

# Organización del Computador 2021

Página Principal / Mis cursos / OrgCompu21 / Parcial 2 / Parcial 2

Navegación por el cuestionario

1	2	3	4	5
6	7	8	9	10
11				

Mostrar una página cada vez

Finalizar revisión

Comenzado el Friday, 11 de June de 2021, 09:00

Estado Finalizado

Finalizado en Friday, 11 de June de 2021, 12:42

Tiempo empleado 3 horas 42 minutos

Calificación 9,08 de 10,00 (91%)

Pregunta 1

Correcta

Puntúa 0,75 sobre 0,75

✗ Marcar pregunta

Considerando que el registro X1 cuenta con el valor -5, luego de ejecutarse el siguiente segmento de código:

```
1>> add x0, x31, x1
2>> cmpl x0, #1
3>> subis x0, x0, #1
4>> b.lo E1
5>> andi x0, x0, #0xF
    E1:
6>> addi x0, x0, #10
7>> lsl x0, x0, #2
    E2:
```

¿Que valor queda almacenado en el registro X0?

(Aclaración: Responder en hexadecimales, en mayúscula, considerando todos los bits del registro y sin ningún carácter extra)

X0 = 0x 0000000000000050 ✓

Pregunta 2

Correcta

Puntúa 1,00 sobre 1,00

✗ Marcar pregunta

Para el siguiente código en C:

```
#define N 10
uint64_t X[N + 1];
uint64_t Y[N + 1];
uint64_t Z[N + 1];
```

i = 0;

do {

Z[i] = X[i] + Y[i];

i++;

} while(i < N);

Elegir cual/cuales de los siguientes códigos LEGv8 es equivalente considerando la siguiente asignación de registros:

X0 <- &X[0]

X1 <- &Y[0]

X2 <- &Z[0]

X3 <- i

X4 <- N

a. //-- add X3, X31, X31

E0:

```
ldur X5, [X0, #0]
ldur X6, [X1, #0]
add X5, X5, X6
stur X5, [X2, #0]
addi X0, X0, #8
addi X1, X1, #8
addi X2, X2, #8
addi X3, X3, #1
subis X31, X3, X4
B.LE E0
```

b. //-- Ninguna es correcta

c. add X3, X31, X31

E0:

```
ldursw X5, [X0, #0]
ldursw X6, [X1, #0]
add X5, X5, X6
stur X5, [X2, #0]
addi X0, X0, #4
addi X1, X1, #4
addi X2, X2, #4
addi X3, X3, #1
subis X31, X3, X4
B.NE E0
```

d. //-- add X3, X31, X4

E0:

```
ldur X5, [X0, #0]
ldur X6, [X1, #0]
add X5, X5, X6
stur X5, [X2, #0]
addi X0, X0, #8
addi X1, X1, #8
addi X2, X2, #8
subis X3, X4, X3
B.LT E0
```

e. //-- Todas son correctas

La respuesta correcta es: //--

add X3, X31, X31

E0:

ldur X5, [X0, #0]

ldur X6, [X1, #0]

add X5, X5, X6

stur X5, [X2, #0]

addi X0, X0, #8

addi X1, X1, #8

addi X2, X2, #8

addi X3, X3, #1

subis X31, X3, X4

B.LE E0

Pregunta 3

Parcialmente correcta

Puntúa 1,88 sobre 2,50

✗ Marcar pregunta

Dada la siguiente sección de un programa en assembler **LEGv8**, asuma que los registros y la memoria contienen los valores mostrados en la tabla al inicio de la ejecución de dicha sección, y que el procesador esta configurado en **little-endian**.

```
    SUBI X4, X4, #1
    LDURSW X9, [X0, #0]
loop: LDURSW X10, [X0, #4]
    SUBS XZR, X9, X10
    B.LT! lab1
    EOR X9, X9, X10
    EOR X10, X9, X10
    EOR X9, X9, X10
lab1: STURW X9, [X0, #4]
    STURW X10, [X0, #0]
    SUBIS X4, X4, #1
    ADDI X0, X0, #4
    ADDI X2, X2, #4
    ADDI X3, X3, #1
    subis X31, X3, X4
    B.NE loop
exit: ...
```

Analizando la ejecución **COMPLETA** del código, indique en la siguiente tabla como queda el contenido de la memoria luego de finalizar la misma.

