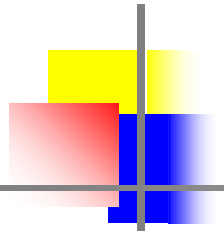
Ch. 08. Design at the Register Transfer Level

2019. 6.10.

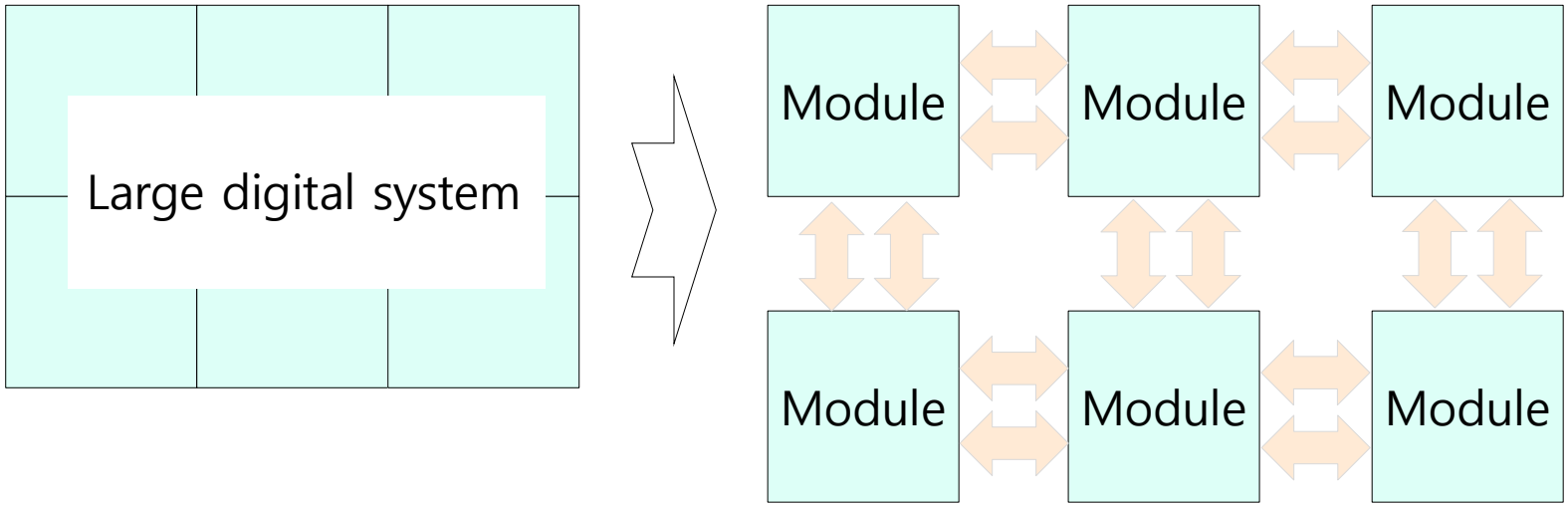
K-S. Sohn

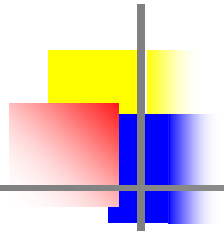


Contents

Purposes of RTL Design

- Describing and designing large, complex digital systems





Module of a Digital System

- Module = Registers + Operations
- 3 components of the specification of a module
  - registers
  - operations on registers (data)
  - control supervising the sequence of operations

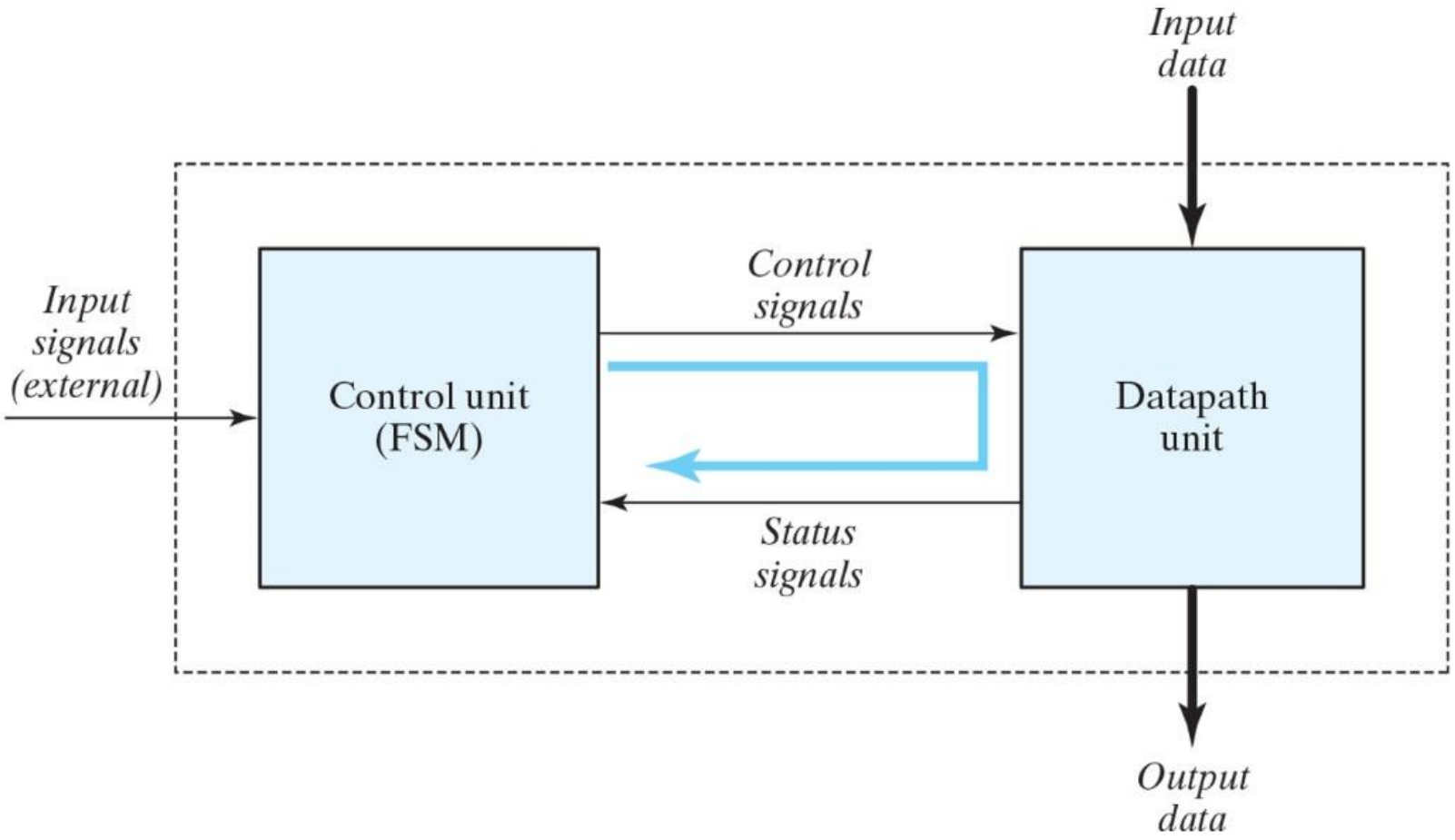
Operations

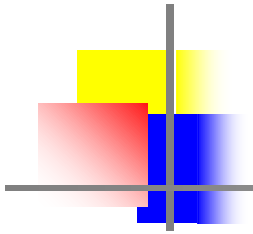
- Type of Operations
  - Transfer
  - Logical
  - Arithmetic
  - Shift
- Operators
  - Verilog operators

Operator Type	Symbol	Operation Performed	Priority Group
Arithmetic	+	addition	1 (unary), 4 (binary)
	–	subtraction	1 (unary), 4 (binary)
	*	multiplication	3
	/	division	3
	**	exponentiation	2
	%	modulus	2
Bitwise or Reduction	~	negation (complement)	1
	&, ~&	AND, NAND (reduction)	1
	|,~|	OR, NOR (reduction)	1
	^, ~^	XOR, XNOR (reduction)	1
	^, ~^, ^~	XOR, XNOR (binary)	9
Logical	!	negation	1
	&&	AND (binary)	11
	||	OR (binary)	12
	&	AND (binary)	8
	|	OR (binary)	10
Shift	>>	logical right shift	5
	<<	logical left shift	5
	>>>	arithmetic right shift	5
	<<<	arithmetic left shift	5
Relational	>	greater than	6
	<	less than	6
	<=	less than or equal	6
	>=	greater than or equal	6
Equality	==	equality	7
	!=	inequality	7
	===	case equality	7
	!==	case inequality	7
Conditional	? :	ternary selection	13
Concatenation	{ } {{}}	joins operands	14

Algorithmic State Machines (ASMs)

