SEE 3243/4243
ASM System Case Studies

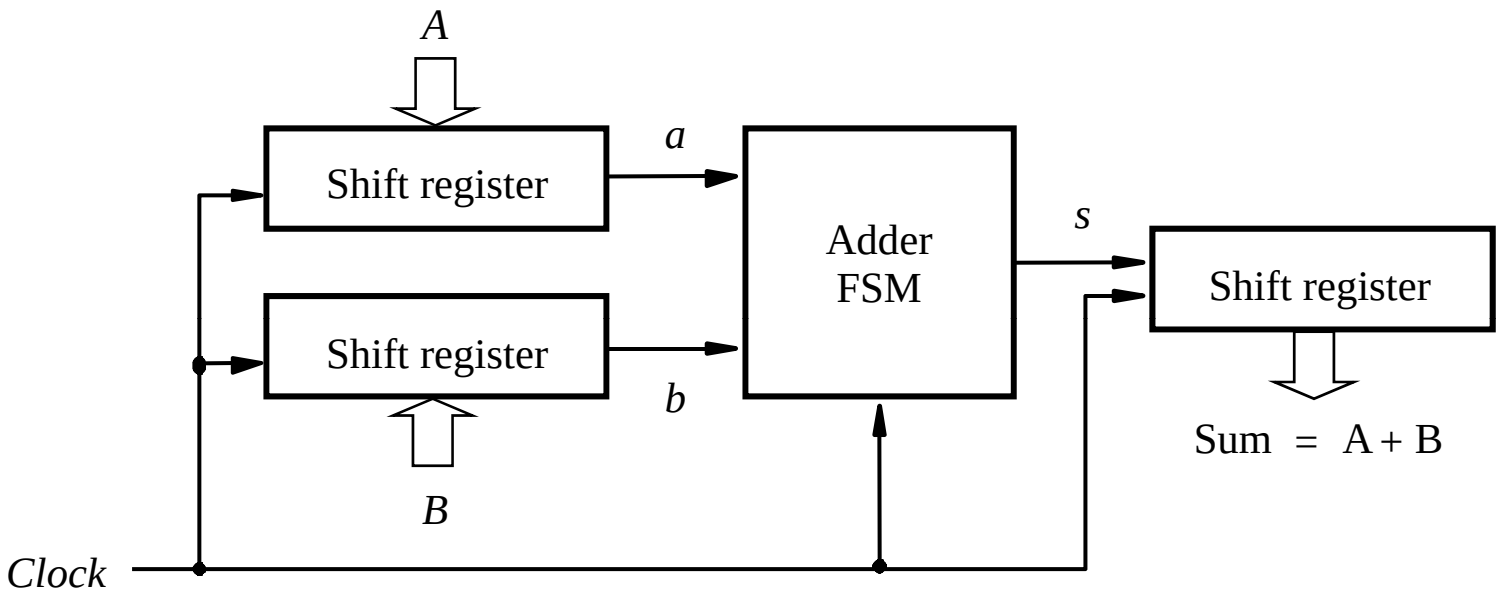
Week 13-14

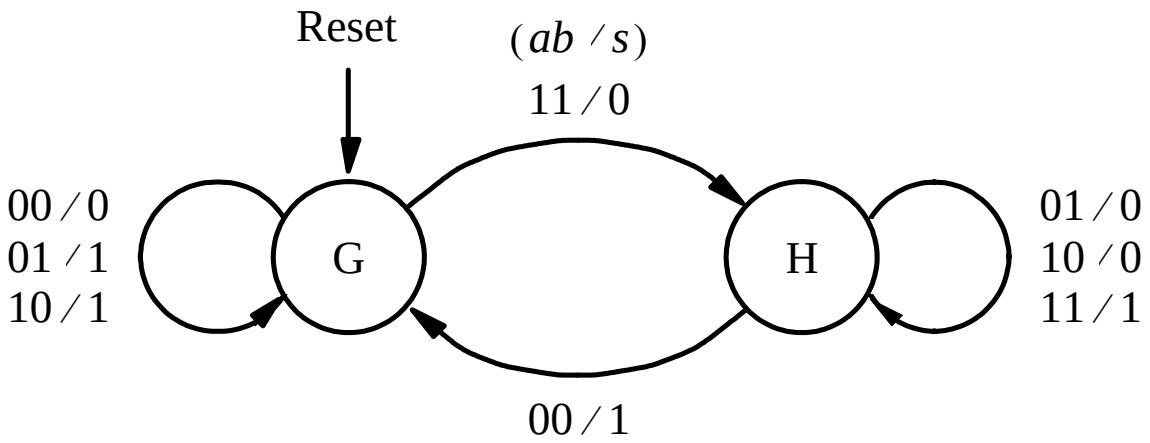
- *Serial Adder*
- *First-1 Finder*
- *Serial Multiplier*

Digital System Design

- Digital system consists of two parts:
  - *Datapath circuit* – used to store, manipulate, and transfer data.
  - *Control circuit* – controls the operation of the datapath. Usually it's built using an ASM.

Block diagram of a serial adder





G: carry-in = 0

H: carry-in = 1

Figure 8.40 State diagram for the serial adder

Present state	Next state				Output *s*			
	ab =00	01	10	11	00	01	10	11
G	G	G	G	H	0	1	1	0
H	G	H	H	H	1	0	0	1

Present state	Next state				Output			
	ab =00	01	10	11	00	01	10	11
y	*y*+				*s*			
0	0	0	0	1	0	1	1	0
1	0	1	1	1	1	0	0	1

Figure 8.41 State table for the serial adder

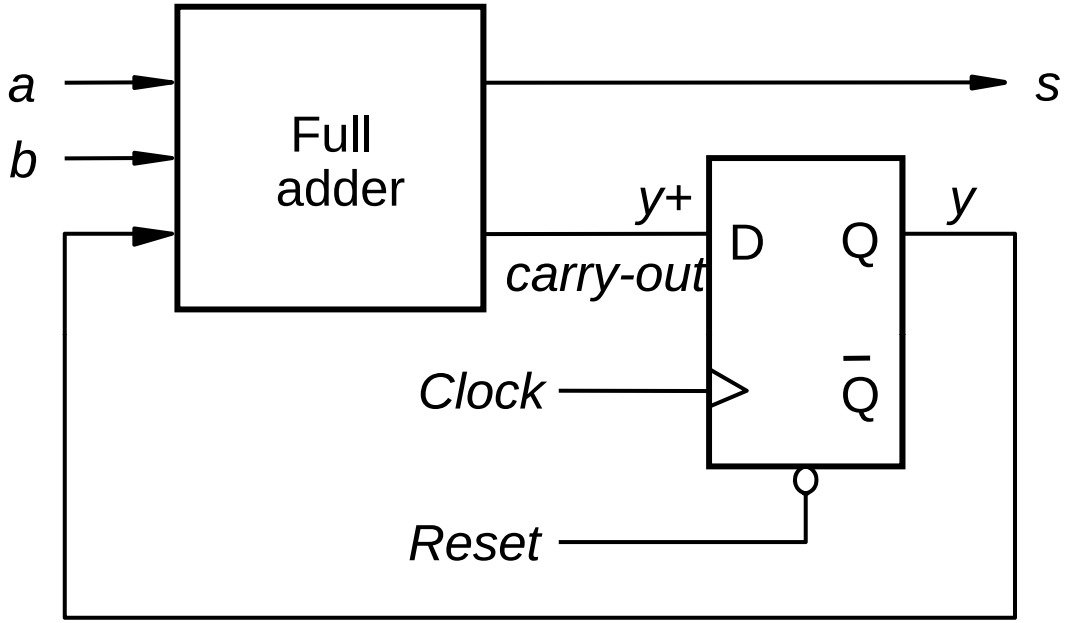


Figure 8.43 Circuit for the adder FSM

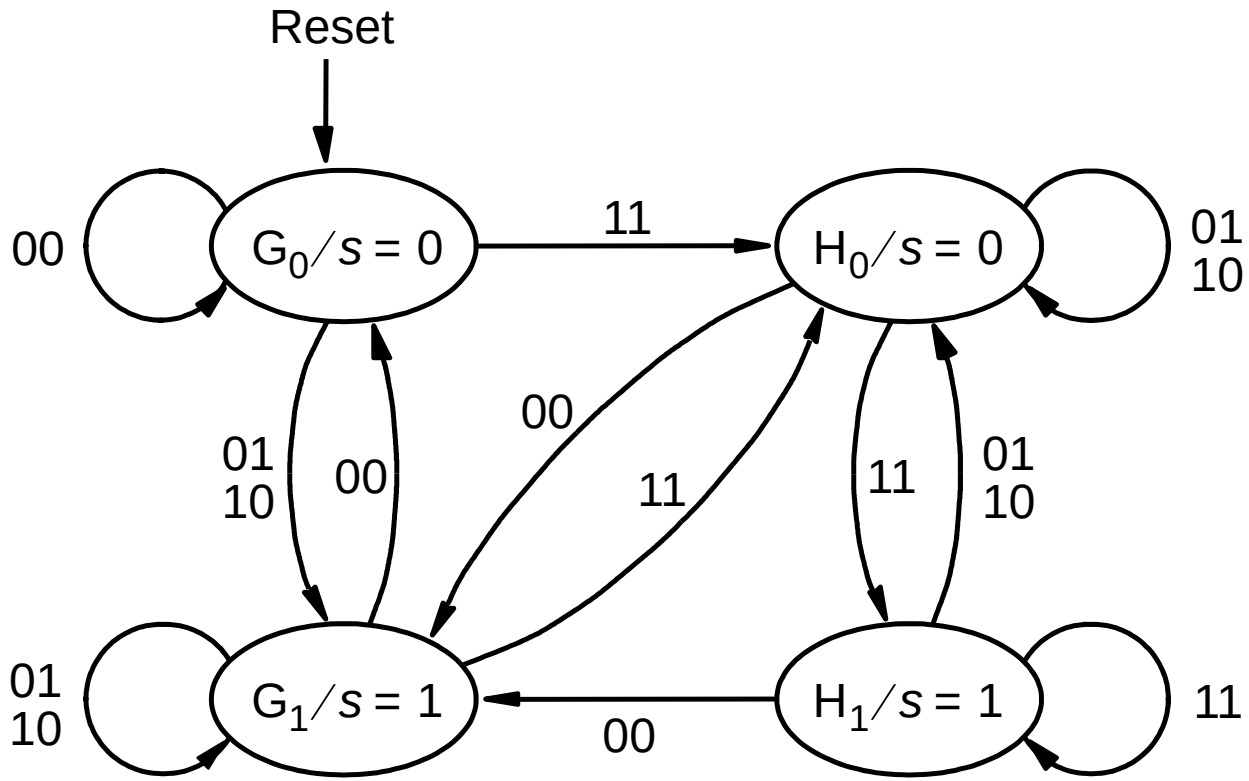


Figure 8.44 State diagram for the Moore-type serial adder FSM

Present state	Nextstate				Output
	ab =00	01	10	11	*s*
G_0	G_0	G_1	G_1	H_0	0
G_1	G_0	G_1	G_1	H_0	1
H_0	G_1	H_0	H_0	H_1	0
H_1	G_1	H_0	H_0	H_1	1

Present state y_2y_1	Nextstate				Output
	ab =00	01	10	11	*s*
	$y_2+\ y_1+$				
00	0 0	01	0 1	10	0
01	0 0	01	0 1	10	1
10	0 1	10	1 0	11	0
11	0 1	10	1 0	11	1

