CS221: Digital Design

http://jatinga.iitg.ernet.in/~asahu/cs221

FSMD and ASM

A. Sahu

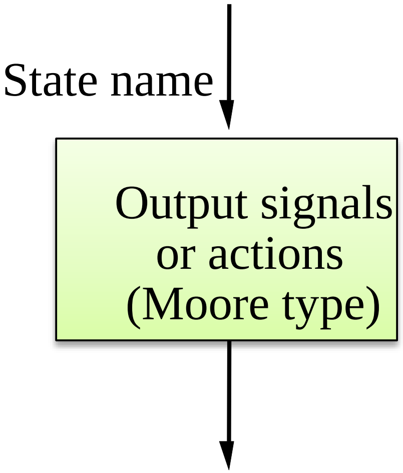
Dept of Comp. Sc. & Engg.

Indian Institute of Technology Guwahati

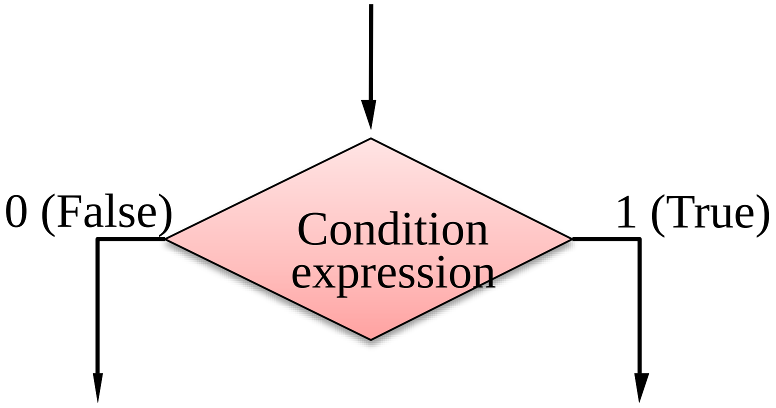
ASM Design

- 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**

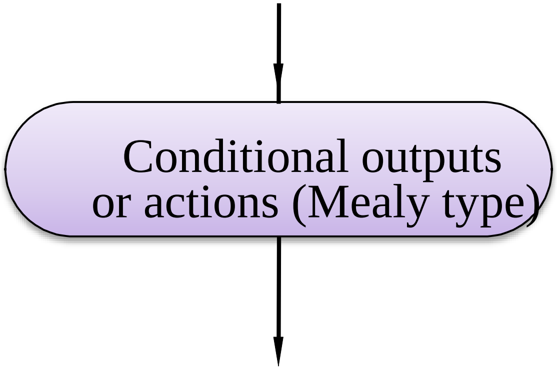
Elements used in ASM charts



(a) State box



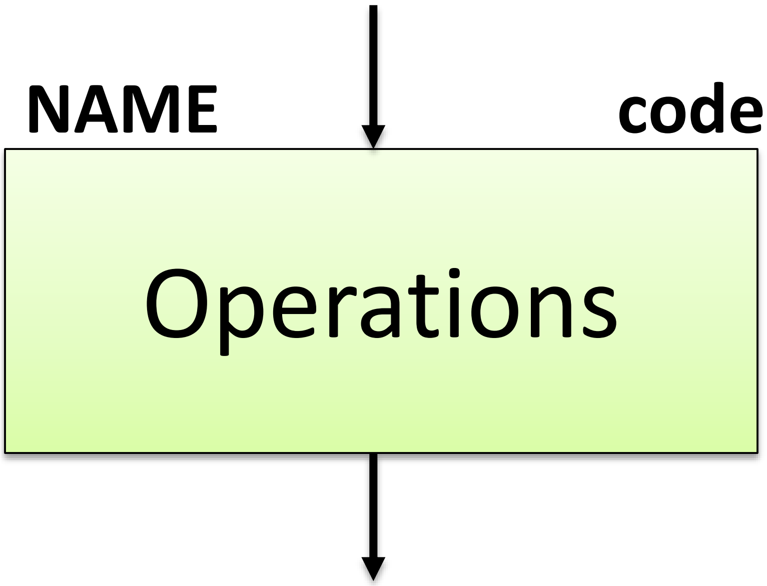
(b) Decision box



(c) Conditional output box

ASM Design : State Box

- State Box – one box per system state



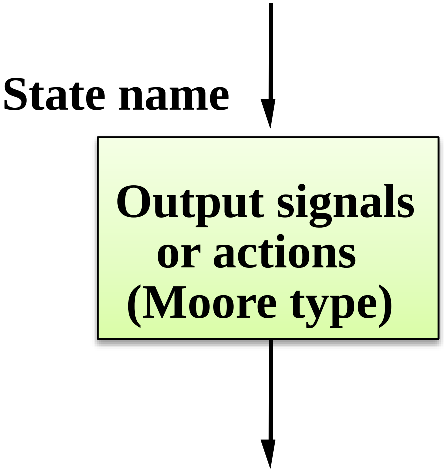
i.e. optioinal binary state code

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 $CLR_{Sum\ Reg}=1$)
- Operations will take place at the end of the clock period

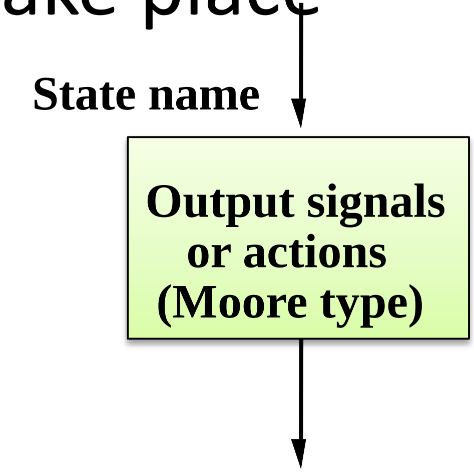
ASM: State Box

- **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).

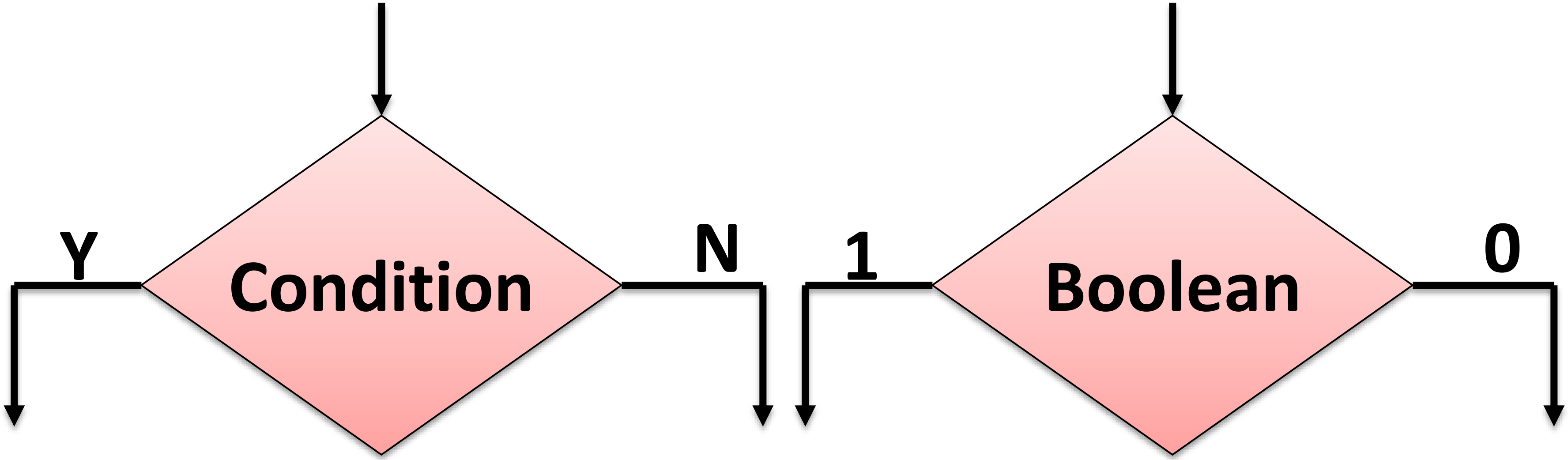


State name

Output signals or actions (Moore type)

