Digital Design

RTL Design

A. Sahu
Dept of Comp. Sc. & Engg.
Indian Institute of Technology Guwahati
http://jatinga.iitg.ac.in/~asahu/cs221/

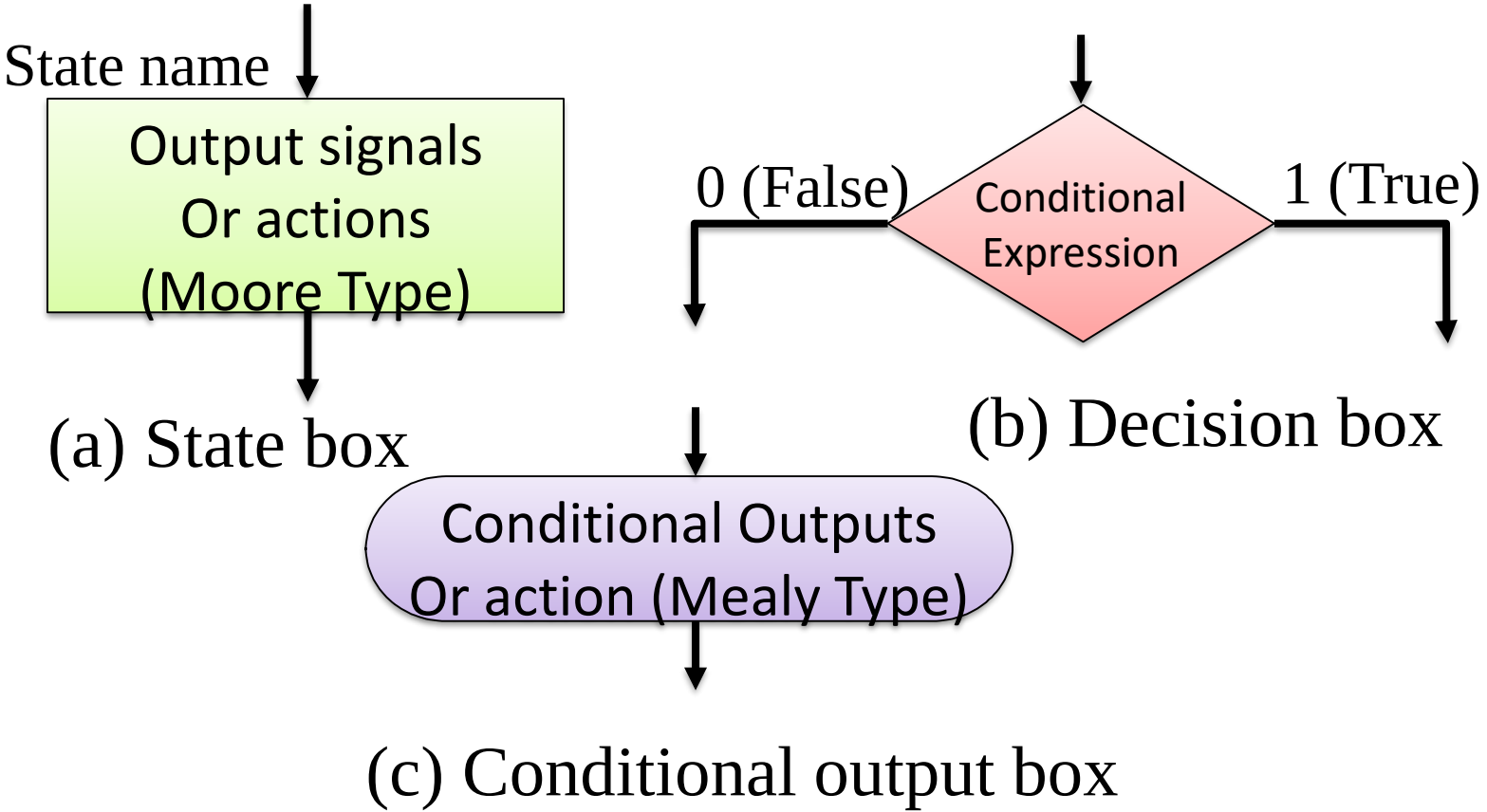
Announcement

- **Quiz on Friday : 8AM to 9AM**
- **Syllabus : topic covered upto 22nd Oct Lecture**
  - **Basic latch, FF, reg, ctr, mem**
  - **FSM based design and Implementation, FSM completeness and Correctness**
  - **FSM Optimization: RM and Partitioning method**
- **Venue : 4201, 4202, 4204, 4205**

Outline

- Simple ASM Examples
- Data Path : R2, R2 and R3
- CP+DP: Mod 14 counter
- Classic Example : Sequential Multiplier
- Verilog Code for Multiplier
- FPGA: Introduction

