Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2343 – Arquitectura de Computadores (I/2020)

Proyecto Semestral: Entrega Práctica 01

ALU de 16 bits

Fecha de entrega: Viernes 03 de Abril a las 20:00 horas

1. Objetivo

La entrega práctica consiste en el diseño, simulación y aplicación en placa de una mini-calculadora de 16 bits cuyo funcionamiento y especificaciones se detallan a continuación: la idea es hacer un circuito capaz de realizar operaciones básicas sobre números de 16 bits y que entregue un resultado de 16 bits más un "carry out". En todo momento se mostrarán en los display de siete segmentos los 8 bits menos significativos de cada operando en hexadecimal (00 a FF), como se ve en la Figura 1:

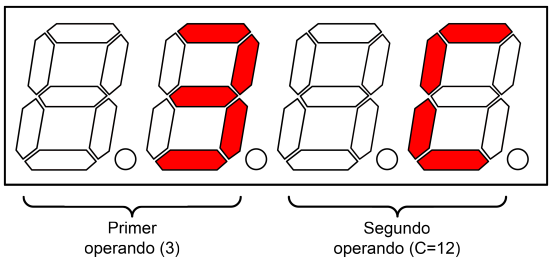

Figura 1: Distribución inicial de los operandos en el display. Nota: Esto será simulado

Para ingresar los operandos al sistema se deben utilizar una combinación de los botones de la placa (simulados): el botón izquierdo más el superior o inferior para aumentar o disminuir de a una unidad el primer operando y el derecho más el superior o inferior para aumentar o disminuir de a una unidad el segundo operando.

Al completar la entrega, la mini-calculadora de 16 bits debe ser capaz de ejecutar las siguientes operaciones considerando ambos operandos: Suma, Resta, And, Or y Xor. Además de lo anterior, debe ser capaz de realizar operaciones adicionales considerando solo el primer operando: Not, ShiftR (1 bit) y ShiftL (1 bit). Al activarse los flags de la ALU (Z, N y/o C) y la señal "clk" deben encenderse correspondientemente los 4 leds disponibles en ese orden. Las operaciones no pueden ser generadas con librerías adicionales, deben usar los operandos lógicos.

Para seleccionar qué operación realizar se deben utilizar los switchs y para ver el resultado de esta operación sin alterar los operadores se debe presionar el botón central (simulado). Cabe destacar que en el display (simulado) se deben mostrar los 16 bits del resultado. Para ilustrar el uso de los botones y switchs de la

placa consideremos el caso que se describe a continuación.

Ejemplo: Supongamos que queremos sumar los números 12 y 9 y que inicialmente los operandos tienen los valores de la Figura 1. Para esto lo primero que tenemos que hacer es realizar la acción de aumentar (con los botones destinados para esto) nueve veces para el primer operando (esto debido a que estaba en 3 y necesitamos dejarlo en 12), luego realizar la acción de disminuir y modificar el valor del segundo operando (dado que el operando estaba inicialmente en C, debemos apretar sus correspondientes botones tres veces para que quede en 9). Una vez hecho esto y solo mientras se mantiene presionado el botón central de la placa, dependiendo de la posición de los switchs de "operación", en los últimos dos dígitos del display se podrá ver el número $C + 9 = 15$ (21 decimal) .

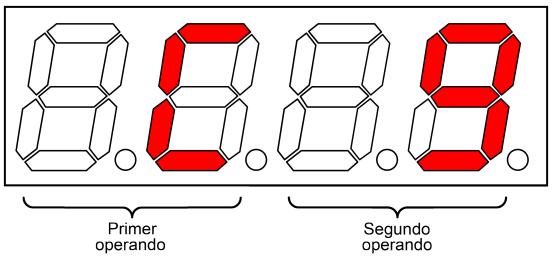

Figura 2: Distribución final de los operandos en el display. Nota: Esto será simulado en Vivado

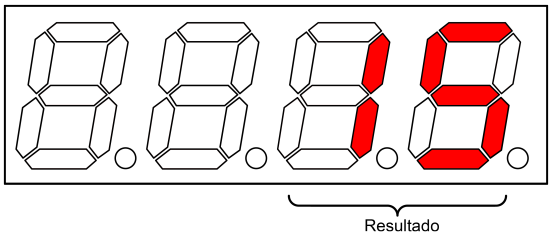

Figura 3: Distribución del resultado en el display para la operación de suma.

1.1. Módulos entregados

Se les entregarán un módulo VHDL que deberán utilizar para desarrollar su entraga:

- `Reg.vhd`

  Este módulo recibe un vector de 16 bits y 4 valores lógicos `clock`, `load`, `up` y `down`. Este componente funciona como un registro 16 bits de flanco de subida.

  La entrada de `clock` corresponde a una señal que va variando de valor encendido y apagado con con una frecuencia determinada y constante.

  Este componente funciona con flanco de subida, el cuál se define como la transición del nivel bajo (cuando la señal esta apagada) a uno alto (cuando la señal esta activada).

  Cuando la entrada `load` este activada y se encuentre la entrada `clock` en un flanco de subida, se guardara en el registro el vector de entrada. Por otro lado, si la entrada `up` este activada y se encuentre la entrada `clock` en un flanco de subida, se guardara en el registro el sucesor del valor previamente almacenado en el registro. Finalmente, si la entrada `down` este activada y se encuentre la entrada `clock`

en un flanco de subida, se guardara en el registro el antecesor del valor previamente almacenado en el registro. Inicialmente el valor almacenado en este registro es igual a cero.

- `Reg`

Entradas					Salidas
clock	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.