- Digital system = Controller + Datapath



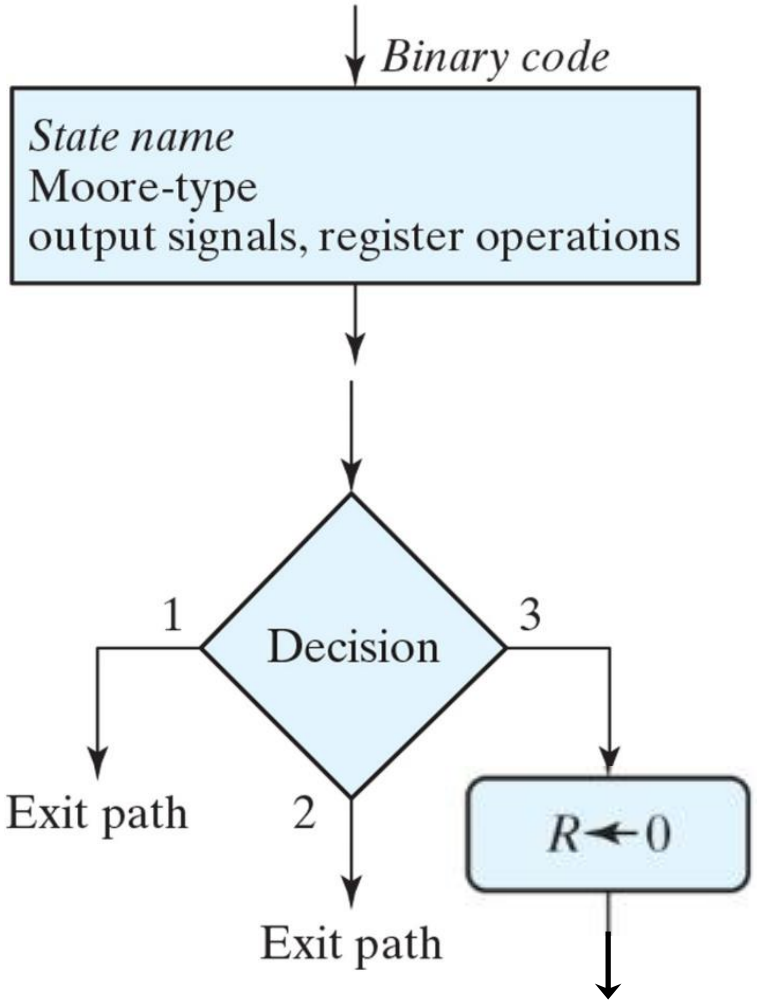


Algorithmic State Machines (ASMs)

- Controller
  - Controls the operations in the datapath unit
  - Described by ASM chart
- Datapath
  - Executes operations on data
  - Described by ASMD chart

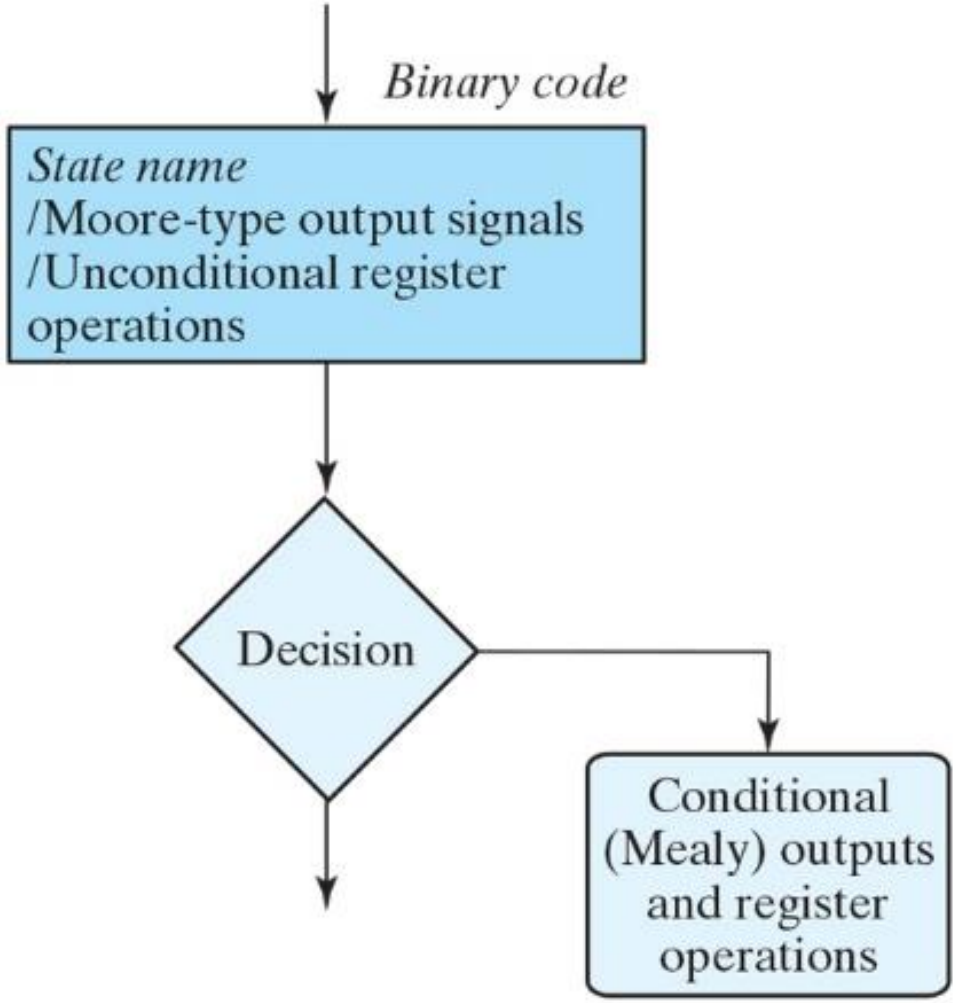
ASM Chart Components

- State box
- Decision box
- Conditional box



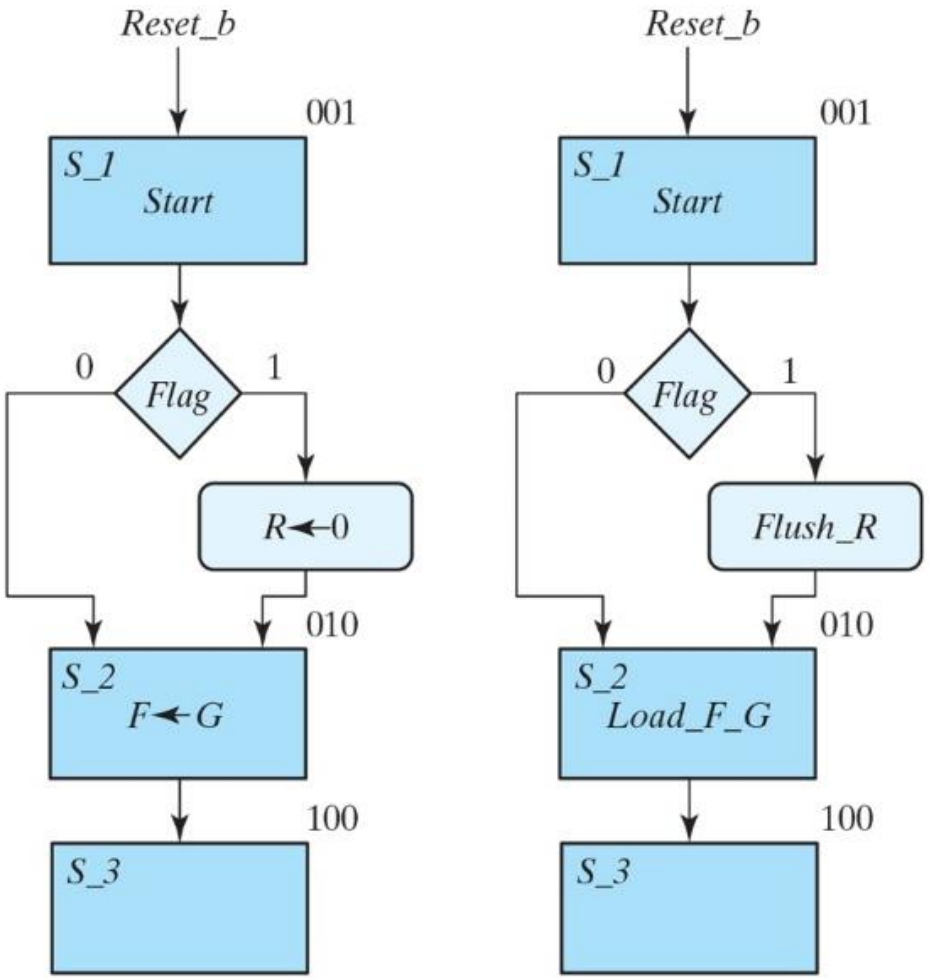
ASM Chart Form

- State box
  - state name, state code
  - Moore type output signals
  - unconditional reg. op.
- Decision box
- Conditional box
  - Mealy output signals
  - conditional reg. op.



