# Informe previo Práctica-5

Apellidos y nombre: ......................................Miquel Torner....................................................... Grupo: .73..

(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 |  | 1000010111111010 | 0x85FA |
| SHL | R7, | R7, | R3 | 0000111011111111 | 0x0EFF |
| ADD | R6, | R6, | R6 | 0000110110110100 | 0x0DB4 |
| MOVI | R0, | -100 |  | 1001000010011100 | 0x909C |
| BZ | R4, | 2 |  | 1000100000000010 | 0x8802 |
| CMPLT | R2, | R2, | R3 | 0001010011010000 | 0x14D0 |
| CMPLEU | R4, | R7, | R1 | 0001111001100101 | 0x1E65 |
| MOVHI | R5, | 0xA4 |  | 1001101110100100 | 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 | 0010000011000011 | ADDI R3, R0, 3 |
| 0x1052 | 0001000001010010 | Instrucción no válida |
| 0x0FCF | 0000111111001111 | SHL R1, R7, R7 |
| 0x7000 | 0111000000000000 | Instrucció no vàlida |
| 0x4200 | 0100001000000000 | ST 0(R1), R0 |
| 0x6282 | 0110001010000010 | STB 2(R1), R2 |
| 0xA4B2 | 1010010010110010 | IN R2, -78 |
| 0x9DF8 | 1001110111111000 | MOVHI R6, -8 |
| 0x80AF | 1000000010101111 | R0, -81 |
| 0x1FF4 | 0001111111110100 | CMPLTU R6, R7, R7 |

#### Pregunta 3

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

* 1. ADDI R3, R1, 7

**Respuesta:** R3 = 8 // PC = 0x00B0

* 1. ADD R3, R4, R5

**Respuesta:** R3 = 1 // PC = 0x00B0

* 1. BNZ R3, -6

PC = 0x00AA

* 1. SHL R7, R7, R2

R7 = 0x0001 // PC = 0x00B0

* 1. SHA R7, R7, R2

R7 = 0x0001 // PC = 0x00B0

* 1. CMPLEU R5, R7, R3

R5 = 0x0001 // PC = 0x00B0

* 1. CMPEQ R5, R7, R3

R5 = 0x0001 // PC = 0x00B0

* 1. BZ R1, -1

PC = 0x00B0

* 1. ADDI R3, R3, -3

R3 = 0xFFFE // PC = 0x00B0

* 1. AND R5, R1, R7

R5 = 0x0001 // PC = 0x00B0

k) LD R2, 30(R5)

R2 = 0x001F // 0x00B0

l) STB 3(R0), R2

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

m) ST -26(R5), R4

MEMw[0xFFE7] = 0x0000 // 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 R0, 0xE4  MOVHI R0, 0x00  SHA R7, R7, R0 | |

|  |
| --- |
| **c)**  MOVI R3, 0x47  MOVHI R3, 0x01 |
| **d)**  MOVI R1, 0xEA  MOVHI R1, 0xFF |
| **e)**  ADDI R0, R2, -1  BZ R0, 2  ADD R3, R5, R5  BNZ R0, 1  SUB R3, R4, R5 |
| **f)**  MOVI R0, 0xFA  MOVHI R0, 0x00  CMPLTU R0, R2, R0  BZ R0, 2  MOVI R0, 0x04  SHL R4, R1, R0 |

#### Pregunta 5

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

a) R4 = 0;

MOVI R4, 0x00

b) V[R2] = R3 \* 2;

MOVI R0, 1

SHA R3, R3, R0

ST 0(R2), R3

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

LD R0, 3(R2)

