CS/ECE/EEE/INSTR F215:Digital Design

Lecture 32: ASM

Sat, 27 Nov 2021

BITS Pilani
Hyderabad Campus

Dr. R. N. Ponnalagu, EEE

Sometimes we are tested,

Not to expose our weakness,

But to discover our strengths.

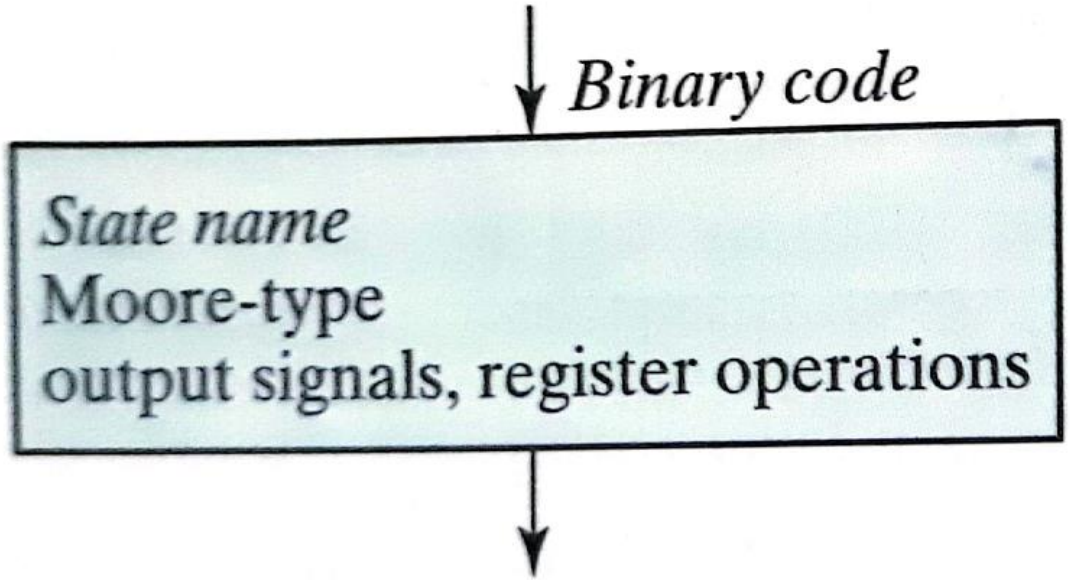
- **Flowchart developed to design Digital Hardware Algorithms is Algorithmic State Machines (ASM) chart**

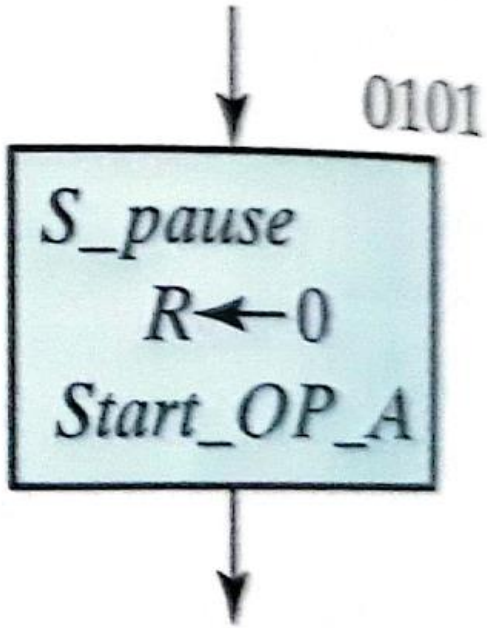
ASM CHART
composed of three basic elements

- **The State box**
- **The decision box**
- **The conditional box**

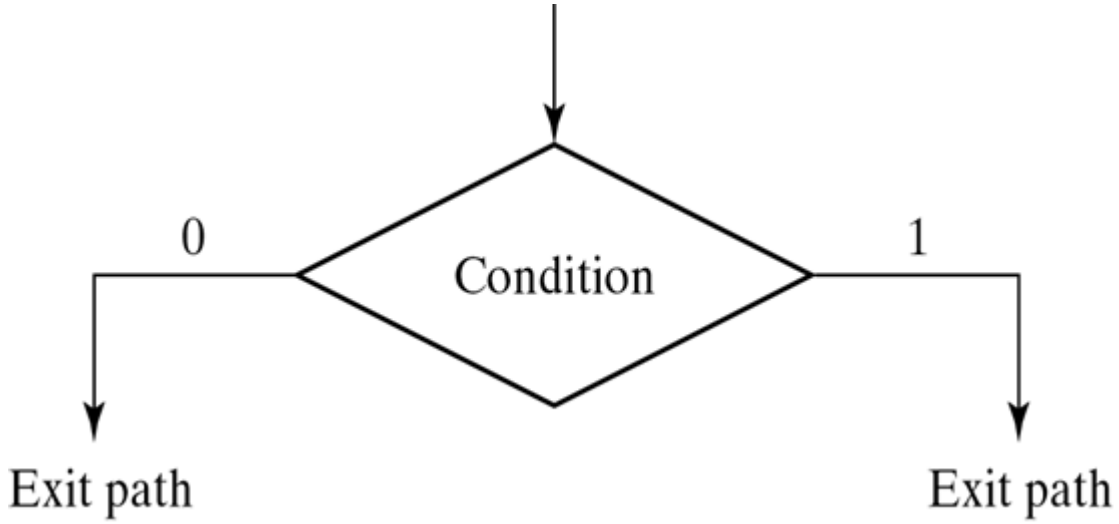
- **A state is indicated by the state box (Rectangle in shape) within which are written i) register operations that occur when state transitions to next state are written and/or ii) the names of the output signal that the control generates while being in the present state.**