ASM Design : Decision Box

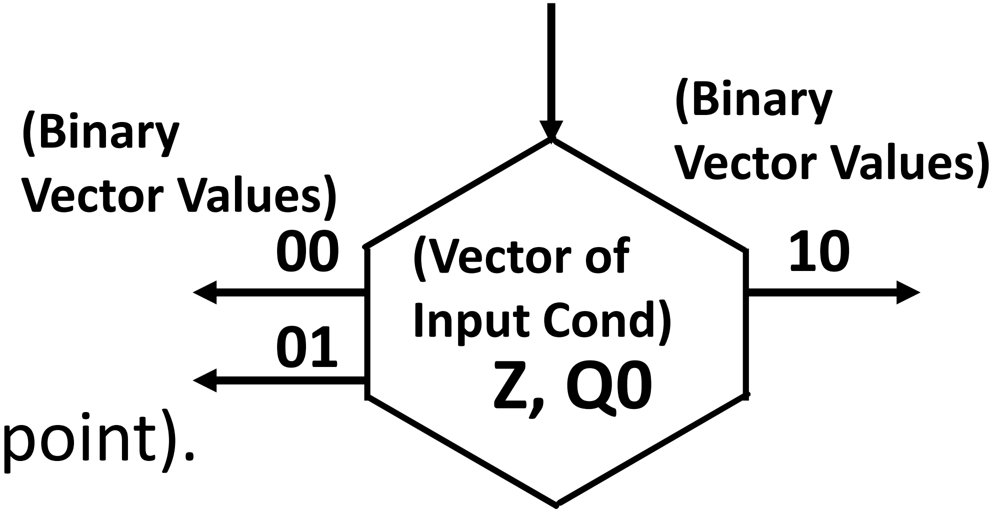
- Decision Box - Basic condition, i.e. logic flow control.
- Only the decision boxes depend on inputs.



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



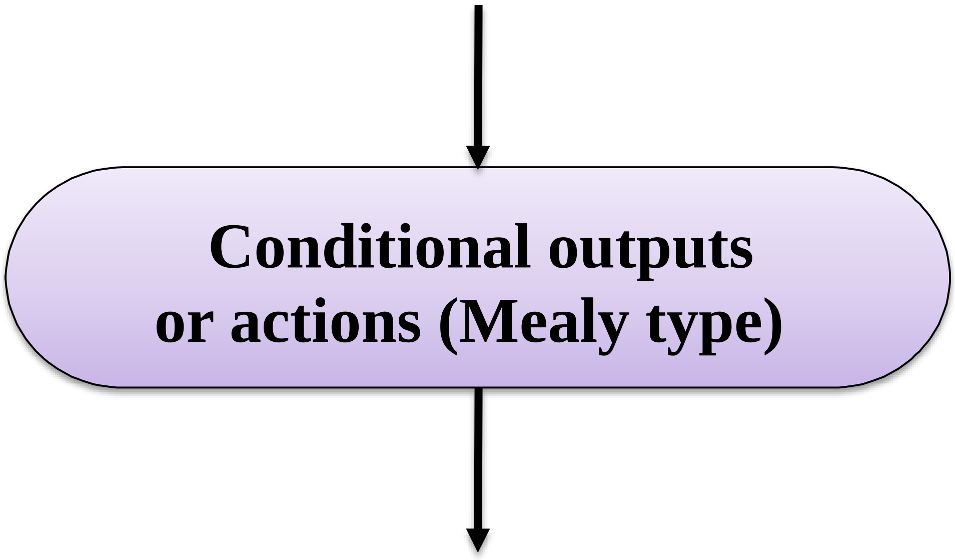
- A hexagon with:
  - One Input Path (entry point).
  - A vector of input conditions, placed in the center of the box, that is tested.
  - Up to 2^n output paths. The path taken has a binary vector value that matches the vector input condition

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.



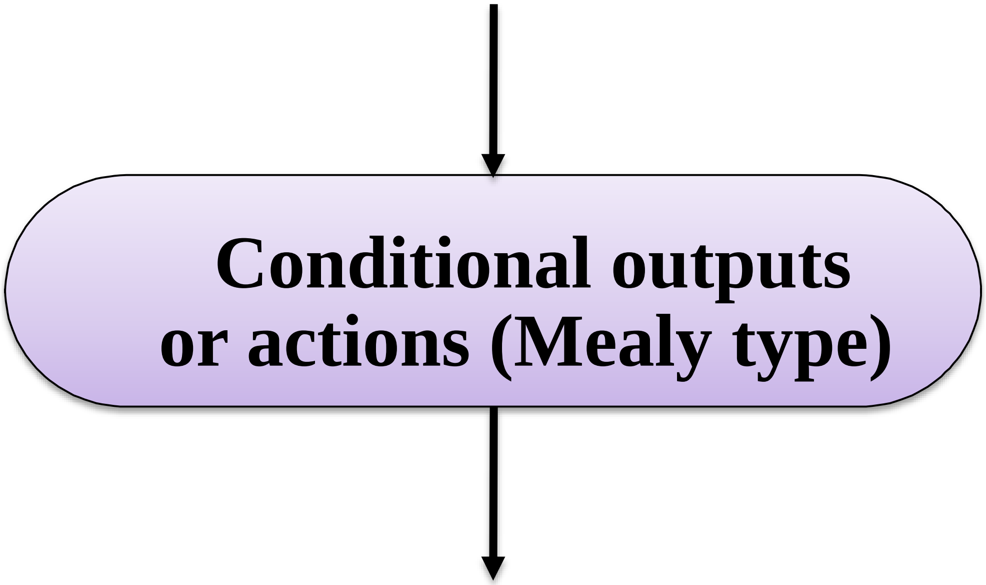
Conditional outputs
or actions (Mealy type)

