

FASE 5

UNIDAD DE CONTROL Y CONSTRUCCIÓN DE LA CALCULADORA

OBJETIVOS DEL APRENDIZAJE:

Tras completar esta práctica debemos ser capaces de:

- Comprender el funcionamiento de los circuitos secuenciales.
- Diseñar circuitos secuenciales basándose en las especificaciones del problema.

REFERENCIAS:

- Bibliografía básica y complementaria.
- Diapositivas del Tema 4 "Circuitos secuenciales".

ELEMENTOS NECESARIOS:

- Circuito de apoyo "SubproyectoControl.circ" (suministrado)
- Circuito SubproyectoMemoria.circ resuelto (NO suministrado, es el desarrollado en la fase 4 que debe estar en el mismo directorio que los ficheros .circ suministrados en la práctica). El circuito resuelto NO es necesario para realizar la práctica con éxito, pero sí para que la calculadora tenga una funcionalidad completa.
- Biestables JK, D y T
- Registros de desplazamiento
- Codificadores y Multiplexores

INTRODUCCIÓN: UNIDAD DE CONTROL PARA SUMA Y RESTA con acumulación

IMPORTANTE: es necesario tener el fichero **SubproyectoMemoria.circ** desarrollado en la fase 4 en el mismo directorio que los ficheros **.circ** suministrados en la práctica.

OBJETIVO:

En esta parte se pretende construir una funcionalidad COMPLETA para las operaciones SUMA y RESTA de nuestra calculadora. De esta forma, nuestro circuito se comportará a nivel de teclado e información mostrada en pantalla tal y como lo haría una calculadora sencilla para las operaciones de suma y resta (no se introducen otras operaciones por no complicar el diseño de la Unidad de Control).

ELEMENTOS DEL SISTEMA

El circuito suministrado en la práctica contiene los elementos digitales desarrollados en fases anteriores en concreto se suministran los siguientes módulos:

MODULOS COMPLETOS (no hay que modificarlos)

1. **ModBus:** Contiene la circuitería de la fase 4 y otros elementos necesarios para conectar la Unidad de Control.
2. **AluMod:** Contiene la Unidad Aritmética Entera realizada en la fase 3, a la que se le han añadido dos registros de 11 bits (A y B) para almacenar los operandos en las operaciones de suma ($A + B$) y resta ($A - B$).
3. **ModCalc:** Integra los elementos anteriores y la Unidad de Control en un solo módulo.
4. **Completa:** Elemento principal del circuito que representa la calculadora, incorporando los botones, el módulo ModCalc y los visualizadores.

MODULO INCOMPLETO (hay que realizarlo)

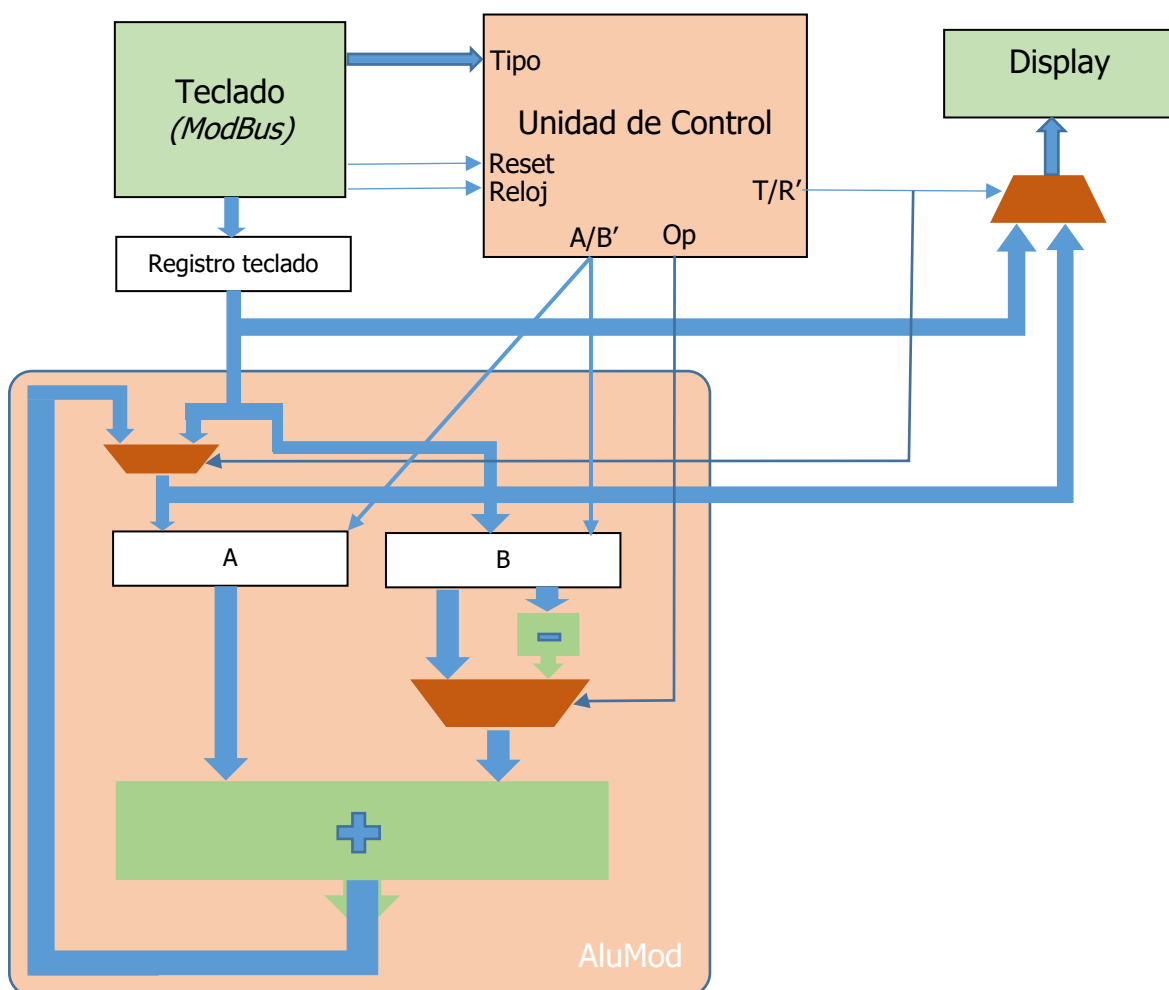
- **UnidadControl:** Es el módulo que realiza la coordinación del resto de módulos gracias a la implementación de un autómata y siguiendo las especificaciones que se dan a continuación.