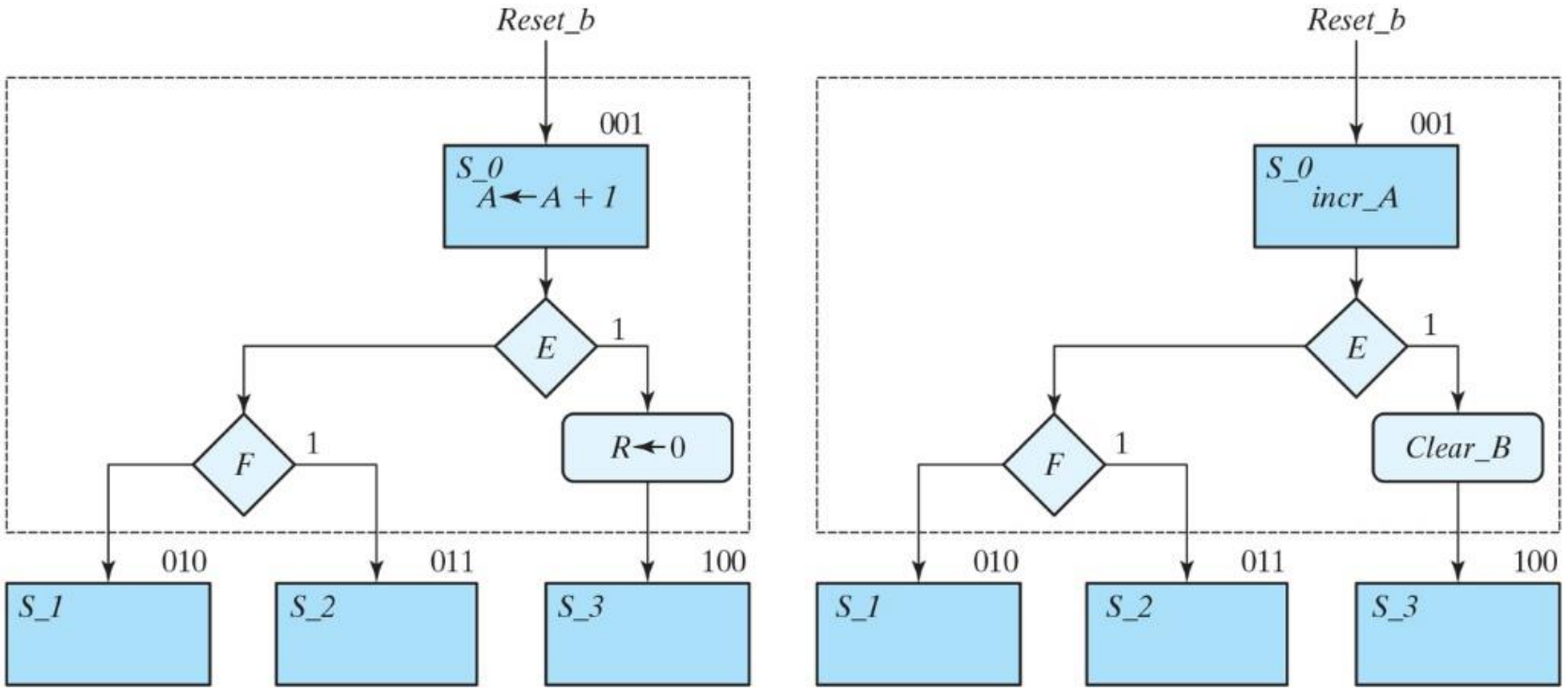
ASM Chart (Ex.)

- ASM with reg. op. and control signals



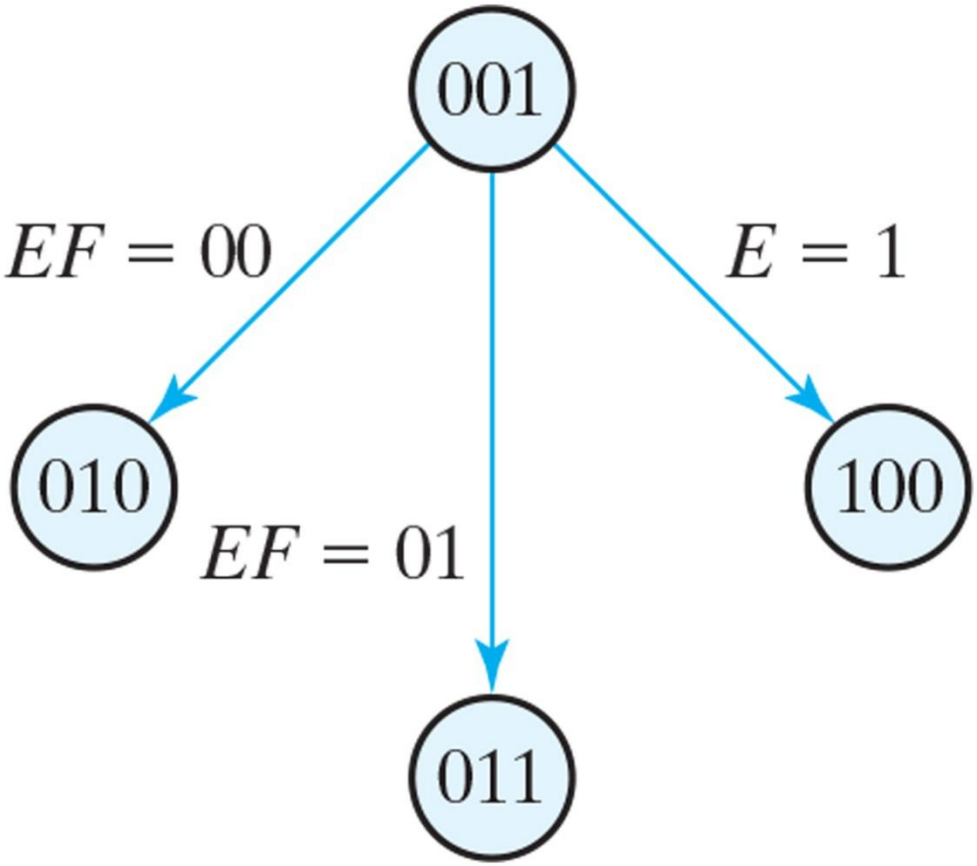
ASM Block

- ASM block conponents
  - one state box
  - optionally, decision box and conditional box



