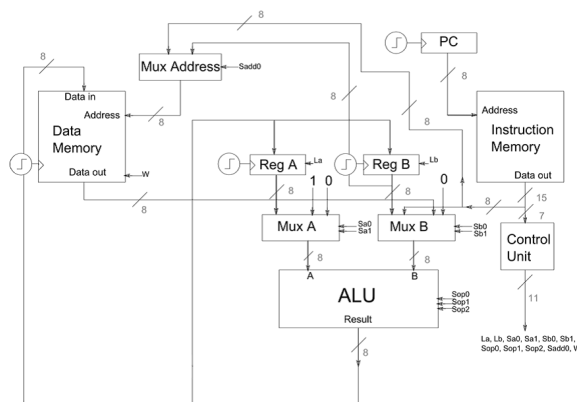


Ayudantía 3

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

- Programabilidad
- Saltos y subrutinas

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



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

- b. ¿Cuántos ciclos toma llamar una subrutina? ¿y cuántos toma retornarla? Justifique.
- c. **(Examen - II/2012)** ¿Qué implicancia tiene en el tamaño de los programas el eliminar la conexión entre memoria de datos y PC (*program counter*) en el computador básico?
- d. **(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?
- e. **(II - 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?
- f. **(II - 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.
- g. **(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**.
- h. **(II - I/2018)** Modifique el *hardware* del computador básico para que las instrucciones **RET** y **POP** tomen un solo ciclo.
- i. 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

```

2. a. **(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.
- b. **(II - II/2016)** En esta pregunta deberá diseñar un computador especializado en el manejo de matrices. El computador debe ser capaz de: i) copiar una matriz desde la memoria de datos a un registro y viceversa, ii) sumar 2 matrices almacenadas en registros distintos y almacenar el resultado en un registro.
 - I. Haga el diagrama del computador, considerando que las matrices pueden tener como máximo $N \times N$ elementos, cada uno de 1 byte.
 - II. Diseñe el *assembly* del computador. Cada instrucción debe estar asociada a un *opcode* y estos a sus respectivas señales de control.
 - III. Agregue tanto al *hardware* como al *assembly* soporte para una instrucción que permita modificar el valor de un elemento arbitrario de una matriz almacenada en la memoria de datos.
- c. **(II - I/2018)** El computador básico solo trabaja con números enteros, sin embargo, para muchas aplicaciones es útil poder trabajar con números decimales. En consecuencia, usted deberá añadir soporte para números decimales al computador básico.
 - I Indique un esquema de números decimales que podría emplear en el computador básico, mencionando sus ventajas y desventajas.
 - II Añada los componentes necesarios para trabajar con el esquema de números decimales argumentados en la parte I., indicando qué hace cada componente y cómo se conecta a las partes existentes del computador básico, detallando su interacción con este. Si bien no es necesario hacer a nivel de compuertas todos los componentes, no se aceptarán componentes mágicos como “Unidad de cómputo decimal”. Sí se aceptará la ALU, registros de n bits y otros componentes desarrollados en clases.