Figure 8.45 State table for the Moore-type serial adder FSM

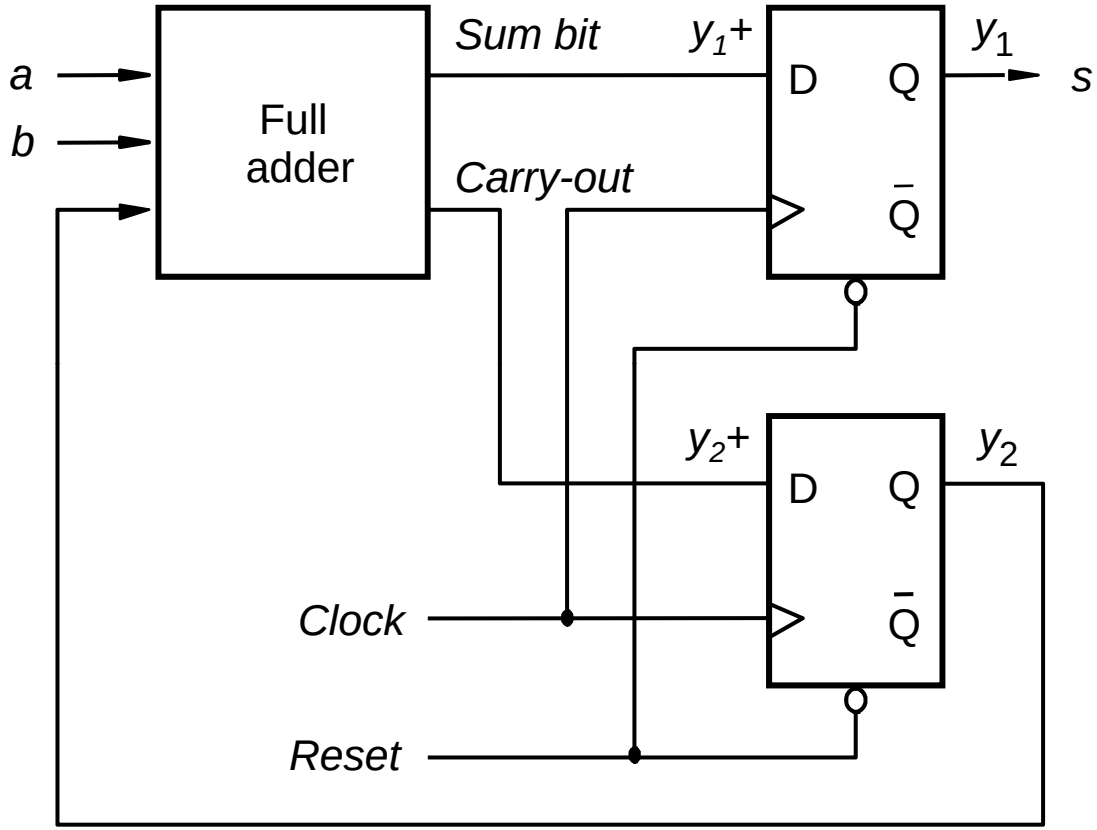


Figure 8.47 Circuit for the Moore-type serial adder FSM

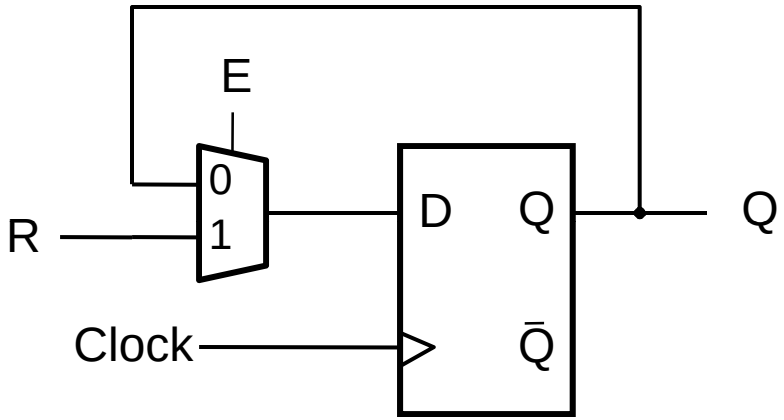
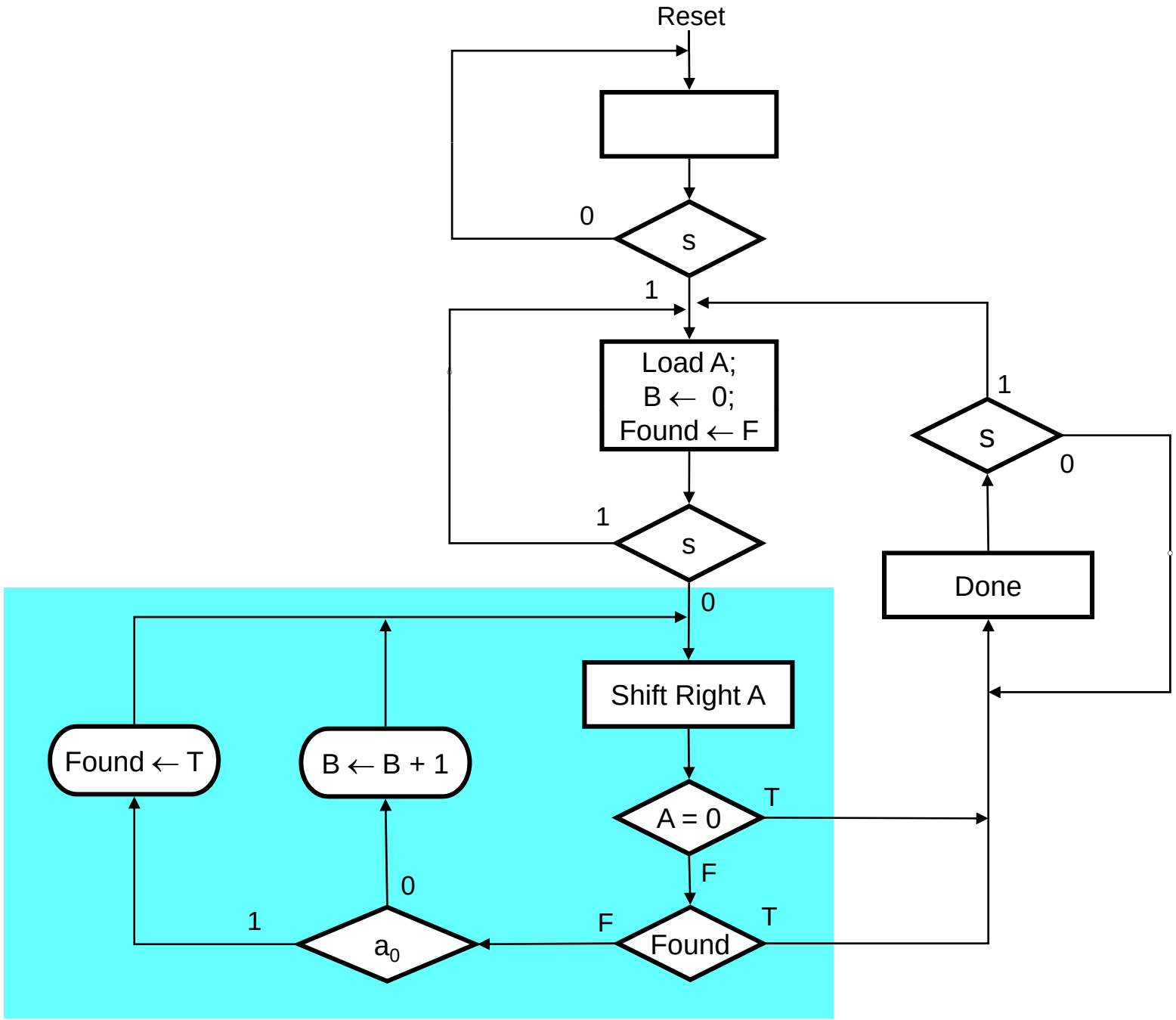


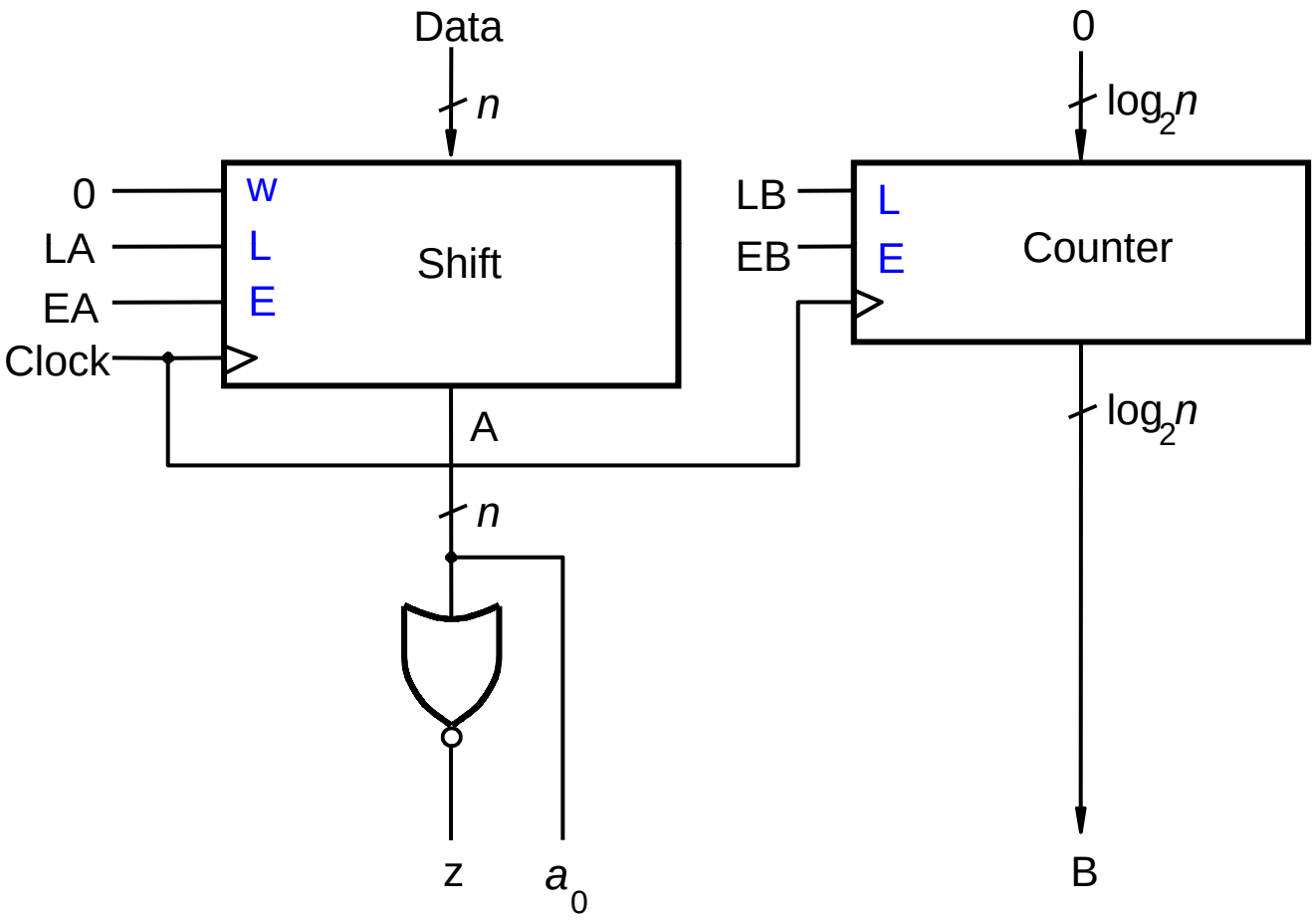
Figure 10.1 A flip-flop with an enable input

Pseudo-Code for First-1 Finder

```
B = 0;
Found = false;
while A ≠ 0 and not found do
    if a0 = 1 then
        Found = true;
    else
        B = B + 1;
    end if
    Right-Shift A;
End while;
```


