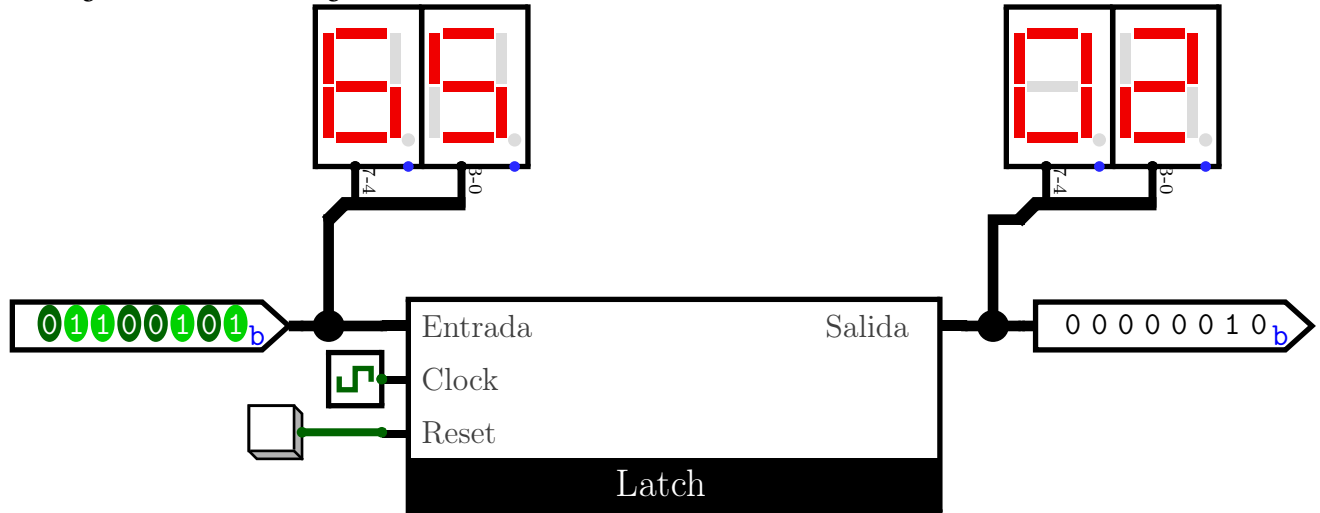


Versión 261C.01

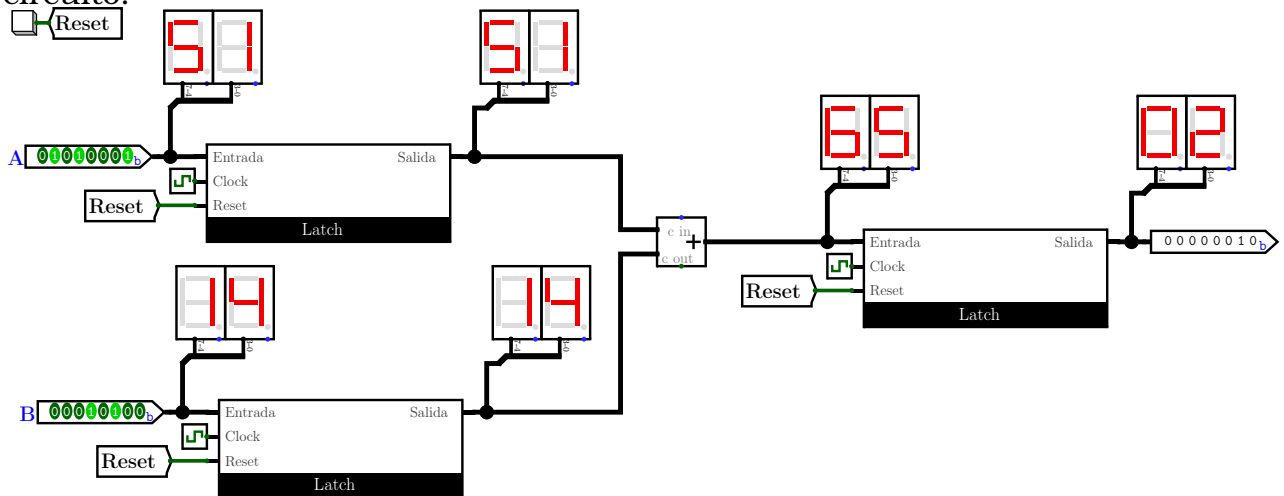
Carrera: INGENIERÍA EN INFORMÁTICA
Asignatura: 3631 - Fundamentos de sistemas embebidos
Tema: Circuitos con memoria
Unidad: 4
Objetivo: Comprender el diseño de circuitos capaces de memorizar
Competencias a desarrollar: <ul style="list-style-type: none">• Concepción, diseño y desarrollo de proyectos de ingeniería en informática.• Gestión, planificación, ejecución y control de proyectos de ingeniería en informática.• Utilización de técnicas y herramientas de aplicación en la ingeniería en informática.• Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.• Desarrollo de una actitud profesional emprendedora.• Aprendizaje continuo• Actuación profesional ética y responsable.• Comunicación efectiva.• Desempeño en equipos de trabajo.• Identificación, formulación y resolución de problemas de ingeniería en informática
Descripción de la actividad: <ol style="list-style-type: none">1. Tiempo estimado de resolución: 1 semana2. Metodología: Ejercicios verificados en simuladores3. Forma de entrega: No obligatoria4. Metodología de corrección y feedback al alumno: Presencial y por Miel.

