

Taller entrega 2

IIC2343

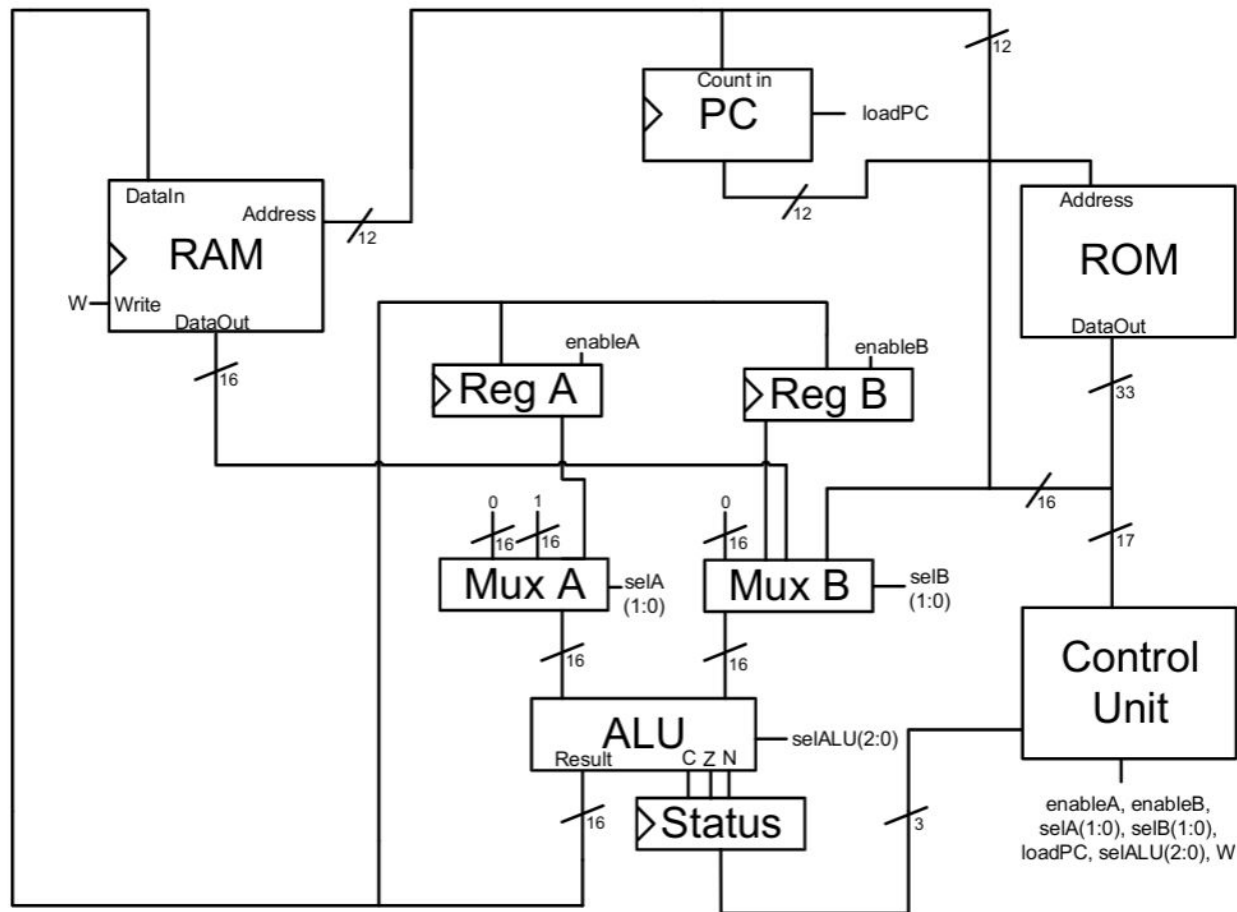
Formalidades

- ° La entrega es el 8 de Abril **20:00 hrs**
- ° La calificación de esta entrega equivale al 50% de la nota final de proyecto, y se puede eliminar.
- ° Esta entrega se puede eliminar (no lo recomendamos, pero se puede)
- ° Es posible sacarse un 7 aún si no cumple

Contenidos

- Componentes de esta entrega (Reg, ROM y RAM)
- Pasos de desarrollo
- Assembler
- Simulación y testeo de esta entrega
- Formato de entrega
- Parte Práctica en el segundo módulo: Simulación de componentes

Diagrama:



Componentes de esta entrega: Reg

- ° Componente ya familiarizado de la entrega pasada.
- ° Para los registros A y B no son necesario activar las señales up y down)se pueden dejar en cero)
- ° Usarlo para crear otros registros, como Status y PC

clock	Entradas			Salidas	
	load	up	down	datain	dataout
Flanco Subida	1	*	*	*	datain
Flanco Subida	0	1	*	*	dataout + 1
Flanco Subida	0	0	1	*	dataout - 1
*	*	*	*	*	dataout

Figura 4: Tabla de verdad de Reg.

