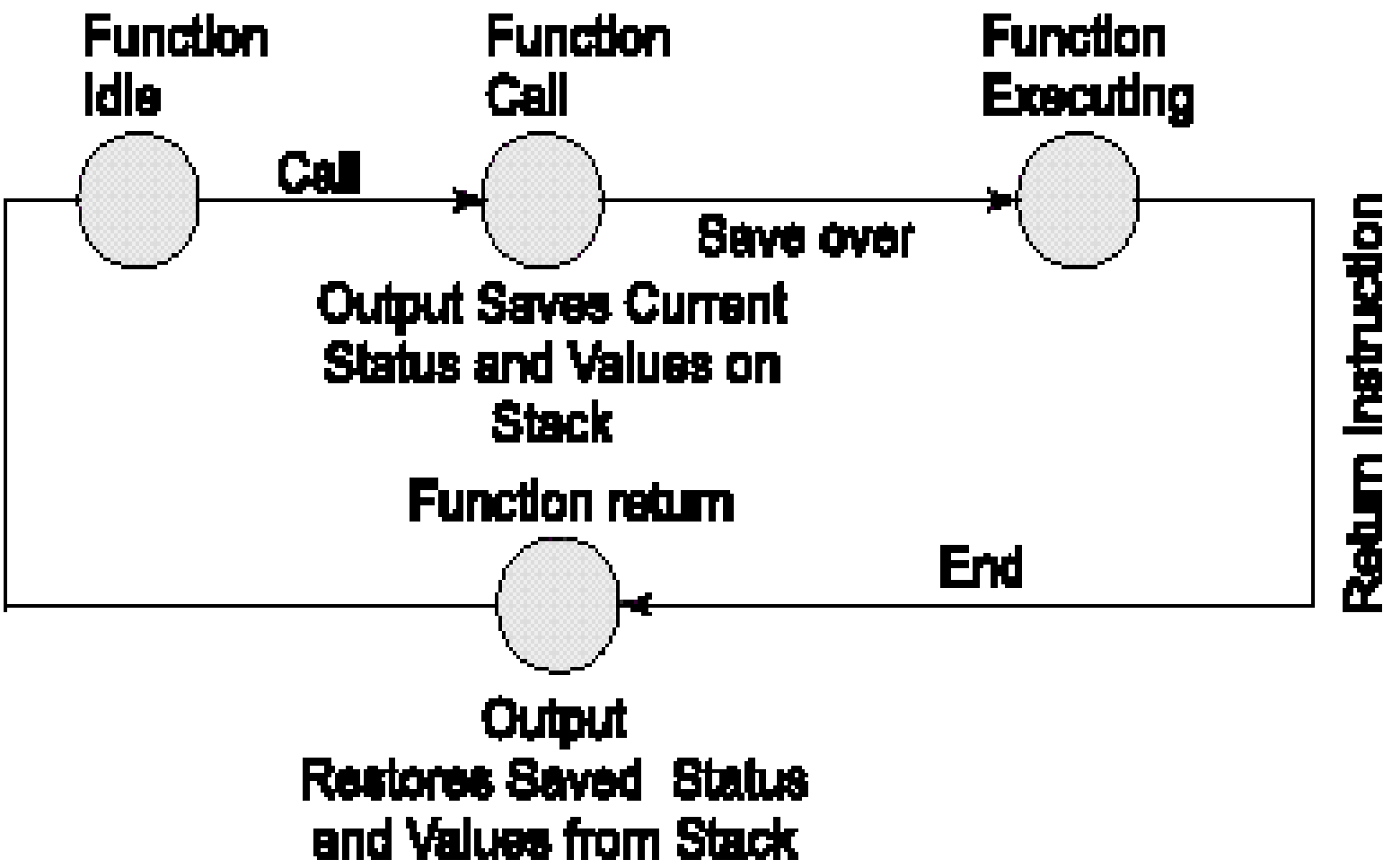
Modeling a process as finite state machine (FSM),

- Followings specified for each state
- The state is one of a finite number of states.
- Finite set of inputs (tokens or event-flags or status flags) with their values for the state.
- Finite actions (for example computations) during the state and finite set of outputs with their possible values (or tokens or event-flags or status flags) and an output (action) *function* for the state that give the outputs.
- *State transition function* for each state to take it to the next state.

