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

IIC2343 - Arquitectura de Computadores (I/2026)

Ayudantía 6: Repaso I1

Ayudantes: Alberto Maturana (alberto.maturana@uc.cl), Mario Rojas (mario.denzel@estudiante.uc.cl)

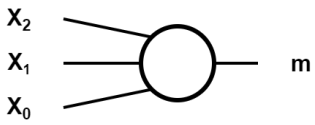
Pregunta 1: Representaciones Numéricas

(a) Escriba la representación en decimal del resultado de la operación 0xF0CA + 0x0F30. Este resultado debe interpretarse como binario en complemento de dos.

(b) Considere una función $C_B(x)$ definida para un número x en base B, tal que $x + C_B(x) = 0$, donde la suma se realiza dígito a dígito en base B y sobre una cantidad fija de cifras. Esta función representa el **complemento en base** B de x. Sabiendo que $C_2(x)$ corresponde al complemento a dos de x (estudiado en clases), defina paso a paso el procedimiento para calcular el complemento en base 4 de un número x, es decir, $C_4(x)$. Además, considere que la representación del cero debe ser única, formada exclusivamente por ceros en todos sus dígitos, evitando representaciones alternativas (como ocurre con el complemento a uno). Justifique cómo su procedimiento garantiza que se cumple $x + C_4(x) = 0$ en base 4, utilizando una cantidad fija de dígitos.

Pregunta 2: Lógica Digital I1-2024-1

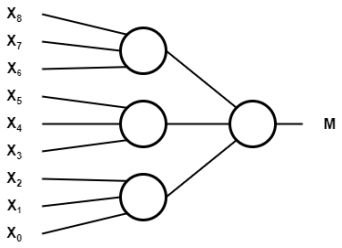
Definiremos una nueva compuerta lógica llamada `Mayoria`. Esta recibirá tres señales de entrada de 1 bit y su salida, de 1 bit, corresponderá al valor que más se repite. Por ejemplo, si tenemos $X_0 = 0$, $X_1 = 0$, $X_2 = 1$, la salida de la compuerta `Mayoria` es 0.



A partir de esta compuerta, responda los incisos (a) y (b).

(a) Diseñe la compuerta lógica `Mayoria`, de 3 entradas, a partir de las siguientes compuertas: `AND`, `OR`, `NOT`, `XOR`, `NAND` o `NOR`.

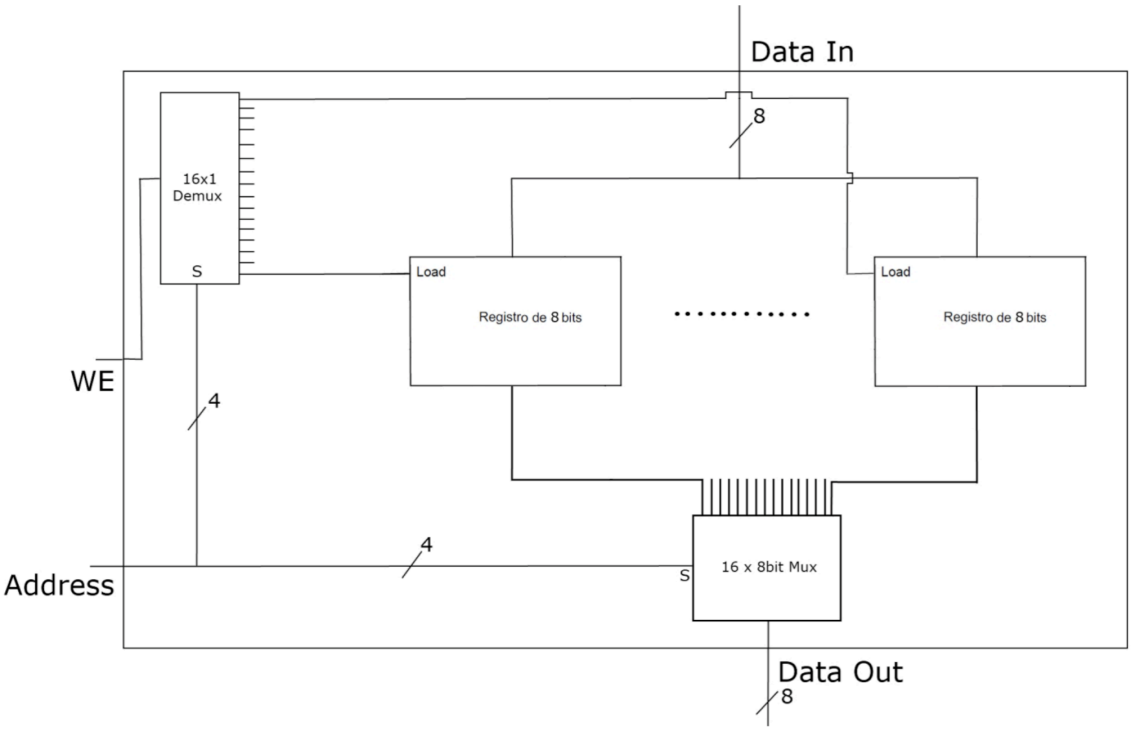
(b) Haciendo uso de la compuerta `Mayoria`, se propone el siguiente circuito para determinar el valor que más se repite entre 9 *inputs*:



Indique si el circuito propuesto resuelve correctamente el problema planteado. Si lo resuelve, justifique por qué; si no lo hace, indique en qué casos no funciona.

Pregunta 3: Almacenamiento: I1 2023-2

A continuación, se adjunta un posible diagrama interno de una RAM de 16 direcciones:

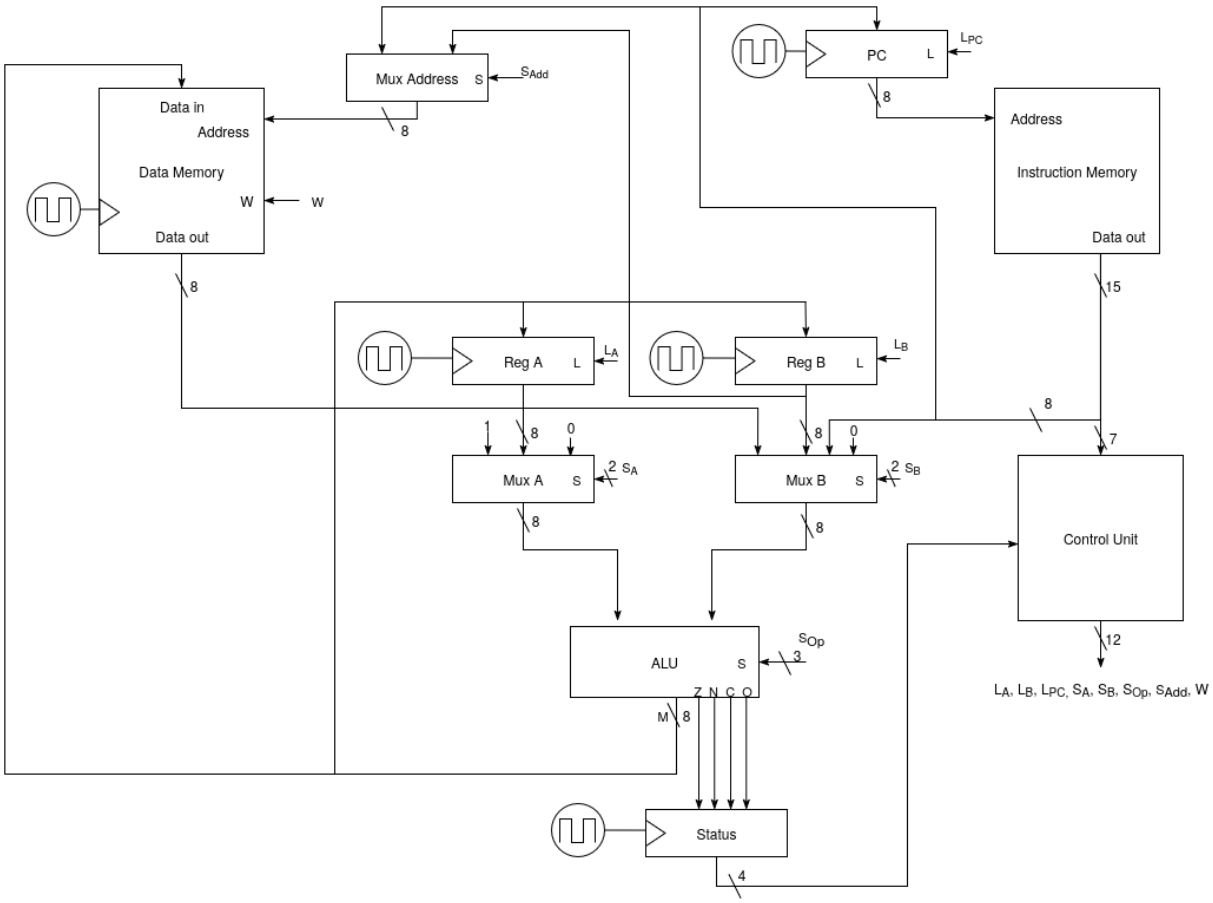


Modifique este diagrama para que reciba entradas $\texttt{DataIn}_1$; $\texttt{DataIn}_2$; $\texttt{Address}_1$; $\texttt{Address}_2$; $\texttt{WE}_1$; $\texttt{WE}_2$, y salidas $\texttt{DataOut}_1$ y $\texttt{DataOut}_2$ para leer y escribir datos en dos direcciones de memoria distintas de forma simultánea.

Pregunta 4: Computador Básico con saltos

(a) Modifique la arquitectura del computador básico para que el registro `STATUS` se actualice solo después de la ejecución de una instrucción `CMP`.

(b) ¿Cómo se podría implementar en el computador básico una señal para que esta indique luego de realizar una operación cuando el resultado es par o impar?



(c) En los ítems **(I)** y **(II)**, se le describirán instrucciones que deberá implementar en el computador básico a partir del diagrama adjunto. Para cada una de ellas, debe incluir la combinación **completa** de señales que las ejecutan. En las señales de carga/escritura/incremento/-decremento, debe indicar si se activan (1) o no (0); en las señales de selección, debe indicar el **nombre** de la entrada escogida ("-" si no afecta). Debe realizar todas las modificaciones de *hardware* necesarias en un solo diagrama. Además, puede implementar instrucciones sin modificar el *hardware* si es que la arquitectura se lo permite, indicando la forma de hacerlo a partir de las señales de control. **Cada instrucción debe ejecutarse en un ciclo**.

