

Informe previo Práctica-5

Apellidos y nombre: Amari Culler Martínez Grupo: 31

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	100010111111010	0x217A
SHL R7, R7, R3	000011101111111	0x0EFF
ADD R6, R6, R6	0000110110110100	0x0DB4
MOVI R0, -100	100100010011100	0x909C
BZ R4, 2	1000100010000100	0x8802
CMPLT R2, R2, R3	0001010011101000	0x14D0
CMPLEU R4, R7, R1	000111101010101	0x1E55
MOVHI R5, 0xA4	1001101111010100	0x9BA4

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	0010100011000011	ADDI R3, R0, 3
0x1052	000100001010010	Instrucción no válida
0x0FCF	0000111111001111	SHL R7, R7, R7
0x7000	0111000100000000	JALR R0, R0
0x4200	0100001000000000	ST (R)R1, R0
0x6282	0110100101000010	STB (R)R1, R2
0xA4B2	1010010010110010	IN R2, 176
0x9DF8	1001110111111000	MOVHI R6, -8
0x80AF	1000000101011111	BZ R0, -175
0x1FF4	0001111111110100	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

d) SHL R7, R7, R2

e) SHA R7, R7, R2

f) CMPLTU R5, R7, R3

g) CMPEQ R5, R7, R3

h) BZ R1, -1

i) ADDI R3, R3, -3

j) AND R5, R1, R7



k) LD R2, 30(R5)

l) STB 3(R0), R2

m) ST -26(R5), R4

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
<p>a)</p>  <p>AND R1, R2, R3</p>	<p>AND R1, R3, R2</p>
<p>b)</p>  <p>SHA R7, R7, -3</p>	<p>SHA R7, R7, -1 SHA R7, R7, -1 SHA R7, R7, -1</p>

32 16 8 4 2 1
-32 1 1

c)



MOVI R3, 327

No es posible en SISA

d)

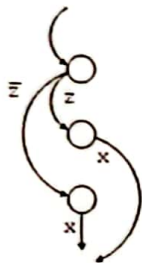


MOVI R1, -22

SUB R1, R1, R1

ADDI R1, R1, -22

e)



SUBI -, R2, 1

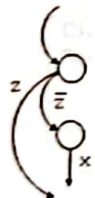
ADD R3, R5, R5

SUB R3, R4, R5

ADDI -, R2, -1

SHAI R3, R5, 1

f)



CMPLTUI -, R2, 250

SHL R4, R1, 4

CMPLTUI -, R2, 250

SHL R4, R1, 2

SHL R4, R1, 2

Pregunta 5

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

a) R4 = 0;

MOVE R4, 0x00

10

b) V[R2] = R3 * 2;

MOVE R1, 0x01

SHI R3, R3, R1

ST 0(R2), R3

Pregunta 6

Algoritmo MUL16 en SIS

```

MOV R5, 0 ; Inicializa resultado

MOV R2, 16 ; Inicializa contador iteraciones

MOV R1, 1 ; Mascara bit 0

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

for: AND R4, R7, R1 ; ¿R7<0> == 1?

      BZ R4, 7 ; 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 -7 ; if (R2 != 0) goto for
  
```

00011
0107

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, -7	0012		0007						
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, -7	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, -7	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								

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

Tiene 4 instrucciones, 6 que se repiten 16 veces en las 16 iteraciones, y 2 instrucciones de guardado del último bit de R7 en 7, y en ordenador SISC cada instrucción tarda 3 ciclos en este caso. Por lo tanto:

$$3 \cdot (4 + 6 \cdot 16 + 2) = 306 \text{ ciclos}$$

- 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 = 0007 R4 = 0000
R2 = 0000 R5 = 000F
R3 = FFFF R6 = R7 = 0000

Pregunta 8

Algoritmo MUL en ensamblador SISA

```

MOV R5, 0

MOV R1, 1 ; Mascara bit 0

MOV R3, -1

for: AND R4, R7, R1 ; ¿R7<0> == 1?

    BZ R4, 7 ; 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 -6
  
```

128 64 32 16 8 4 2 1
1 0 0 0 0 0 0 0

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	MOV R5, 0	000C	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	0081	0005
3	MOV R1, 1	000E						0000		
6	MOV R3, -1	0010		0007						
9	AND R4, R7, R1	0012				FFFF				
12	BZ R4, 7	0014					0007			
15	ADD R5, R5, R6	0016								
18	SHL R6, R6, R1	0018						0086		
21	SHL R7, R7, R3	001A							0102	
24	BNZ R7, -6	001C								0002
27	AND R4, R7, R1	001E								
30	BZ R4, 7	0020					0000			
33	SHL R6, R6, R1	0022								
36	SHL R7, R7, R3	0024							0204	
39	BNZ R7, -6	0026								0007
42	AND R4, R7, R1	0028								
45	BZ R4, 7	002A					0007			
48	ADD R5, R5, R6	002C								
51	SHL R6, R6, R1	002E						0285		
54	SHL R7, R7, R3	0030							0408	
57	BNZ R7, -6	0032								0000

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

Según el número de ciclos de cada instrucción:
 $3 \cdot (3 + 5 \cdot 3 + 2) = 60 \text{ ciclos}$

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

$R1 = 0001$ $R5 = 0285$
 $R2 = XXXX$ $R6 = 0408$
 $R3 = FFFF$ $R7 = 0000$
 $R4 = 0001$

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, R7, R1	0000 1111 0110 0000	0E	60
BZ R4, 1	1000 1000 0000 0001	88	01
ADD R5, R5, R6	0000 1011 1010 1100	0B	AC
SHL R6, R6, R1	0000 1100 0111 0111	0C	77
SHL R7, R7, R3	0000 1110 1111 1111	0E	FF
BKZ R7, -6	1000 1111 1111 1010	8F	FA
IN R0, PRINT-STATUS	1010 0000 0000 0010	A0	02
BZ R0, -2	1000 0000 11111110	80	FE
OUT PRINT-DATA, R5	1010 1011 00000000	AB	00
BKZ R0, -19	1000 0001 11101101	81	ED

KEY-STATUS = 1

KEY-DATA = 0

PRINT-STATUS = 2

PRINT-DATA = 0