Digital System Design ELEC373/473

UNIVERSITY OF LIVERPOOL

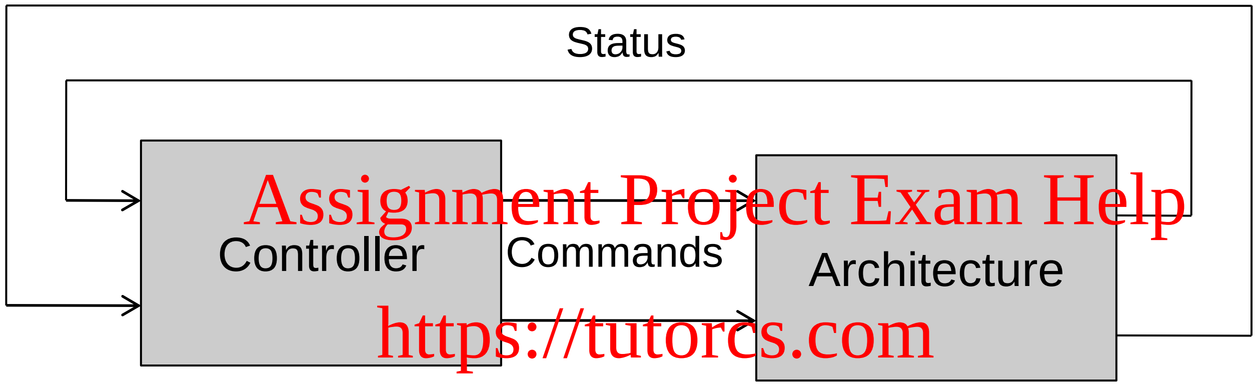
Algorithmic State Machines (ASMs)

(Recap)

Introduction

- As an alternative to using state diagrams, a special type of flow chart, called an algorithmic state machine flow chart or ASM chart, may be used to describe the behaviour of a state machine.
- It is often easier to understand the operation of a digital system by inspection of the ASM chart instead of the equivalent state diagram as the names of the signals are clearly identified.
- The ASM chart is a flowchart whose notation superficially bears a strong resemblance to the conventional software flowchart.
- The ASM chart expresses the concept of a sequence of time intervals in a precise way.
- The software flowchart describes only the sequence of events and not their duration.

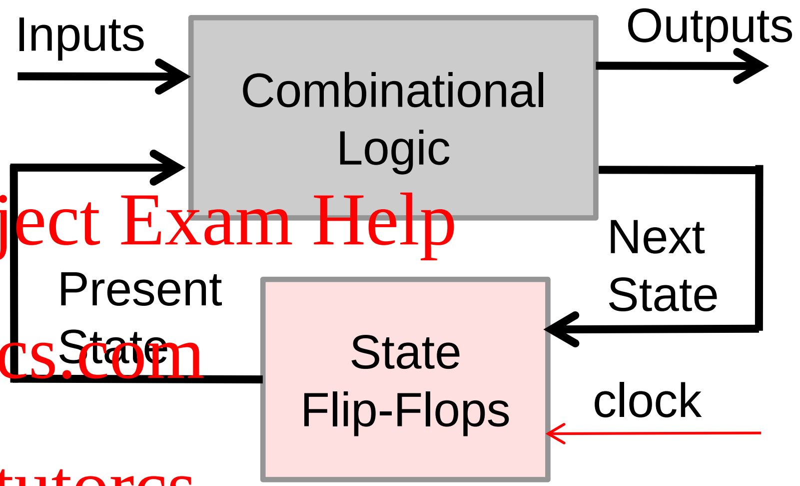
State Machine Structure



- The controller issues properly sequenced commands to the controlled device (the Architecture).
- These commands make the Architecture perform the actions dictated by the control algorithm
- Usually the controller will need status information from the architecture that serves as decision variables for the control algorithm.