E- Registros y contadores

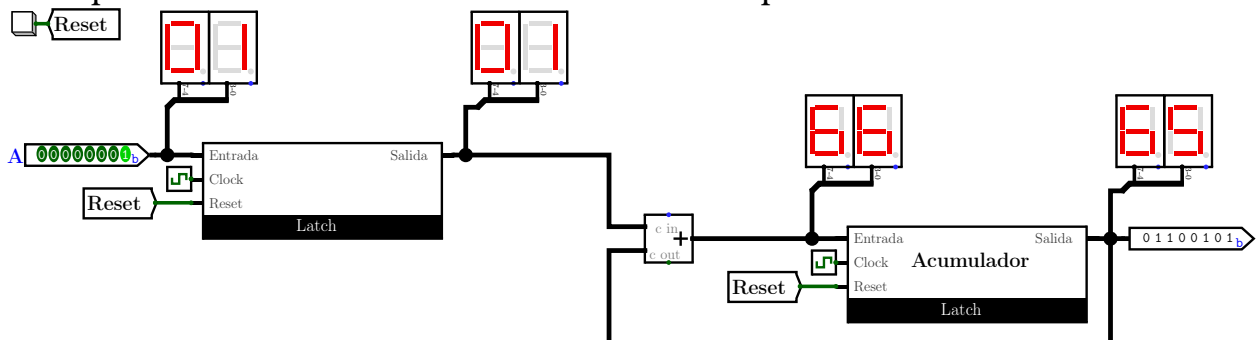
- E.1** Utilizando logisim-evolution implemente un registro Latch de 8 bits (entrada paralelo / salida paralelo). Debe poseer una entrada de Reset que fuerza el valor del Latch a cero de forma asincrónica (es decir, no depende del clock).
Nota: como todo circuito secuencial, va a poseer una entrada de Clock, en este caso funcionando en flanco ascendente



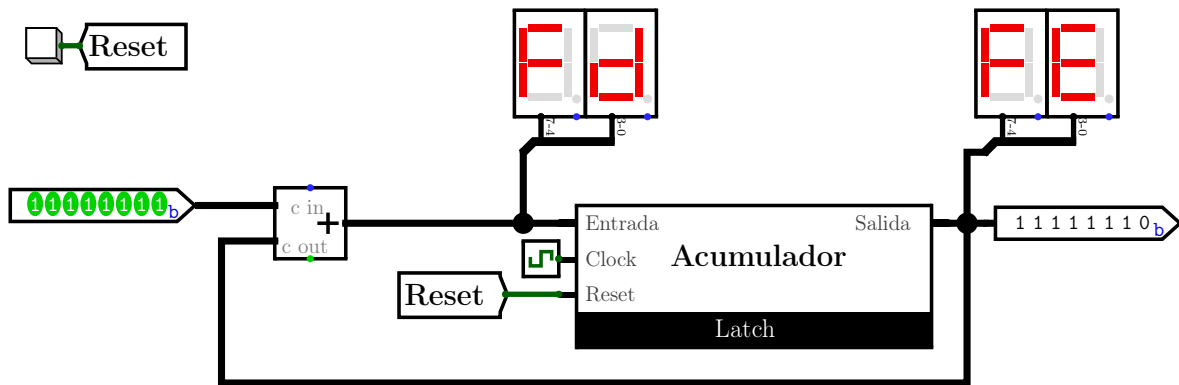
- E.2** Utilizando 3 registros Latch y un sumador de 8 bits, implemente el siguiente circuito.