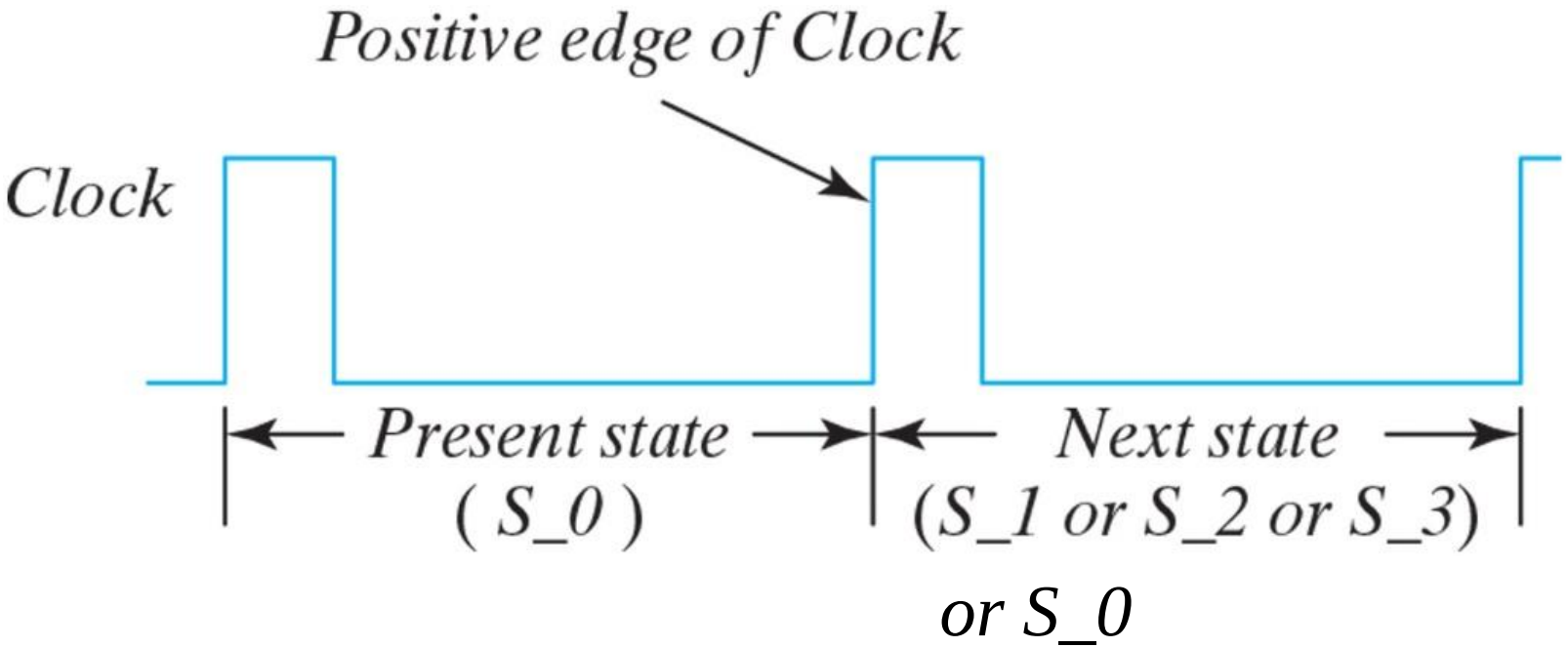
State Diagram vs. ASM Chart

- State diagram equivalent to the ASM chart



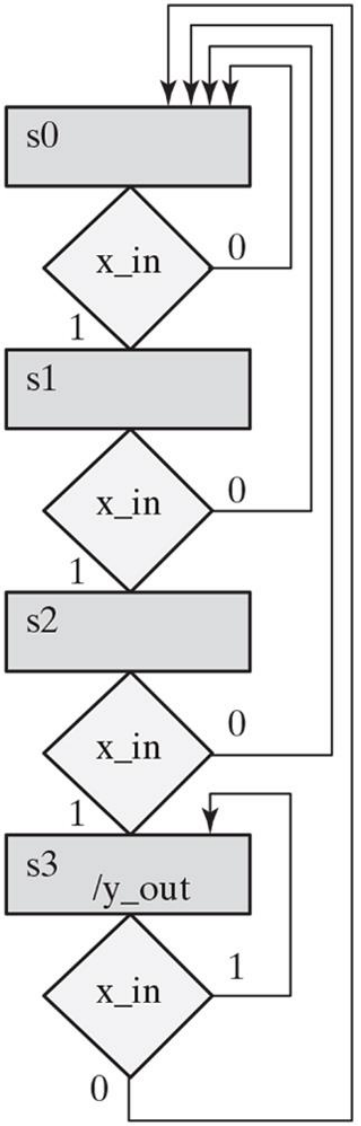
Timing Considerations

- State transition timing



ASM Chart (Practice 8.10)

- ASM chart for 3 consecutive 1 detector

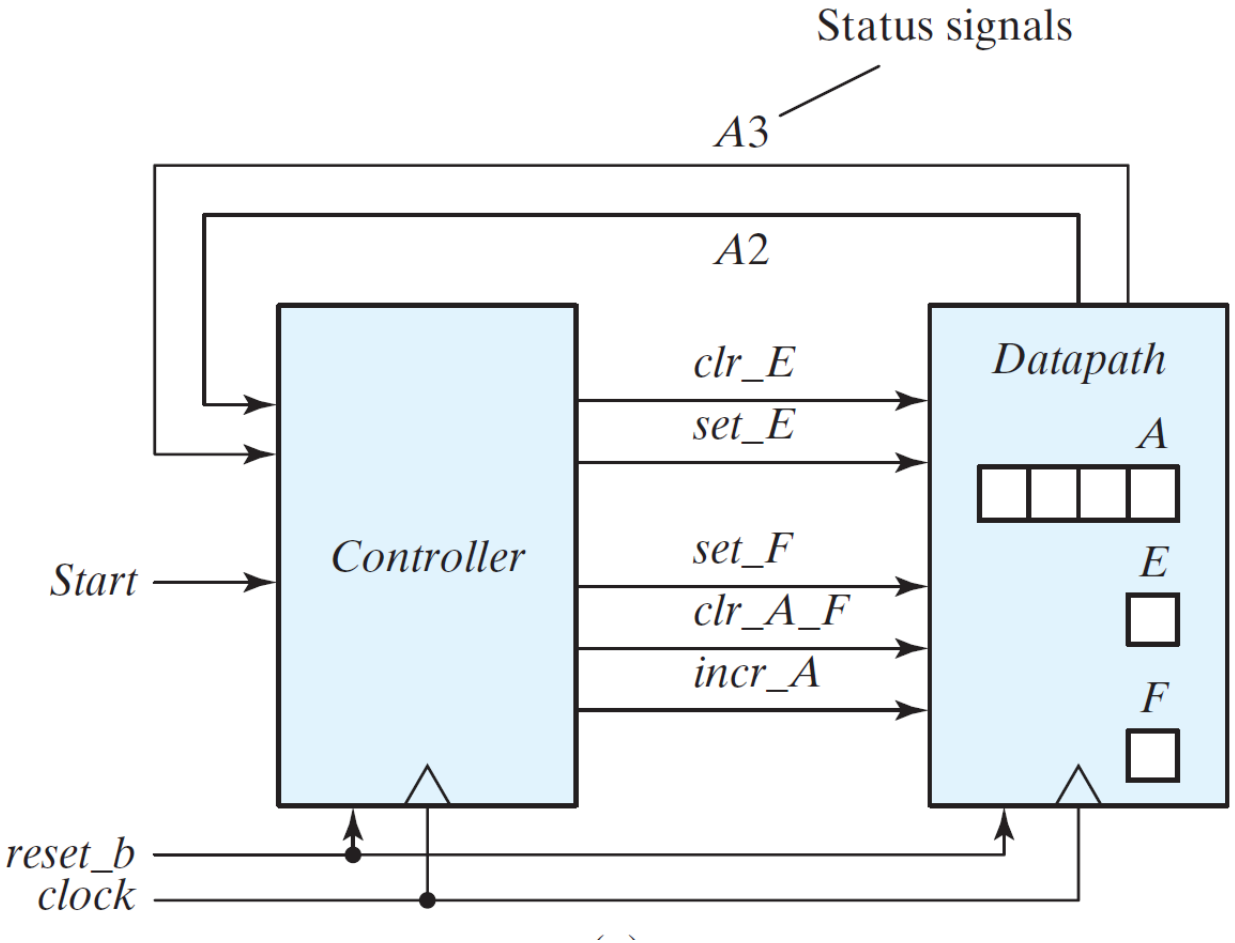


ASMD(ASM and Datapath)

- Example: Design requirements
  - If A2 = 0 ,
    - E is cleared to 0 and the count continues.
  - If A2 = 1 ,
    - E is set to 1;
    - then, if A3 = 0 ,
      - the count continues,
    - but if A3 = 1 ,
      - F is set to 1 on the next clock pulse and the system stops counting.
  - Then, if Start = 0,
    - the system remains in the initial state,
  - but if Start = 1,
    - the operation cycle repeats.

