Digital Logic Systems – Algorithmic State Machines- I

Assist. Prof. Özge ÖZTİMUR KARADAĞ

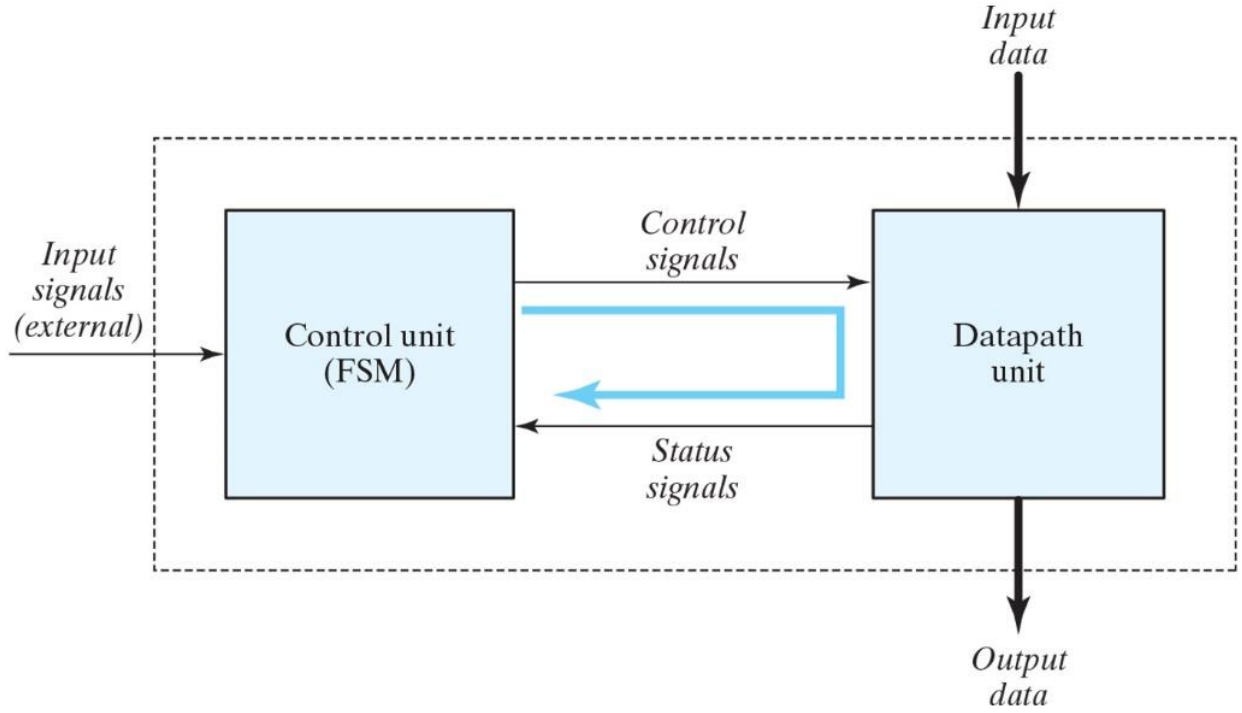
ALKÜ

Logic Design in Digital Systems

- Binary information in a digital system:
- data: discrete elements of information (binary words) that are manipulated by performing arithmetic, logic, shift, and other similar data-processing operations. These operations are implemented with digital components such as adders, decoders , multiplexers, counters. and shift registers.
- control: provides command signals that coordinate and execute the various operations in the data section in order to accomplish the desired data-processing tasks.
- The logic design of a digital system can be divided into two distinct parts:
  - the design of the digital circuits that perform the data -processing operations
  - the design of the control circuits that determine the sequence in which the various actions are performed

Control Logic and the Datapath

- Interaction between the control logic and the datapath in digital systems:
- The *datapath unit,* manipulates data in registers according to the system's requirements
- *The control unit* issues a sequence of commands to the datapath unit.



Control Logic

- generates the signals for sequencing the operations in the datapath unit
- is a finite state machine (FSM). i.e. a synchronous sequential circuit.
- Depending on status conditions and other external inputs the FSM goes to its next state to initiate other operations.

