

Manual del entorno

BOOLE-DEUSTO

13.1 Introducción

En este apéndice presentaremos el entorno BOOLE-DEUSTO de análisis y diseño de sistemas digitales combinacionales y secuenciales a nivel de bit. Este entorno está orientado a las necesidades del alumno y el profesor, sobre todo en los primeros pasos de su aprendizaje.

BOOLE está orientado a una disciplina denominada genéricamente Sistemas Digitales, que consta de tres grandes temas:

- ◆ Fundamentos de Sistemas Digitales: codificación, álgebra de Boole y representación de sistemas digitales.
- ◆ Análisis y Diseño de Sistemas Combinacionales: a nivel de bit y de palabra.
- ◆ Análisis y Diseño de Sistemas Secuenciales: a nivel de bit y de palabra.

El profesor en clase enuncia el método a aplicar y completa algún ejemplo. El alumno en casa debe consolidar lo aprendido, completando nuevos ejercicios. El papel de un entorno educativo en este escenario es doble:

- ◆ Comprobar que los resultados obtenidos por el alumno están bien.
- ◆ Mostrar al alumno cómo se aplica un método; recordarle los pasos.

Para que el entorno lleve a cabo con éxito su trabajo debe tener una serie de características:

- ◆ Completo. Debe contemplar el mayor número posible de métodos.
- ◆ Utilizable. Debe adaptarse a los conocimientos y posibilidades del alumno.
- ◆ Amigable. Debe ser fácil e intuitivo de usar y visual en los resultados.
- ◆ Didáctico. Debe mostrar todos los pasos que le llevan a un resultado.
- ◆ Instalable. Debe ser fácil de instalar.

- ◆ Asequible. Debe tener el menor precio posible.

En no pocas ocasiones los entornos computacionales no ayudan al alumno; es más, le confunden, generando frustración y desconocimiento. A nuestro parecer es mejor no utilizar un entorno computacional si éste es inadecuado.

Estudiemos a la luz de los puntos anteriores la idoneidad didáctica de los entornos profesionales como OrCAD, MAX PLUS II, XILINX Foundation, Electronic WorkBench, etc. En primer lugar recordemos que estos entornos son profesionales; su objetivo es ayudar en el diseño de circuitos electrónicos, no ayudar al alumno. Veamos algunas de sus características:

- ◆ No le interesan los métodos, le interesan sus resultados: los circuitos electrónicos.
- ◆ Sus usuarios finales son los profesionales.
- ◆ Su uso es tan potente como complicado.
- ◆ No son didácticos; sólo buscan los resultados.
- ◆ Son difíciles de instalar.
- ◆ Son generalmente caros.

Resulta tan evidente como sencillo que los objetivos de los entornos profesionales están alejados de las necesidades del profesor y del alumno en el aula. La tabla 13.1 compara ambos entornos, resaltando las diferencias.

ENTORNO BOOLE-DEUSTO	ENTORNOS PROFESIONALES
Didáctico	Profesional
Necesidades del alumno	Necesidades del profesional
Hasta nivel de bit	Hasta nivel de sistema
Proyectos sencillos	Proyectos complejos
El alumno controla a la herramienta	El usuario es dirigido por la herramienta
Interesa el proceso	Interesa el resultado
Sin instalación	Instalación compleja
Es muy fácil de usar	Críptico y difícil de usar
Es gratis y de libre distribución	Coste generalmente elevado
No tiene simulación temporal	Si tiene simulación temporal
No permite la captura gráfica de un circuito	Permite la captura gráfica de un circuito

Tabla 13.1 Comparación entre BOOLE y los entornos profesionales

Todo lo anterior viene a fortalecer la necesidad de un entorno completo y didáctico en la disciplina de sistemas digitales: BOOLE-DEUSTO.

13.2 Aspectos básicos de uso del BOOLE-DEUSTO

El entorno BOOLE-DEUSTO es muy fácil de utilizar, es más, éste es uno de los objetivos principales, incluso más importante que cubrir una mayor parte de la asignatura. El entorno ha de ser útil al alumno y al profesor con poco esfuerzo.

Se puede decir que 15 minutos son más que suficientes para enseñar al alumno a utilizar su parte combinacional, y otro tanto para su parte secuencial.

