

Parcial 2 – 16/06/2023

Ejercicio 1:

Determinar cuáles de las siguientes instrucciones pueden ser ensambladas en LegV8. Justificar las respuestas negativas.

Instrucción	Si/No	Justificación
B 0		
LDUR X1, [XZR, X3]		
MOVK X15, #0xCAFE, LSL#8		
ORRI XZR, X1, #-16		
LSL X30, X30, #33		

Ejercicio 2:

Usando el siguiente ejemplo, escriba un programa en LegV8 que divida X0 por X1 y salte dependiendo si el resultado es mayor o igual a 0.33. 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, #3, LSL#0      //X1 = 3
_____                //
_____                //
_____                //
B.GE end                // Saltar si X0/X1 ≥ 0,33
```

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
    SUBI X0, X0, #1
Loop: CMP X10, X0
      B.GE LOOP_END // GE: mayor o igual
Proc: LSL X11, X10, #3
      ADD X11, X1, X11
      LDUR X12, [X11, #0]
      LDUR X13[X11, #8]
      CPM X13, X12
```

B.GT NO\_XCHG // GT: Mayor

STUR X13 [X11, #0]

STUR X12, [X11, #0]

NO\_XCHG: ADDI X10, X10, #1

B LOOP

LOOP\_END ...

Memoria		
Dirección	Contenido (antes)	Contenido (después)
0x10000100	122	
0x10000108	-30	
0x10000110	70	
0x10000118	10	
0x10000120	45	
0x10000128	200	

Registros	
X0	0x00000004
X1	0x10000100

Responder:

- ¿Cómo queda el contenido de todas las posiciones de memoria mostradas en la tabla al finalizar la sección del programa? Responde completando las columnas en blanco de la tabla de memoria.
- La instrucción contenida en la línea con el label PROC se ejecuta \_\_\_\_\_ 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 2da columna.

(label:)	Dirección: contenido	Desensamblado
	0x00000100: 0x910013E0	ADDI X0, XZR, #4
(loop:)	0x00000104: 0xB89103E9	
	0x00000108: 91001529	ADDI X9, X9, #5
	0x0000010C: B81103E9	STURW X9, [XZR, 0x110]
	0x00000110: 13FFFFFFD	B loop
	0x00000114: 8B1F03E0	ADD X0, XZR, XZR
	0x00000118: D3600400	

- Completar las instrucciones restantes.

- b) ¿Cuántas veces se ejecuta la instrucción con label: loop?
- c) ¿Cuál es el valor de X0 luego de la ejecución del segmento? \_\_\_\_\_

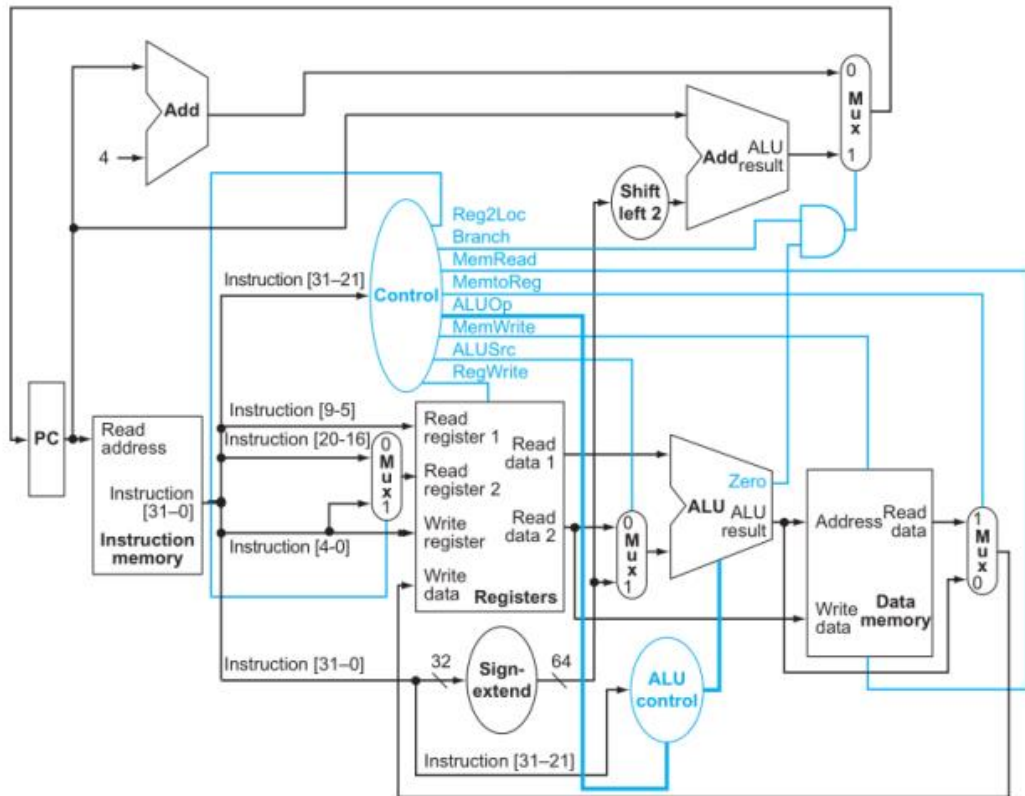
Ejercicio 5:

Considerar que el procesador está ejecutando la instrucción de LegV8: 0xF801816A y el contenido de los registros es:

X10 = 0x20, X11 = 0x23, X12 = 0x54. X13 = 0x00, PC = 0x104

- a) Desensamblar la instrucción.
- b) ¿Qué operación realiza la ALU?
- c) Completar el estado de las señales de control.

Reg2Loc	ALUSrc	MemtoReg	RegWrite	MemRead	MemWrite	Branch



Completar:

Señal	E/S	Nº de bits	Valor
Register, Read data 1			
Register, Read register 2			
Data Memory, address			
Register, write register			
Entrada del pc			
Add, ALUResult			