ASM charts : 3 Elements used

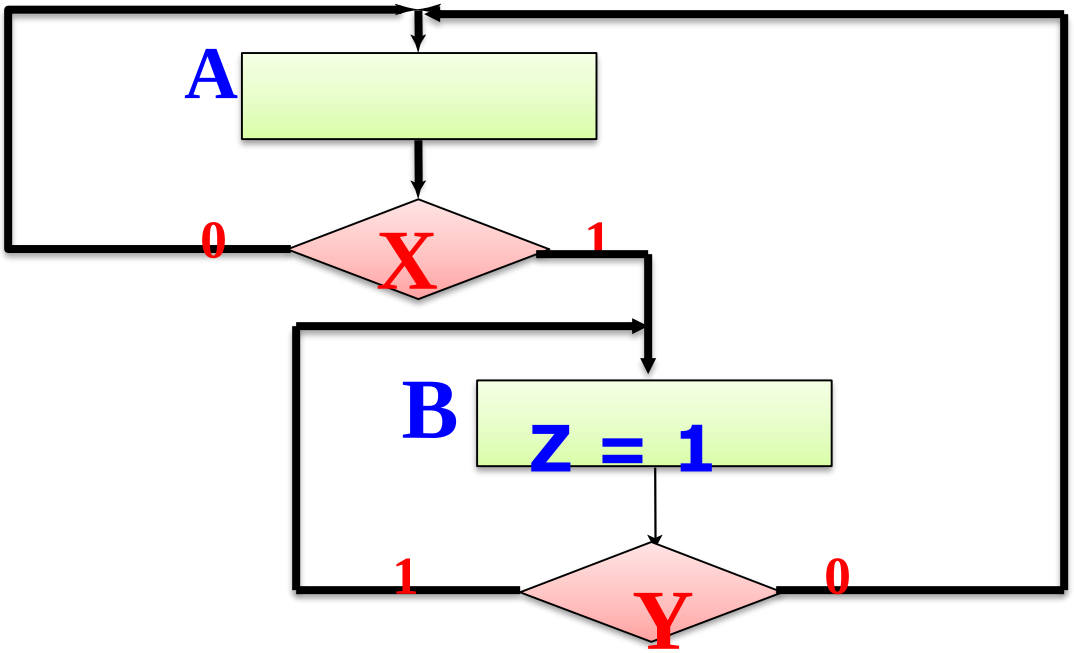


Another Example: Ex 9

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

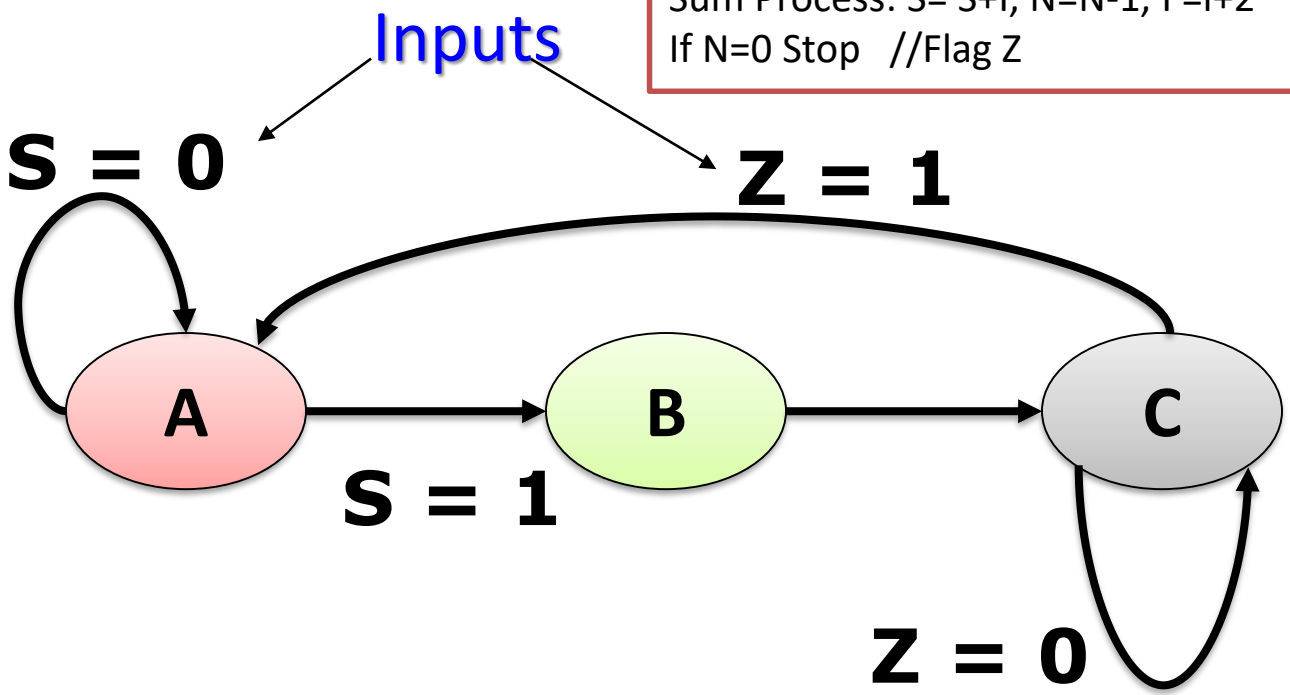
ASM Design Example



Another Example : Ex 10

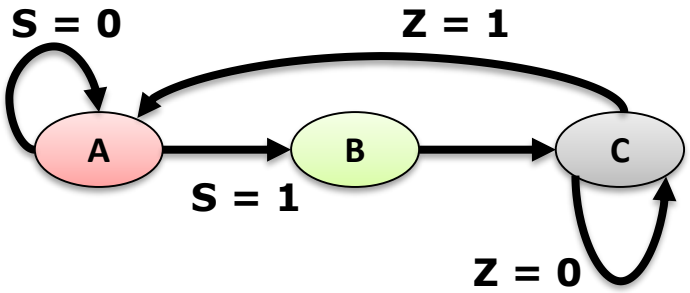
Another Example: From FSM to ASM

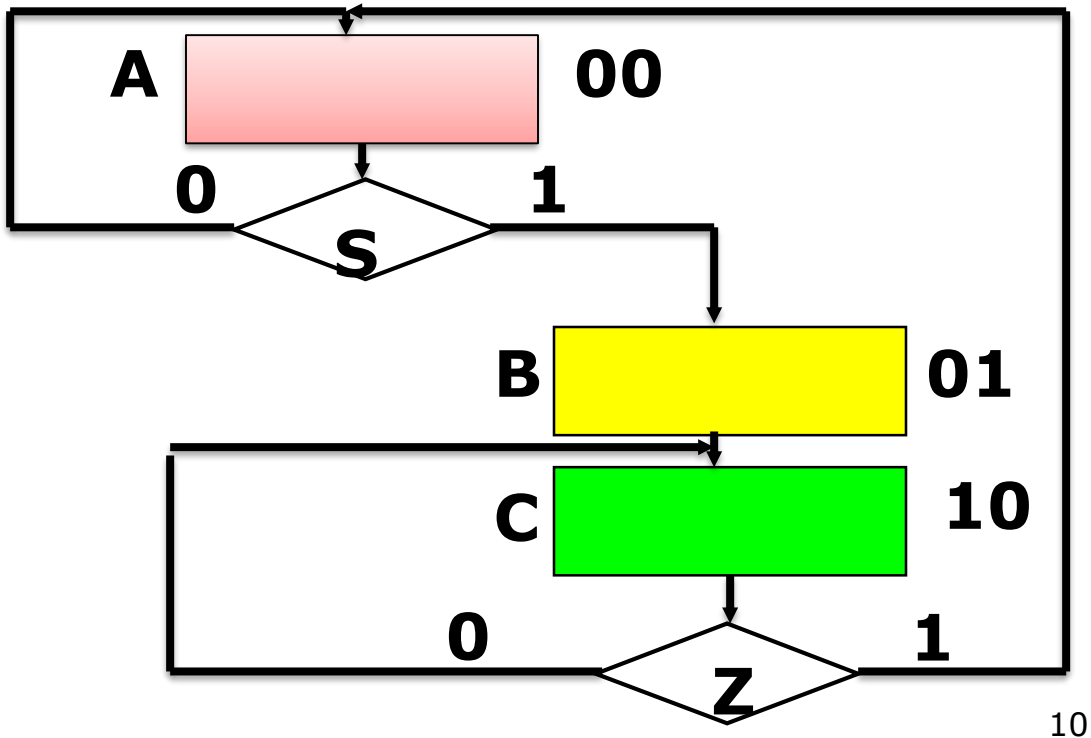
If S =1, Start the Sum Process, initialize N, I=1
Sum Process: S= S+I, N=N-1, I =i+2
If N=0 Stop //Flag Z