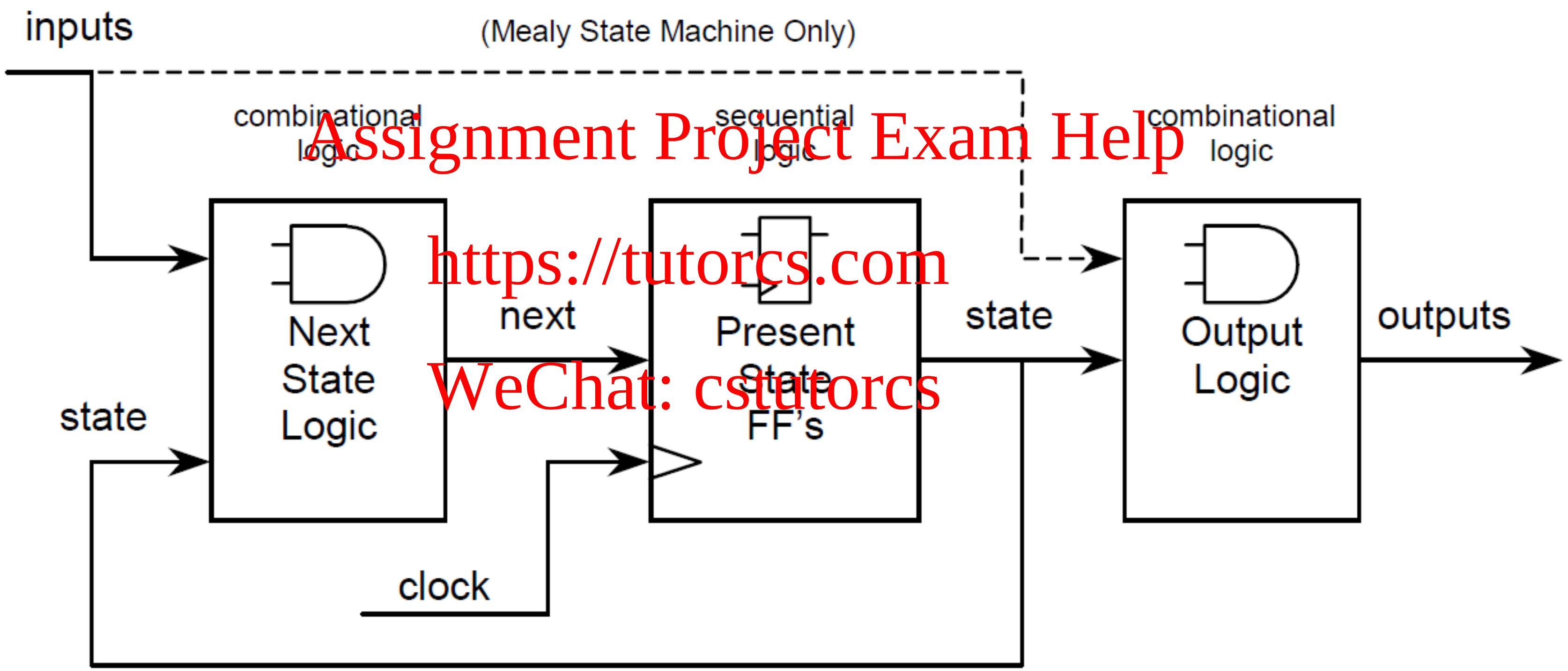
Controller Structure

- All sequential circuits can be divided into a combinational block and a storage element block implemented by Flip-Flops.



- There are two "classes" of state machines:
  - Moore type outputs are a combinational function of only "Present State" signals.
  - Mealy type outputs are a combinational function of both "Present State" and "Input" signals.

