Answers to Selected Problems

CHAPTER 1

1.2 **(a)** 32,768 **(b)** 67,108,864 **(c)** 6,871,947,674

1.3 **(a)** $(4310)_5 = 580$ **(b)** $(198)_{12} = 260$

1.5 **(a)** 6 **(b)** 8 **(c)** 11

1.6 8

1.7 $(62315)_8$

1.9 22.3125 (all three)

1.12 **(a)** 10000 and 110111 **(b)** 62 and 958

1.19 **(a)** 010087 **(b)** 008485 **(c)** 991515 **(d)** 989913

1.24 **(a)**

6	**3**	**1**	**1**	**Decimal**
0	0	0	0	0
0	0	0	1	1
0	0	1	1	2
0	1	0	0	3
0	1	1	0	4 (*or* 0101)
0	1	1	1	5
1	0	0	0	6
1	0	1	0	7 (*or* 1001)
1	0	1	1	8
1	1	0	0	9

1.29 Steve Jobs

1.31 $62 + 32 = 94$ printing characters

1.32 bit 6 from the right

1.33 **(a)** 897 **(b)** 564 **(c)** 871 **(d)** 2,199

CHAPTER 2

2.2 **(a)** x **(b)** x **(c)** y **(d)** 0

2.3 **(a)** B **(b)** $z(x + y)$ **(c)** $x'y'$ **(d)** $x(w + y)$ **(e)** 0

2.4 **(a)** $AB + C'$ **(b)** $x + y + z$ **(c)** B **(d)** $A'(B + C'A)$

2.9 **(a)** $xy + x'y'$

2.11 $F(x, y, z) = \Sigma(1, 4, 5, 6, 7)$

2.12 **(a)** 10100000 **(c)** 00011101 **(d)** 01001110

2.14 **(b)** $(x' + y')' + (x + y)' + (y + z')'$

2.15 $T_1 = A'(B' + C')$

$T_2 = A + BC = T_1'$

2.17 **(a)** $\Sigma(3, 5, 6, 7) = \Pi(0, 1, 2, 4)$

2.18 **(c)** $F = y'z + y(w + x)$

2.19 $\Sigma(1, 3, 5, 7, 9, 11, 13, 15) = \Pi(0, 2, 4, 6, 8, 10, 12, 14)$

2.22 **(a)** $AB + BC = (A + C)B$ **(b)** $x' + y + z'$

CHAPTER 3

3.1 **(a)** $xy' + x'z'$ **(b)** $xy' + z'$ **(c)** $x' + y'z$ **(d)** $x'y + x'z + yz$

3.2 **(a)** $x'y' + xz$ **(b)** $y + x'z$

3.3 **(a)** $xy + x'z'$ **(b)** $x' + yz$ **(c)** $z' + x'y$

3.4 **(a)** y **(b)** $BCD + A'BD'$ **(c)** $ABD + ABC + CD$

(d) $wx + w'x'y$

3.5 **(a)** $xz' + w'y'z + wxy$ **(d)** $BD + B'D' + A'B$ or $BD + B'D' + A'D'$

3.6 **(a)** $B'D' + A'BD + ABC'$ **(b)** $xy' + x'z + wx'y$

3.7 **(a)** $x'y + z$ **(c)** $AC + B'D' + A'BD + B'C$ (or CD)

3.8 **(a)** $F(x, y, z) = \Sigma(3, 5, 6, 7)$ **(b)** $F(A, B, C, D) = \Sigma(1, 3, 5, 9, 12, 13, 14)$

3.9 **(a)** Essential: xz and $x'z'$; Nonessential: $w'x$ and $w'z'$

(b) $F = B'D' + AC + A'BD + (CD$ **or** $B'C)$

3.10 **(c)** $F = BC' + AC + A'B'D$

Essential: BC', AC

Nonessential: AB, $A'B'D$, $B'CD$, $A'C'D$

3.11 **(a)** $F = A'B'D' + AD'E + B'C'D'$

3.12 **(b)** $F = (A + D')(B' + D')$

3.13 **(a)** $F = xy + z' = (x + z')(y + z')$

3.15 **(b)** $F = B'D' + CD' + ABC'D = \Sigma(0, 2, 6, 8, 10, 13, 14)$

3.17 $F' = AC' + BC' + BD$

3.19 **(a)** $F = (w + z')(x' + z')(w' + x' + y')$

3.30 $F = (A \oplus B)(C \oplus D)$

3.35 The HDL description is available on the Companion Website.

Line 1: Dash not allowed, use underscore: Exmpl_3.
Terminate line with semicolon (;).
Line 2: **inputs** should be **input** (no s at the end).
Change last comma (,) to semicolon (;). Output is declared but does not appear in the port list, and should be followed by a comma if it is intended to be in the list of inputs. If *Output* is a mispelling of **output** and is to declare output ports, C should be followed by a semicolon (;) and *F* should be followed by a semicolon (;).
Line 3: *B* cannot be declared as input (Line 2) and output (Line 3). Terminate the line with a semicolon (;).
Line 4: *A* cannot be an output of the primitive if it is an input to the module
Line 5: Too many entries for the not gate (only two allowed).
Line 6: OR must be in lowercase: change to "or".
Line 7: **endmodule** is mispelled. Remove semicolon (no semicolon after endmodule).

CHAPTER 4

4.1 **(a)** $F_1 = A + B'C + BD' + B'D$
$F_2 = A'B + D$

4.2 $F = ABC + A'D$
$G = ABC + A'D'$

4.3 **(b)** 1024 rows and 14 columns

4.4 **(a)** $F = x'y' + x'z'$

4.6 $F = xy + xz + yz$

4.7 **(a)** $w = A \quad x = A \oplus B \quad y = x \oplus C \quad z = y \oplus D$

4.8 $w = AB + AC'D'$

4.10 Inputs: A, B, C, D; Outputs: *w, x, y, z*

$z = D$

$y = C \oplus D$

$x = B \oplus (C + D)$

$w = A \oplus (B + C + D)$

4.12 **(b)** $Diff = x \oplus y \oplus B_{in}$

$B_{out} = x'y + x'B_{in} + yB_{in}$

4.13

	Sum	C	V
(a)	1101	0	1
(b)	0001	1	1
(c)	0100	1	0
(d)	1011	0	1
(e)	1111	0	0

4.14 30 ns

4.18 $w = A'B'C'$

$x = B \oplus C$

$y = C$

$z = D'$

4.22 $w = AB + ACD$

$x = B'C' + B'D' + BCD$

$y = C'D + CD'$

$z = D'$

4.28 **(a)** $F_1 = \Sigma(0, 5, 7)$

$F_2 = \Sigma(2, 3, 4)$

$F_3 = \Sigma(1, 6, 7)$

4.29 $x = D_0'D_1'$

$y = D_0'D_1 + D_0'D_2'$

4.34 $F(A, B, C, D) = \Sigma(1, 6, 7, 9, 10, 11, 12)$

4.35 **(a)** When $AB = 00$, $F = D$

When $AB = 01$, $F = (C + D)'$

When $AB = 10$, $F = CD$

When $AB = 11$, $F = 1$

4.39 The HDL description is available on the Companion Website.

4.42 **(c)** The HDL description is available on the Companion Website.

