CS221: Digital Design

ASM/ FSMD/ RTL Design

A. Sahu

Dept of Comp. Sc. & Engg.

Indian Institute of Technology Guwahati

Outline

- Drawbacks of state diagrams for real systems
- FSMD/ASM
- ASM Specification
- Comparison of FSM Vs ASM
  - Conversion of FSM to ASM, vice versa
- RTL Design

Reference Material for Lec 33, 34, 35

- Chapter 8 of Mano Book
  - Design at Register Transfer Level
  - Classic Example: Booth Multiplication
- Chapter 15 of Kumar Book
  - Algorithmic State Machine

Drawbacks of FSM

- FSM for real systems:
  - Many inputs & many outputs -> awkward to list all of these as each transition arc.
  - On any given arc
    - Typically most inputs are don’t care
    - Typically most outputs are unchanged from the settings in the previous state
    - Tedious & repetitive to list exhaustively

Drawbacks of FSM

- Not a clear structure for illustrating/designing control flow
- What about generic memory/data?
  - Do they really need to be part of the state? If we have many bits of data, this would lead to a huge state
  - **E.g. state diagram for counter or shift register is pointless**
    - **32 bit counter have 2^{32} states**

Drawbacks of FSM

- Some problems analogous to before
- Combinational: Circuit Design Using truth tables (TT); ok/easy for Small Problem
- Adders, Muxes – TT get out of hand
- Design 2 level circuit for a 32 bit Adder
  - 64 inputs and 33 outputs using TT method, worst case delay (2^{63} OR gate)
  - Worst case OR gate size or # of Product term => $2^{N-1}-1$
  - CLA : prefix sum example, use different approach to get the circuit

Drawbacks of FSM: ASM Overview

- Some problems analogous to before
  - Sequential:
    - Small – state diagrams easy
    - Real, Data – state diagrams not helpful
  - **E.g. state diagram for 32 bit counter or shift register is pointless**
    - **32 bit counter have 2^{32} states**
- Solution is
  - ASM/FSMD/RTL Based Design

ASM/FSMD/RTL Based Design

- ASM/FSMD/RTL are similar terms
- ASM : Algorithmic State Machine
- FSMD : FSM with Data Path
- RTL Design : Register Transfer Level Design

- Define what work to be at each clock cycle.

Algorithmic State Machine

Algorithmic State Machine –

- Another representation of a Finite State Machine
- Suitable for FSMs with
  - a larger number of inputs and outputs
- As compared to FSMs expressed using
  - state diagrams and
  - state tables.