REQUERIMIENTOS (funcionamiento de la calculadora):

- El usuario introduce un primer número por teclado (puede sobrepasar la capacidad del display tal y como se hizo en la parte segunda de esta práctica mostrando los 3 últimos dígitos). Mientras introduce el número, este se muestra en la pantalla.
- A continuación, el usuario pulsa bien el botón “+” o bien el botón “-” en función de si lo que quiere hacer es una suma o resta respectivamente.
- Ahora el usuario introduce un segundo número que también se muestra en pantalla conforme se escribe.
- Finalmente, el usuario pulsa el botón “=” y entonces el visor muestra el resultado de la suma o la resta. Si el usuario pulsara de nuevo el botón “=”, se produciría una suma incremental (o resta decremental dependiendo de la operación) del valor tecleado en último lugar sobre el resultado actual y se mostraría en pantalla.

Para el resto de las combinaciones no especificadas se debe intentar seguir el comportamiento normal de una calculadora.

A continuación, se muestra el esquema básico de la Calculadora:



IMPLEMENTACIÓN DE LA UNIDAD DE CONTROL

Para implementar correctamente el funcionamiento antes descrito, necesitamos diseñar UN AUTÓMATA que genere las señales necesarias a nuestro sistema. Este autómata debe ser construido en el Subcircuito "UnidadControl" del fichero proporcionado. Las entradas y salidas (que ya están definidas) son las siguientes:

ENTRADAS

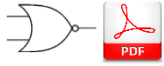
- **Reloj:** Es la señal que propiciará cada cambio en el autómata y se relaciona con la pulsación de cualquier tecla numérica o las teclas: "+", "-" o "=" proporcionada desde el módulo *ModBus*.
- **Tipo:** Esta entrada de dos bits, proporcionada desde el módulo *ModBus*, indica el tipo de tecla pulsada:
 - 00 = tecla numérica
 - 01 = tecla "+"
 - 10 = tecla "-"
 - 11 = tecla "="
- **Reset:** Esta entrada no está incluida como entrada del diseño secuencial (no formará parte de nuestro autómata) pero la usaremos para reiniciar los biestables.

SALIDAS

- **T/R'** = bit que indica qué dato debe mostrar el visualizador:
 - 0: muestra el resultado del sumador
 - 1: muestra el valor almacenado en el registro asociado al teclado
- **A/B'** = bit que indica qué dato se está introduciendo
 - 0: el segundo dato (B)
 - 1: el primer dato (A)
- **Op** = indica qué operación se está realizando
 - 0: suma
 - 1: resta

Ejercicio 1 (80% de la fase)

Con las entradas y salidas especificadas para la Unidad de Control debes construir un autómata de MEALY que gestione la entrada del teclado y genere las salidas adecuadas según el apartado de REQUERIMIENTOS. Para ello debes:



1. Dibujar el diagrama de estados
2. Obtener la tabla de estados sin codificar
3. Obtener la tabla de estados codificada
4. Obtener la tabla de excitación de los biestables para el tipo JK
5. Obtener las ecuaciones de las entradas de los biestables y las salidas
6. Construir el circuito dentro del módulo *UnidadControl* usando el simulador y probarlo usando la calculadora completa.

Ejercicio 2 (20% de la fase)

Responde **razonadamente** a las siguientes preguntas acerca de los módulos suministrados en la práctica:



- a) En el módulo *AluMod*, ¿qué función realizan las dos puertas básicas OR y AND y cómo la hacen?
- b) En el módulo *AluMod*, ¿qué función realiza el MUX que está situado entre los registros A y B?
- c) En el módulo *ModBus*, ¿qué función realiza el codificador de 4 entradas? ¿por qué crees que se ha situado un registro D a su salida?
- d) En el módulo *ModBus*, ¿qué función realiza el MUX que está situado junto a la entrada TR?