- **State is given a symbolic name (Upper left corner of the box), binary code assigned to state (upper right)**



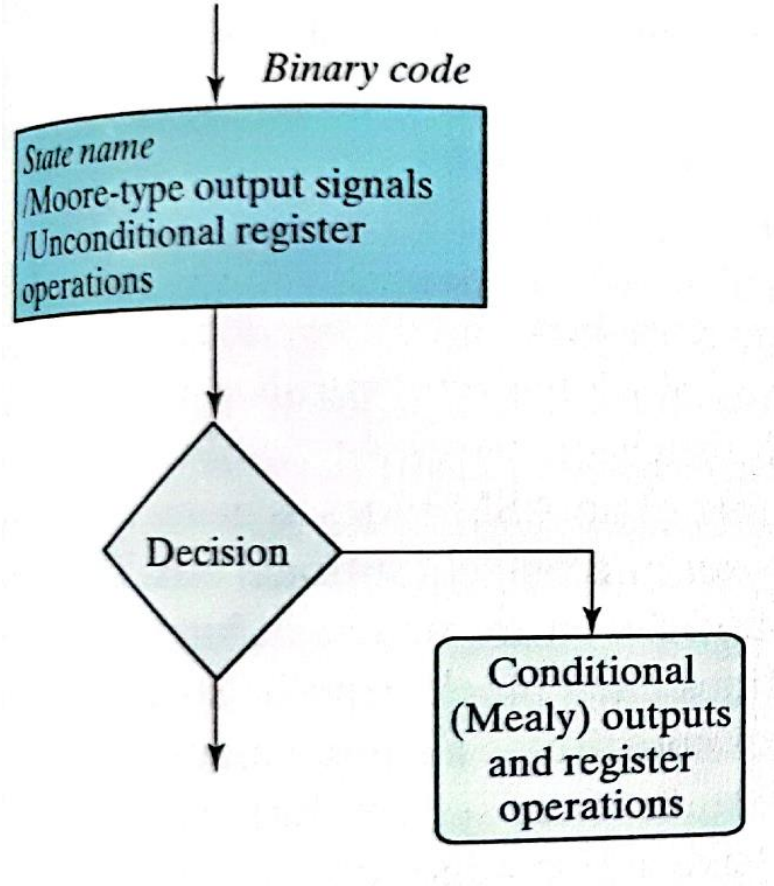


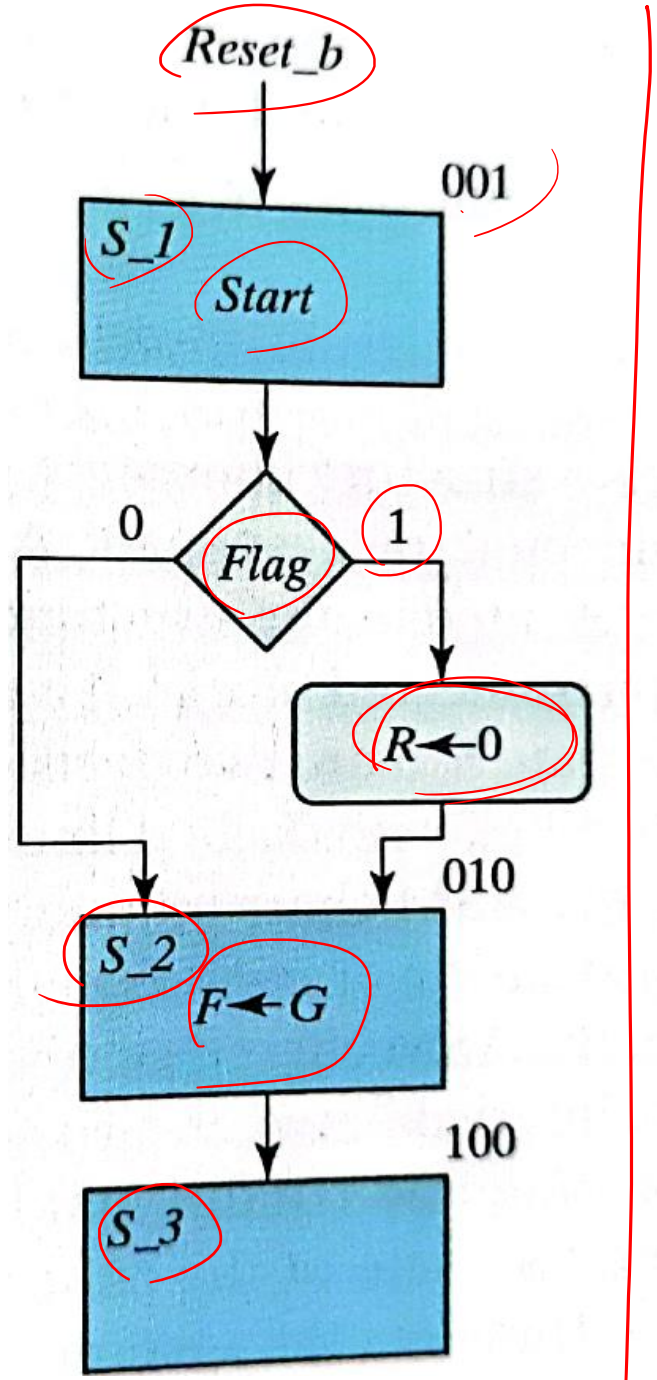
- **Decision box (diamond shaped) indicates the effect of an input on the control subsystem**
- **Condition to be tested is written inside the box**
- **One exit path is taken if the condition is true (label 1) and another if false (label 0)**