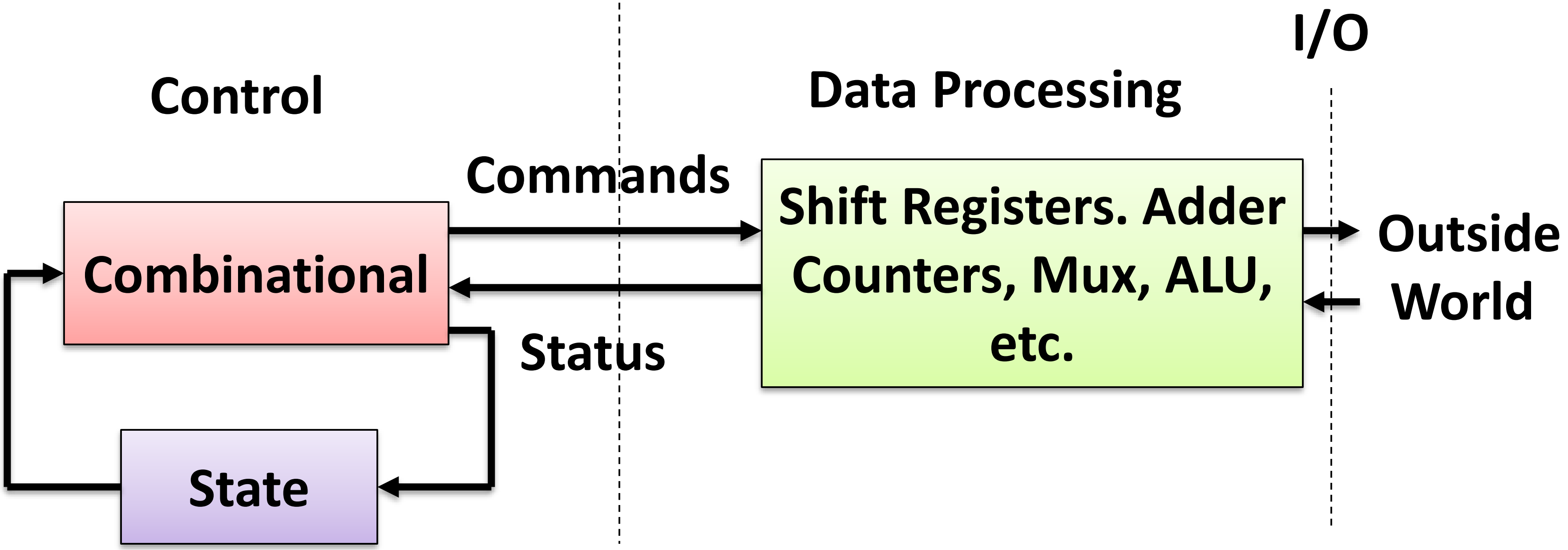
ASM Overview

- We need to separate controller & data processor
  - Controller – What actions need to be taken? What is fundamental operating mode?
  - Processor – Undertake the action. Manipulate the data

The ultimate Goal of this course : Design using Control Path + Data Approach : RTL Design