Alternative controller view



States and Clock

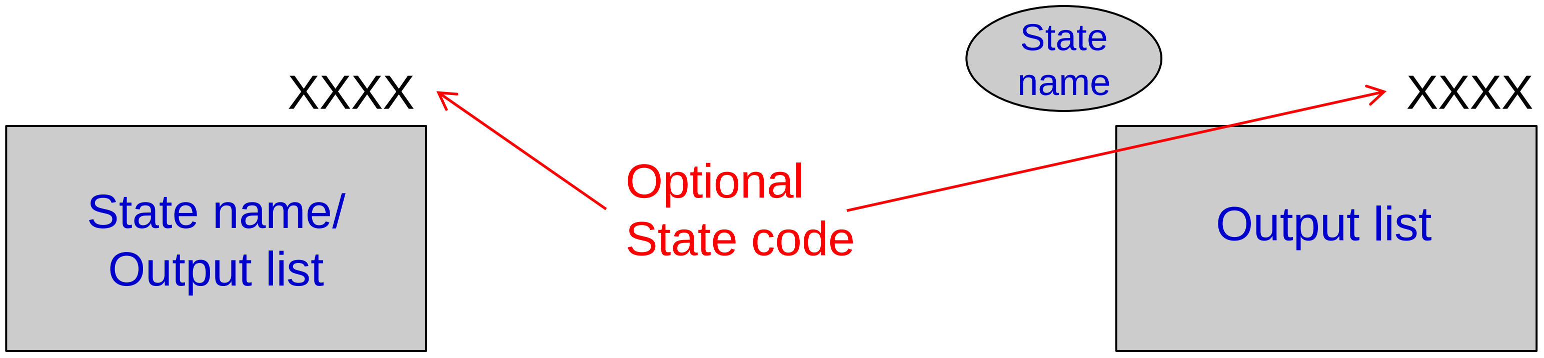
- The algorithmic state machine (ASM) moves through a sequence of states based on the position in the control algorithm (the present state) and the value of the relevant status variables.

- It is the task of the present state of the system to:
  1. Produce any required output signals.
  2. To use appropriate input information to move, at the proper time, to the *next state*.

- In synchronous systems the state transition times are determined solely by the *master clock*.

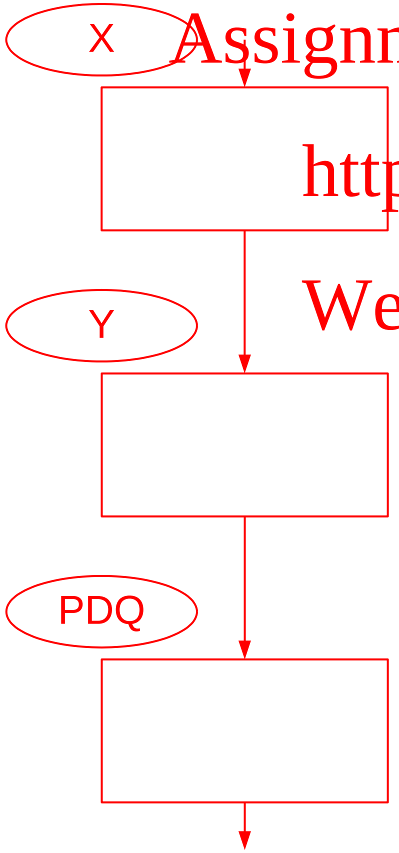
States

- Each active transition of a clock causes a change of state from the *present* state to the *next* state (which could be back to the same state).
- The ASM chart describes the control algorithm such that, given the present state, and the values of the input variables, the next state is determined **unambiguously**.
- The symbol for a state is a rectangle with its symbolic name enclosed in a small circle (or oval) at the upper left corner. (Sometimes the state name is written inside the state box.)
- The *unconditional* outputs are written inside the state box.

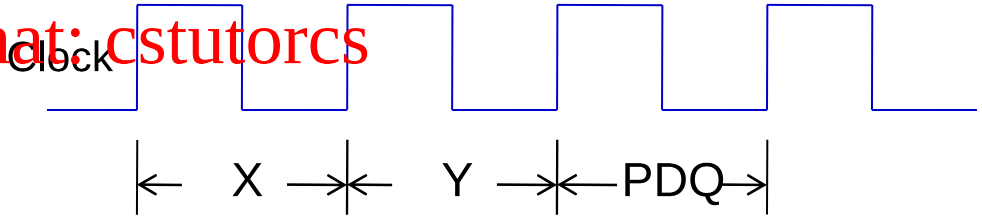


Sequential ASMs

- We could represent a purely sequential algorithm as an ASM chart of a sequence of states.



