



IIC2343 – Arquitectura de Computadores (I/2018)
Proyecto Semestral: Entrega Práctica 01
ALU de 16 bits

Fecha de entrega: Viernes 13 de Abril a las 23:59 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 el display de leds los 8 bits menos significativos de cada operando en hexadecimal (00 a FF), como se ve en la Figura 1:

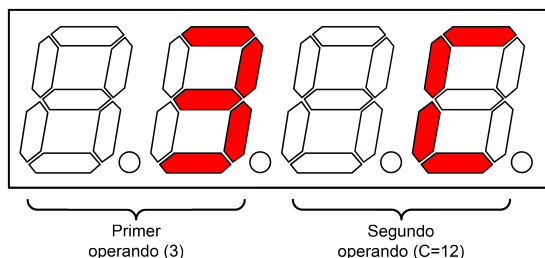


Figura 1: Distribución inicial de los operandos en el display.

Para ingresar los operandos al sistema se deben utilizar una combinación de los botones de la placa: 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, Xor y Multiplicación. 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 “clock” 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 switches que trae la placa y para ver el resultado de esta operación sin alterar los operadores se debe presionar el botón central de la misma. Cabe destacar que en el display se deben mostrar los 16 bits del resultado. Para ilustrar el uso de los botones y switches 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 switches de “operación”, en los últimos dos dígitos del display se podrá ver el número $C + 9 = 15$ (21 decimal) .

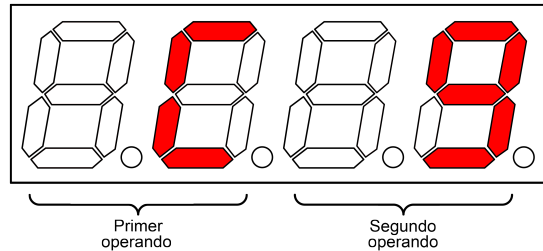


Figura 2: Distribución final de los operandos en el display.

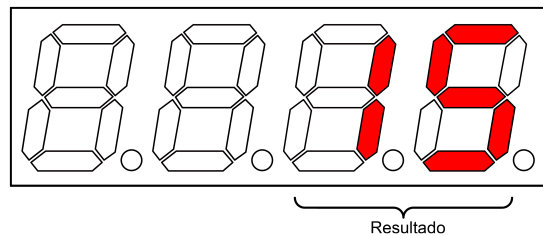


Figura 3: Distribución del resultado en el display para la operación de suma.

2. Materiales

A continuación se detalla lo que se les entregará para esta primera etapa del proyecto:

- Se les entregará los archivos base que usarán para trabajar y desarrollar el proyecto a lo largo del semestre. Qué es cada componente que ya viene en la base, cómo usar estos archivos y cómo cargar la placa se les explicará en la ayudantía del mismo día de la entrega oficial de este enunciado.

3. Desarrollo

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

1. **Operandos:** Crear las instancias de los registros y conectarlos el display y los botones para que puedan ver y modificar los operandos. Usar la señal “clock” para los registros.
2. **Resultado:** Modificar la conexión del display para que el botón central permita ver el resultado.

3. **Operacines:** 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 4 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	$a + b$	$\text{result} < a + b$	$\text{result} = 0$	0
*	*	sub	$a - b$	$a \geq b$	$\text{result} = 0$	$b > a$
*	*	and	$a \text{ and } b$	0	$\text{result} = 0$	0
*	*	or	$a \text{ or } b$	0	$\text{result} = 0$	0
*	*	xor	$a \text{ xor } b$	0	$\text{result} = 0$	0
*	*	mul	$a * b$	0	$\text{result} = 0$	0
*	*	not	not a	0	$\text{result} = 0$	0
*	*	shr	$0 \text{ \& } a(15 \text{ downto } 1)$	$a(0)$	$\text{result} = 0$	0
*	*	shl	$a(14 \text{ downto } 0) \text{ \& } 0$	$a(15)$	$\text{result} = 0$	0

Figura 4: 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. Para esta entrega no deben manipular los displays de 7 segmentos.

4. Puntajes

A continuación se describe el puntaje asociado a cada ítem:

tem	Uso Display	Uso registros	Operaciones	Leds (Z, N, C)	Formalidades	Informe
Puntaje	0.75	0.75	1.5	0.5	0.5	2.0

Figura 5: Tabla de puntajes.

