

PRÁCTICA 2. Trabajo final

Implementación combinacional de un sumador e introducción a los circuitos secuenciales.

Objetivos que se deben alcanzar al realizar la práctica

Después de realizar esta práctica, además de haber mejorado el nivel de consecución de los objetivos necesarios para preparar la práctica (citados en la tabla anterior) y de los objetivos de la práctica anterior relativos al manejo del programa LogicWorks, el alumno será capaz de:

- 1) Implementar en LogicWorks un sumador binario combinacional con propagación del acarreo usando Full-adders. Encapsular el dispositivo usando buses de 16 bits para las entradas y salidas del circuito.
- 2) Obtener el tiempo de propagación del sumador binario usando LogicWorks.
- 3) Implementar en LogicWorks un circuito secuencial a partir de un grafo de estados y comprobar su correcto funcionamiento.
- 4) Obtener el tiempo de ciclo mínimo de un circuito secuencial usando LogicWorks.

Directorio de la práctica

Mi PC\R:\Logic\Ver_4.1\lc\Prac2

Estructura. Esta práctica tiene dos partes bien diferenciadas, de temática muy diferente: 1) implementación de un sumador combinacional conectando Full-adders en propagación del acarreo y 2) introducción al biestable D activado por flanco (secuencial más simple que usaremos para construir otros más complejos) e implementación de un circuito secuencial contador

2.1 Implementación combinacional de un sumador

En esta parte de la práctica, primero vamos a implementar en el simulador LogicWorks el bloque Full-adder usando varios bloques Half-adder, que habéis diseñado en el trabajo previo. Por último, vamos a usar los bloques Full-adder para diseñar e implementar un sumador de números naturales de 16 bits (una de las partes centrales de nuestro computador).

2.1.1 Implementación del Full-adder con Half-adders en LogicWorks

Implementad en LogicWorks el Fa-ha (la implementación correcta de las tres que se daban en la segunda parte de la pregunta 1 del informe previo: con dos Half-adders y alguna puerta). Para ello, utilizad el dispositivo Half-adder (Ha) de la librería de esta práctica 2, LibPrac2. El dispositivo Full-adder dentro del que tenéis que dibujar el circuito que vais a crear con vuestros Half-adders y una puerta Or-2 se encuentra en la librería con el nombre Fa-ha. Como en anteriores prácticas este dispositivo tiene adjunto un circuito con sólo los port connectors y vosotros tenéis que completar el esquema. Después de encapsularlo y salvarlo probad su funcionalidad con el simulador LogicWorks, para asegurarnos que el proceso se ha realizado correctamente.

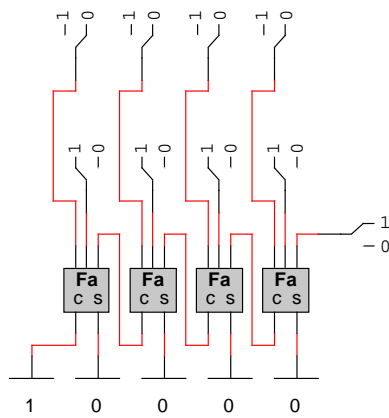
□ Informe final

Pregunta 1

Cuando estéis seguros del correcto funcionamiento del dispositivo encapsulado Fa-ha (con 2 Ha y alguna puerta), avisad al profesor de laboratorio y pedidle que revise vuestro trabajo y firme el informe final.

2.1.2 Implementación del sumador binario combinacional con propagación del acarreo usando LogicWorks

Implementad en LogicWorks el sumador binario combinacional con propagación del acarreo que hemos diseñado en esta práctica, para $n = 4$, usando dispositivos Fa-ha. Colocad Binary Switch en las entradas y Binary Probe en las salidas para comprobar su correcto funcionamiento. El resultado de vuestro trabajo tiene que ser algo parecido a lo siguiente. (Aunque las entradas del Fa-ha son intercambiables a nivel lógico no lo son para medir el tiempo de propagación. Por ello, debéis realizar el conexionado exactamente como se muestra en la figura).



□ Informe final Pregunta 2

Cuando estéis seguros del correcto funcionamiento del circuito sumador con Fa-ha, avisad al profesor de laboratorio y pedidle que revise vuestro trabajo y firme el informe final.