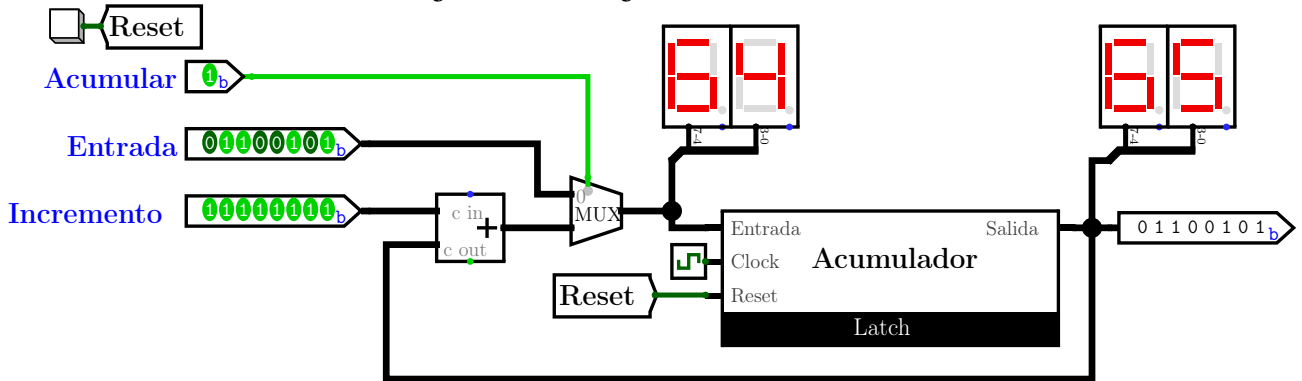
- E.3** Implemente un registro sumador/acumulador, es decir un registro realimentado que acumula un valor de entrada con cada pulso de clock.



- E.4** Implemente un acumulador cuyo valor incremental sea fijo. *Nota: tenga en cuenta que al trabajar los números negativos en complemento a la base, puede sumar -1 para hacer decrementos*



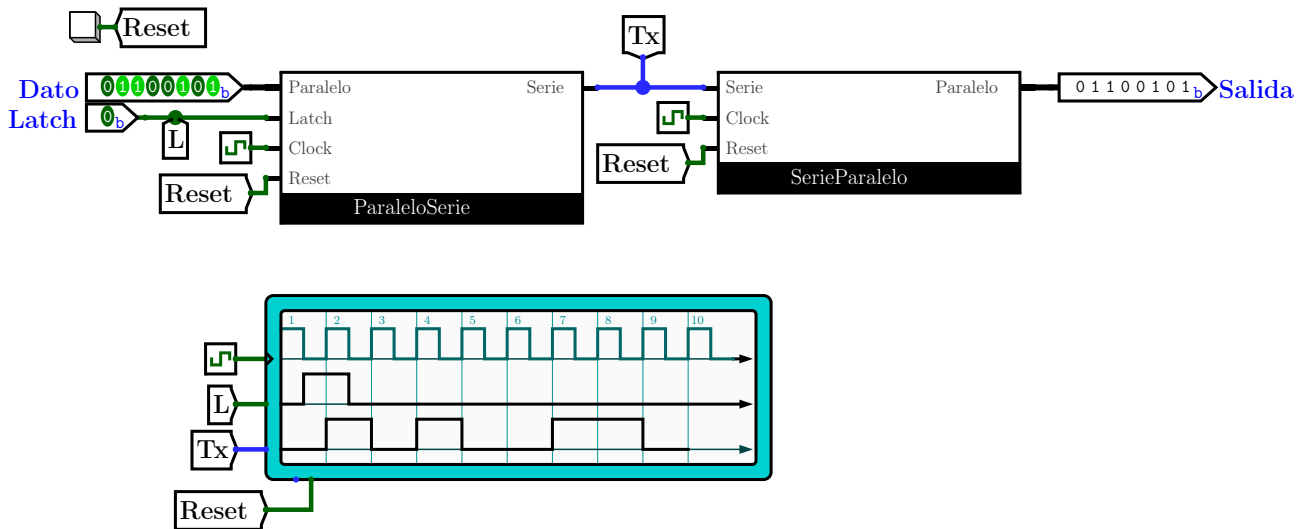
E.5 Modifique el circuito anterior para tener una entrada que permita forzar un valor en el acumulador. Agregue una nueva entrada de selección llamada Acumular que permita seleccionar en el acumulador el valor de entrada o el valor realimentado con el incremento. *Nota: Este circuito permite cargar un valor inicial haciendo que acumular sea cero y generando un pulso de clock. Luego cuando acumular sea uno el circuito funciona de forma realimentada*



