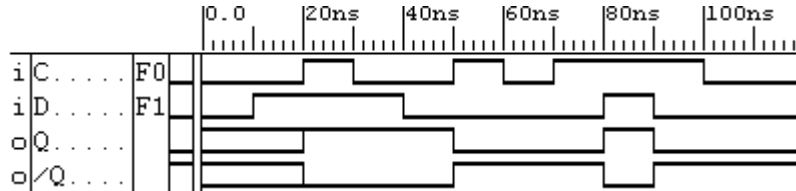


**Problem Solutions to Problems Marked With a \* in  
Logic Computer Design Fundamentals, Ed. 2**

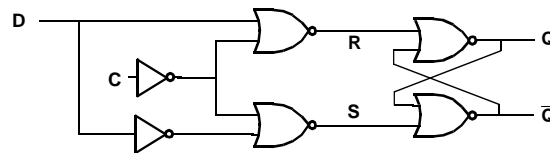
**CHAPTER 4**

© 2000 by Prentice-Hall, Inc.

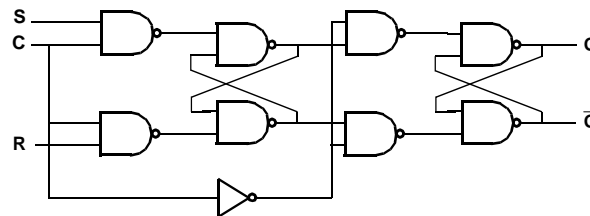
**4-3.** (All simulations performed using Xilinx Foundation Series software.)



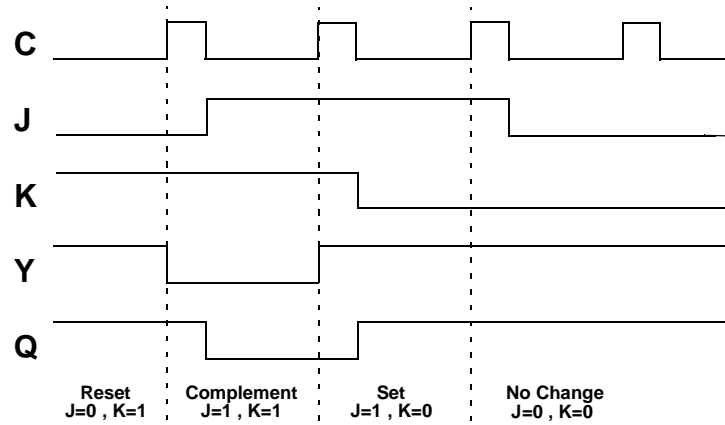
**4-4.**



**4-5.**



**4-6.**



**Problem Solutions – Chapter 4**

**4-10.**

J	K	Q(t)	Q(t+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

S	R	Q(t)	Q(t+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X

D	Q(t)	Q(t+1)
0	0	0
0	1	0
1	0	1
1	1	1

T	Q(t)	Q(t+1)
0	0	0
0	1	1
1	0	1
1	1	0

$Q(t+1) = D$        $Q(t+1) = T \oplus Q$

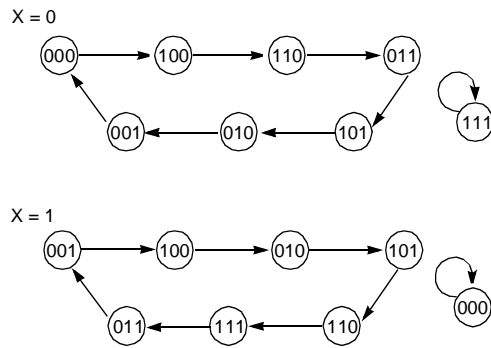
$J_A = B$        $K_A = B\bar{X}$   
 $J_B = \bar{X}$        $K_B = A\bar{X} + \bar{A}X$

$Q(t+1) = J\bar{Q} + \bar{K}Q$        $Q(t+1) = S + \bar{R}Q$

$A(t+1) = J_A\bar{A} + \bar{K}_A A$       =       $B\bar{A} + \bar{B}A + XA$   
 $B(t+1) = J_B\bar{B} + \bar{K}_B B$       =       $\bar{X}\bar{B} + ABX + \bar{A}B\bar{X}$

**4-12.**

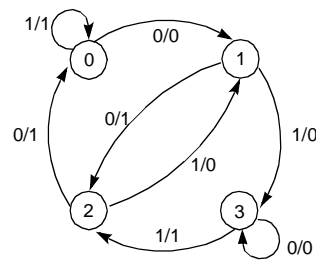
Present state			Input	Next state		
A	B	C	X	A	B	C
0	0	0	0	1	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	1	0	1
0	1	1	0	1	0	1
0	1	1	1	0	0	1
1	0	0	0	1	1	0
1	0	0	1	0	1	0
1	0	1	0	0	1	0
1	0	1	1	1	1	0
1	1	0	0	0	1	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	0	1	1



State diagram is the combination of the above two diagrams.

