Design & Modeling of Digital Systems

HW# 2 Solution

Q1.

(i)

```
ENTITY SHIFTREG4 IS
      PORT (      SEL: IN BIT_VECTOR(1 DOWNTO 0);
            SI, CLK, RESET: IN BIT;
            S: BUFFER BIT_VECTOR(3 DOWNTO 0):= "0000");
END SHIFTREG4;
```

(ii)

```
ARCHITECTURE BEHAV OF SHIFTREG4 IS
SIGNAL S0, S3: BIT;
BEGIN
PROCESS (CLK)
      BEGIN
      IF((CLK = '1' AND CLK'EVENT)) THEN
            IF RESET = '1' THEN
            S <= "0000";
            -- SHIFT LEFT
            ELSE IF (SEL(1) = '0' AND SEL(0) = '0') THEN
                  S <= S(2 DOWNTO 0) & SI;
            -- SHIFT RIGHT
            ELSE IF (SEL(1) = '0' AND SEL(0) = '1') THEN
                  S <= SI & S(3 DOWNTO 1);
            -- ROTATE LEFT
            ELSE IF (SEL(1) = '1' AND SEL(0) = '0') THEN
                  S <= S(2 DOWNTO 0) & S(3);
            -- ROTATE RIGHT
            ELSE
                  S <= S(0) & S(3 DOWNTO 1);
            END IF;
            END IF;
            END IF;
            END IF;
      END IF;
END PROCESS;
END BEHAV;
```

(iii)

```
ARCHITECTURE STRUCT OF SHIFTREG4 IS
COMPONENT MUX2X1
      PORT (SEL: IN BIT; S1, S0: IN BIT; RES: OUT BIT := '0');
END COMPONENT;
COMPONENT MUX4X1
      PORT (SEL: IN BIT_VECTOR (1 DOWNTO 0); S3, S2, S1, S0: IN BIT;
RES: OUT BIT := '0');
END COMPONENT;
COMPONENT FAR
      PORT (CLK, RESET, D: IN BIT; Q: BUFFER BIT := '0');
END COMPONENT;

SIGNAL RES3, RES2, RES1, RES0: BIT;
BEGIN
-- MUXES
U00: MUX4X1 PORT MAP (SEL, S(0), S(2), SI, S(2), RES3);
U01: MUX2X1 PORT MAP (SEL(0), S(3), S(1), RES2);
U02: MUX2X1 PORT MAP (SEL(0), S(2), S(0), RES1);
U03: MUX4X1 PORT MAP (SEL, S(1), S(3), S(1), SI, RES0);
-- FLIP FLOPS
U04: FAR PORT MAP (CLK, RESET, RES3, S(3));
U05: FAR PORT MAP (CLK, RESET, RES2, S(2));
U06: FAR PORT MAP (CLK, RESET, RES1, S(1));
U07: FAR PORT MAP (CLK, RESET, RES0, S(0));
-- FINISH
END STRUCT;

-------------------------------------------------

ENTITY MUX2X1 is
port (      SEL: IN BIT;
      S1, S0: IN BIT;
      RES: OUT BIT:='0');
END MUX2X1;

-------------------------------------------------

ARCHITECTURE BEHAV OF MUX2X1 is
BEGIN
PROCESS (SEL, S0, S1)
      BEGIN
            IF SEL = '0' THEN
                  RES <= S0;
            ELSE
                  RES <= S1;
            END IF;
END PROCESS;
END BEHAV;

-------------------------------------------------
ENTITY MUX4X1 is
port (      SEL: IN BIT_VECTOR(1 DOWNTO 0);
      S3, S2, S1, S0: IN BIT;
      RES: OUT BIT:='0');
```

```
END MUX4X1;

-------------------------------------------------

ARCHITECTURE BEHAV OF MUX4X1 is
BEGIN
PROCESS (SEL(1), SEL(0), S0, S1, S2, S3)
      BEGIN
            IF (SEL(1) = '0' AND SEL(0)='0') THEN
                  RES <= S0;
            ELSE IF (SEL(1) = '0' AND SEL(0)='1') THEN
                  RES <= S1;
            ELSE IF (SEL(1) = '1' AND SEL(0)='0') THEN
                  RES <= S2;
            ELSE
                  RES <= S3;
            END IF;
            END IF;
            END IF;
END PROCESS;
END BEHAV;

-------------------------------------------------



ENTITY FAR is
port (      CLK, RESET: IN BIT;
       D: IN BIT;
       Q: BUFFER BIT:='0');
END FAR;

-------------------------------------------------

ARCHITECTURE Arch1 OF FAR is
BEGIN
PROCESS (CLK)
      BEGIN
      IF((CLK = '1' AND CLK'EVENT)) THEN
            IF RESET = '1' THEN
                  Q <= '0' AFTER 8 ns;
            ELSE
                  Q <= D AFTER 8 ns;
            END IF;
      END IF;
END PROCESS;
END Arch1;
```

Q2.