En los próximos apartados se presentará en detalle el entorno BOOLE-DEUSTO, pero es ahora momento de recalcar los aspectos básicos:

- ♦ Los sistemas han de ser secuenciales o combinacionales, pero siempre a nivel de bit.
- ♦ Los sistemas combinacionales tienen como núcleo a la tabla de verdad, mientras que los secuenciales son autómatas.
- ♦ Los sistemas combinacionales tienen como máximo 10 variables de entrada.

- ♦ Los sistemas combinacionales soportan como máximo y en principio *****
- ♦ Los nombres de las variables pueden ser cambiados.
- ♦ Los autómatas pueden ser de Moore o de Mealy.
- ♦ Al describir un sistema hay que empezar dándole nombre e indicando el número de variables de entrada y de salida.
- ♦ Al salir de una pantalla de captura de datos hay que pulsar siempre **E**valuar (y **S**alir) para que los datos sean actualizados.
- ♦ Las funciones se cargan y se visualizan de una en una, utilizando la barra de desplazamiento.
- ♦ Las tablas de verdad y diagramas de Veitch-Karnaugh no deben tener huecos (se puede utilizar la opción **Completar con**).
- ♦ Se pueden imprimir los resultados tanto para sistemas secuenciales como combinacionales.
- ♦ Los elementos gráficos (circuitos lógicos y diagramas de transición de estados) pueden ser copiados al portapapeles, y de él a cualquier documento.
- ♦ Todo sistema, ya sea combinacional o secuencial, puede ser guardado y cargado como un fichero.
- ♦ Las pantallas que tengan alguna complicación disponen de **Ayuda**.

Para describir el entorno BOOLE nos apoyaremos en ejemplos.

13.3 Instalación y uso

Para instalar BOOLE bastará con copiar el ejecutable, o con copiar y descomprimir el fichero .zip. La instalación no puede ser más sencilla, de hecho fue un requisito a la hora de diseñar el entorno. De este modo el alumno le basta con crear una carpeta (o no), copiar el programa, hacer doble clic sobre él y empezar a trabajar.

En cuanto al uso, éste es libre para cualquier usuario, quedando expresamente prohibida su distribución comercial de cualquier modo sin el consentimiento de los autores (BOOLE está en el Registro de la Propiedad Intelectual).

Se solicita a todos los usuarios que se registren en la dirección electrónica zubia@eside.deusto.es, y así poder enviarles las nuevas versiones del entorno.

13.4 Sistemas combinacionales con BOOLE

Antes de pasar a BOOLE, recordemos qué es diseñar. El diseño de un sistema combinacional pasa por varias fases:

- ♦ Leer y entender el enunciado.
- ♦ Determinar las variables de entrada y salida.
- ♦ Obtener de la tabla de verdad.
- ♦ Obtener las formas normales de cada variable de salida.
- ♦ Obtener los diagramas de Veitch-Karnaugh de cada salida.
- ♦ Simplificar cada V-K, obteniendo la expresión simplificada.
- ♦ Opcionalmente, reescribir la expresión anterior desde las puertas NAND o NOR.
- ♦ Obtener el circuito lógico.
- ♦ Implementar el circuito digital con circuitos integrados.

- ◆ Probar el circuito implementado.

Los dos primeros pasos deben ser dados por el alumno/profesor, y los dos últimos quedan fuera del BOOLE; pertenecen al laboratorio, con programas como Electronic WorkBench, OrCAD, etc. Centrémonos mediante un ejemplo en los seis pasos restantes.

13.4.1 Ejemplo 1 de sistema combinacional

El ejercicio que se propone consiste en diseñar el circuito transcodificador de BCD puro a BCD Aiken.

Las variables de entrada son cuatro, las mismas que de salida. Así pues la tabla de verdad a completar tendrá 16 filas y cuatro salidas.

La figura 13.1 muestra la primera imagen que el alumno ve al activar BOOLE; en esta pantalla elegirá la opción **Sistemas Combinacionales**.



Figura 13.1 Pantalla inicial de BOOLE

La siguiente imagen será la de la figura 13.2. En ella el alumno obligatoriamente debe indicar el nombre del sistema, el número de variables de entrada y el número de variables de salida. Opcionalmente podrá dar un nombre a las variables, teniendo en cuenta que BOOLE asigna un nombre por defecto.