Similar to the ASM of : Sum of First N odd numbers

Another Example: From FSM to ASM



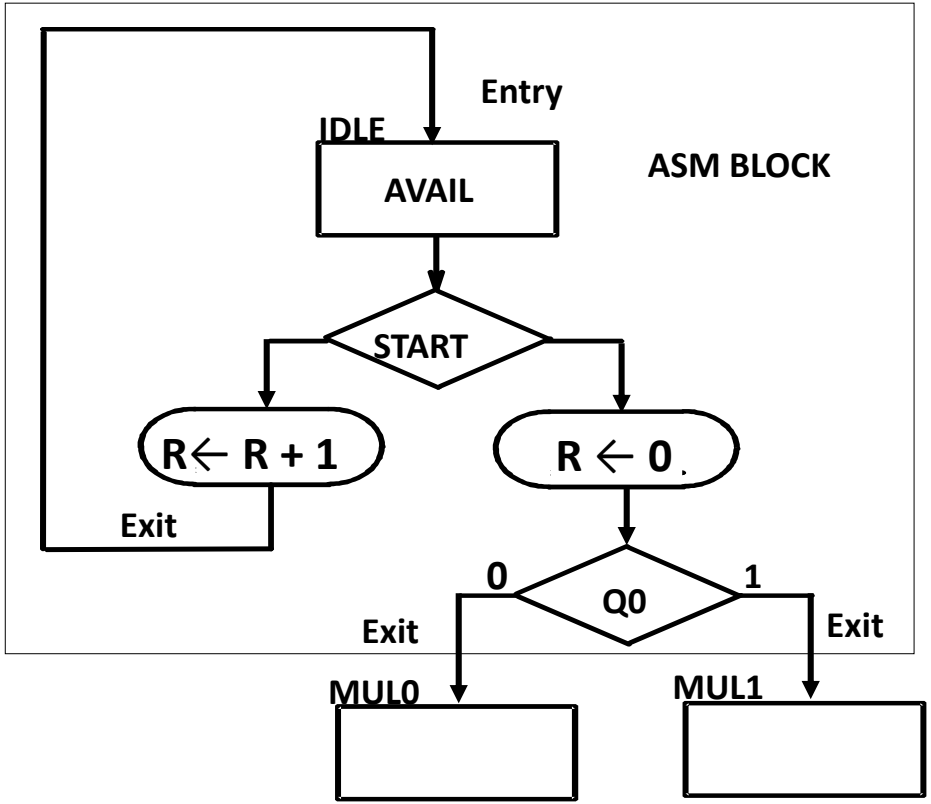


If S =1, Start the Sum Process, initialize N
Sum Process: S= S+I, N=N-1
If N=0 Stop //Flag Z

ASM Block

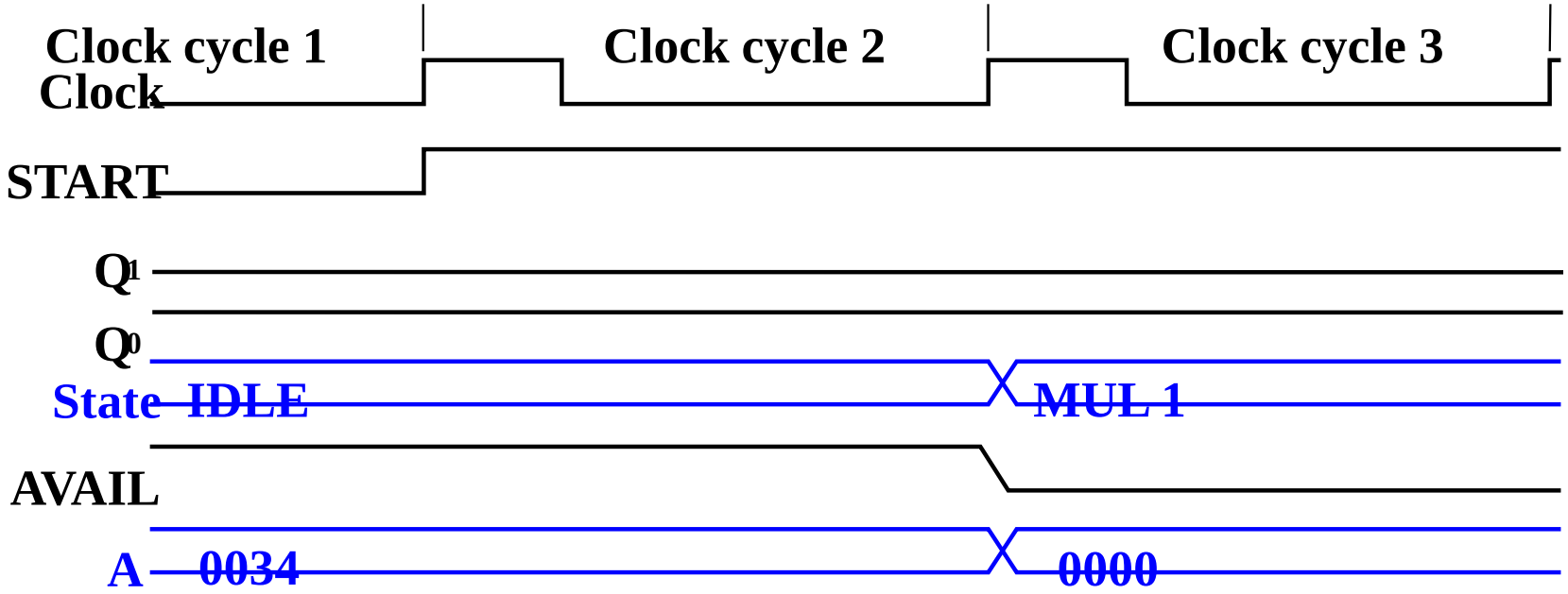
ASM Blocks

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 : Ex 11

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.

