

# Organización del Computador 2021

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>> cmpi 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 hexadecimal, en mayúscula, considerando todos los bits del registro y sin ningún carácter extra)

X0 = 0x

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];
uint64_t i;
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  
 sturw 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 X31, X3, X4  
 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
B.NE loop
exit: ...
    
```

Reg	Value (HEX)
X0	000000000200400
X1	000000000200410
X2	000000000200420
X3	000000000200430
X4	000000000000004
X5	0000000000000125
X6	0000000000000001
X7	EEEEEEEEEEEEEE
X8	0000000000000000
X9	0000000000000000
X10	000000000200400
X11	EEEE0ac2021EEEE
X12	0000000000000000
X13	000000000000cac

Address (HEX)	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0x200400	46	61	4d	41	46	20	2d	20	4f	64	43	32	30	32	31	20
0x200410	b3	ba	ca	fe	ba	c0	ca	c0	1a	35	a5	10	12	34	56	
0x200420	65	73	74	61	73	20	6c	69	6e	65	61	73	20	6e	6f	
0x200430	63	61	6d	62	69	6e	20	6e	75	6e	63	61	20	21	21	

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

**IMPORTANTE:** SE DEBE COMPLETAR LA TOTALIDAD DEL CONTENIDO DE LA MEMORIA, independientemente si se considera que cambió o no su contenido. Todos lo valores numéricos deben ser expresados en HEXADECIMAL con letras minúsculas (Ej: 01, 02, ..., 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
0x200400	46	61	4d	41	4f	64	43	32	46	
0x200410	b3	ba	ca	fe	be	ba	c0	ca	c0	
0x200420	65	73	74	61	73	20	6c	69	6e	
0x200430	63	61	6d	62	69	61	6e	20	6e	

Pregunta 4  
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.

ADDIS X20, X9, #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:

I:

D:

B:

CB:

IM:

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.

STURH X0, [X15, #-16]

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:

I:

D:

B:

CB:

IM:

Pregunta 6  
Parcialmente correcta  
Puntúa 0,38 sobre 0,50  
Marcar pregunta

Desensamble la siguiente instrucción LEGv8 y complete los campos correspondientes al tipo de instrucción resultante.

0xF2DFDFC0

Importante:

- Cargar todos los números inmediatos en formato decimal y las letras en mayúscula.
- Si la instrucción es un salto introducir en el campo el número de instrucciones que se deben saltar.
- Para cargar un número negativo anteponele el signo menos "-".
- Si se utiliza el registro 31, cargarlo como X31 y no como XZR.
- Dejar vacíos los campos de los tipos de instrucciones distintos a la desensamblada.

Por ejemplo, si la instrucción desensamblada es: STUR XZR, [X1, #-16]

Se debe completar:

D: STUR   #

R:

I:

D:

B:

CB:

IM: MOVK    LSL

Pregunta 7  
Parcialmente correcta  
Puntúa 0,50 sobre 0,50  
Marcar pregunta

Desensamble la siguiente instrucción LEGv8 y complete los campos correspondientes al tipo de instrucción resultante.

0xB4FFFE7

Importante:

- Cargar todos los números inmediatos en formato decimal y las letras en mayúscula.
- Si la instrucción es un salto introducir en el campo el número de instrucciones que se deben saltar.
- Para cargar un número negativo anteponele el signo menos "-".
- Si se utiliza el registro 31, cargarlo como X31 y no como XZR.
- Dejar vacíos los campos de los tipos de instrucciones distintos a la desensamblada.

Por ejemplo, si la instrucción desensamblada es: STUR XZR, [X1, #-16]

Se debe completar:

D: STUR   #

R:

I:

D:

B:

CB: CBZ

IM:     LSL

Pregunta 8  
Parcialmente correcta  
Puntúa 0,33 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 puntaje dentro del ejercicio, pero la nota final mínima es 0 puntos.

a. MOVK XZR, #0xFFFF; LSL #16

b. STUR X1, [X32, #0xFF]

c. BL X0, loop

d. LSR XZR, X10, #0

e. EORI X8, X16, #-16

Las respuestas correctas son: BL X0, loop, STUR X1, [X32, #0xFF], EORI X8, X16, #-16

Pregunta 9  
Correcta  
Puntúa 1,50 sobre 1,50  
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 permitido con una sola instrucción de cada tipo alcanzar la **dirección** 0x0152f1e4.

PC (64bit-HEX)	INTRUCCIÓN	CANT DE INSTRUCCIONES MAX HACIA ATRAS (DEC)	LLEGA A 0x0152f1e4?
000000001C00400	CBZ	262144	<input checked="" type="checkbox"/> Si llega
000000100080000	B	33554432	<input checked="" type="checkbox"/> No llega
000000001000040	BL	33554432	<input checked="" type="checkbox"/> No llega

Pregunta 10  
Correcta  
Puntúa 1,75 sobre 1,75  
Marcar pregunta

El diagrama de la figura es similar al datapath con control analizado en clases, pero contiene modificaciones para permitir la ejecución de una instrucción de LEGv8 que no estaba implementada originalmente. Complete el recuadro con el mnemonicó de la instrucción LEGv8 agregada, tal como aparece en la GreenCard:

br

Para que esta instrucción funcione correctamente se agregó una nueva señal de control "NewCont". Complete la siguiente tabla cargando el valor que deben tener esa y las demás señales de control durante la ejecución de esta nueva instrucción (completar cada campo con 1, 0 o X):

Reg2Loc	ALUSrc	MemtoReg	RegWrite	MemRead	MemWrite	Branch	NewCont
X	X	X	0	0	0	0	1
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Suponiendo que los diferentes bloques dentro del procesador tienen las siguientes latencias:

I-Mem/ D-mem	Register File	Mem	ALU	Addr	Single gate	Register Read	Register Setup	Sign extenu	Control	Shift left2	Zero extend	Shift Left
250ps	125ps	20ps	180ps	120ps	7ps	40ps	25ps	60ps	70ps	20ps	60ps	20ps

Cuál es la latencia de la instrucción (completar cada campo con 1, 0 o X):

Latencia =  ps

Pregunta 11  
Correcta  
Puntúa 1,00 sobre 1,00  
Marcar pregunta

En la figura se muestra el datapath con la unidad de control del procesador analizado en clases, en el que está implementado sólo un sub-set de instrucciones LEGv8. Se desea agregar la posibilidad de ejecutar la instrucción LEGv8 **SUBI**. Complete la siguiente tabla cargando el valor que deberían tomar las señales de control durante la ejecución de esta instrucción (completar cada campo con 1, 0 o X):

Reg2Loc	ALUSrc	MemtoReg	RegWrite	MemRead	MemWrite	Branch
X	1	0	1	0	0	0
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Finalizar revisión