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

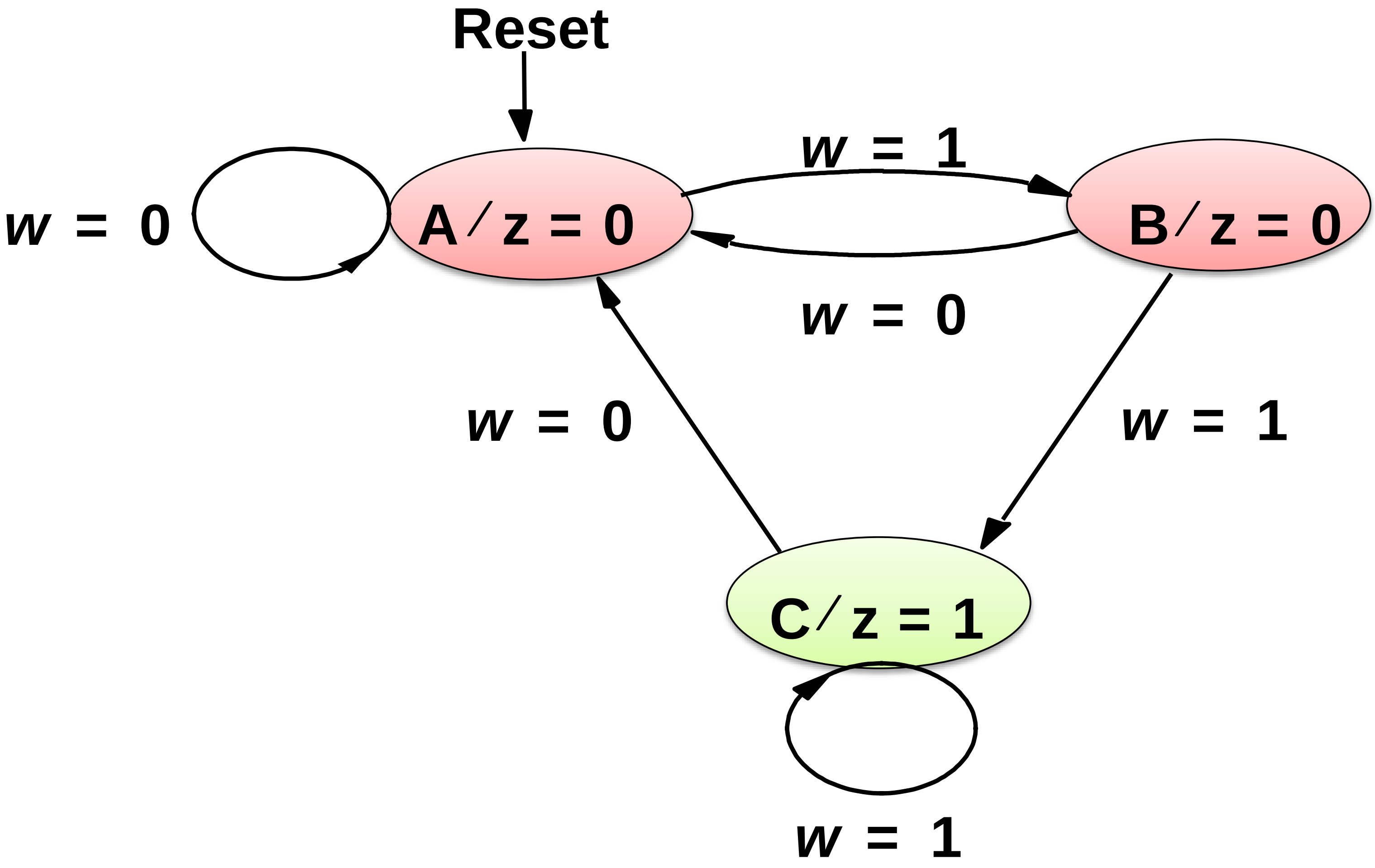
- **Conditional output box**
- Denotes output signals that are of the Mealy type.
- The condition that determines whether such outputs are generated is specified in the decision box.



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**

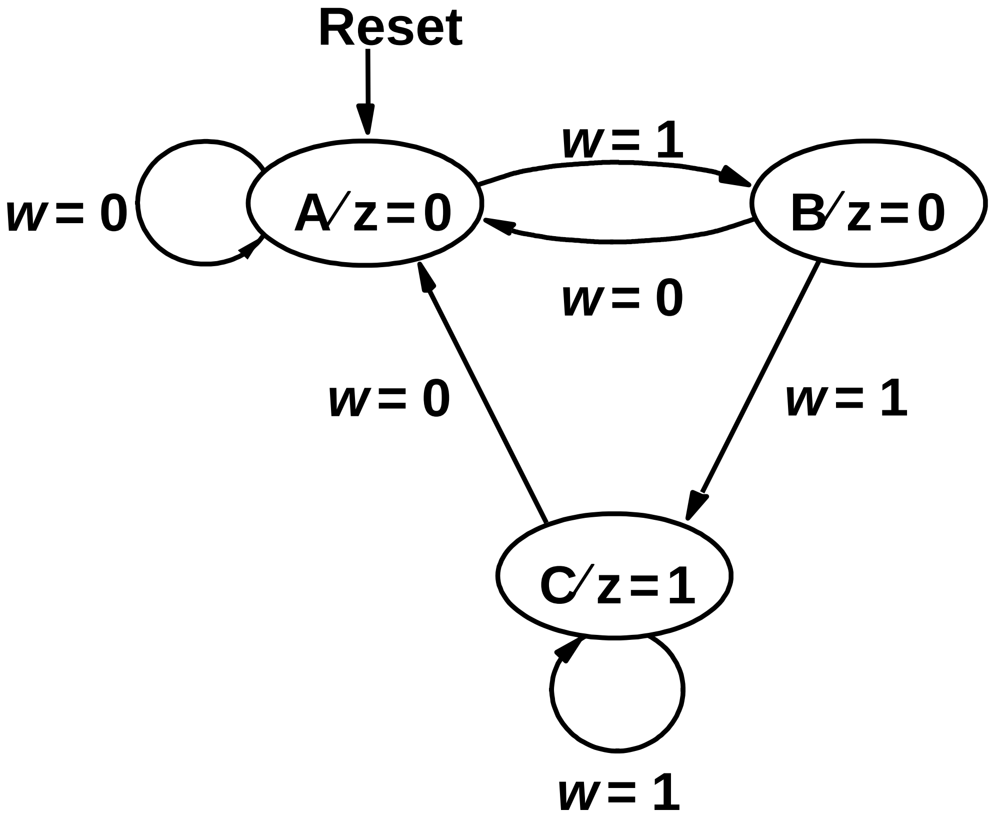
Moore FSM – Example 2: Sequence of two 1’s

