

Práctica 2 - Lógica Digital - Parte B

Organización del Computador 1

Segundo Cuatrimestre 2021

Todas las compuertas mencionadas en esta práctica son de 1 ó 2 entradas, a menos que se indique lo contrario. Usaremos los símbolos detallados a continuación para representar las distintas funciones lógicas: XOR $\rightarrow \oplus$, NAND $\rightarrow \downarrow$, NOR $\rightarrow \downarrow$.

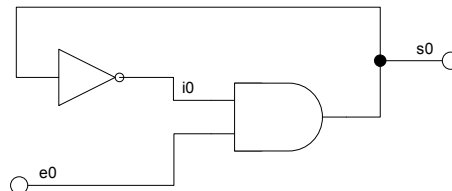
Durante la presente práctica se recomienda fuertemente la utilización de un simulador para experimentar con los componentes y circuitos propuestos y verificar las soluciones. Una recomendación es el Logisim (<http://www.cburch.com/logisim/>).

Circuitos Secuenciales

Ejercicio 14

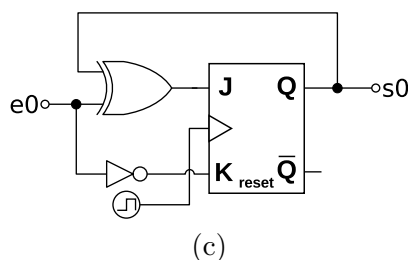
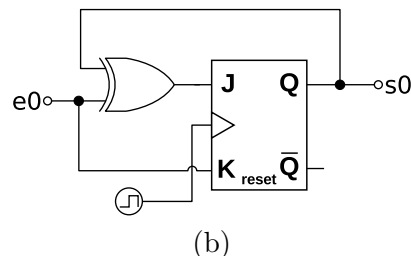
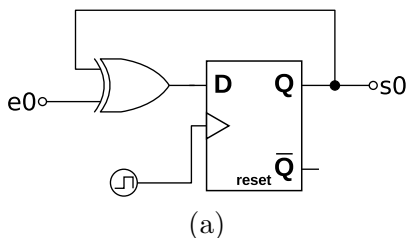
a) Escribir el diagrama temporal para el siguiente circuito secuencial desde 0 ns hasta 65 ns, suponiendo

- un retardo de 15 ns para la compuerta AND,
- un retardo de 5 ns para la compuerta NOT,
- en el tiempo 0 ns la señal e_0 cambia a 1, inicialmente en 0.
- las señales i_0 y s_0 tienen valor 1 y 0 respectivamente en el tiempo 0 ns.
- suponer que los componentes empiezan a estabilizarse cuando sus señales de entrada están estables.

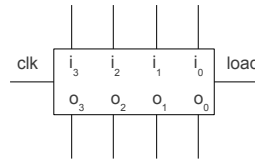


b) ¿Podría alcanzar s_0 un valor estable en el punto anterior? ¿Y en el caso en que e_0 fuera 0 en lugar de 1, se estabilizaría?

Ejercicio 15 Escriba tablas características que especifiquen el comportamiento de cada uno de los siguientes circuitos secuenciales:

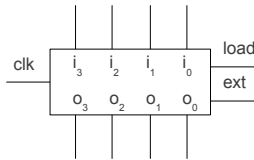


Ejercicio 16 *Registro simple.* Diseñar un registro *simple* de cuatro *bits*. Este tipo de registros es un circuito de seis entradas (i_0 a i_3 , *load*, *clk*) y cuatro salidas (o_0 a o_3), cuyo funcionamiento es el siguiente: cuando la señal *clk* alcanza su flanco ascendente, si *load* está alta, almacena las señales recibidas en i_0 a i_3 , si no, no cambia su contenido. Por las líneas de salida, se emite el valor almacenado en el registro.

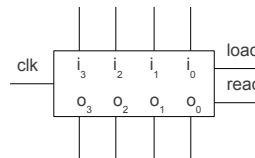


Ejercicio 17 Extensor de signo

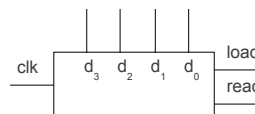
Diseñar un registro *extensor de signo* de dos a cuatro bits. Del mismo modo que un registro simple (ver ejercicio 16) este circuito toma el valor de sus cuatro entradas i_0 a i_3 cuando su señal *clk* atraviesa su flanco ascendente si la entrada *load* vale 1. Por sus líneas de salida (o_0 a o_3) se emite el valor almacenado si la señal *ext* está baja; por el contrario, si vale 1, se emite una representación de cuatro *bits* del número almacenado en los dos *bits* menos significativos del registro, interpretados como un entero codificado en complemento a 2.