2.1.3 Tiempo de propagación del sumador

Vamos a comprobar en el laboratorio si son correctas vuestras respuestas a la pregunta 7 del informe previo. Para ello, en el sumador con Fa-ha, poned un Switch que vaya a la entrada del acarreo del primer Full-adder (c_0) y cuatro Switches más en las entradas de cada uno de los dos números que queremos sumar. Dad nombres a las salidas para poder visualizar en la ventana de tiempos del LogicWorks los cronogramas de las señales de salida.

□ Informe final Pregunta 3

Comprobad mediante los cronogramas del LogicWorks si son ciertas las respuestas que habéis dado a la pregunta 7 del informe previo.

- a) Suponed que en el sumador construido con Fa-ha todas las entradas han valido 0 durante mucho tiempo antes del instante t . En el instante t las entradas pasan a valer: $X = 1010$, $Y = 0101$ y $C_0 = 1$. La salida w_k se estabiliza con su valor correcto en el instante de tiempo $t+d_k$, para $k = 0, \dots, 3$ y la salida c_4 se estabiliza en el tiempo $t+d_4$. ¿Cuánto vale d_k para cada una de las cinco salidas del sumador, c_4, w_3, \dots, w_0 ?:

d_4	d_3	d_2	d_1	d_0

¿Coincide con la respuesta del informe previo? Si no coincide indicad la razón.

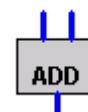
- b) Repetid el ejercicio anterior para $X = 1010$, $Y = 0101$ y $C_0 = 0$.

d_4	d_3	d_2	d_1	d_0

¿Coincide con la respuesta del informe previo? Si no coincide indicad la razón.

2.1.4 Encapsulado del circuito

Ahora vamos a encapsular un circuito sumador como el que habéis implementado pero para $n=16$, que es el tamaño de palabra del computador que construimos en este curso. El símbolo dentro del que encapsularéis el circuito lo hemos dibujado nosotros y se encuentra en la librería LibPrac2, con el nombre de ADD-P2. Si lo copiáis de la librería a un circuito veréis la figura de la derecha.



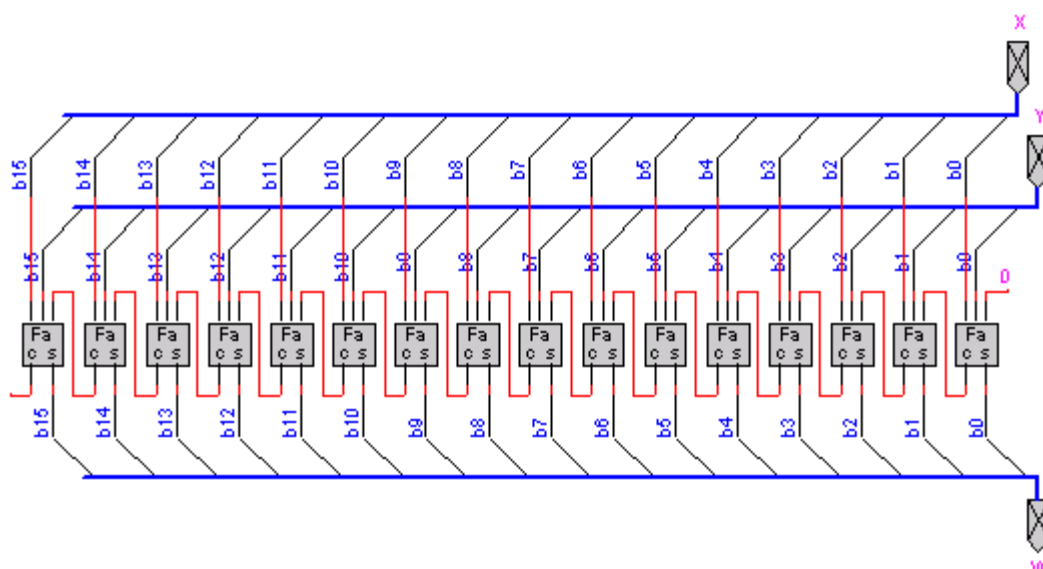
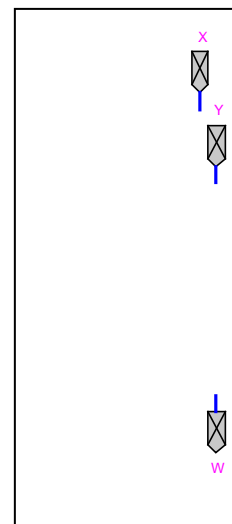
Las patillas del dispositivo no son negras y finas, como ocurría hasta ahora, sino azules y gruesas. Cada una de estas líneas azules representa un bus de n bits ($n=16$ en nuestro caso), una patilla de tipo bus. Esta es una facilidad muy interesante que nos ofrece LogicWorks: de los dispositivos pueden entrar y salir buses para interconectar dispositivos con solamente una línea gruesa que simboliza n bits.