**4-17.**

Present state		Input	Next state		Output
A	B	X	A	B	Y
0	0	0	0	1	0
0	0	1	0	0	1
0	1	0	1	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	0	1	0
1	1	0	1	1	0
1	1	1	1	0	1



Format: X/Y

Problem Solutions – Chapter 4

4-19.

Present state		Input	Next state	
A	B	X	A	B
0	0	0	0	0
0	0	1	1	0
0	1	0	0	1
0	1	1	0	0
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	0	1

$D_A$

		1	
A	1	1	1
		X	

$D_A = A\bar{B} + AX + \bar{B}X$

$D_B$

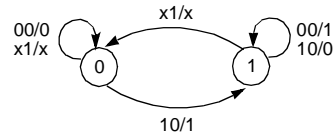
			1
A		1	1
		X	

$D_B = AX + B\bar{X}$

4-20.

Present state	Inputs		Next state	Output
$Q(t)$	X	Y	$Q(t+1)$	Z
0	0	0	0	0
0	0	1	0	X
0	1	0	1	1
0	1	1	0	X
1	0	0	1	1
1	0	1	0	X
1	1	0	1	0
1	1	1	0	X

Format: XY/Z (x = unspecified)



4-24.

Present state		Input	Next state		Output
A	B	X	A	B	Y
0	0	0	0	1	1
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	1	1	0
1	0	0	1	0	0
1	0	1	0	0	0
1	1	0	1	0	0
1	1	1	1	1	0

$D_A$

		1	
A	1	1	1
		X	

$D_A = A\bar{X} + BX$

$D_B$

		1	1
A		1	
		X	

$D_B = BX + \bar{A}\bar{X}$

Y

		1	
A	1	1	
		X	

$Y = \bar{A}\bar{B}$

4-25.

Present state		Input		Next state
A		J	K	A
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
1	0	0	0	1
1	0	0	1	0
1	1	1	0	1
1	1	1	1	0

$D_A$

		1	1
A	1		1
		K	

$D_A = \bar{A}J + AK$

**Problem Solutions – Chapter 4**

**4-30.**

Present state		Input	Next state		FF Inputs			
A	B	X	A	B	J <sub>A</sub>	K <sub>A</sub>	J <sub>B</sub>	K <sub>B</sub>
0	0	0	0	1	0	X	1	X
0	0	1	0	0	0	X	0	X
0	1	0	0	1	0	X	X	0
0	1	1	1	1	1	X	X	0
1	0	0	1	0	X	1	0	X
1	0	1	0	0	X	1	0	X
1	1	0	1	0	X	0	X	1
1	1	1	1	1	X	0	X	0

$$J_A = BX$$

$$K_A = B X + \bar{B} \bar{X}$$

$$J_B = \bar{A} \bar{X}$$

$$K_B = A \bar{X}$$

**4-33.**

Present state		Inputs		Next state		FF Inputs			
A	B	E	X	A	B	J <sub>A</sub>	K <sub>A</sub>	J <sub>B</sub>	K <sub>B</sub>
0	0	0	0	0	0	0	X	0	X
0	0	0	1	0	0	0	X	0	X
0	0	1	0	1	1	1	X	1	X
0	0	1	1	0	1	0	X	1	X
0	1	0	0	0	1	0	X	X	0
0	1	0	1	0	1	0	X	X	0
0	1	1	0	0	0	0	X	X	1
0	1	1	1	1	0	1	X	X	1
1	0	0	0	1	0	X	0	0	X
1	0	0	1	1	0	X	0	0	X
1	0	1	0	0	1	X	1	1	X
1	0	1	1	1	1	X	0	1	X
1	1	0	0	1	1	X	0	X	0
1	1	0	1	1	1	X	0	X	0
1	1	1	0	1	0	X	0	X	1
1	1	1	1	0	0	X	1	X	1