	0	**δ**	**2δ**	**5 ns**	**10 ns**	**10 + δ**	**15 ns**
A	0	0	0	0	1	1	1
	('1' , 10 ns)	('1' , 10 ns)	('1' , 10 ns)	('1' , 5 ns)	('0' , 10 ns)	('0' , 10 ns)	('0' , 5 ns)
B	0	Z	Z	Z	Z	Z	Z
	('Z', δ) ('0' , 25 ns) ('0' , 35 ns)	('0' , 25 ns) ('0' , 35 ns)	('0' , 25 ns) ('0' , 35 ns)	('0' , 20 ns) ('0' , 30 ns)	('0' , 15 ns) ('0' , 25 ns) ('Z' , δ) ('1' , 25 ns) ('0' , 35 ns)	('1' , 25 ns) ('0' , 35 ns)	('1' , 20 ns) ('0' , 30 ns)
C	0	1	1	0	0	1	1
	('1' , δ) ('0' , 5 ns) ('0' , 20 ns)	('1' , δ) ('0' , 5 ns) ('Z' , 20 ns)	('0' , 5 ns) ('Z' , 20 ns)	('Z' , 15 ns)	('Z' , 10 ns) ('1' , δ) ('1' , 5 ns) ('Z' , 20 ns)	('1' , 5 ns) ('Z' , 20 ns)	('Z' , 15 ns)

	15 ns	**20 ns**	**20 + δ**	**25 ns**	**30 ns**	**30 + δ**	**35 ns**
A	1	0	0	0	1	1	1
	('0' , 5 ns)	('1' , 10 ns)	('1' , 10 ns)	('1' , 5 ns)	('0' , 10 ns)	('0' , 10 ns)	('0' , 5 ns)
B	Z	Z	Z	Z	Z	Z	Z
	('1' , 20 ns) ('0' , 30 ns)	('1' , 15 ns) ('0' , 25 ns) ('Z', δ) ('0' , 25 ns) ('0' , 35 ns)	('0' , 25 ns) ('0' , 35 ns)	('0' , 20 ns) ('0' , 30 ns)	('0' , 15 ns) ('0' , 25 ns) ('Z', δ) ('1' , 25 ns) ('0' , 35 ns)	('1' , 25 ns) ('0' , 35 ns)	('1' , 20 ns) ('0' , 30 ns)
C	1	1	1	0	0	1	1

	('Z' , 15 ns)	('Z' , 10 ns) ('1' , δ) ('0' , 5 ns) ('Z' , 20 ns)	('0' , 5 ns) ('Z' , 20 ns)	('Z' , 15 ns)	('Z' , 10 ns) ('1' , δ) ('1' , 5 ns) ('Z' , 20 ns)	('1' , 5 ns) ('Z' , 20 ns)	('Z' , 20 ns)

	35 ns	**40 ns**	**40 + δ**	**45 ns**	**60 ns**	**65 ns**	**65 + δ**	**70 ns**
A	1	0	0	0	0	0	0	0
	('0' , 5 ns)	-----	-----	-----	-----	-----	-----	-----
B	Z	Z	Z	Z	Z	0	0	0
	('1' , 20 ns) ('0' , 30 ns)	('1' , 15 ns) ('0' , 25 ns) ('Z', δ) ('0' , 25 ns) ('0' , 35 ns)	('0' , 25 ns) ('0' , 35 ns)	('0' , 20 ns) ('0' , 30 ns)	('0' , 5 ns) ('0' , 15 ns)	('0' , 10 ns)	('0' , 10 ns)	('0' , 5 ns)
C	1	1	1	0	Z	Z	1	0
	('Z' , 20 ns)	('Z' , 15 ns) ('1' , δ) ('0' , 5 ns) ('Z' , 20 ns)	('0' , 5 ns) ('Z' , 20 ns)	('Z' , 15 ns)	-----	('1' , δ) ('0' , 5 ns) ('0' , 20 ns)	('0' , 5 ns) ('0' , 20 ns)	('0' , 15 ns)

	70 ns	**75 ns**	**75+ δ**	**80**	**95**
A	0	0	0	0	0
	-----	------	-----	------	-----

B	0	0	0	0	0
	('0' , 5 ns)	-----	-----	------	-----
C	0	0	1	0	0
	('0' , 15 ns)	('0' , 10 ns) ('1' , δ) ('0' , 5 ns) ('0' , 20 ns)	('0' , 5 ns) ('0' , 20 ns	('0' , 15 ns)	----