The timing diagram for this sequence of states is:

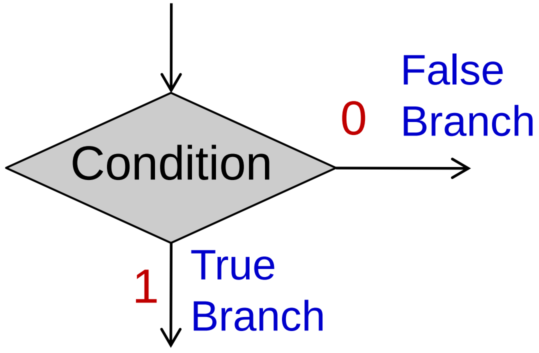


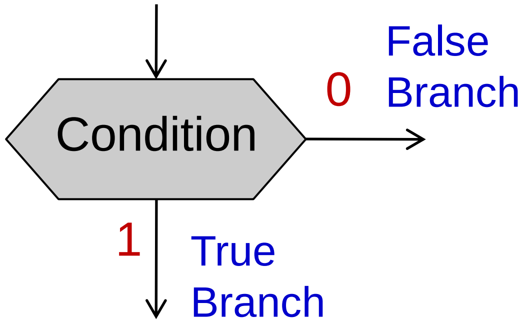
Assuming a positive edge triggered clock, which we will do unless otherwise stated.

You should think of time as rigorously implied in the ASM chart notation.

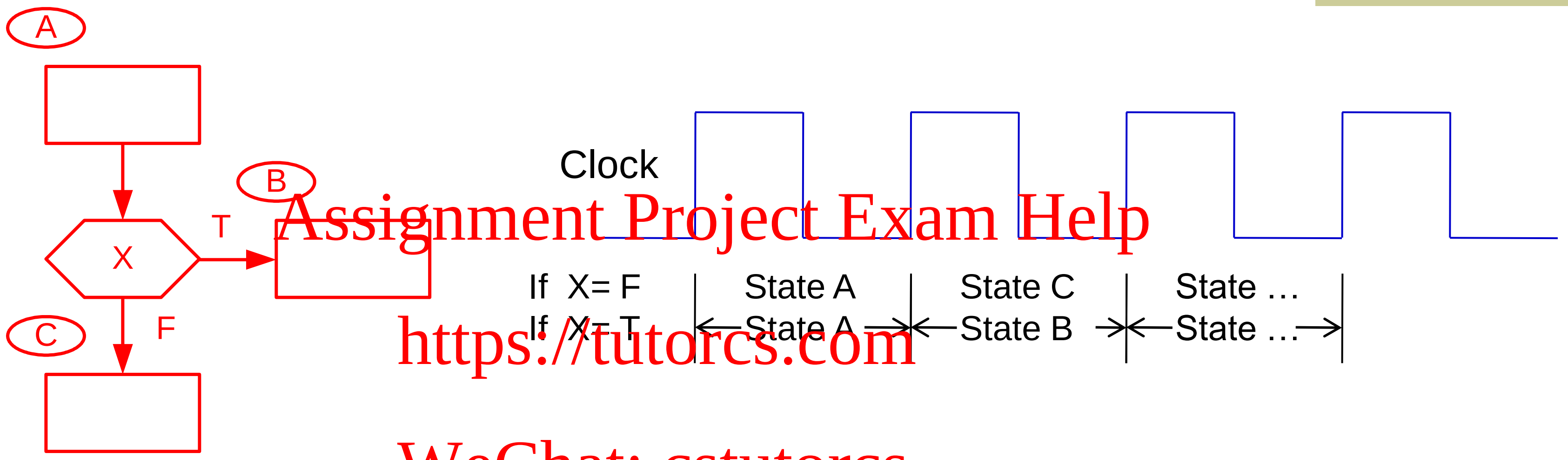
Branches

- Purely sequential ASMs are not usually powerful enough to describe useful algorithms.
- We need some way to express conditional branches so that the next state is determined not only by the present state but also by the present value of one or more test (status) inputs.
- The symbol is the same as in conventional flowcharts for software: the diamond or diamond-sided rectangle.