Componentes de esta entrega: ROM

° Componente visto la semana pasada.

° Dentro de este componente habrá un arreglo de vectores, que representan una memoria de lectura. Los valores de dicho arreglo son explicitados, y serán utilizados para representar las instrucciones del programa en su código de máquina.

° Recordar que la estructura de cada una de las instrucciones de 33 bits queda a criterio de cada grupo.

Entradas	Salidas
address	dataout
*	Mem[address]

Figura 5: Tabla de verdad de la ROM.

Componentes de esta entrega: RAM

- ° Componente nuevo con el que se familiarizan en esta entrega 😊
- ° Similar a la ROM, es un arreglo de vectores, que representan una memoria que se puede escribir en flanco de subida.
- ° Ocupa la señal del clock clk

- RAM

Entradas				Salidas
clock	write	address	datain	dataout
Flanco Subida	1	*	*	Mem[address] = datain
*	*	*	*	Mem[address]

Figura 4: Tabla de verdad de la RAM.

Pasos de Desarrollo: Componentes

° Crear e importar los componentes que se especifican en el enunciado en el archivo Basys3, procurando conectar correctamente las señales de entradas y salidas.

Componentes:

- **ALU**: Idéntica a la entrega pasada.
- **Status, PC**: hacerlo en base a componente Reg.
- **Control Unit (CU)**: componente que procesa las palabras provenientes de la ROM y Status. para generar señales que afectan el comportamiento de todos los componentes de la CPU.
- **MUX**: Selectores con with/select, de la misma forma que la entrega pasada.

Pasos de Desarrollo: Opcodes y Señales

- ° Analizar y crear la forma como la ROM entrega las instrucciones a la Control Unit. (Cuántos bits ocupar, que significará cada bit, o grupo de bits, ¡ocupen sus bits de forma inteligente!)
- ° Se recomienda usar tablas que identifica el código de operación y sus respectivas señales asociadas.

INSTRUCCIÓN	SEÑAL 1	SEÑAL 2	...	SEÑAL N
MOV A,B	1	0		1

Pasos de Desarrollo: Comportamientos de la CU

- ° El comportamiento de su Control Unit dependerá del diseño de su opcode
- ° Es importante que el diseño de su palabra de control sea inteligente. Un mal diseño de palabra puede significar más trabajo en futuras entregas (rehacer la control unit desde cero para una futura entrega)
- ° Se recomienda simular este componente con distintos opcodes para asegurarse de que da las señales esperadas

Pasos de Desarrollo: Assembly a Binario

° Existen principalmente dos formas de pasar Assembly a Binario:

- 1.- Manualmente (No recomendado)
- 2.- Utilizando un Assembler (Recomendado)

Pasos de Desarrollo: Assembly a Binario

Manera manual:

> Pros:

- No usan Assembler.

> Contras:

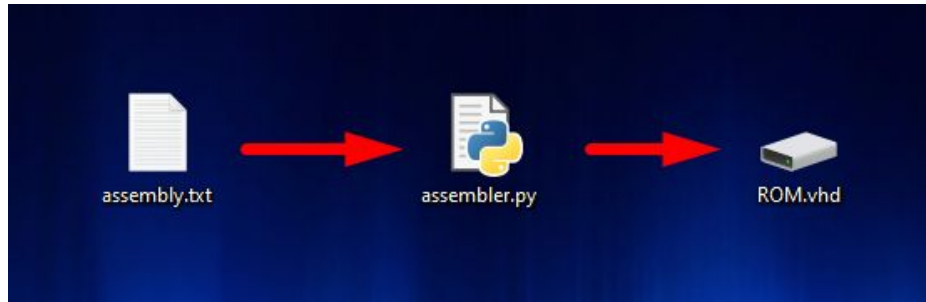
- Transcripción toma mucho tiempo.
- Propenso a error humano.
- Seguimiento de errores es costoso en tiempo.

Pasos de Desarrollo: Assembly a Binario

2.- El uso de un Assembler consiste en que ustedes creen un programa/script que automatice la transcripción de assembly a binario.

¡El assembler no es evaluado!

```
C:\>python assembler.py assembly.txt ROM.vhd
```



Pasos de Desarrollo: Assembly a Binario

Uso de Assembler:

> Pros:

- Pueden programarlo en su lenguaje de preferencia
- Transcripción assembly-binario prácticamente instantánea.
- Fácilmente expandible

> Contras:

- Tienen que programarlo...

