Sharif University of Technology
Department of Computer Engineering

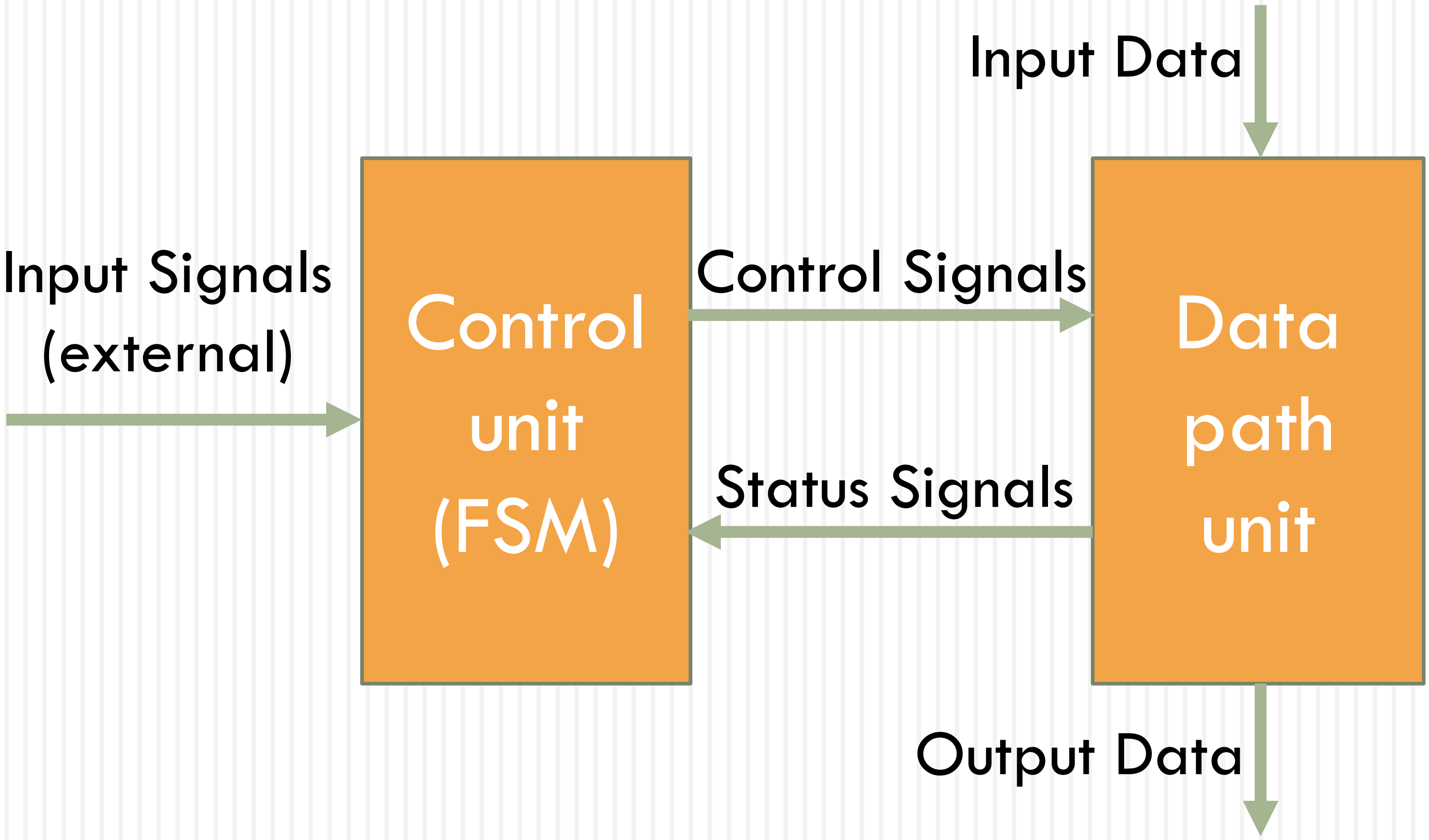
Digital System Design
Algorithmic State Machine (ASM)

Siavash Bayat-Sarmadi

Introduction

- Digital system design
  - Design of digital circuits performing data-processing operations
    - Arithmetic, logic, etc.
  - Design of control circuit
    - Determines the sequence of performing above operations

Introduction (Cont.)

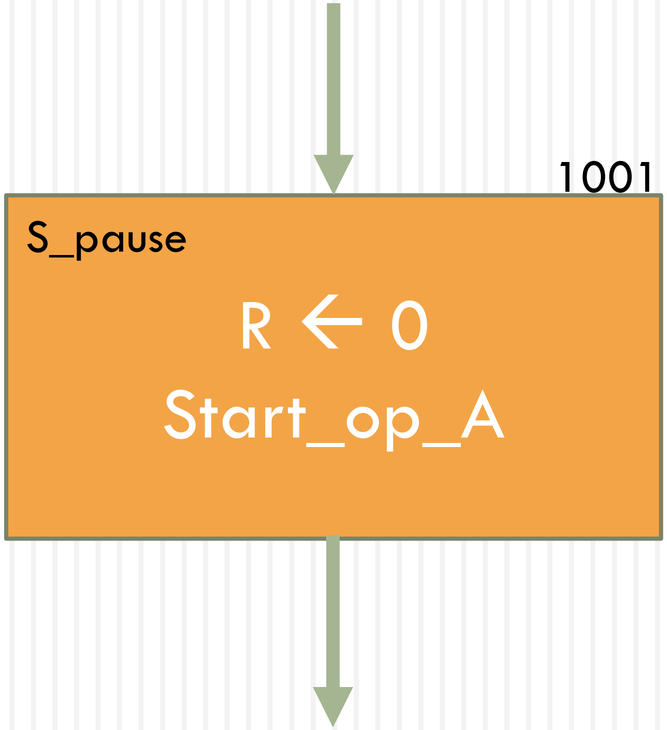


Algorithmic State Machine (ASM) Chart

- A flowchart to define digital hardware alg.
- Resembles conventional flowchart
  - Different interpretation
- Demonstrates
  - State transitions
  - Events that occur while going from one state to another
- Three boxes
  - State box
  - Decision box
  - Conditional box

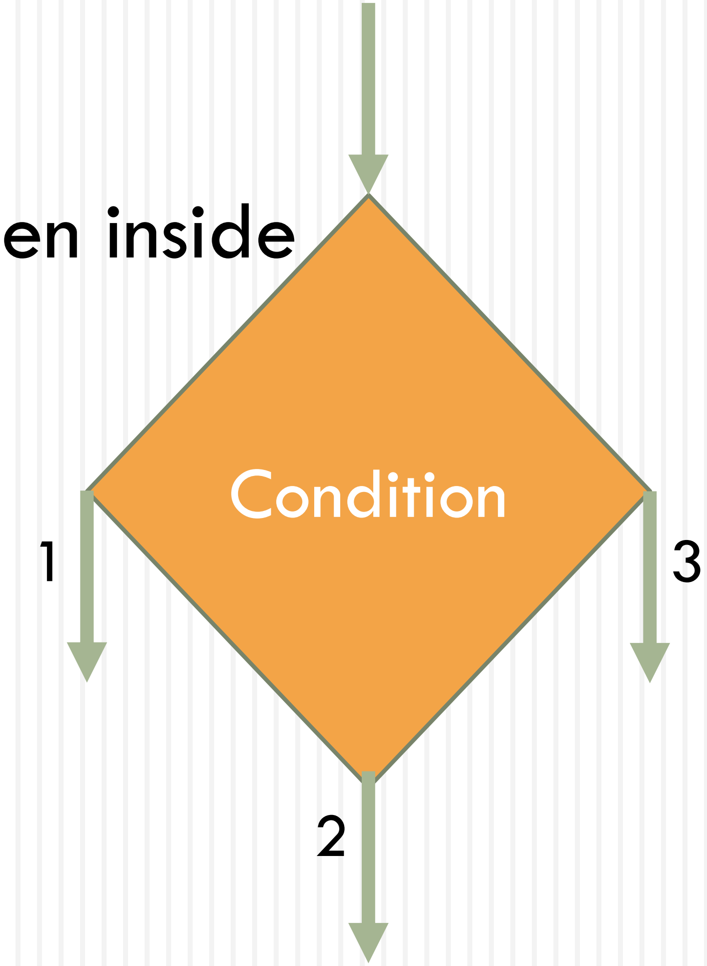
ASM Chart Boxes

- State box
  - A state in the control sequence
  - Contains
    - Register operations
      - Confusing
        - Executed while going to the next state
    - Moore outputs
  - Can be assigned
    - Name or symbol
    - Binary code