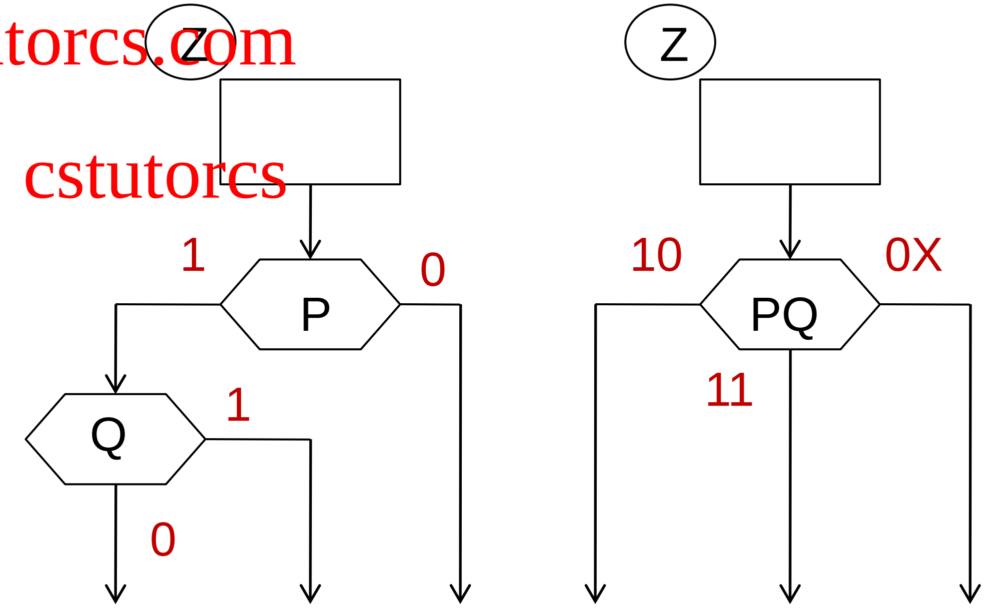
Branches – cont.



- The decision to jump to either state B or state C is made during state A and the jump occurs at the end of state A.
- In hardware implementations the voltage representing the input X must be stable for some period before the decision is made.
- The test does not require a separate clock period, it is done in parallel with the actions of the parent state rectangle and thus is part of the parent state.

Multi-way branches

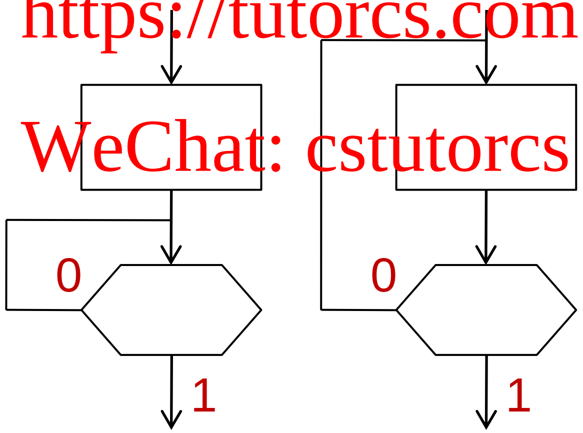
- We may draw a sequence of diamonds or have more than two paths coming from the same diamond.
- Figure (a) conveys the wrong feeling that the test of variable P is of a higher priority than the test of Q.
- For every valid combination of the input variables, there must be exactly one exit path defined.



(a) (b)

A state must be in every path

- At every active clock edge new values will be loaded into the state flip flops.
- The ASM chart should clear state what is the next state.
- Thus there needs to be a state within every path.



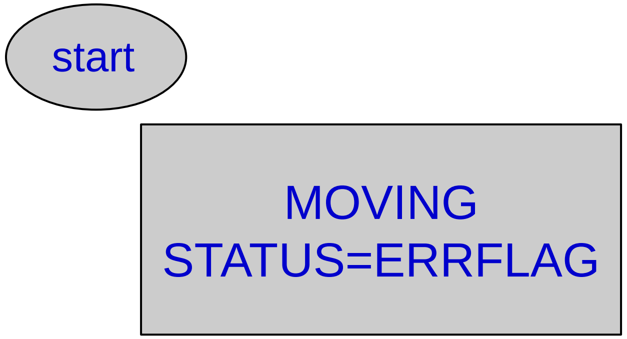
Incorrect Loop   Correct Loop

Unconditional Outputs

- The function of a controller is to send properly sequenced outputs to the controlled device according to some algorithm.
- To indicate an unconditional output, a command description is placed within the appropriate state rectangle.

