Organización de computadoras

Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
Segundo Cuatrimestre de 2017

Segundo Examen Parcial 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: $X = -1 \times 2^5 \times 1{,}486328125 = -47{,}5625$ *e* $Y = 1 \times 2^{-10} \times 2{,}1171875 = 0{,}0021820068359375$*).*

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: $X = 1 \times 2^{-3} \times 1{,}353515625 = 0{,}169189453125$ *e* $Y = -1 \times 2^{-2} \times 1{,}00390625 = 0{,}2509765625$*).*

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
(2) mov (R1), #0100
(3) mov 0100(R1), R1
(4) mov R2, #0500
(5) mov @0100(R1), #0500
(6) mov (0200), 0300
(7) mov R3, 0200
(8) mov R3, @0100(R3)

Interpretación

#xxxx	Inmediato
R	Registro
(R)	Registro indirecto
xxxx	Absoluto
xxxx(R)	Indexado
(xxxx)	Memoria indirecto
@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, lbl3
      JZ R1, lbl3
      JZ R2, lbl3
      SUB R5, R1, R2
      JG R5, lbl2
lbl1: ADD R3, R3, R2
      DEC R1
      JG R1, lbl1
      JMP R4
lbl2: ADD R3, R3, R1
      DEC R2
      JG R2, lbl2
lbl3: STORE R3, 0(R0)
      HLT
```

Op.	Descr.	Form.	Pseudocódigo
0	add	**I**	R[d] ← R[s] + R[t]
1	sub	**I**	R[d] ← R[s] - R[t]
2	and	**I**	R[d] ← R[s] & R[t]
3	xor	**I**	R[d] ← R[s] ^ R[t]
4	lsh	**I**	R[d] ← R[s] << R[t]
5	rsh	**I**	R[d] ← R[s] >> R[t]
6	load	**I**	R[d] ← mem[offset + R[s]]
7	store	**I**	mem[offset + R[d]] ← R[s]
8	lda	**II**	R[d] ← addr
9	jz	**II**	if (R[d] == 0) PC ← PC + addr
A	jg	**II**	if (R[d] > 0) PC ← PC + addr
B	call	**II**	R[d] ← PC; PC ← addr
C	jmp	**III**	PC ← R[d]
D	inc	**III**	R[d] ← R[d] + 1
E	dec	**III**	R[d] ← R[d] - 1
F	hlt	**III**	exit

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

a) Ensamblar el programa a partir de la dirección 00h.

b) 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.

c) 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.

d) ¿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?