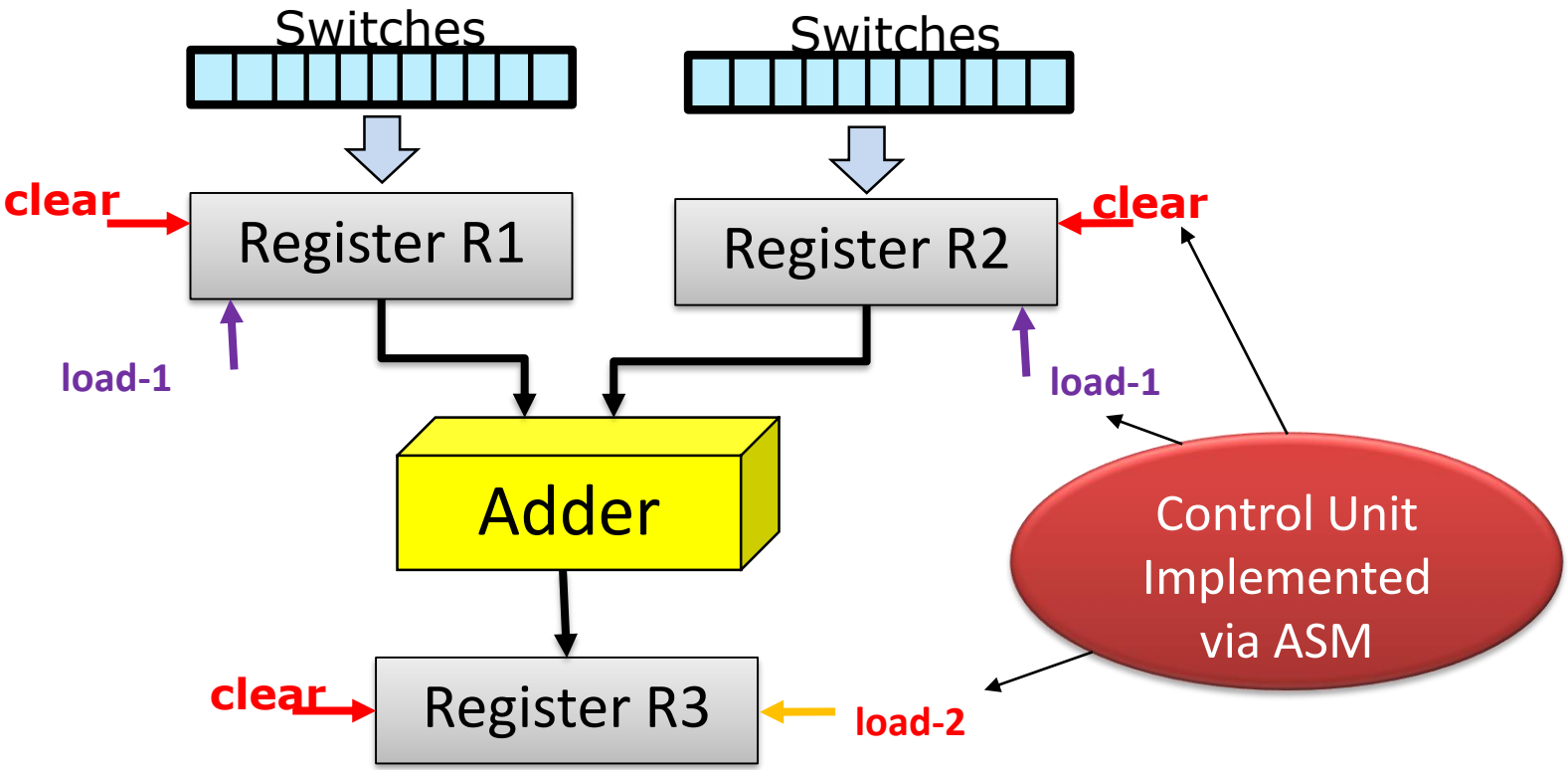
ASM : DP+CP Example

- 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

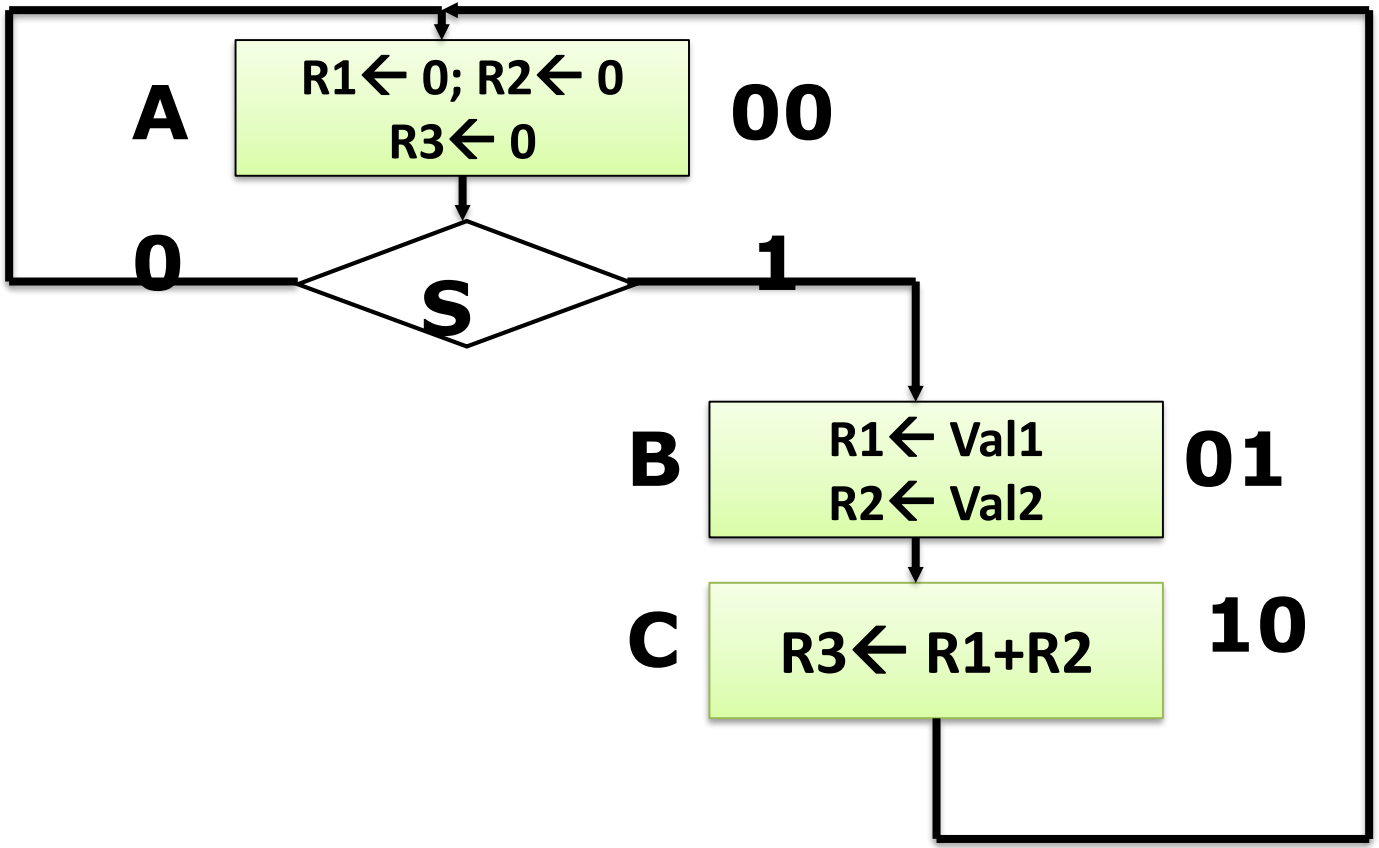
ASM : DP+CP Example

- 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



ASM : DP+CP Example Controller

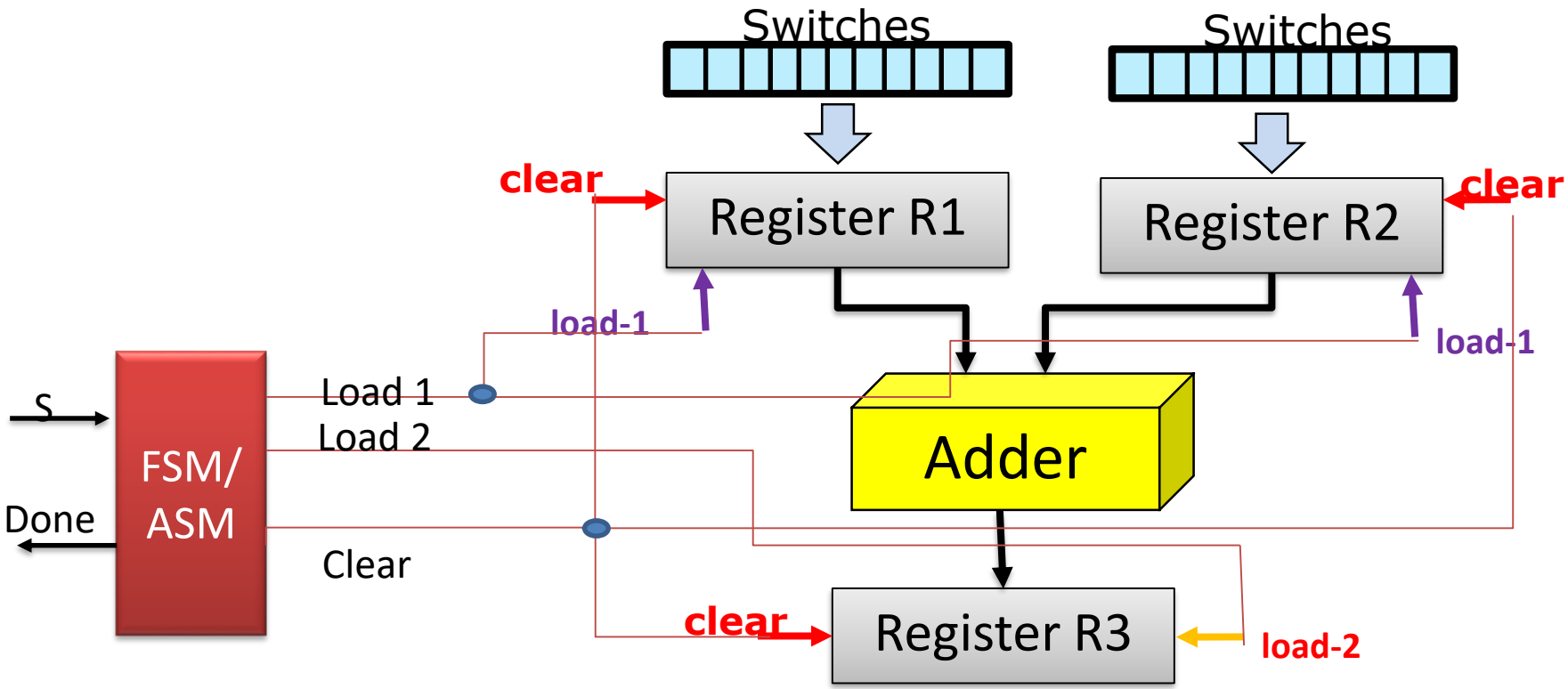
PS		S	NS	
0	0	0	0	0
0	0	1	0	1
0	1	X	1	0
1	0	X	0	0
1	1	X	0	0

PS		CLR	L1	L2
