Algorithmic State Machines (ASM)

Binary information in Digital Systems:

Data
Control

➔Data processing tasks:
Addition, decoding, counting etc

➔Control information provides command signals monitoring data processing tasks.

Various modules are interconnected to form digital system

Logic Design consists of:

➔**Design of Data processing circuits**

➔**Design of control circuits**

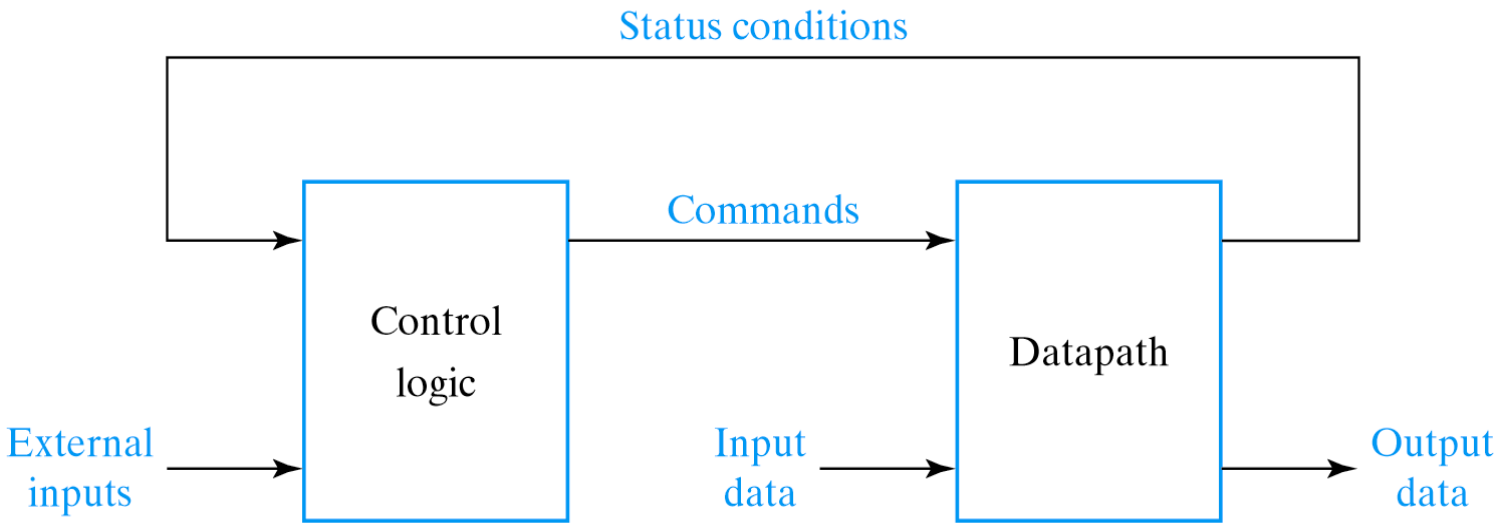


Fig. 8-2 Control and Datapath Interaction

➔**Control sequence and data processing Tasks are specified by means of Hardware Algorithm.**

➔**A Special Flowchart developed to design Digital hardware algorithms is called Algorithmic State Machines (ASM)**

➔**A conventional flowchart describes sequence of Procedural steps without concern for their time relationship**

➔**An ASM chart describes the sequence of events *as well as* the timing relationship between states of sequential controller**

ASM CHART
Composed of three basic elements

➔**The State box**

➔**The decision box**

➔**The conditional box**

A state is indicated by the state box within which Register operations are written

Or

The output signal names that the control generates while being in the state are written.