Reset
0
s
1
Load A;
B ← 0;
Found ← F
1
s
0
Shift Right A
A = 0
T
F
Found
T
F
a0
1
0
B ← B + 1
Found ← T
1
S
0
Done

Data path for the bit counter



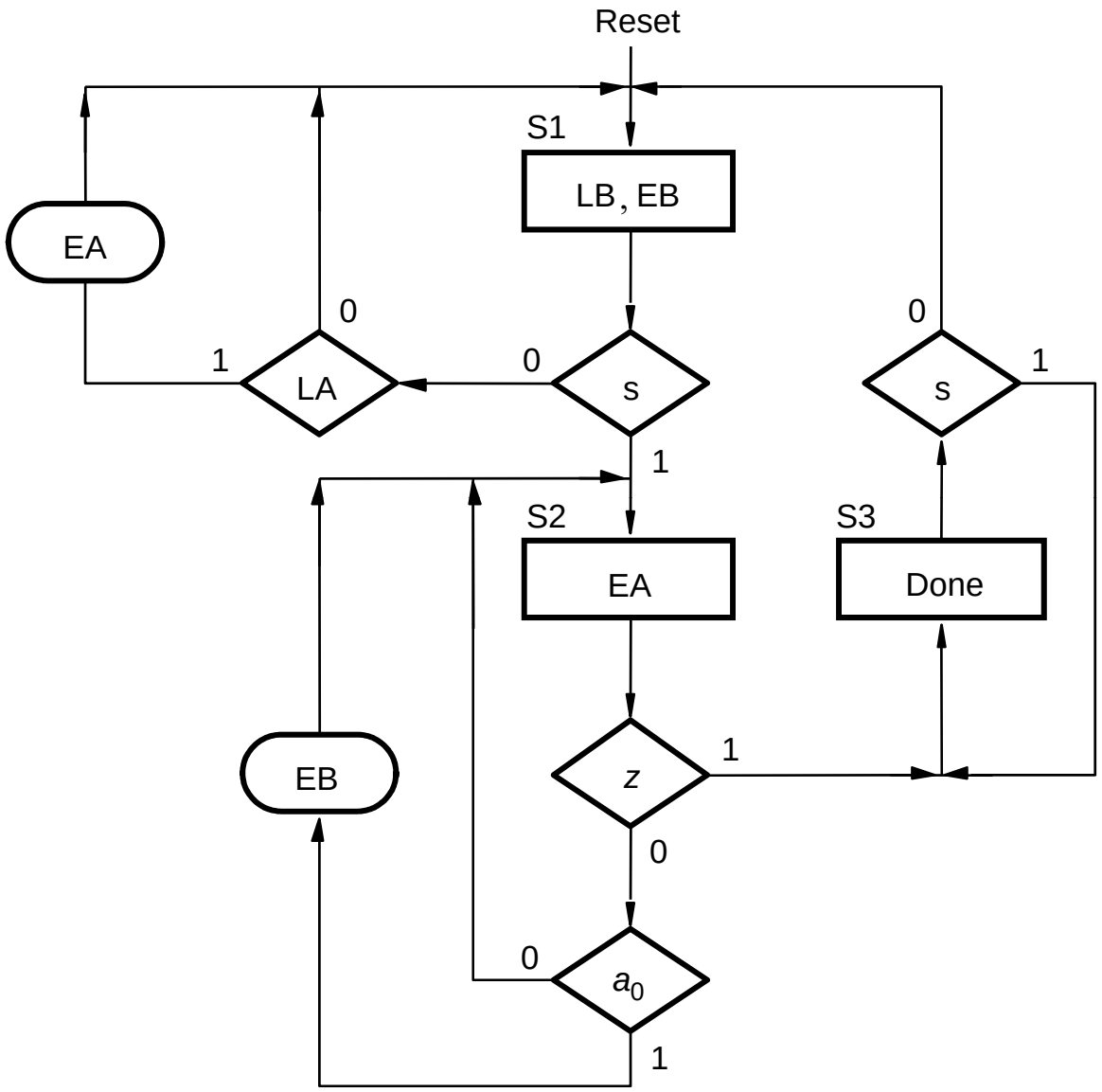


Figure 10.12 ASM chart for the bit counter control circuit

Figure 10.14 Simulation results for the bit-counting circuit

Decimal		Binary	
13		1 1 0 1	Multiplicand
× 11		× 1 0 1 1	Multiplier
13		1 1 0 1	
13		1 1 0 1	
143		0 0 0 0	
		1 1 0 1	
		1 0 0 0 1 1 1 1	Product

(a) Manual method

```
P = 0;
for i = 0 to n – 1 do
    if b_i = 1 then
        P = P + A;
    endif;
    Left-shift A;
endfor;
```

