



Tecnológico de Monterrey

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Ciudad de México

- Tarea 3 Programación en Ensamblador -

Modelación de Sistemas Mínimos y Arquitecturas Computacionales

Grupo 233

Profesores:

Dr. Yoel Ledo

José Luis Almeida Esparza - A01028493

13 de octubre de 2023

2.-Ensambla y ejecuta los ejemplos 4.2 al 4.4 del libro de texto.

4.2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

Load Addr

Store Next

Load Num

Subt One

Store Ctr

Loop, Load Sum

AddI Next

Store Sum

Load Next

Add One

Store Next

Load Ctr

Subt One

Store Ctr

Skipcond 000

2014

OUTPUT MODE: DEC

MAR

014

MBR

0000

PC

008

IN

0000

OUT

0000

19

20

21

22

23

24

25

26

27

28

29

30

31

32

Jump Loop

Halt

Addr, Hex 117

Next, Hex 0

Num, Dec 5

Sum, Dec 0

Ctr, Hex 0

One, Dec 1

Dec 10

Dec 15

Dec 20

Dec 25

Dec 30

MAR

014

MBR

0000

PC

008

IN

0000

OUT

0000

Performed one step

| | +0 | +1 | +2 | +3 | +4 | +5 | +6 | +7 | +8 | +9 | +A | +B | +C | +D | +E | +F |
|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 000 | 1011 | 2012 | 1013 | 4016 | 2015 | 1014 | B012 | 2014 | 3016 | 2012 | 1015 | 4016 | 2015 | 8000 | 9005 | |
| 010 | 7000 | 0117 | 011A | 0005 | 0000 | 0001 | 0001 | 000A | 000F | 0014 | 0019 | 001E | 0000 | 0000 | 0000 | 0000 |
| 020 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 | 0000 |

4.3

Redacta brevemente la respuesta a los conceptos del capítulo 4: “Review of essential terms and concepts”.

Reformulación de conceptos y términos clave:

1. Función de la CPU: La CPU, conocida como la Unidad Central de Procesamiento, desempeña el papel principal en el funcionamiento de un sistema informático, actuando como su cerebro. Su tarea principal consiste en ejecutar operaciones matemáticas, lógicas y de control de datos, supervisando la operación global de la computadora.
2. Propósito del Datapath (Ruta de Datos): El Datapath se utiliza como el núcleo de control para supervisar los procesos. Contiene unidades funcionales, como las unidades de cálculo aritmético y lógico, y registra el procesamiento de datos.
3. Función de la Unidad de Control: La Unidad de Control dirige las actividades de la CPU al interpretar y gestionar las operaciones e instrucciones emitidas por otras unidades. Además, se encarga de garantizar la correcta manipulación y procesamiento de los datos.
4. Registros y Tipos: Los registros son áreas de almacenamiento de alta velocidad ubicadas dentro de la CPU, que incluyen registros de datos, registros de direcciones y registros de control.
5. Determinación de Funciones de la ALU: La Unidad Aritmética y Lógica (ALU) realiza operaciones matemáticas y lógicas, interpretando las operaciones a llevar a cabo según las instrucciones codificadas proporcionadas por la Unidad de Control.
6. Cuellos de Botella del Bus: El Bus, que es el sistema de comunicación a través del cual fluye la información, puede volverse ineficiente y lento cuando se incrementa la cantidad y la velocidad de procesamiento.
7. Diferencia entre Bus Punto a Punto y Bus Multipunto:
 - Bus Punto a Punto: Conecta directamente componentes específicos.
 - Bus Multipunto: Permite la conexión de múltiples componentes, facilitando la comunicación compartida entre ellos.
8. Importancia del Protocolo del Bus: El Protocolo del Bus establece la forma de comunicación y transferencia de información entre los componentes, garantizando la precisión y eficiencia de la transmisión.
9. Diferencias entre Buses de Datos, Direcciones y Control:
 - Bus de Datos: Transfiere información entre los componentes.
 - Bus de Direcciones: Lleva consigo direcciones de memoria.
 - Bus de Control: Gestiona señales de control, como las de lectura y escritura.

10. Ciclo del Bus: Un ciclo del bus ocurre cuando los datos se transfieren de un componente a través del bus. Este ciclo involucra tres etapas clave: la Fase de Dirección (donde se establecen las direcciones de transmisión), la Fase de Datos (donde se envían datos a través del bus) y la Fase de Control (que determina el tipo de operación que se llevará a cabo). Estos ciclos son cruciales para sincronizar las transferencias de datos en un sistema informático.

4.- Realiza un programa que nos diga la longitud de un string, del cual conoces la dirección donde inicia y termina cuando encuentra el NULL en el contenido de memoria; asume que el string está en ASCII

```

1 //Jose Luis Almeida Esparza A01028493
2
3 //set string start address
4 Input
5 Store stringStartAddress
6 Store tempAddress
7
8 //Ask for ascci chars until input null (0)
9 inputLoop, Input
10     Skipcond 400
11     Jump storeChar //if input != 0
12     //else
13     Jump getLength
14
15
16
17
18
19 storeChar, StoreI tempAddress //store input(last AC) into the adress that tempAddress points to
20
21     //increment tempAddress by 1
22     Load tempAddress
23     Add one
24     Store tempAddress
25     Jump inputLoop
26
27
28
29 getLength, Load stringStartAddress
30     Store tempAddress
31     LoadI stringStartAddress
32     Skipcond 400
33     Jump stringLoop //if string start address ptr value !=0
34     //else
35     Jump haltProgram
36
37
38
39 stringLoop, Load stringLength //add one to stringlength
40     Add one
41     Store stringLength
42
43
44     //increment temp adress by one
45     Load tempAddress
46     Add one
47     Store tempAddress
48
49
50     //if tempadress ptr val == 0 halt else go back to stringLoop
51     LoadI tempAddress
52     Skipcond 400
53     Jump stringLoop
54
55
56
57
58 //halt and print length
59 haltProgram, Load stringLength
60     Output
61     Halt
62
63
64