➔The state is given a symbolic name and a binary code is assigned to state

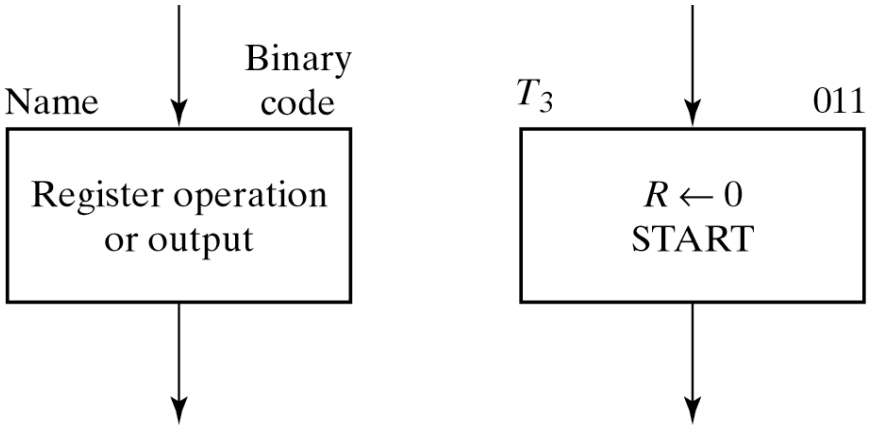


Fig. 8-3 State Box

'START' may indicate an output signal that starts an operation

➔**Decision box indicates the effect of an input on the control subsystem**

➔**Input condition to be tested is written inside the box**

➔**One exit path is taken if the condition is true and another if false**

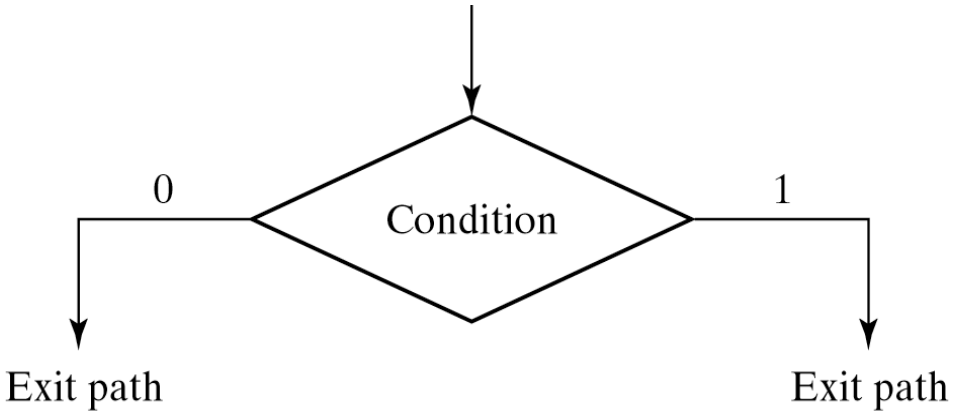


Fig. 8-4 Decision Box

➔**The conditional box in unique to ASM chart**

➔**Input path to conditional box comes from one of exit paths of a decision box**

➔**The register operations or outputs listed inside a conditional box are generated during a given State provided the input condition is satisfied.**