Hardware Algorithm

- The control sequence and datapath tasks of a digital system are specified by means of a hardware algorithm.
- An algorithm consists of a finite number of procedural steps that specify how to obtain a solution to a problem.
- A hardware algorithm is a procedure for solving the problem with a given piece of equipment.
- The most challenging and creative pan of digital design is the formulation of hardware algorithms for achieving required objectives

Hardware Algorithm

- How to represent an algortihm?
  - A flowchart is a convenient way to specify the sequence of procedural steps and decision paths for an algorithm.

- Algorithmic State Machine(ASM) Chart: flowchart that has been developed specifically to define digital hardware algorithms.

- State Machine: Another term for a sequential circuit which is the basic structure of a digital system.

Algorithmic State Machines(ASM)

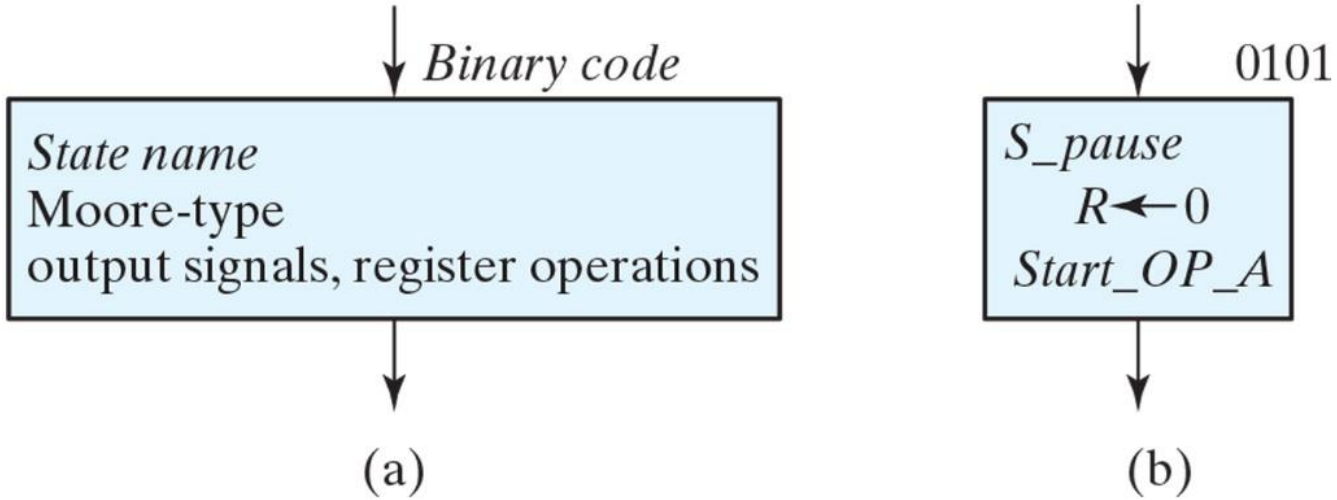
- A conventional flowchart that describes the procedural steps and decision paths of an algorithm in a sequential manner, without taking into consideration their time relationship.
- The ASM chart describes the sequence of events, as well as the timing relationship between the states of a sequential controller and the events that occur while going from one state to the next (i.e., the events that are synchronous with changes in the state).
- How is the ASM used for design?
  - Let’s introduce its parts.

ASM Chart

- A special flowchart for representing sequential processes in a digital system.
- Three main items:
  - State box
  - Decision box
  - Conditional box