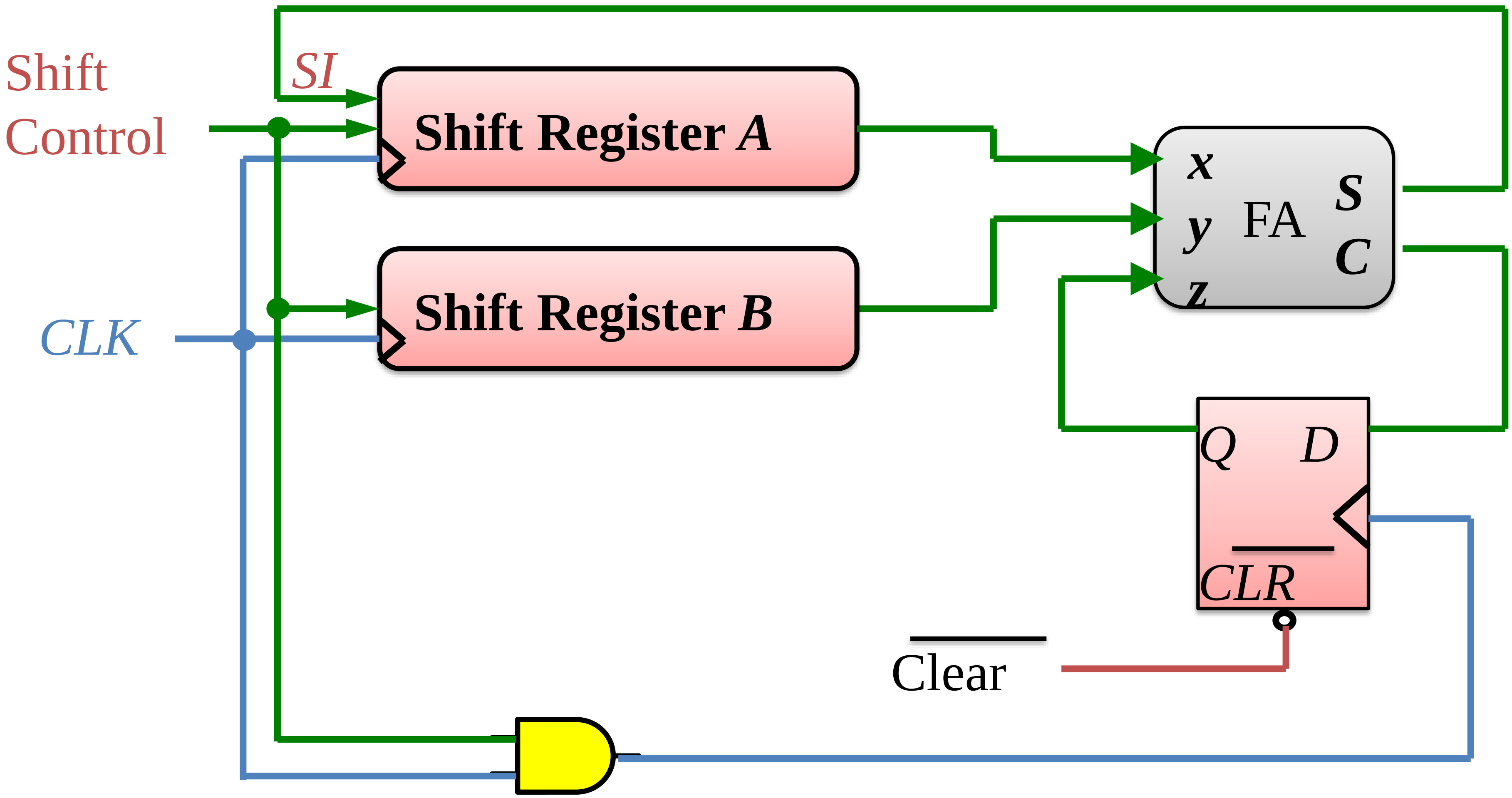
ASM Overview

- Control and data path interaction



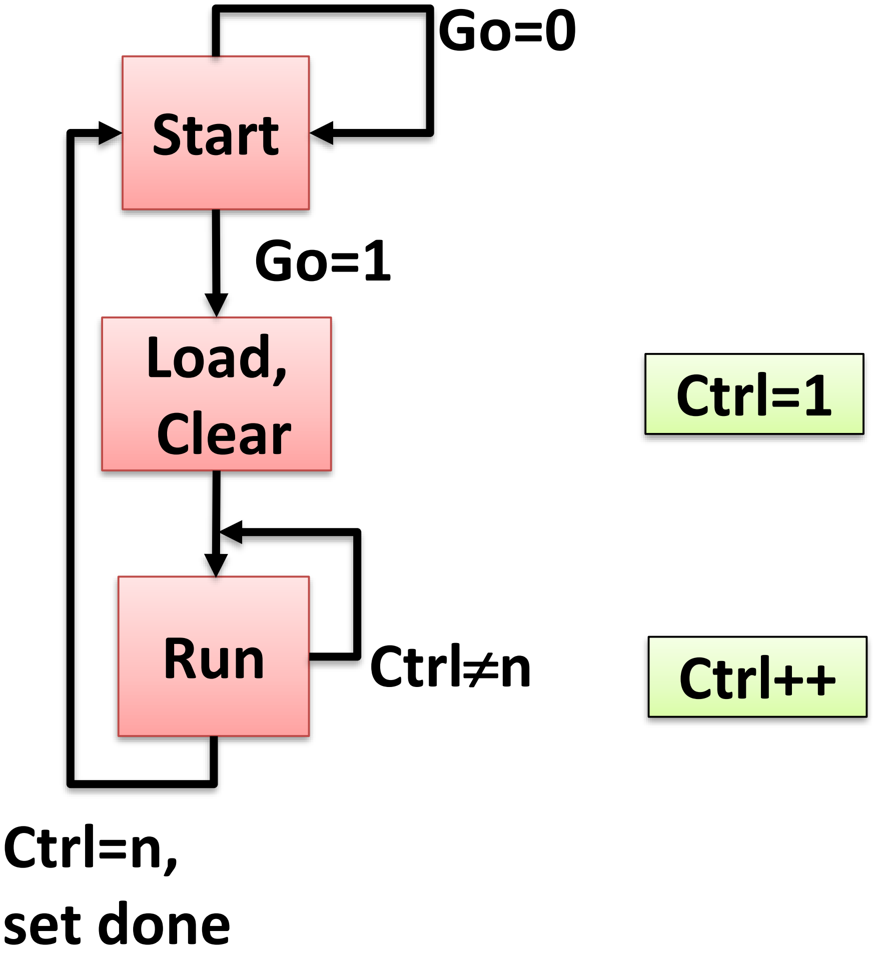
- Our circuit is now explicitly separated