(b) Pseudo-code

Figure 10.15 An algorithm for multiplication

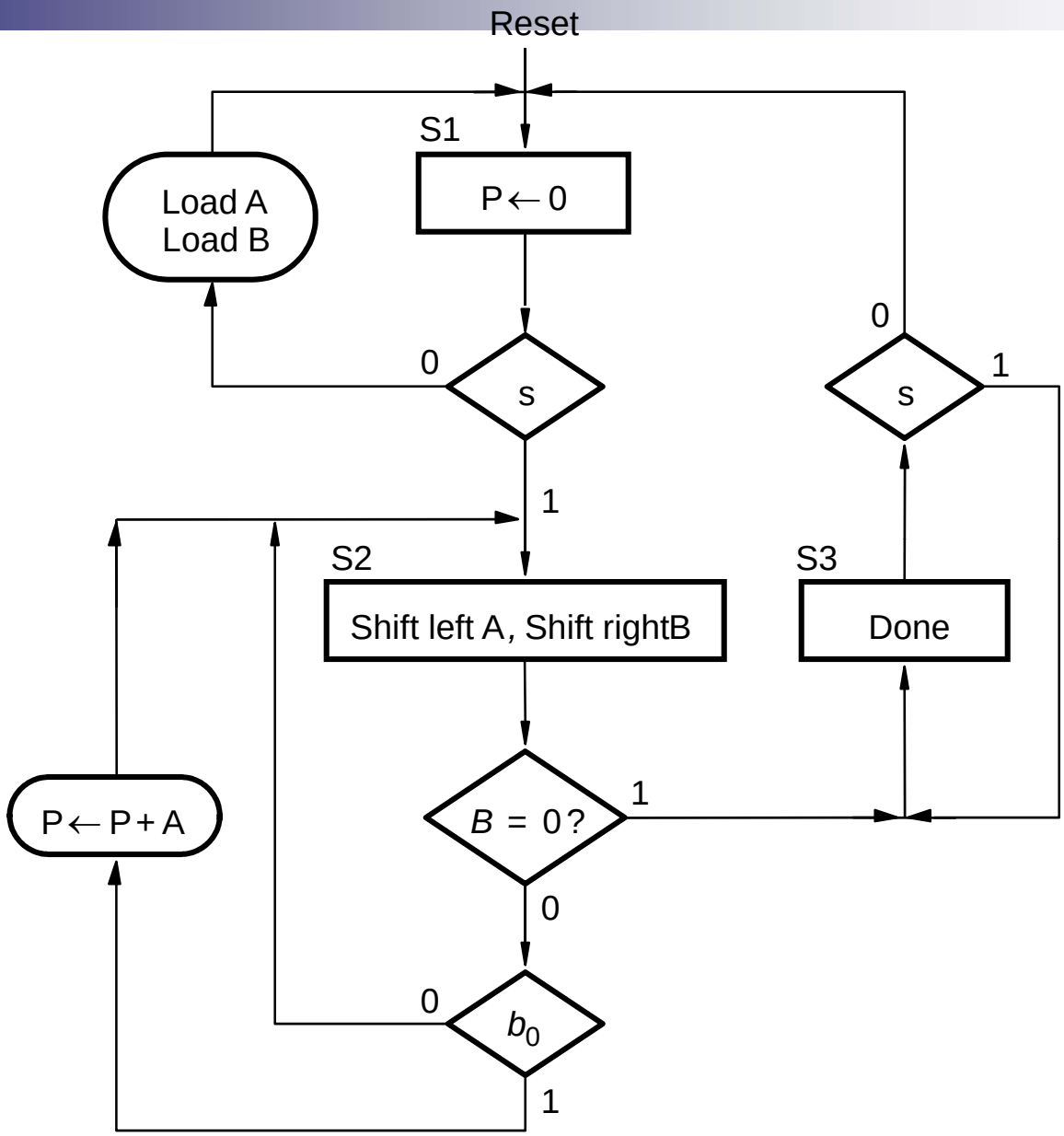


Figure 10.16 ASM chart for the multiplier

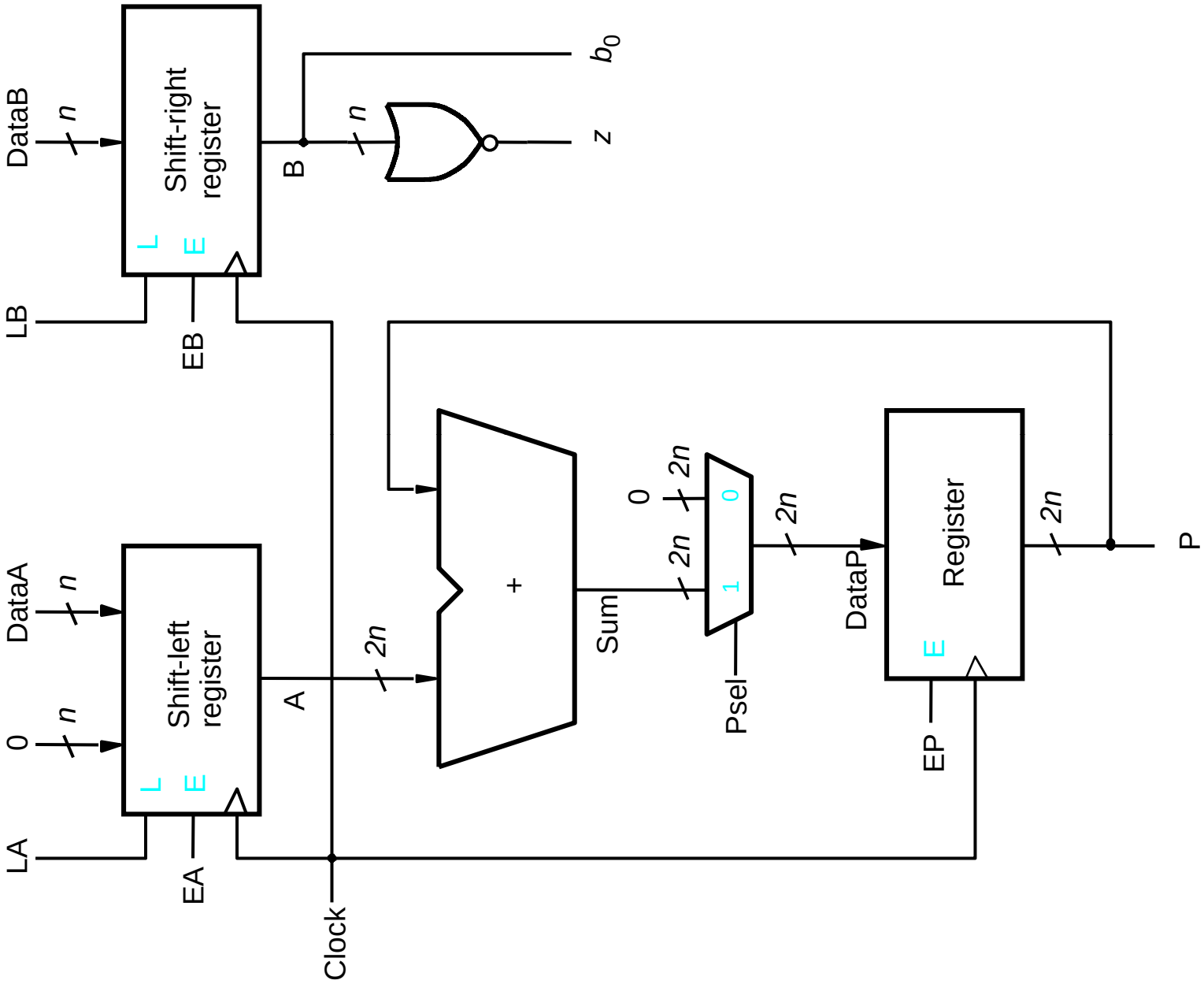


Figure 10.17 Datapath circuit for the multiplier

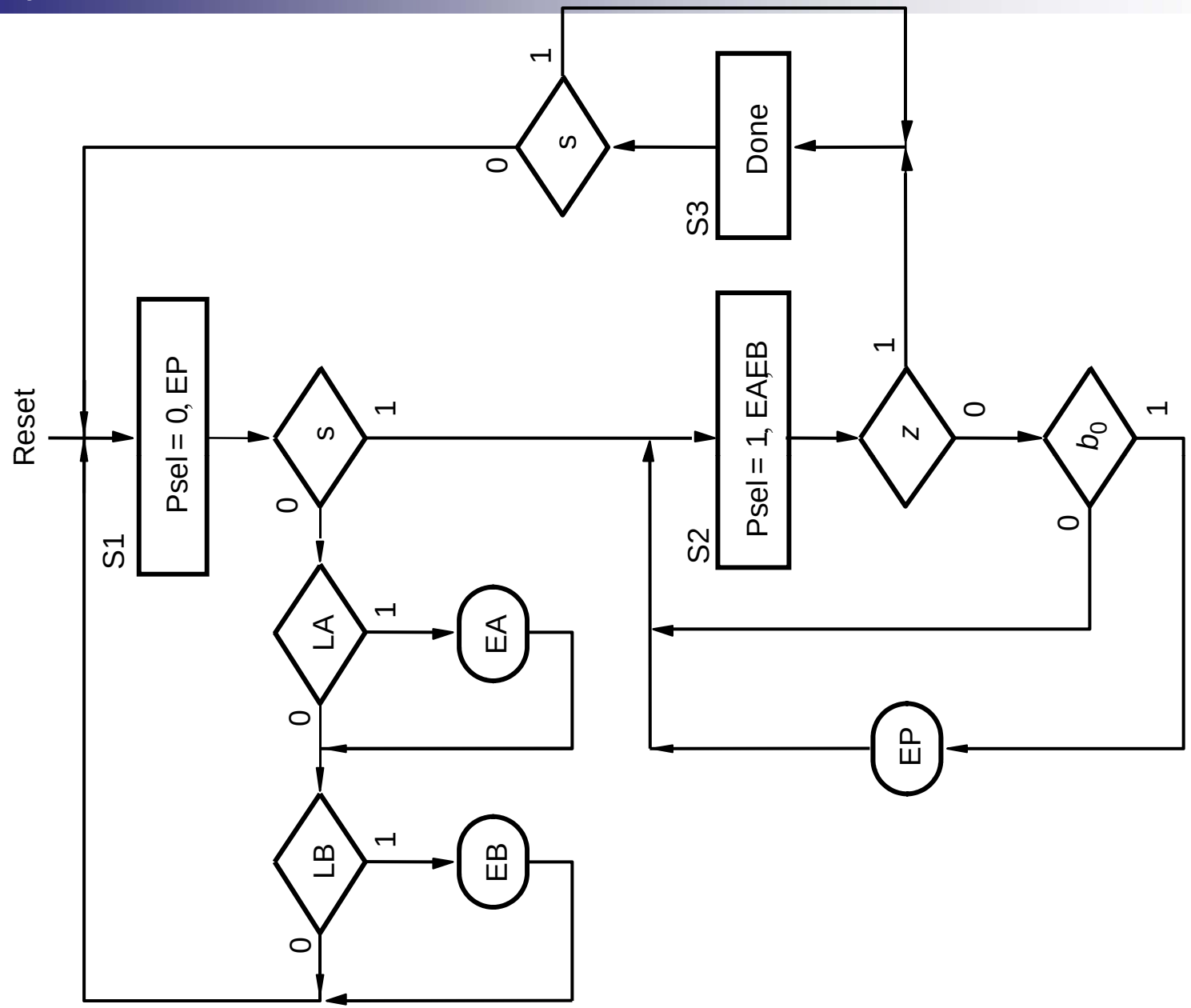


Figure 10.18 ASM chart for the multiplier control circuit

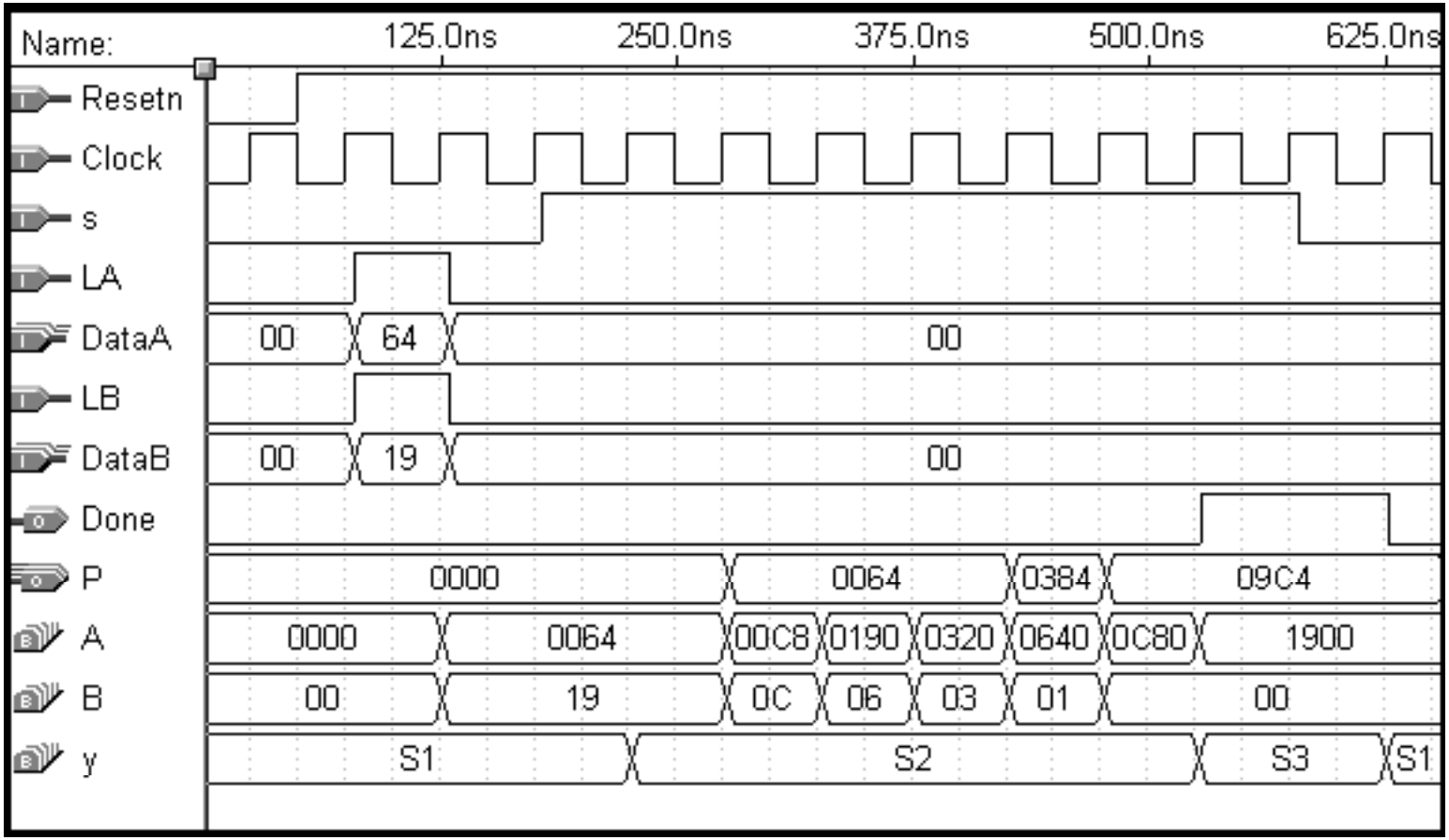
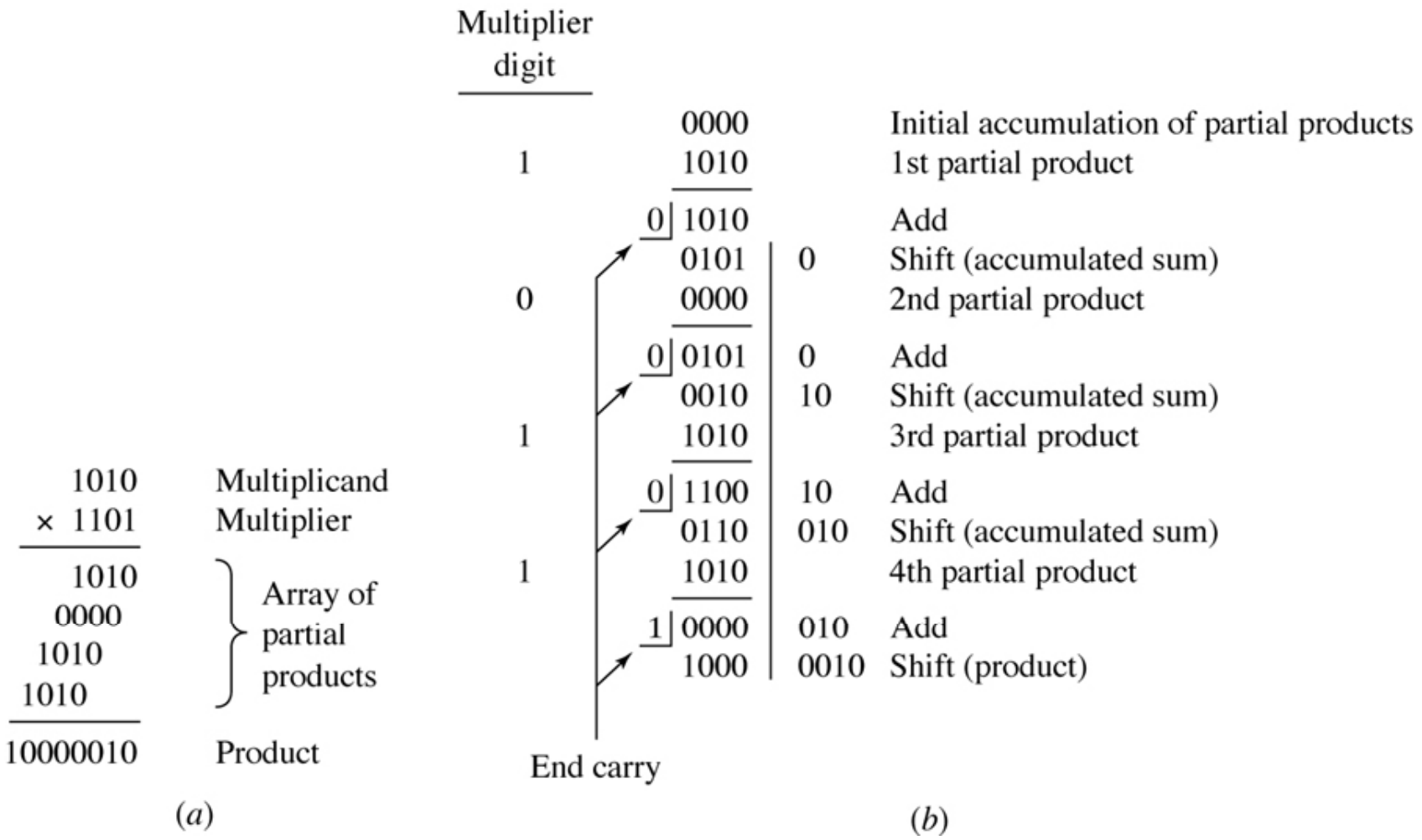


