



**ORGANIZACIÓN DE COMPUTADORAS**  
Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur  
Segundo Cuatrimestre de 2017



<b>Segundo Examen Parcial</b>		
Lic. en Ciencias de la Computación – Ing. en Computación – Ing. en Sistemas de Información		
Apellido y Nombre: (en ese orden)	LU:	Hojas entregadas: (sin enunciado)
Profesor:		
NOTA: Resolver los ejercicios en hojas separadas. Poner nombre, LU y número en cada hoja.		

**Ejercicio 1.** Implementar la siguiente expresión aritmética  $B = (A \times (D + C)) + (A \times (D + C)^2)$ , siendo  $A, B, C$  y  $D$  etiquetas que denotan *direcciones de memoria*, y asumiendo que se cuenta con las instrucciones **add** y **mpy**, para las siguientes arquitecturas:

- a) Una arquitectura de **0–direcciones** (tipo pila), contando con la instrucción **dup** (duplica el tope de la pila). Determinar la profundidad de la pila alcanzada.
- b) Una arquitectura estilo **RISC**, registro a registro, sin restricción en la cantidad de registros, y con instrucciones **lda**, **ld** y **st**. Indicar la cantidad de accesos a memoria realizados.
- c) Una arquitectura de **1–dirección + registro** (tipo Intel), sin restricción en la cantidad de registros y con la instrucción **mov**. Indicar la cantidad de accesos a memoria requeridos.

**Ejercicio 2.** En el marco de la norma **IEEE 754**, considerando la representación en punto flotante de media precisión: mantisa fraccionaria en signo magnitud con hidden bit, exponente en exceso y base 2 y la siguiente distribución de bits:

Sig (1bit)	Exponente (8 bits)	Mantisa (10 bits)
------------	--------------------	-------------------

Dados los números  $X = (1\ 10000100\ 0011111001)$  e  $Y = (0\ 01110101\ 1000111100)$  realizar el producto  $X \times Y$  aplicando redondeo por proximidad hacia los pares y hacia  $+\infty$ , explicando cada uno de los pasos involucrados e indicando claramente qué se hace con los bits **G**, **R** y **S** del resultado y con **R** y **S** al redondear. El resultado debe ser expresando según la representación enunciada. Finalmente, convierta el número hallado a decimal e indique el error existente entre este valor y el obtenido al operar la multiplicación directamente sobre  $X$  e  $Y$  en decimal.

(Pista:  $X = -39,78125$  e  $Y = 0,0015220642090$ ).

**Ejercicio 3.** Considerando la representación en punto flotante propuesta para el ejercicio anterior, y los números  $X = (0\ 01111100\ 0010110101)$  e  $Y = (1\ 01111101\ 1101000110)$ , realizar la suma  $X + Y$  aplicando redondeo por *proximidad unbiased* (hacia los pares), explicando cada uno de los pasos involucrados e indicando claramente qué se hace con los bits **G**, **R** y **S** del resultado y con **R** y **S** al redondear. El resultado debe ser expresando según la representación enunciada. Finalmente, convierta el número hallado a decimal e indique el error existente entre este valor y el obtenido al operar la suma directamente sobre  $X$  e  $Y$  en decimal.

(Pista:  $X = 0,147094727$  e  $Y = 0,454589844$ ).

**Ejercicio 4.** Determinar cuál es el contenido final de cada uno de los registros y posiciones de memoria involucrados en la siguiente secuencia de instrucciones. Indicar en cada caso, el número de instrucción que origina cada cambio. Asumir que el primer operando es el destino y el segundo la fuente de información para la operación.

(1) mov R1, #0200	Interpretación
(2) mov (R1), #0100	#xxxx Inmediato
(3) mov 0100(R1), R1	R Registro
(4) mov R2, #0500	(R) Registro indirecto
(5) mov @0100(R1), #0500	xxxx Absoluto
(6) mov (0200), 0300	xxxx(R) Indexado
(7) mov R3, 0200	(xxxx) Memoria indirecto
(8) mov R3, @0100(R3)	@xxxx(R) Pre-indexado indirecto

**Ejercicio 5.** Considerando el siguiente programa para la arquitectura OCUNS, en la que toda lectura/escritura sobre la dirección FF es redireccionada a la E/S estándar:

LDA R0, FFh	
LOAD R1, 0(R0)	
LOAD R2, 0(R0)	
XOR R3, R3, R3	
LDA R4, 1b13	
JZ R1, 1b13	
JZ R2, 1b13	
SUB R5, R1, R2	
JG R5, 1b12	
1b11: ADD R3, R3, R2	
DEC R1	
JG R1, 1b11	
JMP R4	
1b12: ADD R3, R3, R1	
DEC R2	
JG R2, 1b12	
1b13: STORE R3, 0(R0)	
HLT	

OP.	DESCR.	FORM.	PSEUDOCÓDIGO
<b>0</b>	add	<b>I</b>	$R[d] \leftarrow R[s] + R[t]$
<b>1</b>	sub	<b>I</b>	$R[d] \leftarrow R[s] - R[t]$
<b>2</b>	and	<b>I</b>	$R[d] \leftarrow R[s] \& R[t]$
<b>3</b>	xor	<b>I</b>	$R[d] \leftarrow R[s] \wedge R[t]$
<b>4</b>	lsh	<b>I</b>	$R[d] \leftarrow R[s] \ll R[t]$
<b>5</b>	rsh	<b>I</b>	$R[d] \leftarrow R[s] \gg R[t]$
<b>6</b>	load	<b>I</b>	$R[d] \leftarrow \text{mem}[\text{offset} + R[s]]$
<b>7</b>	store	<b>I</b>	$\text{mem}[\text{offset} + R[d]] \leftarrow R[s]$
<b>8</b>	lda	<b>II</b>	$R[d] \leftarrow \text{addr}$
<b>9</b>	jz	<b>II</b>	if $(R[d] == 0)$ $PC \leftarrow PC + \text{addr}$
<b>A</b>	jg	<b>II</b>	if $(R[d] > 0)$ $PC \leftarrow PC + \text{addr}$
<b>B</b>	call	<b>II</b>	$R[d] \leftarrow PC; PC \leftarrow \text{addr}$
<b>C</b>	jmp	<b>III</b>	$PC \leftarrow R[d]$
<b>D</b>	inc	<b>III</b>	$R[d] \leftarrow R[d] + 1$
<b>E</b>	dec	<b>III</b>	$R[d] \leftarrow R[d] - 1$
<b>F</b>	hlt	<b>III</b>	exit

FORMATO	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>I</b>	0	×	×	×		dest. d				src. s						src. t / off.
<b>II</b>	1	0	×	×		dest. d										address addr
<b>III</b>	1	1	×	×		dest. d										-

- Ensamblar el programa a partir de la dirección 00h.
- Si se reubicara el código máquina obtenido en el inciso (a) a partir de la dirección 20h, ¿qué referencias a memoria requieren ser ajustadas? Justificar adecuadamente.
- Suponiendo que los valores ingresados por teclado son 04h y 02h, realice una traza mostrando la evolución del contenido de cada registro, para luego, describir el propósito del programa en su conjunto.
- ¿Qué sucede con el resultado retornado si los valores ingresados fueran 02h y 04h? ¿Cuál es la diferencia? ¿Existe alguna restricción para los datos de entrada en cuanto al correcto funcionamiento del programa?