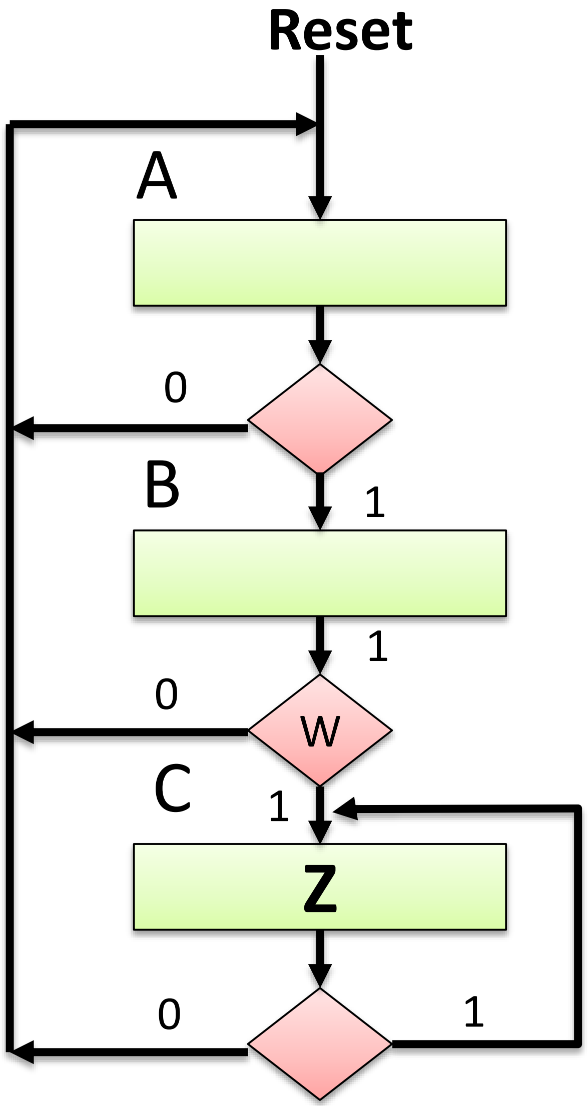


Moore FSM – Example 2: Sequence of two 1’s

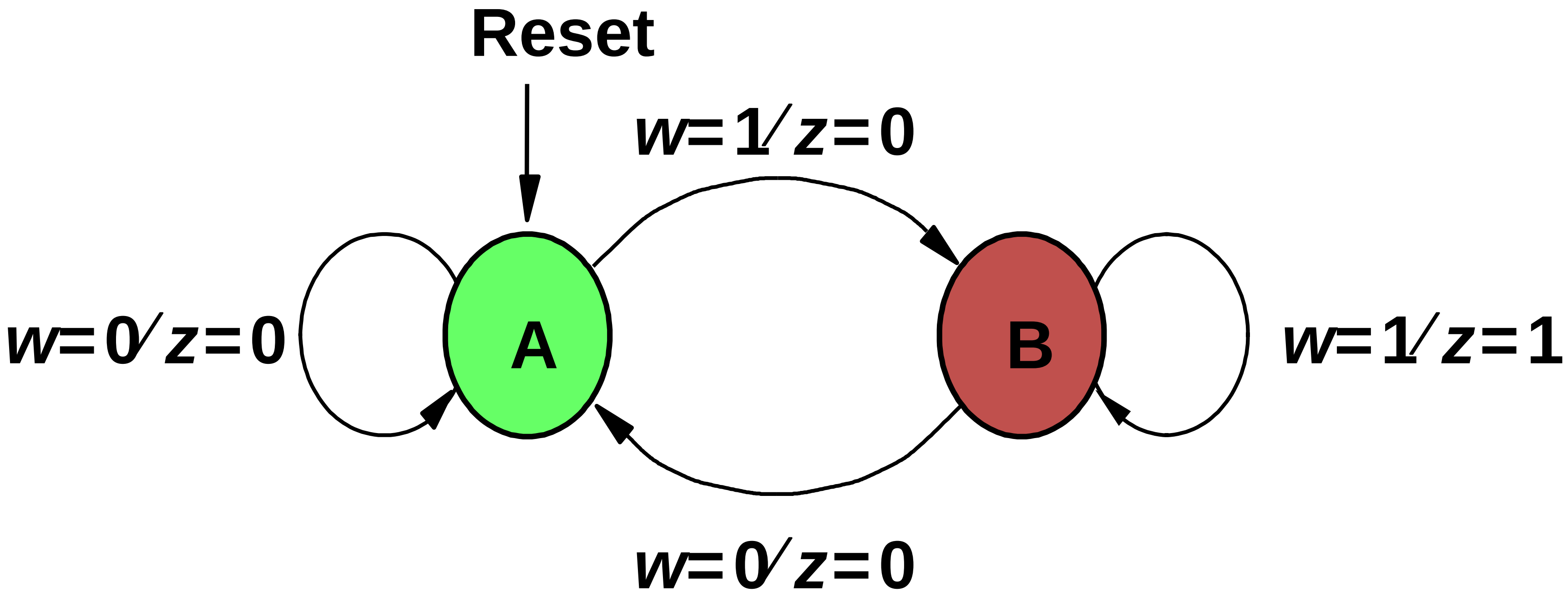
Present state	Next state		Output
	$w = 0$	$w = 1$	z
A	A	B	0
B	A	C	0
C	A	C	1