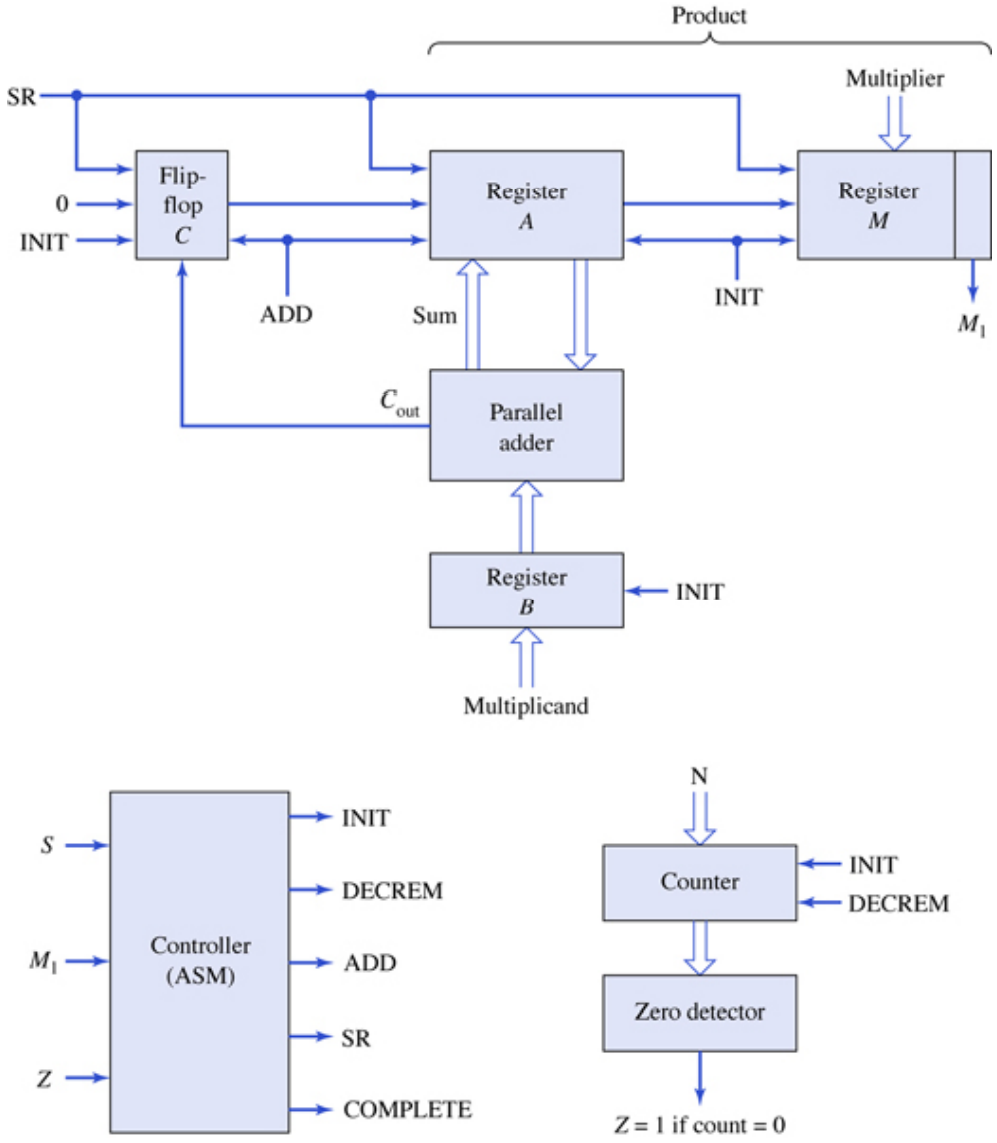
Figure 10.20 Simulation results for the multiplier circuit

More Efficient Version of Multiplication

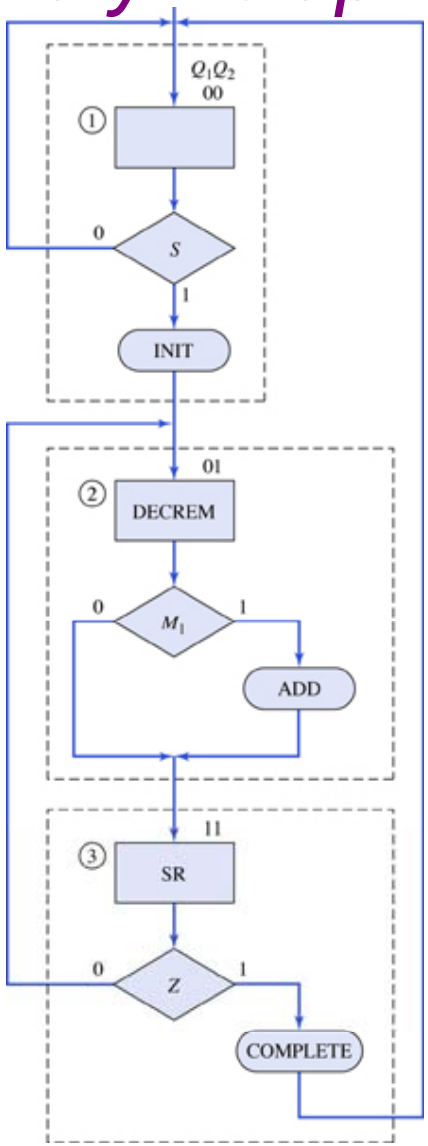


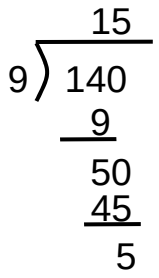
(*a*) Pencil-and-paper approach. (*b*) Add-shift approach.

Architecture for a binary multiplier

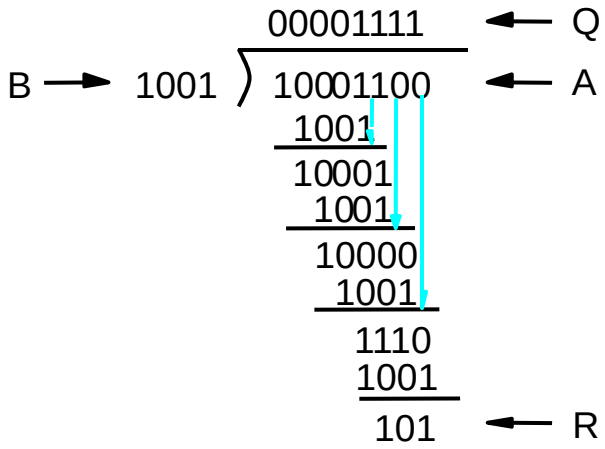


ASM chart for a binary multiplier.





(a) An example using decimal numbers



(b) Using binary numbers

$R = 0$;
for $i = 0$ to $n - 1$ do
    Left-shift $R||A$;
    if $R \geq B$ then
        $q_i = 1$;
        $R = R - B$;
    else
        $q_i = 0$;
    endif;
endfor;

