



Centro de Enseñanza Técnica Industrial

Plantel Colomos

Ingeniería en Desarrollo de Software

Nombre Alumno: José Rafael Ruiz Gudiño

Registro: 20110374

Arquitectura de Computadoras

Practica 3

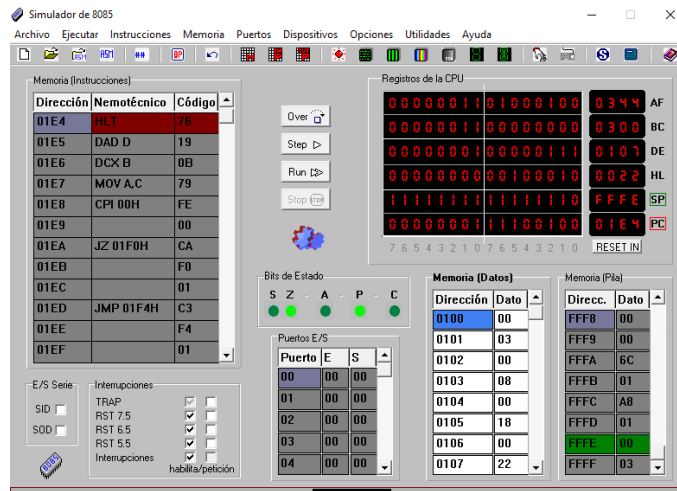
3°P

T/M

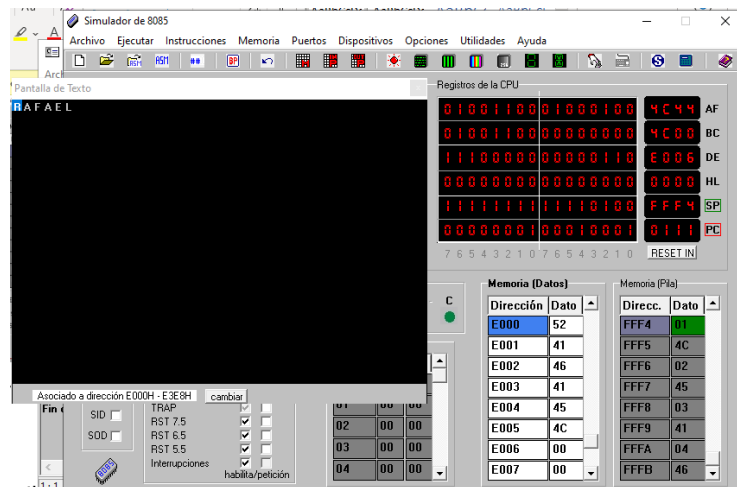
19/04/2021

Prácticas que se presentaron

Práctica 3.7



Práctica 3.1



Código de las prácticas

Práctica 3.1

```
.DATA E000H
DB 'R'
DB 'A'
DB 'F'
DB 'A'
DB 'E'
DB 'L'
.ORG 100H
MVI D, E0H ; Punteros de dirección del nombre (1000H)
MVI E, 00H
MVI C, 07H ; Letras + 1 para salir del ciclo.
CICLO:
DCR C
```

```

JZ SALIR
LDAX D      ; Se mueve al Acumulador lo que apunta D
MOV B,A     ; Se mueve a B el contenido del Acumulador
PUSH B      ; Se hace push de B pero incluye a C también (Se pone en
memoria de pila)
INR E      ; Se incrementa E para que apunte a la siguiente letra
JMP CICLO
SALIR:
HLT

```

Práctica 3.2

```

.DATA 1120H
DB 2FH
.ORG 1130H
MVI B, 11H
MVI C, 20H ; DIRECCION que se va a sumar
MVI H, 11H
MVI L, 00H ; Direccion a donde se va a mandar el resultado
LDAX B ; se manda la direccion a A
MVI B, FFH ; numero a sumar
ADD B ; se suma a A
MOV E, A ; ; se mueve a E
JC CARRY ; se verifica si no hay un desbordamiento
MOVDIREC:
MOV M, A ; se mueve a direccion lo de A
mov A, D ; SE VA A D lo de A
INR L ; se incrementa l para ir a la siguiente direccion
MOV M, A ; se envia el dato a la direccion siguiente
HLT ; se detiene

CARRY: ; se incrementa D para compensar el desbordamiento
INR D
JMP MOVDIREC