Remember : Serial Addition

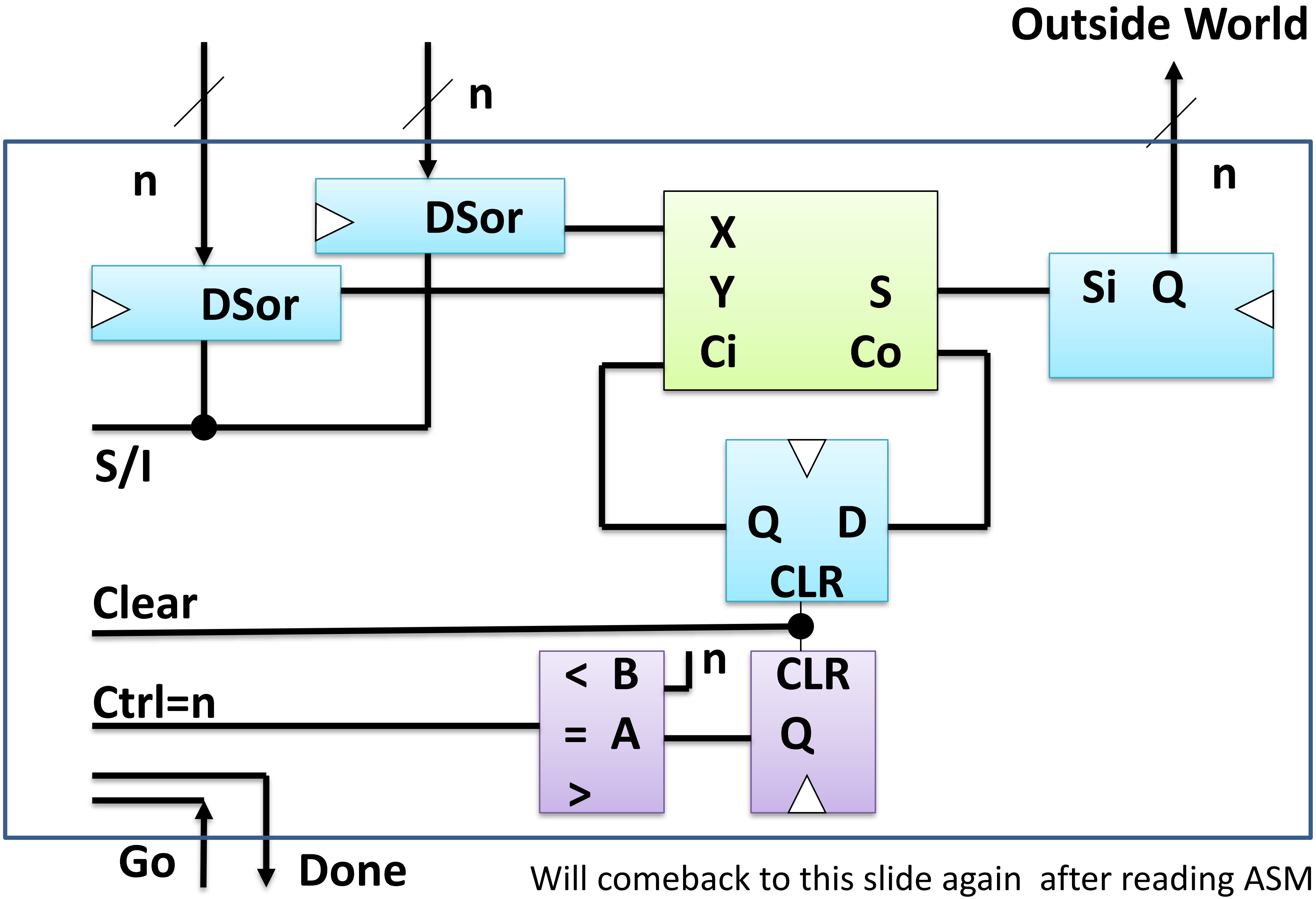


ASM Overview : High Level

- Ex. Serial Addition
- Control Part/Path



Serial Addition : Data Path



Will comeback to this slide again after reading ASM

ASM Design : Data processing

- What sorts of manipulations of the input and output data are requested?
- How many/what sorts of things need to be stored?
- How to design
  - Ad hoc/creative/by insight
  - List requested operations/manipulations
  - Include initialization controls
  - Include status lines

ASM Design : Control logic

- All of the commands to the data proc. logic need to be controlled,
- And the status lines need to be monitored and acted upon.
- ASM charts are like state diagrams, but without specific drawbacks.
  - Don’t list all inputs for each transition – don’t care inputs
  - Don’t list all outputs for each state – not changed outputs