(c) Pseudo-code

Figure 10.21 An algorithm for division

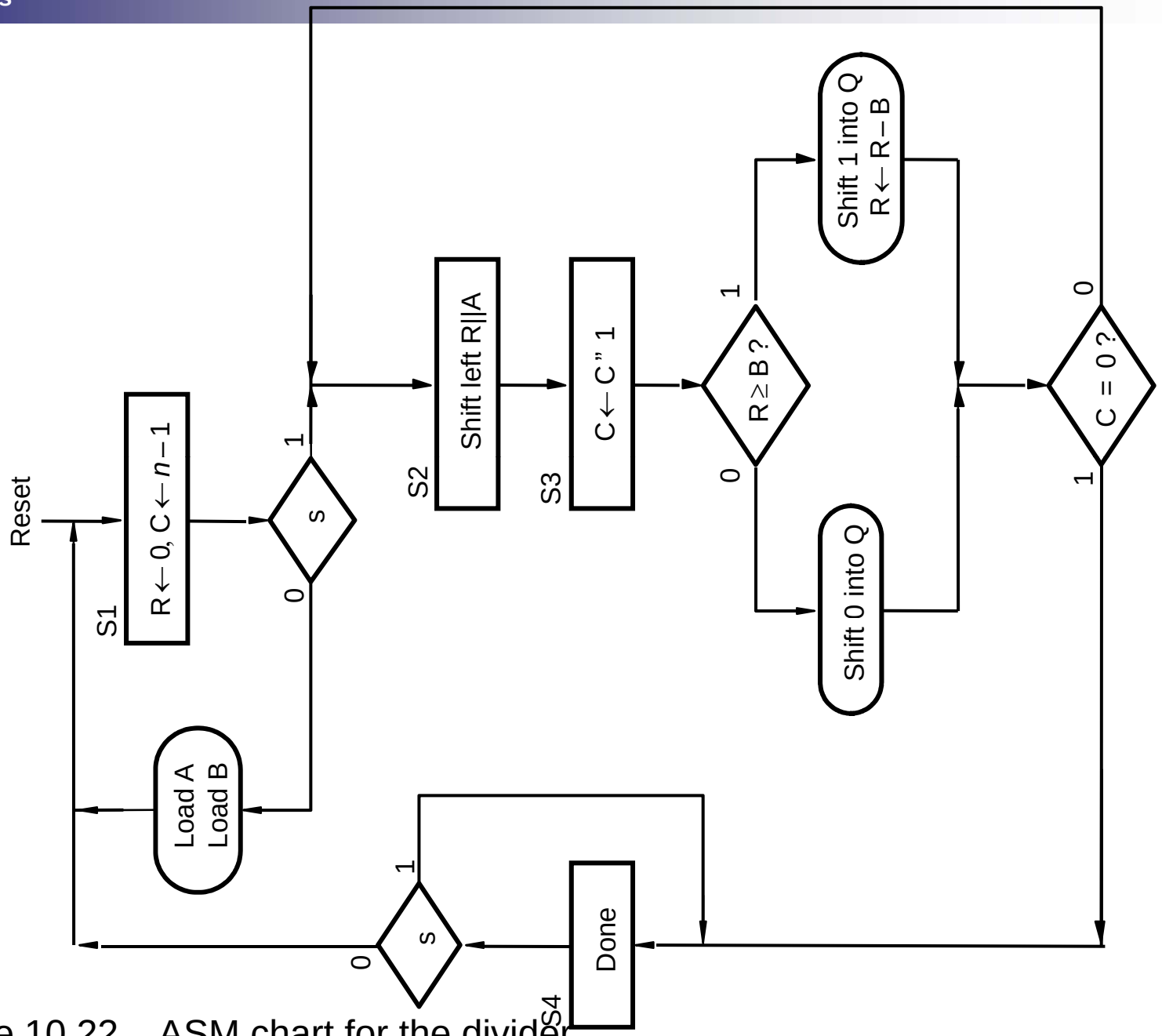


Figure 10.22 ASM chart for the divider

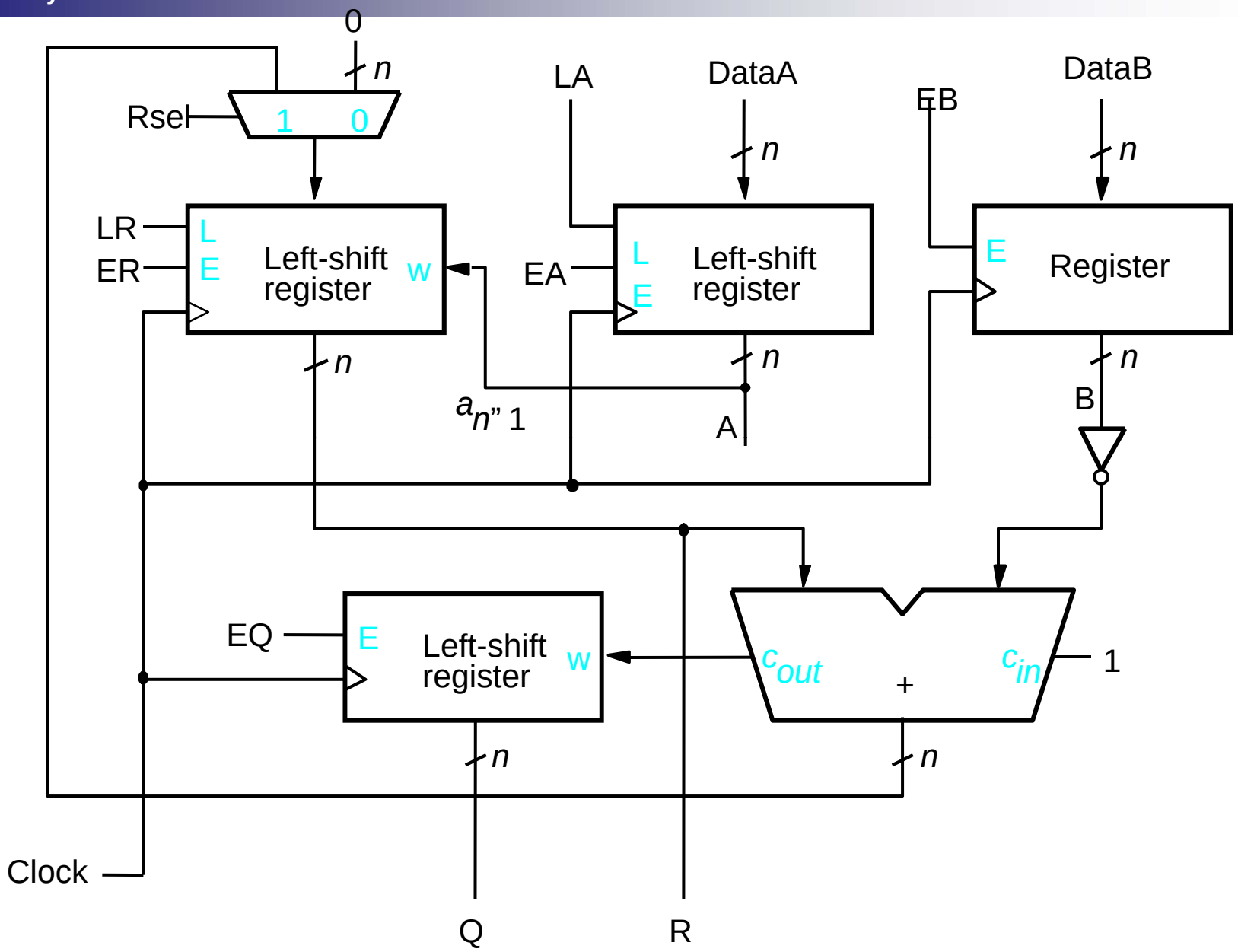


Figure 10.23 Datapath circuit for the divider

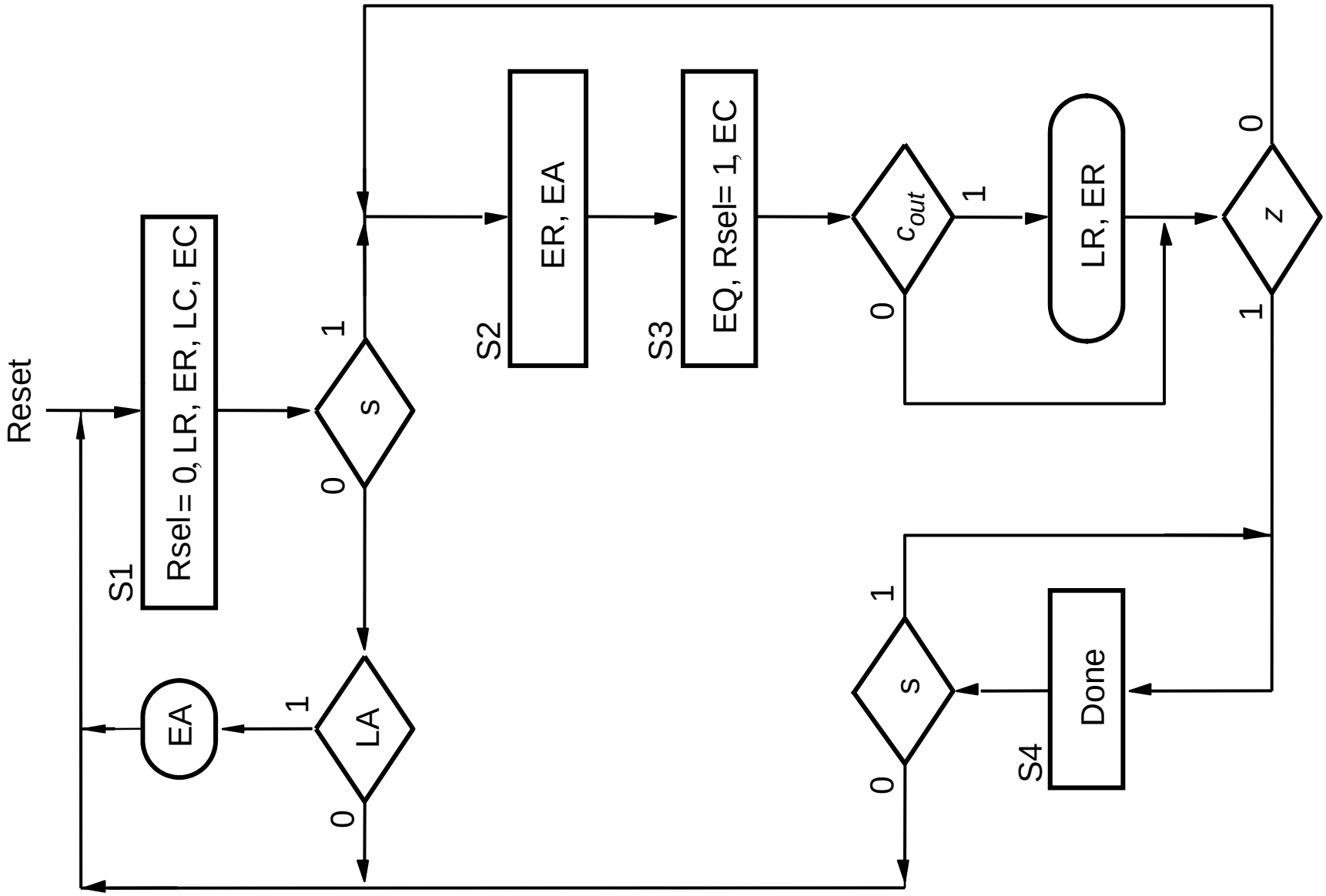
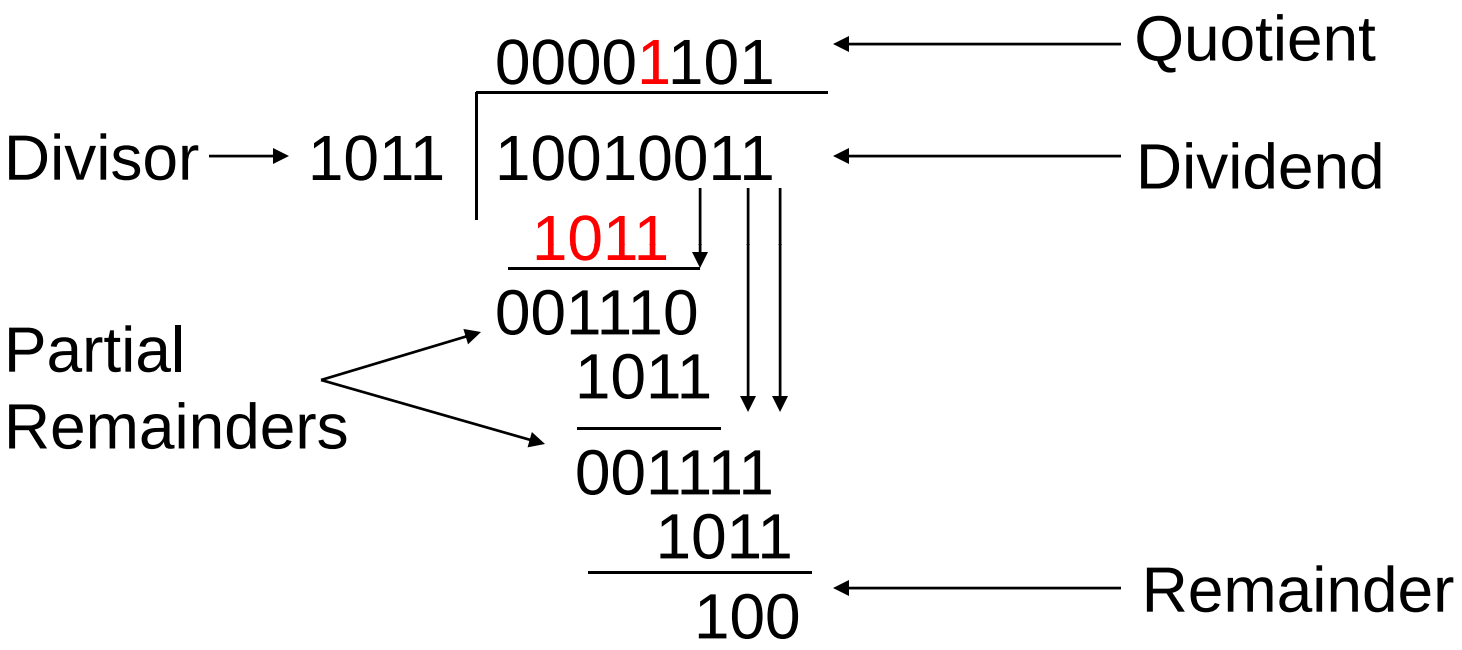


Figure 10.24 ASM chart for the divider control circuit

Division of Unsigned Binary Integers



B → 1001 ⟌ 10001100 ← A

	Clock cycle	R	rr_0	A/Q
	Load A, B	0 0 0 0 0 0 0 0	0	1 0 0 0 1 1 0 0
0	Shift left	0 0 0 0 0 0 0 0	1	0 0 0 1 1 0 0 0
1	Shift left, $Q_0 \leftarrow 0$	0 0 0 0 0 0 0 1	0	0 0 1 1 0 0 0 0
2	Shift left, $Q_0 \leftarrow 0$	0 0 0 0 0 0 1 0	0	0 1 1 0 0 0 0 0
3	Shift left, $Q_0 \leftarrow 0$	0 0 0 0 0 1 0 0	0	1 1 0 0 0 0 0 0
4	Shift left, $Q_0 \leftarrow 0$	0 0 0 0 1 0 0 0	1	1 0 0 0 0 0 0 0
5	Subtract, $Q_0 \leftarrow 1$	0 0 0 0 1 0 0 0	1	0 0 0 0 0 0 0 1
6	Subtract, $Q_0 \leftarrow 1$	0 0 0 0 1 0 0 0	0	0 0 0 0 0 0 1 1
7	Subtract, $Q_0 \leftarrow 1$	0 0 0 0 0 1 1 1	0	0 0 0 0 0 1 1 1
8	Subtract, $Q_0 \leftarrow 1$	0 0 0 0 0 1 0 1	0	0 0 0 0 1 1 1 1