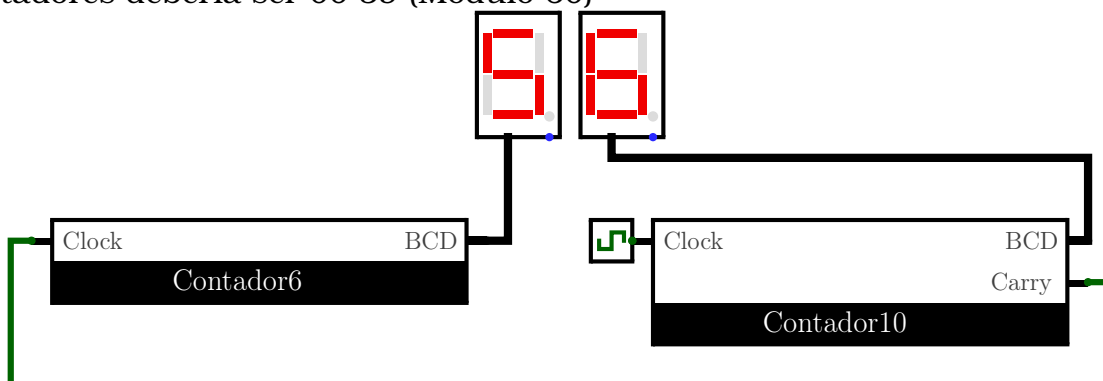
E.6 Modifique el circuito anterior agregando un MUX que seleccione el valor de incremento entre cero y otro valor. La señal de selección de este MUX se genera detectando si el valor acumulado es cero, en cuyo caso el valor de incremento es cero. De esta forma el acumulador mantiene su valor cuando llega a cero, y solo puede volver a acumular cuando ingresa por la entrada un valor distinto de cero.

E.7 Teniendo en cuenta los últimos circuitos diseñe un multiplicador por sumas sucesivas. Ej: Si se desea multiplicar 3x4 se realiza 3+3+3+3. Va a utilizar un acumulador para la suma sucesivas y otro que decrementa hasta cero y controla los valores de realimentación. *Nota: la idea del ejercicio es utilizar lo visto de secuenciales y combinacionales para resolver este problema más complejo. Haga las modificaciones que crea necesarias para resolver el problema. Recomendamos seguir con otros ejercicios y volver a este luego de completar la guía.*

E.8 Implemente un registro ParaleloSerie. Este registro posee una entrada de datos de 8 bits y una señal de Latch de un bit. Cuando Latch=1 entonces el valor de paralelo se almacena en el registro con el siguiente flanco de clock. Si Latch=0 entonces cada flanco de clock desplaza a derecha el valor almacenado en el registro que sale por la salida Serie. Luego implemente un circuito SerieParalelo.



- E.9** Diseñe e implemente un contador binario asincrónico módulo 16 de cuenta descendente. Grafique en logisim-evolution los valores utilizando el osciloscopio digital.
- E.10** Diseñe e implemente un contador binario síncrono módulo 16 de cuenta ascendente. Grafique en logisim-evolution los valores utilizando el osciloscopio digital.
- E.11** Diseñe e implemente un contador código Johnson módulo 10. Grafique en logisim-evolution los valores utilizando el osciloscopio digital.
- E.12** Diseñe e implemente un contador one-hot módulo 5. Grafique en logisim-evolution los valores utilizando el osciloscopio digital.
- E.13** Diseñe un contador módulo 10 (Contador10) con una salida de 4 bits (llamada BCD) y una salida llamada Carry de forma tal que Carry=1 cuando Contador10 pasa de 1001 a 0000. Luego Diseñe un contador módulo 6 (Contador6) y conecte el circuito de la siguiente forma. El módulo de cuenta de ambos contadores debería ser 00 59 (Módulo 60)



- E.14** Diseñe un reloj que pueda contar desde 00:00:00 hasta 23:59:59. Simule usando auto-tick en logisim-evolution. Tenga en cuenta que los contadores que cuenten de 00 a 59 son todos iguales, pero el contador que cuenta de 00 a 23 tiene que detectar el caso en donde ambos contadores llegan a 23 y volver a 00. Es decir, el dígito más significativo cuenta 0,1 y 2, pero el menos significativo cuenta de 0 a 9 cuando el más significativo es distinto de 2 y cuenta de 0 a 3 cuando el otro es igual a dos... eso quiere decir que va a tener un "reset" combinado.