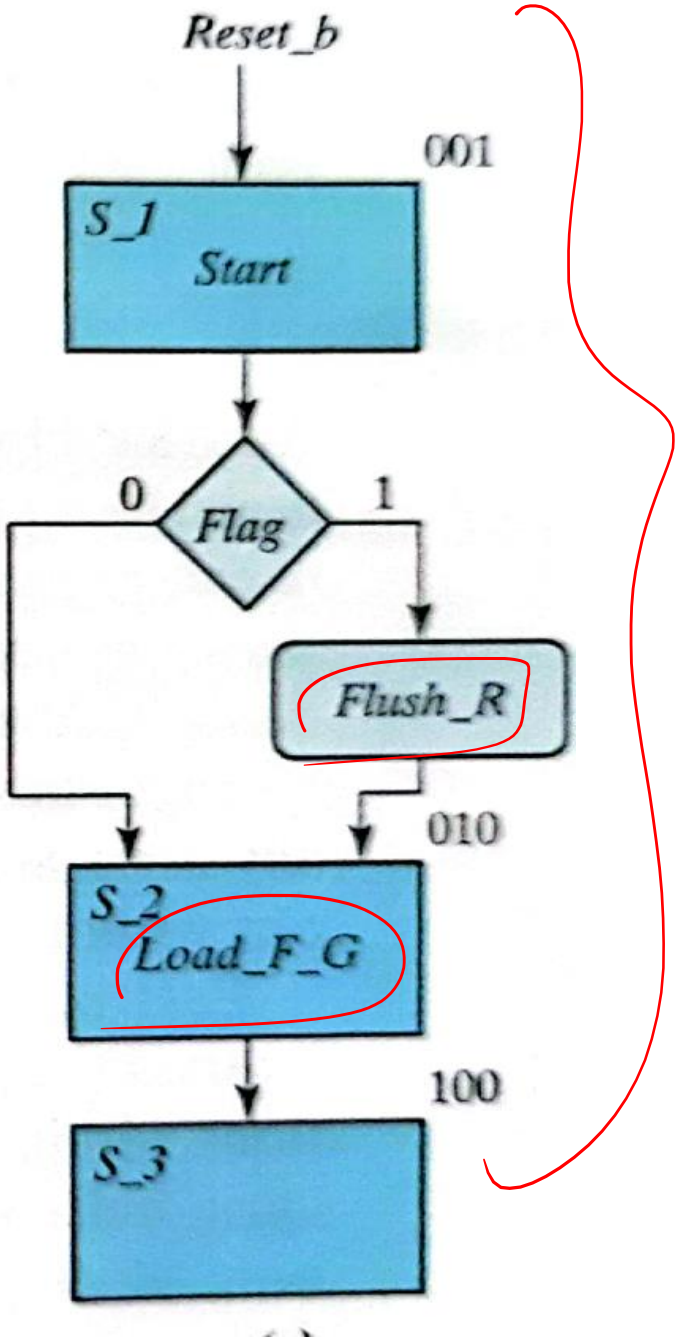


- **State box and decision box are similar to those used in conventional flowchart**

- **The conditional box (Shape: rounded corners) is unique to ASM chart**
- **Input path to conditional box comes from one of exit paths of a decision box**
- **The outputs listed inside a conditional box are generated during a given state and the register operations are associated with a transition from the state**