ASM Chart for Moore FSM

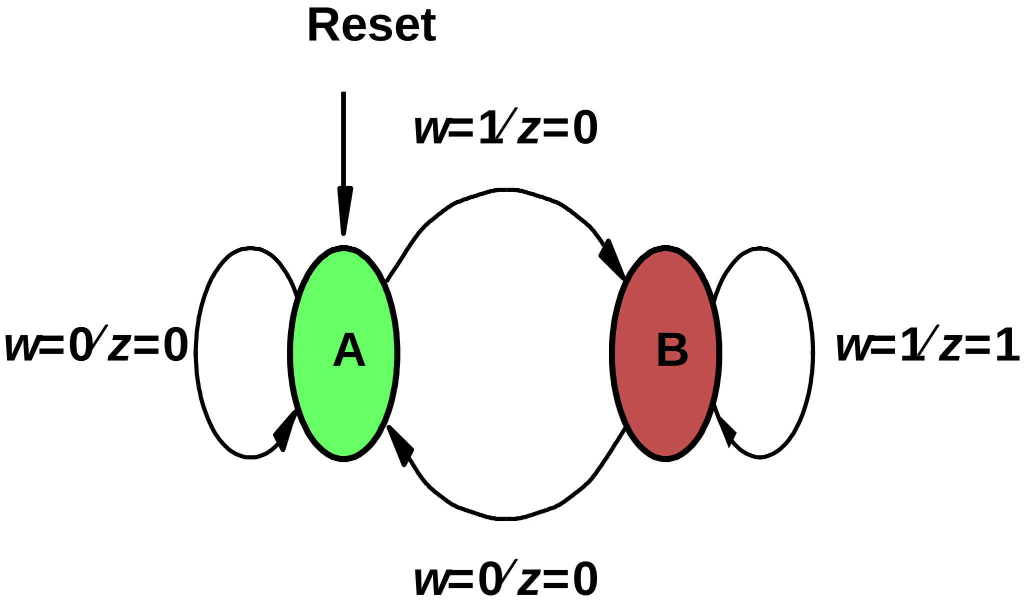


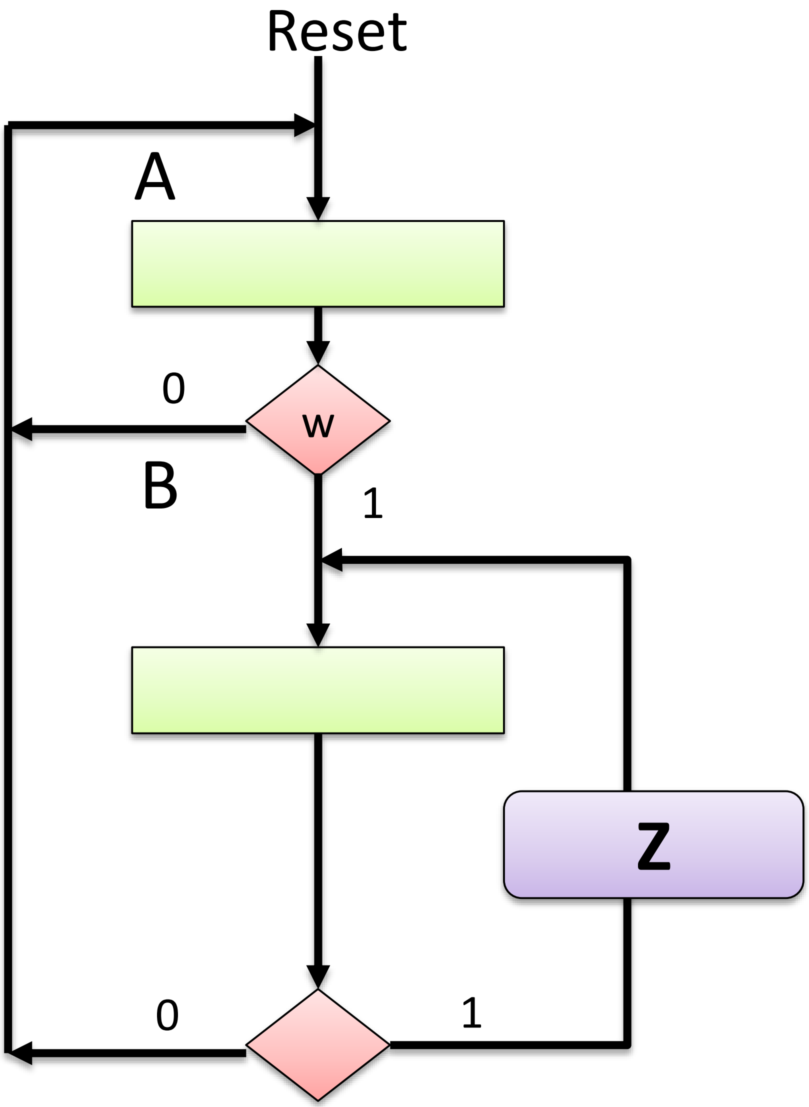


Mealy FSM –Example 3: Sequence of two 1's



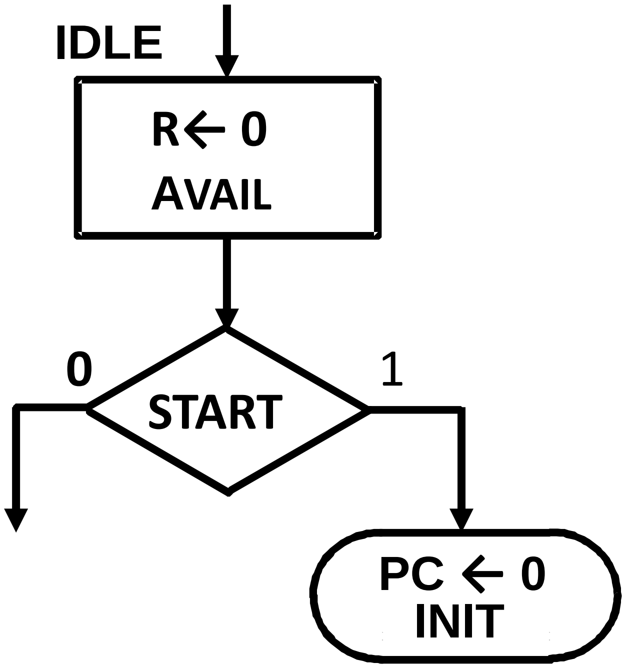
ASM Chart for Mealy FSM – Example 3





ASM: Connecting All type Boxes Together

- By connecting boxes together, we begin to see the power of expression.
- What are the:
  - Inputs?
  - Outputs?
  - Conditional Outputs?
  - Transfers?
  - Conditional Transfers?



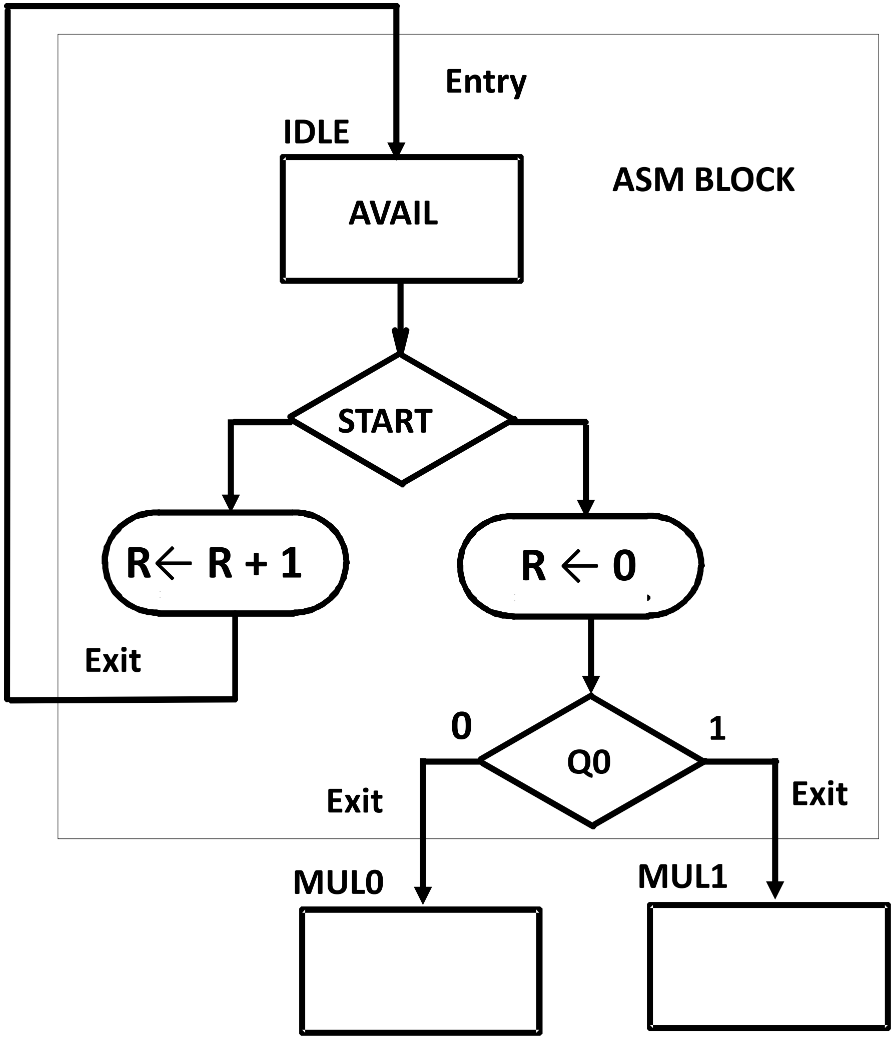
ASM Blocks

- One state box along with all decision and conditional output boxes connected to it is called an ASM Block.
- The ASM Block includes all items on the path from the current state to the same or other states.