MOVI R1, 0x0A

ST 0(R1), R0

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 R0, 0x40

MOVHI R0, 0x01

CMPLT R0, R0, R1

BZ R0, 2

ADD R2, R2, R2

BNZ R0, 1

ADD R5, R2, R5

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

}

MOVI R2, 0x03

MOVI R6, 0x00

CMPLE R0, R2, R5

BZ R0, 5

ADD R7, R2, R1

ADDI R7,R7, 25

ST 0(R7), R6

ADDI R2, R2, 1

BNZ R0, -7

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

}

MOVI R3, 0x00

MOVI R2, 0x0

MOVI R7, 0x10

CMPLT R0, R3, R7

BZ R0, 4

ADD R1, R3, R2

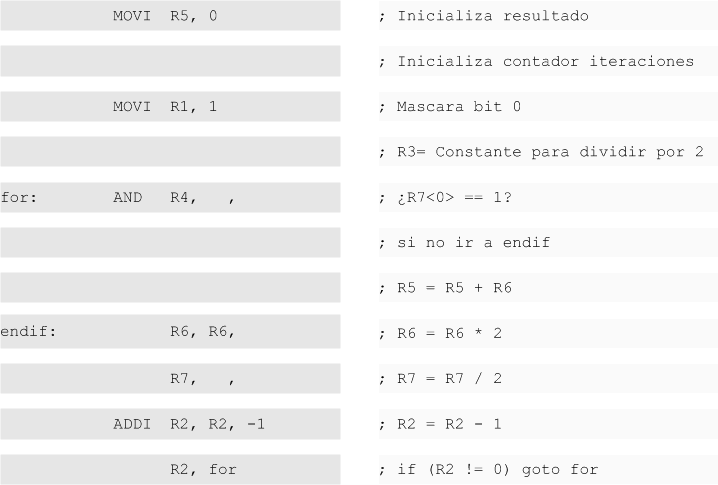
ST 0(R1), R2

ADDI R3, R3, 1

BNZ R0, -6

#### Pregunta 6

###### Algoritmo MUL16 en SISA



BNZ

SHL R1

SHL R7 R3

BZ R4, endif

ADD R5, R5, R6

R7 R1

MOVI R3, -1

MOVI R2, 0x10

**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, endif | 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, for | 0020 |  |  | 000F |  |  |  |  |  |
| 33 | AND R4, R7, R1 | 0014 |  |  |  |  |  |  |  |  |
| 36 | BZ R4, endif | 0016 |  |  |  |  | 0000 |  |  |  |
| 39 | SHL R6, R6, R1 | 001A |  |  |  |  |  |  |  |  |
| 42 | SHL R7, R7, R3 | 001C |  |  |  |  |  |  | 0009 |  |
| 45 | ADDI R2, R2, -1 | 001E |  |  |  |  |  |  |  | 0001 |
| 48 | BNZ R2, for | 0020 |  |  | 000E |  |  |  |  |  |
| 51 | AND R4, R7, R1 | 0014 |  |  |  |  |  |  |  |  |
| 54 | BZ R4, endif | 0016 |  |  |  |  | 0001 |  |  |  |
| 57 | ADD R5, R5, R6 | 0018 |  |  |  |  |  |  |  |  |

1. ¿Cuántos ciclos tarda en ejecutarse el código completo en el computador SISC?

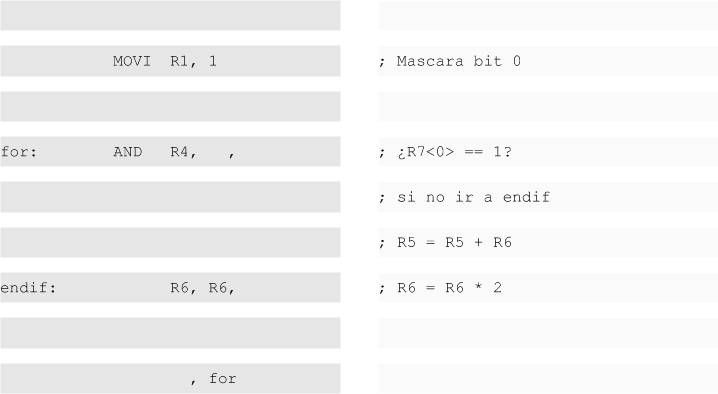
Per que s’executi el codi complet han de passar 306 cicles (4 instruccions inicials + 16 vegades el bucle (2 iteracions de 7 instruccions i la resta de 6))