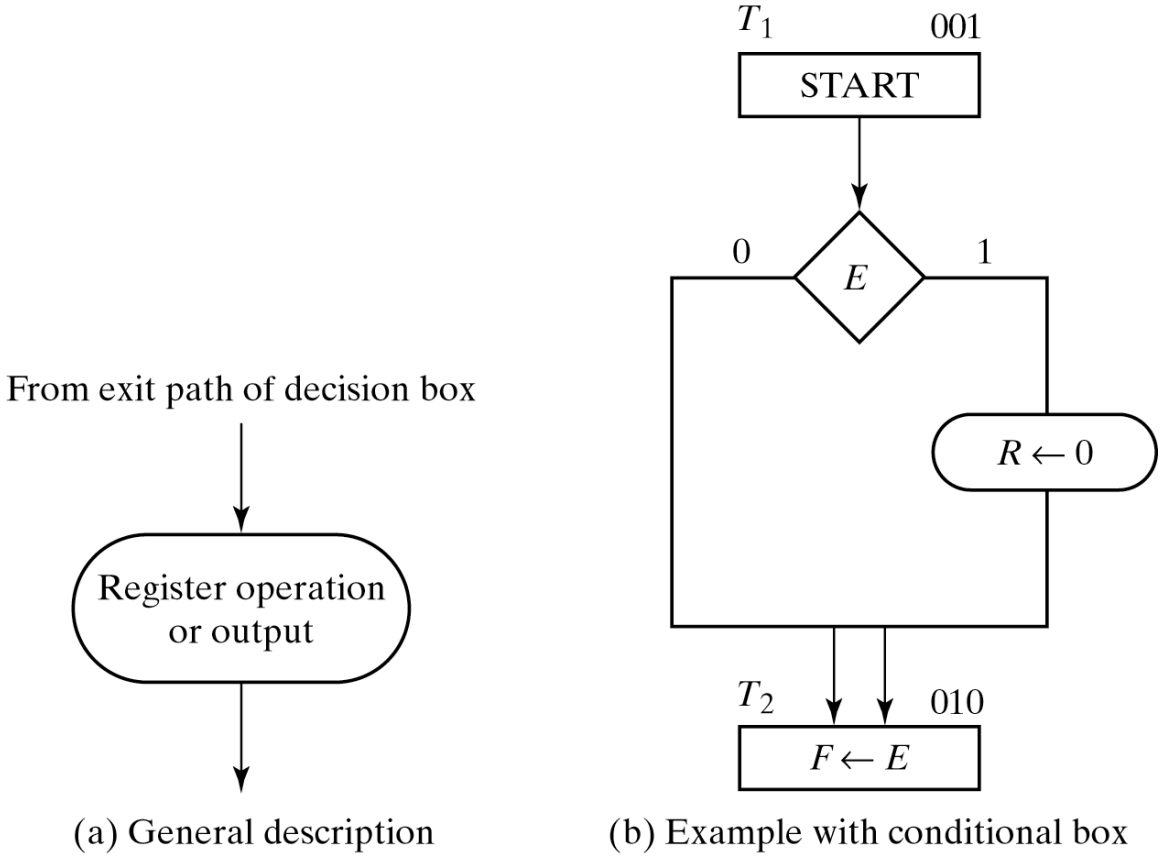


Fig. 8-5 Conditional Box

ASM Block

➔**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**

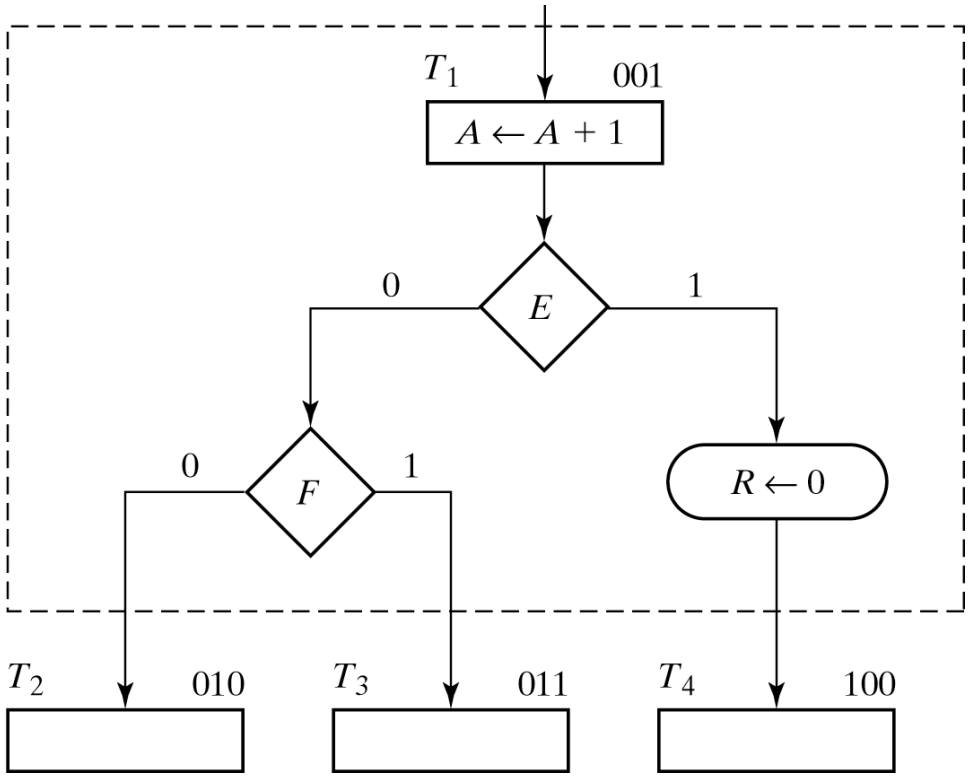


Fig. 8-6 ASM Block

➔Each ASM block in the ASM chart describes the state of the system during one clock pulse interval

➔The operations within the state and conditional boxes are executed with a common clock pulse while the system is in state T1.

➔The same clock pulse also transfers the system controller to one of the next states.

➔ASM chart is very similar to state diagram.

