

El objetivo de este trabajo practico es diseñar una calculadora básica utilizando un teclado matricial y los displays de 7 segmentos.

Las especificaciones mínimas de la calculadora son las siguientes:

- Aceptar dos operandos, números enteros de 4 dígitos, en notación decimal, que el usuario introduce usando el teclado
- Mostrar el valor que el usuario esta ingresando, a medida que presiona el teclado, en un display de 7 segmentos de al menos 4 dígitos
- Limpiar el display cuando espera el segundo operando
- Mostrar el resultado correcto una vez que se presione la tecla “=” luego del segundo operando
- Poder realizar sumas y restas no signadas. No se necesita considerar condiciones de overflow o underflow

Algunas especificaciones relacionadas a la implementación:

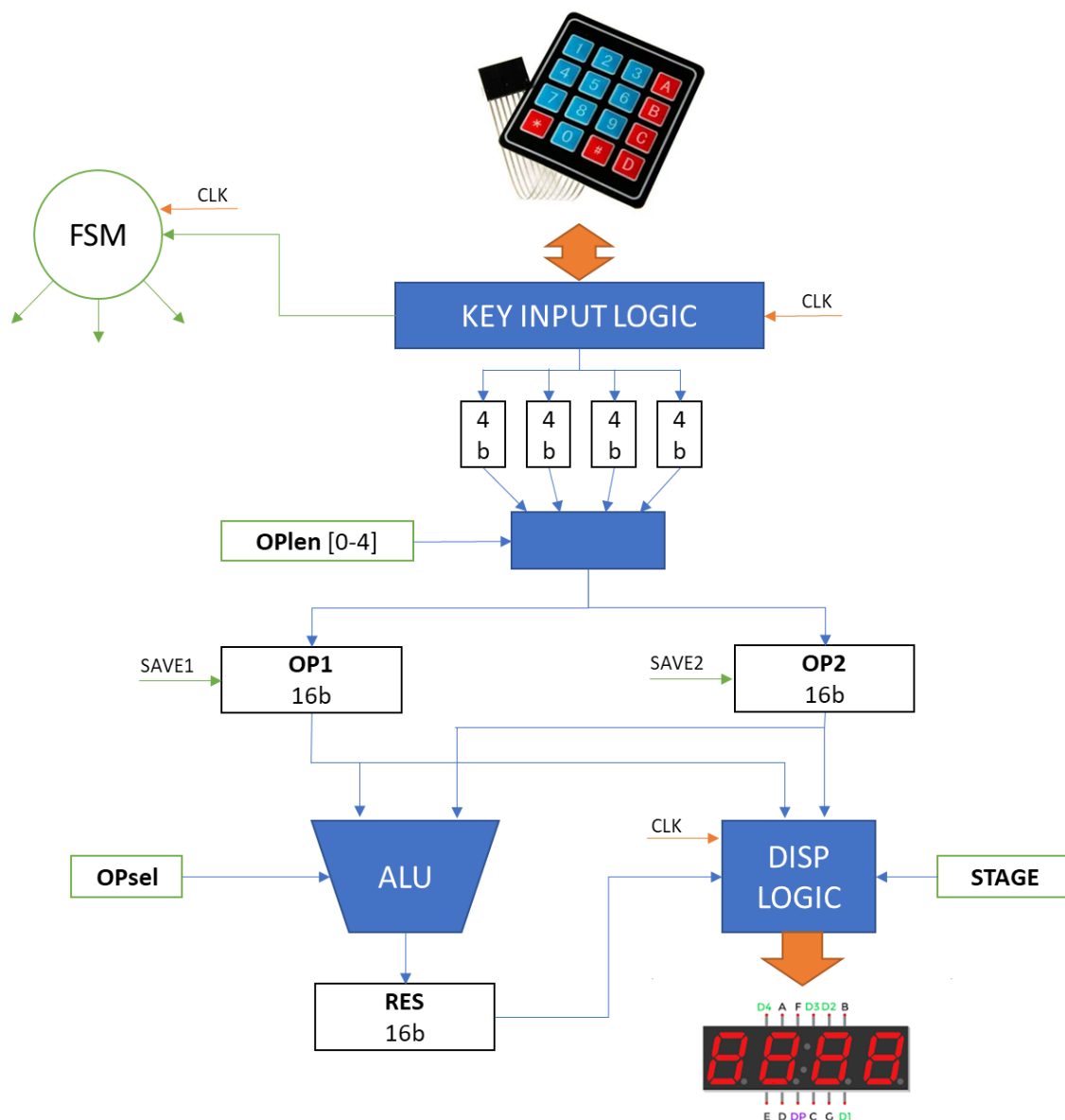
- Utilizar la FPGA UPduino para la mayor parte de la implementación, con entrada y salida serie para teclado y display (sujeto a la implementación de las placas del TP1)
 - ATENCION: la FPGA soporta I/O de 3.3V únicamente!
 - Respete el protocolo serie definido en el TP1
- Utilizar el circuito del display del TP2
 - Los dígitos del display deben estar multiplexados; la implementación depende del diseño de la placa del TP2
- La calculadora se recomienda implementar como una máquina de estados para su comportamiento general

Diseño de alto nivel

Se recomienda utilizar un diseño de alto nivel, similar al siguiente, para implementar la calculadora.