Ejercicio 18 *Registro de salida restringida.* Diseñar un registro *de salida restringida* de cuatro bits. Este tipo de registros es un circuito de siete entradas (i_0 a i_3 , *load*, *clk* y *read*) y cuatro salidas (o_0 a o_3), muy similar al registro simple (ejercicio 16) pero que sólo emite su salida por las líneas o_0 a o_3 si *read* está alta cuando *clk* alcanza su flanco ascendente. Dicha salida se debe mantener hasta el próximo flanco ascendente en *clk* donde *read* esté baja.



Ejercicio 19 *Registro bidireccional.* Diseñar un registro *bidireccional* de cuatro bits. Este tipo de registros es un circuito con tres entradas (*load*, *read*, *clk*) y cuatro señales de entrada y salida (d_0 a d_3). Su funcionamiento es el siguiente: si la señal *load* vale 1 cuando *clk* alcanza su flanco ascendente, almacena los valores recibidos en d_0 a d_3 ; en cambio, si *read* está alta, se emite el valor almacenado en el registro por esas mismas líneas¹. Las señales *read* y *load* nunca valen 1 simultáneamente.

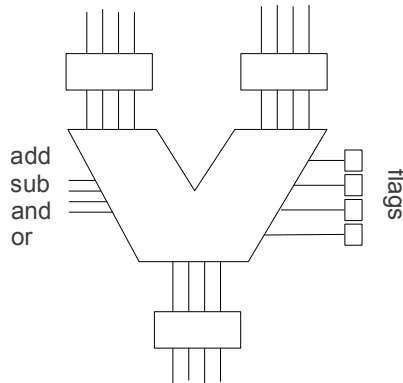


¹ Ayuda: utilice componentes de tres estados.

Ejercicio 20 ALU

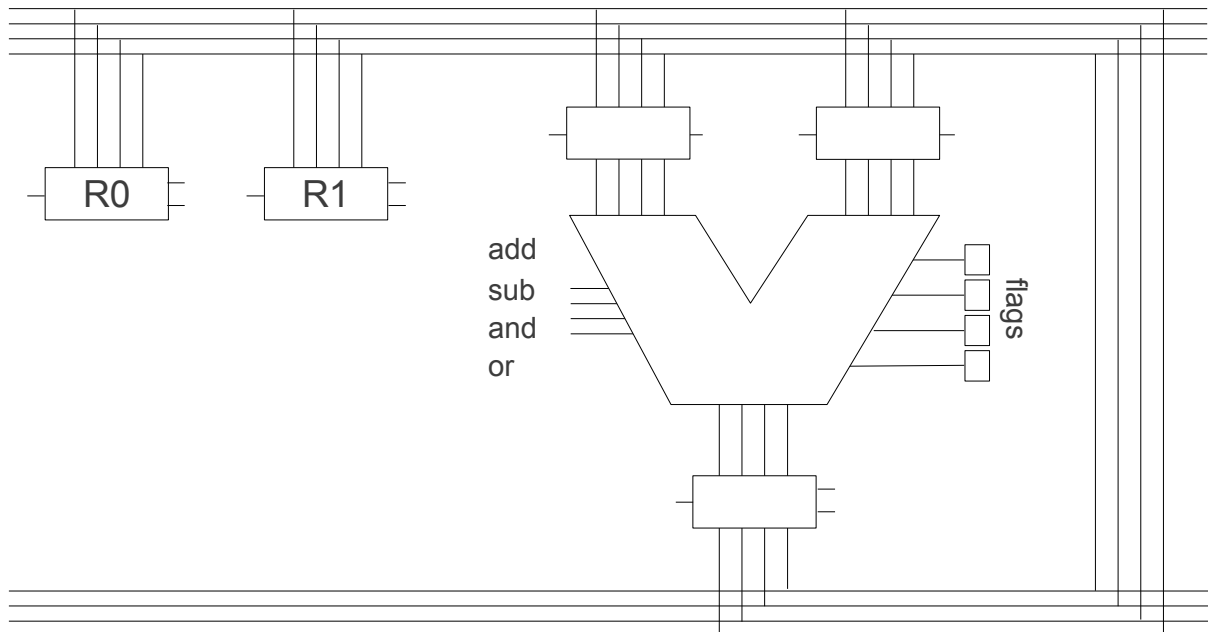
Diseñe una ALU con las siguientes características:

- cuatro señales de entrada que indican la operación a realizar: **add**, **sub**, **and**, **or**;
- dos registros simples, donde se almacenan los operandos a utilizar;
- un registro de salida restringida, donde se almacena el resultado;
- cuatro *flags* cuyos valores son determinados por la última operación realizada.