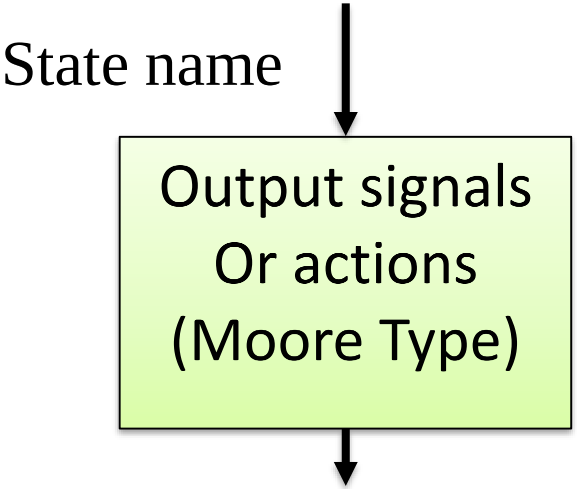
ASM Design

- How to design - ASM chart/state diagram (**for small problems**)
  - State assignment
  - State table
  - Kmap-gates/FF/Reg Mux Dec/EPROM, or, creatively, a combination of them

ASM Design : ASM Chart

- ASM charts are like flowcharts, with a few crucial differences.
- Be careful, especially with timing.
- Three type components/Box
  - **State Box**
  - **Decision Box**
  - Combinational Box/Transition Box/**Conditional Box**

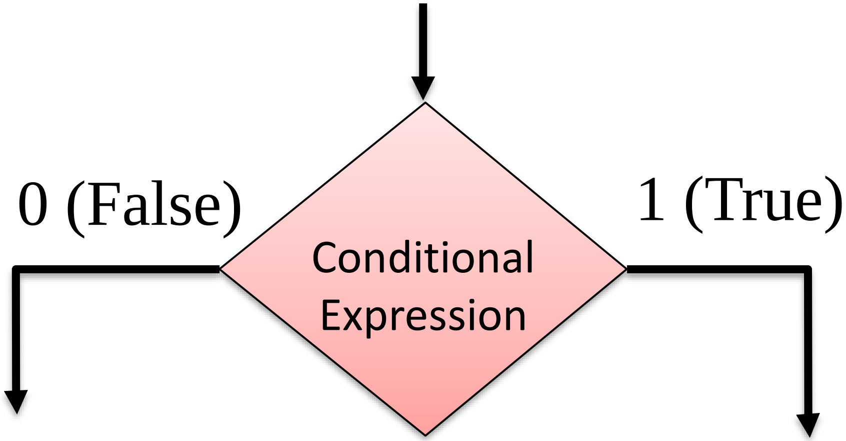
ASM charts : 3 Elements used



State name

Output signals
Or actions
(Moore Type)

(a) State box

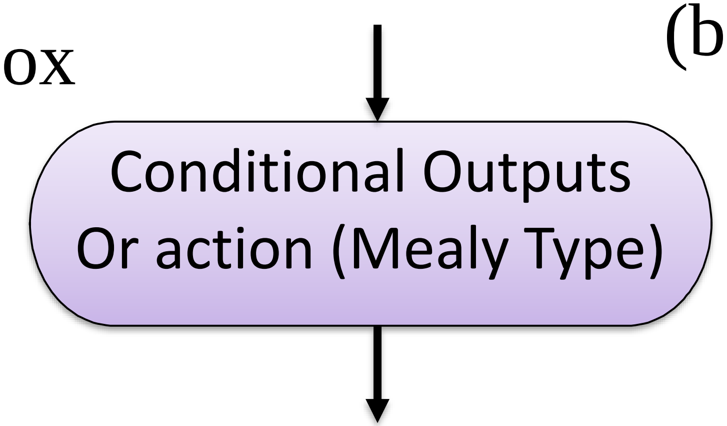


0 (False)

Conditional
Expression

1 (True)