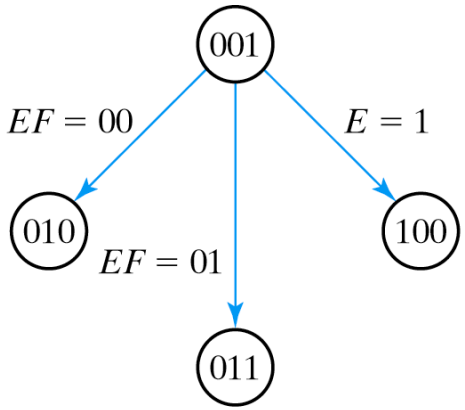


Fig. 8-7 State Diagram Equivalent to the ASM Chart of Fig. 8

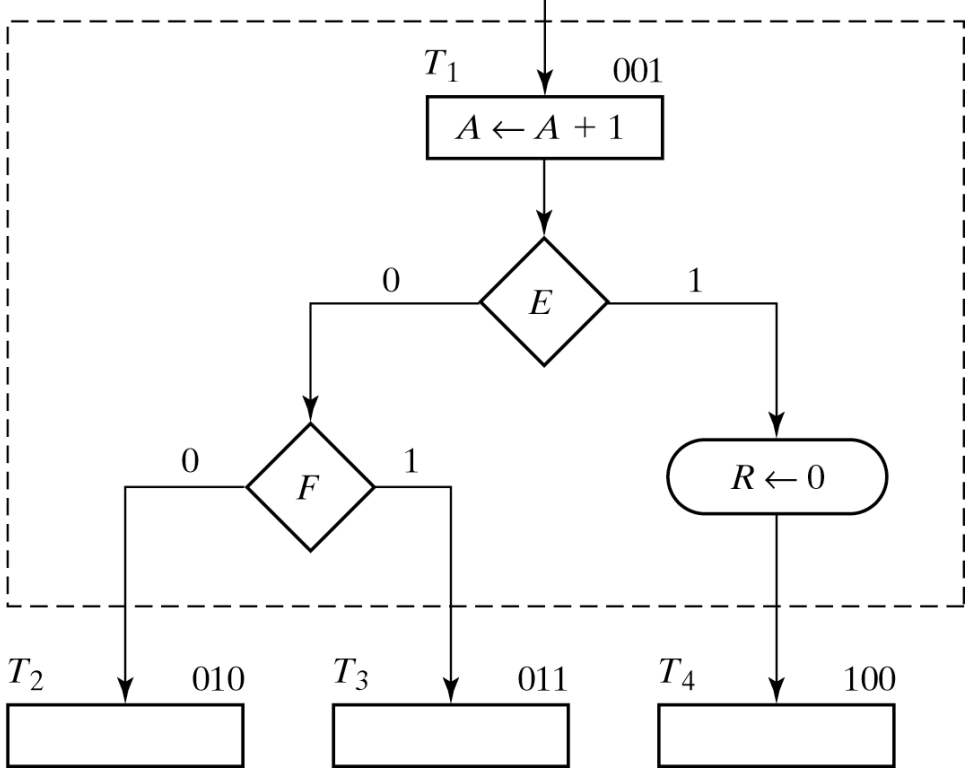


Fig. 8-6 ASM Block

- Major difference between a conventional flowchart and an ASM chart is interpreting the timing relations.
- In conventional flowchart the listed operations follow one after the other in time sequence.
- ASM chart consider the entire ASM block as one unit.
- All operations within a block must occur in synchronism during clock edge.