4.50 The HDL description is available on the Companion Website.

4.56
```
assign match = (A == B);          // Assumes reg [3: 0] A, B;
```

4.57 The HDL description is available on the Companion Website.

CHAPTER 5

5.4 **(b)** $PQ' + NQ$

5.7 $S = x \oplus y \oplus Q$

$Q(t + 1) = xy + xQ + yQ$

5.8 A counter with a repeated sequence of 00, 01, 10

5.9 **(a)** $A(t + 1) = xA' + AB$

$B(t + 1) = xB' + A'B$

5.10 **(c)** $A(t + 1) = xB + x'A + yA + y'A'B'$

$B(t + 1) = xA'B' + x'A'B + yA'B'$

5.11

Present state:	00	00	01	00	01	11	00	01	11	10	00	01	11	10	10
Input:	0	1	0	1	1	0	1	1	1	0	1	1	1	1	0
Output:	0	0	1	0	0	1	0	0	0	1	0	0	0	0	1
Next state:	00	01	00	01	11	00	01	11	10	00	01	11	10	10	00

5.12

Present state	Next state 0	Next state 1	Output 0	Output 1
a	f	b	0	0
b	d	a	0	0
d	g	a	1	0
f	f	b	1	1
g	g	d	0	1

5.13 **(a)**

State:	a	f	b	c	e	d	g	h	g	g	h	a
Input:	0	1	1	1	0	0	1	0	0	1	1	
Output:	0	1	0	0	0	1	1	1	0	1	0	

(b)

State:	a	f	b	a	b	d	g	d	g	g	d	a
Input:	0	1	1	1	0	0	1	0	0	1	1	
Output:	0	1	0	0	0	1	1	1	0	1	0	

5.15 $D_Q = Q'J + QK'$

5.16 $D_A = Ax' + Bx$

$D_B = A'x + Bx'$

5.18 $J_A = K_A = (BF + B'F')E$

$J_B = K_B = E$

5.19 **(a)** $D_A = A'B'x\_in$

$D_B = A + C'x\_in' + BCx\_in$

$D_C = Cx\_in' + Ax\_in + A'B'x\_in'$

$y\_out = A'x\_in$

5.23 **(a)** $RegA = 125, \quad RegB = 125$

(b) $RegA = 125, \quad RegB = 30$

5.26 **(a)**

$Q(t + 1) = JQ' + K'Q$

When $Q = 0, Q(t + 1) = J$

When $Q = 1, Q(t + 1) = K'$

```
module JK_Behavior (output reg Q, input J, K, CLK);
 always @ (posedge CLK)
          if (Q == 0)          Q <= J;
          else                 Q <= ~K;
endmodule
```

5.31 The HDL description is available on the Companion Website.
Note: The statements must be written in an order that produces the effect of concurrent assignments.

CHAPTER 6

6.4 0110; 0011; 0001; 1000; 1100; 1110; 0111; 1011

6.8 $A = 0010, 0001, 1000, 1100$. Carry $= 1, 1, 1, 0$

6.9 **(b)** $J_Q = x'y; K_Q = (x' + y)'$

6.14 **(a)** 4

6.15 30 ns; 33.3 MHz

6.16 $1010 \rightarrow 1011 \rightarrow 0100$

$1100 \rightarrow 1101 \rightarrow 0100$

$1110 \rightarrow 1111 \rightarrow 0000$

6.17 $D_{A0} = A_0 \oplus E$

$D_{A1} = A_1 \oplus (A_0E)$

$D_{A2} = A_2 \oplus (A_1A_0E)$

$D_{A3} = A_3 \oplus (A_2A_1A_0E)$

6.19 **(b)** $D_{Q1} = Q_1'$

$D_{Q2} = Q_2Q_1' + Q_8'Q_2'Q_1$

$D_{Q4} = Q_4Q_1' + Q_4Q_2' + Q_4'Q_2'Q_1$

$D_{Q8} = Q_8Q_1' + Q_4Q_2Q_1$

6.21 $J_{A0} = LI_0 + L'C$

$K_{A0} = LI_0' + L'C$

6.24 $T_A = A \oplus B$

$T_B = B \oplus C$

$T_C = AC + A'C'$ (not self-starting)

$= AC + A'B'C$ (self-starting)

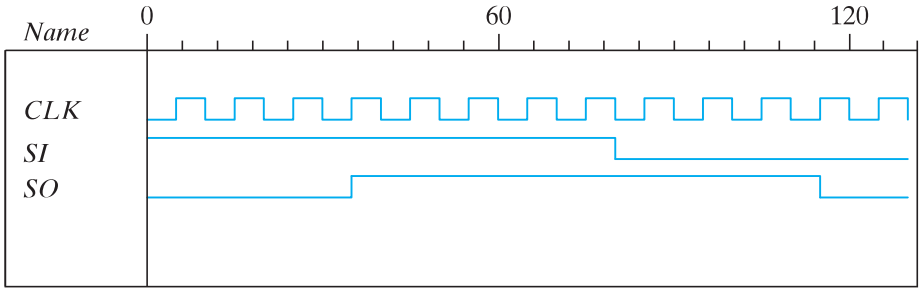
6.26 The clock generator has a period of 12.5 ns. Use a 2-bit counter to count four pulses.

6.28 $D_A = A \oplus B$

$D_B = AB' + C$

$D_C = A'B'C'$

6.34 The HDL description is available on the Companion Website. Simulations results for Problem 6.34 follow:



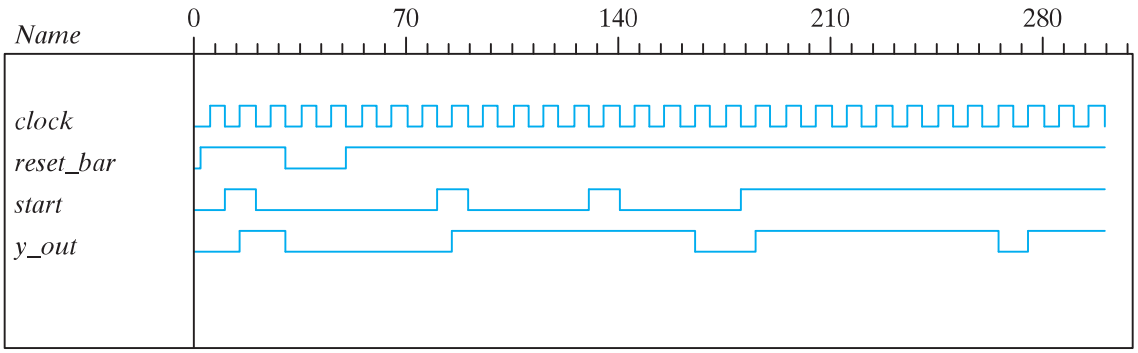
6.35 **(b)** The HDL description is available on the Companion Website.