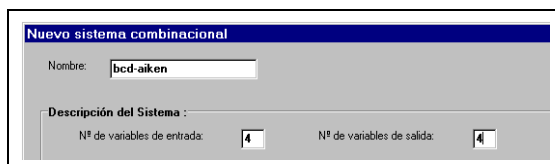


Figura 13.2 Declaración del sist. combinacional

Una vez declarado el sistema, el alumno debe describirlo. La figura 13.3 muestra todas las opciones que ofrece BOOLE; en este caso optaremos por la opción **Tabla de Verdad Manual**.

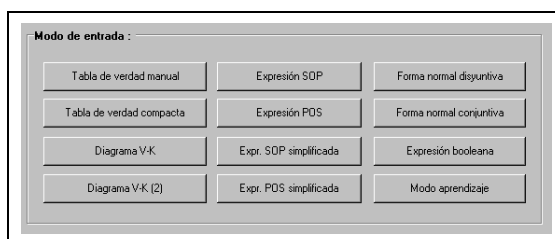


Figura 13.3 Operaciones booleanas de BOOLE

Al activar esta opción el alumno verá las 16 filas ordenadas, debiendo escribir él las salidas mediante clics de ratón. Al ponerse encima de una casilla, cada clic de ratón cambia el valor booleano. Además no hay que cumplimentar toda la tabla, se pueden escribir sólo los 1's y rellenar automáticamente el resto con 0's, o escribir sólo los 1's y 0's y rellenar el resto con condiciones libres.

La figura 13.4 muestra la tabla de verdad del transcodificador. Para que el sistema sea cargado es necesario activar **Evaluar**, y luego **Salir**. A partir de este momento todas las operaciones de BOOLE serán aplicadas a este sistema.

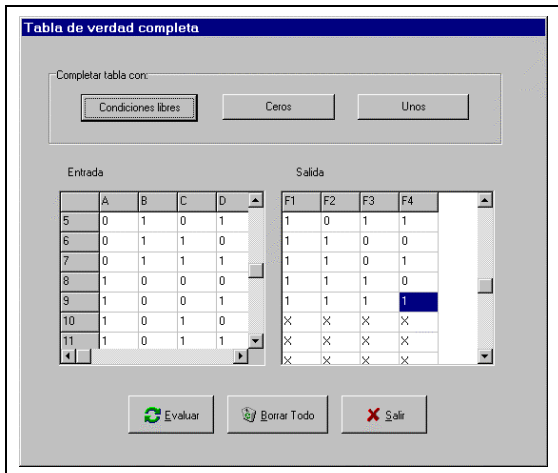


Figura 13.4 Tabla de Verdad Manual

Al volver a la figura 13.3, si el alumno activara **Forma Normal Disyuntiva** obtendría la figura 13.5. En ella sólo se ve una salida, la primera, para ver las restantes basta con pulsar en la barra de desplazamiento de la derecha.

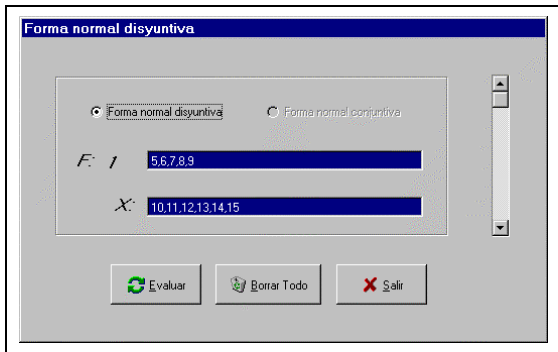


Figura 13.5 Forma Normal Disyuntiva

Si el alumno quisiera, podría cambiar el sistema borrando los números de F1 y escribiendo otros nuevos. Pero sigamos con el sistema tal y como lo hemos descrito al principio.

Si después de pulsar **Salir** en la anterior figura, el alumno pulsara **V-K** en la pantalla principal, se encontraría con la figura 13.6. Para ver los V-K de las restantes salidas, bastará con pulsar en la barra de desplazamiento. Además de la opción V-K

el alumno dispone de V-K 2; con esta opción obtendría un V-K ordenado de distinta manera (según el código Gray). De esta forma **BOOLE** busca adaptarse a las necesidades de los distintos profesores y alumnos.