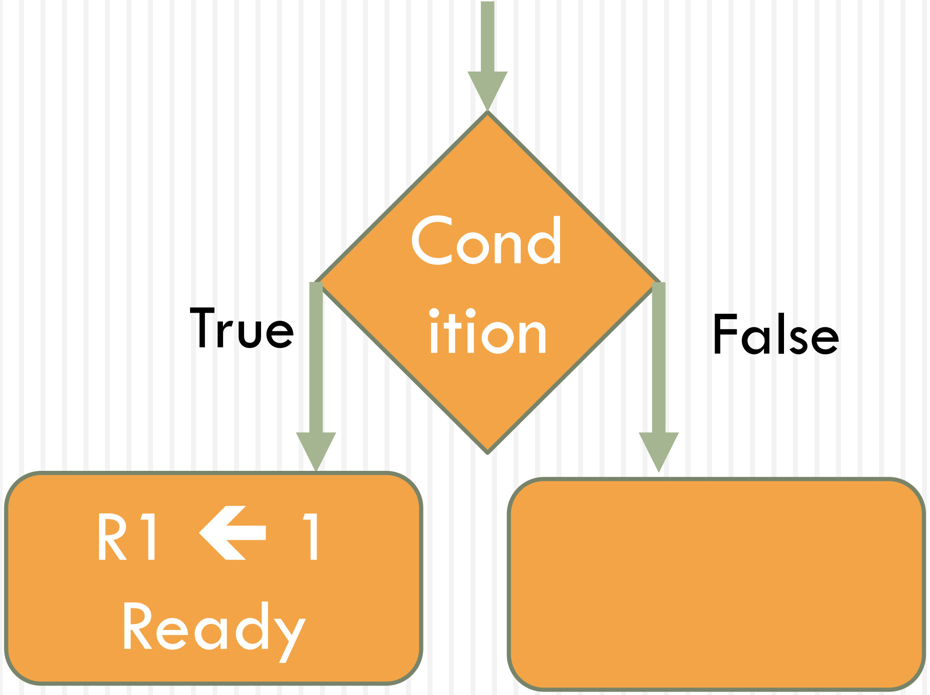
ASM Chart Boxes (Cont.)

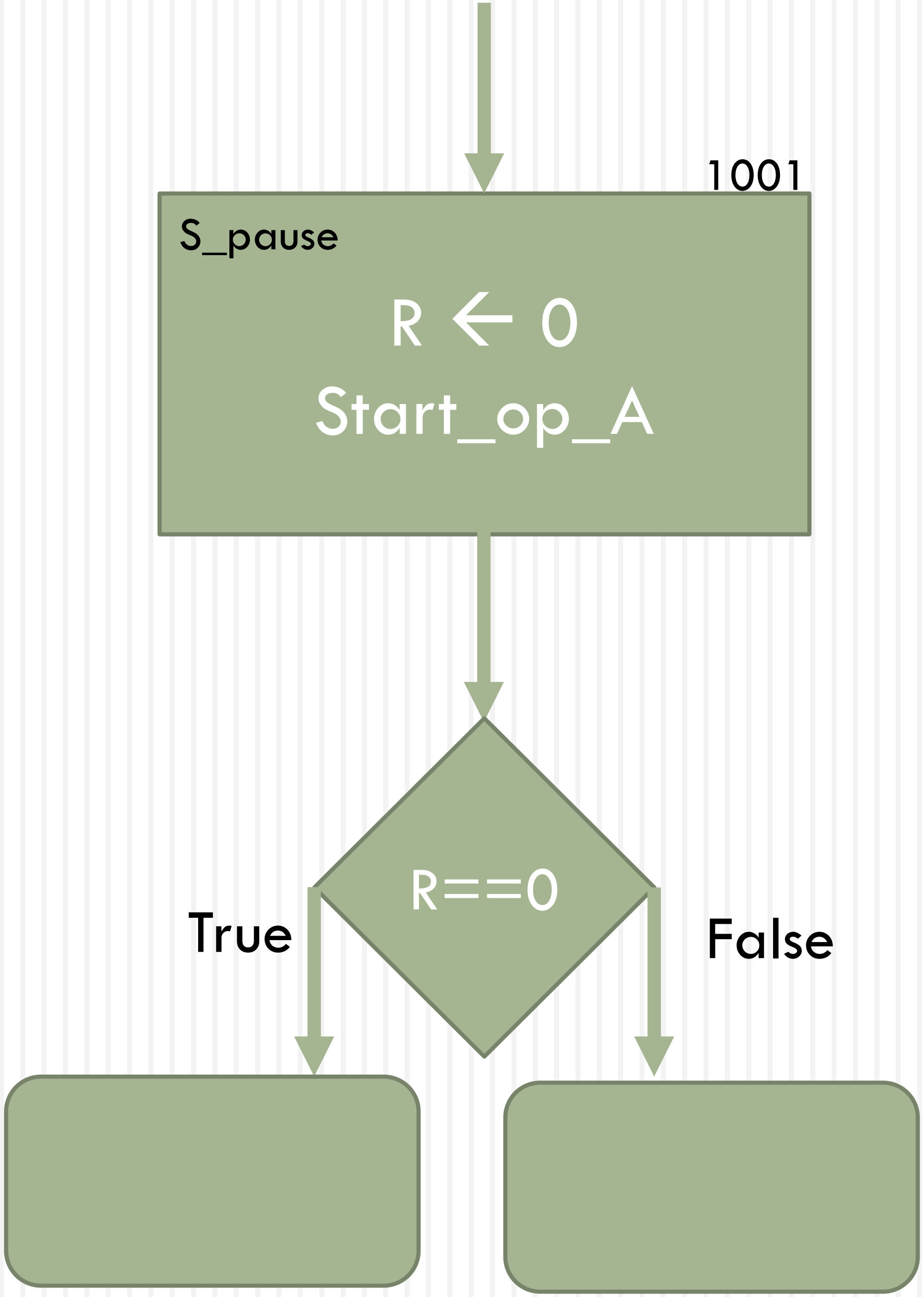
- Decision box
  - Two or more exit paths
  - Condition to be tested written inside



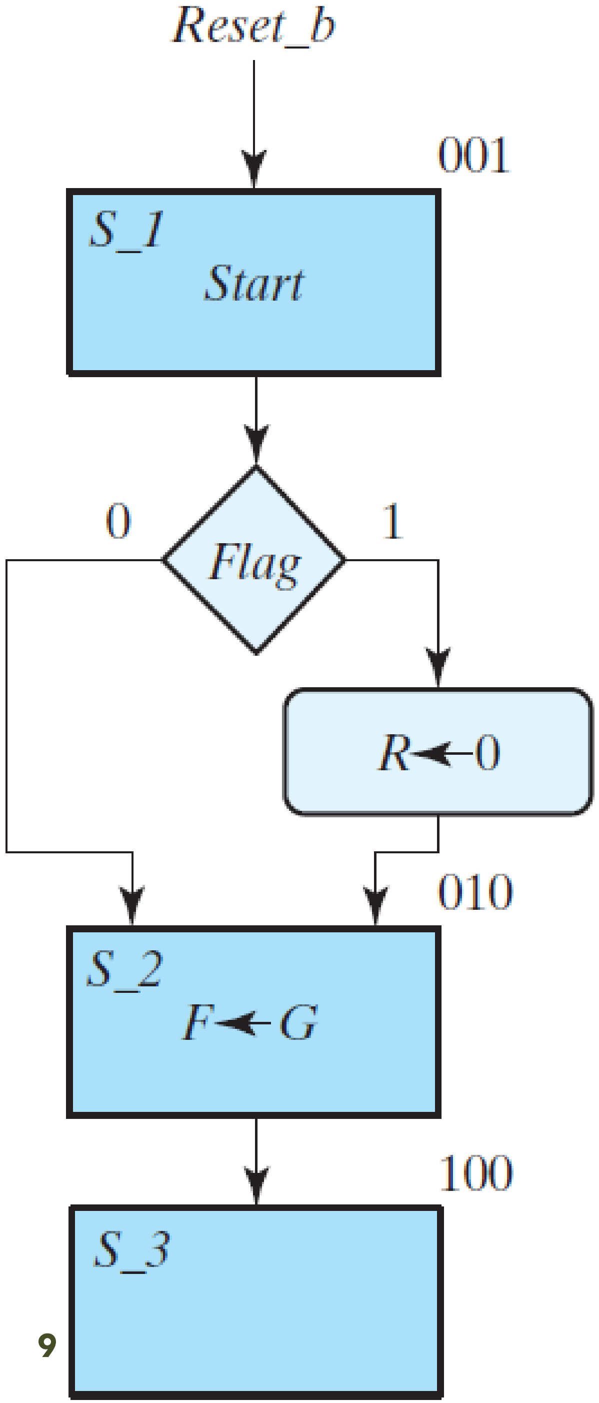
ASM Chart Boxes (Cont.)