La mayor parte de los bloques se puede realizar en Verilog o Python HDL, para luego conectarlos de manera correcta en el módulo de “top level”.

Nota de implementación: Una opción es que la lógica sea toda BCD para simplificar el diseño. En este caso, prestar atención al implementar las operaciones aritméticas.



FSM

La FSM del sistema controla la etapa en la que se encuentra la calculadora (esperando operando 1, esperando operando 2, mostrando resultado).

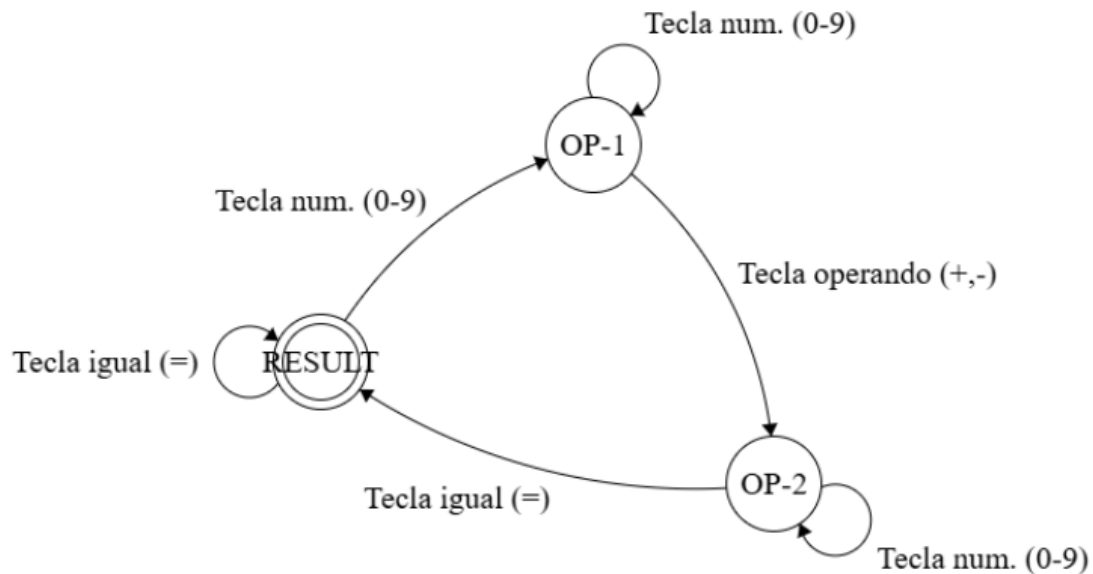
Una vez mostrando resultado, la calculadora debería estar esperando un nuevo operando para volver a comenzar.

- Las entradas de la FSM es la tecla que proviene del teclado
- Las salidas de la FSM controlan:
 - Cantidad de dígitos del operando 1
 - Cantidad de dígitos del operando 2
 - Señal de SAVE1/SAVE2 para almacenar el operando actual en su registro
 - Guardado de la operación (suma, resta, etc.)

- Selección de la salida del display (operando 1, operando 2, resultado)

La FSM se debe alimentar con un clock lo suficientemente rápido para poder detectar y responder a un usuario relativamente ágil con las teclas.

Un ejemplo posible de la FSM seria cm



KEY INPUT y decodificación de operando

La lógica de entrada debe, en base al valor recibido dl teclado, almacenar el valor seleccionado por el usuario en uno de los 4 registros de entrada (operandos de máx. 4 dígitos BCD).

También debe enviar a la FSM señales cuando:

- El usuario presiona un valor numérico
- El usuario presiona la operación suma
- El usuario presiona la operación resta
- El usuario presiona el “=”

En base a la cantidad de teclas numéricas presionados por el usuario, el bloque de decodificación toma los [0-4] valores ingresados por el usuario y lo transforman en un valor binario, o BCD de 16 bits.

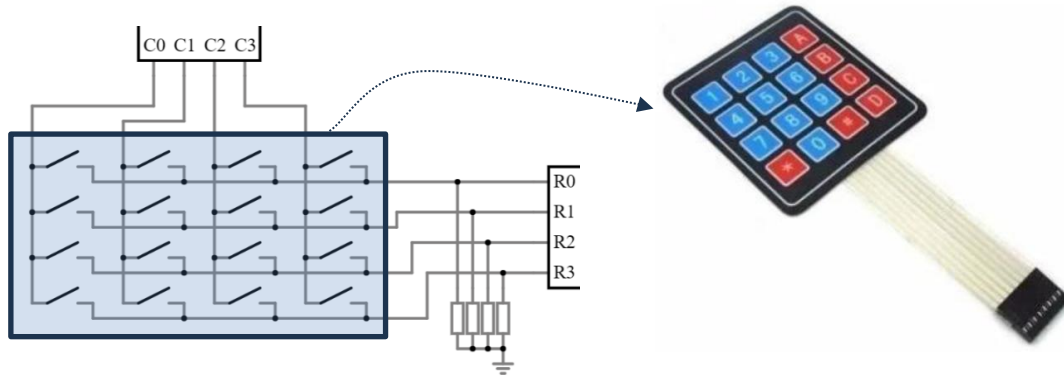
El estado de la máquina de estado decide donde se envía el operando decodificado, OP1 u OP2.