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	X	X	X

D1=Q1’.Q0 D2=S.Q1’.Q0’

CLR=Q1’.Q0’ L1=Q0’.Q1 L2=Q1.Q0’

Data Path


Switches
Switches
clear
Register R1
Register R2
clear
load-1
load-1
S
Load 1
Load 2
FSM/
ASM
Done
Clear
Adder
clear
Register R3
load-2

ASM Charts: An Complete Example
Ref: Mano Book

Ex 12: The Dozenth One

ASM Charts: An Example

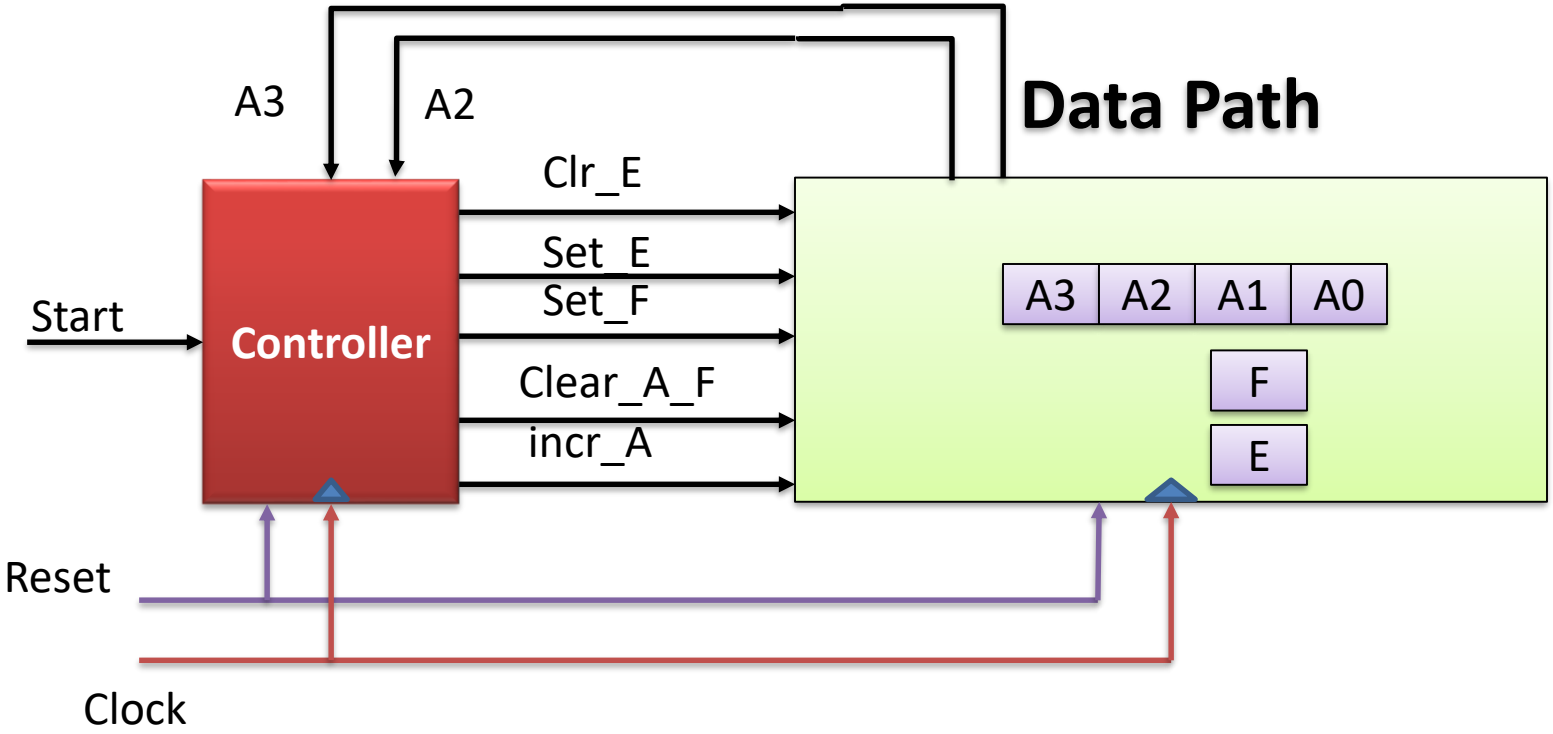
Mod 14 counter:

