

Parcial 2 2023-06-16

• Ejercicio 1

Determinar cuáles de las siguientes instrucciones pueden ser ensambladas en LEGv8

Justificar las respuestas negativas

Instruccion	SI/NO	Justificacion
BR 0	Si	
STUR X3, [X1, XZR]	NO	
MOVK X15, #0xC0CA, LSL #24	NO	sólo admite 00 16 32 48
ORRI X0, X1, #-32	NO	NO admite negativos
LSL X30, X31, #32	Si	Shift es de 6 bits \Rightarrow 00 a 63

• Ejercicio 2

Usando el siguiente ejemplo, escriba un programa en LEGv8 que divida x0 por x1 y salte dependiendo si el resultado es mayor igual a 0.20. Se pueden usar todas las instrucciones de LEGv8 excepto las de punto flotante. Debe resolverse en 5 instrucciones o menos.

MOVZ X0, #1, LSL #0 // X0 = 1

MOVZ X1, #4, LSL #0 // X1 = 4

~~Mov X2, #10, LSL #0~~

~~MUL X0, X0, X2~~

~~Udiv X1, X0, X1~~

~~CMPJ X1, #2~~



$X0/X1 \geq 0.20 \Leftrightarrow X0 * 5 \geq X1$

LSL X2, X0, #2
Add X0, X0, X0
CMP X0, X1

B.GE end // Saltar si X0/X1 \geq 0.20

• Ejercicio 3

Dada la siguiente sección de un programa en assembler LEGv8, el registro X1 contiene la dirección base de un arreglo A, mientras que X0 contiene el tamaño de dicho arreglo.

Asuma que los registros y la memoria contienen los valores mostrados en la tabla al inicio de la ejecución de dicha sección.

```

        ADDI X10, XZR, #0       $X10 = 0$ 
        SUBI X0, X0, #1         $X0 = 3$ 
LOOP:   CMP X10, X0
        B.GE LOOP_END // GT: mayor o igual
PROC:   LSL X11, X10, #3        $X11 = X10 * 8$ 
        ADD X11, X1, X11       $X11 = 0x10000100 + X11$ 
        LDUR X12, [X11, #0]
        LDUR X13, [X11, #8]
        CMP X13, X12
        B.GT NO_XCHG // GT: mayor
        STUR X13, [X11, #0]
        STUR X12, [X11, #8]
NO_XCHG: ADDI X10, X10, #1
        B LOOP
LOOP_END: ...

```

Memoria

Dirección	Contenido (antes)	Contenido (después)
0x10000100	-15	-30
0x10000108	-30	-15
0x10000110	70	10
0x10000118	10	70
0x10000120	200	200
0x10000128	80	80

Registros	
X0	0x00000004
X1	0x10000100

Responder

a) ¿Cómo queda el contenido de todas las posiciones de memoria mostradas en la tabla al finalizar la sección del programa? (Responder completando la columna en blanco de la tabla de memoria)

b) La instrucción contenida en la línea con la label "PROC:" se ejecuta 3 veces

Ejercicio 4

Considere el segmento de memoria que se muestra en la primera columna de la forma [dirección : contenido] que contiene codificado un programa en ISA LEGv8. Parte del programa desensamblado se presenta en la segunda columna.

(label) Dirección : Contenido	Desensamblado
B89103E9 1011 1006 1001 0001 0000 0011 1110 1001 OPC: 5C9 LDURSW 61Re D Dt Address = 1 0001 0000 = 0x110 OP = 0x0 Rn = x1 = xZR Rt = x9 =	ADDI x0, XZR, #4
0x00000100 : 0x910013E0 0x00000104 : 0xB89103E9 256+16 0x00000108 : 0xB1000529	LDURSW x9, [XZR, #0x110] SUBI x9, x9, #1
0x0000010C : 0XB81103E9 xD3600400 01 0011 0110 0000 0000 1100 0000 0000 1 PC: 0x69B = LSL Ro R. amb = #1 n = x0 d = x0	STURW x9, [XZR, 0x110]
0X00000110 : 0x14000002	B SKIP
0x00000114 : 0x8B1F03E0	ADD X0, XZR, XZR
(skip) 0x00000118 : 0xD3600400	LSL x0, x0, #1

0x14000001
 Branch
 0001 0100 0000 0000 0000 0000
 0000 0000 0001
 OPC = 000101 Branch
 BR Address = 1
 → Salta a 0x00000118 en vez de a SWIP.

Se pide:

a) Completar el desensamblado de las instrucciones faltantes y analice la ejecución del mismo

b) ¿Cuántas veces se ejecuta la instrucción contenida en la dirección 0x00000114 ?

1 veces.

c) ¿Cuál es el valor de X0 luego de la ejecución del segmento? X0: 0 por Add x0, xZR, xZR

Ejercicio 5

Considere que el procesador está ejecutando la instrucción de LEGv8 0xF84101BD y el contenido de los registros es X10 = 0x20 , X11=0x23 , X12=0x54 , X13=0x00 , PC=0x204

a) Desensamblar la instrucción: LDUR x29, [x13, #0x10]

b) ¿Qué operación realiza la ALU? : Suma

c) Completar la siguiente tabla con el estado de las señales de control

0xF84101BD
 1111 1000 0100 0001
 0000 0001 1011 1101
 OPC = 0x7C2 = LDUR
 Dt Address = 16 = 0x10
 OP = 0x0
 Rn = x13 Rd = x29

Reg2Loc	ALUSrc	MentoReg	RegWrite	MemRead	MenWrite	Branch
<u>0/x</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>

d) Completar la siguiente tabla con el valor de las señales indicadas, si es de entrada o salida del bloque y la cantidad de bits de la señal

Señal	E/S	Numero de bits	valor
Register, read data 1	S	64	0x00
Register, read register 2	E	5	X
Data memory, address	E	64	0x10
Register, write register	E	5	X
Entrada del PC	E	64	0x208
Add, ALU result	S	64	0x10

$0 \times 00 + 0 \times 10$

$0 \times 208 + 4$

