



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

IIC2343 – Arquitectura de Computadores

Ayudantía 1

Profesor: Yadran Francisco Eterovic Solano

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

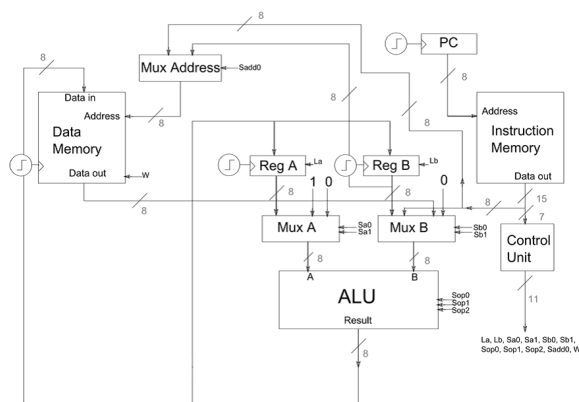
Temas a tratar

Los temas a tratar dentro de esta ayudantía son:

- Programabilidad
- Saltos y subrutinas

Precalentamiento

1. (I1 - II/2017) Considere el siguiente diagrama de bloques del computador básico, aún incompleto.



- a. Explica el rol del multiplexor *Address*, da un ejemplo de su funcionamiento.
- b. Explica qué es necesario agregar para permitir instrucciones de tipo salto incondicional; y explica cómo funcionaría en ese caso un salto incondicional.

- c. Explica qué es necesario agregar a tu respuesta en **b.** para permitir instrucciones de tipo salto condicional; y explica cómo funcionaría en ese caso un salto condicional.
2. Considere el siguiente programa:

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

En base a este:

- a. Construya un programa en **Assembly** que obtenga el mismo resultado (considere que x , y y z parten con sus valores almacenados en memoria).
 - b. 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**.
 - c. A partir de los programas anteriores, explique el flujo resultante de cada uno de ellos en el diagrama del computador en **1**.
3. **(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.
4. **(Examen - I/2017)** Modifique la arquitectura del computador básico para que el registro **STATUS** se actualice solo después de la ejecución de una instrucción **CMP**.
5. **(Apuntes - Saltos y subrutinas)** ¿Cómo se podría implementar en el computador básico la opción de que este avise luego de realizar una operación cuando el resultado es par o impar?
6. ¿Cuántos ciclos toma llamar una subrutina? ¿y cuántos toma retornarla? Justifique.

Preguntas

1. 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. **(I1 - I/2016)** Modifique el computador básico para dar soporte a la instrucción **CLEAR**, que setea en 0 el valor de todos los registros (excepto **PC** y **STATUS**) y el de todas las palabras de la memoria de datos. Indique la(s) señal(es) de control de la nueva instrucción.
- c. **(Propuesto) (Examen - II/2016)** Modifique la arquitectura del computador básico para que funcione con lógica ternaria en vez de binaria. Más específicamente, modifique los tamaños de los elementos (buses, registros, señales de control, etc.) de modo que el nuevo computador tenga una capacidad similar a la versión binaria. Asuma que existen todos los componentes vistos en clases en versión ternaria. **Nota:** No es válido utilizar los valores ternarios como si fueran binarios.

2. a. **(I1 - I/2016)** ¿Qué pasaría si se quita el registro **STATUS** del computador básico y se conectaran directamente las señales **ZNCV** a la unidad de control?
- b. **(I1 - I/2017)** Si se elimina la instrucción **CMP** del computador básico, ¿cómo deben modificarse las instrucciones de salto, sin alterar el hardware, para que estas no dependan del resultado de la última instrucción ejecutada? Escriba detalladamente todas las modificaciones necesarias y sus implicancias. Asuma que solo es necesario resolver el caso de la comparación de los registros A y B y que no es posible sobrescribir los registros para realizar la comparación.
- c. **(Propuesto)** Diagrame y explique una posible implementación de la instrucción **JGT Dir** (opcode 1000000) al interior de la unidad de control.
- d. **(Propuesto)** A partir del siguiente código Assembly, explique el flujo resultante dentro del computador básico, indicando el valor final de los registros A y B:

```

DATA:
    r 3 ; Resultado final
CODE:
    MOV A,(r)
    MOV B,2
    PUSH B
    CALL func
    POP B
    JMP 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
finish: ; Termina el programa

```