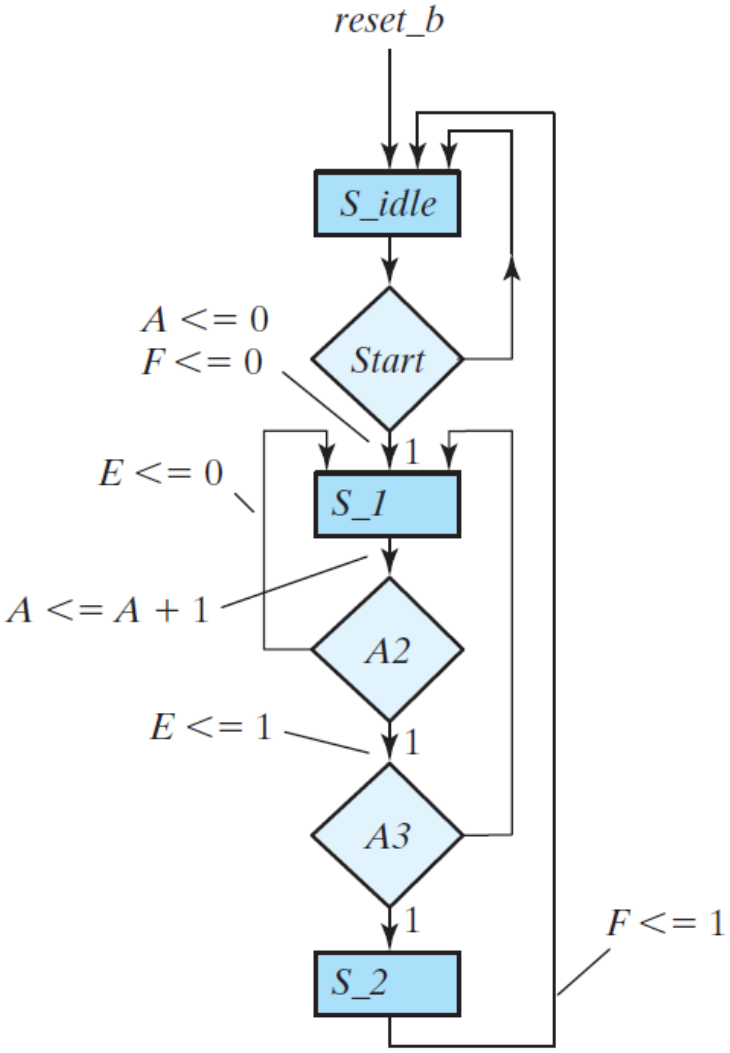
The Example

- Block diagram



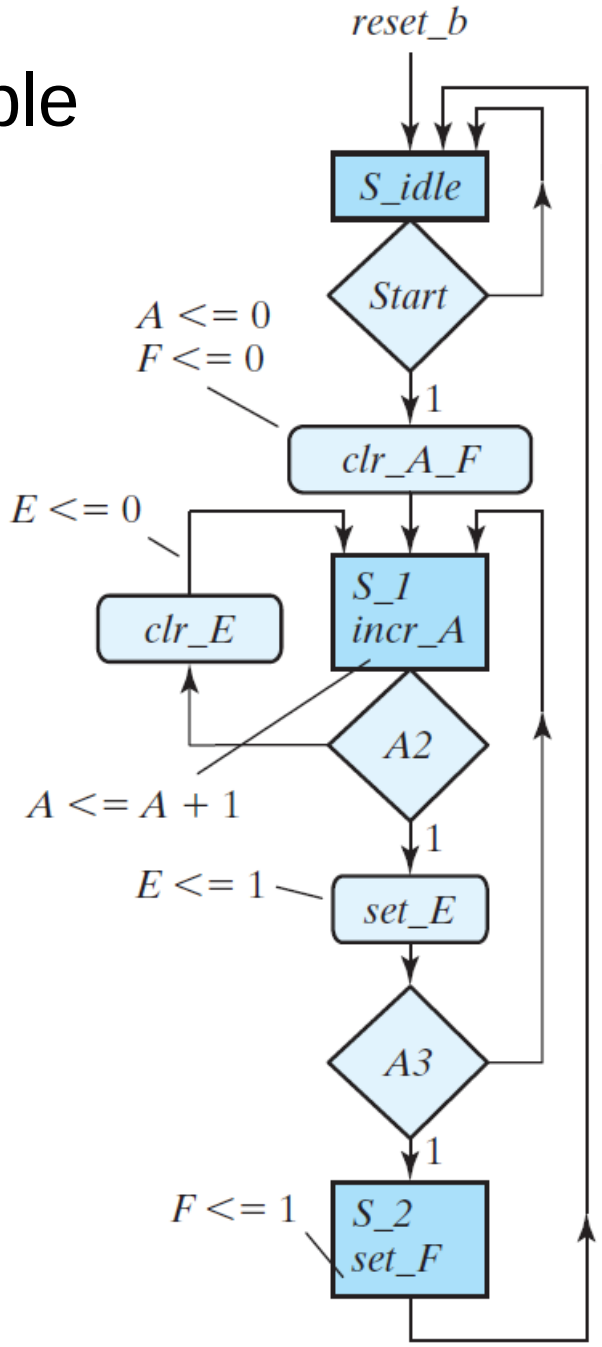
ASM of the Example

- ASM chart



ASMD of the Example

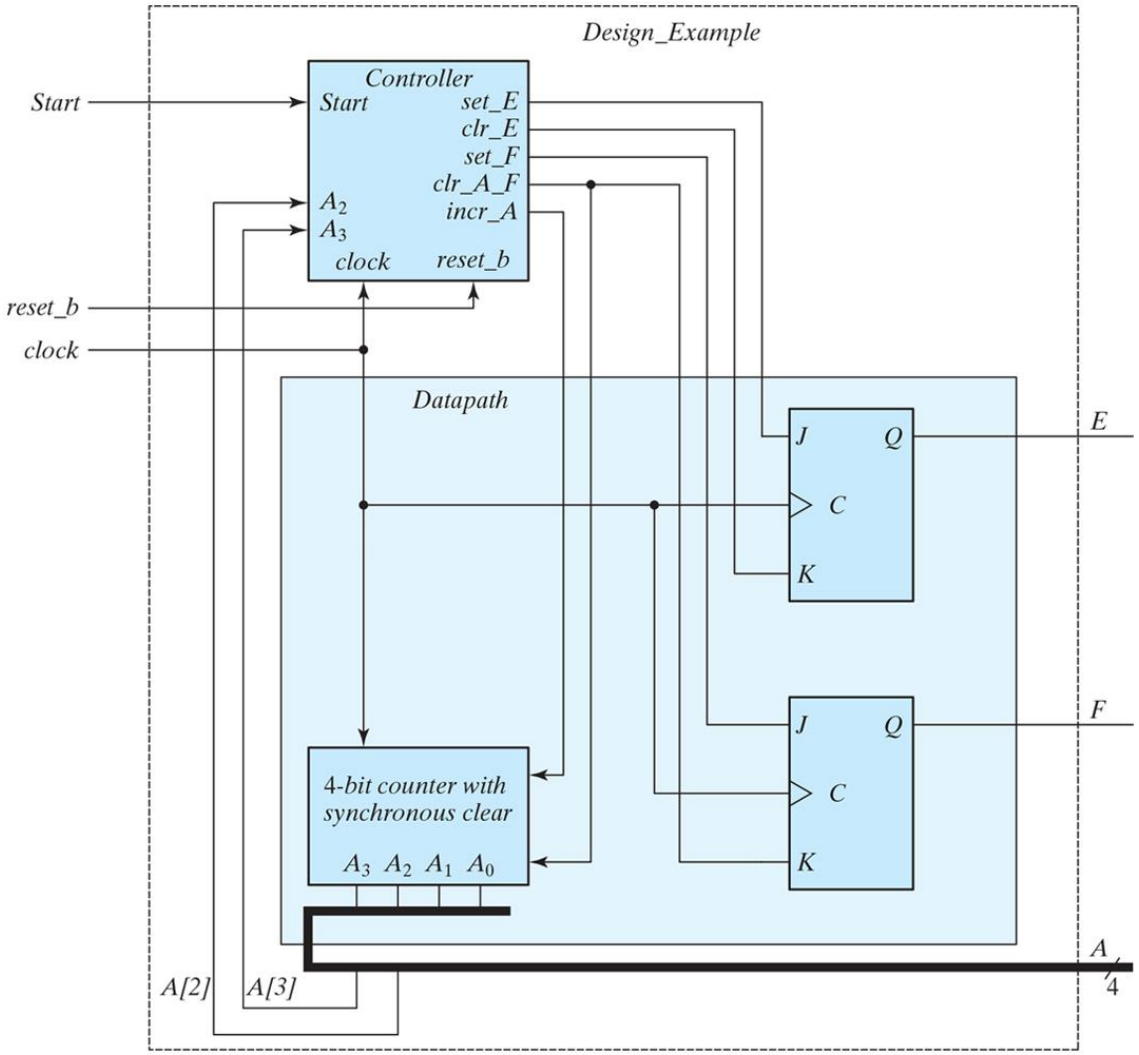
- ASMD chart



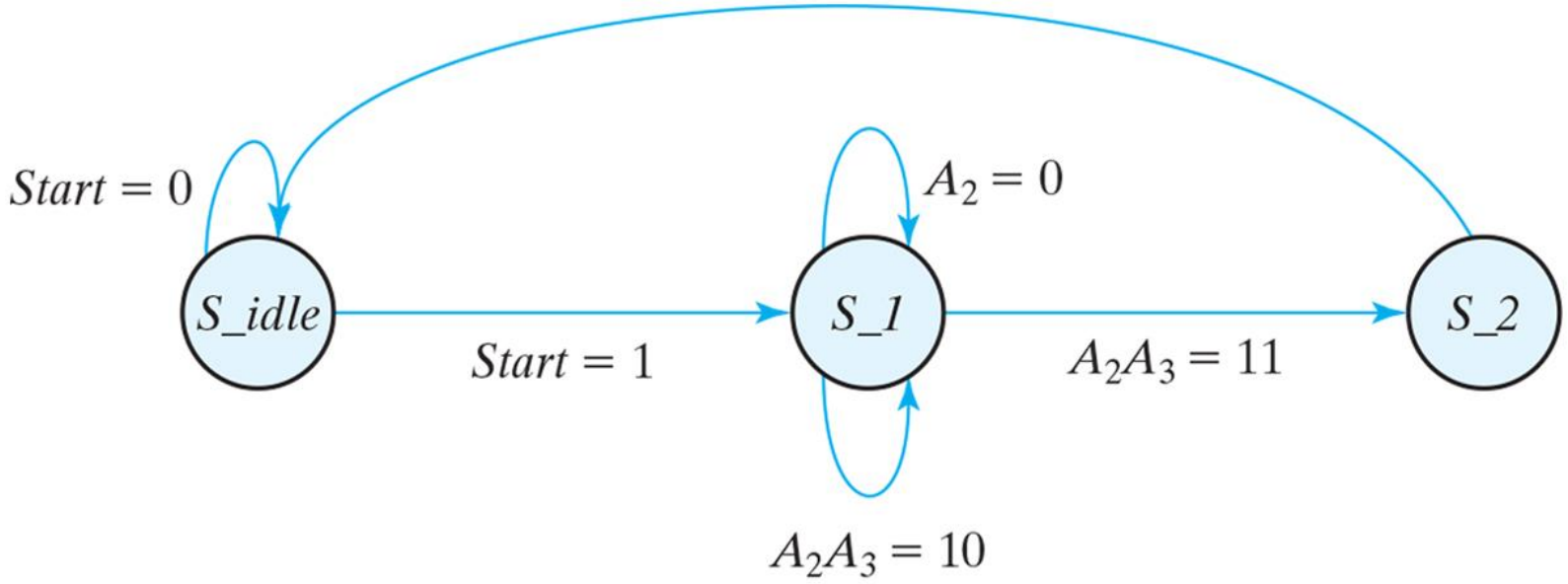
Sequence of Operations of the Example

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

Block Diagram of the Example

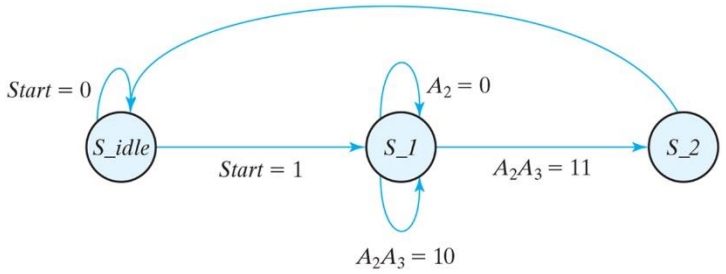


State Diagram from ASMD



RTL Description from ASMD

S_idle	⟶ *S_1, clr_A_F:*	$A \leftarrow 0, F \leftarrow 0$
S_1	⟶ *S_1, incr_A:*	$A \leftarrow A + 1$
	if $(A_2 = 1)$ then *set_E:*	$E \leftarrow 1$
	if $(A_2 = 0)$ then *clr_E:*	$E \leftarrow 0$
S_1	⟶ *S_2, incr_A:*	$F \leftarrow 1$
S_2	⟶ *S_idle:*	$A \leftarrow A + 1$