- Conditional box
  - Unique to ASM chart
  - Must get its inputs from a decision box
  - Written inside
    - Mealy outputs
    - Register operations
      - State transition





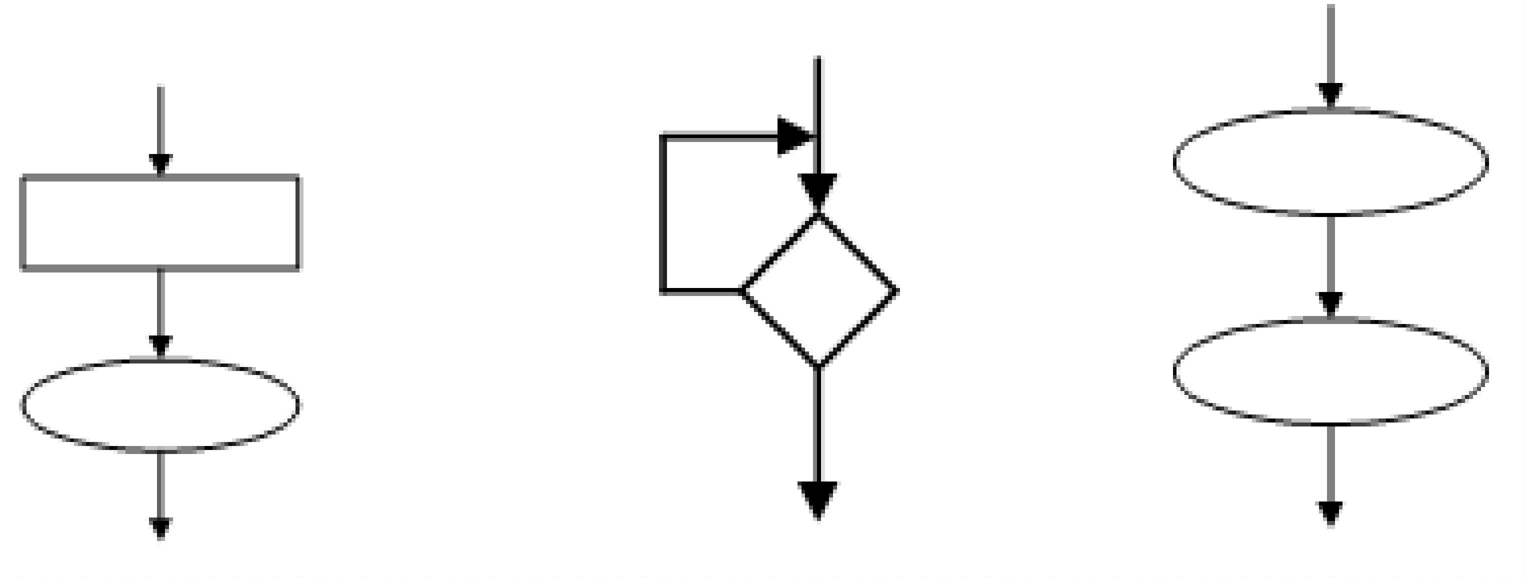
- Which path is taken?



- Control signal 'Start' asserted in S_1
- The value of 'Flag' is checked
  - If Flag == 1, R is to be cleared
  - Otherwise, remains unchanged
  - Next state is S_2

ASM Chart (Cont.)