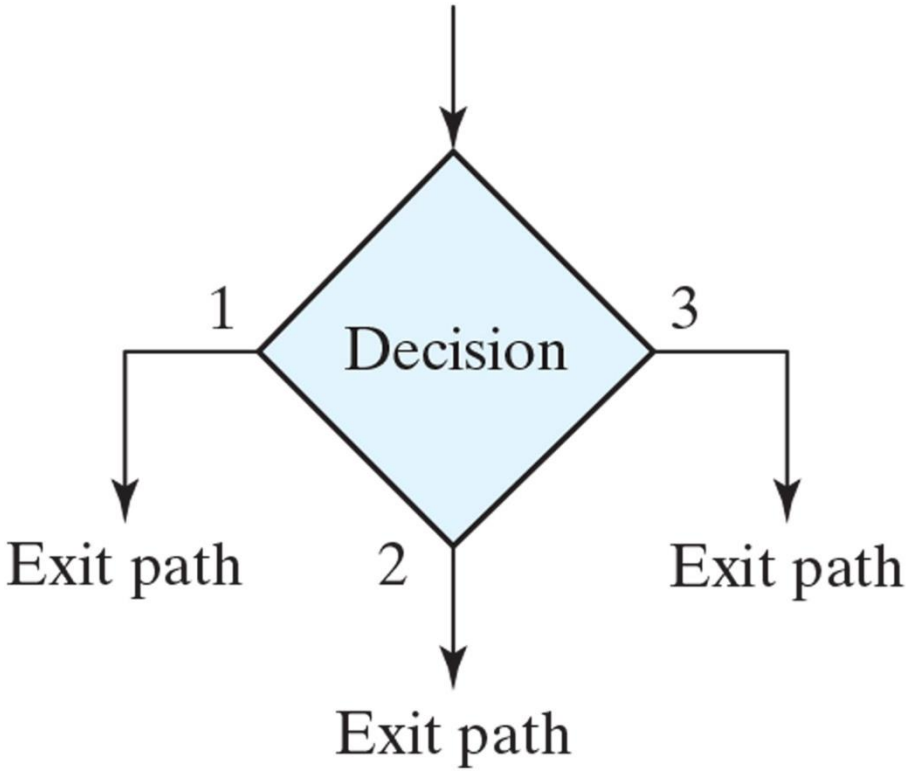
ASM Chart

- State Box
  - a)general representation
  - b)example



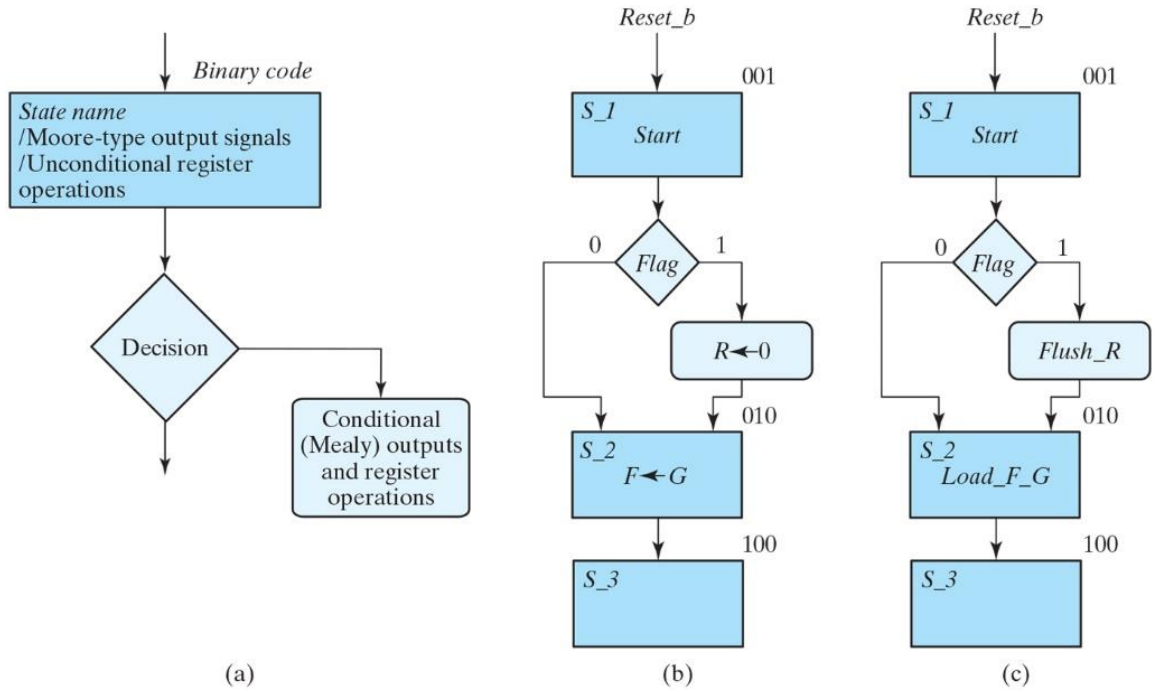
ASM Chart

- Decision Box describes the effect of an input (i.e .• a primary, or external, input or a status. or internal, signal) on the control subsystem.
  - Input to be tested is placed in the box.