I. `SWAP A,B`. Almacena en el registro `A` el valor del registro `B`; y almacena en el registro `B` el valor del registro `A`, *i.e.* intercambia sus valores.

II. `JGTCMP A,B,label`. Salta a la dirección asociada a `label` si, y solo si las *flags* `Z` y `N` provenientes del registro `Status` son iguales a 0 y, al mismo tiempo, ejecuta la operación `A - B` en la ALU sin actualizar los registros `A` y `B`, independiente de si hay salto o no.

(d) A continuación, se adjuntan dos fragmentos de código **(1)** y **(2)** escritos en el Assembly del computador básico del curso, con la diferencia de que **(2)** incorpora las instrucciones implementadas en los ítems anteriores. Para cada fragmento, indique los valores de los registros `A` y `B` al finalizar la ejecución del código. Si no incluye desarrollo o no justifica cómo llegó a los valores obtenidos, **no se otorgará puntaje**.

(1)

```
DATA:
  var1 -1
  var2 1
  var3 -1
  var4 1
CODE:
  MOV A,125  ; 125 = 01111101b
  NOT A,A
  SHL A,A
  SHR A,A
  MOV B,A
  MOV A,(B)
  JGT case_1
  JMP case_2
  case_1:
    MOV A,-1
    JMP end
  case_2:
    MOV A,1
  end:
```

(2)

```
DATA:
  var1 5
  var2 16
CODE:
  init:
    MOV A,(var1)
    MOV B,(var2)
    CMP A,B
    JGTCMP A,B,case_1
    JLT case_2
    case_1:
      JMP continue
    case_2:
      SWAP A,B
    continue:
      SWAP A,B
      NOT A,A
      SWAP A,B
      INC B
      ADD A,B
```

Feedback ayudantía

Escanee el QR para entregar feedback sobre la ayudantía.