```

Práctica 3.3

```

.DATA 1100H
DB 3BH
DB 5AH
.DATA 1105H
DB 2FH
DB 33H
.ORG 1110H
MVI B, 11H
MVI C, 00H ; direccion de los numeros
MVI D, 11H
MVI E, 05H; direccion del contenido a restar
LDAX D ; se va el conteido de la direccion 1105H a A
MOV H,A ; Se va H el contenido de A
LDAX B ; se va el conteido de la direccion 1100H a A
SUB H ; Se resta lo de la direccion 1105h a 1100H
STAX B ; se va a 1100h el resultado
INR C
INR E ; se incrementa para ir a la siguiente direccion
LDAX D ;se va el conteido de la direccion 1106H a A
Mov H,A ; Se va H el contenido de A
LDAX B ; se va el conteido de la direccion 1101H a A

```

```

SUB H ; Se resta lo de la direccion 1106h a 1101H
STAX B ;se va a 1101h el resultado
HLT ; se detiene el programa

```

Práctica 3.4

```

.DATA 0002H
DB EEH ; inicialización del multiplicando
DB 4H ; inicialización del multiplicador

.ORG 100H
MOV H, 00H
MOV L, 00H

MVI B,00H ; Puntero B y C dan la posición 0002H
MVI C,02H

LDAX B ; Mueve a A el dato que apunta B Y C. (MULTIPLICANDO) 0002H
MOV D,A ; D tiene el MULTIPLICANDO
INR C ; SE INCREMENTA PARA OBTENER EL DATO DEL MULTIPLICADOR 0003H
LDAX B ; Mueve a A el registro 0003H (MULTIPLICADOR)
MOV E,A ; E tiene el MULTIPLICADOR

; D MULTIPLICANDO, E MULTIPLICADOR, TOTAL = HL (HIGH) Y (LOW)

MULTI:
MOV A,E ;A tiene el multiplicador
CPI 00H ; SI A = 00H finaliza el ciclo
JZ FIN

MOV A,L ; Parte menos significativa L.
ADD D ; se suma A + D en A (Sumo el multiplicando)
MOV L,A ; Guardo en L la parte menos significativa

DCR E ; se decrementa el multiplicador
JC ACARREO ; Si hay un desborde ya no tiene espacio L
JMP MULTI ;Sino vuelvo a hacer la suma
ACARREO:
INR H ; se Incrementa cada vez que haya desbordamiento (Parte
significativa)
JMP MULTI ; Regreso a la sumatoria hasta que el multiplicador sea 0.

FIN:

; En la posición 0004H se guarda la parte mas significativa (high) y en
0005H la parte menos significativa (low)
MOV A,H ; SE PONE EN A LA PARTE HIGH
MOV B, L ; EN B LA PARTE LOW

MVI D, 00H
MVI E, 04H ; direccion de memoria a mandar
STAX D ; se manda high
MOV A,B ; se manda la parte low a A
INR E ; Se incrementa E para ir a la siguiente direccion de memoria
STAX D ; se manda low a memoria
HLT

```

Práctica 3.5

```

; Fórmula tiempo del Intel 8085:  $T = 1/f = \frac{1}{2} \text{ MHz} = 5\mu\text{s}$ 
; formula de TD(tiempo delay) = (T=periodo del reloj) (LoopT-States=suma
de los t-states) (N10= Valor del delay)
; TD = (.5us) (87) (x) = 2ms -> x = 2ms / (.5us) (87) = 45D = 2DH
.ORG 0100H
INICIO:
MVI A, 00H ; 7T
MVI B, 2DH ; 7T

LOOP:
CPI 01H ; 7T
JZ FINAL ; 10T
INR A ; 4T
LOOP1:
DCR B ; 4T
JNZ LOOP1 ; 10T
JMP LOOP ; 10T

FINAL:
INR C ; 4T
MOV A,C ; 4T
OUT 00H ; 10T
JMP INICIO ; 10T
HLT

```

Práctica 3.6

```

.DATA 1000H
DB 12H ;X ; El programa hace sumatoria hasta del numero siguiente hasta
hacer carry
; ejemplo 12 = 13+14+15+16+17+18+19...hasta carry
.ORG 1005H
MVI B, 10H
MVI C, 00H ;direccion
LDAX B ; se va el dato 06h a A
MOV C, A ; SE VA C
SUMATORIA:
INR C ; SE INCREMENTA C 06H+1= 07H N>X 06H
ADD C ; SE VA SUMANDO N+1
JC FIN ; SI HAY ACARREO SE TERMINA
JMP SUMATORIA
FIN:
MOV E, A ; SE VA A E EL RESULTADO
INR D ; SE INCREMENTA D SI HUBO CARRY
MVI H, 10H
MVI L, 01H ; direccion memoria para el mas significativo
MOV A,D ; se va a HIGH
MOV M, A ;SE VA MEMORIA HIGH
INR L ; SE INCREMENTA LA DIRECCION
MOV A,E ; SE VA A ACUMULADOR EL DATO LOW
MOV M, A ; SE VA A MEMORIA LOW
HLT

```

Practica 3.7

```

.DATA 0098H
DB 01H;A 0098H
DB 02H;X 0099H

```

```

DB 00H;
DB 00H; RESULTADO 1 se guarda (A*I) en 100h y 0101H
DB 00H;
DB 00H; RESULTADO 2 (X^I) en 102h y 103H
DB 00H;
DB 00H; RESULTADO 3 (X^I) (A*I) en 104h y 0105H
DB 00H;
DB 00H; RESULTADO FINAL de la sumatoria en 106h 0107H
DB 03H; N 0108H
.ORG 0115H
;B tiene n
MVI B,00H
PUSH B
MAIN:
;CONTADOR I
POP B
INR B
;GUARDAR EL CONTADOR
PUSH B
;-- OPERACION (A*I)
MVI H,00H
MVI L,00H
;PUNTERO PARA EL VALOR DE A
MVI D,00H
MVI E, 98H

;SE SACA EL VALOR DE A
LDAX D
;SE MANDA EL DATO AL MULTIPLICADOR

MVI D,00H
MOV E,A
;SE RECUPERA EL CONTADOR
POP B
PUSH B
;SE MANDA EL CONTADOR A MULTIPLICAR
MOV C,B
MVI B,00H

CALL MULTI
;PUNTERO PARA GUARDAR DATO
MVI D,01H
MVI E, 00H
;GUARDA EL VALOR UNO
MOV A,H
STAX D
;PUNTERO PARA GUARDAR DATO
MVI D,01H
MVI E, 01H
;GUARDA EL VALOR UNO
MOV A,L
STAX D
;-- FIN DE OPERACION (A*I)
POP B
PUSH B
PUSH B
;-- INICIO DE OPERACION (X^I)

```

```

; INICIAR EL VALOR DE RESULTADO
; TEST
POP B
DCR B
PUSH B
; PUNTERO PARA EL VALOR DE X
MVI D, 00H
MVI E, 99H
; SE SACA EL VALOR DE X
LDAX D
MOV C, A
MVI B, 00H
MVI L, 00H
MVI H, 00H
POTENCIA:
; PUNTERO PARA EL VALOR DE X
MVI D, 00H
MVI E, 99H
; SE SACA EL VALOR DE X
LDAX D
MOV E, A
MVI D, 00H
POP H
PUSH H
MOV A, H
MVI L, 00H
MVI H, 00H
CPI 00H
JZ NOFLAT
JMP FLAT
NOFLAT:
MOV L, E
MOV H, D
JMP FINPOTENCIA
FLAT:
CALL MULTI
POP B
DCR B
PUSH B
MOV A, B
CPI 00H
JZ FINPOTENCIA
MOV B, H
MOV C, L
JMP POTENCIA
FINPOTENCIA:
POP B
; PUNTERO PARA GUARDAR DATO
MVI D, 01H
MVI E, 02H
; GUARDA EL VALOR UNO
MOV A, H
STAX D
; PUNTERO PARA GUARDAR DATO
MVI D, 01H
MVI E, 03H
; GUARDA EL VALOR UNO