Si hacéis doble clic ahora en el símbolo del ADD aparece el circuito que se muestra a continuación.

El circuito interno consta solamente de los PORT CONNECTORS de los buses de entrada X e Y (dispositivos PORTIN de 16 bits, que son más grandes que los que ya conocéis, PortIn, de un bit) y del bus de salida W (dispositivo PORTOUT, también más grande que los PortOut de un bit). No cambiéis el nombre de estos PORT CONNECTORS pues debe coincidir con el nombre de las patillas de buses que nosotros hemos puesto al crear el símbolo del dispositivo ADD-P2 (estos nombres deben coincidir al igual que ocurría con los nombres de los Port Connectors y las patillas de un bit).



Todos los buses que vamos a usar en este curso son de 16 bits y las 16 señales internas (los 16 bits del bus) se denominan siempre: b15, b14, ..., b2, b1, b0, independientemente del nombre del bus. Si no seguimos este convenio nuestros diseños no funcionarían, ya que esos son los nombres que hemos dado a los bits de los buses de los conectores PORTIN y PORTOUT.

Al final de esta sección vais a completar el circuito interno del dispositivo ADD-P2 con un circuito sumador con propagación del acarreo para $n=16$ bits. Así, cuando hagáis doble clic en el símbolo se abrirá su interior y aparezca un circuito como el que se muestra a continuación. Conseguir esto es vuestro objetivo ahora.



Para acceder a los cables de un bit que hay dentro de un bus, hay que usar un Breakout. Para poner un breakout en un circuito tenéis que ir al menú Schematic -> New Breakout. Nos aparece una ventana que nos pregunta

- Pin List. Aquí tenéis que copiar: b0..15 (exactamente así, de lo contrario no funcionará)
- Pin Spacing. Podéis usar el número 1, 2, 3, 4, etc. Dependiendo de lo separados que queráis que salgan los cables del bus (probad distintas opciones para ver el resultado). Si usáis el espaciado 6 como hemos usado nosotros en el circuito anterior, el diseño os cabrá en una sola página y quedará muy bonito.

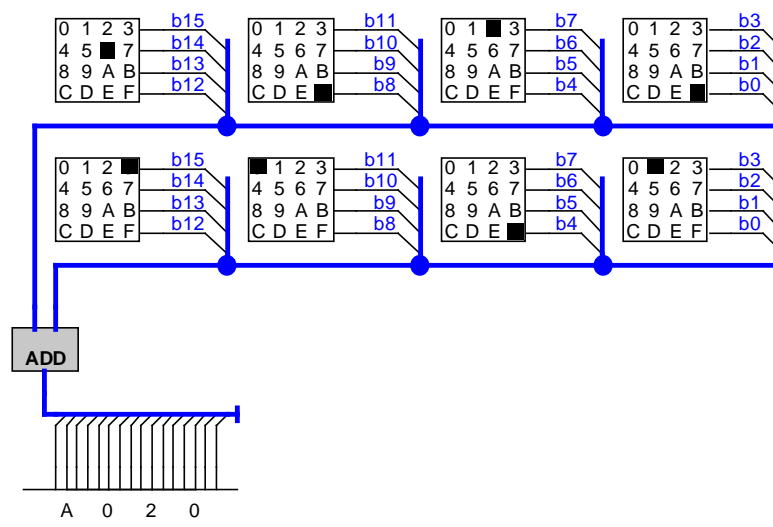
Se puede dibujar un bus de n bits (que es una línea azul gruesa) para unir, por ejemplo, el bus del PORTIN X con un breakout que obtendrá sus bits internos, o para unir dos patillas de tipo bus de dos dispositivos en un circuito complejo. Para ello se estira de la patilla de tipo bus del dispositivo (como se hace para el caso de cables de 1 bit), o se usa el botón  (equivalente al botón  usado para dibujar cables de un bit).

