



**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: <i>Resolver los ejercicios en hojas separadas. Poner nombre, LU y número en cada hoja.</i>		

**Ejercicio 1.** Hacer.

**Ejercicio 2.** Hacer.

**Ejercicio 3.** Hacer.

**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	Interpretación
R1	100h	100h	500h	(2) mov R6, R	#xxxx Inmediato
R2	200h	200h	400h	(3) mov R6, (R)	R Registro
R3	300h	400h	100h	(4) mov R6, xxxx	(R) Registro indirecto
R4	400h	600h	400h	(5) mov R6, (xxxx)	xxxx Absoluto
				(6) mov R6, (R2)xxxx	(xxxx) Memoria indirecto
				(7) mov R6, @300(R)	(R)xxxx Base
					@xxxx(R) 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 **04h**, 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.