(b) Decision box

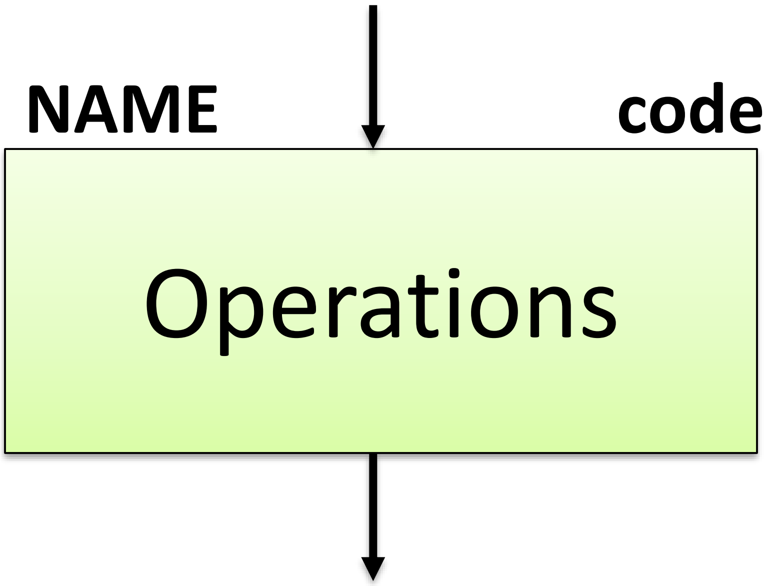


Conditional Outputs
Or action (Mealy Type)

(c) Conditional output box

ASM Design : State Box

- State Box – one box per system state



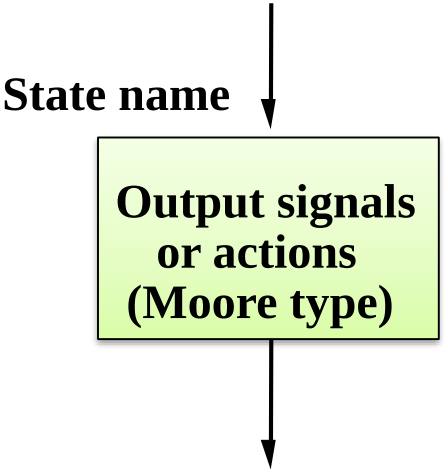
i.e. optional binary state code

ASM Design : State Box

- Operation notation:
  - Sum <- 0 or Carry <- 0 or LOAD A
  - Combinational variable: S=0, T=S+V
- Idea: keep operations abstract & high level.
  - Don’t work in detailed language of processing logic (i.e. write Sum <- 0, not $\text{CLR}_{\text{Sum Reg}}$=1)
- Operations will take place at the end of the clock period

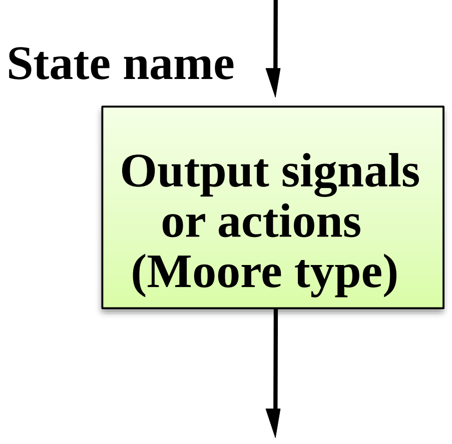
ASM: State Box

- **State box** – represents a state.
- Equivalent to a node in a state diagram or a row in a state table.
- Contains **register transfer** actions or output signals
- **Moore-type outputs are listed inside of the box.**



ASM: State Box

- It is customary to write only the name of the signal that has to be asserted in the given state,
  - e.g., z instead of z<=1.
- Also, it might be useful to write an action to be taken,
  - e.g., count <= count + 1,
- And only later translate it to asserting a control signal that causes a given action to take place
  - (e.g., enable signal of a counter).