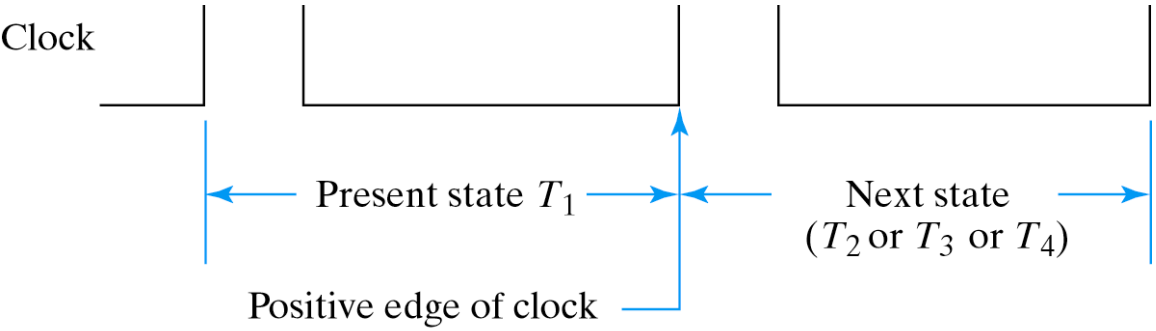


Fig. 8-8 Transition Between States

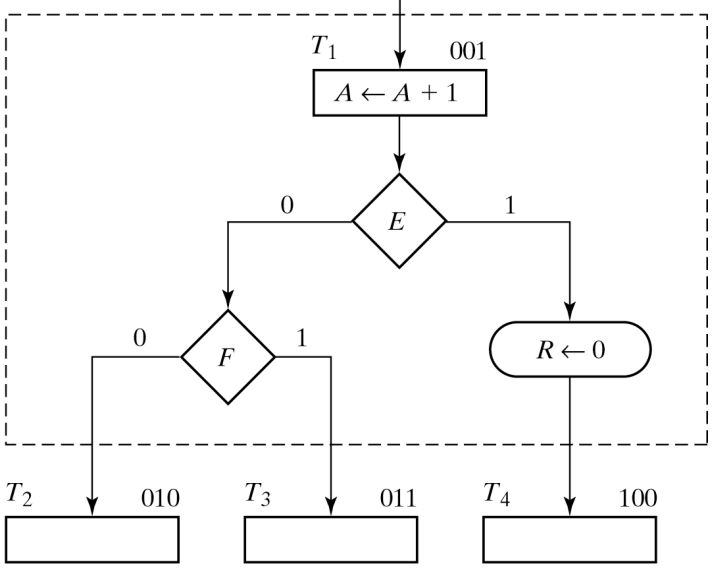


Fig. 8-6 ASM Block

During transition
A is incremented, If E = 1 , R is cleared
Depending on values of EF control transferred to T2 or T3 or T4

Design Example:

Design a Digital system with two FFs E and F, one 4-bit binary counter A (A4A3A2A1). A start signal S initiates the system operation by clearing counter A and flip-flop F. Counter incremented by one starting from next clock pulse, continues to increment until operations stop.

If A3 = 0, E is cleared to 0 and count continues

If A3 = 1, E set to 1; then if A4=0, count continues, but if A4 = 1, F set to 1 on *next clock pulse* and system stops counting. If S = 0 system remains in initial state, but if S = 1 operation cycle repeats.