ASM Blocks

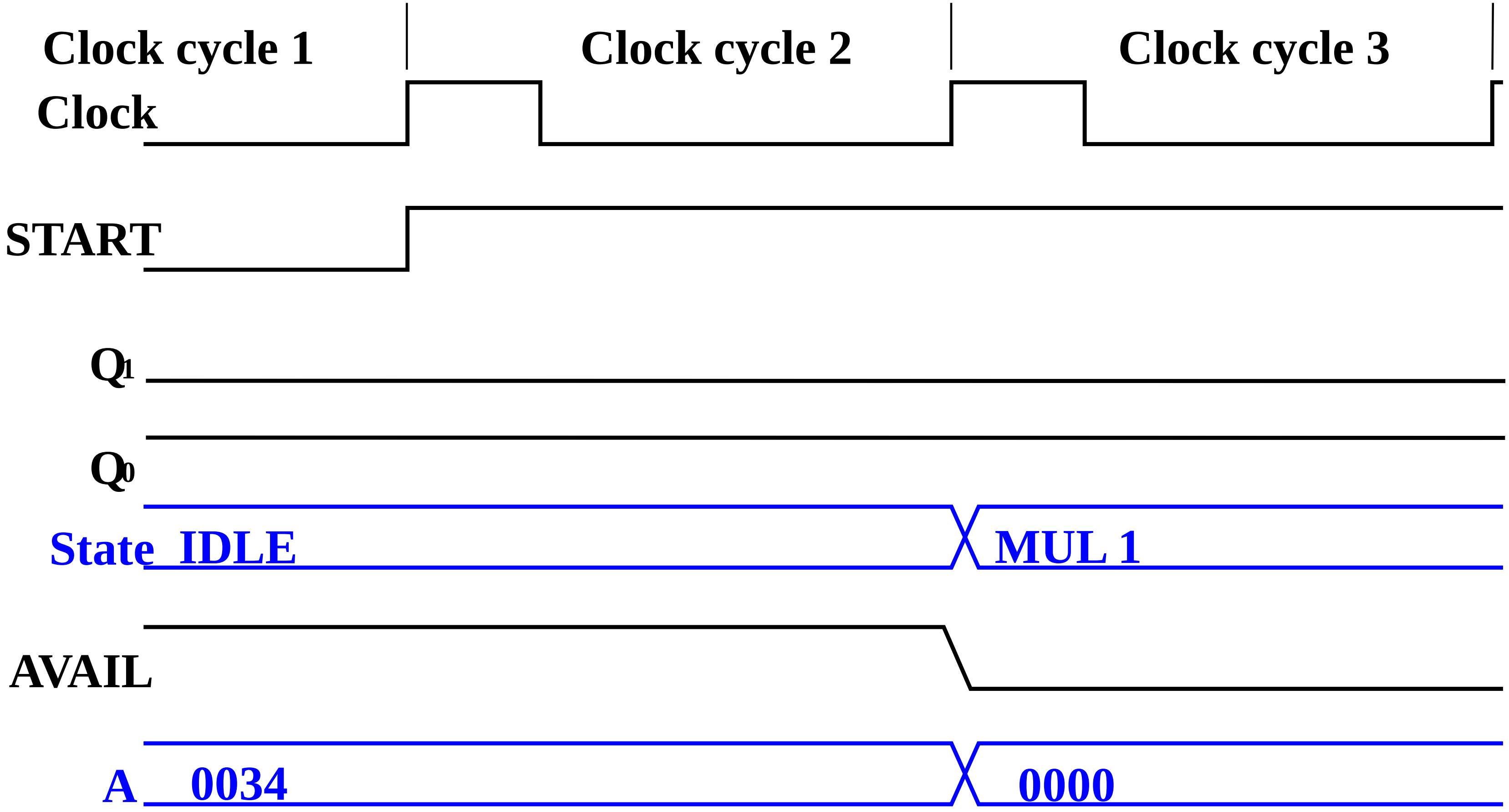
One ASM block

execute in one cycle

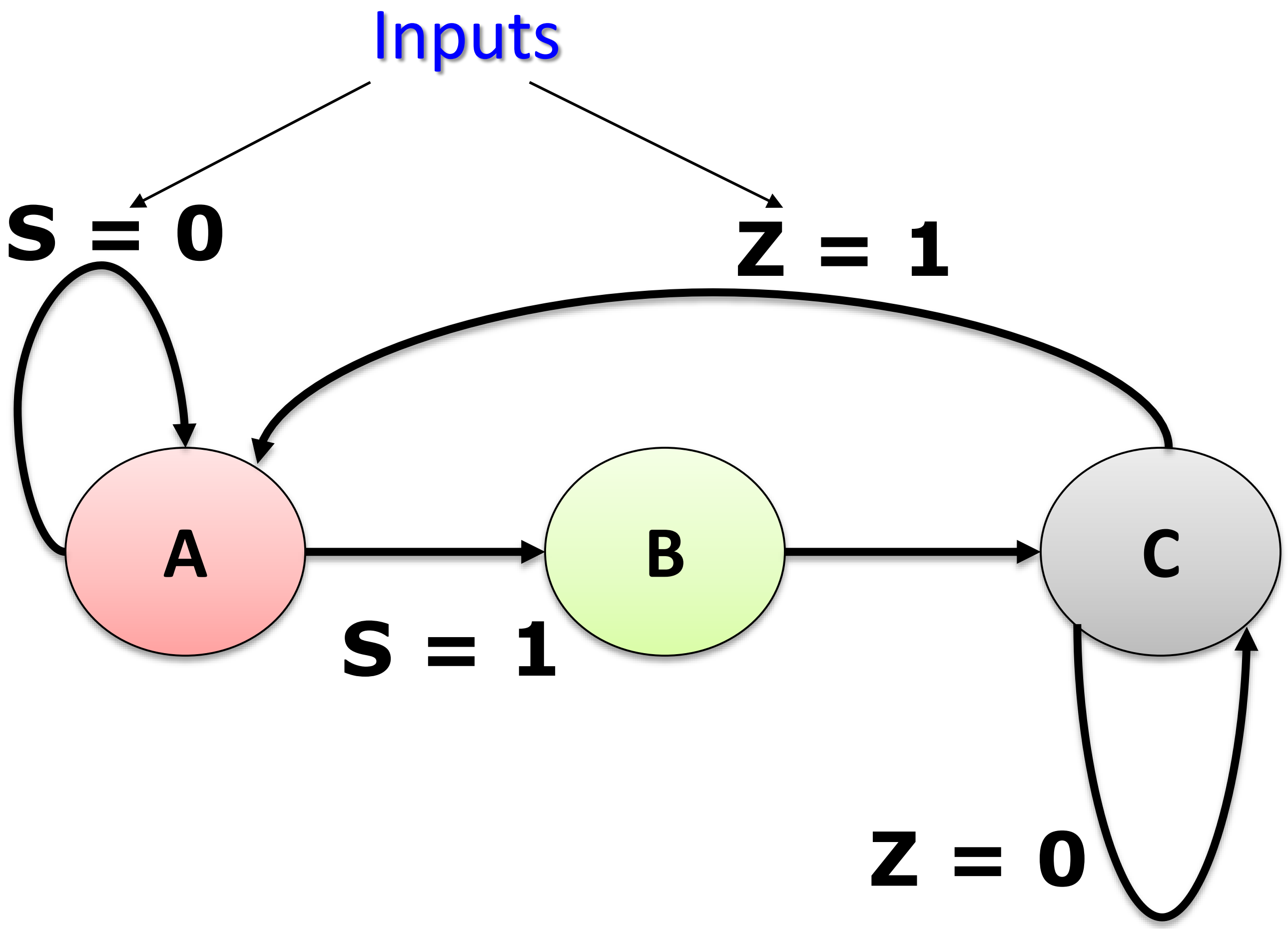


ASM Timing

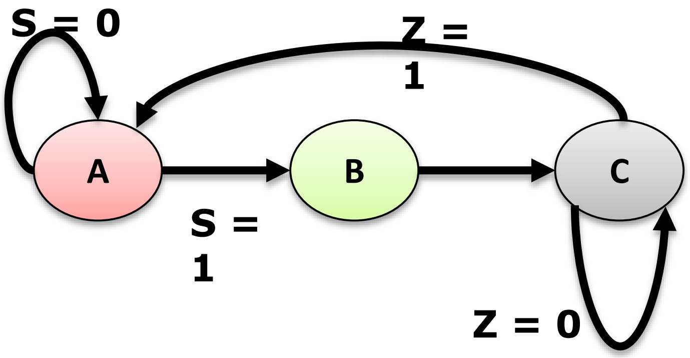
- Outputs appear while in the state
- Register transfers occur at the clock while exiting the state - New value occur in the next state!