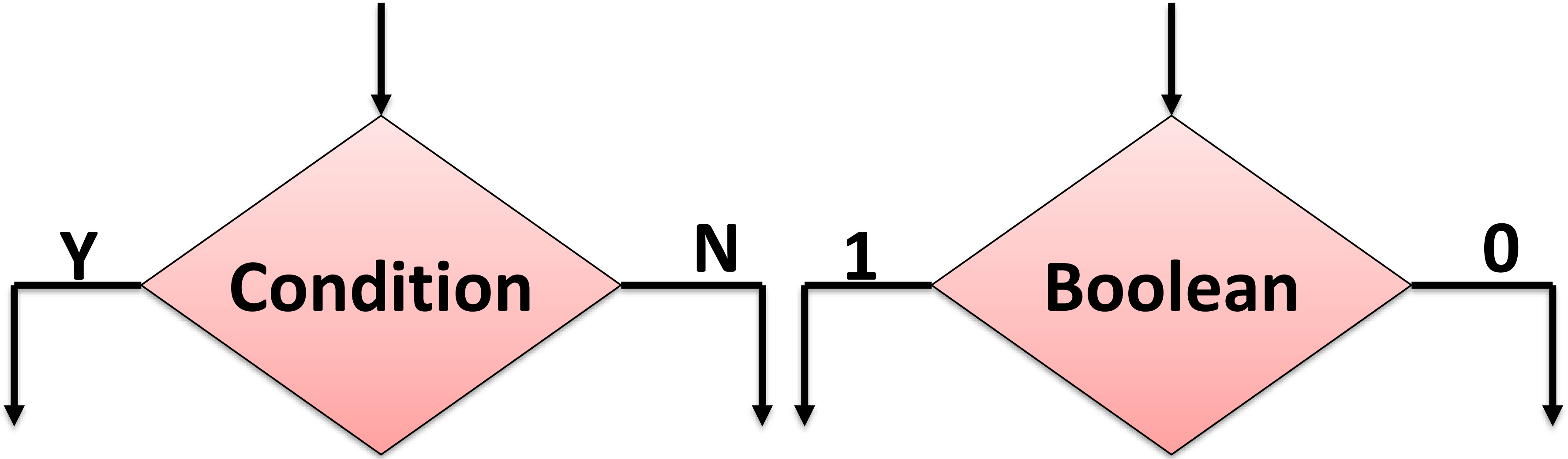


State name

Output signals or actions (Moore type)

ASM Design : Decision Box

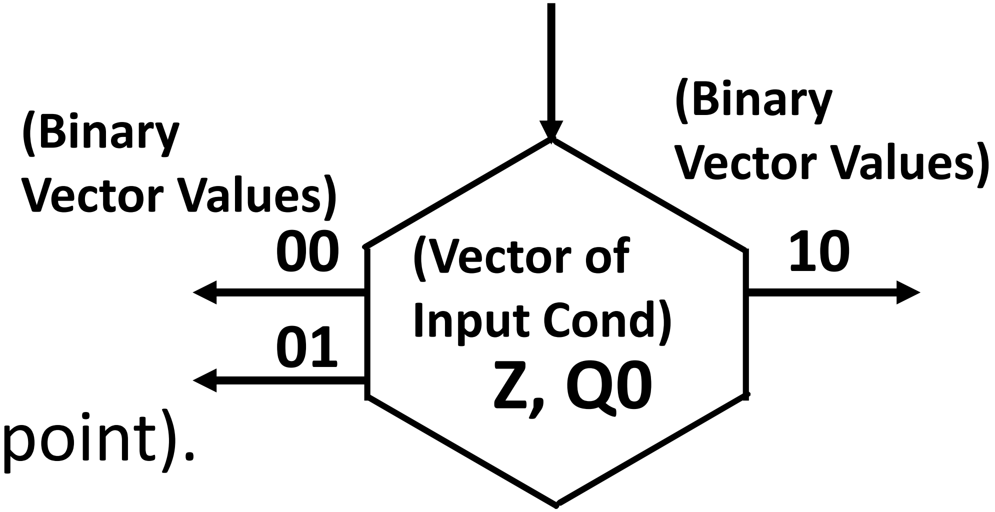
- Decision Box - Basic condition, i.e. logic flow control.
- Only the decision boxes depend on inputs.



ASM Design : Decision Box

- **Decision box** – indicates that
  - a given condition is to be tested and
  - the exit path is to be chosen accordingly
- The condition expression may include
  - One or more inputs to the FSM.