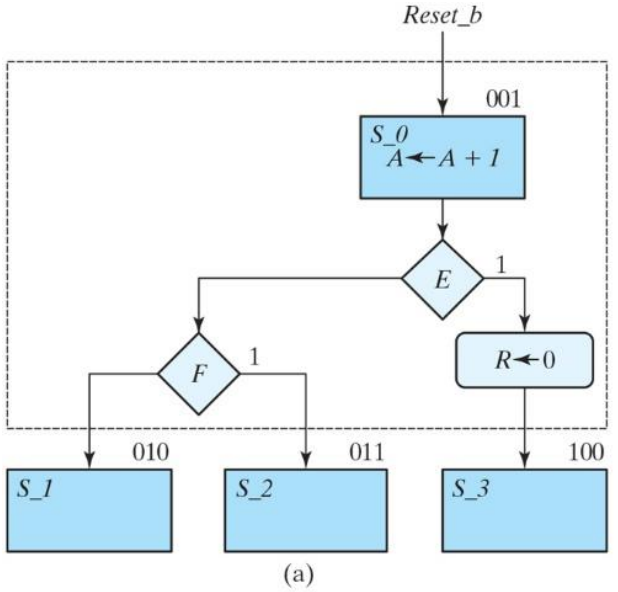
ASM Chart

- Conditional Box:
  - Has rounded corner.
  - The input path to the conditional box must come from one of the exit paths of a decision box.
  - The outputs listed inside the conditional box are generated as Mealy-type signals during a given state; the register operations listed in the conditional box. are associated with a transition from the state.
  - a)general representation
  - b)sample (data path and control)
  - c)sample (control)