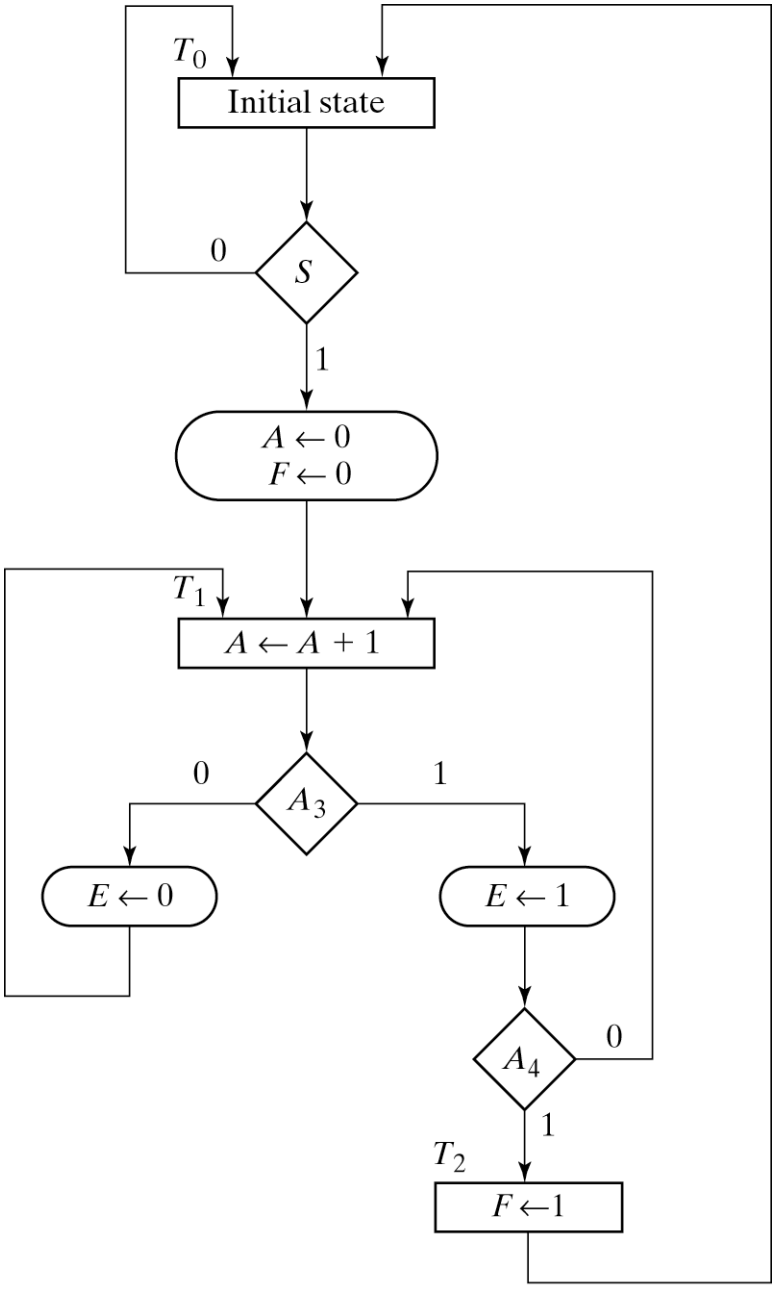


Fig. 8-9 ASM Chart for Design Example

➔**When no operations the system in initial state T0 waiting for S**

➔**When input S = 1, counter A and flip-flop F cleared and Counter goes to state T1**

➔**Block with T1 has two decision and two conditional boxes**

➔**Counter incremented with every clock pulse, at the same time one of three operations occur during clock transition**

Either E cleared and control stays in T1 (A3 = 0); or
E is set and control stays in T1 (A3A4 = 10); or
E is set and control goes to T2 (A3A3 = 11).

➔**When in T2 F is set and circuit goes to T0.**

- Every block in the ASM chart specifies the operations performed during one common clock pulse.
- Operations specified within the state and conditional boxes in the block are performed in the datapath section.
- The change from one state to next is performed in the control section.

Counter				Flipflops			
A4	A3	A2	A1	E	F	Conditions	State
0	0	0	0	1	0	A3= 0, A4=0	T1
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	

Counter				Flipflops			
A4	A3	A2	A1	E	F	Conditions	State
0	1	0	0	0	0	A3=1, A4= 0	T1
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	

Counter				Flipflops			
A4	A3	A2	A1	E	F	Conditions	State
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	A3=1, A4 = 1	

Counter				Flipflops			
A4	A3	A2	A1	E	F	Conditions	State
1	1	0	0	0	0	A3=1, A4 = 1	
1	1	0	1	1	0		T2
1	1	0	1	1	1		T0

- The requirements of design of the datapath are specified inside the state and conditional boxes

- Control logic is determined by the decision boxes and required state transitions.

Data path consists of
-4 bit binary counter
-Two flip-flops
-Gates

Counter is incremented with every clock cycle when control is in state T1.
Cleared when control is in state T0 and S is equal to 1. ➔ uses an AND gate
The other two conditional operations use two other AND gates for setting or clearing E.
F is set unconditionally during T2.