$$J_A = E(BX + \bar{B} \bar{X})$$

$$K_A = E(BX + \bar{B} \bar{X})$$

$$J_B = E$$

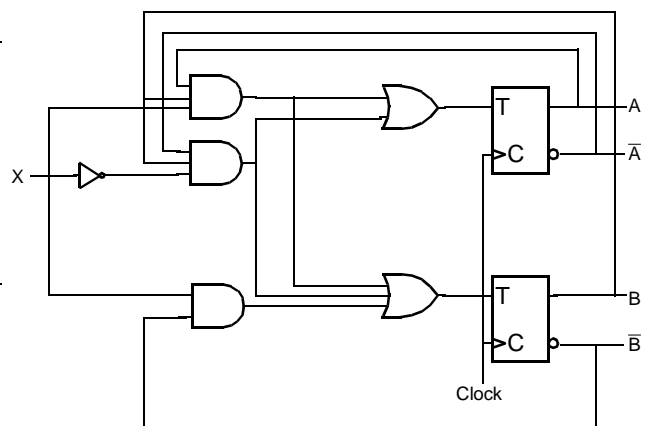
$$K_B = E$$

**4-36.**

Present state		Input	Next state		FF Inputs	
A	B	X	A	B	T <sub>A</sub>	T <sub>B</sub>
0	0	0	0	0	0	0
0	0	1	0	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	0	0	0
1	0	1	1	1	0	1
1	1	0	1	1	0	0
1	1	1	0	0	1	1

$$T_A = ABX + \bar{A}B\bar{X}$$

$$T_B = ABX + \bar{A}B\bar{X} + \bar{B}X$$



**Problem Solutions – Chapter 4**

**4-37**

```
library IEEE;
use IEEE.std_logic_1164.all;
```

```
entity mux_4to1 is
  port (
    S: in STD_LOGIC_VECTOR (1 downto 0);
    D: in STD_LOGIC_VECTOR (3 downto
0);
    Y: out STD_LOGIC
  );
end mux_4to1;
```

-- (continued in the next column)

```
architecture mux_4to1_arch of mux_4to1 is
begin
```

```
  process (S, D)
  begin
    case S is
      when "00" => Y <= D(0);
      when "01" => Y <= D(1);
      when "10" => Y <= D(2);
      when "11" => Y <= D(3);
      when others => null;
    end case;
```

```
  end process;
end mux_4to1_arch;
```

**4-40.**

```
library IEEE;
use IEEE.std_logic_1164.all;
entity jkff is
  port (
    J,K,CLK: in STD_LOGIC;
    Q: out STD_LOGIC
  );
end jkff;
```

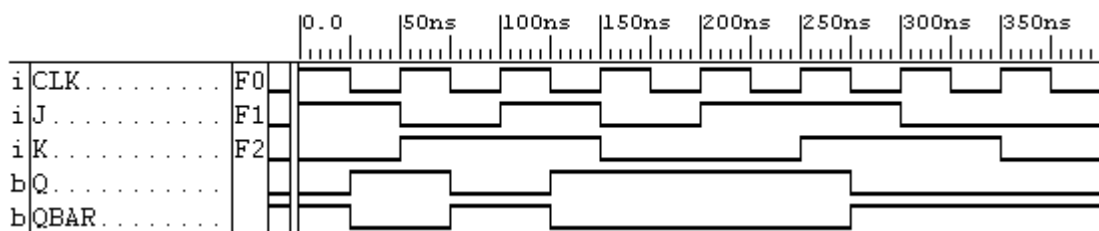
```
architecture jkff_arch of jkff is
  signal q_out: std_logic;
begin
```

```
  state_register: process (CLK)
  begin
    if CLK'event and CLK='0' then --CLK falling edge
```

-- (continued in the next column)

```
    case J is
      when '0' =>
        if K = '1' then
          q_out <= '0';
        end if;
      when '1' =>
        if K = '0' then
          q_out <= '1';
        else
          q_out <= not q_out;
        end if;
      when others => null;
    end case;
  end if;
end process;
```

```
  Q <= q_out;
end jkff_arch;
```



**4-45.**

```
module problem_4_45 (S, D, Y);
```

```
  input [1:0] S;
  input [3:0] D;
  output Y;
  reg Y;
```

// (continued in the next column)

```
  always @(S or D)
  begin
```

```
    if (S == 2'b00) Y <= D[0];
    else if (S == 2'b01) Y <= D[1];
    else if (S == 2'b10) Y <= D[2];
    else Y <= D[3];
```

```
  end
endmodule
```

---

**Problem Solutions – Chapter 4**

---

**4-47.**

```
module JK_FF (J, K, CLK, Q);
    always @(negedge CLK)
        case (J)
            0'b0: Q <= K ? 0: Q;
            1'b1: Q <= K ? ~Q: 1;
        endcase
endmodule

// (continued in the next column)
```