Después de esta pequeña introducción sobre los buses en LogicWorks vamos a dibujar el circuito en la página que se abre al hacer doble clic sobre el símbolo ADD y que sólo contiene los tres PORT CONNECTORS. Ahora ya podéis crear tres breakout, dos para conectarlos a los PORTIN X e Y y otro para conectarlo a W, como se muestra en la figura anterior. El siguiente paso consiste en pegar los 16 Full-adder a los cables que salen de los breakout.

Una vez terminado el circuito ya podéis cerrarlo. Ahora podéis seleccionar el símbolo ADD para salvar los cambios que habéis hecho en su circuito interior en la librería LibPrac2, de la misma forma que ya conocéis de la práctica anterior para los dispositivos que tienen patillas de un bit.

2.1.5 Comprobación del circuito encapsulado

Para comprobar el dispositivo con el circuito encapsulado tenéis que abrir el circuito Probe-ADD.cct. que hemos creado para que probéis cómodamente el funcionamiento de vuestro sumador. Aparecerá un esquema como este.



Tenéis que sustituir el dispositivo ADD, cuyo circuito interno sólo tiene los ports connectors, por vuestro dispositivo ADD-P2. Probad varios números haciendo clic en los teclados hexadecimales y observad la salida en los displays hexadecimales. Comprobad que el sumador funcione correctamente.

□ Informe final Pregunta 4

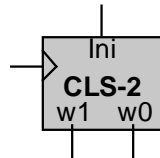
Cuando estéis seguros del correcto funcionamiento del dispositivo encapsulado ADD-P2, avisad al profesor de laboratorio y pedidle que revise vuestro trabajo y firme el informe final.

2.2 Introducción a los circuitos secuenciales

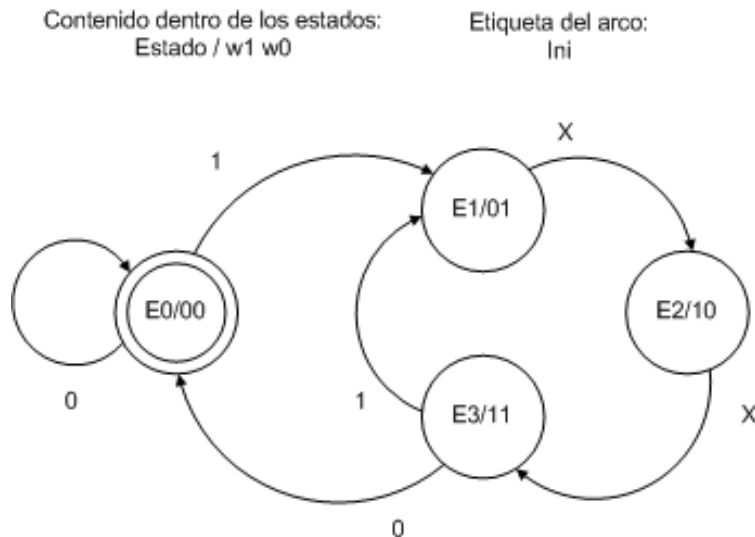
En el trabajo previo de esta segunda parte de la práctica ya estudiamos el comportamiento del biestable D activado por flanco. También diseñamos un circuito secuencial de Moore sencillo, un generador de secuencia, que a continuación vamos a implementar en LogicWorks.

2.2.1 Implementación en LogicWorks de un Generador de secuencia

El generador de la secuencia 1, 2, 3, que se repite indefinidamente, quedará encapsulado en este bloque.



Ini es la entrada y w1 w0 son las dos salidas que codifican en binario un número natural. El comportamiento del circuito queda especificado en el siguiente grafo de estados, como definimos en el Trabajo previo.