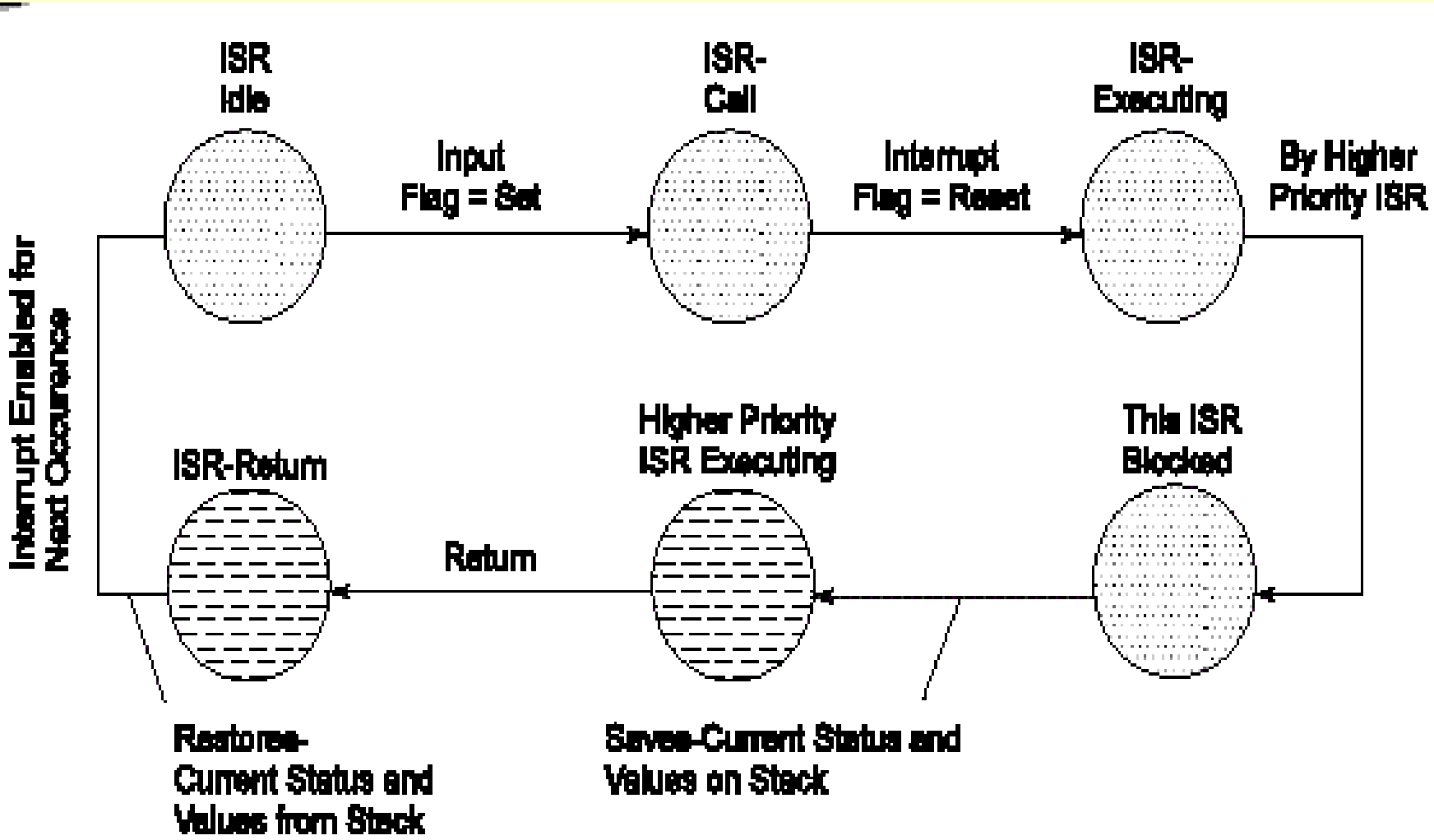
FSM model for the timer states



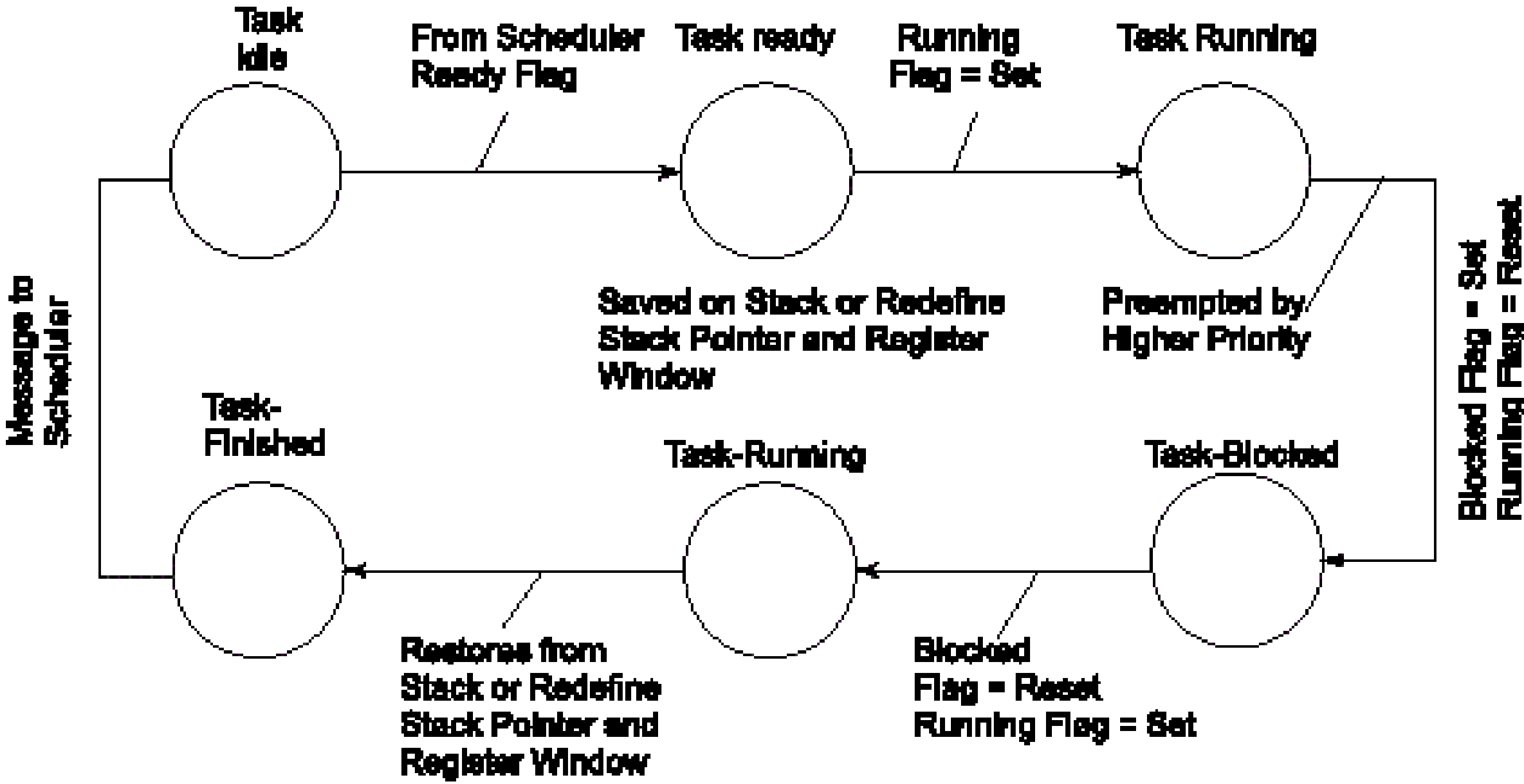
FSM model for a ‘C’ Function



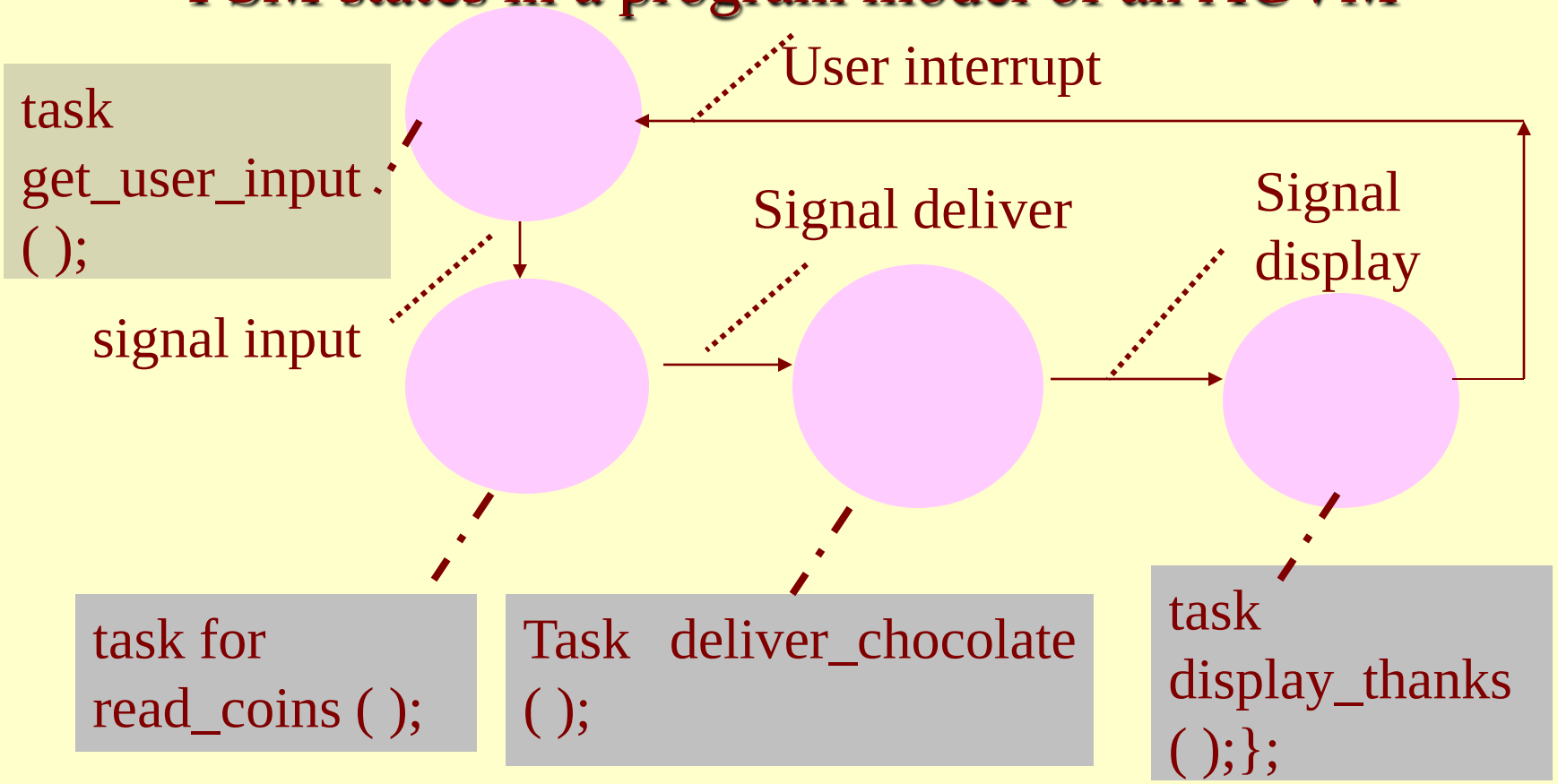
FSM model for an ISR



FSM model for a Task in a Multitasking System



FSM states in a program model of an ACVM



→Arrows show state transition to next state on a software interrupt signal for the transition

Moore and Mealy Models

- A set of outputs represents a state in Moore model
- A set of outputs represents a state-transition in Mealy model

Moore Model

- A flag indicating sate condition or a set of codes being executed or a set of values of certain parameters identifies a next state on mapping the inputs, variables and previous states using the next-state transition function.

Mealy Model

- A state can generate multiple tokens (outputs, messages, flags interrupts or semaphores). An output or set of outputs and variables identifies a next state on mapping the inputs, variables and previous states using the output-state transition (action) function.

FSM

- When there is only one directed incoming arc and one outgoing arc at a state

FSM

- (1) A *state* from a process flow in FSM can lead to any one of the several possible subsequent states. This will depend on the output function and state transition function at the state.

FSM

- (2)There can be several directed graphs from a *state*. But there can only be one directed graph towards a state from one definite state.

Summary

We learnt

- Program modeling for real time (event controlled or response time constrained) programs can be done by the Finite State Machine(s) model

We learnt

- A finite state machine model assumes the finite number of states and reduces the programming tasks to the following. (i) Coding for each state transition function and each output function (ii) Knowing the time periods taken by the process at each state transition function and between each state, when programming for real time.

We learnt

- (ii) The FSM model is appropriate for one process at a time, for the sequential flows from one state to the next state, and for a controlled flow of the program. The FSM is found to easily model the states in processes in real time systems.

End of Lesson 3 of Chapter 8

on

State Machine Programming Models For Event Controlled Programs