Vector Decision Box



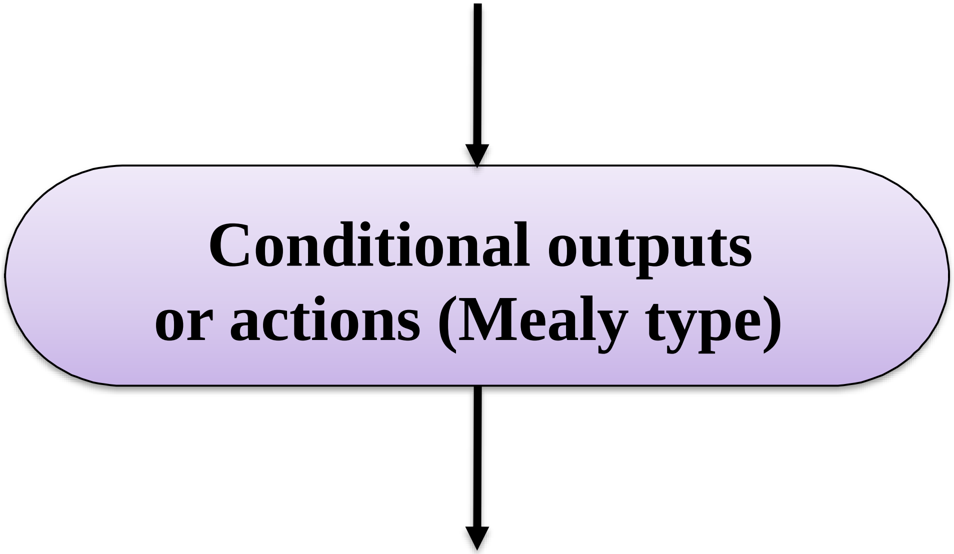
- A hexagon with:
  - One Input Path (entry point).
  - A vector of input conditions, placed in the center of the box, that is tested.
  - Up to 2^n output paths. The path taken has a binary vector value that matches the vector input condition

ASM Design

- Keep conditions as general as possible.
- Prefer: Carry high? Over $Q_{FF\#5}$=1?

ASM Design: Conditional Box

- Conditional Box - An action/operation
  - – to be undertaken conditioned on some earlier decision box.

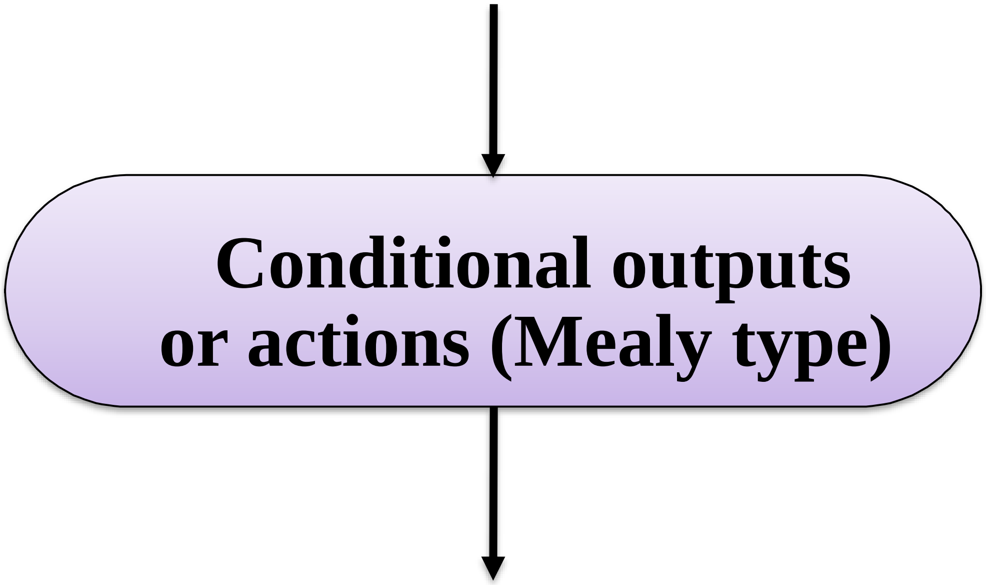


ASM Design Vs Flowchart

- Conditional boxes do not appear in normal flowcharts.
- The essential difference is timing:
  - Flowcharts are sequential
  - ASM charts are not. All of the operations associated with a given state take place simultaneously.

ASM Design: Conditional Output Box

- **Conditional output box**
- Denotes output signals that are of the Mealy type.
- The condition that determines whether such outputs are generated is specified in the decision box.



ASMs representing simple FSMs

- Algorithmic state machines can model both
  - Mealy FSM
  - Moore Finite State Machines
- **They can also model machines that are of the mixed type**