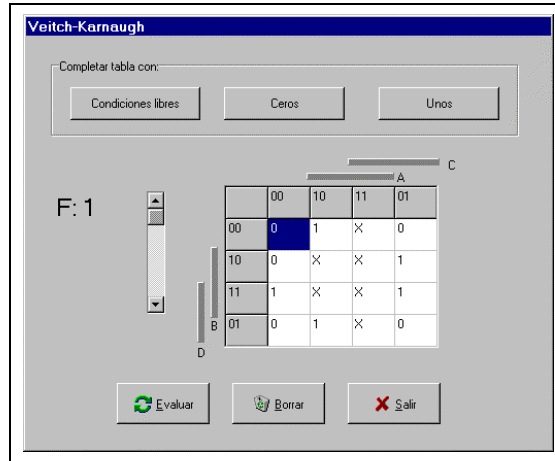


Figura 13.6 Diagrama de Veitch-Karnaugh

El anterior V-K tiene por objetivo facilitar la simplificación visual de la función. Al pulsar **Salir** en la figura 13.6 y **Expr. SOP Simplificada** en la pantalla principal de la figura 13.3, el alumno obtendrá las expresiones simplificadas de cada una de las salidas. Dichas expresiones lo serán en forma de suma de productos (SOP), aunque también podría haber optado por POS. Las expresiones obtenidas son mínimas necesariamente (el algoritmo implementado es recursivo y exacto), aunque el conjunto no tiene por qué serlo, es decir, simplifica cada función por separado.

La figura 13.7 muestra la expresión mínima de la salida 2 (para ver el resto pulsaremos en la barra de desplazamiento).

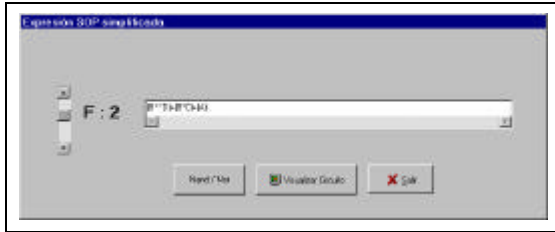


Figura 13.7 Expresión SOP Simplificada

Si el alumno quisiera reescribir las salidas desde las puertas NAND o NOR, le bastaría con activar **Nand/Nor** en la anterior figura. La figura 13.8 es un ejemplo de ello.



Figura 13.8 Expresiones NAND/NOR

Para ver los circuitos lógicos el alumno deberá pulsar **Visualizar Circuito** en la figuras 13.7 o 13.8, resultando las figuras 13.9 y 13.10.

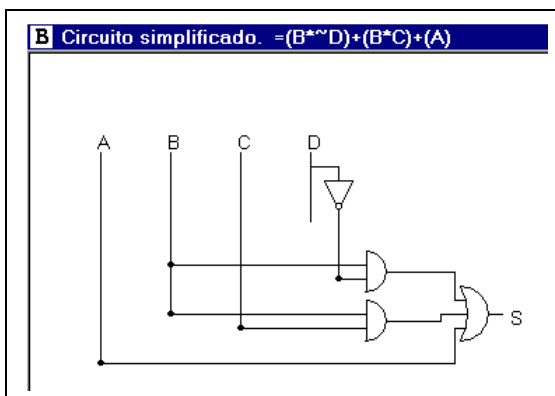


Figura 13.9 Circuito Lógico AND-OR

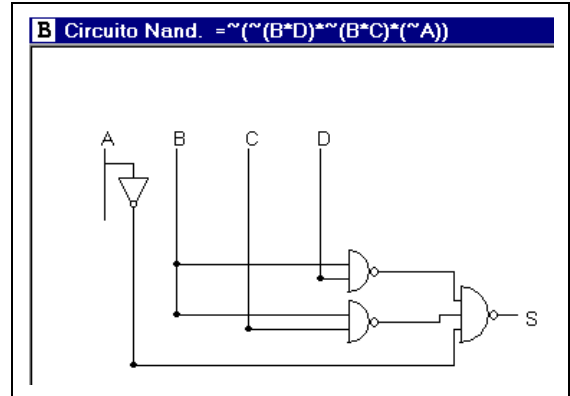


Figura 13.10 Circuito Lógico NAND

Llegado a este punto el alumno habrá completado todo el proceso de diseño paso por paso. Aunque si hubiera querido también habría podido introducir la tabla de verdad y seguidamente pulsar **Expr. SOP Simplificada**, ahorrándose los pasos intermedios; actitud que se debe evitar mientras se esté aprendiendo a diseñar.

