

PRÁCTICO 6 - Assembler de LEGv8 básico

Ejercicio 1:

Dadas las siguientes sentencias en "C":

- a) $f = g + h + i + j$;
- b) $f = g + (h + 5)$;
- c) $f = (g + h) + (g + h)$;

- 1.1) Escribir la secuencia **mínima** de código assembler LEGv8 asumiendo que f, g, h, i y j se asignan en los registros $X0, X1, X2, X3$ y $X4$ respectivamente.
- 1.2) Dar el valor de cada variable en cada instrucción assembler si f, g, h, i y j se inicializan con valores de 1, 2, 3, 4, 5, en base 10, respectivamente.

Ejercicio 2:

Luego, dadas las siguientes sentencias en assembler LEGv8:

- a) `ADD X0, X1, X2`
- b) `ADDI X0, X0, #1`
`ADD X0, X1, X2`

- 2.1) Escribir la secuencia **mínima** de código "C" asumiendo que los registros $X0, X1$ y $X2$ contienen las variables f, g y h respectivamente.
- 2.2) Dar el valor de cada variable en cada instrucción assembler si f, g y h se inicializan con valores de 1, 2, 3, en base 10, respectivamente.

Ejercicio 3:

Dadas las siguientes sentencias en "C":

- a) $f = -g - f$;
- b) $f = g + (-f - 5)$;

- 3.1) Escribir la secuencia mínima de código assembler LEGv8 asumiendo que f y g se asignan en los registros $X0$ y $X1$ respectivamente.
- 3.2) Dar el valor de cada variable en cada instrucción assembler si f y g se inicializan con valores de 4 y 5, en base 10, respectivamente.

Ejercicio 4:

Dadas las siguientes sentencias en assembler LEGv8:

- a) `SUB X1, XZR, X1`
`ADD X0, X1, X2`
- b) `ADDI X2, X0, #1`
`SUB X0, X1, X2`

- 4.1) Escribir la secuencia mínima de código "C" asumiendo que los registros $X0, X1$, y $X2$ contienen las variables f, g , y h respectivamente.
- 4.2) Dar el valor de cada variable en cada instrucción assembler si f, g , y h se inicializan con valores de 1, 2 y 3, en base 10 respectivamente.

Ejercicio 5:

Dadas las siguientes sentencias en “C”:

- a) $f = -g - A[4];$
 b) $B[8] = A[i - j];$

5.1) Escribir la secuencia **mínima** de código assembler LEGv8 asumiendo que f, g, i y j se asignan en los registros X0, X1, X2 y X3 respectivamente, y que la dirección base de los arreglos A y B se almacenan en los registros X6 y X7 respectivamente.

5.2) ¿Cuántos registros se utilizan para llevar a cabo las operaciones anteriores?

Ejercicio 6:

Dadas las siguientes sentencias en assembler LEGv8:

<p>a) LSL X2, X4, #1 ADD X0, X2, X4 ADD X0, X0, X4</p>	<p>b) LSL X9, X3, #3 ADD X9, X6, X9 LSL X10, X4, #3 ADD X10, X7, X10 LDUR X12, [X9, #0] ADDI X11, X9, #8 LDUR X9, [X11, #0] ADD X9, X9, X12 STUR X9, [X10, #0]</p>
--	--

6.1) Escribir la secuencia mínima de código “C” asumiendo que los registros X0, X1, X2, X3 y X4 contienen las variables f, g, h, i y j respectivamente, y los registros X6, X7 contienen las direcciones base de los arreglos A y B.

6.2) Para las instrucciones LEGv8 anteriores, re-escriba el código para minimizar (de ser posible) la cantidad de instrucciones manteniendo la funcionalidad.

Ejercicio 7:

Dadas las siguientes sentencias en assembler LEGv8:

```
ADDI X9, X6, #8
ADD X10, X6, XZR
STUR X10, [X9, #0]
LDUR X9, [X9, #0]
ADD X0, X9, X10
```

7.1) Asumiendo que los registros X0, X6 contienen las variables f y A (dirección base del arreglo), escribir la secuencia mínima de código “C” que representa.

7.2) Asumiendo que los registros X0, X6 contienen los valores 0xA, 0x100, y que la memoria contiene los valores de la tabla, encuentre el valor del registro X0 al finalizar el código assembler.

Dirección	Valor
0x100	0x64
0x108	0xC8
0x110	0x12C

Ejercicio 8:

Dado el contenido de los siguientes registros:

a) X9 = 0x55555555, y X10=0x12345678

b) X9 = 0x00000000AAAAAAAA, y X10=0x1234567812345678

8.1) ¿Cuál es el valor del registro X11 luego de la ejecución del siguiente código assembler en LEGv8?

```
LSL X11, X9, #4
ORR X11, X11, X10
```

8.2) ¿Cuál es el valor del registro X11 luego de la ejecución del siguiente código assembler en LEGv8?

```
LSL X11, X10, #4
ANDI X11, X11, #0xFFF
```

8.3) ¿Cuál es el valor del registro X11 luego de la ejecución del siguiente código assembler en LEGv8?

```
LSR X11, X9, #3
ANDI X11, X11, #0x555
```

Ejercicio 9:

Suponga que el registro X9 contiene el *Exception Syndrome Register* (ESR). Dé una secuencia **mínima** de instrucciones LEGv8 para poner en X10 el número que codifica la clase de excepción *Exception Class* (EC).

EXCEPTION SYNDROME REGISTER (ESR)

Exception Class (EC)	Instruction Length (IL)	Instruction Specific Syndrome field (ISS)
31	26	25
		24
		0

Ejercicio 10:

Suponga que el registro X9 contiene un número entero representado en complemento a dos. Dé una secuencia mínima de operaciones a realizar para devolver en X10 un 1 *si y sólo si* el contenido de X9 es negativo.

Ejercicio 11:

Utilizar MOVZ, MOVK para cargar los registros:

10.1) {X0 = 0x1234000000000000}

10.2) {X1 = 0xBBB0000000000AAA}

10.3) {X2 = 0xA0A0B1B10000C2C2}

10.4) {X3 = 0x0123456789ABCDEF}