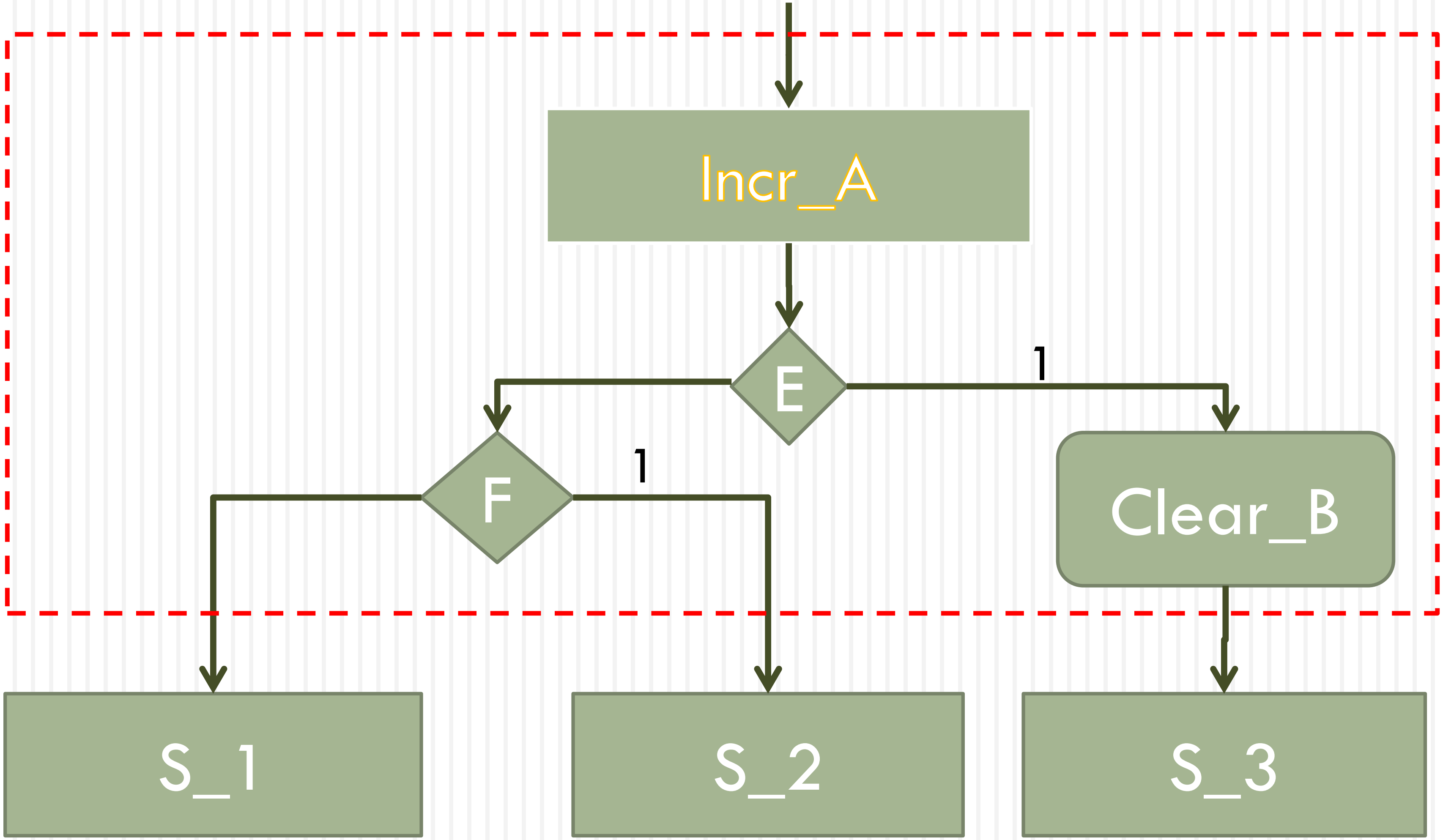
- Some illegal connections



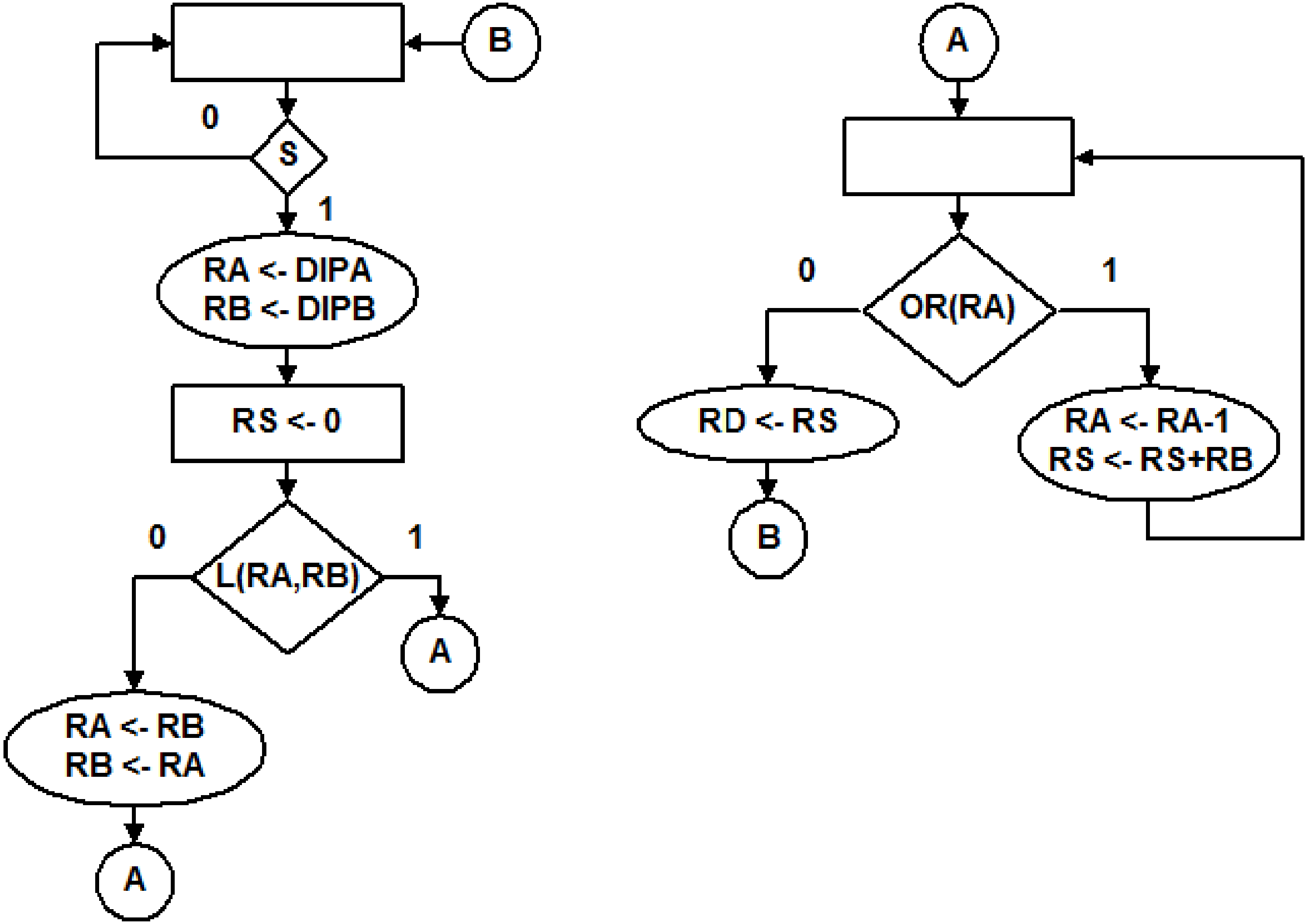
ASM Block

- One state box & all the decision and conditional boxes connected to the exit path of it
- Only one entrance
- Represents what happens in the system during one clock cycle

ASM Block (Cont.)



Example: Multiplier



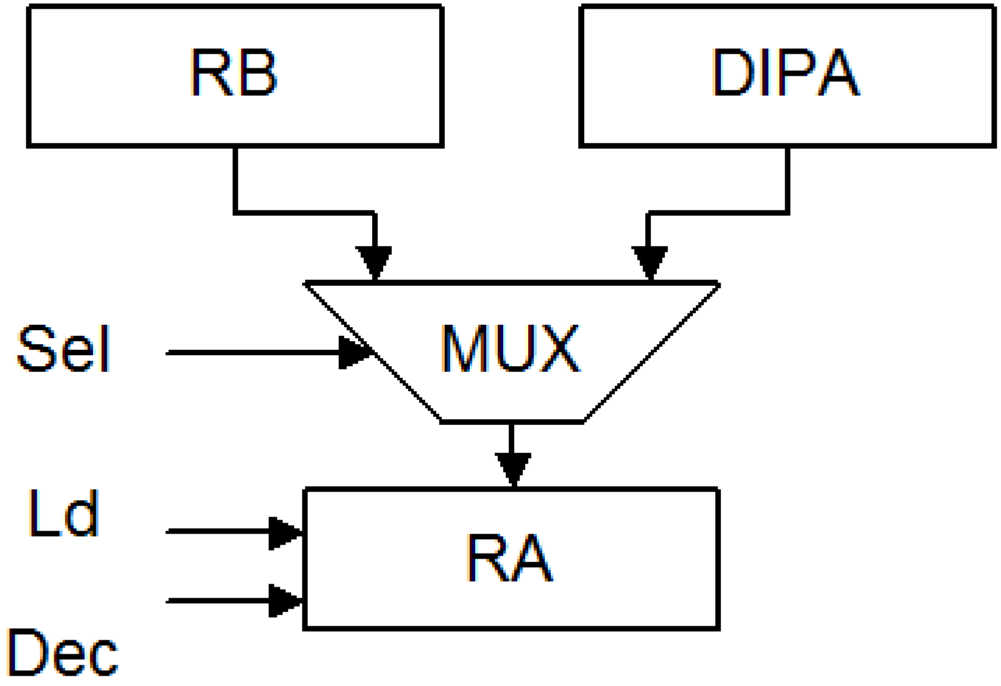
Example (Cont.)