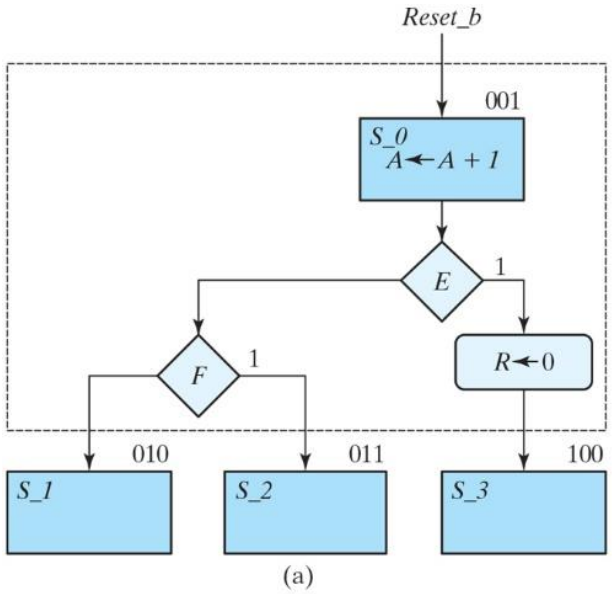
ASM Chart

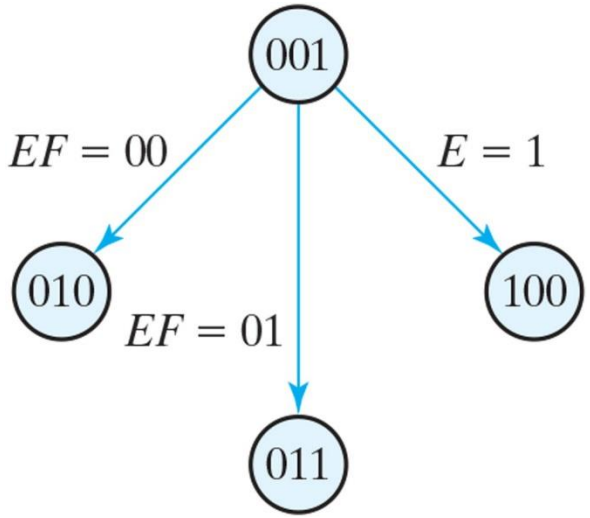
- ASM Block: An ASM block is a structure consisting of one state box and all the decision and conditional boxes connected to its exit path.
- An ASM block has one entrance and any number of exit paths represented by the structure of the decision boxes.
- An ASM chart consists of one or more interconnected blocks.
- A state box without any decision or conditional boxes constitutes a simple block.
- Each block in the ASM chart describes the state of the system during **one clock-pulse interval**.
- The operations within the state and conditional boxes initiated by a common clock pulse when the state of the controller transitions from *S_0* to its next state. The same clock pulse transfers the system controller to one of the next states, S_1. S_2. or S_3. as dictated by the binary values of *E* and *F.*



ASM Chart

- The ASM chart is similar to a state diagram.
- Each state block is equivalent to a state in a sequential circuit.
- The decision box is equivalent to the binary information written along the directed lines that connect two states in a state diagram.





ASM Chart

- Symbolic representation of register operations

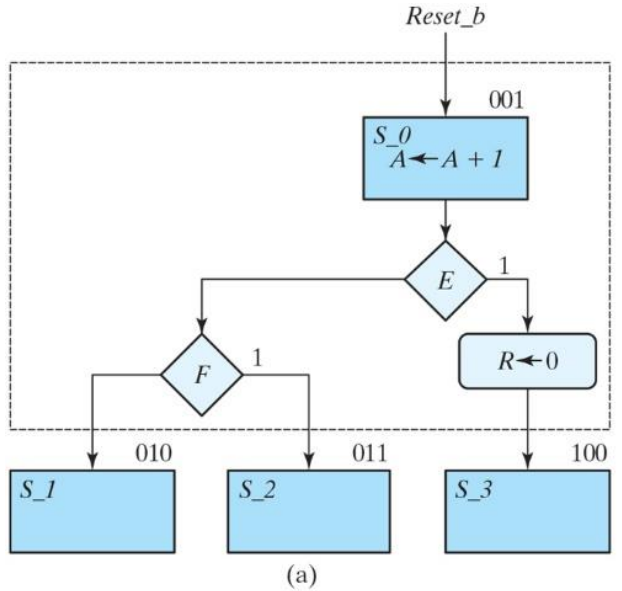
Symbolic Representation	Definition
A←B	Transfer content of B to register A.
R←0	Reset R
F←1	Set F
A←A+1	Count up A
A←A-1	Count down A
A←A+B	Add content of register B to A

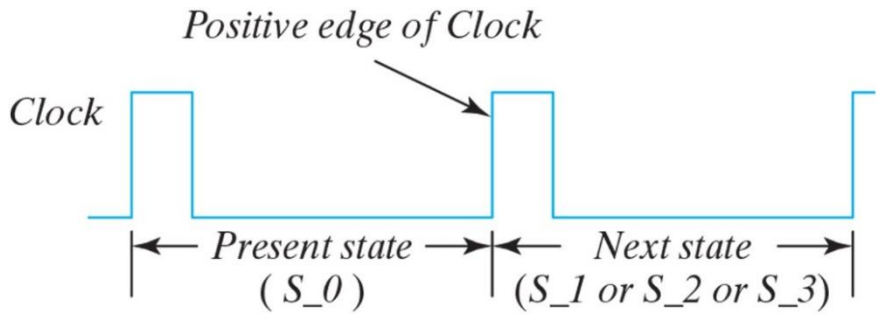
Considering the Timing

- All the registers and flip-flops are controlled with a single clock generator.
- Clock signals are applied to all registers in datapath and control logic.
- It is assumed that all clock signals are synchronized and all flip-flops are positive edge triggered.
- The difference between a classical flow chart and an algorithmic state machine is the interpretation about the timing.

