



IIC2343 - Arquitectura de Computadores (II/2020)

Ayudantía 4

14 de Septiembre de 2020

1. Ex - 2019 - 1

Para cada uno de los siguientes programas escritos en assembly del computador básico, indica si durante la ejecución se produce el salto o no. Justifica con los valores de las señales Z, N, C y V entregados por la ALU.

I.	III.	V.
MOV A,3	MOV A,5	MOV A, 12
CMP A,4	MOV B,-5	SUB A, 1
JMP Fibonacci	XOR A,B	JCR JurgenMandaSaludos
...	JNE BonusLealtad	...
II.	IV.	VI.
MOV A,1	MOV A,7	MOV A,127
MOV B,-9	MOV B,-2	ADD A,127
SHR A, A	INC B	JOV Gemoji
JEQ Arquiyudantes	JLT SuHansidad	...
...	...	

2. I1 2018 - 1

Considera el código en assembly que se muestra más abajo.

- Explica el propósito de, y cómo funciona, la secuencia de instrucciones PUSH B, CALL func y POP B; y cuál es el efecto de que no haya una instrucción PUSH A (justo antes) ni una instrucción POP A (justo después).
- Explica el propósito de, y cómo funciona, la secuencia de instrucciones CMP A,0 y JEQ end, y en particular qué ocurre cuando el contenido del registro A es 0.

Tanto en (a) como en (b), identifica las componentes del computador básico que están involucradas y explica qué rol juegan.

DATA: r 3 CODE : MOV A, (r) MOV B, 2 PUSH B CALL func POP B JMP finish finish:	func : shift: MOV A,B CMP A, 0 JEQ end DEC A MOV B,A MOV A, (r) SHL A,A MOV (r),A JMP shift end : RET
---	--

Figura 1: Código pregunta 2.

3. I1 2019-2

El siguiente programa escrito en python implementa un algoritmo de conteo de los bits distintos entre dos números. Traduzca el programa al assembly del computador básico, cumpliendo lo siguiente:

- Su programa debe ser una traducción directa del programa en python, es decir, además de traducir la funcionalidad (conteo de los bits distintos) debe hacerlo implementando el mismo algoritmo descrito en el programa, incluyendo las funciones como subrutinas.
- Su programa debe definir (al menos) las variables var_a, var_b que se ocuparán como operandos para el conteo de los bits distintos, y la variable var_c que almacenará el resultado.

```

1 def contar_bits_seteados(n):
2     cuenta = 0
3     while n!=0:
4         cuenta += n & 1
5         n >>= 1
6     return cuenta
7
8 def contar_bits_distintos(var_a, var_b):
9     return contar_bits_seteados(var_a^var_b)
10
11
12 # Programa principal
13 def main():
14     var_a = 10
15     var_b = 20
16     var_c = 0
17     var_c = contar_bits_distintos(var_a,var_b)
18
19 if __name__ == "__main__":
20     main()

```