- Datapath
  - For RA

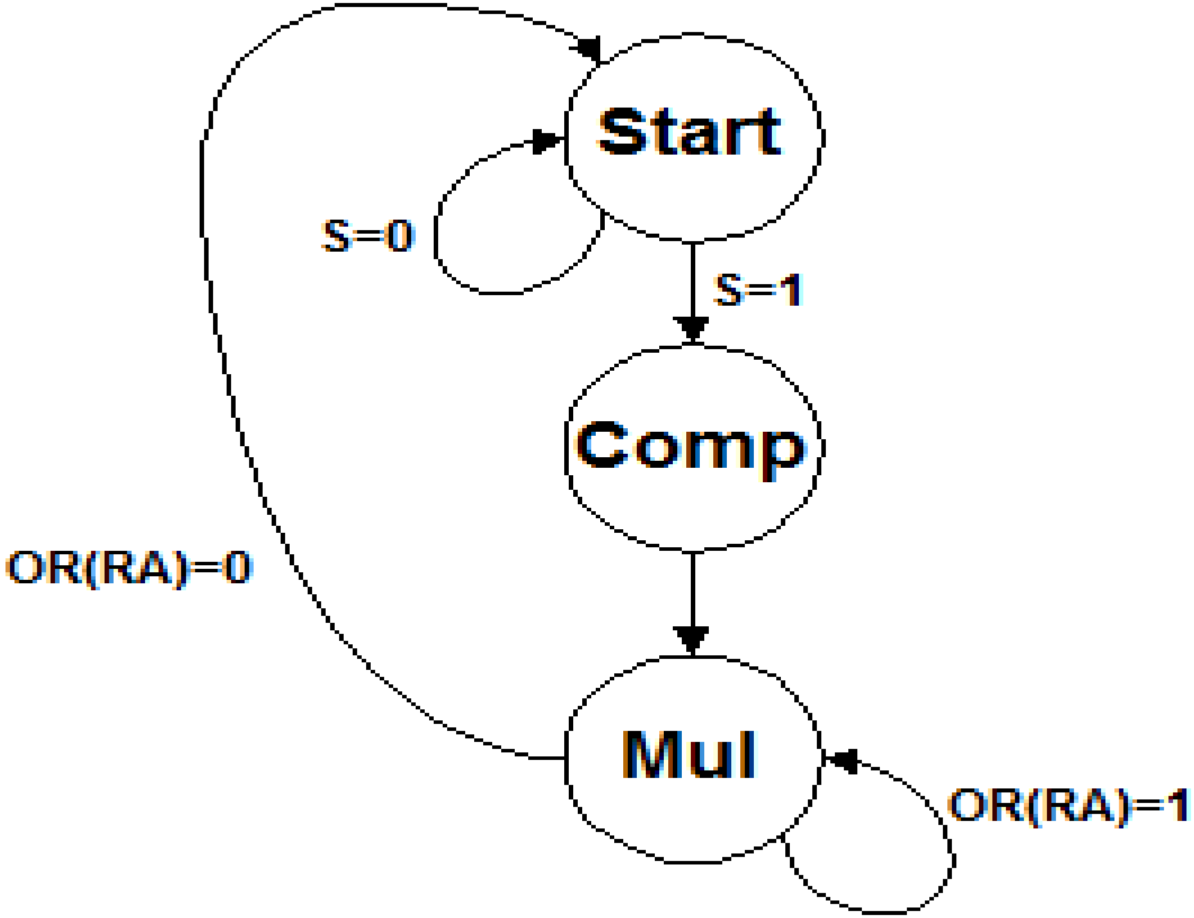
RA<-DIPA

RA<-RB

RA<-RA-1



Control Unit (Cont.)



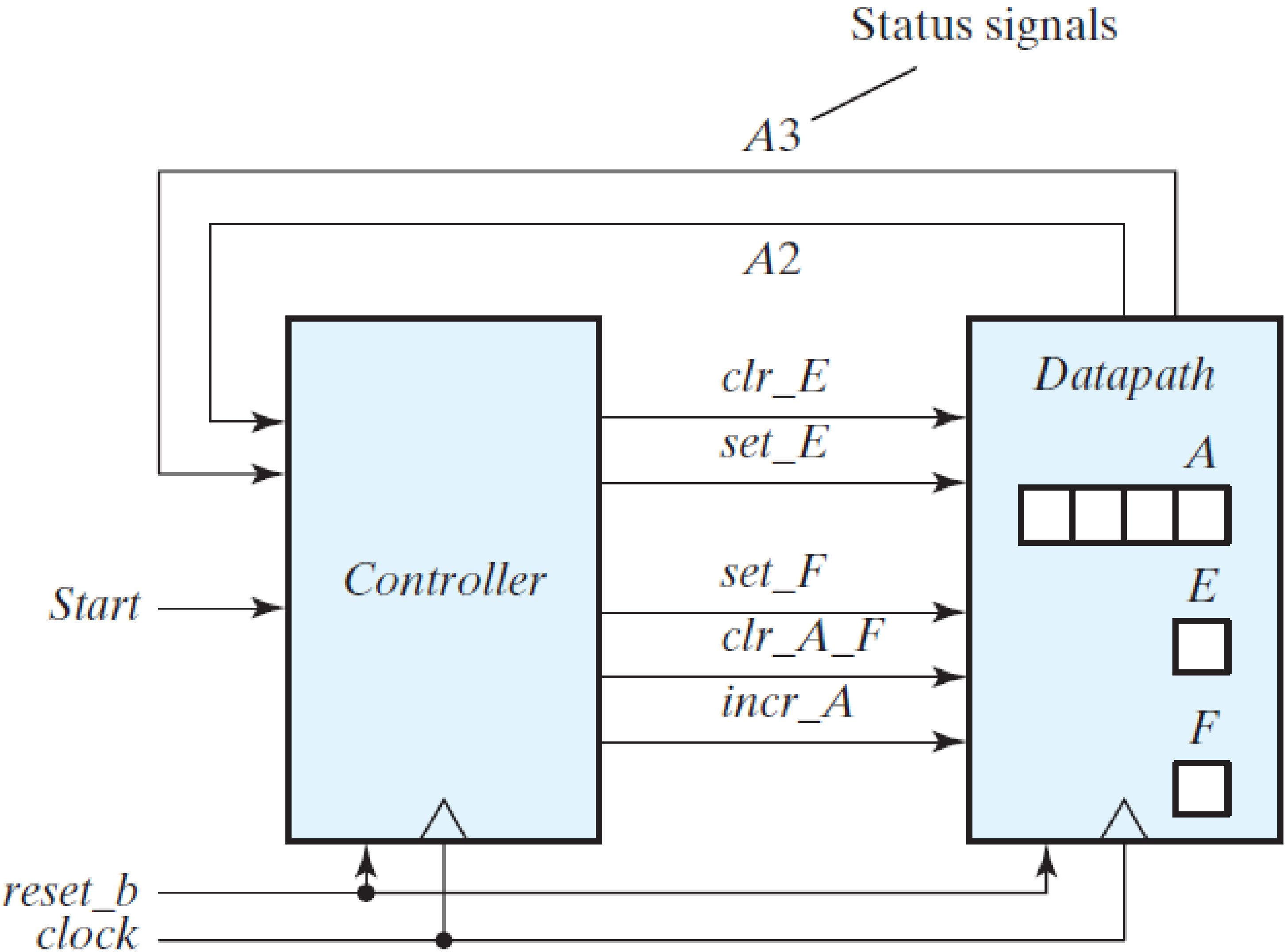
Algorithmic state machine and datapath (ASMD)

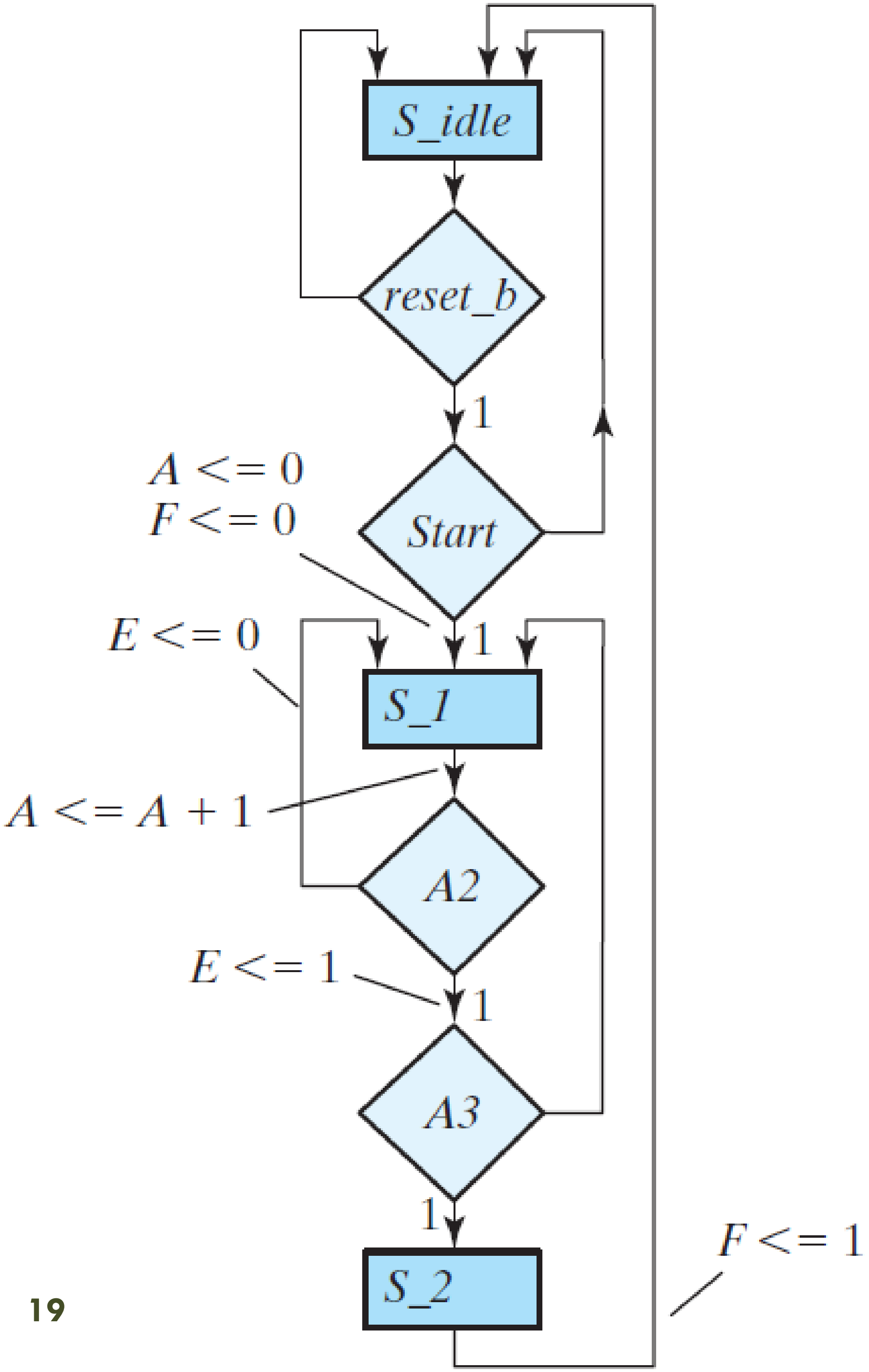
- To clarify the information displayed by ASM charts
- Differences with ASM
  - No register operation in state boxes
  - Edges annotated with register operations
  - Conditional boxes identify signals controlling register operations