Considering the Timing

- If this was a regular flow chart, processes could be assumed to be successive.
- An ASM block is considered as a single unit. All the processes in the block take place synchronously while the system goes from state S_0 to the next state.
- In the example, the following operations occur **simultaneously** at the next positive edge of the clock:
  - Register *A* is incremented.
  - If E=1 register R is cleared.
  - Control transfers to the next state.



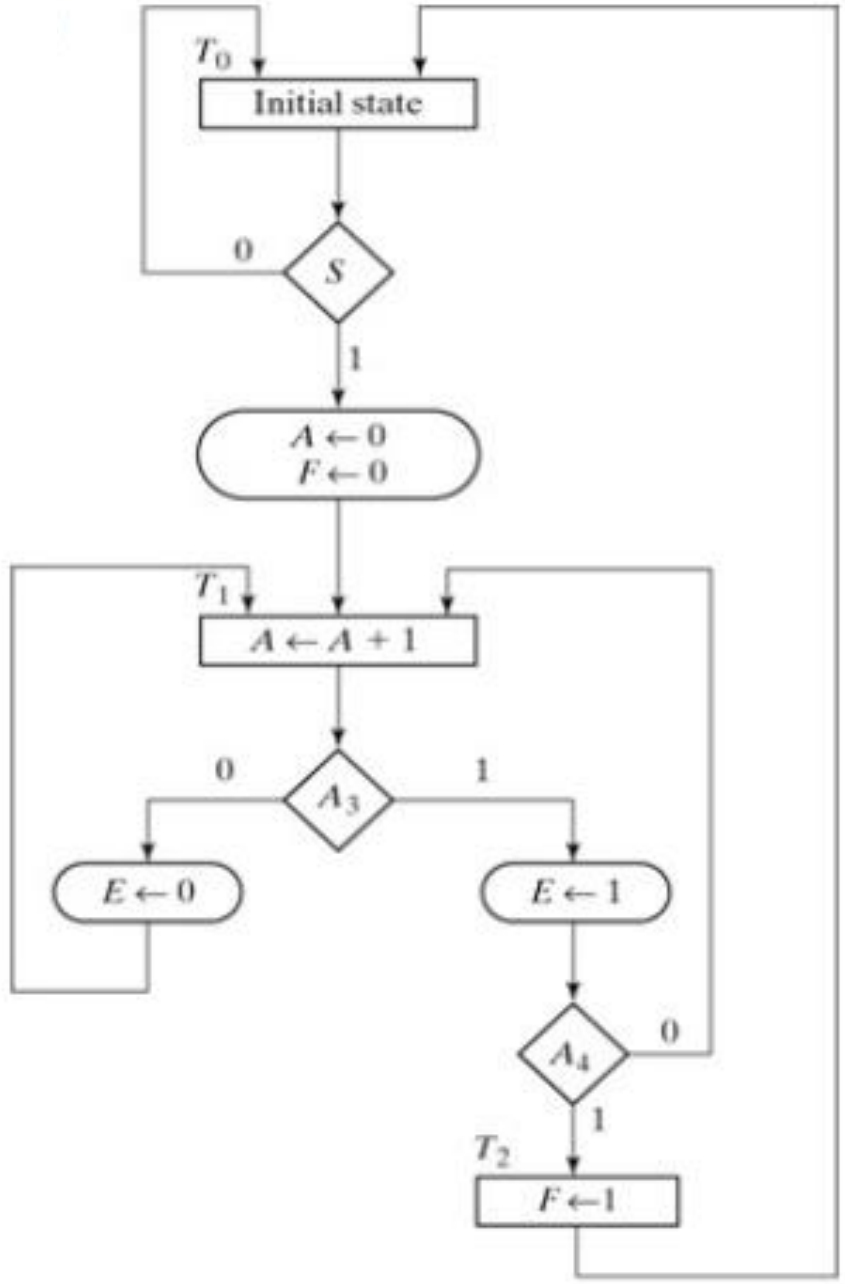


Design Example

- We want to design a digital system with two flip-flops; E and F and a 4 bit binary counter A.
- Flip-flops in A are A_4,A_3,A_2,A_1.(A_4 for the most significant bit)
- A signal *S,* initiates the system's operation by clearing the counter *A* and flip- flop *F.*
- At each subsequent clock pulse the counter is incremented by 1 until the operations stop.
- A_3 and A_4 determine the sequence of operations:
  - If A_3=0 E is cleared to 0 and count continues.
  - If A_3=1 E is set to 1; then if A_4=0 the count continues, if A_4=1 F is set to 1 on the next clock pulse and the system stops counting.

