

# Informe previo Práctica-5

Apellidos y nombre: DURAN LÓPEZ MARC Grupo: 33

Apellidos y nombre: ..... Grupo: .....

(por orden alfabético)

## Pregunta 1

(Contesta solo a los apartados que consideres oportunos para mejorar tu aprendizaje)

Lenguaje ensamblador	Lenguaje máquina (L.M.) (binario)	L.M. (hexa)
ADDI R2, R0, -1	0010000010111111	0x20BF
ADDI R5, R0, -120	Instrucción no válida	---
BNZ R2, -6	1000 0101 1111 1010	85FA
SHL R7, R7, R3	0000 1110 1111 1111	0EFF
ADD R6, R6, R6	0000 1101 1011 0100	0DB4
MOVI R0, -100	1001 0000 1001 1000	909C
BZ R4, 2	1000 1000 0000 0010	8802
CMPLT R2, R2, R3	0001 0100 1101 0000	14D0
CMPLEU R4, R7, R1	0001 1110 0110 0101	1E65
MOVHI R5, 0xA4	1001 1011 1010 0100	9BA4

## Pregunta 2

(Contesta solo a los apartados que consideres oportunos para mejorar tu aprendizaje)

Lenguaje máquina (hexa)	Lenguaje máquina (L.M.) (binario)	Lenguaje ensamblador
0x20C3	0010 0000 1100 0011	ADDI R3, R0, 3
0x1052	- - - - -	Instrucción no válida
0x0FCF	0000 1111 1111 0011	SHL R1, R7, R7
0x7000	0111 0000 0000 0000	SALR R0, R0, 0
0x4200	0100 0010 0000 0000	ST 0(R1), R0
0x6282	0110 0010 0100 0001	STB 2(R1), R2
0xA4B2	1010 0100 1011 0010	IN R2, B2
0x9DF8	1001 1101 1111 1000	MOVHI R6, 0xF8
0x80AF	1000 0000 1010 1111	BZ R0, 0xAF
0x1FF4	0001 1111 1111 1010	CMPLTU R6, R7, R7



### Pregunta 3



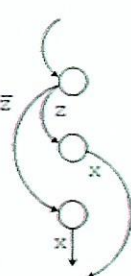

(Contesta solo a los apartados que consideres oportunos para mejorar tu aprendizaje)

- a) ADDI R3, R1, 7  
Respuesta: R3 = 8 // PC = 0x00B0
- b) ADD R3, R4, R5  
Respuesta: R3 = 1 // PC = 0x00B0
- c) BNZ R3, -6  
PC = 0x00A4
- d) SHL R7, R7, R2  
R7 = 1 // PC = 0x00B0
- e) SHA R7, R7, R2  
R7 = 1 // PC = 0x00B0
- f) CMPLTU R5, R7, R3  
R5 = 1 // PC = 0x00B0
- g) CMPEQ R5, R7, R3  
R5 = 1 // PC = 0x00B0
- h) BZ R1, -1  
PC = 0x00B0
- i) ADDI R3, R3, -3  
R3 = -2 // PC = 0x00B0
- j) AND R5, R1, R7  
R5 = 1 // PC = 0x00B0
- k) LD R2, 30(R5)  
R2 = MEM<sub>w</sub>[0x001F] // PC = 0x00B0
- l) STB 3(R0), R2  
MEM<sub>w</sub>[0x0003] = 0 // PC = 0x00B0
- m) ST -26(R5), R4  
MEM<sub>w</sub>[0xFFE7] = 0 // PC = 0x00B0

### Pregunta 4

(Contesta solo a los apartados que consideres oportunos para mejorar tu aprendizaje)

Fragmento de grafo con mnemotécnicos para la palabra de control	Fragmento de programa en lenguaje ensamblador SISA
a) 	AND R1, R2, R3
b) 	MOVI R6, -3 SHA R7, R7, R6

<p>c)</p> 	<pre> MOVHI    R3, 327 </pre>	<pre> MOVHI    R3, 0x47 MOVHI    R3, 0x01 </pre>
<p>d)</p> 	<pre> MOVHI    R1, -22 </pre>	<pre> MOVHI    R1, 0xEA </pre>
<p>e)</p> 	<pre> SUBI     -, R2, 1 ADD      R3, R5, R5 SUB      R3, R4, R5 </pre>	<pre> ADDI     R2, R2, -1 BNZ      R2, 1 ADD      R3, R5, R5 BZ       R2, 1 SUB      R3, R4, R5 </pre>
<p>f)</p> 	<pre> CMPLTU   -, R2, 250 SHLL     R4, R1, 4 </pre>	<pre> MOVHI    R0, 0xFA MOVHI    R0, 0x00 CMPLTU   R7, R2, R0 BZ       R7, 1 MOVHI    R3, 4 SHL      R4, R1, R3 </pre>