Example

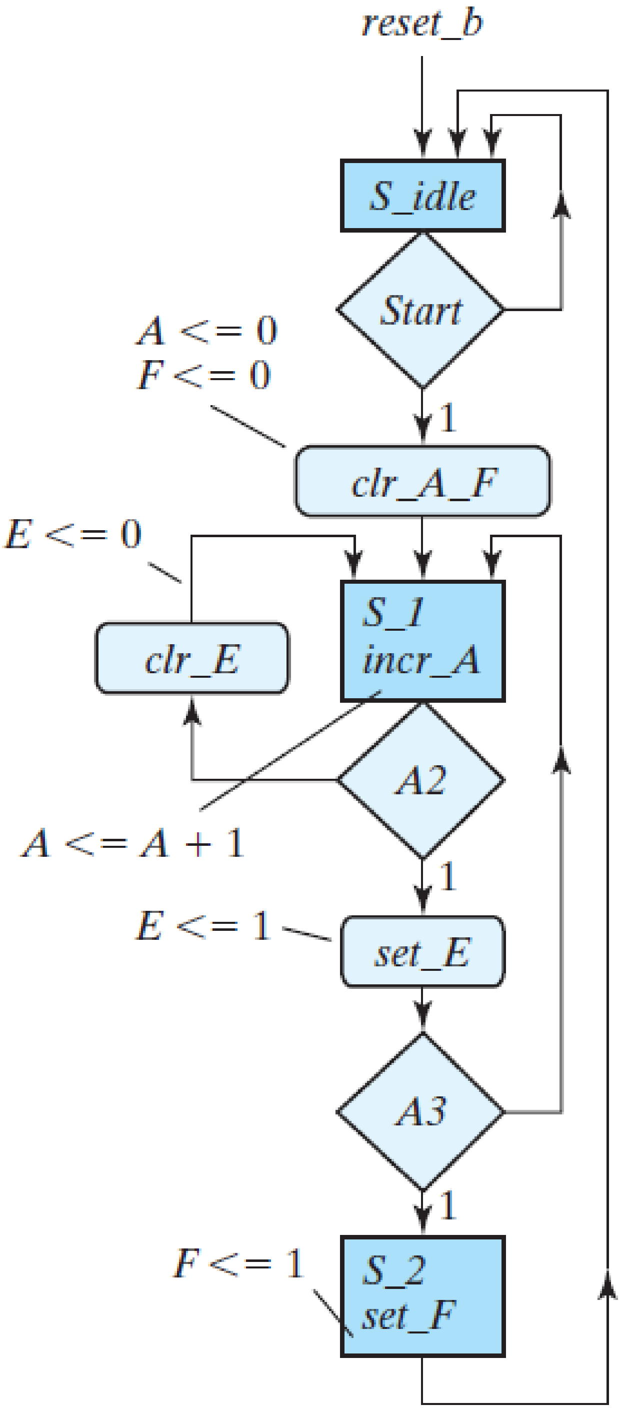
- Specifications
  - Datapath
    - Two JKFFs, i.e., E and F
    - One four-bit binary counter, i.e., A[3:0]
  - *Start*
    - Initiation by clearing A and F
  - At each subsequent clock A incremented
    - If $A_2 = 0$, E cleared, count continues
    - If $A_2 = 1$, E set
      - If $A_3 = 0$, count continues
      - If $A_3 =1$, F set, counting stops

Example (Cont.)

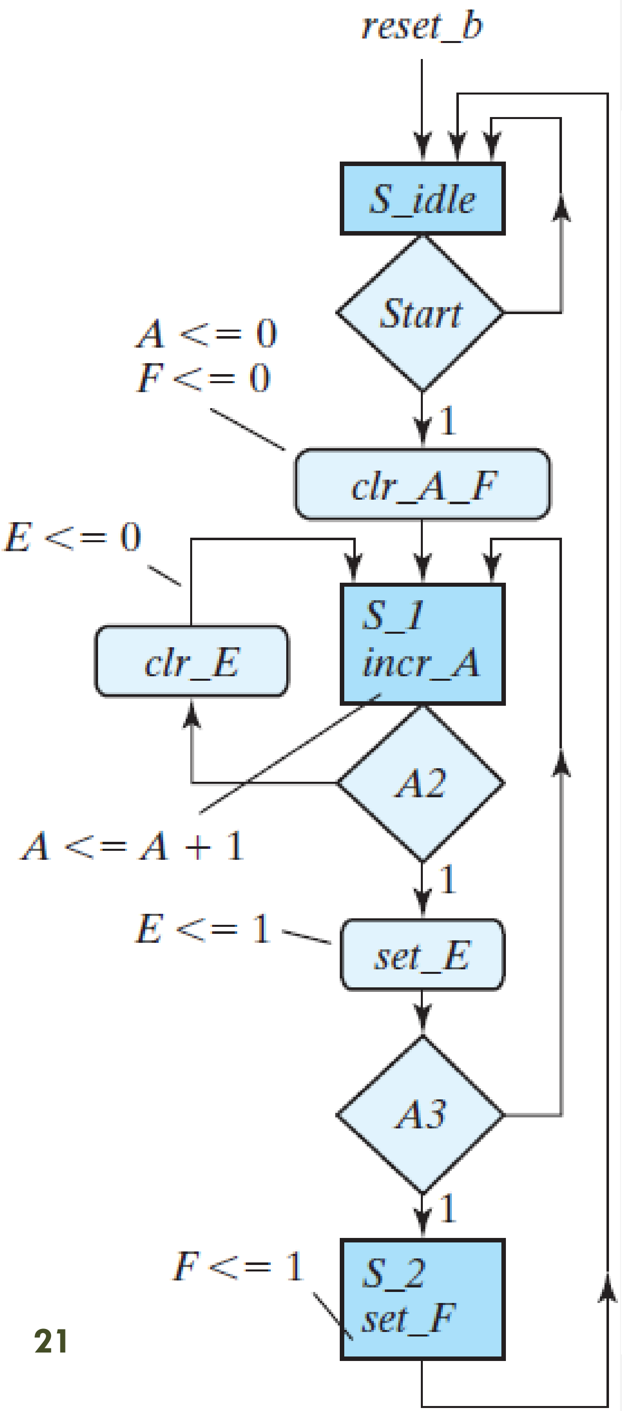




- Synchronous reset
- Register operations annotated on edges leaving
  - A state box
    - Occur unconditionally
    - Controlled by a Moore-type control signal
      - E.g., A ← A + 1
  - A decision box
    - Occur conditionally
    - Controlled by a Mealy-type control signal
      - E.g., E ← 1