Design Example…

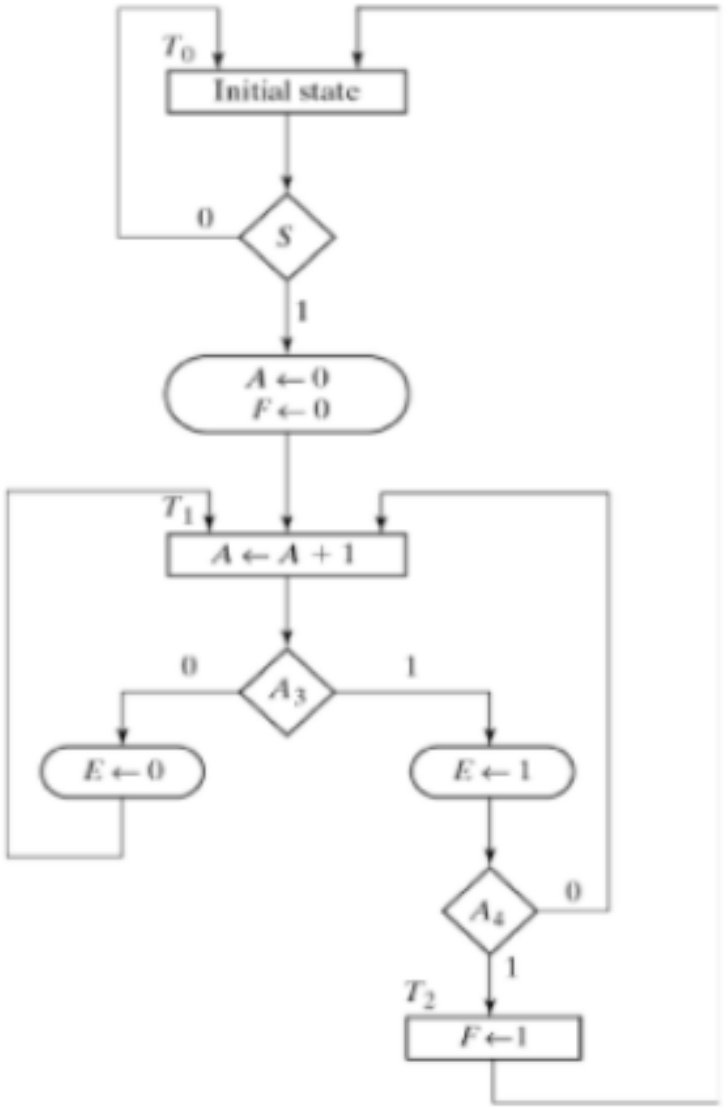
- We want to design a digital system with two flip-flops; E and F and a 4 bit binary counter A.
- Flip-flops in A are A_4,A_3,A_2,A_1.(A_4 for the most significant bit)
- A signal *S,* initiates the system's operation by clearing the counter *A* and flip- flop *F.*
- At each subsequent clock pulse the counter is incremented by 1 until the operations stop.
- A_3 and A_4 determine the sequence of operations:
  - If A_3=0 E is cleared to 0 and count continues.
  - If A_3=1 E is set to 1; then if A_4=0 the count continues, if A_4=1 F is set to 1 on the next clock pulse and the system stops counting.