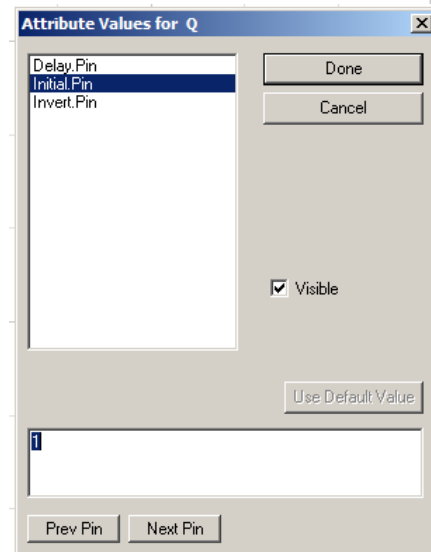
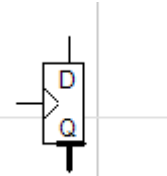



Implementad en LogicWorks el circuito que habéis diseñado en la pregunta 9 del informe previo.

Encontraréis los dispositivos flip-flop (biestable D) y puertas And-2, Or-2 y Or-4 en la LibPrac2. También se encuentra en esta librería el dispositivo CLS-2, que sólo tiene ports connectors en su circuito interno, para que encapsuléis vuestro diseño.

Recordad que en el diseño de un circuito secuencial es necesario inicializar las salidas Q de los biestables, para que cuando comience el funcionamiento los biestables se pongan en el estado inicial del circuito. Los biestables de nuestras librerías de LogicWorks se encuentran por defecto inicializados a 0. Es decir, al hacer reset en el simulador, la salida de los biestables es 0. Así que si el estado inicial de un biestable tiene que ser 0, como ocurre en esta práctica, no es necesario modificar el valor que está por defecto.

No obstante, si al hacer reset quisiéramos que un biestable concreto se inicialice a 1, habría que haber modificado previamente la definición de ese biestable con el siguiente procedimiento. Haced clic con el botón derecho del cursor en la patilla Q del biestable (en la parte de la patilla cercana a la caja). La patilla se selecciona como indica la figura y aparece una ventana de dialogo. Seleccionad Attributes y aparece otra ventana en la que tenéis que seleccionar InitialPin, Visible y poner un 1 la caja del valor, como muestra la figura. Por último haced clic en Done y ya está. Aparece un 1 en azul al lado de patilla para recordar que ese biestable se inicializará a 1 al hacer reset.



Por último, antes de comenzar la simulación no olvidéis de hacer clic en el botón de reset del simulador, , para inicializar los biestables de la unidad de control al valor que tienen asignado.

□ Informe final

Pregunta 5

Cread un circuito con el dispositivo CLS-2 que habéis construido, con un Binary Switch conectado a la entrada Ini, un BinarySwitchClk conectado a la entrada de reloj (este switch es distinto, tiene un pequeño retardo para que los cronogramas se vean con más claridad, no hace falta que entendáis el porqué) y un Binary Probe conectado a cada una de las salidas. Comprobad que el funcionamiento del circuito es el correcto creando secuencias de señales Ini y Clk parecidas a las de la pregunta 10 del informe previo. Comparad los resultados con los del informe previo. Cuando estéis seguros del correcto funcionamiento del dispositivo encapsulado CLS-2, avisad al profesor de laboratorio y pedidle que revise vuestro trabajo y firme el informe final.

Informe final Práctica-2

Apellidos y nombre: Grupo:

Apellidos y nombre: Grupo:

(por orden alfabético)

Pregunta 1 (1 punto)

Fa-ha (con 2 Ha y alguna puerta)

Comentario del profesor:

Firma del profesor:

Pregunta 2 (1 punto)

Sumador de 4 bits

Comentario del profesor:

Firma del profesor:

Pregunta 3

a) (1.5 puntos)

d ₄	d ₃	d ₂	d ₁	d ₀

(0.5 puntos) ¿Coincide con la respuesta del informe previo? Si no coincide indicad la razón.

b) (1.5 puntos)

d ₄	d ₃	d ₂	d ₁	d ₀

(0.5 puntos) ¿Coincide con la respuesta del informe previo? Si no coincide indicad la razón.

Pregunta 4 (1 punto)

ADD-P2

Comentario del profesor:

Firma del profesor:

Pregunta 5 (3 puntos)

CLS-2

Comentario del profesor:

Firma del profesor: