Ejercicios ensamblador

Luisa Cecilia Flores Fernández

Julián de Jesús López López is697509

Mariana Sierra Vega is702782

10 de octubre del 2017

Fundamentos de Microprocesadores

Profesor: Mario Alberto Peredo Durán

# Ejercicios de práctica en ensamblador

## Ejercicio 1

ORG 0000H

SJMP MAIN1

ORG 0040H

MAIN1: MOV A,#0FH

MOV R0,#12H

MOV R1,#34H

MOV R2,#56H

MOV R3,#78H

MOV R4,#09H

MOV R5,A

END

## Ejercicio 2

ORG 0000H

SJMP INICIO

ORG 0040H

INICIO: MOV A, #25H

ADD A, #1H

ADD A, #1H

ADD A, #1H

ADD A, #1H

ADD A, #1H

ADD A, #1H

ADD A, #1H

ADD A, #1H

ADD A, #1H

ADD A, #1H

ADD A, #1H

ADD A, #1H

DEC A

DEC A

DEC A

DEC A

DEC A

DEC A

DEC A

DEC A

DEC A

DEC A

DEC A

DEC A

END

## Ejercicio 3

ORG 0000H

SJMP MAIN1

ORG 0040H

MAIN1: ADD A,#1H

ADD A,#1H

ADD A,#1H

ADD A,#1H

ADD A,#1H

ADD A,#1H

ADD A,#1H

ADD A,#1H

ADD A,#1H

ADD A,#1H

SJMP MAIN1

END

## Ejercicio 4

ORG 0000H

SJMP MAIN4

ORG 0040H

MAIN4: MOV A,R4

ADD A,R6

MOV R2,A

END

## Ejercicio 5

DA0 EQU 30H

DA1 EQU 31H

DA2 EQU 32H

DA3 EQU 33H

DA4 EQU 34H

DA5 EQU 35H

DA6 EQU 36H

DA7 EQU 37H

DA8 EQU 38H

DA9 EQU 39H

RES EQU 40H

ORG 0000H

SJMP INICIO

ORG 0040H

INICIO: MOV DA0, #05H

MOV DA1, #06H

MOV DA2, #10H

MOV DA3, #02H

MOV DA4, #08H

MOV DA5, #15H

MOV DA6, #34H

MOV DA7, #67H

MOV DA8, #23H

MOV DA9, #11H

MOV A, DA0

ADD A,DA1

ADD A,DA2

ADD A,DA3

ADD A,DA4

ADD A,DA5

ADD A,DA6

ADD A,DA7

ADD A,DA8

ADD A,DA9

MOV RES, A

END

## Ejercicio 6

ORG 0000H

JMP MAIN

ORG 0040H

MAIN: MOV A,#10H

INC A

RL A

MOV A,#33H

XCH A,@R0 ;cuando R0 = 44

XRL A,#11H

ADD A,#11H

RR A

MOV B,#3H

DIV AB

SUBB A,#1H

JMP MAIN

END

## Ejercicio 7

ORG 0000H

SJMP MAIN7

ORG 0040H

MAIN7: MOV A,R0 ;Primero movemos al acumulador

RR A ;Recorremos 4 veces para que los bits más significativos se vuelvan los menos significativos

RR A

RR A

RR A

ANL A,#00001111b

MOV R1,A

MOV A,R0

ANL A,#00001111b

MOV R2,A

END

## Ejercicio 8

TEMP EQU 50H

RES EQU 51H

ORG 0000H

SJMP INICIO

ORG 0040H

INICIO: MOV A, R2

ANL A, #00001111b

MOV TEMP, A

MOV A, R1

ANL A,#00001111b

RR A

RR A

RR A

RR A

ORL TEMP, A

MOV R0, TEMP

END

## Ejercicio 9

CONTN EQU R3

CONTP EQU R4

CONT0 EQU R5

ORG 00H

JMP main

ORG 40H

main: MOV DPTR, #1940H

MOVX A, @DPTR

MOV R0, A

INC DPTR

MOVX A, @DPTR

MOV R1, A

INC DPTR

MOVX A, @DPTR

MOV R2, A

MOV DPH, R0

MOV DPL, R1

CICLO: MOVX A, @DPTR

JNZ CHECAR

INC CONT0

JMP REGRESO

CHECAR:

JB 0E7H, CHECAR

INC CONTP

JMP REGRESO

SUMAN:

INC CONTN

REGRESO: INC DPTR

DJNZ R2, CICLO

MOV A, CONTN

MOV DPTR, #1943H

MOVX @DPTR, A

MOV A, CONT0

MOV DPTR, #1944H

MOVX @DPTR, A

MOV A, CONTP

MOV DPTR, #1945H

MOVX @DPTR, A

END

## Ejercicio 10

CONT EQU 21H

ORG 0000H

SJMP MAIN10

ORG 0040H

MAIN10: MOV CONT,R0 ;Mover del registro 0 el número a evaluar

BEGIN: JB CONT.0, CONTINUE

ADD A,CONT

CONTINUE: DJNZ CONT, BEGIN

MOV R1,A

END;

## Ejercicio 11

CONTADOR EQU 20H

CONTADOR2 EQU 21H

INICIO1 EQU 1AH

INICIO2 EQU 1BH

FIN1 EQU 1CH

FIN2 EQU 1DH

ORG 0000H

SJMP INICIO

ORG 0040H

INICIO: MOV CONTADOR2, #1H

MOV CONTADOR, #00H

MOV R0, #1AH

MOV R1, #00H

MOV R2, #1CH

MOV R3, #00H

SJMP ESCRITURA

ESCRITURA: MOV DPH, R0

MOV DPL, R1

MOVX A, @DPTR

MOV DPH, R2

MOV DPL, R3

MOVX @DPTR, A

CJNE A, #0FFH, KEEP

SJMP $

KEEP: CJNE R1, #0FFH, AUMENTAR

CJNE R0, #1AH, FIN

INC R0

MOV R1, #00H

INC R2

MOV R3, #00H

SJMP ESCRITURA

AUMENTAR: INC R1

INC R3

SJMP ESCRITURA

FIN: SJMP $

END

## Ejercicio 12

ORG 0000H

JMP MAIN

ORG 0040H

;Version 1

MAIN: MOV R2, #7H

CICLO: ADD A, R0

DJNZ R2, CICLO

MOV R1, A

;Version2

MOV R0, #19H

MOV R2, #7H

MOV A, R0

END

## Ejercicio 13

DPS EQU 0A2H

ORG 0000H

JMP BEGIN

ORG 0040H

BEGIN: MOV DPTR,#1902H ;GUARDAR CANTIDAD DE DATOS EN LA LISTA R2

MOVX A,@DPTR

MOV R2,A

MOV DPS,#0 ;VERSION 0 DIRECCIONES

MOV DPTR,#1900H ;GUARDAR LA DIRECCIÓN DEL INICIO DE LA LISTA

MOVX A,@DPTR

MOV R3,A ;R3 DIRECCIÓN HIGH DEL APUNTADOR A LA LISTA

INC DPTR

MOVX A,@DPTR ;R4 DIRECCIÓN LOW DEL APUNTADOR A LA LISTA

MOV R4,A

DEC DPL

SETB AC

DATOS: MOV DPS,#1

MOV DPL,R3 ;OBTENER EL DATO

MOVX A,@DPTR

MOV R5,A ;R5 DATO HIGH DEL APUNTADOR A LA LISTA

MOV DPL,R4

MOVX A,@DPTR

MOV R6,A ;R6 DATO LOW DEL APUNTADOR A LA LISTA

JB AC,SETMENOR

MOV A,R5

CJNE A,00H,COMPARAR ;SI EL DE LA LISTA ES MENOR AL GUARDADO COMO MENOR

MOV A, R6

CJNE A,01H,COMPARAR

SUM: INC R3

INC R3

INC R4

INC R4 ;CAMBIA LAS DIRECCIONES DEL APUNTADOR ACTUAL

DJNZ R2,DATOS ;SI AÚN NO ES CERO REPITE EL PROCESO

MOV DPTR,#1903H

MOV A,R0

MOVX @DPTR,A

MOV A,R1

INC DPTR

MOVX @DPTR,A

JMP $

SETMENOR: MOV PSW, #00H ;PASA EL MENOR A MEMORIA

MOV DPS,#0

MOV DPTR,#1903H

MOVX A,@DPTR

MOV DPS,#1

MOV DPL,A

MOV A,R5

MOVX @DPTR,A

MOV A,R6

INC DPTR

MOVX @DPTR,A

MOV A,R5 ;R0R1 MENOR

MOV R0,A

MOV A,R6

MOV R1,A

JMP SUM

COMPARAR: JC SETMENOR

JMP SUM

END

## Ejercicio 14.

NUM EQU 20H

STOPH EQU 21H

STOPL EQU 22H

ORG 0000H

SJMP INICIO

ORG 0040H

INICIO: MOV DPTR, #1B00H

MOVX A, @DPTR

MOV NUM, A ;R0 tiene la longitud de la lista

MOV STOPH, #1BH

MOV STOPL, #01H

MOV A, NUM

ADDC A, STOPL

MOV STOPL, A

DEC STOPL

MOV A, #00H

ADDC A, STOPH

MOV STOPH, A

INC DPTR ;DPTR = 1B01H

CICLO: MOVX A, @DPTR

MOV R1, A ;R1 = Primer elemento de la lista

MOV R3, DPH

MOV R4, DPL

INC DPTR

MOVX A, @DPTR

MOV R5, DPH

MOV R6, DPL

MOV R2, A ;R2 =Contiene el segundo numero

MOV A, R1

SUBB A, R2

JC OTRA

MOV A, R1

MOV DPH, R5

MOV DPL, R6

MOVX @DPTR, A

MOV DPH, R3

MOV DPL, R4

MOV A, R2

MOVX @DPTR, A

;XCH A, R2

;MOV R1, A

MAYOR: INC DPTR

MOV A, DPH

CJNE A, STOPH, CICLO

MOV A, DPL

CJNE A, STOPL, CICLO

SJMP $

OTRA: MOV A, R2

MOV R1, A

MOV A, R5

MOV R3, A

MOV A, R6

MOV R4, A

SJMP CICLO

END

## Ejercicio 15

DPS EQU 0A2H

ORG 00H

JMP main

ORG 40H

main:

MOV DPTR,#1A00H

;Lee la direccion de inicio en forma R0R1

MOVX A,@DPTR

MOV R0,A

INC DPTR

MOVX A,@DPTR

MOV R1,A

;Lee la direccion de fin en forma R2R3

INC DPTR

MOVX A,@DPTR

MOV R2,A

INC DPTR

MOVX A,@DPTR

MOV R3,A

INC DPTR

;Comienza a leer el bloque de memoria

MOV DPS,#1

MOV DPH,R0

MOV DPL,R1

;Conteo unos = R4R5

cicloByte:

MOVX A,@DPTR

;MOV 20H,A ;Mueve el valor a verificar a ubicacion accesible por bits

JNB ACC.0, bit1;JNB 20H, bit1

ACALL incCheck

bit1:

JNB ACC.1,bit2;JNB 21H, bit2

ACALL incCheck

bit2:

JNB ACC.2, bit3;JNB 22H, bit3

ACALL incCheck

bit3:

JNB ACC.3,bit4;JNB 23H, bit4

ACALL incCheck

bit4:

JNB ACC.4, bit5;JNB 24H, bit5

ACALL incCheck

bit5:

JNB ACC.5,bit6;JNB 25H, bit6

ACALL incCheck

bit6:

JNB ACC.6,bit7;JNB 26H, bit7

ACALL incCheck

bit7:

JNB ACC.7, contCiclo;JNB 27H, contCiclo

ACALL incCheck

contCiclo:

INC DPTR

MOV A, DPL

CJNE A, 03H, cicloByte ;CJNE DPL, R3, cicloByte

MOV A, DPH

CJNE A, 02H, cicloByte ;CJNE DPH, R2, cicloByte

;Copiar los valores de la cuenta de unos a 1A04H y 1A05H

MOV DPS,#0

MOV A,R4

MOVX @DPTR,A

INC DPTR

MOV A,R5

MOVX @DPTR,A

forceEnd: SJMP $

incCheck:

;Verifica si ya se llenó el byte bajo de conteo R5

MOV A,R5

CPL A

JZ incHP

INC R5

JMP incLP

incHP:

;Verifica si ya se llenó el byte alto de conteo R4

MOV A,R4

CPL A

JZ forceEnd

INC R4

incLP:

RET

END

## Ejercicio 16

ORG 0000H

JMP MAIN

ORG 0040H

MAIN: MOV DPTR,#1C00H ;Guardar el número a evaluar

MOVX A, @DPTR

MOV R0,A ;R0R1

INC DPTR

MOVX A, @DPTR

MOV R1,A

MOV A,R0

JZ CERO\_L

CONTINUE: MOV A, R0 ;PRIMERO REVISAMOS SI EL BIT MÁS SIGNIFICATIVO ES 1

RLC A

MOV P1.0,C

RRC A

JB P1.0,RES ;SI ES UNO MANDA EL RESULTADO

MOV A, R1 ;SI NO, DESPLAZAMOS EL LOW

RLC A

MOV R1,A

MOV A, R0 ;DESPLAZAMOS EL HIGH

RLC A

MOV R0,A

INC R2

JMP CONTINUE

RES: MOV DPTR,#1C02H

MOV A,R0

MOVX @DPTR,A

INC DPTR

MOV A,R1

MOVX @DPTR,A

INC DPTR

MOV A,R2

MOVX @DPTR,A

JMP $

CERO\_L: MOV A,R1

JZ CERO\_RES

JMP CONTINUE

CERO\_RES: MOV DPTR,#1C02H ;el resultado = 0

MOV A,#00H

MOVX @DPTR,A

INC DPTR ;el resultado\_low = 0

MOVX @DPTR,A

INC DPTR ;desplazamientos = 0

MOVX @DPTR,A

JMP $

END

## Ejercicio 17

ORG 0000H

JMP main

ORG 0040H

main:

MOV DPTR, #0500H

MOV A, R4

MOV B, #02H

MUL AB

MOV R4, A

MOV R5, A

suma:

MOVX A, @DPTR

INC DPTR

MOV B, A

MOVX A, @DPTR

ADD A, R0

MOV R0, A

MOV A, B

ADDC A, R1

MOV R1, A

DJNZ R4, suma

promedio:

MOV A, R0

SUBB A, R5

MOV R0, A

MOV A, R1

SUBB A, #00H

MOV R1, A

INC R4

CJNE R1, #00H, promedio

MOV A, R0

MOV 30H, R5

CJNE A, 30H, escr

fin:

MOV A, R4

MOV R6, A

MOV A, R0

MOV R7, A

JMP $

escr:

JC fin

JMP promedio

END

## Ejercicio 18

L1H EQU 30H

L1L EQU 31H

DPS EQU 0A2H

ORG 0000H

JMP MAIN

ORG 0040H

MAIN: MOV DPS,#0

MOVX A,@DPTR ;R0 LOCALIDAD DEL NUMERO L1

MOV R0,A

INC DPTR

MOVX A,@DPTR ;R1 INICIO LISTA L1

MOV R1,A

MOV DPS,#1

MOV A,R6 ;OBTENER LA DIRECCIÓN DE MEMORIA

MOV DPL,A

MOVX A,@DPTR ;OBTENER EL DATO EN ESA MEMORIA DEL NUMERO L2

MOV R6,A

MOV A,R7 ;LO MISMO CON EL INICIO DE LA LISTA L2

MOV DPL,A

MOVX A,@DPTR

MOV R7,A

CLR P1.0

COMPARAR: MOV A,R1 ;COMPARAR L1 VS L2

CJNE A,07H,DIFERENTES

DIFERENTES: JC MENOR

JMP MAYOR

MENOR: MOV DPS,#0 ;GUARDAR EL DPS # 0 PARA QUE NO SE PIERDA

MOV L1H,DPH

MOV L1L,DPL

MOV DPH,R4 ;INGRESAR A R4 DIRECCION DEL RES H

MOV DPL,R5 ;RES L

MOV A,R1 ;MOVER EL A (R1) A ESA DIRECCION

MOVX @DPTR,A ;PONERLE A ESA DIRECCIÓN EL VALOR DE LA LISTA MENOR

INC DPTR ;REGRESAR EL DPTR #0 A LA LISTA 1

MOV R5,DPL

MOV R4,DPH

MOV DPH,L1H

MOV DPL,L1L

DJNZ R0,L1\_MOVER

JB P1.0,FIN

SETB P1.0

JMP MAYOR

L1\_MOVER: INC DPTR ;REGRESAR EL R1 AL DATO SIGUIENTE

MOVX A,@DPTR

MOV R1,A

JB P1.0, MENOR

JMP COMPARAR

MAYOR: MOV DPS,#1

MOV L1H,DPH

MOV L1L,DPL

MOV DPH,R4

MOV DPL,R5

MOV A,R7

MOVX @DPTR,A

INC DPTR

MOV R5,DPL

MOV R4,DPH

MOV DPH,L1H

MOV DPL,L1L

DJNZ R6,L2\_MOVER

JB P1.0, FIN

SETB P1.0

JMP MENOR

L2\_MOVER: INC DPTR

MOVX A,@DPTR

MOV R7,A

JB P1.0, MAYOR

JMP COMPARAR

FIN: JMP $

END

## Ejercicio 19

DPS EQU 02AH

PCAD EQU 30H ;DIRECCIÓN CADENA

PSUB EQU 31H ;DIRECCIÓN SUBCADENA

NUM EQU 32H ;NUMERO TOTAL

ORG 0000H

JMP MAIN

ORG 0040H

MAIN: MOV PCAD,#40H ;GUARDAR TODOS LOS APUNTADORES

MOV PSUB,#45H

MOV NUM,R2

DJNZ R2,POINT\_POS ;SI LA POS NO ES 0 RECORRER

POINT\_POS: INC PCAD ;INCREMENTAMOS EL APUNTADOR DE LA CADENA

DJNZ R2,POINT\_POS ;SI NO ES 0 VOLVER A DISMINUIR

MOV R3, PCAD

MOV A,R3

MOV R0,A ;COMO SI FUE 0

MOV A,@R0 ;HAY QUE COMPARAR EL DATO EN RAM

CJNE A,#30H,STACK ;SI NO ES IGUAL QUE 30H = 0 EN ASCII -> METER A STACK

STACK: MOV DPL,A ;MOVERL AL DPL

PUSH DPL ;METER A LA PILA

INC R4 ;CONTAR CUANTO METES A LA PILA

INC R3 ;CONTINUAS CON LA SIGUIENTE POSICIÓN

MOV A,R3

MOV R0,A

MOV A,@R0 ;HAY QUE COMPARAR EL DATO EN RAM

CJNE A,#30H,STACK ;SI NO ES IGUAL QUE 30H = 0 EN ASCII -> METER A STACK

SUB\_CAD: MOV R1,PSUB

MOV A,@R1

CJNE A,#30H,IN ;SI NO ES EL FINAL DE LA CADENA INSERTAR EL VALOR

MOV A,R4

MOV R5,A

INC R4

DEC PCAD

DJNZ R4,MOVE\_POINTER ;VOLVER A AGREGAR TODO

JMP $

IN: MOV R0,PCAD ;EN LA POSICIÓN DE PCAD

MOV @R0,A ;INSERTAR LO QUE ESTE EN EL ACUMULADOR

INC PSUB ;INCREMENTAR 1 EN PSUB

INC PCAD ;INCREMENTAR 1 EN PCAD

JMP SUB\_CAD

MOVE\_POINTER: INC PCAD

DJNZ R5,MOVE\_POINTER

POPSTACK: POP DPL

MOV R0,PCAD

MOV @R0,DPL

DEC PCAD

DJNZ R4,POPSTACK

JMP $

END