Ejercicio 21

Dado el siguiente circuito, indique mediante un diagrama de tiempos la secuencia de activaciones y desactivaciones de señales de control necesarias para que el valor almacenado en el registro bidireccional (ejercicio 19) **R0** se sume al valor del registro bidireccional **R1** y el resultado se almacene en el registro **R0**.



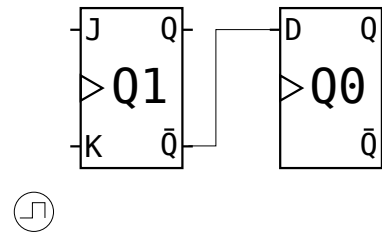
Ejercicios tipo parcial

Ejercicio 22 Para el presente ejercicio puede utilizar los siguientes componentes: SUMADOR COMPLETO de 1 *bit*, compuertas lógicas de 2 entradas y negadores de 1 *bit*. Está prohibido utilizar componentes de 3 estados.

- a) Armar un componente que tenga como entrada 1 número binario (A) en complemento a 2 de 3 *bits* y que calcule su inverso aditivo ($-A$), en caso de que exista.

- b) Armar un componente que tenga como entrada 2 números binarios (A y B) en complemento a 2 de 3 *bits* y que calcule $A - B$. Además se pide que el componente tenga 4 salidas adicionales correspondientes a los flags Z, C, V y N con su interpretación habitual.
- c) Dada la siguiente tabla que se encontró incompleta y el circuito también incompleto, completarlos para que sean consistentes.

$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	1	?
0	1	0	?
1	0	0	?
1	1	0	?



- d) ¿El circuito anterior cambia su valor a cada clock? ¿O para alguna entrada es estable?

Ejercicio 23² Una fábrica de lavarropas nos pide el desarrollo de la circuitería para su nuevo modelo super-económico, con una única función de lavado de treinta minutos de duración.

Los elementos con los que debe contar el lavarropas son los siguientes:

- Una llave de encendido: **START/PAUSE** y un botón de **RESET**.
- Una pantalla que muestra el tiempo restante en minutos.
- Un sistema de seguridad por sobrepeso que no permite arrancar en caso de sobrecarga.

El comportamiento es el siguiente: el lavarropas se encuentra listo para funcionar al ser enchufado, por ello muestra un 30 en su pantalla. Al accionar la llave de **START**, si el sistema no está sobrecargado, el contador debe ir disminuyendo hasta llegar a 0, con la salida S_0 activa. El usuario puede deshabilitar la llave de **START** para cargar más ropa. Luego, al re-activarse debe continuar desde donde se interrumpió. Al finalizar debe apagarse la salida S_0 . En caso de tener peso de más, no debe arrancar o continuar luego de ser interrumpido. El sensor de sobrepeso mantiene una salida alta mientras el peso supere el límite establecido. El botón de **RESET** debe regresar el estado del lavarropas al estado inicial, **sólo** en caso de no estar andando.

La empresa posee una amplia experiencia en este tipo de circuitos por lo que posee varias cosas desarrolladas previamente. Entre ellas, podemos nombrar:

- Una pantalla de dos dígitos que muestra el número ingresado por las 6 entradas que posee, interpretadas como un número sin signo.
- Flip-Flops D con entradas de *set* y *reset*.
- Circuitos Contadores de 16 bits, con *reset*.
- Clock de 100HZ, con entrada de *enable*.

Se pide:

- a) Construir un *registro restador* de 6 bits, con entrada de *clock* y *reset* que lo pone en el valor 30.³
- b) Construir un circuito combinatorio de 16 entradas y una salida, tal que la salida se active cuando el número recibido sea $(6000)_{10}$.
- c) Construir el circuito del lavarropas solicitado.

²Ejercicio tomado en el primer parcial del verano de 2010.

³Ayuda: vale usar un *full-adder* de 6 bits.

Ejercicio 24 La conjetura de Collatz, es un famoso problema matemático aún no resuelto. Esta conjetura enuncia la siguiente función $f : \mathbb{N} \mapsto \mathbb{N}$, aplicable a cualquier número entero positivo:

$$f(n) = \begin{cases} \frac{n}{2} & \text{si } n \text{ es par} \\ 3n + 1 & \text{si } n \text{ es impar} \end{cases}$$

Se dice que si se toma cualquier número y se aplica esta función reiteradas veces, el resultado siempre converge a 1.

- a) Construir un circuito combinatorio que realice la función $f(n)$ para una entrada de 5 bits.
- b) Construir un circuito secuencial, que aplique reiteradas veces la función anterior por cada ciclo de reloj.
- c) Modificar el circuito anterior de forma que si el valor de entrada es 1, entonces la salida también sea 1.