- There is a 4 bit counter (A)
- E specify: less than 12_{10} or less than 4_{10}
- EF=11 specify : value =12_{10},
  - **It time to reset after next counting**
- E depends of A_2, F depends on A_3 and A_2

If A_2=1 ➔ E=1, else E=0

A_3A_2=1, ➔ F =1, counter reset

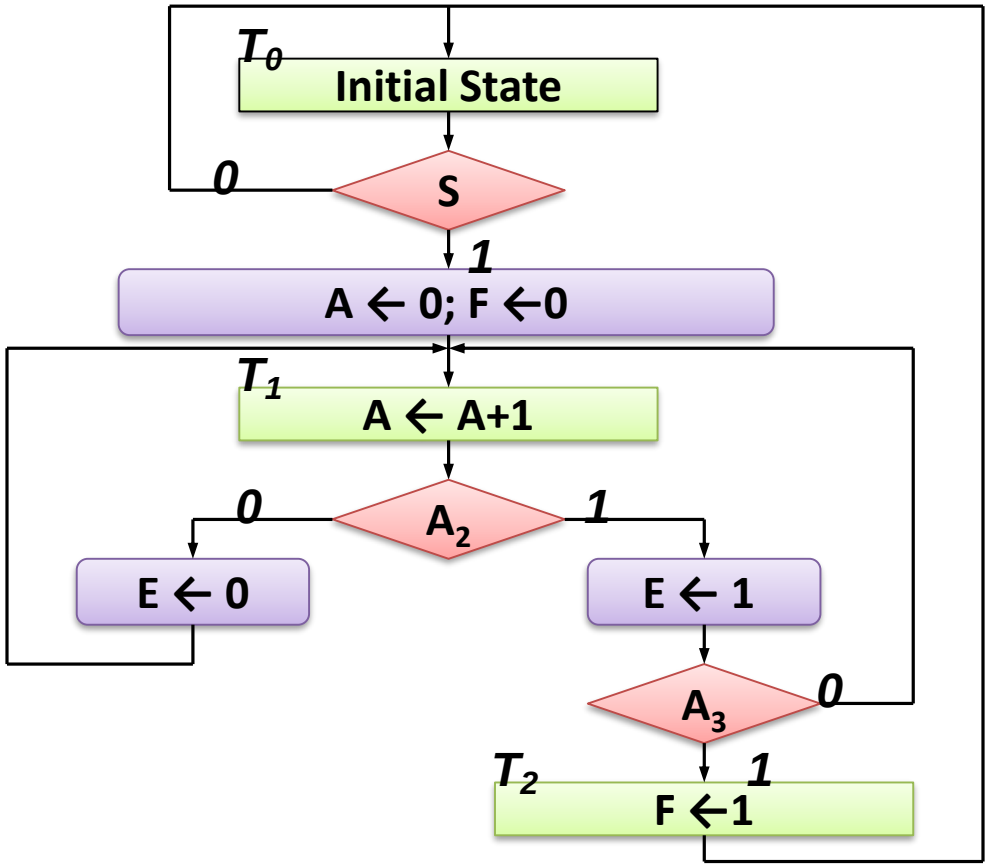
AMS DP+CP : to be High Level



ASM Charts: An Example

- A is a register;
- A_i stands for i^{th} bit of the A register.

  $A = A_3A_2A_1A_0$
- E and F are single-bit flip-flops.