6.37 The HDL description is available on the Companion Website.

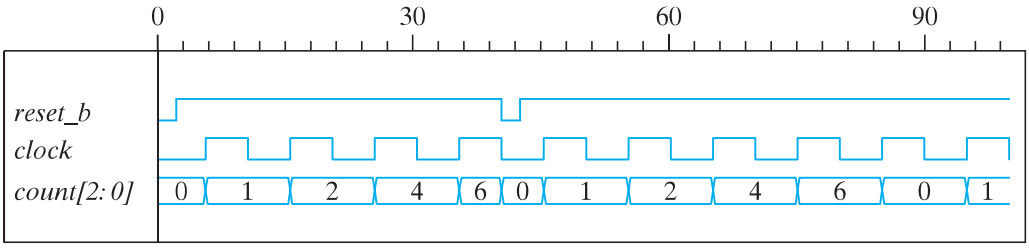
6.38 **(a)** The HDL description is available on the Companion Website.

6.42 Because A is a register variable, it retains whatever value has been assigned to it until a new value is assigned. Therefore, the statement $A <= A$ has the same effect as if the statement was omitted.

6.45 The HDL description is available on the Companion Website. Simulations results for Problem 6.45 follow:



6.50 **(b)** The HDL description is available on the Companion Website. Simulations results for Problem 6.50 follow:



CHAPTER 7

7.2 **(a)** 2^{13} **(b)** 2^{31} **(c)** 2^{26} **(d)** 2^{21}

7.3 Address: 1 0001 1011 = 011B (hex)

Data: 100 1011 1100 = 4BC (hex)

7.7 **(a)** 7 × 128 decoders, 256 AND gates **(b)** $x = 46; y = 112$

7.8 **(a)** 8 chips **(b)** 18; 15 **(c)** 3 × 8 decoder

7.10 0001 1011 1011 1

7.11 101 110 011 001 010

7.12 **(a)** 0101 1010; **(b)** 1100 0110; **(c)** 1111 0100

7.13 **(a)** 6 **(b)** 7 **(c)** 7

7.14 **(a)** 0101010

7.16 24 pins

7.20 Product terms: yz', xz', $x'y'z$, xy', $x'y$, z

7.25 $A = yz' + xz' + x'y'z$

$B = x'y' + yz + y'z'$

$C = A + xyz$

$D = z + x'y$

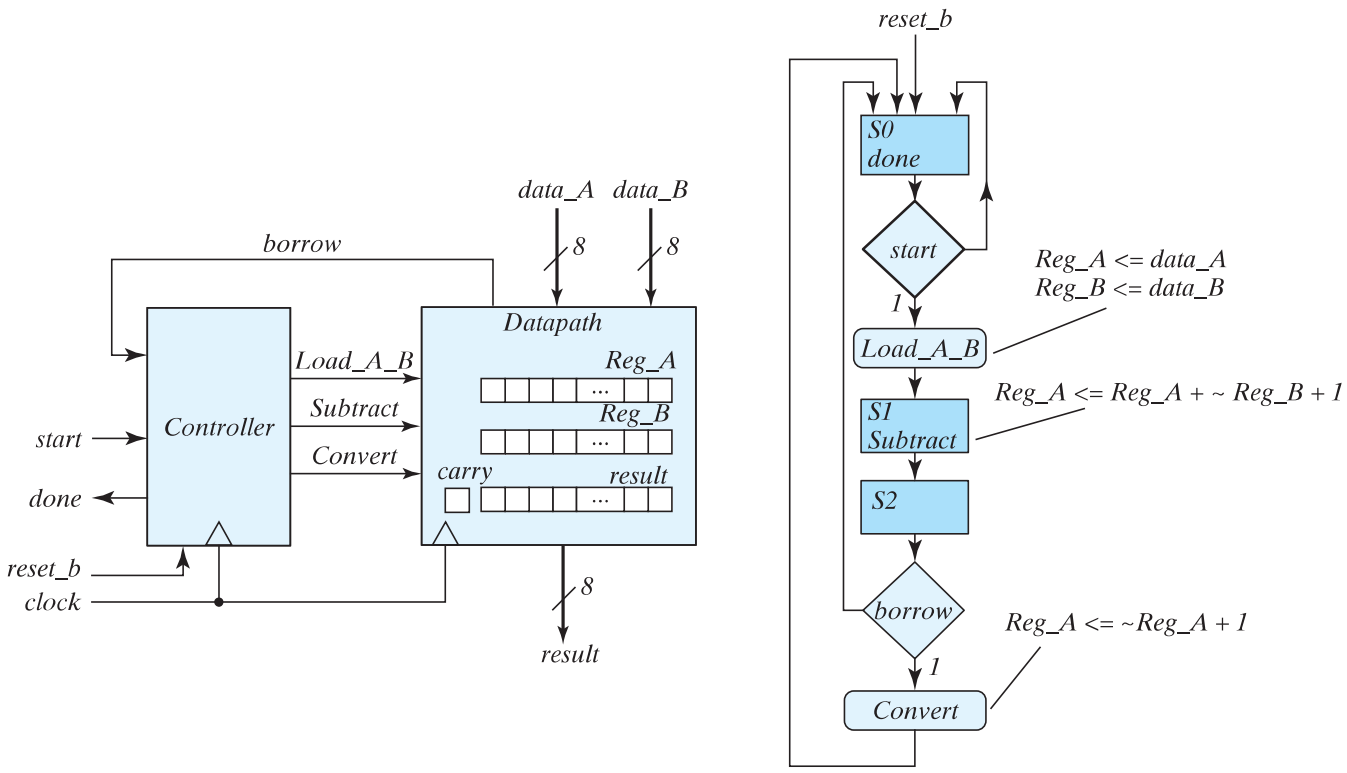
CHAPTER 8

8.1 **(a)** The transfer and increment occur concurrently, i.e., at the same clock edge. After the transfer, *R2* holds the contents that were in *R1* before the clock edge, and *R2* holds its previous value incremented by 1.
(b) Decrement the content of *R3* by one.
(c) If ($S_1 = 1$), transfer content of *R1* to *R0*. If ($S_1 = 0$ and $S_2 = 1$), transfer content of *R2* to *R0*.