### Pregunta 5

(Contesta solo a los apartados que consideres oportunos para mejorar tu aprendizaje)

a)  $R4 = 0;$   
`MOVHI R4, 0`

b)  $V[R2] = R3 * 2;$   
`MOVHI R0, 1`  
`SHL R3, R3, R0`  
`V[R2] = R3`

c)  $V[10] = V[R2 + 3];$

d) if ( $R3 \leq R1$ )  $R3 = R1 - 1;$

CMPLT R7, R3, R1

BZ R7, 1

ADDI R3, R1, -1

e) if ( $R1 \geq 320$ )  $R2 = R2 + R2;$

else  $R5 = R2 + R5;$

MOVI R6, 0x40

MOVHI R6, 0x0A

CMPLT R7, R1, R6

BNZ R7, 1

ADD R2, R2, R2

BZ R7, 1

ADD R5, R2, R5

f) for ( $R2 = 3; R2 \leq R5; R2 = R2 + 1$ ) {  
     $V[R1 + R2 + 25] = 0;$

MOVI R2, 3

CMPLT R7, R2, R5

BZ R7, 5

ADDI R3, R2, 25

ADD R3, R1, R3

$V[R3] = 0$

ADDI R2, R2, 1

BNZ R2, -7

g) for ( $R3 = 0; R3 < 16; R3 = R3 + 1$ ) {  
     $V[R3 + R2] = 0;$

MOVI R3, 0

MOVI R4, 16

CMPLT R7, R3, R4

BZ R7, 4

ADD R5, R3, R2

$V[R5] = 0$

ADDI R3, R3, 1

BNZ R3, -6



## Pregunta 6

### Algoritmo MUL16 en SISA

```

        MOVI R5, 0           ; Inicializa resultado

        MOVI R2, 16          ; Inicializa contador iteraciones

        MOVI R1, 1           ; Mascara bit 0

        MOVI R3, -1          ; R3= Constante para dividir por 2

for:    AND  R4, R7, R1       ; ¿R7<0> -- 1?

        BZ  R4, 1            ; si no ir a endif

        ADD R5, R5, R6        ; R5 = R5 + R6

endif:  SHL R6, R6, R1        ; R6 = R6 * 2

        SHL R7, R7, R3        ; R7 = R7 / 2

        ADDI R2, R2, -1       ; R2 = R2 - 1

        BNZ R2, for          ; if (R2 != 0) goto for
    
```

## Pregunta 7

Ciclo Fetch	Instrucción en ensamblador que se va a ejecutar	Estado de los registros, en el ciclo en que se hace el Fetch de la instrucción (en hexadecimal)								
		PC	R0	R1	R2	R3	R4	R5	R6	R7
0	MOVI R5, 0	000C	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	0003	0005
3	MOVI R2, 16	000E						0000		
6	MOVI R1, 1	0010			0010					
9	MOVI R3, -1	0012		0001						
12	AND R4, R7, R1	0014				FFFF				
15	BZ R4, 1	0016					0001			
18	ADD R5, R5, R6	0018								
21	SHL R6, R6, R1	001A						0003		
24	SHL R7, R7, R3	001C							0006	
27	ADDI R2, R2, -1	001E								0002
30	BNZ R2, -7	0020			000F					
33	AND R4, R7, R1	0014								
36	BZ R4, 1	0016					0000			
39	SHL R6, R6, R1	001A								
42	SHL R7, R7, R3	001C							000C	
45	ADDI R2, R2, -1	001E								0001
48	BNZ R2, -7	0020			000E					
51	AND R4, R7, R1	0014								
54	BZ R4, 1	0016					0001			
57	ADD R5, R5, R6	0018								

000F

a) ¿Cuántos ciclos tarda en ejecutarse el código completo en el computador SISC?

Una iteración son 10 u 11 instrucciones dependiendo del valor de R7. Para el valor de R7 dado en el ejercicio anterior tarda en hacer la multiplicación de 2 números de 16 bits 303 ciclos, ya que para cada instrucción son 3 ciclos, puesto que no hay ninguna acceso a memoria.

b) ¿Cuál es el estado del computador (el valor de los registros del procesador que se han modificado) después de ejecutarse el código completo?