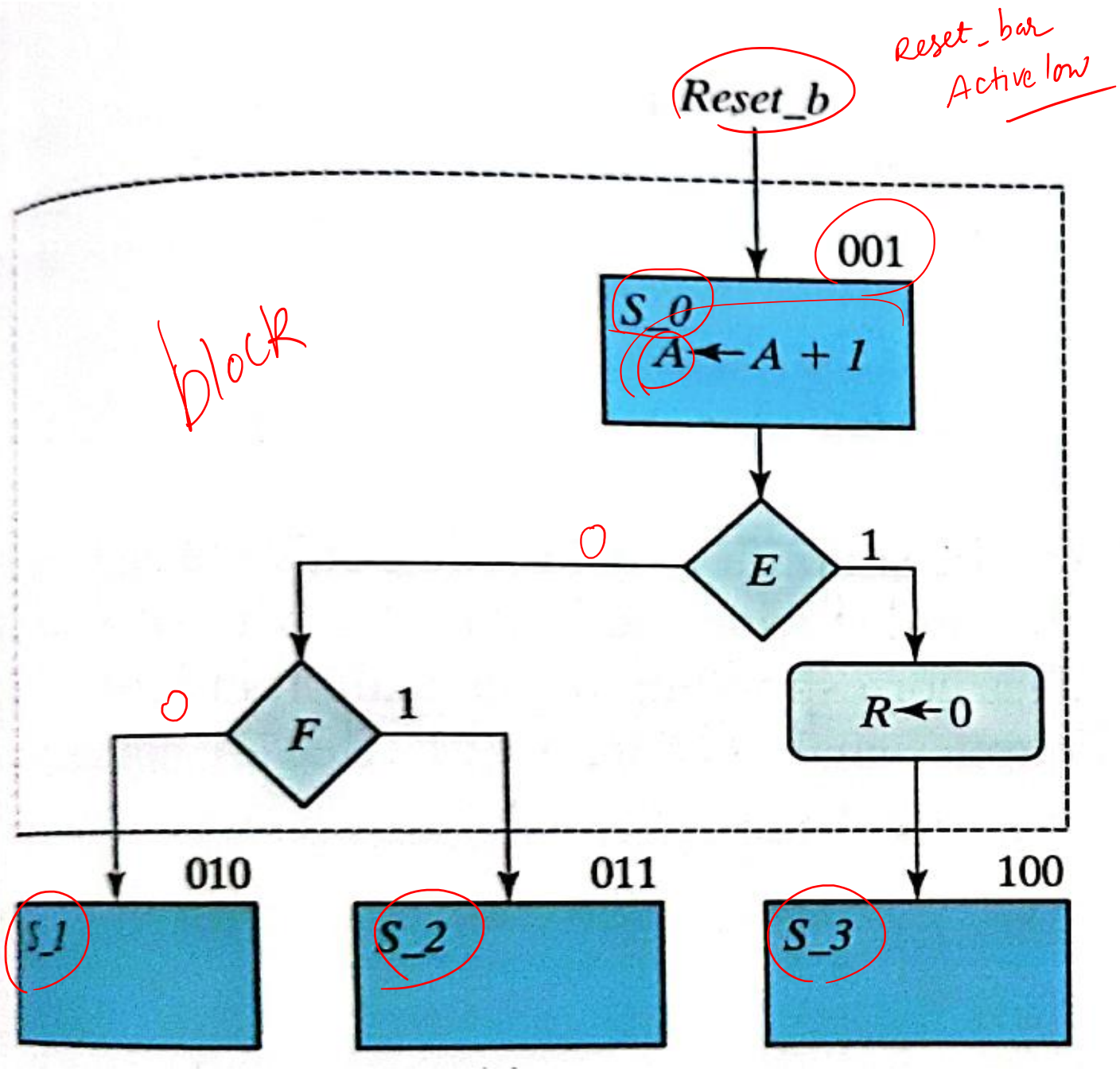


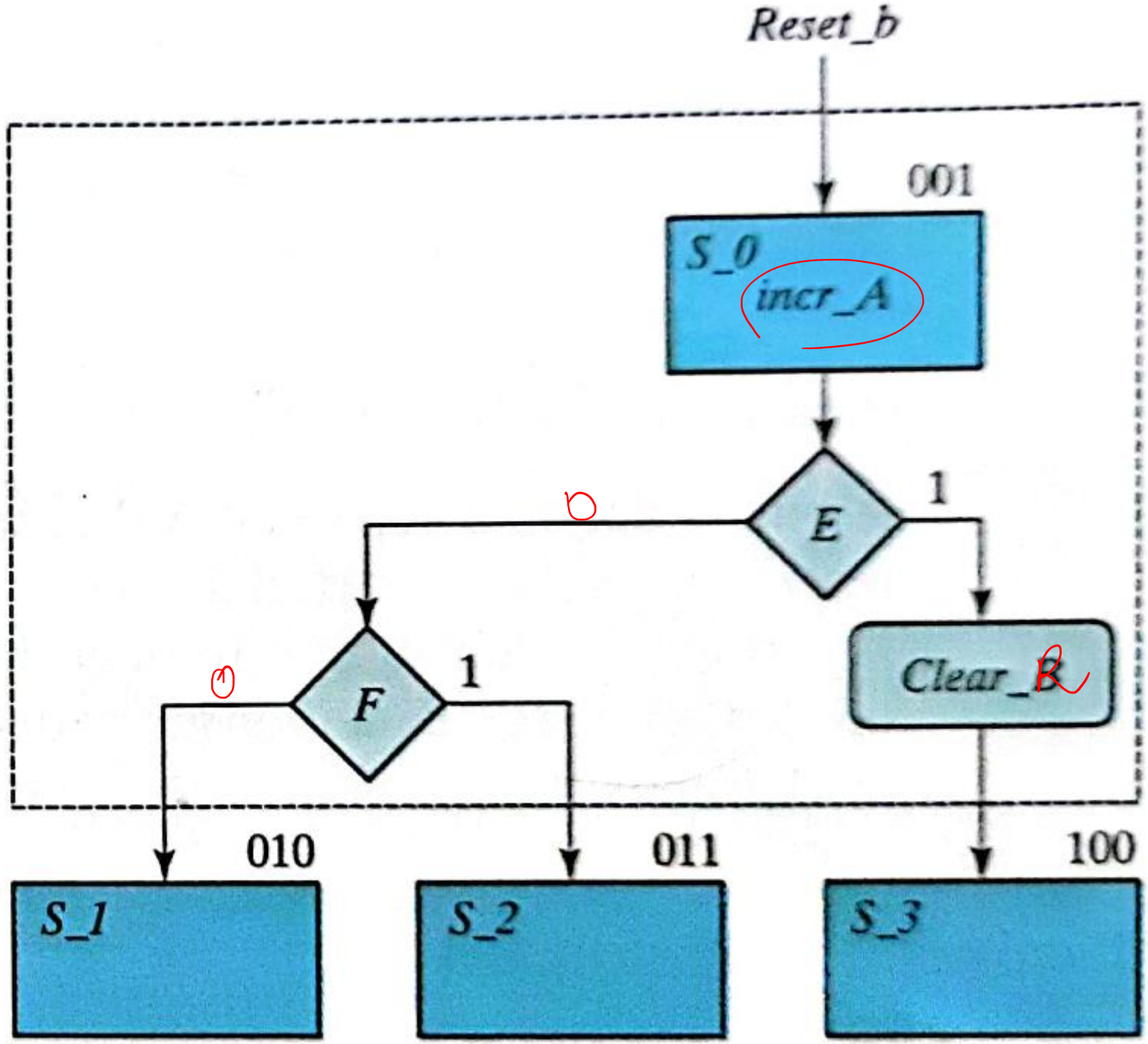

Reset_b
001
S_1
Start
0
Flag
1
R←0
010
S_2
F←G
100
S_3

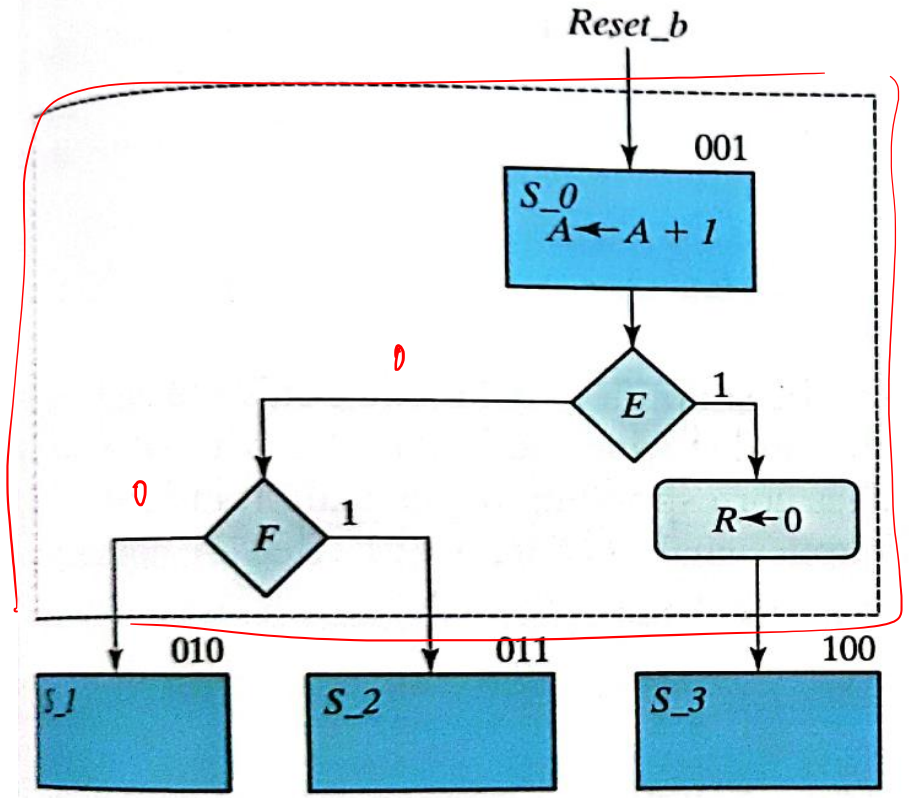

Reset_b
001
S_1
Start
0
Flag
1
Flush_R
010
S_2
Load_F_G
100
S_3

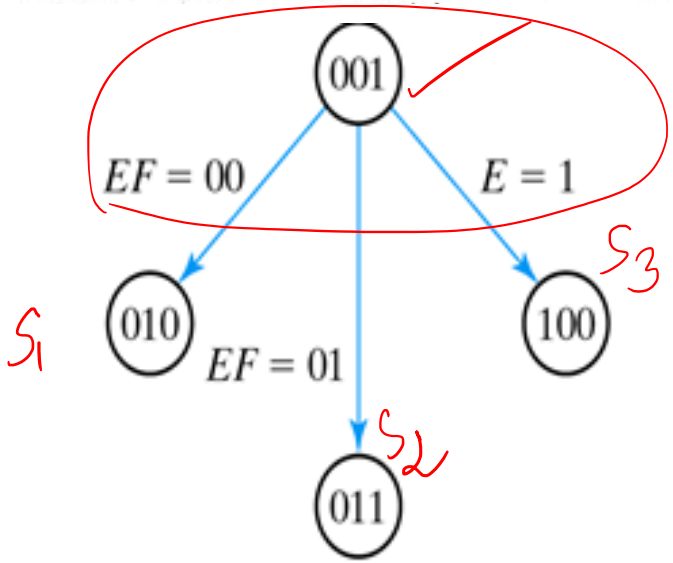
ASM Block in ASM Chart

- A structure consisting of one state box and all the decision and conditional boxes connected to its exit path.
- An ASM block has one entrance and any number of exit paths represented by the structure of the decision boxes
- ASM Chart may have one or more interconnected blocks.
- Each block in the ASM chart describes the state of the system during one clock pulse. (interval between two successive active edges of clock)
- The operations within the state and conditional boxes are executed with a common clock pulse while the system transits from one state to next.
- Same clock pulse transfers system controller to next states.
- ASM chart is very similar to state diagram.