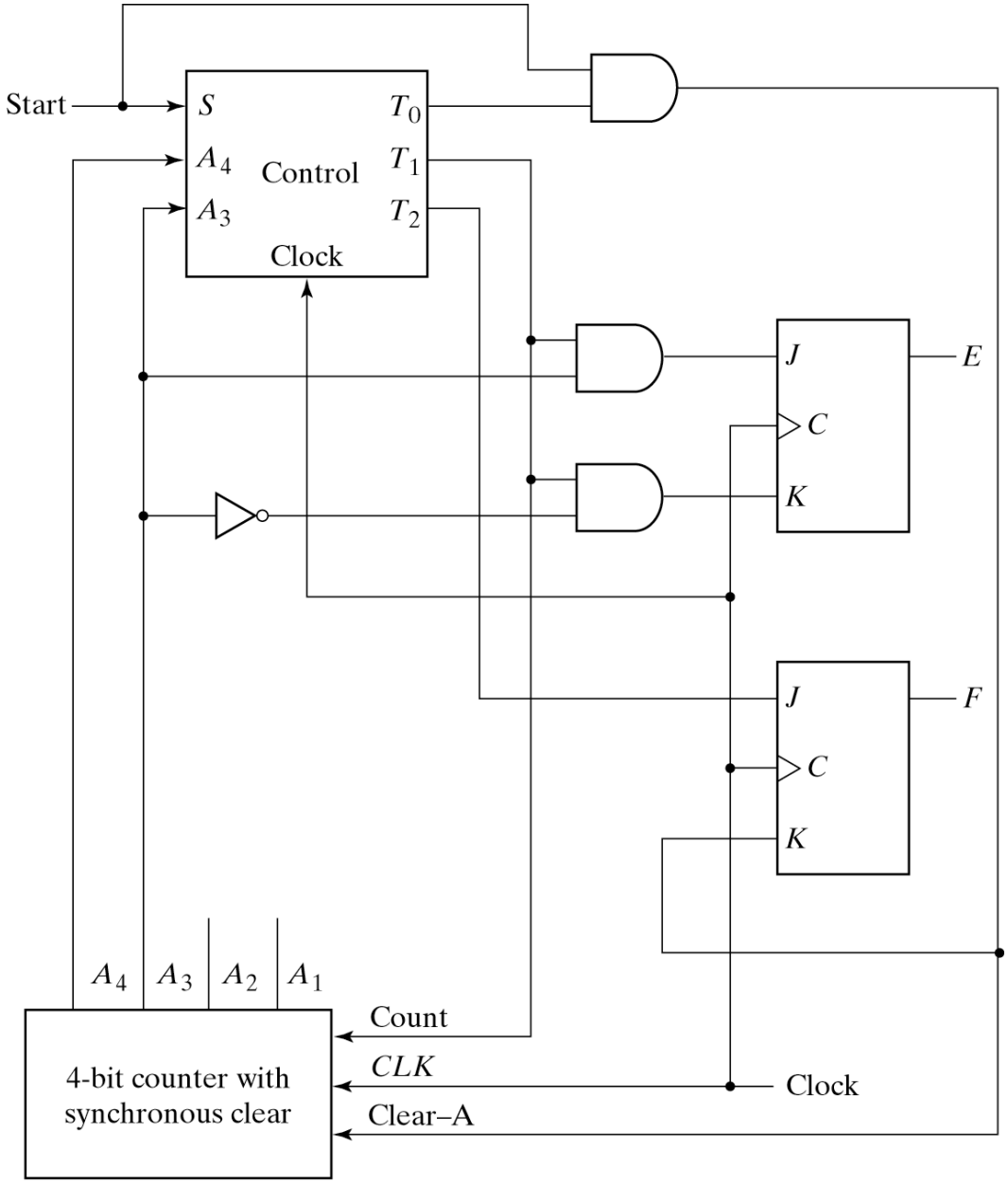
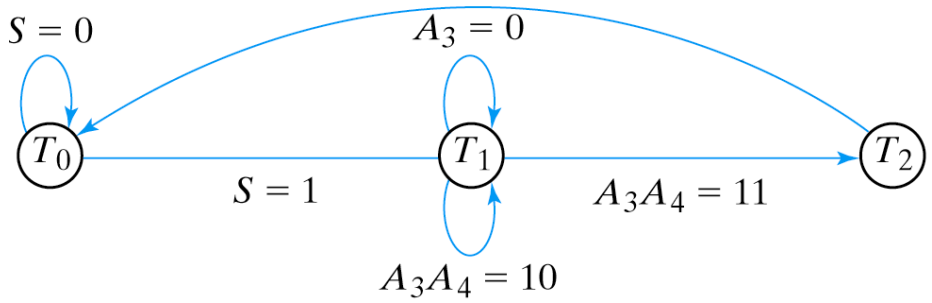


Fig. 8-10 Datapath for Design Example



(a) State diagram for control

Register transfer operations:

T0: if (S=1) then A ← 0, F ←0

T1: A ← A + 1

if (A3 = 1) then E ← 1
if (A3 = 0) then E ← 0

T2: F ← 1

State Table:

Present-State Symbol	Present State G1	G0	Inputs S	A3	A4	Next State G1	G0	Outputs T0	T1	T2
T0	0	0	0	X	X	0	0	1	0	0
T0	0	0	1	X	X	0	1	1	0	0
T1	0	1	X	0	X	0	1	0	1	0
T1	0	1	X	1	0	0	1	0	1	0
T1	0	1	X	1	1	1	1	0	1	0
T2	1	1	X	X	X	0	0	0	0	1

Design Using D FFs.