Figure 10.25 An example of division using $n = 8$ clock cycles

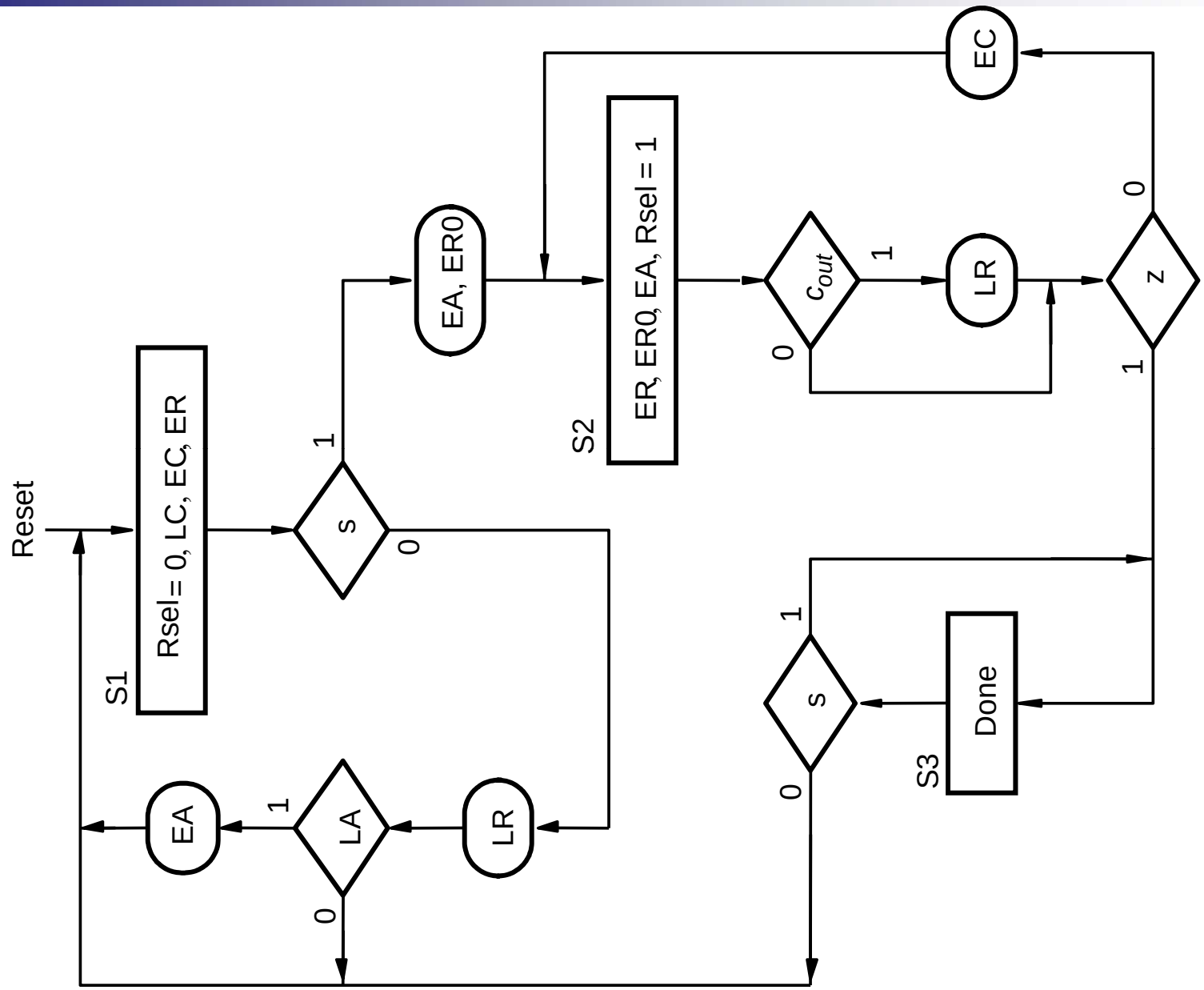


Figure 10.26 An example of division using $n = 8$ clock cycles

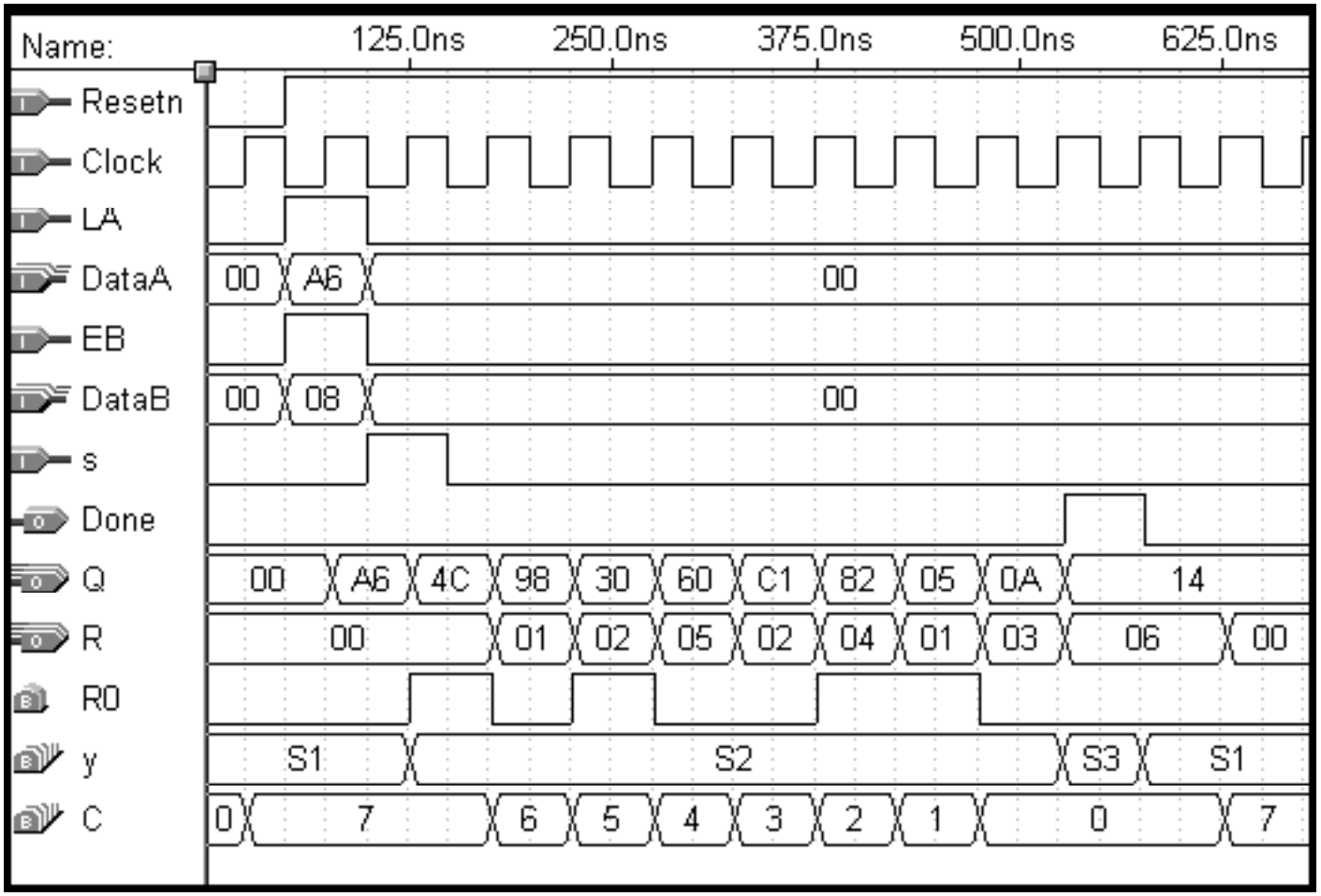


Figure 10.29 Simulation results for the divider circuit

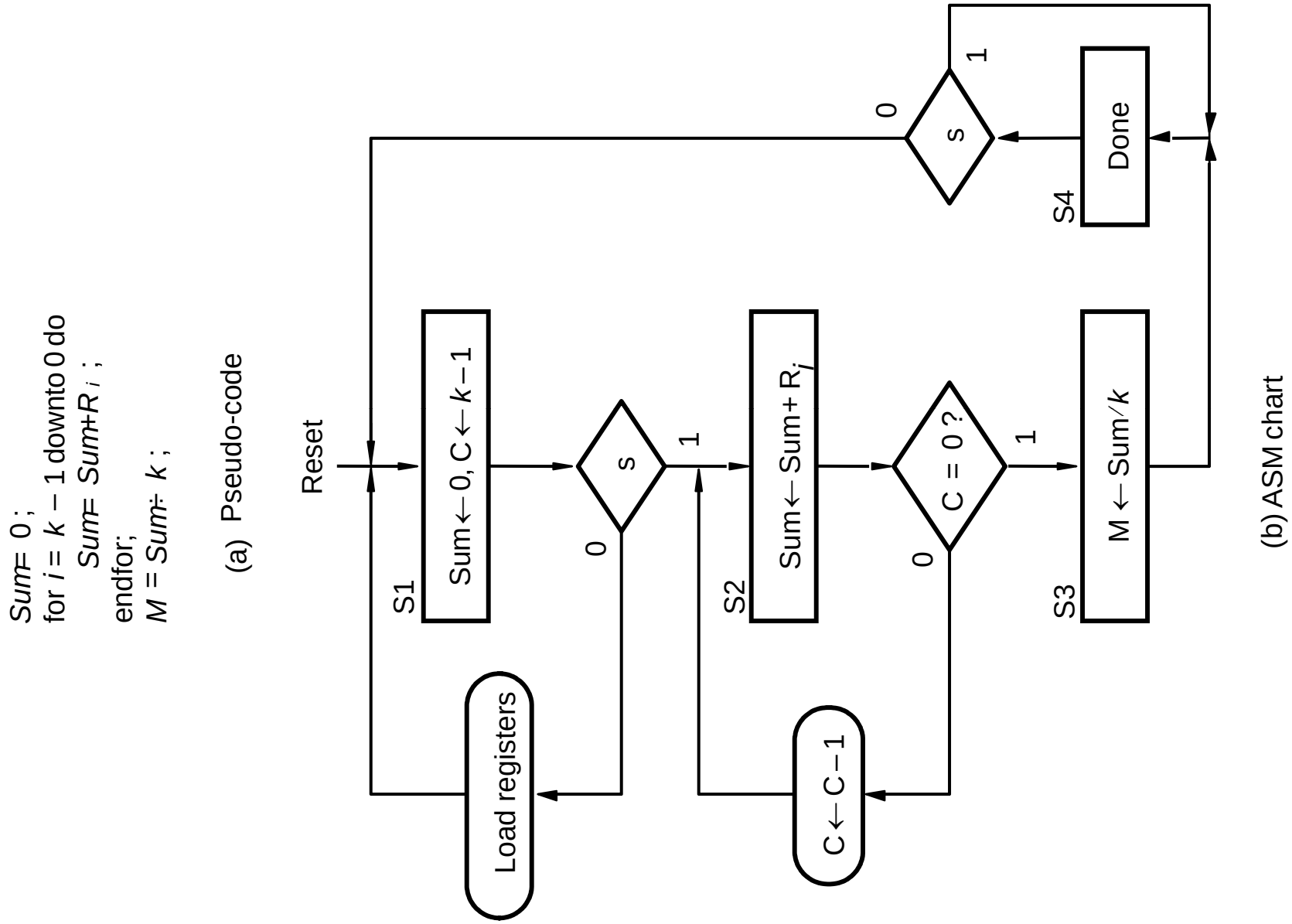


Figure 10.30 An algorithm for finding the mean of k numbers

Figure 10.31 Datapath circuit for the mean operation