```

```

MOV A,L
STAX D
;-----FIN DE LA OPERACION (X^I)
;-----INICIO DE LA
OPERACION(X^I) (A*I)
;PUNTERO PARA EL VALOR DE RESULTADO 2 MS
MVI D,01H
MVI E,02H

;SE SACA EL VALOR DE RESULTADO 2 MS
LDAX D
;SE MANDA EL DATO AL MULTIPLICADOR
MOV B,A
;PUNTERO PARA EL VALOR DE RESULTADO 2
MVI D,01H
MVI E,03H
;SE SACA EL VALOR DE RESULTADO 2
LDAX D
;SE MANDA EL DATO AL MULTIPLICADOR
MOV C,A
;PUNTERO PARA EL VALOR DE RESULTADO 1 MS
MVI D,01H
MVI E,00H
;SE SACA EL VALOR DE RESULTADO 1 MS
LDAX D
;SE MANDA EL DATO AL MULTIPLICADOR
MOV H,A
;PUNTERO PARA EL VALOR DE RESULTADO 1 mS
MVI D,01H
MVI E,01H
;SE SACA EL VALOR DE RESULTADO 1 mS
LDAX D
;SE MANDA EL DATO AL MULTIPLICADOR

MOV L,A
MOV D,H
MOV E,L
MVI H,00H
MVI L,00H
CALL MULTI
;PUNTERO PARA GUARDAR DATO
MVI D,01H
MVI E, 04H
;GUARDA EL VALOR UNO
MOV A,H
STAX D
;PUNTERO PARA GUARDAR DATO
MVI D,01H
MVI E, 05H
;GUARDA EL VALOR UNO
MOV A,L
STAX D
;-- FIN DE LA OPERACION(X^I) (A*I)
;-- SUMATORIA DEL RESULTADO
;PUNTERO PARA EL VALOR DE MS RESULTADO 3 (X^I) (A*I)
MVI D,01H
MVI E,04H

```



```
;SE SACA EL VALOR DE RESULTADO MS RESULTADO 3 (X^I) (A*I)
LDAX D
```

```
;SE MANDA EL DATO PARA SUMAR
```

```
MOV B,A
```

```
;PUNTERO PARA EL VALOR DE mS RESULTADO 3 (X^I) (A*I)
MVI D,01H
MVI E,05H
```

```
;SE SACA EL VALOR DE RESULTADO mS RESULTADO 3 (X^I) (A*I)
LDAX D
```

```
;SE MANDA EL DATO PARA SUMAR
```

```
MOV C,A
;PUNTERO PARA EL VALOR FINAL
MVI D,01H
MVI E,06H
```

```
;SE SACA EL VALOR FINAL
LDAX D
```

```
;SE MANDA EL DATO PARA SUMAR
```

```
MOV H,A
```

```
;PUNTERO PARA EL VALOR FINAL
MVI D,01H
MVI E,07H
```

```
;SE SACA EL VALOR FINAL
LDAX D
```

```
;SE MANDA EL DATO PARA SUMAR
```

```
MOV L,A
```

```
DAD B
```

```
;PUNTERO PARA GUARDAR DATO
MVI D,01H
MVI E, 06H
```

```
;GUARDA EL VALOR UNO
MOV A,H
STAX D
```

```
;PUNTERO PARA GUARDAR DATO
MVI D,01H
MVI E, 07H
;GUARDA EL VALOR UNO
MOV A,L
STAX D
```

```
;----- FIN DE LA SUMATORIA DEL RESULTADO
```

```

POP B
PUSH B
MOV A,B
CPI 03H ; ===N
JZ FINFIN
JMP MAIN
FINFIN:
HLT
MULTI:
;EL RESULTADO SALE EN H Y L
;EL B Y C DEBE TENER EL NUMERO DE CICLOS
;EN D Y E, TIENE QUE ESTAR EL DATO
UNO:
DAD D
DCX B
MOV A,C
CPI 00H
JZ CEROUNO
JMP DOS
CEROUNO:
MOV A,B
CPI 00H
RZ
DOS:
JMP UNO

```

Conclusión

La actividad fue realmente un reto para mi debido a que es la primera vez que he trabajado con ensamblador y con un simulador de microprocesador, por lo que fue complicado ya que tuve que ver como funcionaba el lenguaje y simulador, también tuve que aprender cómo manejar los datos para hacer los programas ya que es más complicado que en un lenguaje de alto nivel, aquí se tiene que jugar con los registros y memorias para poder hacer los programas. Sin embargo, con esta actividad aprendí una gran cantidad de cosas sobre ensamblador, ya que no sabia que a nivel ensamblador puedes manejar los datos en direcciones de memoria, registros, memoria pila, etc.