Flip-flop Inputs
DG1 = T1A3A4

DG0 = T0S +T1

Output Functions

T0 = G0’
T1 = G1’G0
T2 = G1

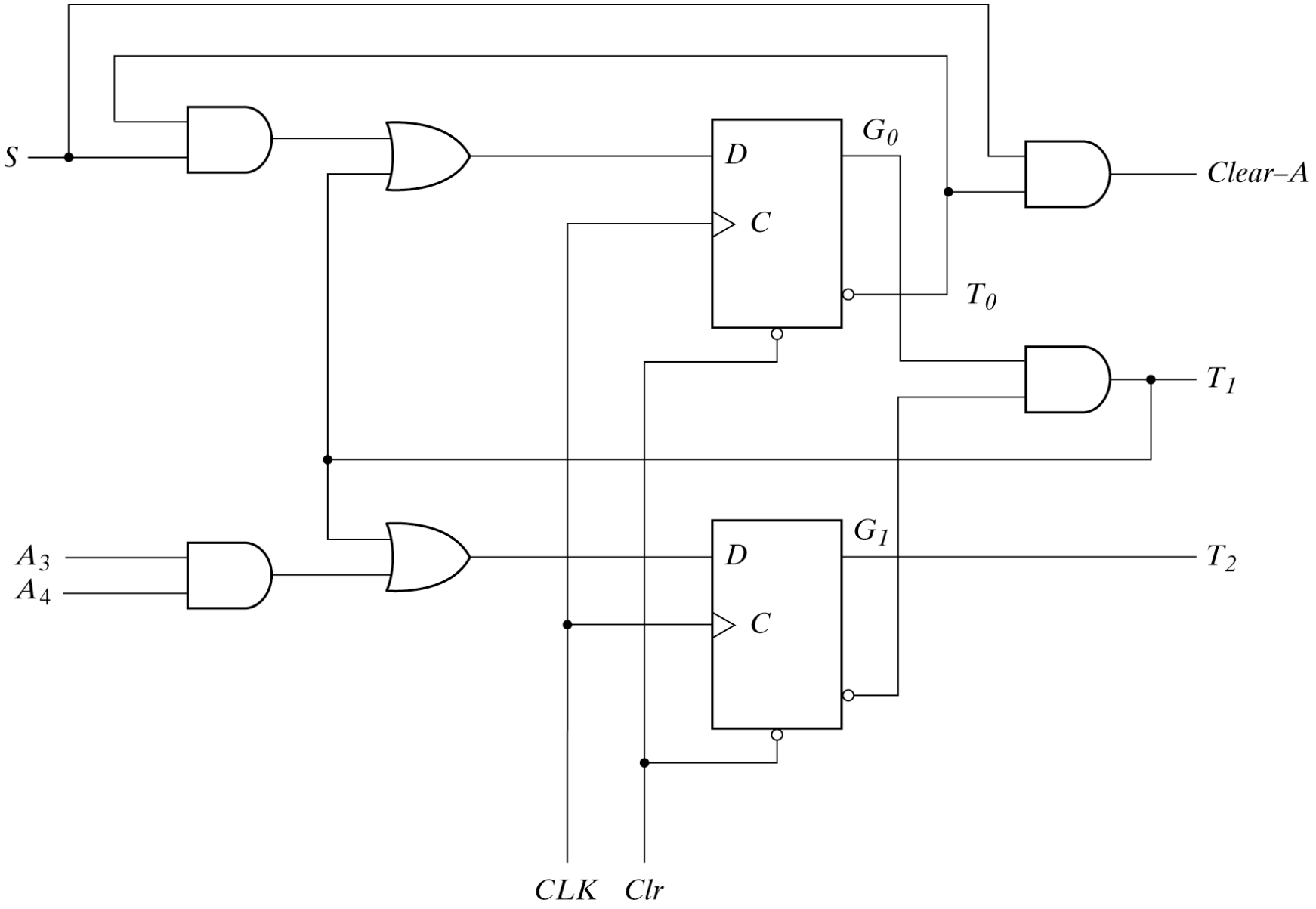


Fig. 8-12 Logic Diagram of Control