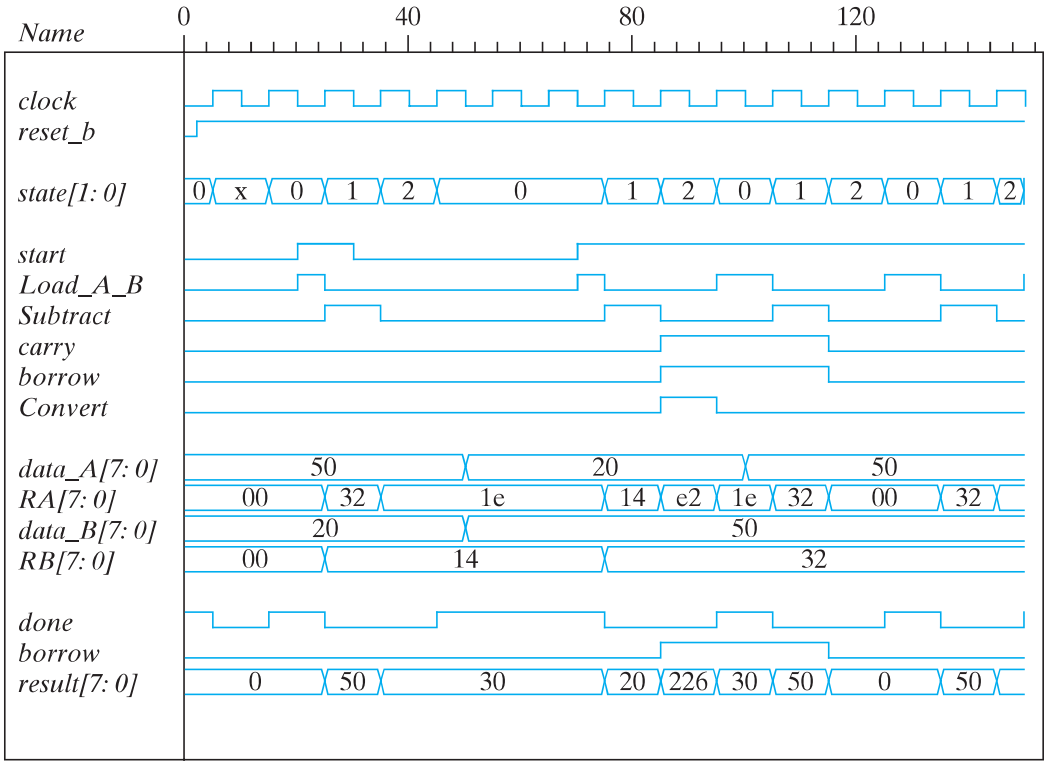
8.7 RTL notation:

S0: Initial state: if (start = 1) then (*RA* ← data_*A*, *RB* ← data_*B*, go to *S*1).
S1: {Carry, *RA*} ← *RA* + (2's complement of *RB*), go to *S*2.
S2: If (borrow = 0) go to *S0*. If (borrow = 1) then *RA* ← (2's complement of *RA*), go to *S*0.

Block diagram and ASMD chart:

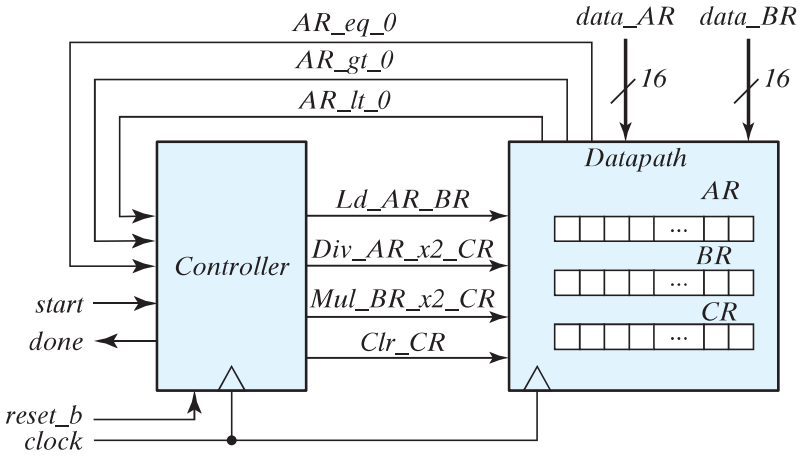


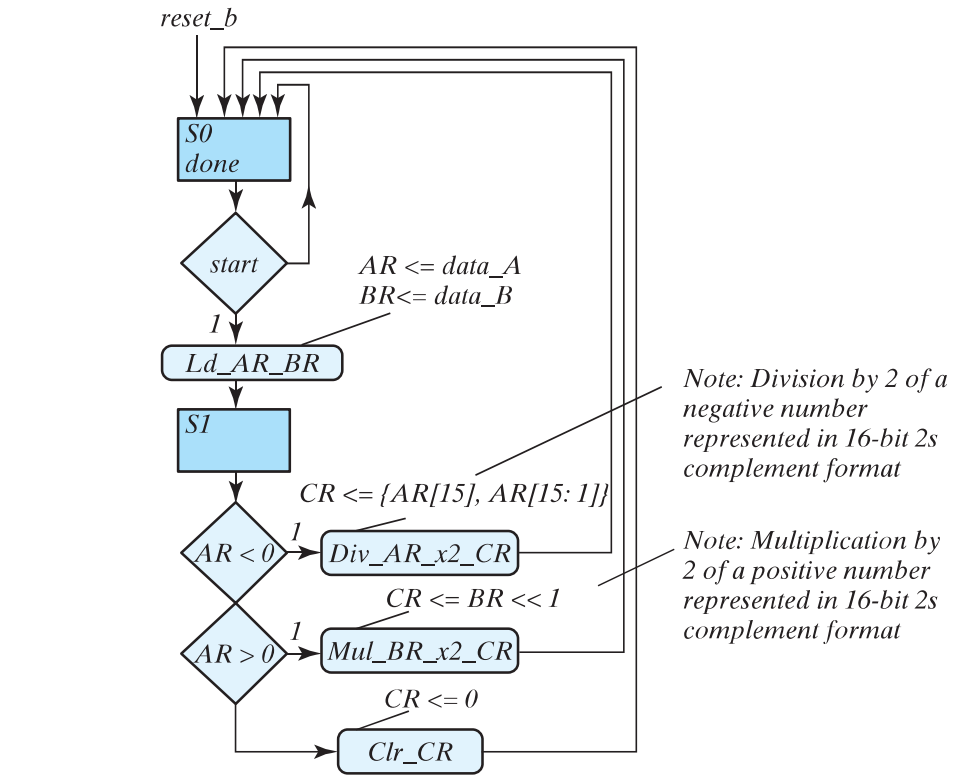
The HDL description is available on the Companion Website. Simulations results for Problems 8.7 follow:



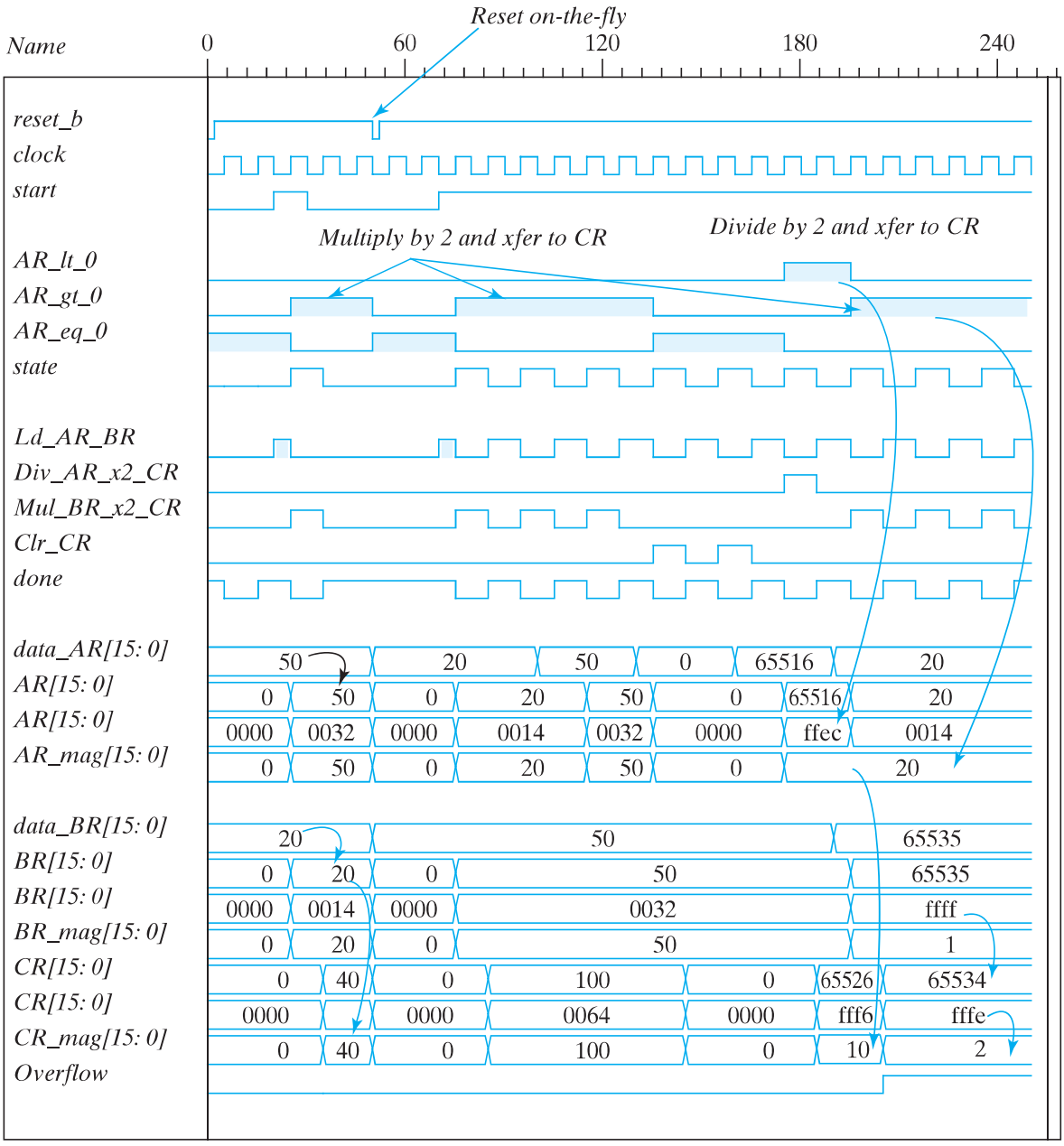
8.8 RTL notation:

S0: if (start = 1) $AR \leftarrow$ input data, $BR \leftarrow$ input data, go to *S1*.
S1: if (AR [15]) = 1(sign bit negative) then $CR \leftarrow AR$ (shifted right, sign extension).
else if (positive non-zero) then (Overflow $\leftarrow BR([15] \oplus [14])$, $CR \leftarrow BR$(shifted left)
else if ($AR = 0$) then ($CR \leftarrow 0$).





The HDL description is available on the Companion Website. Simulations results for Problem 8.8 follow:



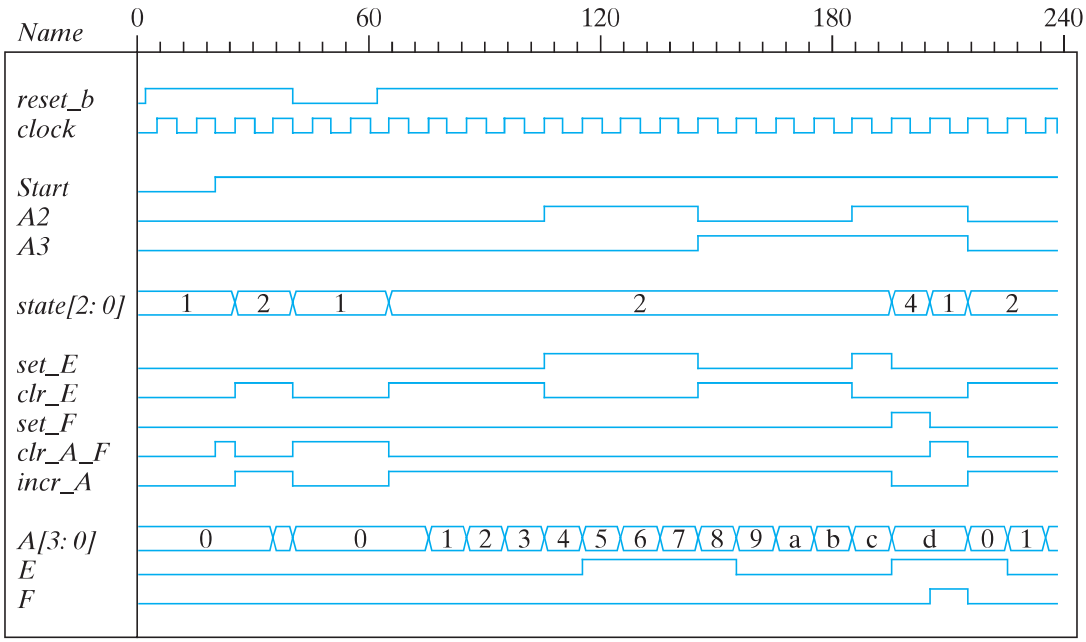
8.9 Design equations:

$$D_{S\_idle} = S\_2 + S\_idle\ Start'$$

$$D_{S\_1} = S\_idle\ Start + S\_1(A2\ A3)'$$

$$D_{S\_2} = A2\ A3\ S\_1$$

The HDL description is available on the Companion Website. Simulations results for Problem 8.9 follow:



8.11 $D_A = A'B + Ax$

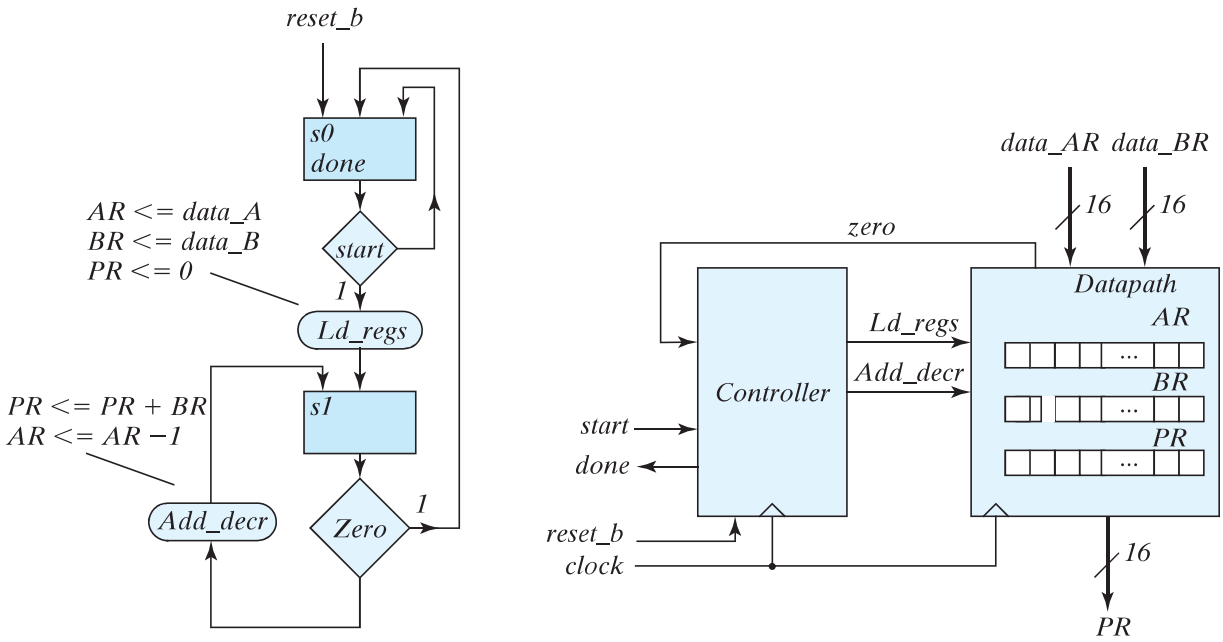
$D_B = A'B'x + A'By + xy$

8.16 RTL notation:

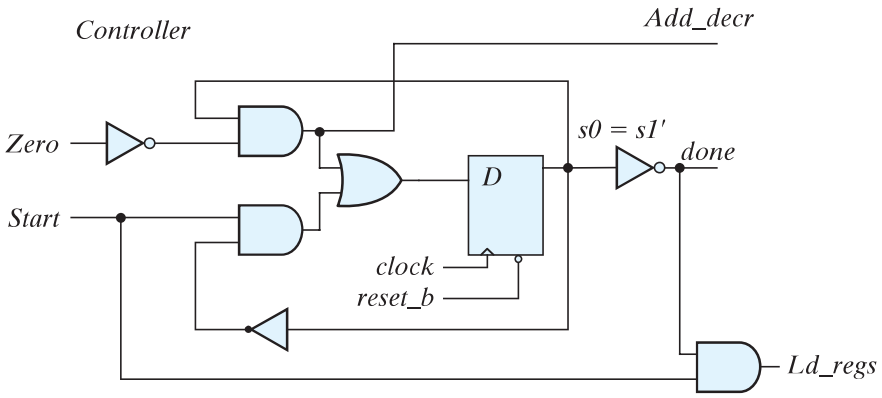
s0: (initial state) If *start* = 0 go back to state *s0*, If(*start* = 1) then $BR \leftarrow multiplicand$, $AR \leftarrow multiplier$, $PR \leftarrow 0$, go to *s*1.

s1: (check *AR* for Zero) *Zero* = 1 if *AR* = 0, if (*Zero* = 1) then go back to *s0* (*done*) If (*Zero* = 0) then go to *s1*, $PR \leftarrow PR + BR$, $AR \leftarrow AR - 1$.

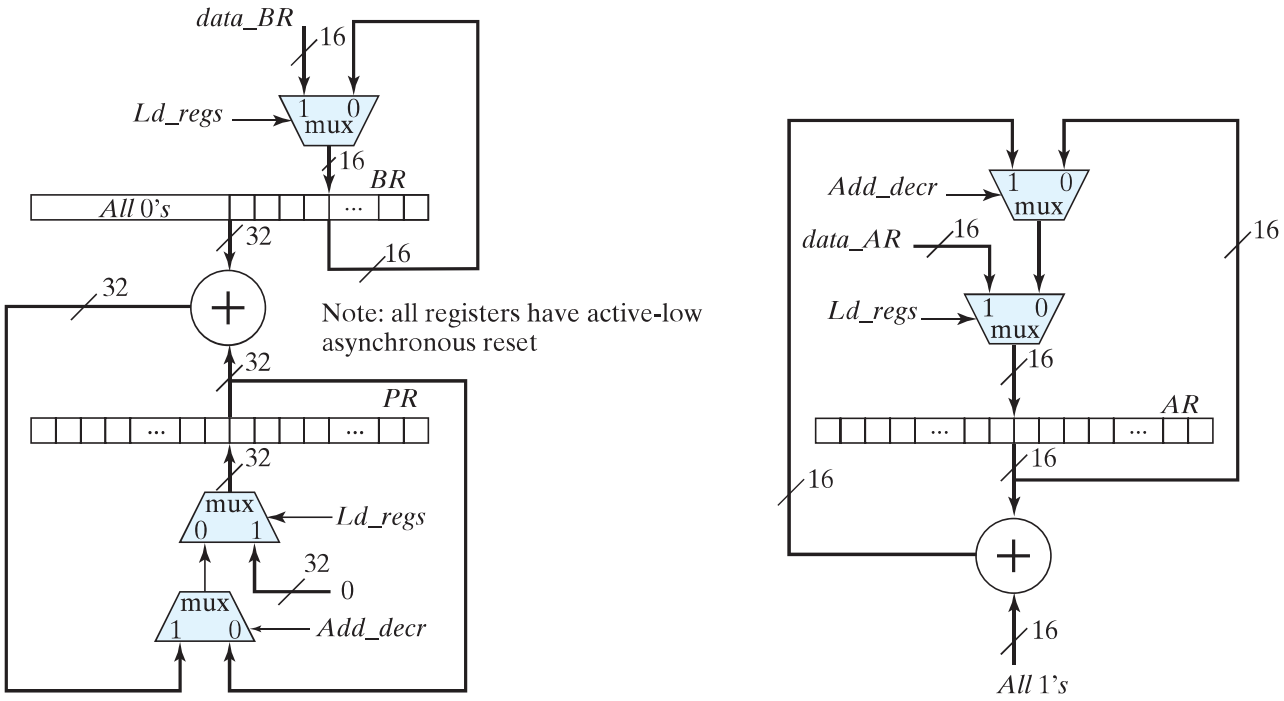
The internal architecture of the datapath consists of a double-width register to hold the product (PR), a register to hold the multiplier (AR), a register to hold the multiplicand (BR), a double-width parallel adder, and single-width parallel adder. The single-width adder is used to implement the operation of decrementing the multiplier unit. Adding a word consisting entirely of 1s to the multiplier accomplishes the 2's complelment subtraction of 1 from the multiplier. Figure 8.16 (a) below shows the ASMD chart, block diagram, and controller of othe circuit. Figure 8.16 (b) shows the internal architecture of the datapath. Figure 8.16 (c) shows the results of simulating the circuit.