$R_1 = 0001$  ;  $R_2 = 0000$  ;  $R_3 = FFFF$  ;  $R_4 = 0000$  ;  $R_5 = 000F$  ;  $R_6 = 0000$  ;  $R_7 = 0000$

### Pregunta 8

Algoritmo MUL en ensamblador SISA

```

MOVl R5, 0

MOVl R1, 1           ; Mascara bit 0

MOVl R3, -1

for:  AND  R4, R7, R1      ; ¿R7<0> == 1?

      BZ   R4, 1          ; si no ir a endif

      ADD  R5, R5, R6      ; R5 = R5 + R6

endif: SHL  R6, R6, R1     ; R6 = R6 * 2

      SHL  R7, R7, R3

      BNZ  R7, for
  
```

### Pregunta 9

Ciclo Fetch	Instrucción en ensamblador que se va a ejecutar	Estado de los registros, en el ciclo en que se hace el Fetch de la instrucción (en hexadecimal)								
		PC	R0	R1	R2	R3	R4	R5	R6	R7
0	MOVl R5, 0	000C	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	0081	0005
3	MOVl R1, 1	000E						0000		
6	MOVl R3, -1	0010		0001						
9	AND R4, R7, R1	0012				FFFF				
12	BZ R4, 1	0014					0001			
15	ADD R5, R5, R6	0016								
18	SHL R6, R6, R1	0018						0081		
21	SHL R7, R7, R3	001A							0102	
24	BNZ R7, -6	001C								0002
27	AND R4, R7, R1	0012								
30	BZ R4, 1	0014					0000			
33	SHL R6, R6, R1	0018								
36	SHL R7, R7, R3	001A							0304	
39	BNZ R7, -6	001C								0001
42	AND R4, R7, R1	0012								
45	BZ R4, 1	0014					0001			
48	ADD R5, R5, R6	0016								
51	SHL R6, R6, R1	0018						0385		
54	SHL R7, R7, R3	001A							0708	
57	BNZ R7, -6	001C								0000

a) ¿Cuántos ciclos tarda en ejecutarse el código completo en el computador SISC?

Para el valor dado de R7, tarda en hacer la multiplicación 57 ciclos.

b) ¿Cuál es el estado del computador (el valor de los registros del procesador que se han modificado) después de ejecutarse el código completo?

R1 = 0001 ; R3 = FFFF ; R4 = 0001 ; R5 = 0385 ; R6 = 0708 ; R7 = 0000

[R4 ↔ R5]

### Pregunta 10

Lenguaje Ensamblador	Lenguaje Máquina (L.M.) (binario)	L.M. Byte-1 (Hexa)	L.M. Byte-0 (Hexa)
Begin: IN R6, KEY-STATUS	1010 110 0 00000001	AC	01
BZ R6, -2	1000 110 0 11111110	8C	FE
IN R6, key-Data	1010 110 0 00000000	AC	00
IN R7, key-Status	1010 111 0 00000001	AE	01
BZ R7, -2	1000 111 0 11111110	8E	FE
IN R7, key-Data	1010 111 0 00000000	AE	00
MOVI R5, 0	1001 101 0 00000000	9A	00
MOVI R1, 1	1001 001 0 00000001	92	01
MOVI R3, -1	1001 011 0 11111111	96	FF
AND R4, R3, R1	0000 111 001 100 00	0E	60
BZ R4, 1	1000 100 0 00000001	88	01
ADD R5, R5, R6	0000 101 110 101 100	0B	AC
SHL R6, R6, R1	0000 110 001 110 111	0C	77
SHL R7, R7, R3	0000 111 011 111 111	0E	FF
BNZ R7, -6	1000 111 1 1111 1010	8F	FA
IN R0, Print-Status	1010 000 0 00000010	A0	02
BZ R0, -2	1000 000 0 11111110	80	FE
OUT Print-Data, R5	1010 101 1 00000000	AB	00
BZ R7, -19	1000 111 0 11101101	8E	ED

MOVI R4, 0 → 1001 1000 0000 0000 98 00  
 AND R5, R7, R1 → 0000 111 001 101 00 0E 68  
 BZ R5, 1 → 1000 101 0 00000001 8A 01  
 ADD R4, R4, R6 → 0000 100 110 100 100 09 A4  
 OUT Print-Data, R4 1010 100 1 00000000 A9 00