Course Code : BCSL-022

Course Title : Assembly Language Programming Lab

Assignment Number : BCA(2)/BCSL022/Assign/2022-23

Maximum Marks : 50

Weightage : 25%

Last Dates for Submission : 31st October, 2022 (For July, 2022 Session)

: 15th April, 2023 (For January, 2023 Session)

Q1: Design a two-bit down counter circuit that does reverse counting. It should have states 11, 10, 01, and 00. The initial state of the counter may be assumed to be 11. The counter will be in following successive states: 11, 10, 01, 00,11, 10, 01, 00, 11,... Use J-K flip flop to design the circuit. You must design the circuit using state transition diagram and Karnaugh's maps.

Ans.

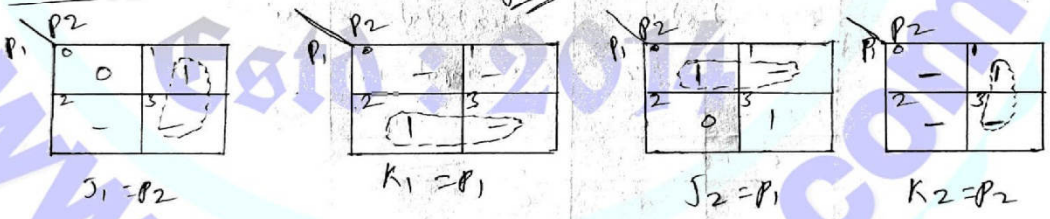
Ans.1

State Transition

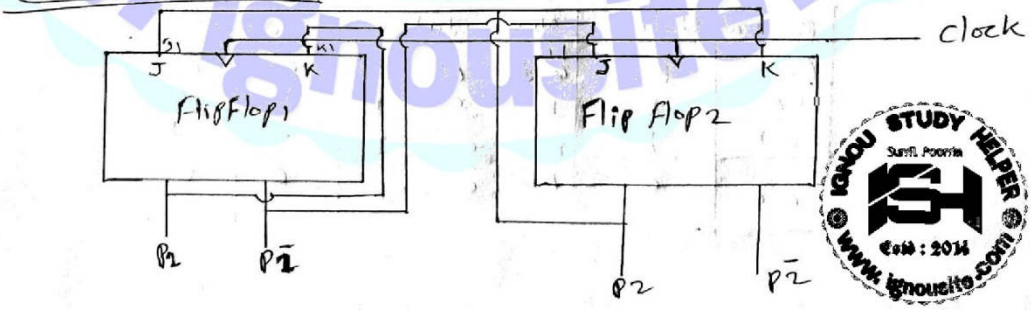
Present state		Next state		Flip-flop input			
P_1	P_2	P_1+	P_2+	J_1	K_1	J_2	K_2
1	0	0	0	–	1	0	–
0	1	0	1	1	–	–	1
0	0	1	0	0	–	1	–

P	P+	J	K
1	1	–	0
1	0	–	1
0	1	1	–
0	0	0	–

Karnaugh's map



J-K flip flop circuit:-



$J_1 = P_2$ $\quad J_2 = P_1$

$K_1 = P_1$ $\quad K_2 = P_2$

output

$R_1 = P_1$ $\quad R_2 = P_2$

$P^+ = J\bar{P} + \bar{K}P$

$P^+ = J_1\bar{P} + \bar{K}_1P_1$ $\quad P_2 = J_2^+\bar{P_2} + \bar{K}_2P_2$

$P_1^+ = P_2\bar{P_1} + \bar{P_1}P_1$ $\quad P_2^+ = P_1\bar{P_2} + \bar{P_2}P_2$

$P_1^+ = P_2P_1$ $\quad P_2^+ = P_1\bar{P_2}$

State Transition table Ignou study helper

present state		Next state		output state	
P_1	P_2	00 input 10		R_1	R_2
1	0	00	01	1	0
0	1	00	01	0	1
0	0	00	01	0	0

Present State		Next State			output	
P_1	P_2	10	00	01	R_1	R_2
10	A	A	C	B	1	0
01	B	A	C	B	0	1
00	C	A	C	B	0	0

Q2: Write and run following programs using 8086 assembly language.

(a) Write and run an 8086-assembly language program that accepts two input digits from the keyboard and converts it into a two-digit packed BCD number. This resultant number should be stored in a byte location in the memory. For example, if you input digit '3' and '5' then it will be converted to packed BCD number 35, which is 001101012. This result should be stored in a byte memory location.

Ans.