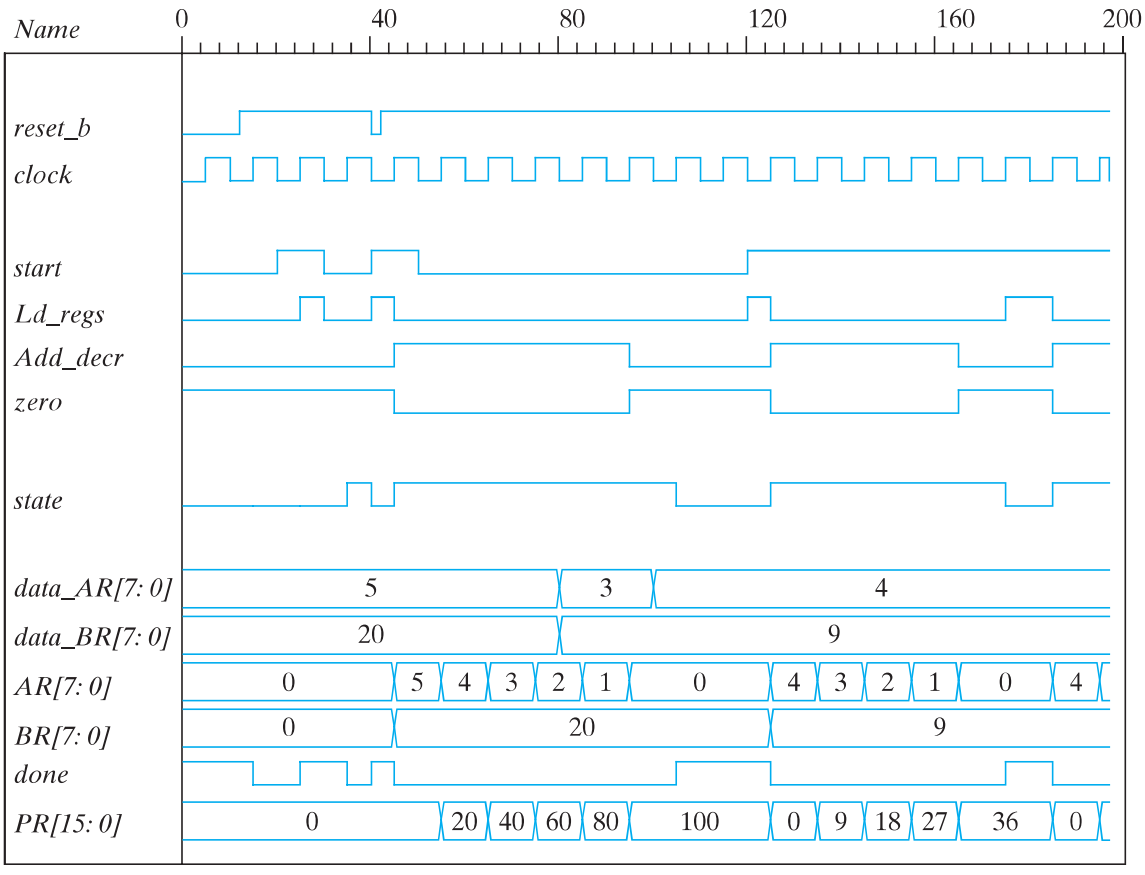
Note: Form Zero as the output of an OR gate whose inputs are the bits of the register AR.



(a) ASMD chart, block diagram, and controller



(b) Datapath



(c) Simulation results

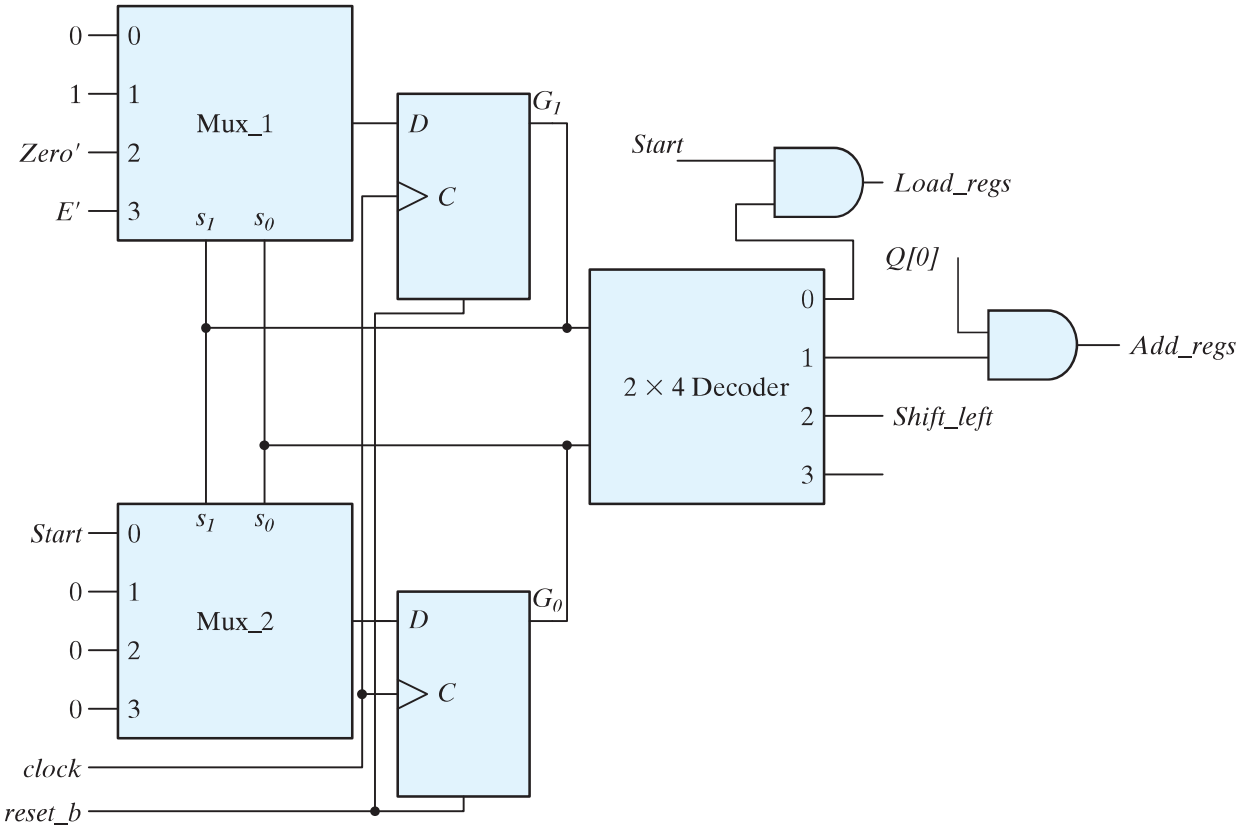
8.17 $(2^n - 1)(2^n - 1) < (2^{2n} - 1)$ for $n \geq 1$

8.18 **(a)** The maximum product size is 32 bits available in registers A and Q.
(b) P counter must have 5 bits to load 16 (binary 10000) initially.
(c) Z (zero) detection is generated with a 5-input NOR gate.