**IMPORTANT:** SE DEBE COMPLETAR LA TOTALIDAD DEL CONTENIDO DE LA MEMORIA, independientemente si se considera que cambió o no su contenido. Todos los valores numéricos deben ser expresados en HEXADECIMAL con letras minúsculas (Ej: 01, 02, ... 09, 0a, 0b, 0c,...). El tamaño de la palabra de memoria es de 1 byte.

Address (HEX) 0 1 2 3 4 5 6 7 8 9 a b c d e f

0x200400 46 61 4d 41 4e 20 2d 20 42 64 43 32 30 32 33 20

0x200410 b3 ba ca fe be ba c0 ca c0 1a 35 a5 10 12 34 56

0x200420 65 73 74 61 73 20 66 69 66 65 61 73 20 66 20

0x200430 63 61 6d 62 69 66 61 66 65 63 61 20 21 21

Pregunta 4

Parcialmente correcta

Puntúa 0,50 sobre 0,50

✗ Marcar pregunta

Importante:

- Los campos deben completarse en binario, respetando la cantidad de bits de cada uno de ellos.
- Si hay campos que no se utilizan en la instrucción dada, completarlos con ceros.
- Dejar vacíos los campos de los tipos de instrucciones distintos a la ensamblada.

R

Reg Value (HEX)

X0 0000000000000000

X1 0000000000000000

X2 0000000000000000

X3 0000000000000000

X4 0000000000000004

X5 0000000000000001

X6 0000000000000000

X7 ffffffff000000000000

X8 0000000000000000

X9 0000000000000000

X10 0000000000000000

X11 ffffff0dc2021ffff

X12 0000000000000000

X13 0000000000000000aca

Pregunta 5

Parcialmente correcta

Puntúa 0,50 sobre 0,50

✗ Marcar pregunta

Ensamble la siguiente instrucción LEGv8 y complete los campos correspondientes al tipo de instrucción dada. Los campos están en orden, según se indica en la Green Card.

ADDS X20, X0, #15

Importante:

- Los campos deben completarse en binario, respetando la cantidad de bits de cada uno de ellos.

Si hay campos que no se utilizan en la instrucción dada, completarlos con ceros.

Dejar vacíos los campos de los tipos de instrucciones distintos a la ensamblada.

R

Reg Value (HEX)

X0 0000000000000000

X1 0000000000000000

X2 0000000000000000

X3 0000000000000000

X4 0000000000000000

X5 0000000000000000

X6 0000000000000000

X7 0000000000000000

X8 0000000000000000

X9 0000000000000000

X10 0000000000000000

X11 0000000000000000

X12 0000000000000000

X13 0000000000000000

Pregunta 6

Parcialmente correcta

Puntúa 0,38 sobre 0,50

✗ Marcar pregunta

Seleccionar cuáles de las siguientes instrucciones **NO** pueden ser codificadas en LEGv8.

Tener en cuenta: la selección de opciones incorrectas resta puntuaje dentro del ejercicio, pero la nota final mínima es de 0 puntos.

a. MOVX X20,X0, #0FFF,LSL#16

b. STUR X1,[X20, #0FF]

c. BL X0, loop

d. LSR X20, X16, #0

e. EORI X20, X16, #16

Las respuestas correctas son: BL X0, loop, STUR X1,[X20, #0FF], EORI X20, X16, #16

Pregunta 7

Parcialmente correcta

Puntúa 0,50 sobre 0,50

✗ Marcar pregunta

Considerando que cada instrucción está guardada en la dirección de memoria indicada por el PC, calcular cuantas instrucciones como **MÁXIMO** puede avanzar luego de realizar un salto hacia **atrás**. Luego indicar si es posible con una sola instrucción de cada tipo alcanzar la dirección 0x0152f1e4?

PC (64bit-HEX)	INTRUCCIÓN	CANT DE INSTRU