

## Exercici 1

<b>ADDI R2, R0, -1</b>	0010 000 010 111111	0x20BF
<b>ADDI R5, R0, -120</b>	---	---
<b>BNZ R2, -6</b>	1000 010 1 11111010	0x85FA
<b>SHL R7, R7, R3</b>	0000 111 011 111 111	0x0EFF
<b>ADD R6, R6, R6</b>	0000 110 110 110 100	0x0DB4
<b>MOVI R0, -100</b>	1001 000 0 10011100	0x909C
<b>BZ R4, 2</b>	1000 100 0 00000010	0x8802
<b>CMPLT R2, R2, R3</b>	0001 010 011 010 000	0x14D0
<b>CMPLEU R4, R7, R1</b>	0001 111 001 100 101	0x1E65
<b>MOVHI R5, 0xA4</b>	1001 101 1 10100100	0x9BA4

## Exercici 2

<b>0x20C3</b>	0010 000 011 000011	ADDI R3, R0, 3
<b>0x1052</b>	0001 000 001 010 010	---
<b>0x0FCF</b>	0000 111 111 001 111	SHL R1,R7,R7
<b>0x7000</b>	0111 000 000 000000	JALR R0,R0,0
<b>0x4200</b>	0100 001 000 000000	ST 0(R1),R0
<b>0x6282</b>	0110 001 010 000010	STB 2(R1),R2
<b>0xA4B2</b>	1010 010 0 10110010	IN R2,0xB2
<b>0x9DF8</b>	1001 110 1 11111000	MOVHI R6,0xF8
<b>0x80AF</b>	1000 000 0 10101111	BZ R0,-81
<b>0x1FF4</b>	0001 111 111 110 100	CMPLTU R6,R7,R7