Reset_b
Reset_bar
Active low
001
S_0
A ← A + 1
block
E
0
1
F
0
1
R ← 0
010
011
100
S_1
S_2
S_3

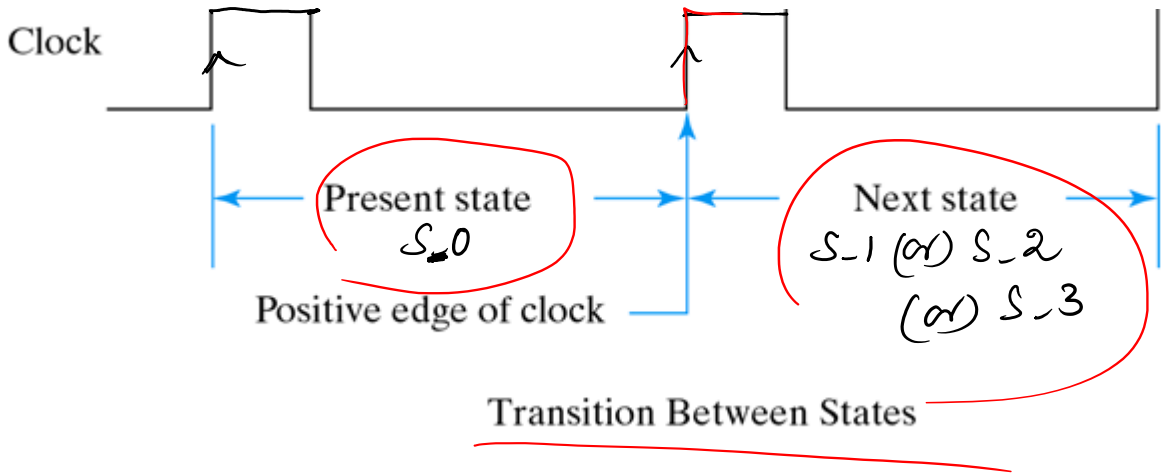

Reset_b
001
S_0
incr_A
E
1
0
F
1
0
Clear_R
010
011
100
S_1
S_2
S_3





- Each state block is equivalent to a state in a sequential circuit.
- Decision box is equivalent to the binary information written along the directed lines that connects two states in a state diagram
- ASM chart can be easily converted to state diagram and then proceed with the design
- The unconditional and conditional operations that must be performed in the data path unit are not indicated in the state diagram of the controller

- In a digital system clock pulses are applied not only to registers of the data path but also to all the FFs in the state machine implementing control unit.
- Inputs are also synchronized to clock and change state in response to an edge transition (synchronous sequential circuit)
- Major difference between a conventional flowchart and an ASM chart is interpreting the timing relations among the various operations
- In conventional flowchart the listed operations follow one after the other in a sequence. No concept of timing and synchronization.
- In contrast, ASM chart consider the entire block as one unit.
- All operations within a block must occur in synchronism during clock edge transition.

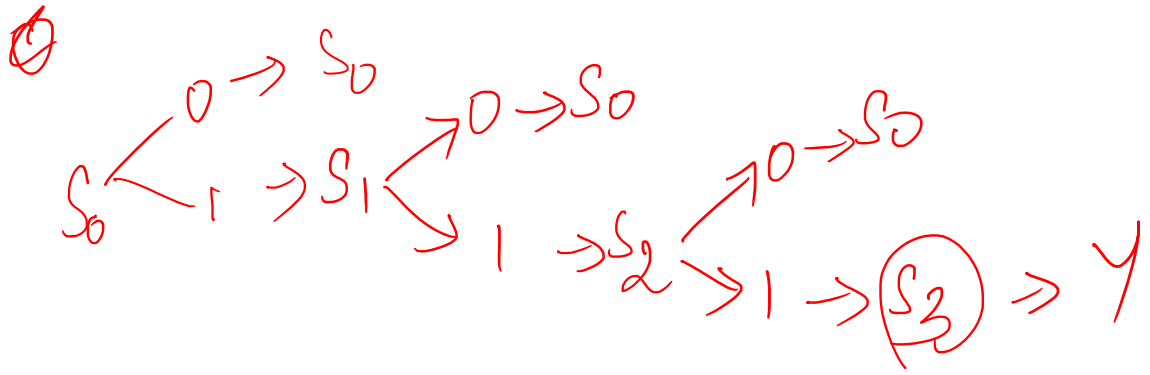


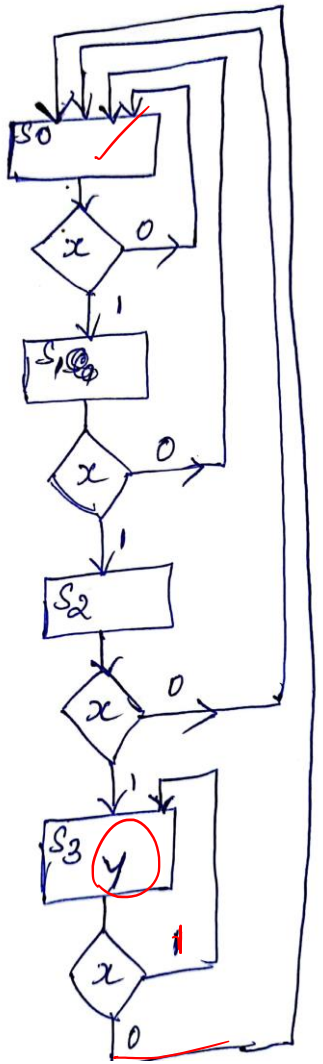
During transition

A is incremented, If E = 1 , R is cleared

Depending on values of E and F control transferred to states S_1 or S_2 or S_3

Draw an ASM chart for a synchronous state machine that is to monitor input *x* and assert output *y* after three consecutive 1s are received and remain asserted until a 0 is received.