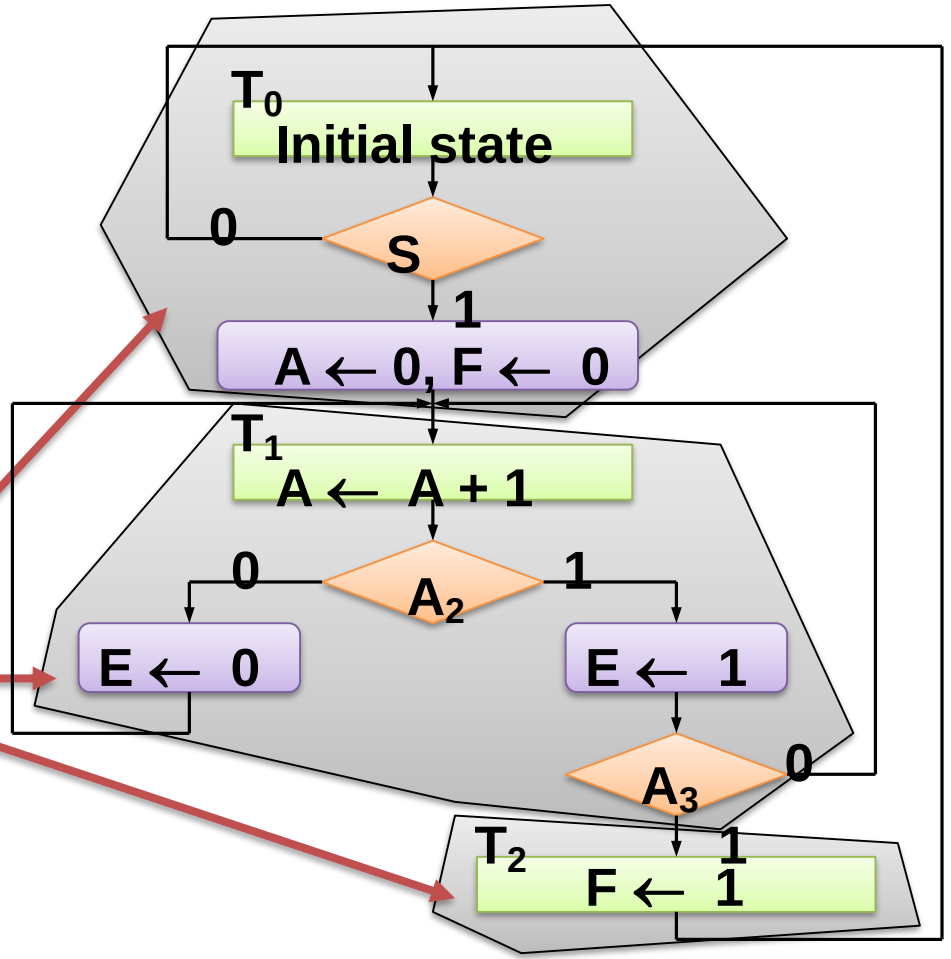


Timing in ASM Charts

- Operations of ASM can be illustrated through a timing diagram.
- Two factors which must be considered are
  - Operations in an **ASM block** occur at the same time in ***one clock cycle***
  - Decision boxes are dependent on the status of the ***previous clock cycle*** (that is, they do not depend on operations of current block)

Timing in ASM Charts

3 ASM blocks

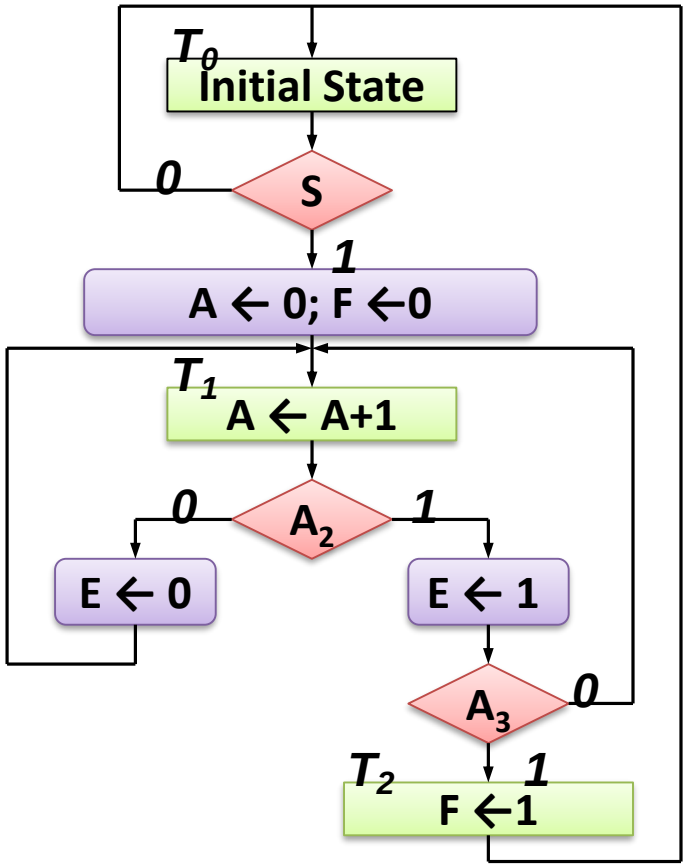


Timing in ASM Charts

- Operations of ASM can be illustrated through a timing diagram.
- Two factors which must be considered are
  - Operations in an ASM block occur at the same time in ***one clock cycle***
  - Decision boxes are dependent on the status of the ***previous clock cycle*** (that is, they do not depend on operations of current block)

Timing in ASM Charts

CTR	E, F	Conditions	State
0000	1,0	A_2=0,A_3=0	T1
0001	0,0	--	--
0010	0,0	---	--
0011	0,0	---	--
0100	0,0	A_2=1,A_3=0	--
0101	1,0	--	--
0110	1,0	--	--
0111	1,0	--	--
1000	1,0	A_2=0,A_3=1	--
1001	0,0	--	--
1010	0,0	--	--
1011	0,0	--	--
1100	0,0	A_2=1,A_3=1	--
1101	1,0		T2
1101	1,1		T0



ASM Chart => Digital System

- ASM chart describes a digital system. From ASM chart, we may obtain:
  - Controller logic (via State Table/Diagram)
  - Architecture/Data Processor

1. **Design of controller is determined**
   - from the decision boxes and the required state transitions.
2. **Design requirements of data processor**
   - can be obtained from the operations specified with the state and conditional boxes.

ASM Chart to Controller

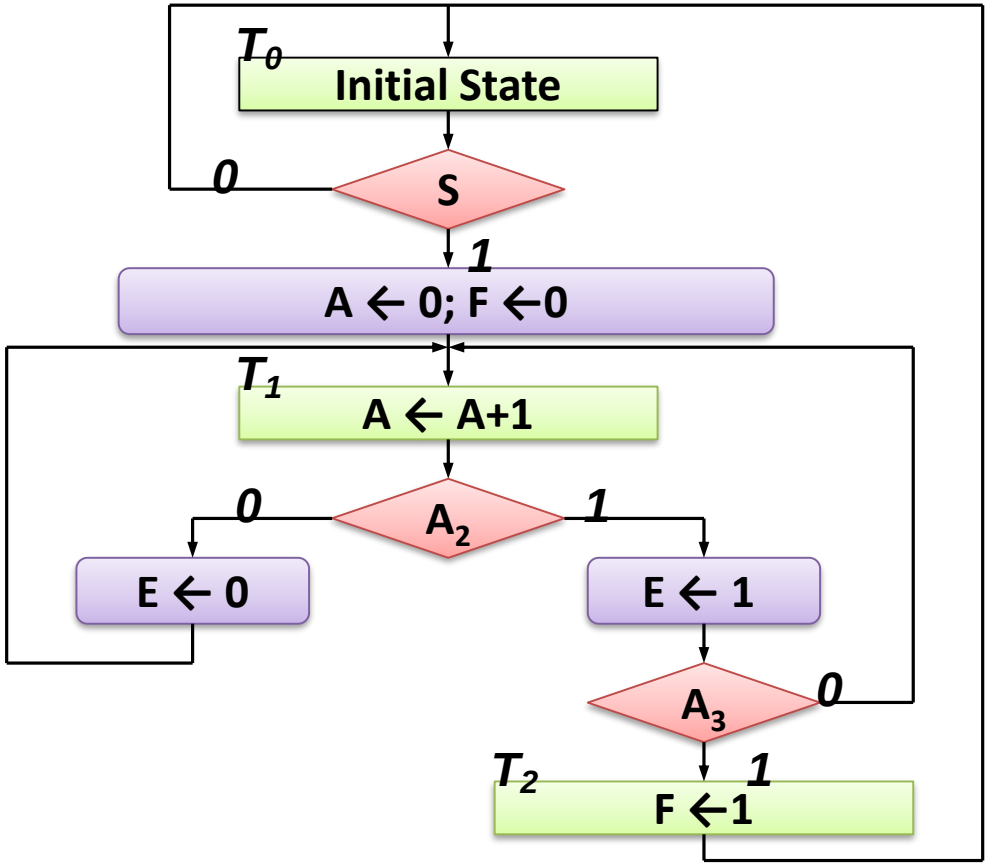
- Procedure:
  - Step 1: Identify all states and assign suitable codes.
  - Step 2: Formulate state table using