8.20 $2(n + 1)t$

8.21

State codes:	G1	G0
S_idle	0	0
S_add	0	1
S_shift	1	0
unused	0	0



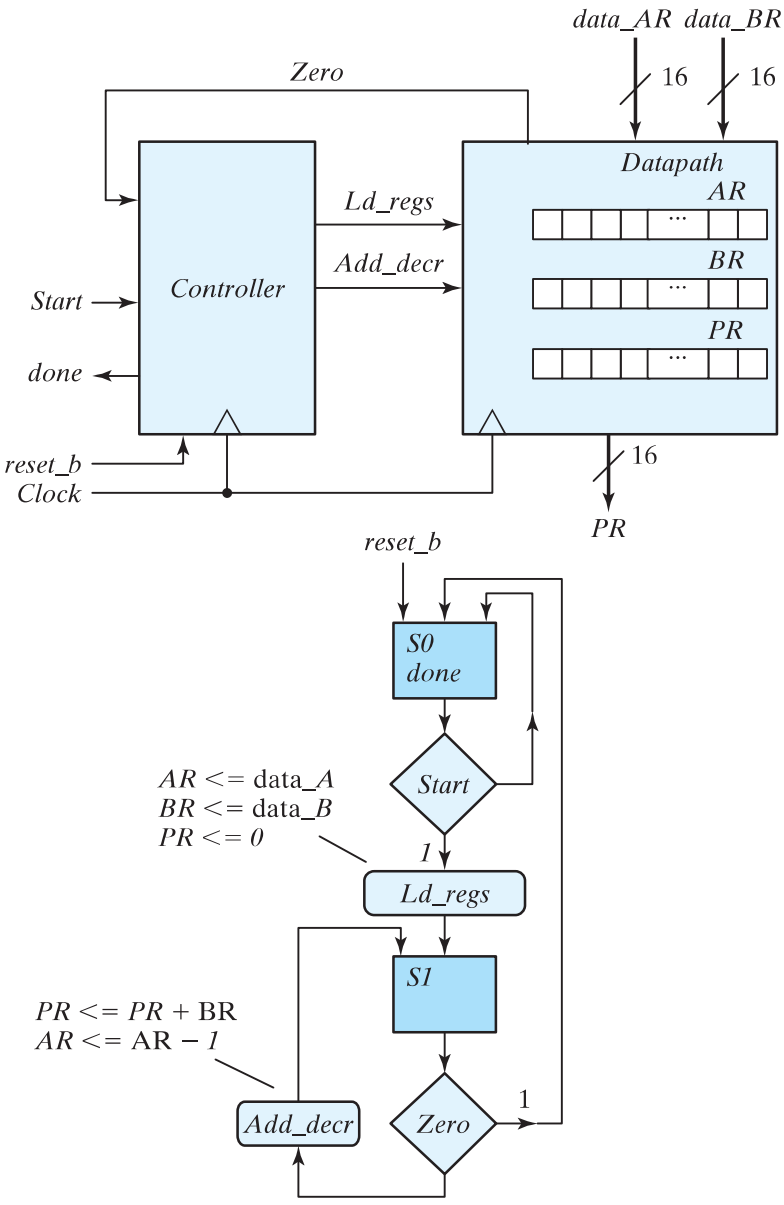
8.30 **(a)** $E = 1$ **(b)** $E = 0$

8.31 A = 0110, B = 0010, C = 0000.

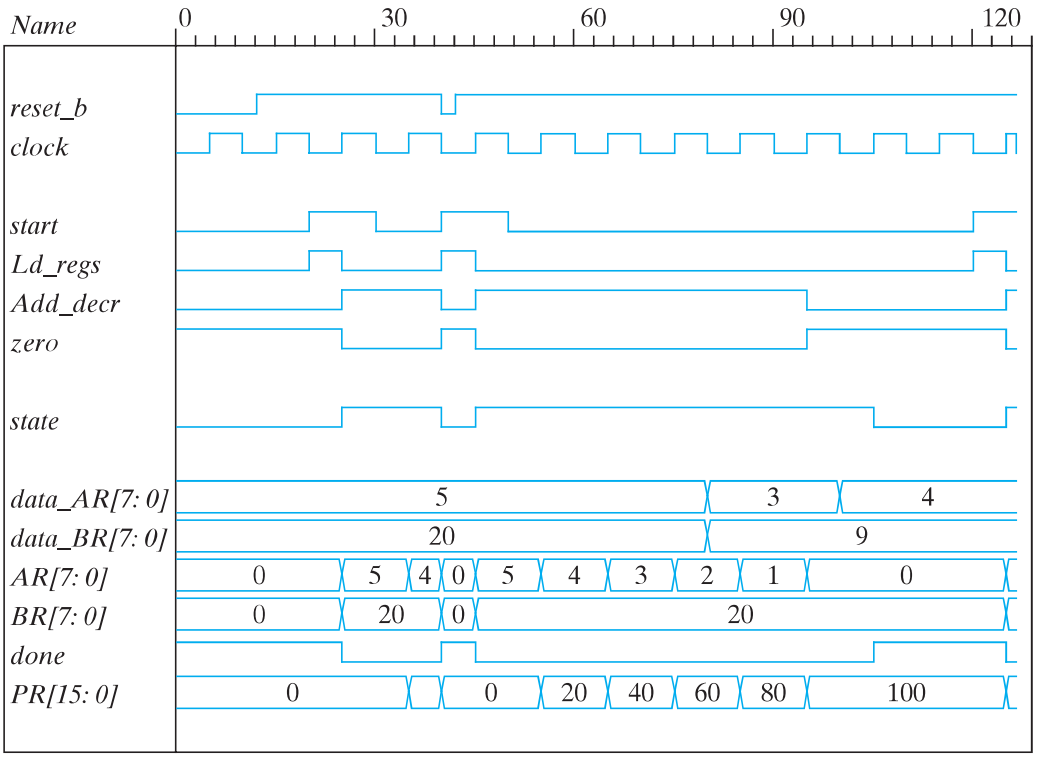
A*B = 1100	A | B = 0110	A && C = 0
A + B = 1000	A∧B = 0100	| A = 1
A − B = 0100	&A = 0	A < B = 0
~C = 1111	~|C = 1	A > B = 1
A & B = 0010	A || B = 1	A != B = 1

8.39

Block diagram and ASMD chart:



The HDL description is available on the Companion Website. Simulation results for Problem 8.39 follow:


Name
0
30
60
90
120
reset_b
clock
start
Ld_regs
Add_decr
zero
state
data_AR[7: 0]
5
3
4
data_BR[7: 0]
20
9
AR[7: 0]
0
5
4
0
5
4
3
2
1
0
BR[7: 0]
0
20
0
20
done
PR[15: 0]
0
0
20
40
60
80
100