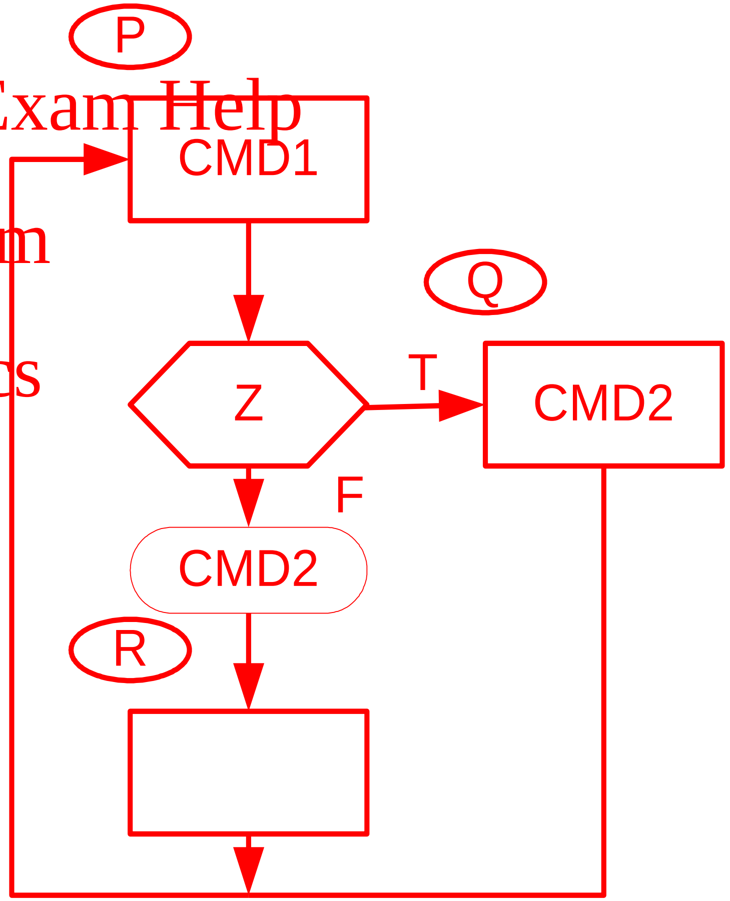
- The first line, MOVING, calls for the assertion of the signal MOVING, during the state, i.e. MOVING = TRUE.

- The last line means that the output STATUS is to have the value of the variable ERRFLAG (T or F) during this state.



Conditional Outputs

- Sometimes we want a command to occur only when some other condition exists.
- We call such a command a conditional output and specify it with an oval.
- Output CMD1 will appear for one state time whenever the ASM is in state P. CMD2 will occur for one state time whenever the ASM is in state Q. Also, when in state P CMD2 will occur if test input Z is false.
- In this example CMD2 is an unconditional output in state Q and a conditional output in state P.



Summary of ASM Symbols

- Test inputs may serve two functions in ASM charts:
    1. They may help specify the next state
    2. They may control the issuing of conditional outputs.
- Ovals for conditional outputs and diamonds for test inputs belong to the parent state; since the activities occur concurrently during the state time.
- A state thus consists of its rectangle, which is always present, and any test diamonds and conditional output ovals associated with that state.
- Unconditional outputs are a function only of the parent state. Conditional outputs depend on both the state and the path within the state.

Homework - Room Light Controller

- Design a digital system that will turn on a light as the first person enters a room, and turn off the light as the last person leaves. Assume that there is a single door fitted with two photocells that generate TTL-compatible outputs.
- One photocell is on the inner side of the door and the other is on the outer side. Light beams shine on each photocell, producing a false output from the cell; a true output from the photocell arises when the light beam is interrupted.
- Assume that once a person starts through the door, the process is completed, and that only one person enters or leaves at a time. Your design should be able to cope with up to 15 people in the room.

- Hints
  - Design from the top down i.e. draw a block diagram.
  - Think what you can add to the architecture to make the controller simpler?
  - My controller design has just 4 states.