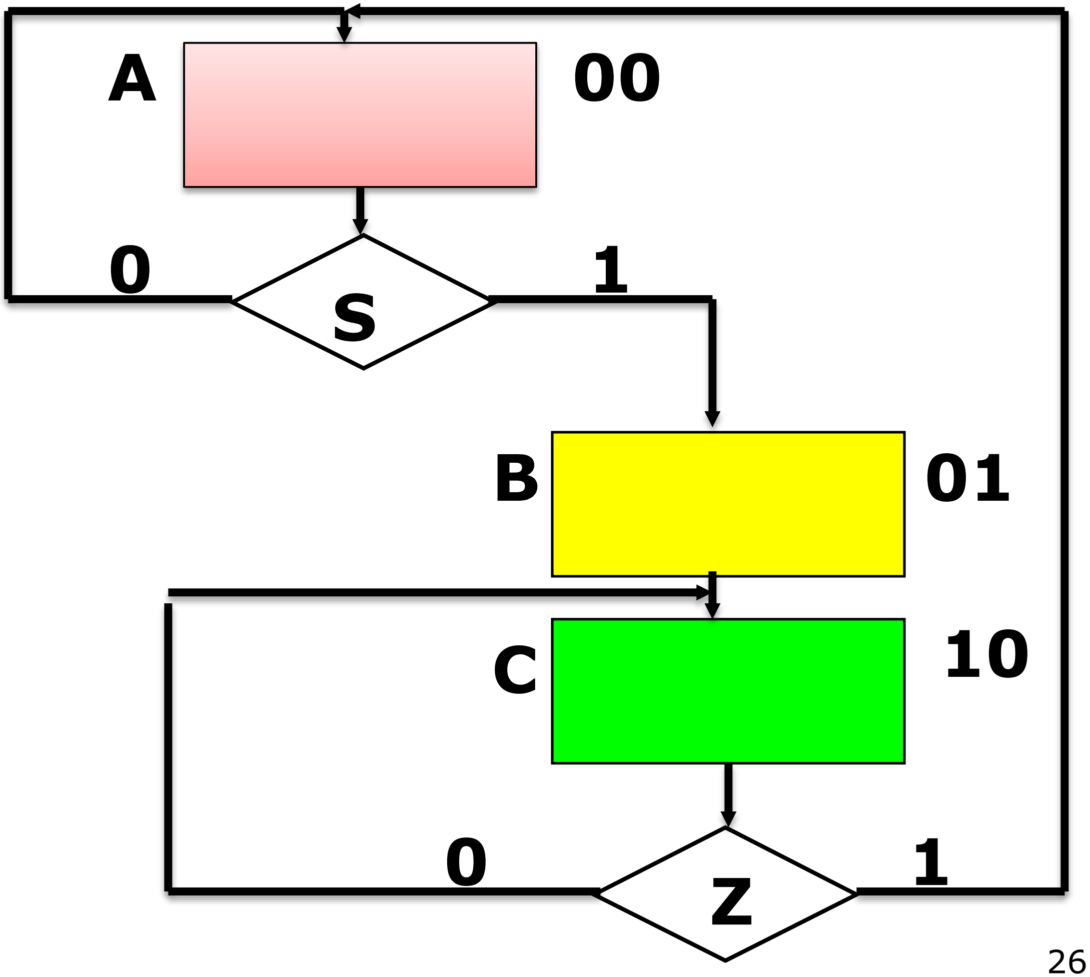


Another Example: From FSM to ASM



Another Example: From FSM to ASM

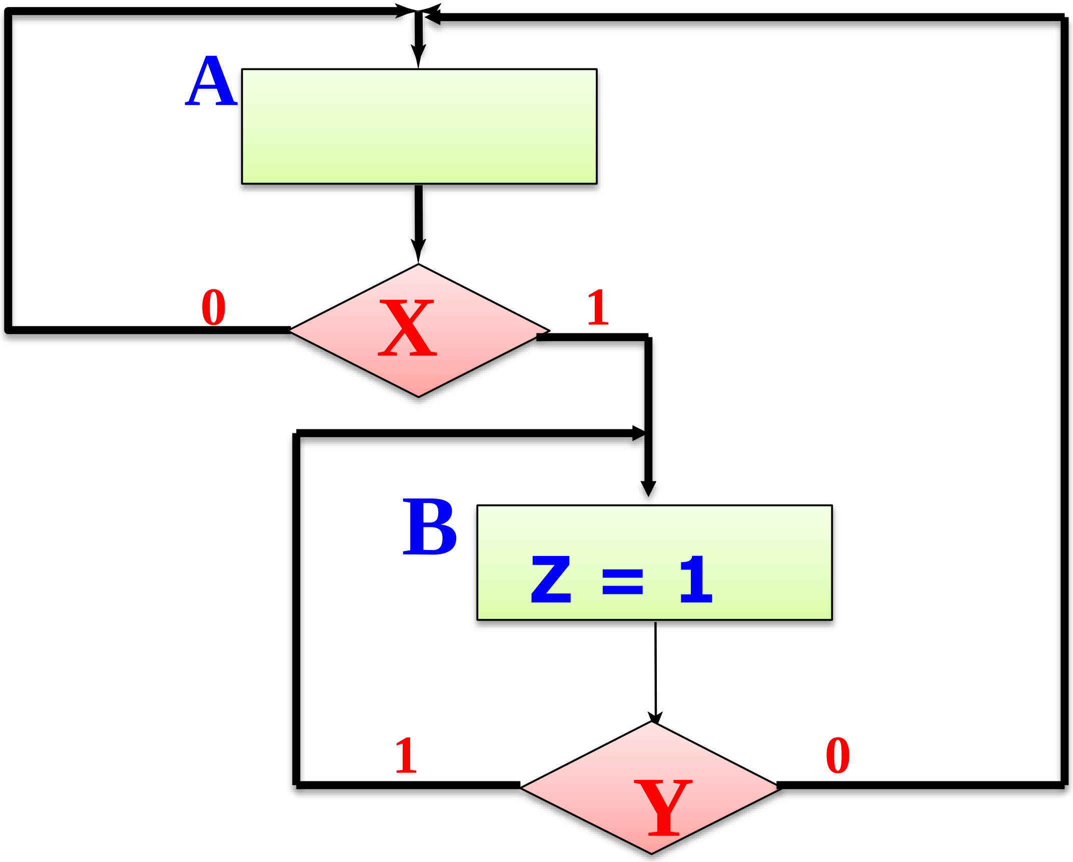




ASM Design Example

- Find the ASM chart corresponding to the following description
  - **There are two states A, B**
  - **If in state A and input X is `0' then the next state is A**
  - **If in state A and input X is `1' then the next state is B**
  - If in state B and input Y is `1' then the next state is B
  - If in state B and input Y is `0' then the next state is A
  - **Output Z is equal to `1' while the circuit is in state B**
- Solution:
  - Total States → 2
  - Two Inputs → X, Y
  - One Output → Z

ASM Design Example



ASM : DP+CP Example

- Find the Data Path and ASM for the following problem:
  - We **first** need to load two registers (R1, R2) with some value.
  - We will **then** need to add the two Registers (R1, R2) and save the result in Register R3.
  - All these operations **should occur if** a “**start**” Signal is activated.

