Homework 7

Athazahra Nabila Ruby - KKI - 2306173113

1. a. Register is sequential logic which can be defined by a state table. It is frequently used to perform simple data storage and data movement and processing operations.

   b. A register is composed of multiple flip-flops to store binary data. Flip-flops are used due to their ability to store binary information, operate synchronously with clock signals, facilitate sequential logic circuits, enable controlled data movements, and handle parallel data.

   c. A microoperation is an elementary operation performed on the information stored in one or more registers

   d. I. Transfer = move data from one register to another
      ex. = $R_1 \leftarrow R_2$
      II. Arithmetic = perform arithmetic operation on data in register
      ex. = $R_1 \leftarrow R_1 + R_2$
      III. Logic = manipulate data or use bitwise logical operations
      ex. = $R_1 \leftarrow R_1 \wedge R_2$
      IV. Shift = shift data in registers
      ex. $R_1 \leftarrow$ cir R_2 → circular shift right

2. $R_1 = 0100\ 0101$
   $R_2 = 1101\ 0111$

   a. $R_0 \leftarrow R_1 + \overline{R_2} + 1$ = $R_1 - R_2$

   1 2 1 1 1 2
   0 ✗ ✗ ✗ ✗ ✗ ✗ 2
   1 0100 0101
   1101 0111 −
   $R_0 = 0110\ 1110$

   b. $R_0 \leftarrow \overline{R_1 \wedge R_2}$
   $R_1 \wedge R_2 = 0100\ 0101$
   complement
   $R_0 = 1011\ 1010$

   c. $R_0 \leftarrow$ sl R_1 = $1000\ 1010$ → shift left
   $R_1 = 0100\ 0101$
   $R_0 = 1000\ 1010$ → zero fill

   d. $R_0 \leftarrow \overline{R_1} \oplus \overline{R_2}$
   $\overline{R_1} = 1011\ 1010$
   $\overline{R_2} = 0010\ 1000$ ⊕
   $R_0 = 1001\ 0010$

   • 0 ⊕ 0 = 0 ; 0 ⊕ 1 = 1
   • 1 ⊕ 1 = 0 • 1 ⊕ 0 = 1

   e. $R_0 \leftarrow$ asr R_2
   $R_2 = 1101\ 0111$
   $R_0 = 1110\ 1011$

Athazahra Nabila Ruby - KK1 - 2306173113

3. a. $R_A \leftarrow \overline{R_A} + \overline{R_C}$

$\overline{R_A}$ = 0101 1010

$\overline{R_C}$ = 1001 0110 +

$\overline{R_A} + \overline{R_C}$ = 1111 0000

Current : R_A = 1111 0000

R_B = 1100 1100 } doesn't change

R_C = 0110 1001

b. $R_B \leftarrow$ cir $\overline{R_B}$ (circular shift right)

$\overline{R_B}$ = 0011 0011

Cir $\overline{R_B}$ = 1001 1001

Current : R_B = 1001 1001

R_A = 1111 0000 } doesn't change

R_C = 0110 1001

c. $R_C \leftarrow R_B \oplus R_C$

R_B = 1001 1001

R_C = 0110 1001 $\oplus$

$R_B \oplus R_C$ = 1111 0000

·) 0 ⊕ 0 = 0 ·) 0 ⊕ 1 = 1

·) 1 ⊕ 1 = 0 ·) 1 ⊕ 0 = 1

Current : R_C = 1111 0000

R_A = 1111 0000 } doesn't change

R_B = 1001 1001

d. $R_A \leftarrow R_A + \overline{R_C} + 1$ ($R_A - R_C$)

R_A = 1111 0000

R_C = 1111 0000 −

$R_A - R_C$ = 0000 0000

Current : R_A = 0000 0000

R_B = 1001 1001 } doesn't change

R_C = 1111 0000

e. $R_B \leftarrow \overline{R_B \vee R_C}$

R_B = 1001 1001

$\overline{R_C}$ = 0000 1111 ∨

·) 0 ∨ 0 = 0 ·) 0 ∨ 1 = 1

·) 1 ∨ 1 = 1 ·) 1 ∨ 0 = 1

$R_B \vee R_C$ = 1001 1111 complement

$\overline{R_B \vee R_C}$ = 0110 0000

Current : R_B = 0110 0000

R_A = 0000 0000 } doesn't change

R_C = 1111 0000

f. $R_C \leftarrow \overline{\overline{R_B} \oplus \overline{R_C}}$

$\overline{R_B}$ = 1001 1111

$\overline{R_C}$ = 0000 1111 ⊕

$\overline{R_B} \oplus \overline{R_C}$ = 1001 0000 complement

$\overline{\overline{R_B} \oplus \overline{R_C}}$ = 0110 1111

Current : R_C = 0110 1111

R_A = 0000 0000 } doesn't change

R_B = 0110 0000

4. a. $C_x \cdot \overline{C_y}$: $A \leftarrow \overline{A} \vee B$

$C_x \quad C_y$: $A \leftarrow \overline{B}$

$\overline{C_x} \cdot \overline{C_y}$: $A \leftarrow A \oplus B$

$\overline{C_x} \cdot C_y$: $A \leftarrow \overline{A \wedge B}$

b.

Control input		Present State	Input	next state
C_x	C_y	$A(t)$	$B(t)$	$A(t+1)$
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0