## Exercici 3

**a) ADDI R3,R1,7**

R3 = 0x0008 // PC = 0x00B0

**b) ADD R3,R4,R5**

R3 = 0x0001 // PC = 0x00B0

**c) BNZ R3,-6**

R3 = 0x0001 // PC = 0x00A6

**d) SHL R7,R7,R2**

R7 = 0x0001 // PC = 0x00B0 ! NO MODIFICA RES

**e) SHA R7,R7,R2**

R7 = 0x0001 // PC = 0x00B0 ! NO MODIFICA RES

**f) CMPLEU R5,R7,R3**

R5 = 0x0001 // PC = 0x00B0 ! NO MODIFICA RES

**g) CMPEQ R5,R7,R3**

R5 = 0x0001 // PC = 0x00B0 ! NO MODIFICA RES

**h) BZ R1,-1**

R1 = 0x0001 // PC = 0x00B0 ! NO MODIFICA RES

**i) ADDI R3,R3,-3**

R3 = 0x000E // PC = 0x00B0

**j) AND R5,R1,R7**

R5 = 0x0001 // PC = 0x00B0 ! NO MODIFICA RES

**k) LD R2,30(R5)**

R2 = MEMw[0x001F] // PC = 0x00B0

**l) STB 3(R0),R2**

MEMb[0x0003] = 0x0000 // PC = 0x00B0

**m) ST -26(R5),R4**

MEMw[0xFFE7] = 0x0000 // PC = 0x00B0

## Exercici 4

**a) AND R1,R2,R3**

AND R1,R2,R3

**b) SHAI R7,R7,-3**

MOVI R3,0xFD

SHAI R7,R7,R3

**c) MOVI R3,327**

MOVI R3,0x47

MOVHI R3,0x01

**d) MOVI R1,-22**

MOVI R1,0xEA

**e) SUBI -,R2,1; ADD R3,R5,R5; SUB R3,R4,R5 // Mirar el graf**

ADDI R9,R2,-1

BNZ R9,1

ADD R3,R4,R5

SUB R3,R4,R5

**f) CMPLTUI -,R2,250; SHL R4,R1,4 // Mirar el graf**

MOVI R50,0xFA

MOVHI R50,0x00

MOVI R40,0x04

CMPLTU R3,R2,R50

BZ R3,1

SHL R4,R1,R40

## Exercici 5

**a) R4 = 0;**  
MOVI R4,0x00

**b) V[R2] = R3\*2;**  
MOVI R0,0x01  
SHL R3,R3,R0  
ST 0(R2),R3

**c) V[10] = V[R2+3];**  
LDB R10,3(R2)  
STB 10(R4),R10

**d) if (R3 <= R1) R3 = R1 - 1;**  
CMPLE R0,R3,R1  
BZ R0,1  
ADDI R3,R1,-1

**e) if (R1 >= 320) R2 = R2 + R2; else R5 = R2 + R5;**  
MOVI R32,0x40  
MOVHI R32,0x01  
CMPLE R0,R32,R1  
BZ R0,1  
ADD R2,R2,R2  
ADD R5,R2,R5

**f) for (R2 = 3; R2 <= R5; R2 = R2 + 1) V[R1 + R2 + 25] = 0;**  
MOVI R2,0x03  
CMPLT R0,R2,R5  
BZ R0,7  
MOVI R6,0x19  
MOVI R10,0  
ADD R6,R6,R1  
ADD R6,R6,R2  
STB 0(R6),R10  
ADDI R2,R2,1  
BNZ R0,-9

**g) for (R3 = 0; R3 < 16; R3 = R3 + 1) V[R3 + R2] = 0;**  
MOVI R3,0  
MOVI R16,0x10  
CMPLT R0,R3,R16  
BZ R0,5  
ADD R5,R3,R2  
MOVI R100,0  
STB 0(R5),R100  
ADDI R3,R3,1  
BNZ R0,-6

## Exercici 6

```

    MOVI R5,0      ; Resultat
    MOVI R2,16     ; Comptador 16
    MOVI R1,1      ; Constant per multi
    MOVI R3,-1     ; Constant per div
for:  AND R4,R7,R1  ; Comprovar si LSB==1
      BZ R4,endif   ; si no es, anar a endif
      ADD R5,R5,R6
endif: SHL R6,R6,R1 ; Multiplicar
       SHL R7,R7,R3 ; Dividir
       ADDI R2,R2,-1
       BNZ R2,for   ; Mentre no sigui 0, repetir

```

## Exercici 7

Ciclo Fetch	Instrucció en ensamblador que se va a executar	Estados de los registros, 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"					NO SALTA			
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"			SALTA					
36	BZ R4,1	"0016"					"0000"			
39	ADD R5,R5,R6						SALTA			
42	SHL R6,R6,R1	"001A"								
45	SHL R7,R7,R3	"001C"							"000C"	
48	ADDI R2,R2,-1	"001E"								"0001"
51	BNZ R2,-7	"0020"			"000E"					
54	AND R4,R7,R1	"0014"			SALTA					
57	BZ R4,1	"0016"					"0001"			
60	ADD R5,R5,R6	"0018"					NO SALTA			
63	SHL R6,R6,R1	"001A"						"000F"		
66	SHL R7,R7,R3	"001C"							"0018"	
69	ADDI R2,R2,-1	"001E"								"0000"
72	BNZ R2,-7	"0020"			"000D"					

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

# En aquest algoritme totes les instruccions són ràpides (3 cicles)

Instruccions independents: 4

Iteracions de 7 instruccions: 2

Iteracions de 6 instruccions: 16(Totals)-2(De 7 iter)=14

Cicles Fetch:  $4+2*7+14*6 = 102$

**TOTAL CICLES:**  $102*3 = 306$  cicles

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

PC	R0	R1	R2	R3	R4	R5	R6	R7
"0020"	XXXX	"0001"	"0000"	"FFFF"	"0000"	"000F"	"0000"	"0000"