    State from state boxes

    Inputs from decision boxes

    Outputs from operations of state/conditional boxes.
  - Step 3: Obtain state/output equations and draw circuit

ASM Chart to Controller



T_0
T_1
T_2

Assign codes to states:

$T_0 = 00$

$T_1 = 01$

$T_2 = 11$

Present state		inputs			Next state		outputs		
G_1	G_0	S	A_2	A_3	G_1^+	G_0^+	T_0	T_1	T_2
0	0	0	X	X	0	0	1	0	0
0	0	1	X	X	0	1	1	0	0
0	1	X	0	X	0	1	0	1	0
0	1	X	1	0	0	1	0	1	0
0	1	X	1	1	1	1	0	1	0
1	1	X	X	X	0	0	0	0	1

Inputs from conditions in decision boxes.
Outputs = present state of controller.

ASM Chart to Architecture/Data Processor

- Architecture is more difficult to design than controller.
- Nevertheless, it can be deduced from the ASM chart. In particular, the OPS from the ASM chart determine:
  - What registers to use?
  - How they can be connected?
  - What operations to support?
  - How these operations are activated?
- Guidelines:
  - Always use high-level units
  - Simplest architecture possible

ASM Chart to Architecture/Data Processor

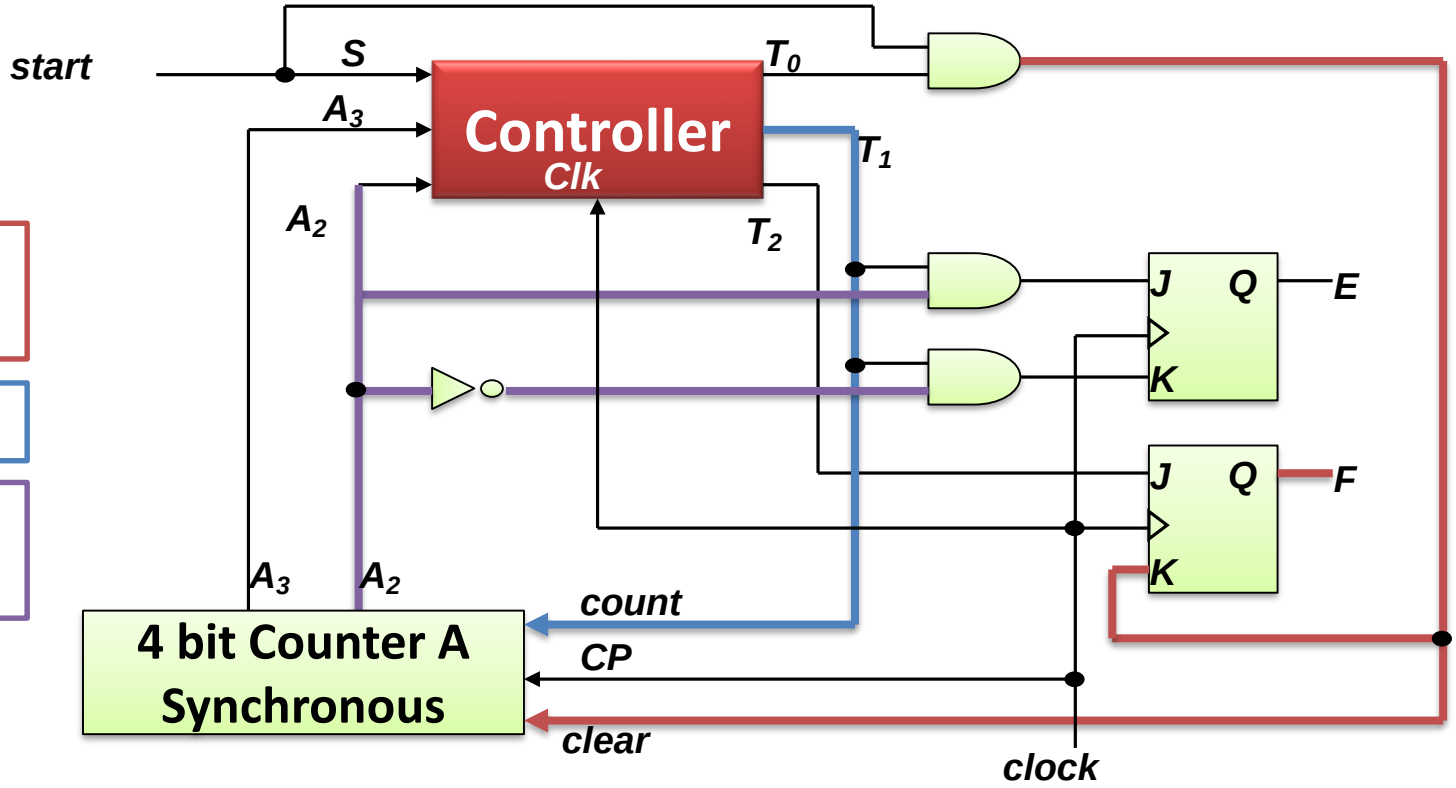
- **Various OPS are:**
  - Counter incremented ($A \leftarrow A + 1$) when state = T_1.
  - Counter cleared ($A \leftarrow 0$) when state = T_0 and S = 1.
  - E is set ($E \leftarrow 1$) when state = T_1 and $A_2 = 1$.
  - E is cleared ($E \leftarrow 0$) when state = T_1 and $A_2 = 0$.
  - F is set ($F \leftarrow 0$) when state = T_2.
  - F is cleared ($F \leftarrow 0$) when state = T_0 and S = 1.
- **Deduce Data Path:**
  - One 4-bit register A (e.g.: 4-bit synchronous counter with clear/increment).
  - Two flip-flops needed for E and F (e.g.: JK/D flip-flops).

ASM Chart => Architecture/Data Processor

$F \leftarrow 0; A \leftarrow 0$ @T_0 and $S = 1$

$A \leftarrow A + 1$ @ T_1

$E \leftarrow 1$ @T_1 and $A_2 = 1$



Thanks