Antes de pasar a diseñar otro sistema combinacional, el alumno podrá guardar el sistema, imprimir los resultados obtenidos u obtener el código correspondiente al sistema en OrCAD-PLD. La figura 13.11 muestra dichas opciones en la pantalla principal.

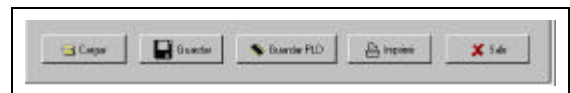


Figura 13.11 Servicios generales de BOOLE

Si se optara por **Guardar PLD**, se obtendría un programa (ver figura 13.12) listo para ser compilado en OrCAD y finalmente grabado en la PAL elegida.

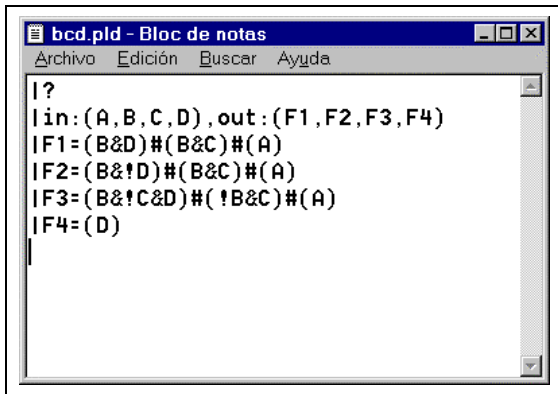


Figura 13.12 Programa en OrCAD-PLD

13.4.2 Ejemplo 2 de sistema combinacional

En este caso se va a diseñar con BOOLE un codificador 8:3 con prioridad.

La figura 13.13 declara al sistema por su nombre y número de variables de entrada y salida.

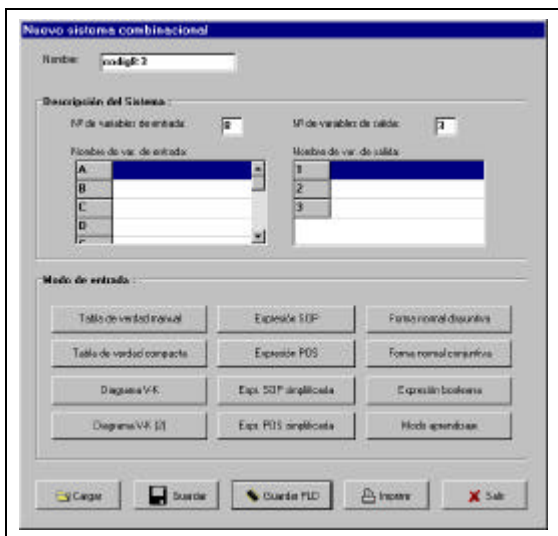


Figura 13.13 Pantalla principal al completo

Optaremos en este caso por describir el sistema con **Tabla de Verdad Compacta**. Como se puede ver en la figura

13.14, en este caso el alumno es responsable de la entrada y la salida. A cambio puede asociar condiciones libres en la entrada, es decir, escribir varias filas en una sola. De este modo en vez de responder a 256 filas, el alumno escribirá solamente 8. Esta opción es muy cómoda (y peligrosa) en determinadas ocasiones.

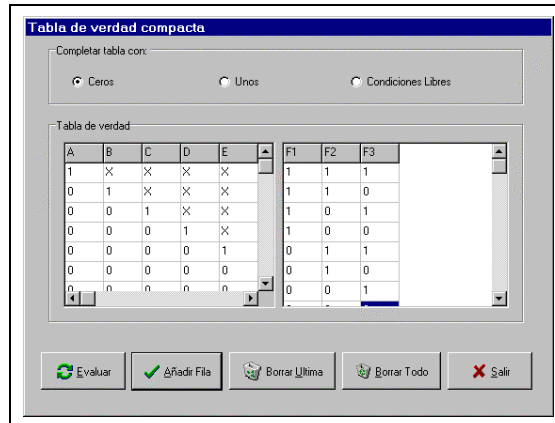


Figura 13.14 Tabla de Verdad Compacta

Si al utilizar esta opción y las condiciones libres, el alumno no escribiera todas las filas BOOLE las tomaría como condiciones libres, 0's ó 1's, según lo elegido en la parte superior de la figura 13.14.