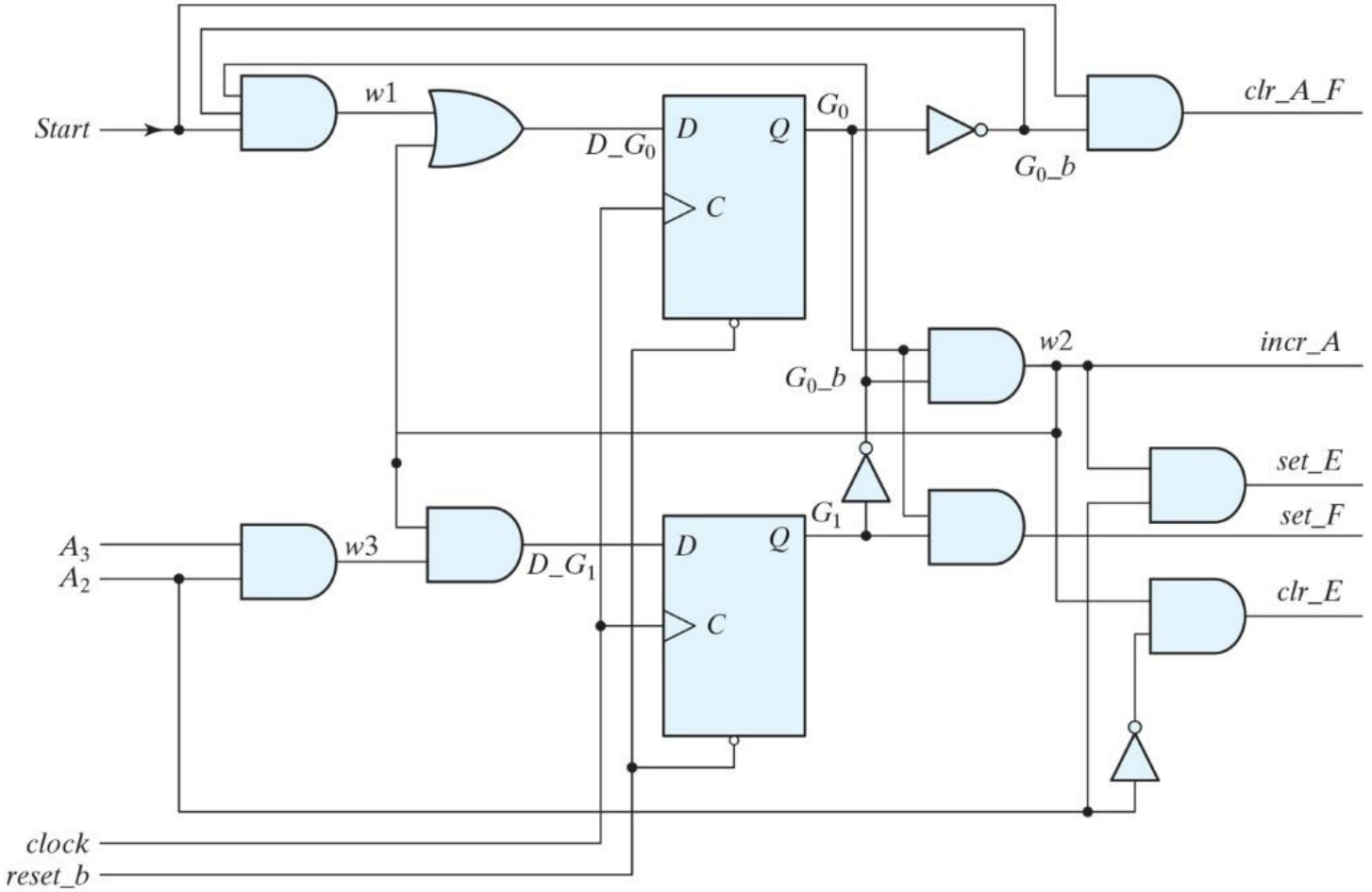


State Table from ASMD

	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

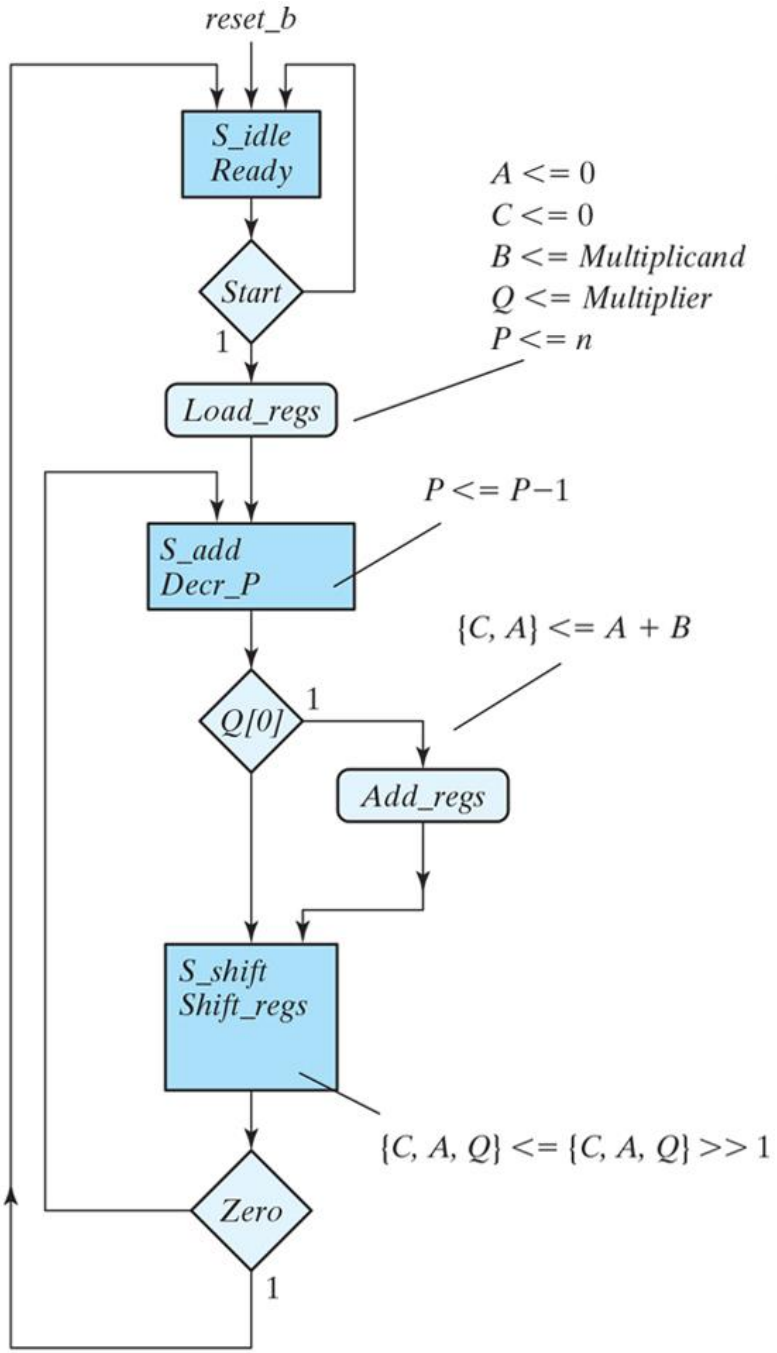
Logic Diagram of the Example

- Controller logic diagram



Binary Multiplier

- ASMD chart

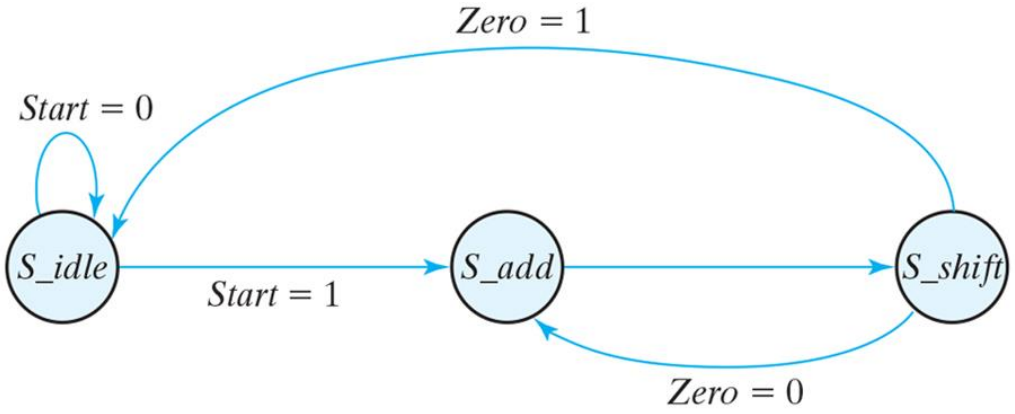


Numerical Exampl of Binary Multiplier

Multiplicand $B = 10111_2 = 17_H = 23_{10}$ **Multiplier $Q = 10011_2 = 13_H = 19_{10}$**

	C	A	Q	P
Multiplier in Q	0	00000	10011	101
$Q_0 = 1$; add B		10111		
First partial product	0	10111		100
Shift right CAQ	0	01011	11001	
$Q_0 = 1$; add B		10111		
Second partial product	1	00010		011
Shift right CAQ	0	10001	01100	
$Q_0 = 0$; shift right CAQ	0	01000	10110	010
