

Resumen

El objetivo de este trabajo practico es diseñar una calculadora básica utilizando el teclado matricial y 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:

- Se debe utilizar la FPGA UPduino para la mayor parte, pero no toda la lógica
 - ATENCION: la FPGA soporta I/O de 3.3V únicamente!
- Se debe utilizar el circuito del teclado del TP1
- Se debe multiplexar al menos los 4 dígitos del display para reducir la cantidad de conexiones de salida
- La calculadora debe implementar una maquina de estados para su comportamiento general

El trabajo esta dividido en varias partes para ayudar con el diseño de la calculadora, aunque las mismas no son prescriptivas: Si desea implementar la calculadora de otra manera, dentro de los parámetros especificados anteriormente, es posible con previa discusión con la cátedra.

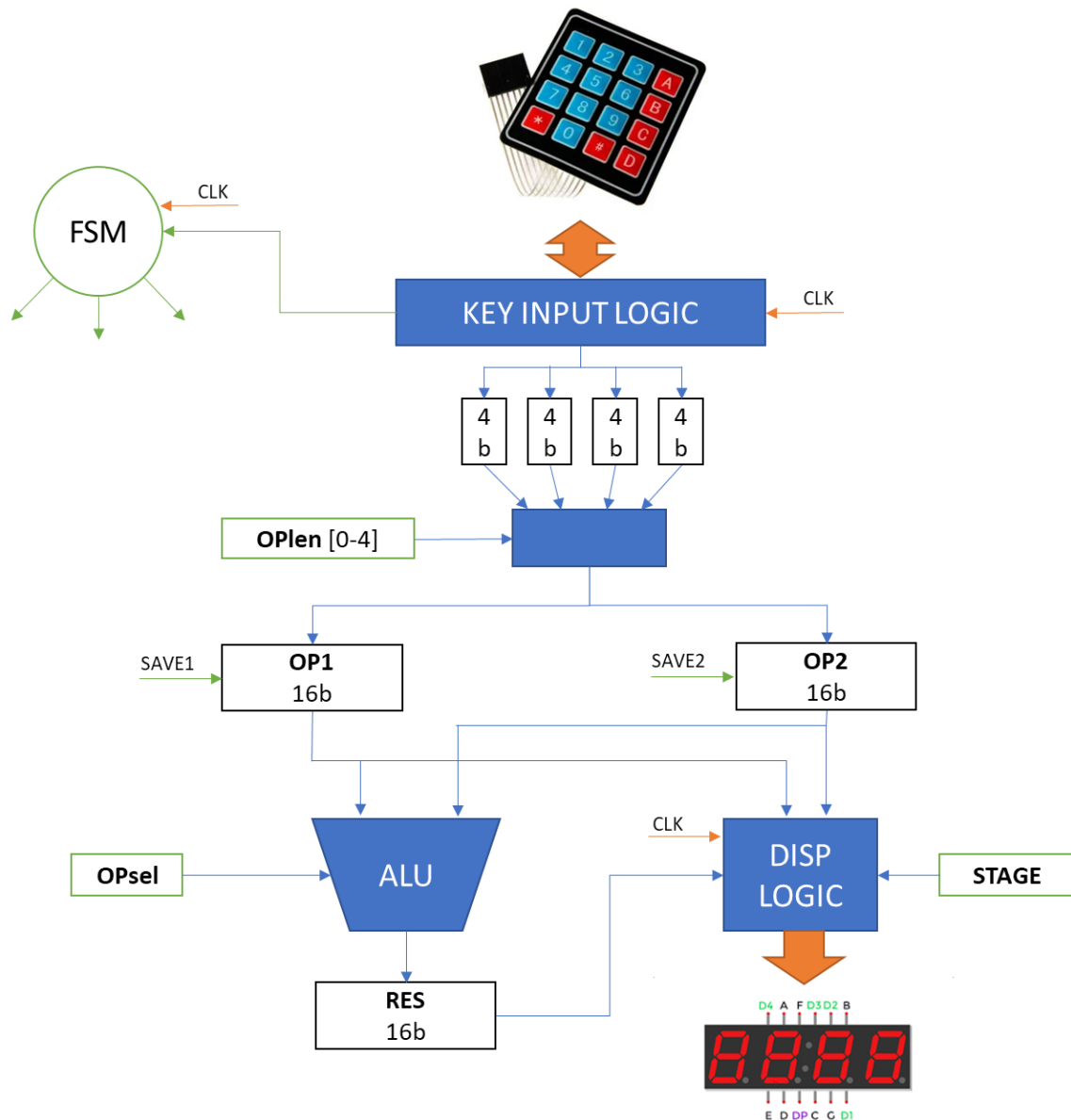
No se permite realizar una implementación tipo "código" de una calculadora.

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, para luego conectarlos de manera correcta en el módulo de "top level".

Se recomienda que toda la lógica sea BCD para simplificar el diseño. Cuidado al implementar las operaciones aritméticas!



FSM

La FSM del sistema debe controlar 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 son las teclas que el usuario presiona
- 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
- Selección 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.

El barrido del teclado matricial puede hacerse en la misma FSM, una FSM separada, o en la lógica del bloque **KEY INPUT**. Considerar si se necesita otro clock.

KEY INPUT y decodificación de operando

Este bloque debe controlar el teclado, y en base a las entradas recibidas, debe 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 BCD de 16 bits.

El estado de la maquina 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:

1. Tecla [2] -> 0002
2. Tecla [5] -> 0025
3. Tecla [0] -> 0250

ALU

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

- Suma BCD
- Resta BCD

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 modulo debe tomar valores de 16 bits BCD internos, y generar las señales necesarias para mostrarlos en cuatro displays de 7 segmentos. Se elije cual de los registros mostrar en base a la etapa en la que se encuentra (STAGE). El display debe actualizarse al cambiar el valor del registro.

Las señales de salida de la FPGA deben ser la menor cantidad posible:

- 7 salida, una por segmento de un dígito
- 4 salidas, una por dígito a mostrar

Los 4 dígitos deben ser multiplexados, a una frecuencia suficiente para que el “parpadeo” apreciable del display sea mínimo.

ATENCION: Se debe diseñar e implementar componentes externos para manejar el display, ya que la FPGA no soporta la corriente necesaria.

Entrega

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

- Código de Verilog implementado de manera modular con el comportamiento de la calculadora
- Simulaciones de Verilog 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 de TP1 (se permite rehacer la interfaz de teclado directamente en el nuevo PCB si se desea)
 - Lógica “de potencia” para el display
 - 4 displays de 7 segmentos
 - 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
 - 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
- Control de brillo del display variable
 - Extra por control digital con el teclado
- Canal de conexión serie (1/2/3 cables de señal) con el display y/o teclado
- Etc.