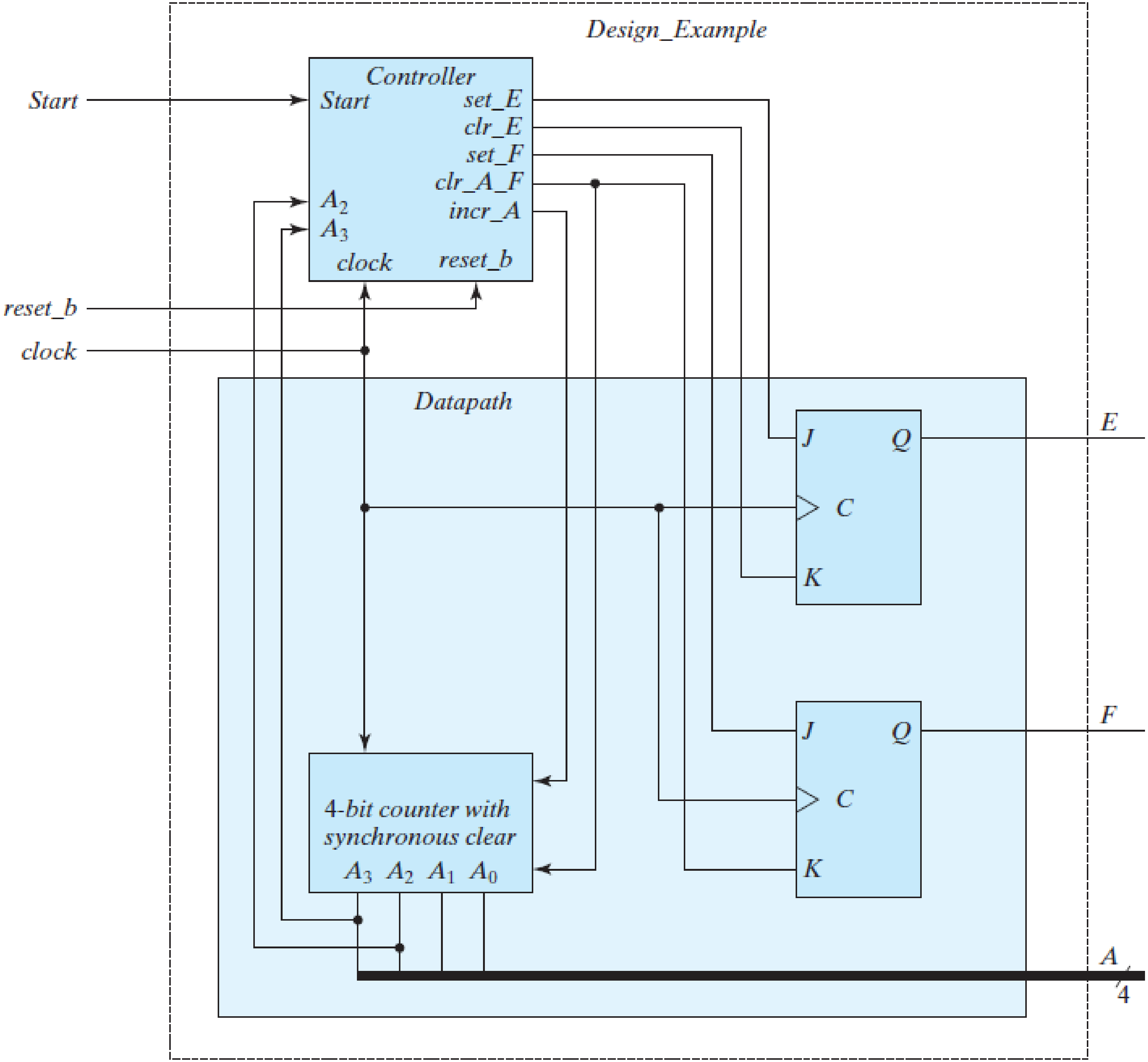


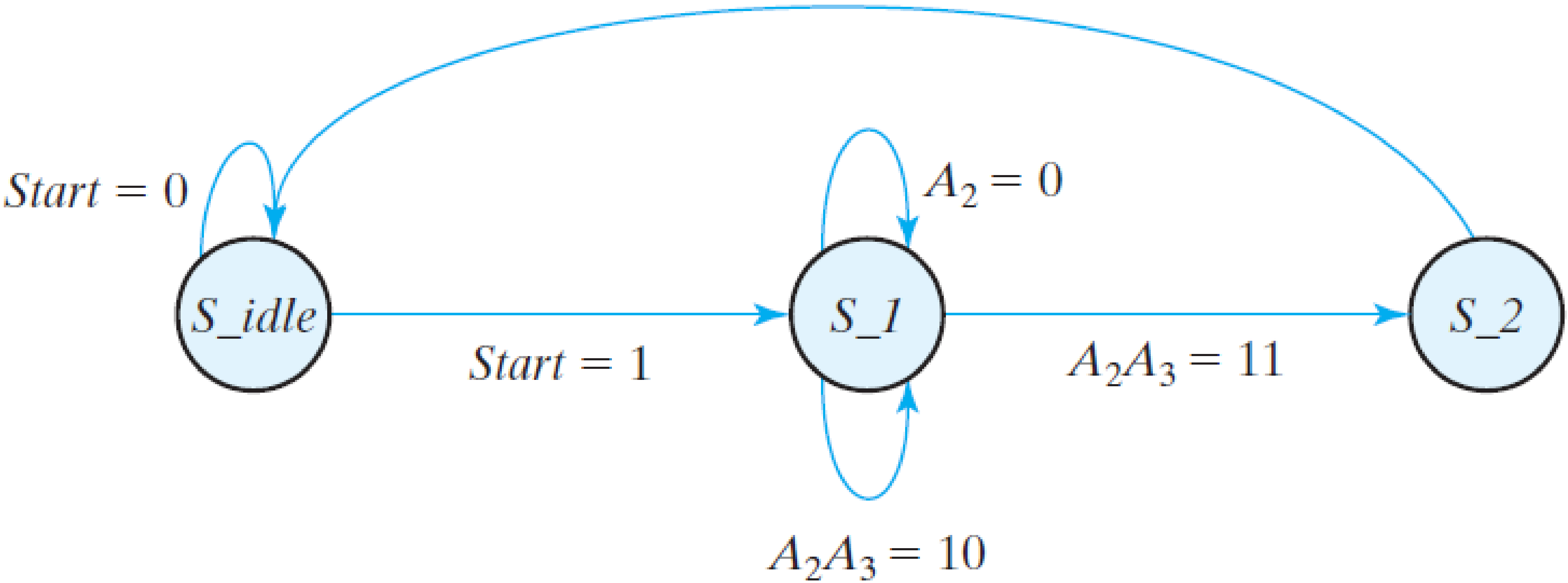
- Asynchronous reset
- Conditional boxes inserted
  - Demonstrate control signals in the chart