Design Example…

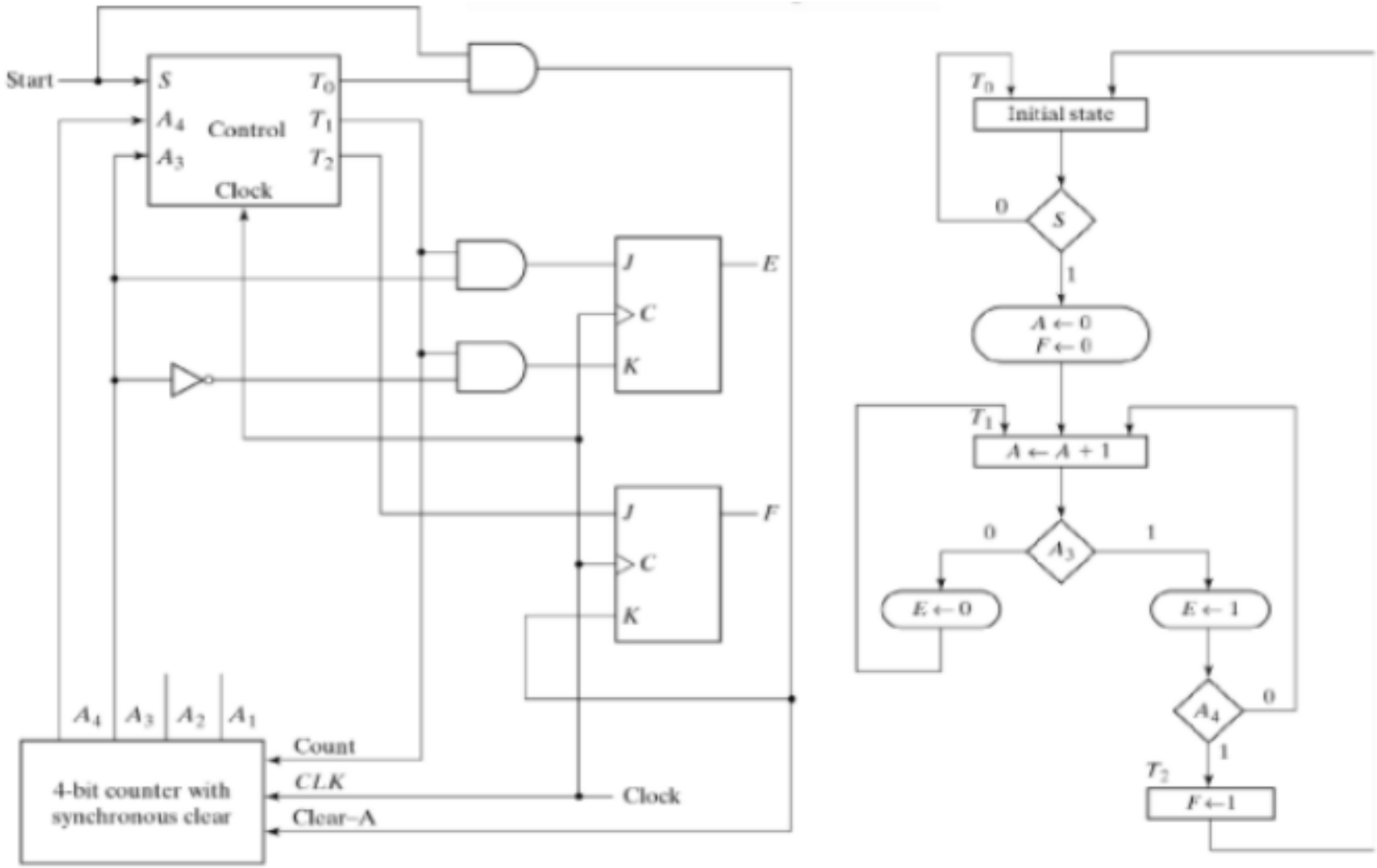
- Zama

Counter				Flip-flops		Conditions	State
A4	A3	A2	A1	E	F		
0	0	0	0	1	0	A3=0, A4=0	T_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	A3=1, A4=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	A3=0, A4=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		
1	1	0	1	1	0		T_2
1	1	0	1	1	1		T_0



Design Example..

- Datapath



References

- Morris Mano, Digital Design