$Q_0 = 0$; shift right CAQ	0	00100	01011	001
$Q_0 = 1$; add B		10111		
Fifth partial product	0	11011		
Shift right CAQ	0	01101	10101	000
Final product in $AQ = 0110110101_2 = 1b5_H$				

State Diagram and RTL Description



(a)

State Transition		Register Operations
From	To	
S_idle		*Initial state*
S_idle	*S_add*	$A <= 0, C <= 0, P <= dp\_width$ $A <= Multiplicand, Q <= Multiplier$
S_add	*S_shift*	$P <= P - 1$ $if\ (Q[0])\ then\ (A <= A + B, C <= C_{out})$
S_shift		*shift right {CAQ}*, $C <= 0$

(b)

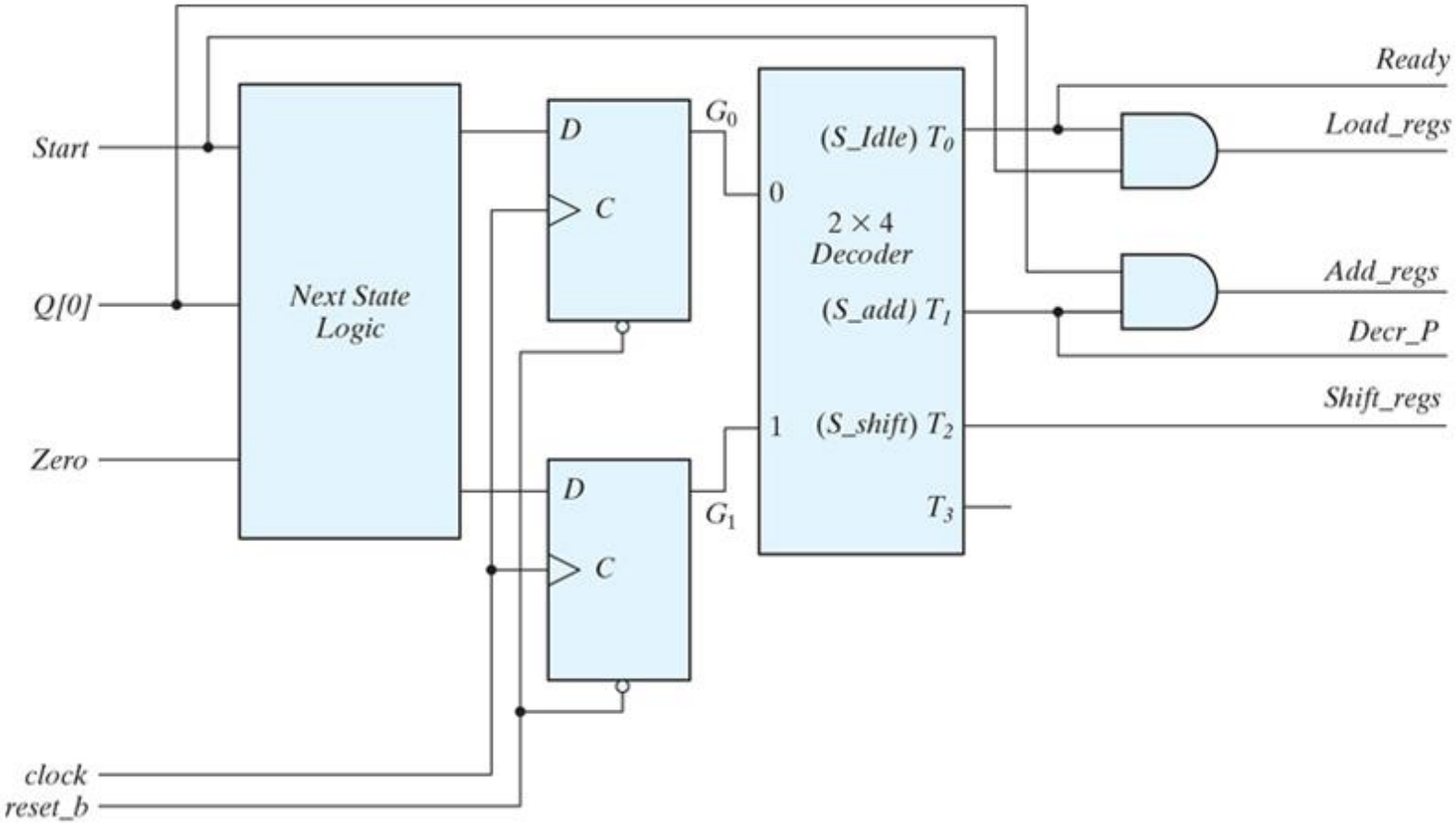
State Assignment for Control

State	Binary	Gray Code	One-Hot
S_idle	00	00	001
S_add	01	01	010
S_shift	10	11	100