## Exercici 8

```

        MOVI R5,0
        MOVI R1,1
        MOVI R3,-1
for:    AND R4,R7,R1
        BZ R4,endif
        ADD R5,R5,R6
endif:  SHL R6,R6,R1
        SHL R7,R7,R3
        BNZ R7,for

```

## Exercici 9

Ciclo Fetch	Instrucción en ensamblador que se va a ejecutar	Estados de los registros, 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	"0081"	"0005"
3	MOVI R1,1	"000E"						"0000"		
6	MOVI R3,-1	"0010"		"0001"						
9	AND R4,R7,R1	"0012"				"FFFF"				
12	BZ R4,1	"0014"					"0001"			
15	ADD R5,R5,R6	"0016"					NO SALTA			
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"								SALTA
30	BZ R4,1	"0014"					"0000"			
33	ADD R5,R5,R6						SALTA			
36	SHL R6,R6,R1	"0018"								
39	SHL R7,R7,R3	"001A"							"0204"	
42	BNZ R7,-6	"001C"								"0001"
45	AND R4,R7,R1	"0012"								SALTA
48	BZ R4,1	"0014"					"0001"			
51	ADD R5,R5,R6	"0016"					NO SALTA			
54	SHL R6,R6,R1	"0018"						"0285"		
57	SHL R7,R7,R3	"001A"							"0408"	
60	BNZ R7,-6	"001C"								"0000"

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

# En aquest algoritme totes les instruccions són ràpides (3 cicles)

Instruccions independents: 3

Iteracions de 6 instruccions: 2

Iteracions de 5 instruccions: 1

Cicles Fetch:  $3+2*6+1*5 = 20$

**TOTAL CICLES:  $20*3 = 60$  cicles**

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

PC	R0	R1	R2	R3	R4	R5	R6	R7
"001C"	XXXX	"0001"	XXXX	"FFFF"	"0001"	"0285"	"0408"	"0000"

## Exercici 10

PC (Hexa)	Lenguaje Ensamblador	Lenguaje Maquina (L.M.)(binario)	L.M. Byte-1 (Hexa)	L.M. Byte-0 (Hexa)
"0000"	Begin: IN R6, KEY-STATUS	1010 110 0 00000001	"AC"	"01"
"0002"	BZ R6, Begin	1000 110 0 11111110	"8C"	"FE"
"0004"	IN R6, KEY-DATA	1010 110 0 00000000	"AC"	"00"
"0006"	POLLING-2: IN R7, KEY-STATUS	1010 111 0 00000001	"AE"	"01"
"0008"	BZ R7, POLLING-2	1000 111 0 11111110	"8E"	"FE"
"000A"	IN R7, KEY-DATA	1010 111 0 00000000	"AE"	"00"
"000C"	MOVI R5,0	1001 101 0 00000000	"9A"	"00"
"000E"	MOVI R1,1	1001 001 0 00000001	"92"	"01"
"0010"	MOVI R3,-1	1001 011 0 11111111	"96"	"FF"
"0012"	for: AND R4,R7,R1	0000 111 001 100 000	"0E"	"60"
"0014"	BZ R4,endif	1000 100 0 00000001	"88"	"01"
"0016"	ADD R5,R5,R6	0000 101 110 101 100	"0B"	"AC"
"0018"	endif: SHL R6,R6,R1	0000 110 001 110 111	"0C"	"77"
"001A"	SHL R7,R7,R3	0000 111 011 111 111	"0E"	"FF"
"001C"	BNZ R7,for	1000 111 1 11111010	"8F"	"FA"
"001E"	POLLING-3: IN R0, PRINT-STATUS	1010 000 0 00000010	"A0"	"02"
"0020"	BZ R0,POLLING-3	1000 000 0 11111110	"80"	"FE"
"0022"	OUT PRINT-DATA,R5	1010 101 1 00000000	"AB"	"00"
"0024"	BNZ R0,Begin	1000 000 1 11101101	"81"	"ED"