Figure 10.32 ASM chart for the control circuit

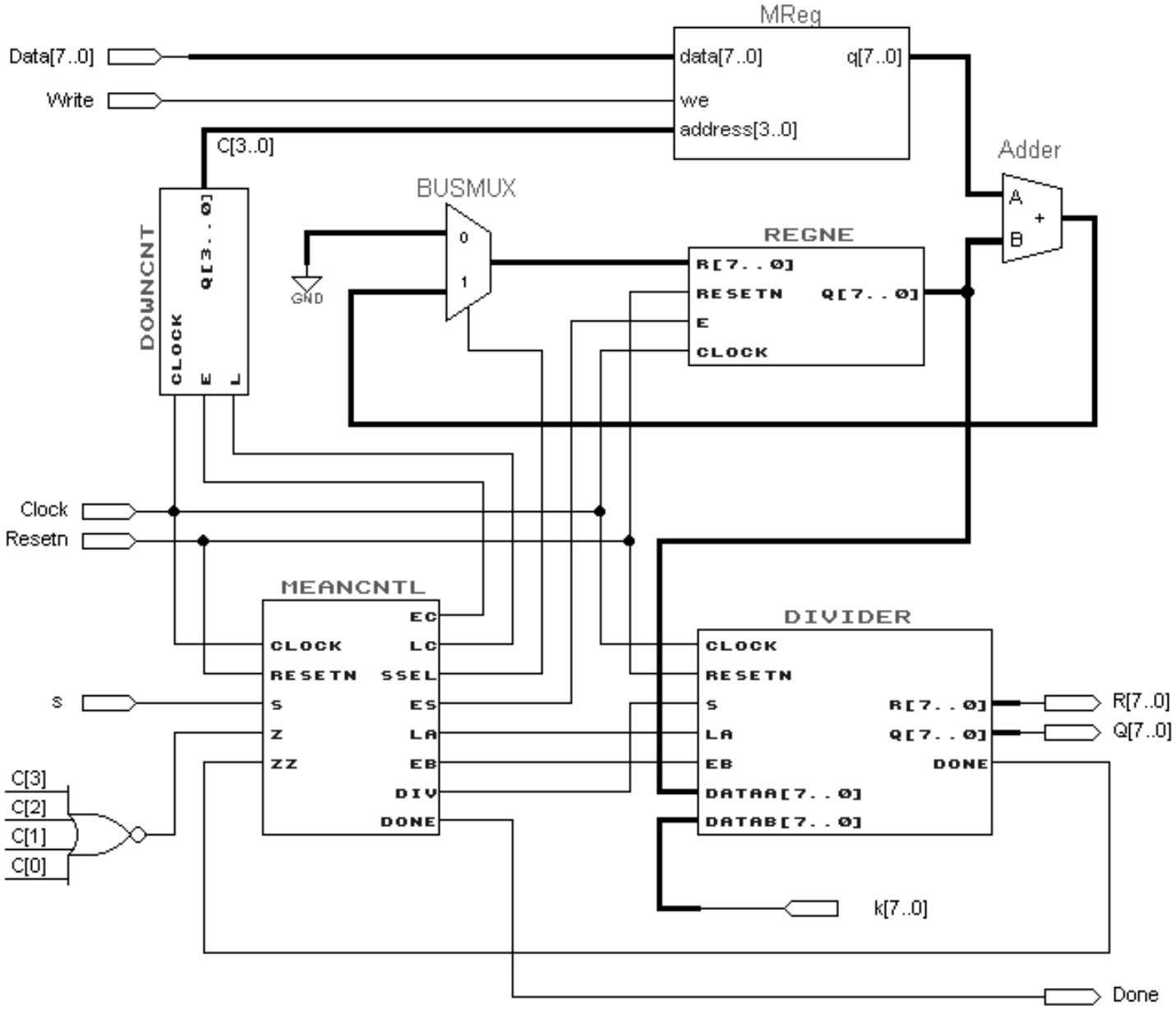


Figure 10.33 Schematic of the mean circuit with an SRAM block

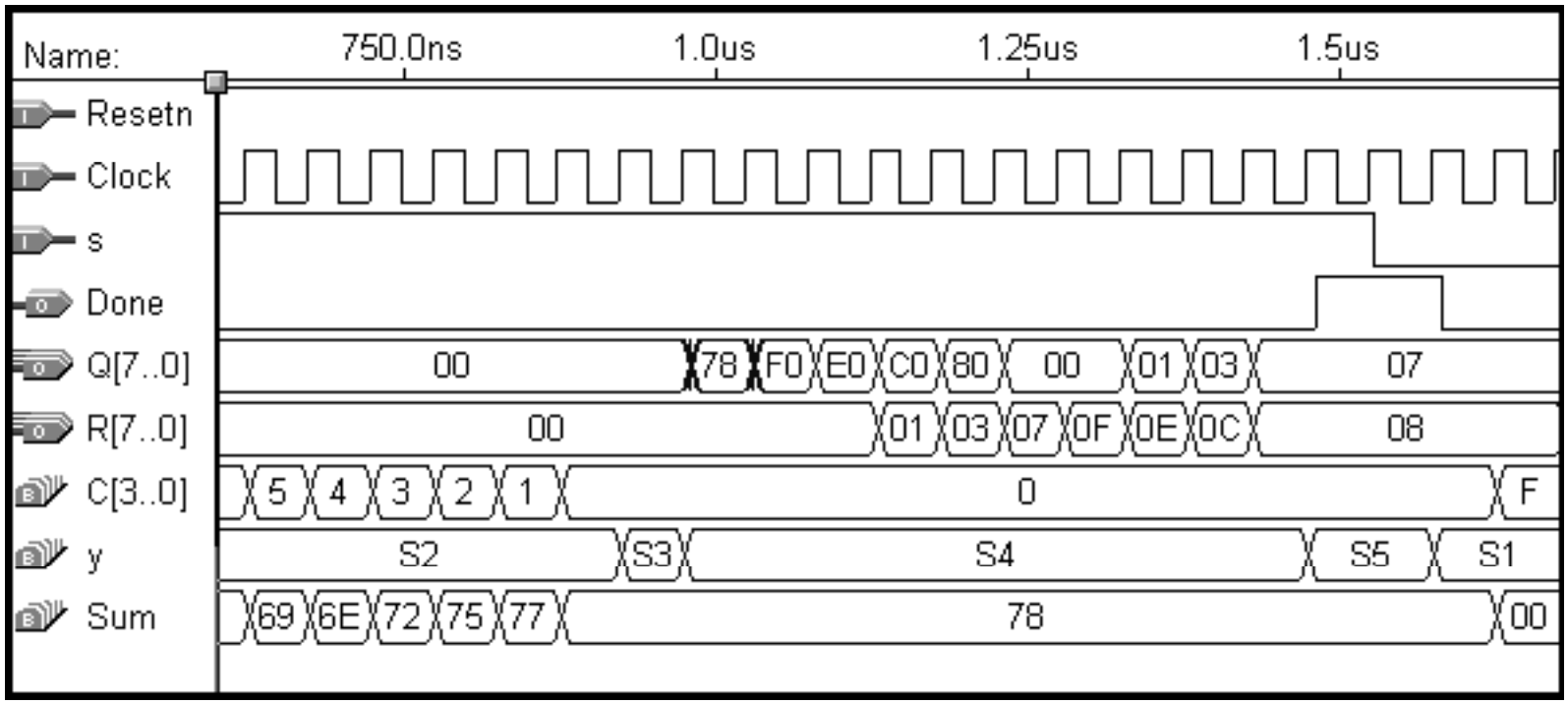


Figure 10.34 Simulation results for the mean circuit using SRAM

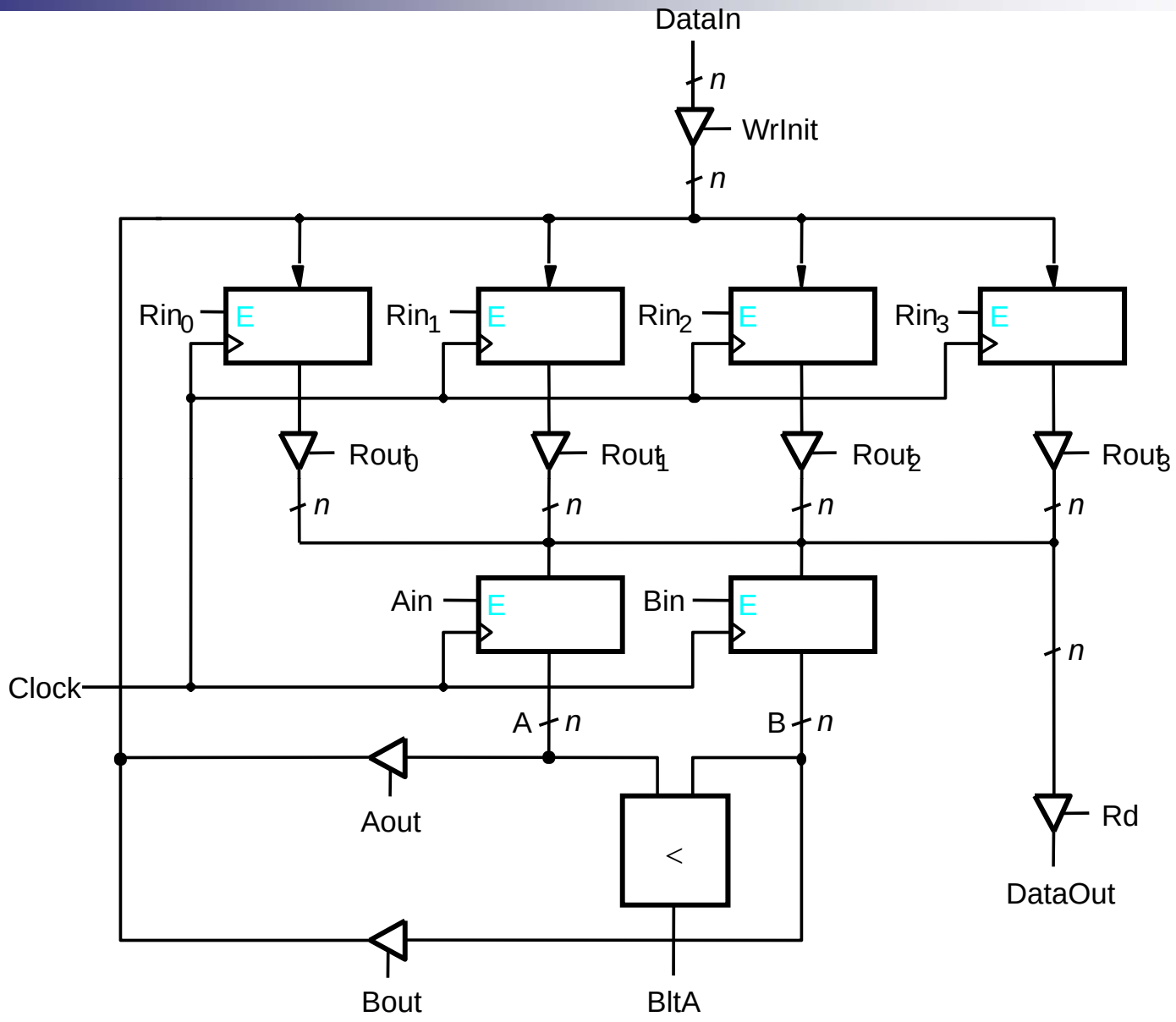


Figure 10.42 Using tri-state buffers in the datapath circuit

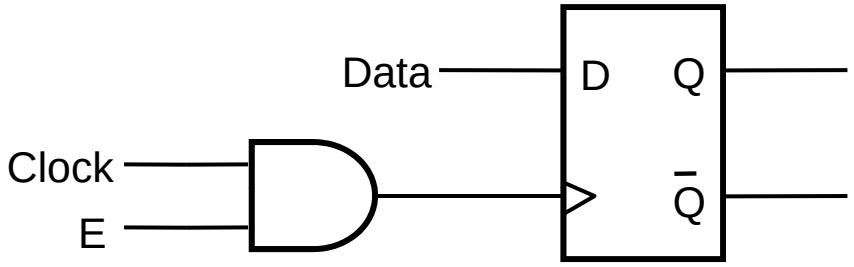


Figure 10.43 Clock enable circuit