5. Informe

Además de desarrollar la calculadora, deben realizar un informe que contenga lo siguiente:

5.1. Preguntas de contenido (1,5 ptos)

1. Explique el comportamiento de los componentes creados.
2. Explique cómo su proyecto suma o resta.
3. Explique cómo su proyecto multiplica
4. Interprete las señales de control pedidas.

5.2. Actividades de cada integrante (0,5)

El informe debe tener un párrafo por integrante que indique qué hizo cada integrante del grupo durante la entrega. Además, deben explicar cuáles fueron las dificultades que tuvieron durante el desarrollo de la entrega.

6. Formalidad

Finalmente, deben asegurarse de que su repositorio siga la siguiente estructura: Deben guardar en la carpeta **Proyecto Base** deben almacenar el código fuente de sus componentes y placa. Además su repositorio debe contar con una carpeta llamada **Entregas**, dentro de la cual deben estar sus entregas correspondientes a cada evaluación con el nombre de ésta. Por ejemplo, para esta primera entrega, deben guardar en su carpeta **Entregas** los archivos **.bit** e **informe.pdf**. Además de esto deben preocuparse de usar un lenguaje correcto y apropiado en todos sus archivos.

7. Entrega

Deben entregar:

- Todos los archivos involucrados en su proyecto en un repositorio.
- El informe con todo lo indicado, en formato PDF.

La entrega del proyecto (código, informe y adicionales) es por medio del repositorio en Github tal que el Viernes 13 de Abril a las 23:59 horas deben tener en la rama Master todos los archivos correspondientes a su grupo. Durante la corrección, se considerará el orden del repositorio y redacción y ortografía del informe (puntaje de formalidad).

Orden de repositorio: mantener una estructura de carpetas ordenadas que indiquen claramente los archivos de la entrega. Además, no subir archivos de compilación al repositorio (.sim, .xil)

Entregas atrasadas serán **penalizadas con 0.5 puntos** por cada hora (o fracción) de atraso.

Para entrega 2 y 3: El día anterior a la entrega, se subirán los archivos para la evaluación tanto en la página del curso como en un issue en el Syllabus del curso. Son 6 archivos que corresponden a algoritmos para evaluar. Deben generar los archivos .bit para cada uno de ellos. Luego, durante la entrega presencial, se van a probar esos algoritmos y se pedirá la compilación completa (transformar el .txt a instrucciones de la ROM, insertar estas instrucciones en el computador y generar el bitstream en Vivado) de uno de ellos. Toda acotación especial se debe incluir en el informe de cada entrega o en un README.

8. Contacto

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

- Francesca Lucchini: flucchini@uc.cl
- Felipe Pezoa: fipezoa@uc.cl
- Hernán Valdivieso: hfvaldivieso@uc.cl
- Luis Leiva: lileiva@uc.cl

9. Evaluación

Cada entrega del proyecto se evaluará de forma grupal y se ponderará por un porcentaje de coevaluación para calcular la nota de cada alumno.

Dado lo anterior, dentro de las primeras **24 horas** posteriores a cada entrega, **todos los alumnos** deberán completar de forma **individual y obligatoria** el formulario web que los ayudantes pondrán a su disposición, repartiendo un máximo de 4 puntos (aunque cambia según la cantidad de estudiantes en el grupo; el máximo corresponde al número de compañeros), con hasta un decimal, entre sus compañeros y una diferencia máxima entre el mayor y menor puntaje de 1. La suma de todos los puntos obtenidos por el integrante, sp , será utilizada para el cálculo de la nota de cada entrega, lo que puede hacer que este repruebe el curso.

La nota de cada entrega se calcula de la siguiente forma (con un máximo de 7,5):

$$NotaEntrega_{individual} = \min(k_g \times NotaEntrega_{grupal}, NotaEntrega_{grupal} + 0,5)$$

donde,

$$k_g = \frac{sp+3}{7}$$

y sp puede variar de acuerdo a la cantidad de estudiantes que conforman el grupo.

Los alumnos que no cumplan con enviar la coevaluación en el plazo asignado tendrán un **descuento de 0.5 puntos** en su nota de la entrega correspondiente.

10. 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.