Steps:

1. Load the BCD number from the memory location (35FH, arbitrary choice) into the accumulator
2. Temporarily store the accumulator's value in B
3. Obtain BCD_2 by ANDing the accumulator with 0FH and store it in C
4. Restore the original value of the accumulator by moving the value in B to A. AND the accumulator with F0H
5. If the value in the accumulator equals 0, then BCD_2 is the final answer and store it in the memory location, 2020H (arbitrary)
6. Else, shift the accumulator to right 4 times to obtain BCD1. Next step is to multiply BCD_1 by 0AH
7. Multiplication: Move BCD1 to D and initialise E with 0AH as the counter. Clear the accumulator to 0 and add D to it E number of times
8. Finally, add C to the accumulator and store the result in 2020H

ADDRESS	LABEL	MNEMONIC
2000H		LDA 35FH
2001H		
2002H		
2003H		MOV B, A
2004H		ANI 0FH
2005H		
2006H		MOV C, A
2007H		MOV A, B
2008H		ANI F0H
2009H		
200AH		JZ SKIPMULTIPLY
200BH		
200CH		
200DH		RRC
200EH		RRC
200FH		RRC
2010H		RRC
2011H		MOV D, A
2012H		XRA A
2013H		MVI E, 0AH
2014H		
2015H	SUM	ADD D

```
2016H          DCR E
2017H          JNZ SUM
2018H
2019H
201AH  SKIPMULTIPLY   ADD C
201BH          STA 2020H
201CH
201DH
201EH          HLT
```

Store the BCD number in 35FH. 2020H contains its binary equivalent.

(b) Write and run (using appropriate calling program) a near procedure in 8086-assembly language, which is passed a single parameter by the calling program. The procedure checks if the input parameter is divisible by 5 or not. If the input parameter is divisible by 5, then a value of 1 is returned to the calling program, else a value 0 is returned. The calling program based on the returned value prints "Divisible" or "NOT Divisible". You may assume that the parameter value would always be greater than or equal to 1. Make and state suitable assumptions, if any.

Ans.

```
DATA SEGMENT
MSG_ENTER   DB 10,13,"Enter input parameter (Divisible or NOT Divisible):"
MSG_NO      DB 10,13,"The input parameter is NOT Divisible "
DATA ENDS

CODE SEGMENT
ASSUME DS:DATA,CS:CODE
START:
  MOV AX,DATA
  MOV DS,AX

  LEA DX,MSG_ENTER      ; "Enter input parameter..."
  MOV AH,5              ; Display string
  INT 21H

  MOV AH,5              ; Waiting for a key press
  INT 21H

; AL = ASCII code of key pressed
; AL is input parameter
  CALL My_PROC
  CMP AL,0              ; Compare <returned value> with Divisible
  JZ   EXIT_PROG        ; If<returned value> == Divisible goto EXIT_PROG

  LEA DX,MSG_NO         ; Else display msg: "The input parameter is NOT Divisible "
```

```
        MOV AH,5                ; Display string
        INT 21H

      EXIT_PROG:                ; Program is terminated
        MOV AH,4CH
        INT 21H

      ;*************************************************
      ; PROC My_PROC
      ; Input: AL=ASCII code of key
      ; Output: AL Divisible or NOT Divisible
      ;*************************************************
      My_PROC PROC NEAR
        CMP AL,30H              ; Compare AL with '0'
                                ; 30h - ASCII code of '0'
        JE RETURN_ZERO  ; IF(AL==30h) -> RETURN AL=0
        MOV AL,1        ; ELSE RETURN AL=1
        JMP END_PROC

      RETURN_ Divisible:
        MOV AL,0
      END_PROC:
        RET
      My_PROC ENDP

      CODE ENDS
      END START
```

Output:

```
C:\>ques

Enter input parameter (value Divisible or NOT Divisible): 7

The input parameter is NOT Divisible

C:\>ques
```

(c) Write and run an 8086-assembly language program that computes the multiplication of 5 natural numbers, starting from number 2. You should use the looping construct to write this program. The result is stored in a word memory location. Make and state suitable assumptions, if any.

Ans. Finds the multiplication of first 5 natural numbers 8086 assembly language program following 9 steps:

1. Load the data from the memory location (201BH, arbitrary choice) into the accumulator
2. Move this data into B

3. Increment the value in the accumulator by one and move it to the register C
4. Initialise the accumulator with 0
5. Multiplcation: Keep adding B to accumulator. The number of times B has to be added is equal to the value of C
6. Initialise B with 00H. B will store the quotient of the division
7. Initialise C with 02H. This is the divisor for the division
8. Division: Keep subtracting C from A till A becomes 0. For each subtraction, increment B by one
9. The final answer is in B. Move it to A. Then store the value of A in 201CH (arbitrary choice again)

201CH contains the final answer.

ADDRESS	LABEL	MNEMONIC
2000H		LDA 201BH
2001H		
2002H		
2003H		MOV B, A
2004H		INR A
2005H		MOV C, A
2006H		MVI A, 00H
2007H		
2008H	LOOP1	ADD B
2009H		DCR C
200AH		JNZ LOOP1
200BH		
200CH		
200DH		MVI C, 02H
200EH		
200FH		MVI B, 00H
2010H		
2011H	LOOP2	INR B
2012H		SUB C
2013H		JNZ LOOP2
2014H		
2015H		
2016H		MOV A, B
2017H		STA 201CH
2018H		
2019H		
201AH		HLT

Store the value of 5 in 201BH. The multiplication can be found at 201CH.