2. Desarrollo

Para desarrollar esta entrega se recomienda a los grupos realizar los siguientes pasos:

1. **Operandos:** Crear las instancias de los registros y conectarles el display y los botones para que puedan ver y modificar los operandos. Usar la señal "clock" para los registros.
2. **Resultado:** Modificar la coneccion del display para que el boton central permita ver el resultado.
3. **Operaciónes:** Agregar y completar la ALU, conectando los switches para poder seleccionar la operación y los leds para ver los flags. La ALU debe cumplir con lo especificado en la Figura 5 y debe ser desarrollada usando los conocimientos adquiridos en clases.

Entradas			Salidas			
a(15:0)	b(15:0)	sel	result(15:0)	c	z	n
*	*	add	a + b	result < a + b	result = 0	0
*	*	sub	a - b	a >= b	result = 0	b > a
*	*	and	a and b	0	result = 0	0
*	*	or	a or b	0	result = 0	0
*	*	xor	a xor b	0	result = 0	0
*	*	not	not a	0	result = 0	0
*	*	shr	0 & a(15 downto 1)	a(0)	result = 0	0
*	*	shl	a(14 downto 0) & 0	a(15)	result = 0	0

Figura 5: Tabla de verdad de la ALU.

Finalmente deben generar un solo circuito que integre todo lo pedido y que cumpla con la función detallada en la sección 1.

Requerimientos

Para implementar declaraciones condicionales **solamente** se permite hacer uso de bloques `with/select`. El uso de los *statements process*, `case` e `if/else` quedan absolutamente prohibidos. Esto porque se privilegia el uso de selectores y operaciones lógicas básicas para el desarrollo de esta tarea. **Para esta entrega, queda estrictamente prohibido utilizar cualquier tipo de librería aritmética que simplifique la tarea de la suma**.

Nota: Esto solo aplica a los componetes a desarrollar, en los archivos de simulación se permite el uso de process.

- Crear el proyecto
  - Seleccionar las opciones correctas para crear el proyecto en Vivado, que funcione con la placa correspondiente.
  - Importar correctamente el archivo `Basys3.xdc` .
  - Configurar correctamente las *constrains* del archivo `Basys3.xdc`. Descomentando las líneas correctas del archivo.
- (6 pts) Crear el módulo `Basys3.vhd`
  - (3 pts) Crear una *source* llamada `ALU`, que contiene la arquitectura requerida para resolver el problema, la cual será ocupada por Basys3.
  - (1 pts) Uso correcto de los del componente `Reg.vhd` para desarrollar el problema.
  - (1 pts) Correcta conexión de los comportamiento de la entidad Basys3 para desarrollar el problema
  - (1 pts) Contenido del informe
  - Puede crear más *sources* para facilitar el problema
- **Incluir el un archivo Infome.md :** En este debe describir trabajo realizado por cada uno de los miembros de su grupo. Se debe indicar específicamente que hizo cada integrante del grupo y deben explicar que fue lo mas difícil para el grupo en la entrega.

  Además se deberá incluir resultados de los archivos de simulación mostrando que su entidad efectivamente resuelve el problema. Para esto basta con mostrar un caso no trivial (como sería la suma de dos ceros, el shif de cero, etc) para evidenciar que funciona. **Nota:** se recomienda hacer más casos para probar su circuito, pero a la hora de escribir el informe un solo caso basta.

Evaluación de pares

La evaluación de pares estará activa durante hasta el miércoles 01 de Abril a las 23 horas. El link es el siguiente:

`https://forms.gle/KBYJrvNFh9kGHzaz7`.

Esto con el objetivo que nos informen si algún integrante de su grupo no se comunica o no se compromete con el proyecto, para que como equipo docente poder intervenir antes de la entrega y poder encontrar soluciones posibles para el correcto desarrollo de la misma.

Entrega

La entrega se realizará a través de GitHub. El repositorio debe contener una carpeta con su proyecto de Vivado. En el caso de la carpeta del proyecto, deben subir solo la carpeta `.srcs`, el archivo `.xpr` y el archivo `Basys3.xdc`

3. Contacto

Cualquier pregunta sobre el proyecto, ya sean de enunciado, contenido o sobre aspectos administrativos deben comunicarse con los ayudantes creando preguntas en el foro de canvas del curso o directamente con los ayudantes:

- Felipe Valenzuela: frvalenzuela@uc.cl
- Matías López: milopez8@uc.cl
- Cristóbal Herreros ceherreros@uc.cl
- Raúl Del Río Jara rjdelrio@uc.cl

4. Integridad académica

Los alumnos de la Escuela de Ingeniería de la Pontificia Universidad Católica de Chile deben mantener un comportamiento acorde a la Declaración de Principios de la Universidad. En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un Procedimiento Sumario. Es responsabilidad de cada alumno conocer y respetar el documento sobre Integridad Académica publicado por la Dirección de Docencia de la Escuela de Ingeniería.

Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica. Todo trabajo presentado por un alumno para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno, sin apoyo en material de terceros. Por "trabajo" se entiende en general las interrogaciones escritas, las tareas de programación u otras, los trabajos de laboratorio, los proyectos, el examen, entre otros. Si un alumno copia un trabajo, obtendrá nota final 1,1 en el curso y se solicitará a la Dirección de Docencia de la Escuela de Ingeniería que no le permita retirar el curso de la carga académica semestral. Por "copia" se entiende incluir en el trabajo presentado como propio partes hechas por otra persona.

Obviamente, está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la referencia correspondiente.

Lo anterior se entiende como complemento al Reglamento del Alumno de la Pontificia Universidad Católica de Chile. Por ello, es posible pedir a la Universidad la aplicación de sanciones adicionales especificadas en dicho reglamento.