```

5.-Realiza un programa que calcule: $Y = X - 2 * A - 3 * B - 4 * C$

```

1 //Jose Luis Almeida Esparza A01028493
2
3 //Realiza un programa que calcule:  $Y=X-2*A-3*B-4*C$  (programa5.mas).
4
5
6 // Add x
7 Clear
8 Add X
9
10 //add -2*A
11 Subt A
12 Subt A
13
14 // add -3*B
15 Subt B
16 Subt B
17 Subt B
18
19 // add -4*c
20 Subt C
21 Subt C
22 Subt C
23 Subt C
24
25 Store Y
26 Output
27 Halt
28
29
30 Y, DEC 0
31 X, DEC 0
32 A, DEC 0
33 B, DEC 0
34 C, DEC 0
35

```

6.-Realiza un programa que calcule: $Y=5*(A-B+C)$

```

1 //Jose Luis Almeida Esparza A01028493
2
3 //Realiza un programa que calcule:  $Y=5*(A-B+C)$  (programa6.mas).
4
5
6
7 // temp = (A-B+C)
8 Clear
9 Add A
10 Subt B
11 Add C
12
13 Store TEMP
14
15
16 //add 5*temp
17 Clear
18 Add TEMP
19 Add TEMP
20 Add TEMP
21 Add TEMP
22 Add TEMP
23
24
25 Store Y
26 Output
27 Halt
28
29
30 Y, DEC 0
31 TEMP, DEC 0
32 A, DEC 0
33 B, DEC 0
34 C, DEC 0
35

```

7-Realiza un programa que invierta 4 localidades de memoria.

| | | |
|----|--------------------------------------------------------------------------|----|
| 1 | //Jose Luis Almeida Esparza A01028493 | 0 |
| 2 | | |
| 3 | //Problema 7. Realiza un programa que invierta 4 localidades de memoria. | IF |
| 4 | | 0 |
| 5 | Load A | M |
| 6 | Store T1 | C |
| 7 | | |
| 8 | Load B | M |
| 9 | Store T2 | 0 |
| 10 | | |
| 11 | Load C | P |
| 12 | Store T3 | C |
| 13 | | |
| 14 | Load D | IN |
| 15 | Store A | 0 |
| 16 | | |
| 17 | Load T3 | O |
| 18 | Store D | 0 |
| 19 | Load T2 | |
| 20 | Store C | |
| 21 | | |
| 22 | Load T1 | |
| 23 | Store B | |
| 24 | | |
| 25 | Halt | |
| 26 | | |
| 27 | A, DEC 20 | |
| 28 | B, DEC 10 | |
| 29 | C, DEC 40 | |
| 30 | D, DEC 30 | |
| 31 | T1, DEC 0 | |
| 32 | T2, DEC 0 | |
| 33 | T3, DEC 0 | |
| 34 | | |

8.-Realiza un programa que tome la primera localidad de memoria de una secuencia de 6 números, llenando las restantes cinco con valor de la anterior más uno.

```

1 //Jose Luis Almeida Esparza A01028493
2
3 /Programa 8. Realiza un programa que tome la primera localidad de memoria de una secuencia
4 /de 6 números, llenando las restantes cinco con el valor de la anterior más uno
5
6     Load    Num
7     Store 200
8
9     Load 200
10    ADD 1
11    Store 201
12
13    Load 201
14    Add 1
15    Store 202
16
17    Load 202
18    Add 1
19    Store 203
20
21    Load 203
22    Add 1
23    Store 204
24
25    Load 204
26    Add 1
27    Store 205
28    Halt
29    Num, DEC 2
30

```

9.-realiza un programa que tome una serie de cinco localidades de memoria y a cada una sume un valor: La primera 1, la segunda 2, etc.

```

1 //Jose Luis Almeida Esparza A01028493
2 |
3 //load arr start address into temp address
4 Load arrStartAddress
5 Store tempAddress
6
7 //make i the arr size
8 Load i
9 Add arrSize
10 Store i
11
12 //ask for memory addresses 5 times
13 inputLoop, Input
14 StoreI tempAddress
15
16 //increment temp address by one
17 Load tempAddress
18 Add ONE
19 Store tempAddress
20
21 //decrement i by one
22 Load i
23 Subt ONE
24 Store i
25
26
27 Skipcond 400
28 Jump inputLoop //if i != 0 loop again
29
30
31 addLoop, Load i//increment i by 1
32 Add ONE
33 Store i
34
35
36 LoadI arrStartAddress //load the value that arrStartAddress points to (its another address/pointer)
37 Add i // add i to the loaded address
38 Subt ONE // sub 1 to offset
39 Store tempAddress //store the incremented address
40
41
42 LoadI tempAddress //load the value that tempaddress point to
43 Add i //add i to the address it points to
44 StoreI tempAddress //store the value in the pointed address
45
46
47 Load arrSize //load array size
48 Subt i
49 Skipcond 400
50 Jump addLoop //if (arrsize - i) != 0 loop again
51
52
53 Halt
54
55
56
57
58 i, DEC 0
59 ONE, DEC 1
60 arrSize, DEC 5
61 arrStartAddress, DEC 128
62 tempAddress, DEC 128
63
64

```