Sequence of Operations for Design Example

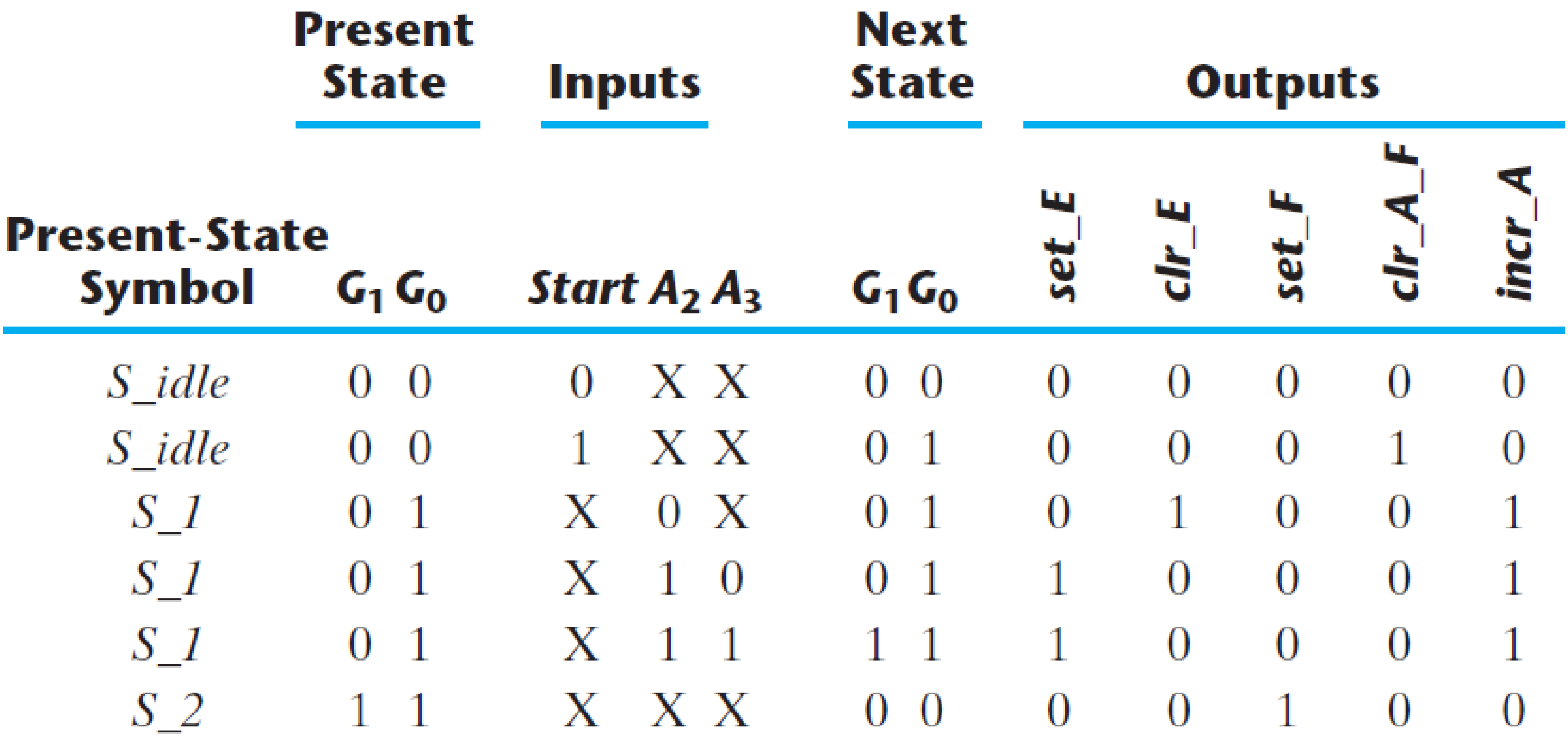
Counter				Flip-Flops			
A_3	A_2	A_1	A_0	E	F	**Conditions**	**State**
0	0	0	0	1	0	$A_2 = 0, A_3 = 0$	S_1
0	0	0	1	0	0		
0	0	1	0	0	0		
0	0	1	1	0	0		
0	1	0	0	0	0	$A_2 = 1, A_3 = 0$	
0	1	0	1	1	0		
0	1	1	0	1	0		
0	1	1	1	1	0		
1	0	0	0	1	0	$A_2 = 0, A_3 = 1$	
1	0	0	1	0	0		
1	0	1	0	0	0		
1	0	1	1	0	0		
1	1	0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


Design_Example
Controller
Start
set_E
clr_E
set_F
clr_A_F
A_2
A_3
incr_A
clock
reset_b
Start
reset_b
clock
Datapath
J
Q
C
K
E
J
Q
C
K
F
4-bit counter with synchronous clear
A_3 A_2 A_1 A_0
A
4

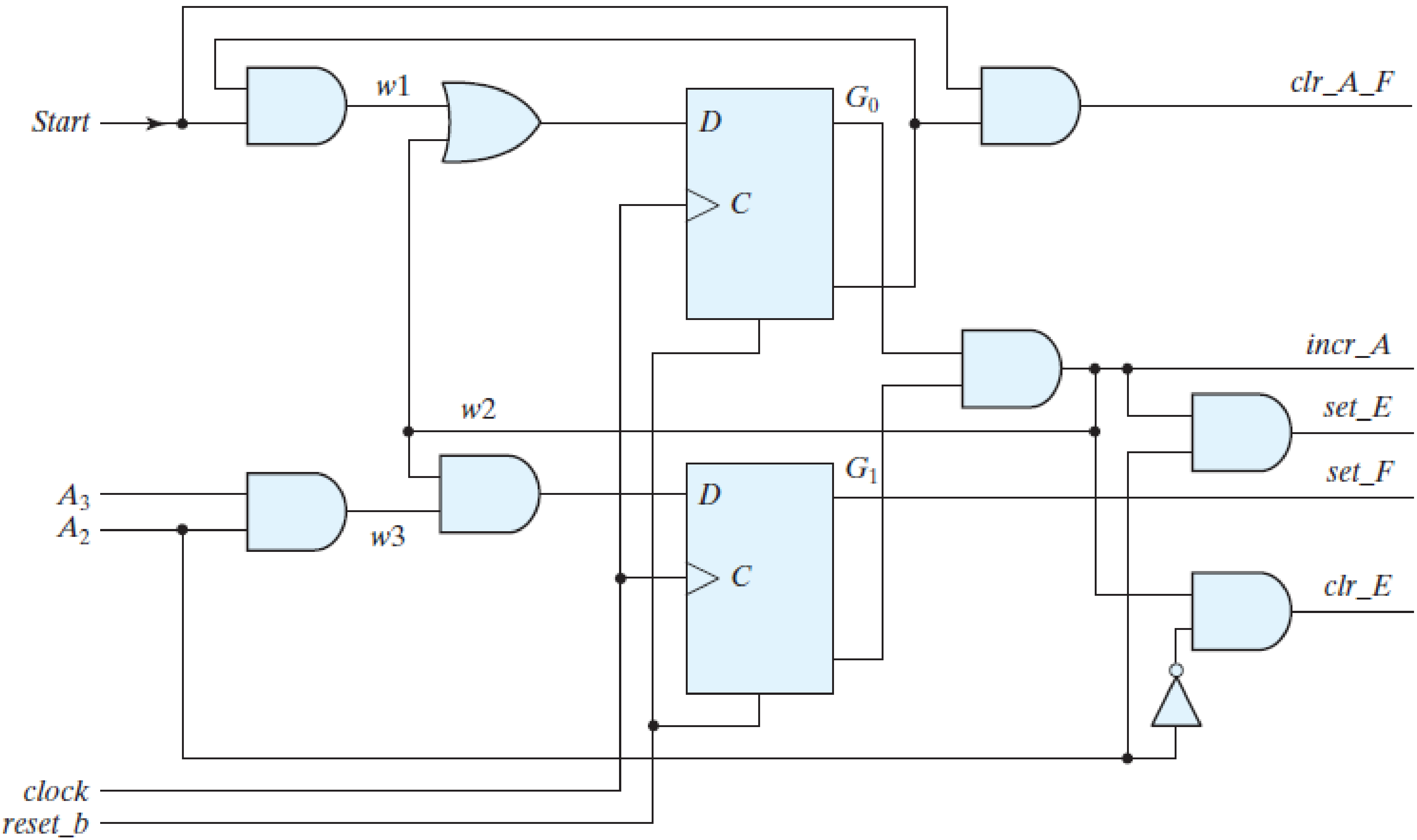


(a)

$S\_idle \longrightarrow S\_1,\ clr\_A\_F$:		$A \longleftarrow 0,\ F \longleftarrow 0$
$S\_1 \longrightarrow S\_1,\ incr\_A$:		$A \longleftarrow A + 1$
	if $(A_2 = 1)$ then set_E:	$E \longleftarrow 1$
	if $(A_2 = 0)$ then clr_E:	$E \longleftarrow 0$
$S\_2 \longrightarrow S\_idle,\ set\_F$:		$F \longleftarrow 1$



	Present State		Inputs			Next State		Outputs				
Present-State Symbol	G_1	G_0	*Start*	A_2	A_3	G_1	G_0	*set_E*	*clr_E*	*set_F*	*clr_A_F*	*incr_A*
S_idle	0	0	0	X	X	0	0	0	0	0	0	0
S_idle	0	0	1	X	X	0	1	0	0	0	1	0
S_1	0	1	X	0	X	0	1	0	1	0	0	1
S_1	0	1	X	1	0	0	1	1	0	0	0	1
S_1	0	1	X	1	1	1	1	1	0	0	0	1
S_2	1	1	X	X	X	0	0	0	0	1	0	0


Start
w1
G_0
D
C
clr_A_F
incr_A
w2
set_E
set_F
G_1
A_3
A_2
w3
D
C
clr_E
clock
reset_b