Pasos de Desarrollo: Assembly a Binario

Uso de Assembler:

Nota: La funcionalidad de su assembler depende de ustedes!

- Si les basta con que genere texto para copiar-pegar, esta bien ✓
- Si quieren que su assembler sea más sofisticado y realice el manejo de archivos, también está bien ✓

En conclusión: Es su assembler, hecho a su manera, para ustedes 😊

Pasos de Desarrollo: Simulación

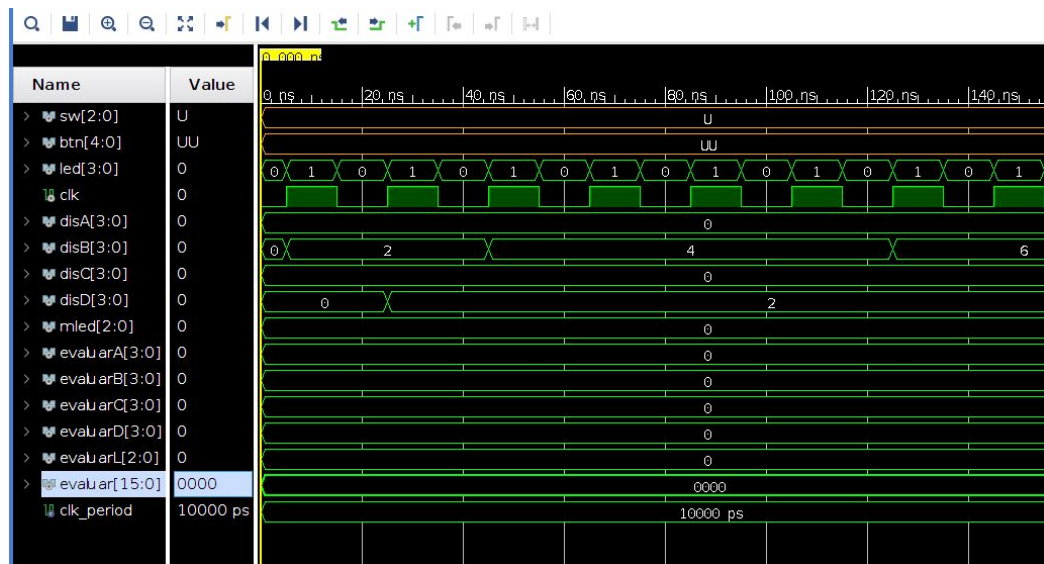
° Finalmente, teniendo todo lo anterior listo, deberán simular y observar si los valores transitivos y finales de los display corresponden a los valores esperados o no.

° Para esto se le entregará un archivo de simulación listo con explicaciones de un archivo de simulación listo con explicaciones de cómo deberán probar sus test. (Estos serán subidos en el transcurso de la semana).

Pasos de Desarrollo: Simulación

° Ejemplo 1 del enunciado en simulación:

```
-- INICIO
evaluarA <= (disA xor "0000");
evaluarB <= (disB xor "0000");
evaluarC <= (disC xor "0000");
evaluarD <= (disD xor "0000");
wait for 20 ns;
-- MOV A,2
evaluarA <= (disA xor "0000");
evaluarB <= (disB xor "0010");
evaluarC <= (disC xor "0000");
evaluarD <= (disD xor "0000");
wait for 20 ns;
-- MOV B,2
evaluarA <= (disA xor "0000");
evaluarB <= (disB xor "0010");
evaluarC <= (disC xor "0000");
evaluarD <= (disD xor "0010");
```



Lo que deben entregar

- ° En sus repositorios de Github deben subir la carpeta del proyecto de Vivado
- ° En dicho debe estar contenido los archivos srcs, con las entidades desarrolladas a lo largo de esta entrega y las anteriores
- ° Un informe en formato Markdown donde se detalla lo que hizo cada persona como también muestras de las simulaciones de que su entidad funciona correctamente.
- ° Los archivos ROM, un archivo por cada programa de ejemplo.

Evaluación de Pares

- ° Esta evaluación se hará durante el desarrollo de la entrega hasta el miércoles 6 de mayo a las 23:00 horas
- ° El objetivo es poder lograr una comunicación efectiva con todo el equipo, y que si sea necesario una intervención del cuerpo docente sea antes de entregar para evitar perjudicar la nota final
- ° Si presenta alguna complicación no dude en informar lo más pronto posible para poder tomar las medidas pertinentes.

¿Dudas?