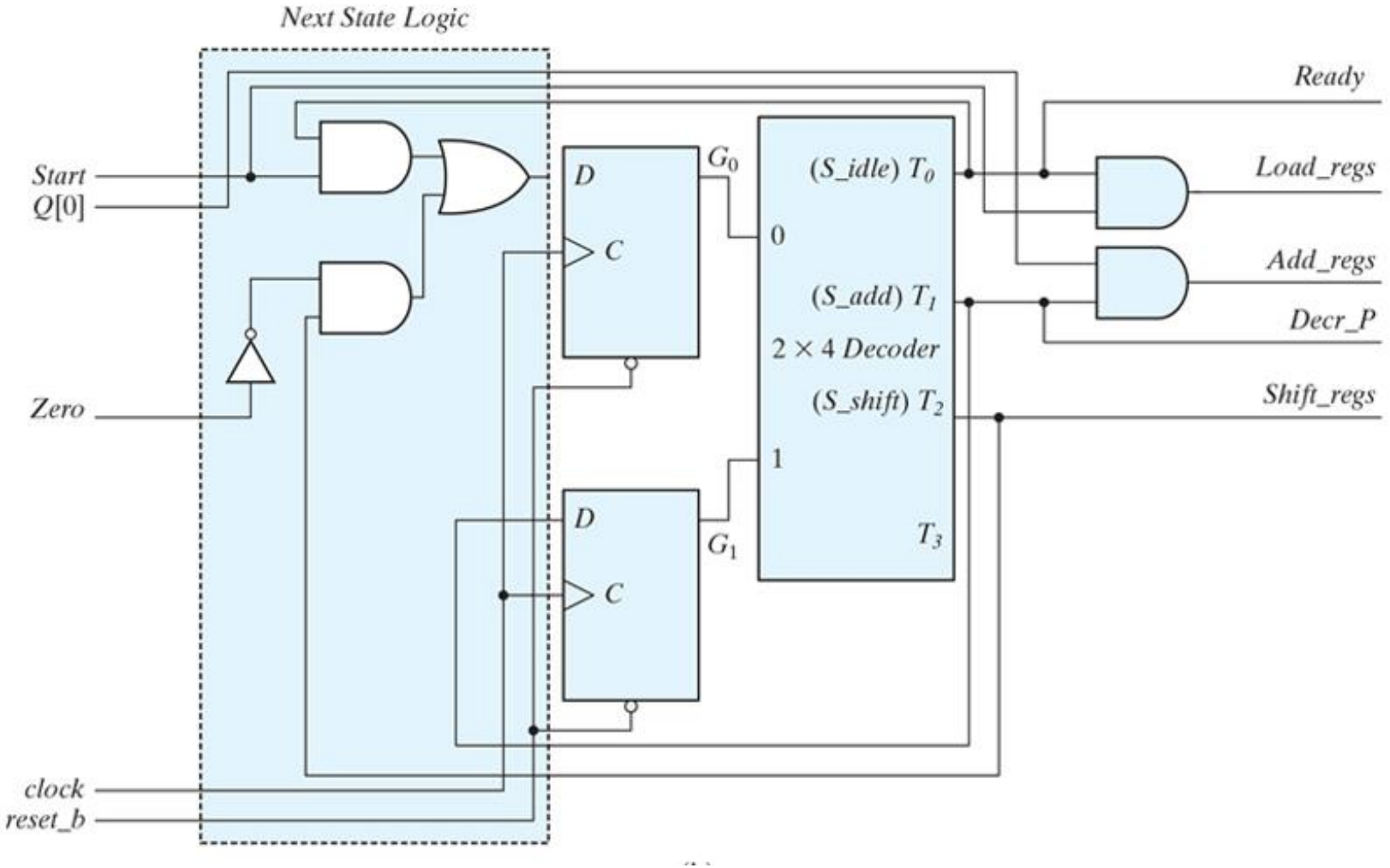
State Table for Control Circuit

	Present State		Inputs			Next State		Outputs				
Present-State Symbol	G_1	G_0	*Start*	*Q[0]*	*Zero*	G_1	G_0	*Ready*	*Load_regs*	*Decr_P*	*Add_regs*	*Shift_regs*
S_idle	0	0	0	X	X	0	0	1	0	0	0	0
S_idle	0	0	1	X	X	0	1	1	1	0	0	0
S_add	0	1	X	0	X	1	0	0	0	1	0	0
S_add	0	1	X	1	X	1	0	0	0	1	1	0
S_shift	1	0	X	X	0	0	1	0	0	0	0	1
S_shift	1	0	X	X	1	0	0	0	0	0	0	1

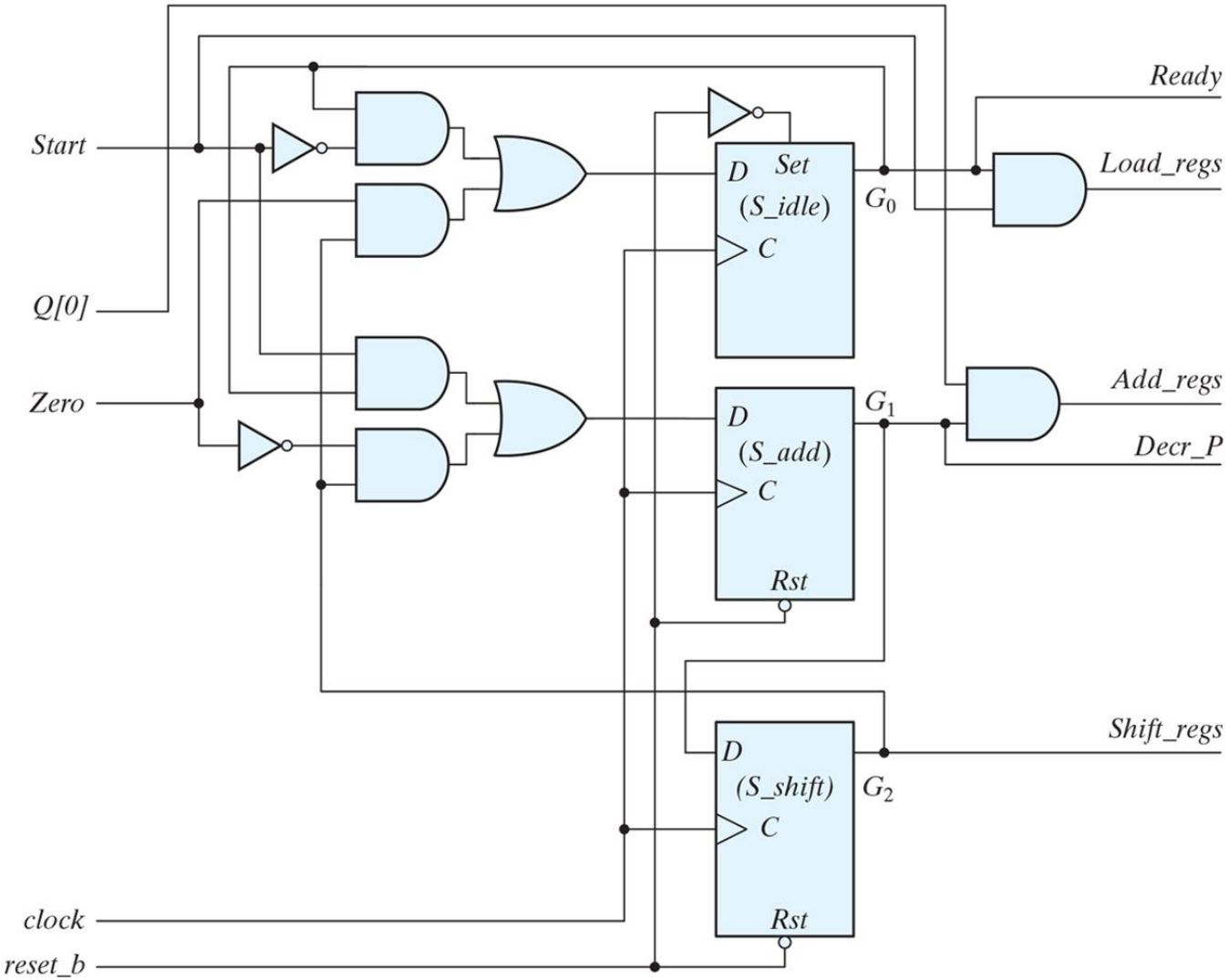
Block Diagram

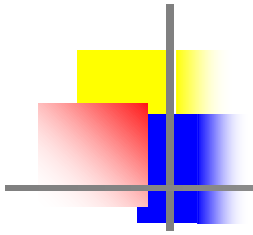


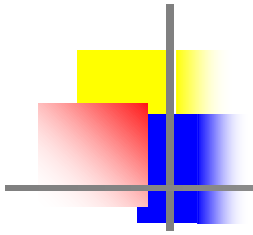
Logic Diagram

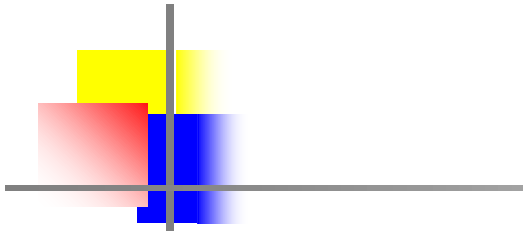


Logic Diagram for One-hot State Controller









Discussion~~~