

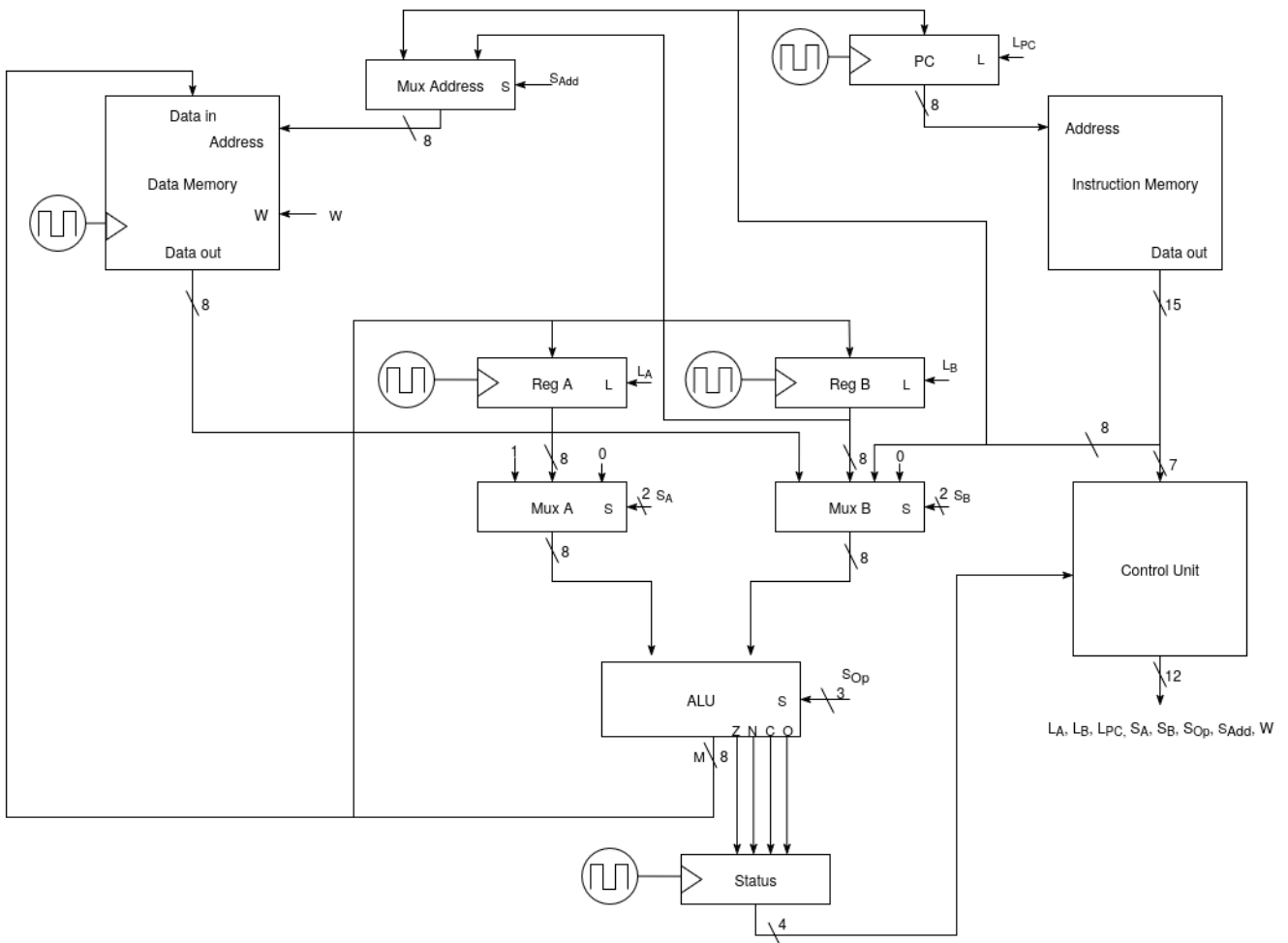


IIC2343 - Arquitectura de Computadores (II/2025)

Ayudantía 5

Ayudantes: Daniela Ríos (danielaarp@uc.cl), Alberto Maturana (alberto.maturana@uc.cl), José Mendoza (jfmendoza@uc.cl)

Computador básico



Pregunta 1: Assembly - Saltos

- (a) Describa qué hace el fragmento de código, escrito en el Assembly del computador básico del curso. Indique los valores de los registros A y B al finalizar la ejecución del código.

```
DATA:
  x -5
  r 2

CODE:
  MOV A,(x)
  CMP A,0
  JEQ zero
  CMP A,0
  JLT neg
  JMP power

zero:
  MOV A,(r)
  SHR A,A
  MOV (r),A
  JMP end

neg:
  NOT A,A
  ADD A,1
  JMP power

power:
  CMP A,1
  JEQ end
  MOV B,A
  MOV A,(r)
  SHL A,A
  MOV (r),A
  MOV A,B
  SUB A,1
  JMP power

end:
  MOV A,(r)
```

- (b) Programe en *assembly* un código que calcule la multiplicación de dos números positivos. Para ello, se proporciona la sección **DATA**, la cual contiene los números x e y . Guarde el resultado final en la dirección **r**.

```
DATA:
  r 0
  x 9
  y 7
```

Pregunta 2: Análisis - Saltos

El siguiente fragmento, escrito en el Assembly del computador básico del curso, se encarga de revisar si un arreglo **arr1** es igual al doble del inverso de otro arreglo **arr2**, es decir, que el primer elemento de **arr1** sea igual al doble del último elemento de **arr2**; que el segundo elemento de **arr1** sea igual al doble del penúltimo elemento de **arr2**, y así sucesivamente. En caso de cumplirse, se tendrá que **res == 1** al finalizar la ejecución, y **res == 0** en otro caso. No obstante, el código da un cómputo equivocado. Identifique qué instrucción(es) está(n) errada(s) y explique justificadamente cómo corregirla(s) para que el código actualice **res** correctamente.

DATA:

```
res    1
len    4
arr1   6
        10
        16
        4
arr2   2
        8
        5
        3
iter   0
i1     0
i2     0
temp   0
```

CODE:

```
MOV A, arr1
MOV (i1), A
MOV A, arr2
ADD A, (len)
MOV (i2), A

loop:
MOV B, (i1)
MOV A, (B)
SHR A,A
MOV (temp), A

MOV B, (i2)
MOV A, (B)
CMP A, (temp)
JNE no_son_iguales

INC (iter)
INC (i1)
MOV A, (i2)
SUB A, 1
MOV (i2), A

MOV A, (iter)
CMP A, (len)
JEQ end
JNE loop

no_son_iguales:
MOV A, 0
MOV (res), A

end:
```

Pregunta 3: Modificación computador básico con saltos

Modifique la arquitectura del computador básico para implementar las instrucciones:

- `MOV A, (B+offset)`
- `MOV (B+offset), A`
- `MOV B, (B+offset)`
- `MOV (B+offset), B`

Es decir, instrucciones de direccionamiento indirecto con registro B y *offset*, siendo este último un literal. Para cada instrucción, deberá incluir la combinación completa de señales que la ejecutan. Por cada señal de carga y escritura, deberá indicar si se activan (1) o no (0); en las señales de selección, deberá indicar el nombre de la entrada escogida (“-” si no afecta).

Feedback ayudantía

Escanee el QR para entregar feedback sobre la ayudantía.