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

R1 = 0x0001; R2 = 0x0000; R3 = 0xFFFF; R4 = 0x0000; R5 = 0x000F; R6 = 0x8000; R7 = 0x0000

#### Pregunta 8

###### Algoritmo MUL en ensamblador SISA



BNZ R7

SHL R7, R7, R3

SHL R1

ADD R5, R5, R6

BZ R4, endif

R7 R1

MOVI R3, -1

MOVI R5, 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 | 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, endif | 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, for | 001C |  |  |  |  |  |  |  | 0002 |
| 27 | AND R4, R7, R1 | 0012 |  |  |  |  |  |  |  |  |
| 30 | BZ R4, endif | 0014 |  |  |  |  | 0000 |  |  |  |
| 33 | SHL R6, R6, R1 | 0018 |  |  |  |  |  |  |  |  |
| 36 | SHL R7, R7, R3 | 001A |  |  |  |  |  |  | 0204 |  |
| 39 | BNZ R7, for | 001C |  |  |  |  |  |  |  | 0001 |
| 42 | AND R4, R7, R1 | 0012 |  |  |  |  |  |  |  |  |
| 45 | BZ R4, endif | 0014 |  |  |  |  | 0001 |  |  |  |
| 48 | ADD R5, R5, R6 | 0016 |  |  |  |  |  |  |  |  |
| 51 | SHL R6, R6, R1 | 0018 |  |  |  |  |  | 0285 |  |  |
| 54 | SHL R7, R7, R3 | 001A |  |  |  |  |  |  | 0408 |  |
| 57 | BNZ R7, for | 001C |  |  |  |  |  |  |  | 0000 |

1. ¿Cuántos ciclos tarda en ejecutarse el código completo en el computador SISC?

Triga un total de 60 en executar-se.

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

R1 = 0x0001; R3 = 0xFFFF; R4 = 0x0001; R5 = 0x0285;

R6 = 0x0408; R7 = 0x0000

#### 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, Begin | 1000 110 0 11111110 | 8C | FE |
| IN R6, KEY-DATA | 1010 110 0 00000000 | AC | 00 |
| Poll: IN R7, KEY-STATUS | 1010 110 0 00000001 | AC | 01 |
| BZ R7, Poll | 1000 110 0 11111110 | 8C | FE |
| IN R7, KEY-DATA | 1010 110 0 00000000 | AC | 00 |
| MOVI R5, 0 | 1001 001 0 00000000 | 92 | 00 |
| MOVI R1, 1 | 1001 101 0 00000001 | 9A | 01 |
| MOVI R3, -1 | 1001 011 0 11111111 | 96 | FF |
| for: AND R4, R7, R1 | 0000 111 101 100 000 | 0F | 60 |
| BZ R4, endif | 1000 100 0 00000001 | 88 | 01 |
| ADD R5, R5, R6 | 0000 001 110 001 100 | 03 | 8C |
| endif: SHL R6, R6, R1 | 0000 110 101 110 111 | 0D | 77 |
| SHL R7, R7, R3 | 0000 111 011 111 111 | 0E | FF |
| BNZ R7, for | 1000 111 1 11111010 | 8F | FA |
| Poll2: IN R0, PRINT-STATUS | 1010 000 0 00000010 | A0 | 02 |
| BZ R0, Poll3 | 1000 000 0 11111110 | 80 | FE |
| OUT PRINT-DATA, R5 | 1010 001 1 00000000 | A3 | 00 |
| BZ R7, Begin | 1000 111 0 11101101 | 8E | ED |

La taula inclou els canvis R1 🡨🡪 R5 els LM (bin i hex)