Se espera que el usuario introduzca números de una manera natural; es decir, que los dígitos ya ingresados se desplacen hacia la izquierda con cada nuevo valor:

Entrada	Display (opc. 1)	Display (opc. 2)
<div>1</div>	1	1
<div>15</div>	15	15
<div>153</div>	153	153

Teclado matricial

El teclado matricial (bloque azul) está disponible como en la foto.



Se debe generar pulsos secuenciales en C0-C3 y leer el valor en R0-R3 para encontrar el valor final del botón presionado. Diseñe bloques en la FPGA para generar las señales secuenciales en las columnas (usando salidas), y leer el valor de las filas (usando entradas).

Para simplificar, puede considerar que el valor a enviar (KEY[3..0]) se puede formar directamente combinando los dos bits del valor de columna seleccionado (C[1..0]) y los dos bits de fila (R[1..0]), aunque el valor binario no coincida con la etiqueta del botón.

Utilice registros dentro de la FPGA para retener el valor presionado por el usuario. También se pueden detectar de manera opcional condiciones de error, por ejemplo, dos botones presionados al mismo tiempo.

Seleccione una frecuencia de barrido del teclado apropiada, de tal manera que el uso del mismo sea natural.

Notas de implementación: Recuerden que se necesita incluir pull-downs (o pull-ups, según corresponda) en las entradas de la FPGA si los switches son de un solo polo, como los del teclado.

Se debe implementar un anti-rebote en la FPGA (o externamente con lógica discreta) de manera que cada presión de una tecla sea interpretada correctamente como un solo dígito.

De manera opcional, puede usar componentes de lógica discreta en la placa (shift register, encoder/decoder. etc.) para reducir la cantidad de pines necesarios de la FPGA.

Se recomienda que el módulo de teclado implementado en la FPGA tenga una salida que indique cuando el usuario presionó y soltó la tecla, para poder diferenciar entre presionar de manera continua, y presionado múltiple de la misma tecla.

ALU

La Arithmethical Logical Unit debe proveer al menos las dos operaciones básicas:

- Suma
- Resta

Las operaciones pueden hacerse en BCD o en Binario, según el diseño de la etapa de entrada.

El bloque de la ALU debe ser, idealmente, puramente combinacional, dependiendo la salida de las tres entradas: Operando 1, Operando 2 y Operación.

Lógica de DISPLAY

Este módulo corresponde al del display del TP2, que se comunica con la FPGA mediante el canal serie, usando pines de GPIO como Salida para generar las señales de CLK_DISPLAY, CLK_DATA, DATA y DATA_READY (o las necesarias para su diseño).

La información a mostrar en el display debe enviarse de manera serial al display. Considerar el formato para que los valores se vean correctamente en los displays.

Entrega

Se espera que cada grupo tenga, al finalizar el trabajo práctico:

- Código (Verilog/HDL) implementado de manera modular con el comportamiento de la calculadora
- Simulaciones (Verilog/Python) para funcionalidad básica, por ejemplo:
 - Entrada de un operando
 - Estados de la FSM de control
 - Etc.
- Placa (PCB), archivos de diseño, y fabricación de la misma con al menos:
 - Interfaz hacia la placa de teclado
 - Interfaz hacia el display del TP2
 - Zócalo para conectar la FPGA UPduino (no soldar a la placa!)
 - Circuito de alimentación para la lógica externa y la FPGA, incluyendo capacitores de bypass y desacople según considere necesarios
- Informe del diseño donde se encuentren las consideraciones y detalles del diseño implementado
- Prototipo funcional utilizando la FPGA provista en la placa anterior

Se considerará para la calificación:

- Criterios de diseño de lógica digital, máquinas de estado
- Modularidad del código Verilog/HDL
 - Se debe minimizar el uso de código estilo secuencial, IF-THEN
- Selección de Clock(s) en la FPGA, teclado, display
- Diseño del PCB
- Consideraciones eléctricas
 - Tensiones de alimentación
 - Corriente y frecuencia del display multiplexado
 - Capacitores de desacople y bypass

Extras

Se pueden considerar las siguientes mejoras, completamente opcionales:

- Encadenar operaciones (utilizar el resultado como próximo operando)
- Mayor cantidad de funciones (multiplicación, división, etc.)
- Operandos más grandes (>4 dígitos)
- Manejo de errores: overflow, underflow, falta de operando, etc.
- Soporte para números con punto decimal
- Buzzer de teclado
- Etc.