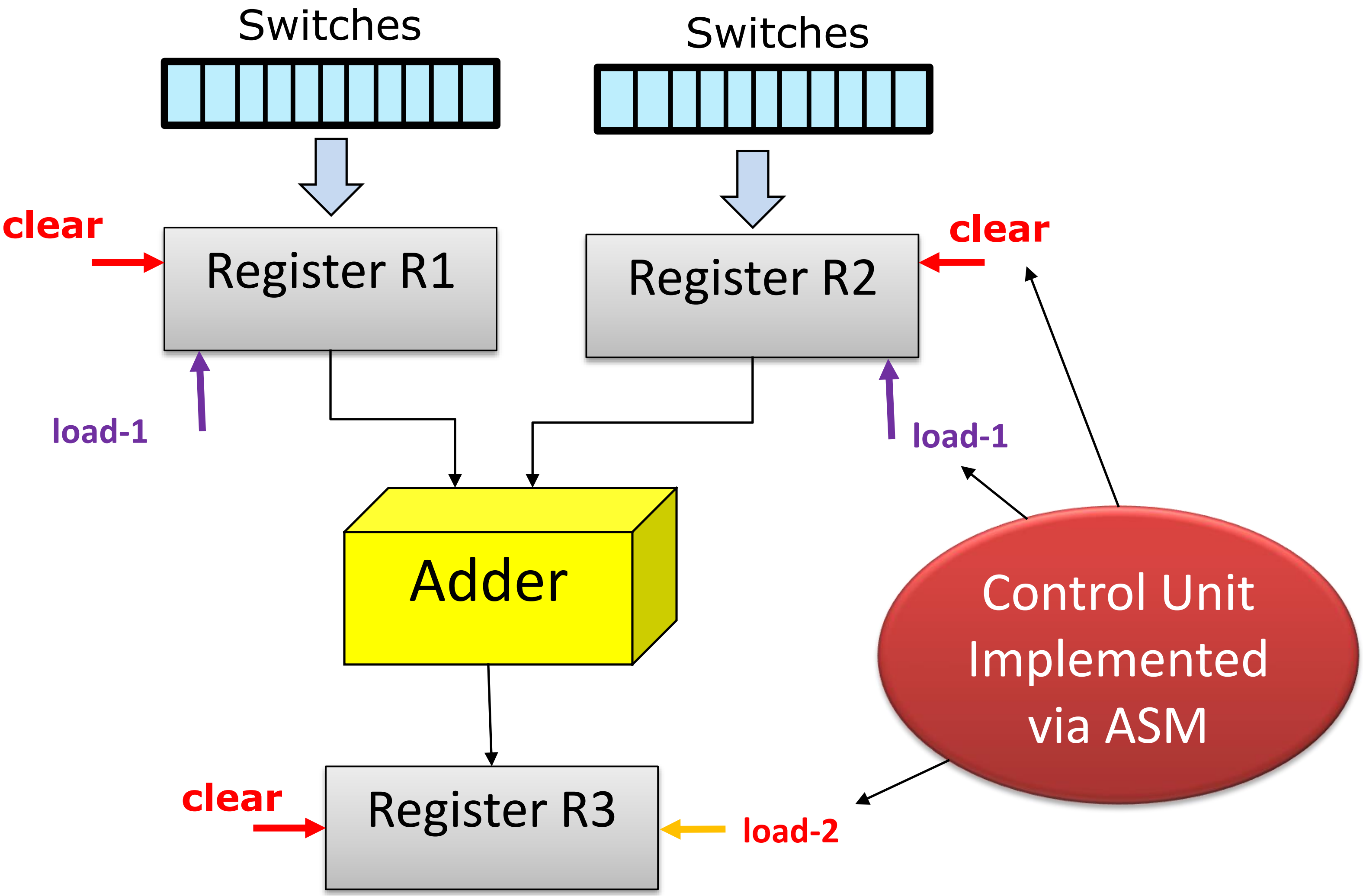
ASM : DP+CP Example

- Translation to Hardware:
  - We need to clear the registers first.
  - If the "**start**" signal is set to 0, I do nothing
  - Else If the "**start**" signal is set to 1, I will **load** R1, R2 with values
  - Next, enable R3 to be loaded (load-2)by the results of R1+R2

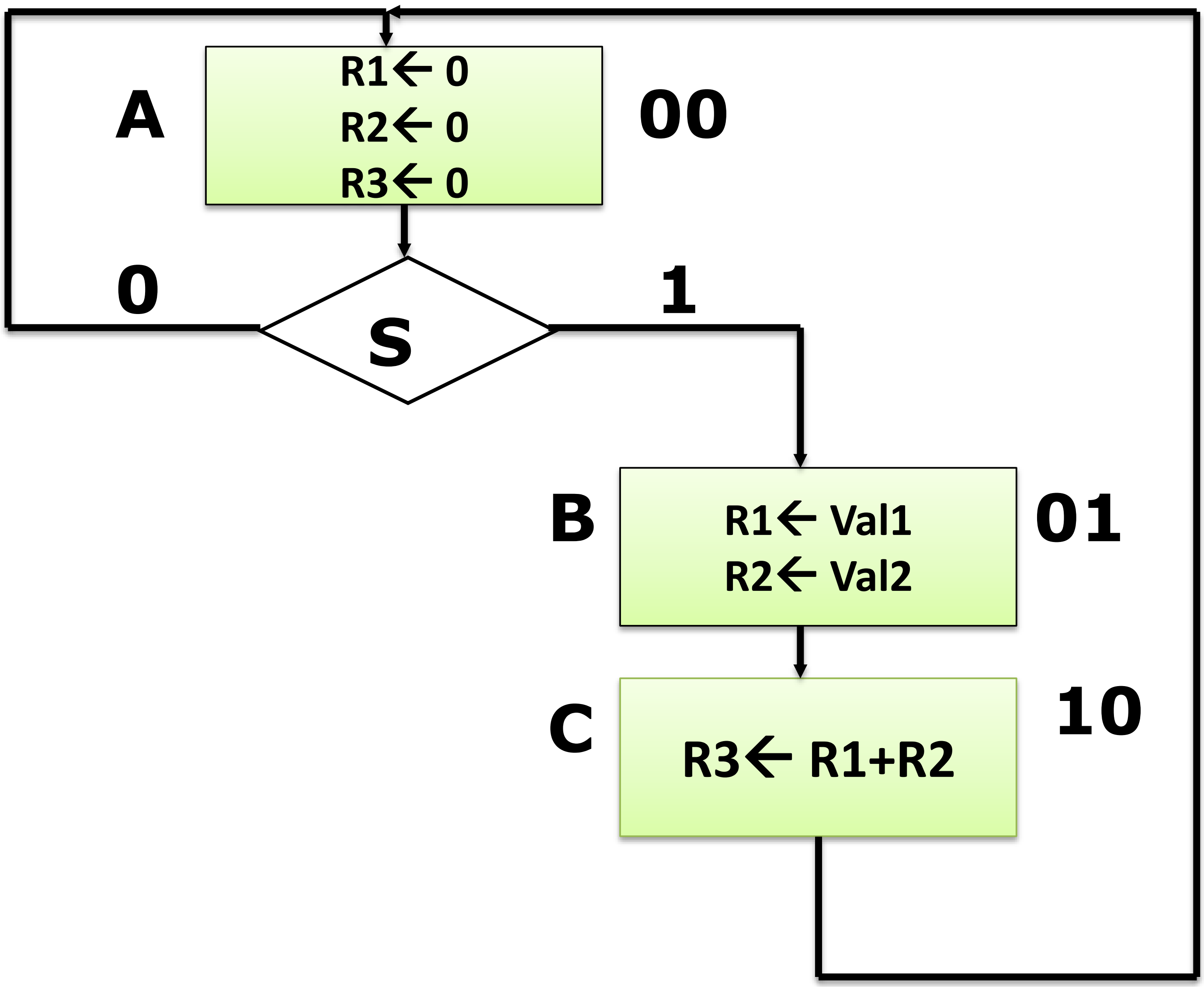
ASM : DP+CP Example

- Inputs/Outputs:
  - **start** is an input signal (Use a switch)
  - **clear** is an output signal generated and connected to R1, R2, R3
  - **load-1** is an output signal generated and connected to R1, R2
  - **load-2** is an output signal generated and connected to R3

Data Path


Switches
Switches
clear
Register R1
Register R2
clear
load-1
load-1
Adder
Control Unit Implemented via ASM
clear
Register R3
load-2

ASM : DP+CP Example



Thanks