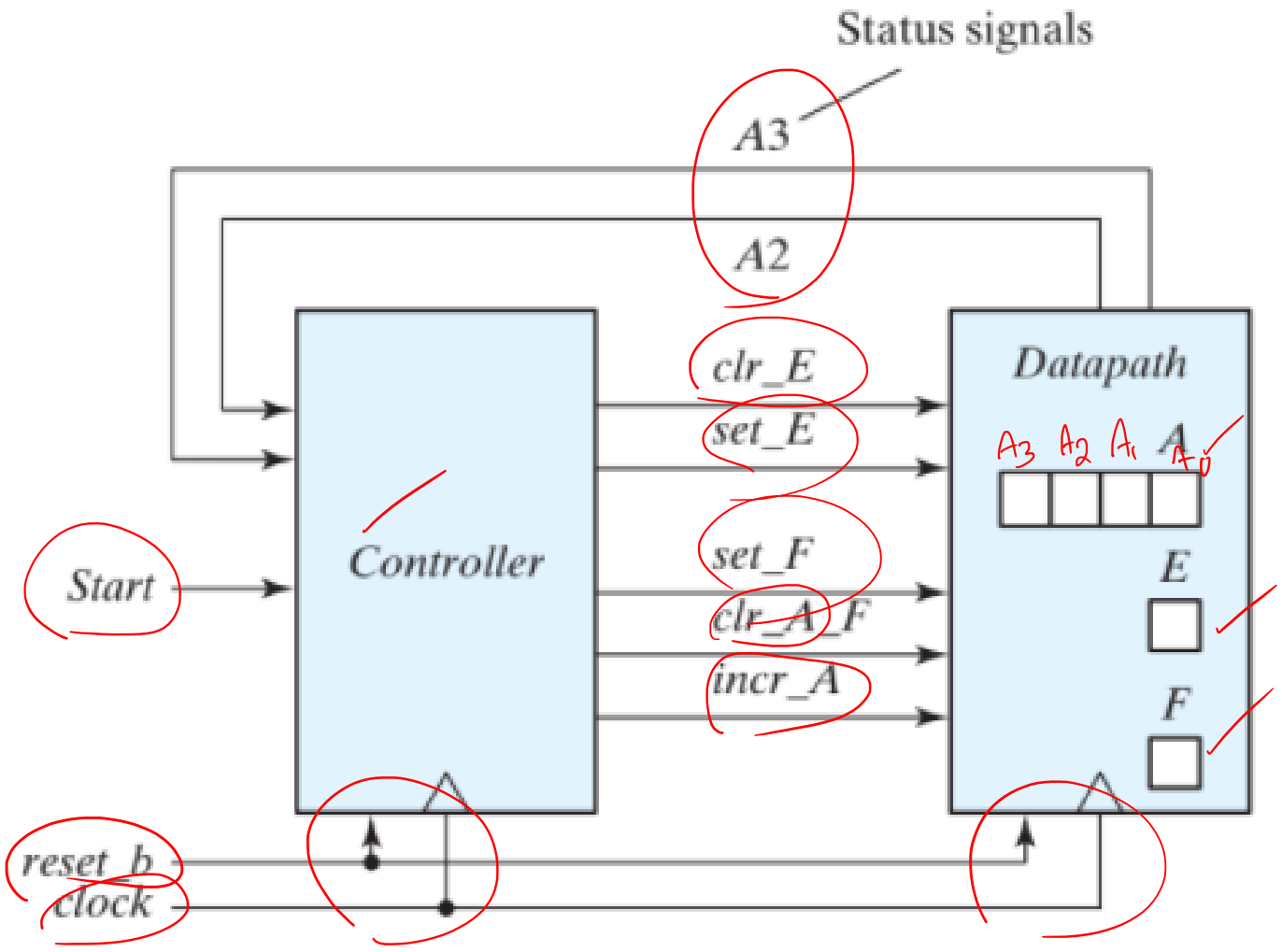




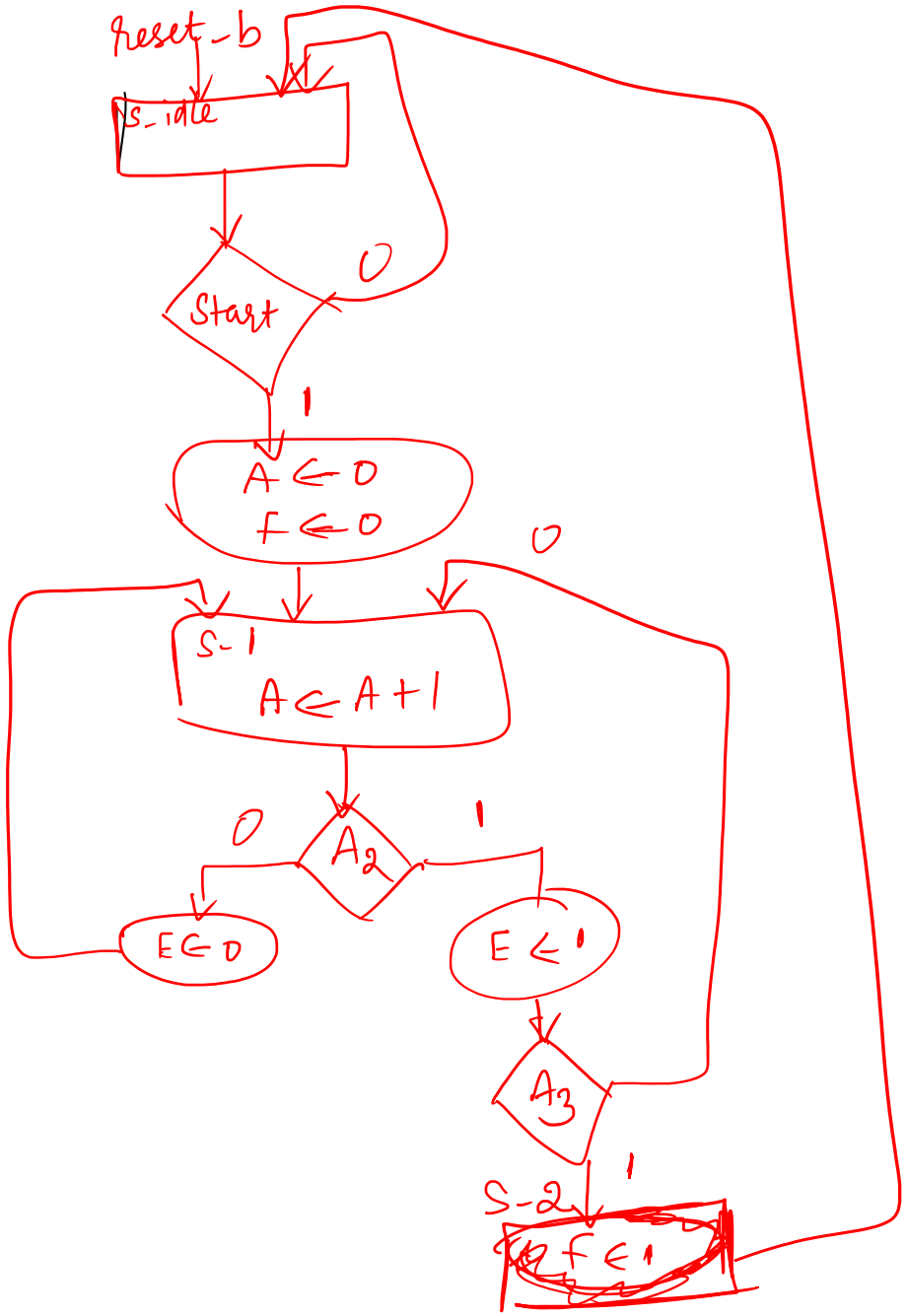
Example

- Design a Digital system having two FFs E and F, one 4-bit binary counter A (individual FFs :~~A4A3A2A1~~ $A_3 A_2 A_1 A_0$).
- Initial state is the reset state: S_idle (state reached by application of reset_b)
- A Start signal initiates the operation by clearing counter A and flip-flop F.
- Counter incremented by one starting from next clock pulse, continues to increment until operations are such that it stops.
- Counter bits A2 and A3 determine sequence of operations
  - ✓ If A2 = 0, E is cleared to 0 and count continues
  - ✓ If A2 = 1, E set to 1; then if A3=0, count continues,
  - ✓ but if A3 = 1, F is set to 1 on next clock pulse and system stops counting.
- If Start = 0 system remains in initial state, but if Start = 1 cycle repeats.

Block diagram of the design



- When no operations the system is in initial state *S_idle* awaiting for *Start* signal
- When input *Start* = 1, state changes to *S_1* and the counter A and flip-flop F are cleared. The register operations occur unconditionally (Moore type).
- Register A is incremented at every clock edge while machine is state *S_1*.
- In state *S_1* when the counter is incremented with every clock pulse and at the same time, one of three operations occur during clock transition either
  - ✓ E is cleared and control stays in state *S_1* (A2 = 0); or
  - ✓ E is set and control stays in state *S_1* (A2A3 = 10); or
  - ✓ E is set and control goes to state *S_2* (A2A3 = 11).
- Hence block with state *S_1* has two decision and two conditional boxes
- In state *S_2*, a Moore type control signal is asserted to set flipflop F and state changes to *S_idle*.


reset_b
S_idle
Start
0
1
A ⟵ 0
f ⟵ 0
0
S_1
A ⟵ A+1
A2
0
1
E ⟵ 0
E ⟵ 1
A3
S_2
1
f ⟵ 1

Sequence of operations

Counter				Flipflops			
A3	A2	A1	A0	E	F	Conditions	State
0	0	0	0	1	0	A2= 0, A3=0	*S_1*
0	0	0	1	0	0		
0	0	1	0	0	0		
0	0	1	1	0	0		
0	1	0	0	0	0	A2=1, A3= 0	
0	1	0	1	1	0		
0	1	1	0	1	0		
0	1	1	1	1	0		
1	0	0	0	1	0	A2=1, A3= 0	
1	0	0	1	0	0		
1	0	1	0	0	0		
1	0	1	1	0	0		
1	1	0	0	0	0	A2=1, A3= 1	
1	1	0	1	1	0		*S_2*
1	1	0	1	1	0		*S_idle*

Sequence of Operations for Design Example

Counter				Flip-Flops			
A_3	A_2	A_1	A_0	E	F	Conditions	State
0	0	0	0	1	0	$A_2 = 0, A_3 = 0$	*S_1*
0	0	0	1	0	0		
0	0	1	0	0	0		
0	0	1	1	0	0		
0	1	0	0	0	0	$A_2 = 1, A_3 = 0$	
0	1	0	1	1	0		
0	1	1	0	1	0		
0	1	1	1	1	0		
1	0	0	0	1	0	$A_2 = 0, A_3 = 1$	
1	0	0	1	0	0		
1	0	1	0	0	0		
1	0	1	1	0	0		
1	1	0	0	0	0	$A_2 = 1, A_3 = 1$	
1	1	0	1	1	0		*S_2*
1	1	0	1	1	1		*S_idle*