



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

IIC2343 – Arquitectura de Computadores

## Ayudantía 2

Profesores: Hans-Albert Löbel Díaz, Jorgen Dieter Heysen Palacios

Ayudante: Germán Leandro Contreras Sagredo (glcontreras@uc.cl)

### Temas a tratar

Los temas a tratar dentro de esta ayudantía son:

- Almacenamiento de datos.
- Programabilidad.

### Preguntas

1. a. (I1 - I/2018) Construya un circuito que obtenga la siguiente tabla de verdad, pero solo usando compuertas OR y NOT:

R	S	$Q(t+1)$	$\overline{Q(t+1)}$
0	0	-	-
1	0	0	1
0	1	1	0
1	1	$Q(t)$	$\overline{Q(t)}$

**Cuadro 1:** Tabla de verdad del *latch* RS.

- b. (I1 - I/2013) Diseñe, utilizando todos los elementos de circuitos lógicos vistos en clases, un contador secuencial de 2 bits que se incrementa con cada flanco de subida de la señal de control.
- c. (I1 - II/2012) Implemente mediante compuertas lógicas, elementos de control y *flip-flops*, una memoria RAM de 16 palabras de 1 byte.
- d. (I1 - I/2018) Diseñe un registro de 1 byte haciendo uso de *flip-flops* D y una señal de control L que habilite la sobreescritura del estado  $Q$  solo cuando  $L = 1$ . Luego, explique cómo puede modificar el circuito realizado para acceder individualmente (tanto para lectura como escritura) a cada uno de los bits del componente.

2. a. (I1 - I/2016) ¿Cuál es la frecuencia máxima que puede tener el *clock* del computador básico? ¿Qué pasa si un *clock* con una frecuencia mayor a la máxima es conectado al computador básico?
- b. Diseñe un circuito que le permita a la unidad de control del computador básico identificar un *opcode*, retornando una señal de 1 bit que indique si corresponde a la instrucción esperada o no.
- c. Considere el siguiente programa:

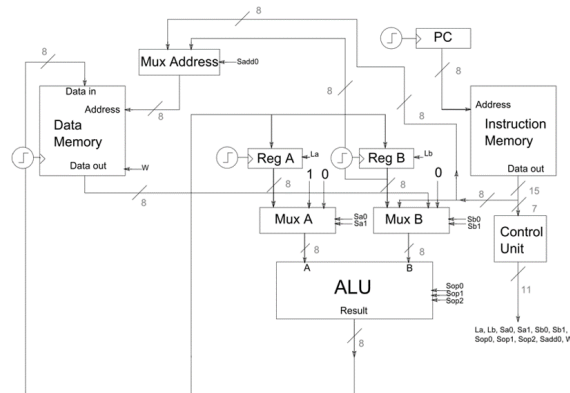
```

x = 2
y = 4
z = 0 # Variable auxiliar
z = x
x = x + y
y = y - z

```

En base a este:

- I. Construya un programa en **Assembly** que obtenga el mismo resultado (considere que  $x$ ,  $y$  y  $z$  parten con sus valores almacenados en memoria).
- II. Ahora, programe en **Assembly** un código que obtenga el mismo resultado de  $x$  e  $y$ , pero sin hacer uso de la variable  $z$  en el segmento **DATA**.
- III. A partir de los programas anteriores, explique el flujo resultante de cada uno de ellos en el diagrama del computador básico que se muestra a continuación.



**Figura 1:** Computador básico visto hasta ahora.

- d. (I1 - I/2016) ¿En qué casos es posible soportar la instrucciones **ADD B,Lit** en el computador básico, sin modificar su *hardware* ni sobrescribir datos? Para los casos negativos, indique qué modificaciones al *hardware* y/o **Assembly** se deberían hacer para soportarla.
- e. (I1 - II/2015) Modifique el diagrama del computador básico de manera que soporte la ejecución de la instrucción **GOTO dir**, que fuerza que la siguiente instrucción en ejecutarse sea la ubicada en la dirección **dir**.
- f. (I1 - I/2018) Modifique el computador básico para que acepte el comando **MOV A, [DIR]**, que toma el valor almacenado en la dirección **DIR** y luego considera ese valor como una dirección, va a esa dirección y almacena su valor en **A**.