



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



<b>Recuperatorio 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  $B = C + D + A^2 + (A \times D)$ , 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 simplificada: 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  $X = (0\ 01111101\ 0000000111)$  e  $Y = (0\ 01111010\ 0000001001)$  realizar el producto  $X \times Y$  aplicando redondeo por proximidad 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 multiplicación directamente sobre  $X$  e  $Y$  en decimal.

(Pista:  $X = 0,251708984$ ,  $Y = 0,031524658$ ,  $X \times Y = 0,00793503969907761$ ).

**Ejercicio 3.** Considerando la representación en punto flotante propuesta para el ejercicio anterior, y los números  $X = (0\ 01111100\ 0010000000)$  e  $Y = (0\ 01111001\ 0000110011)$ , realizar la suma  $X + Y$  aplicando redondeo 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 suma directamente sobre  $X$  e  $Y$  en decimal.

(Pista:  $X = 0,140625$ ,  $Y = 0,016403198$ ,  $X + Y = 0,157028198242187$ ).

**Ejercicio 4.** Dados los valores indicados tanto para el banco de registros como para las etiquetas de memoria, indicar para cada modo de direccionamiento, el registro **R** y/o el número hexadecimal **xxxx** necesarios para mover el operando **100h** al registro **R6**. Luego, indicar en cada paso cuántos accesos a memoria se realizan por la instrucción.

Reg.	Cont.	Dir.	Cont.	(1) mov R6, #xxxx	(2) mov R6, R	(3) mov R6, (R)	(4) mov R6, xxxx	(5) mov R6, (xxxx)	(6) mov R6, (R2)xxxx	(7) mov R6, @300(R)	Interpretación
R1	100h	100h	500h	#xxxx	R	(R)	xxxx	(xxxx)	(R)xxxx	@xxxx(R)	Inmediato
R2	200h	200h	400h								Registro
R3	300h	400h	100h								Registro indirecto
R4	400h	600h	400h								Absoluto
											Memoria indirecto
											Base
											Pre-indexado indirecto

**Ejercicio 5.** Considerando el programa A para la arquitectura OCUNS, en la que toda lectura/escritura sobre la dirección **FF** es redireccionada a la E/S estándar, y los pseudocódigos 1 y 2 indicados a continuación:

Programa A:

```

LDA R0, FFh
LOAD R1, 0(R0)
ADD R2, RF, RF
JZ R1, 1b12
1b11: ADD R2, R2, R1
DEC R1
JG R1, 1b11
1b12: STORE R2, 0(R0)
HLT

```

Pseudocódigo 1

```

if (R1 <= 4) R2++;
else R2--;

```

Pseudocódigo 2

```

R3 = 0;
for(R4 = 0; R4 < 10; R4++)
    R3 += R4;

```

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 a partir de la dirección **00h**.
- Suponiendo que se ingresa por teclado el valor **03h**, realice una traza del programa A mostrando la evolución del contenido de cada registro y del PC (paso a paso), y luego describa el propósito del programa en su conjunto.
- Indique una secuencia de instrucciones para la arquitectura OCUNS, que sea equivalente al pseudocódigo 1.
- Indique una secuencia de instrucciones para la arquitectura OCUNS, que sea equivalente al pseudocódigo 2.