Una vez cargado el sistema, el alumno puede pasar por todas las fases del ejemplo anterior o activar directamente **Expr. SOP Simplificada**. Las figuras 13.15 y 13.16 muestran la expresión simplificada de la salida 3 (la de menos peso) y el circuito lógico correspondiente, respectivamente.

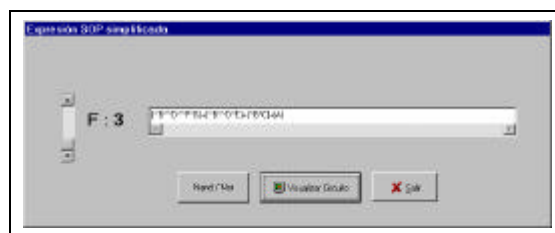


Figura 13.15 Expresión SOP Simplificada

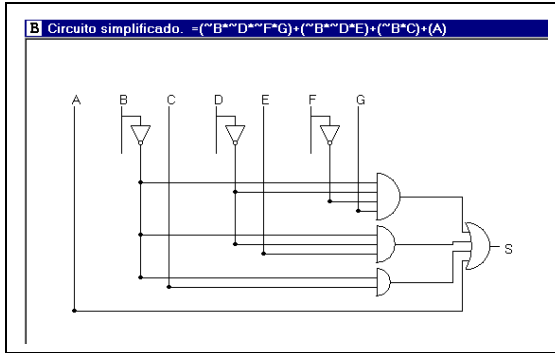


Figura 13.16 Circuito Lógico AND-OR

Los ejemplos han permitido describir los pasos a dar para diseñar un sistema combinacional. Ahora bien la parte combinacional del entorno BOOLE permite otros usos de gran valor didáctico, como se ve a continuación.

13.4.3 Análisis de sistemas combinacionales

En el caso de sistemas combinacionales, y en general, se puede decir que analizar es seguir el camino inverso de diseñar, es decir, dado un circuito o unas ecuaciones obtener su tabla de verdad para comprobar su idoneidad. El primer caso queda fuera de los objetivos del BOOLE, (cae dentro de los entornos profesionales: Electronic WorkBench, OrCAD, etc), mientras que el segundo si es contemplado por BOOLE.

Por ejemplo, el alumno puede comprobar si las expresiones por él obtenidas se comportan como un circuito de complemento a 9 (C3-0) para una entrada BCD (E3-0).

$$C3 = (\sim E3 * \sim E2 * \sim E1) \quad C1 = (E1)$$

$$C2 = (E2 * \sim E1) + (\sim E2 * E1) \quad C0 = (\sim E0)$$

Para comprobar dichas ecuaciones, el alumno habrá declarado el sistema con su nombre, 4 entradas y otras tantas salidas, y seguidamente activará **Expresión Booleana**. Para introducir las expresiones booleanas habrá que seguir las indicaciones que aparecen en la **Ayuda** de la pantalla.

Introducidas una a una las cuatro funciones (utilizando la barra de desplazamiento) se obtiene la figura 13.17.

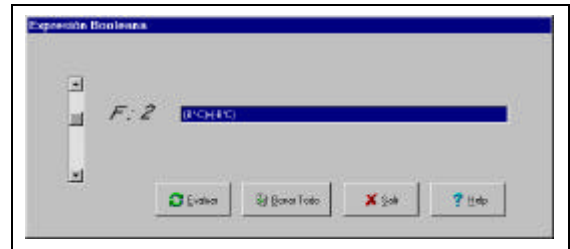


Figura 13.17 Captura de Expresión Booleana

Después de pulsar **Evaluar** y **Salir** el alumno volverá a la pantalla principal, donde al activar **Tabla de Verdad Manual** obtendrá la tabla correspondiente a dichas ecuaciones, comprobando su validez. La figura 13.18 muestra la tabla de verdad de las ecuaciones anteriores y su validez.

Completar tabla con:								
<input type="button" value="Condiciones libres"/> <input type="button" value="Ceros"/> <input type="button" value="Unos"/>								
Entrada				Salida				
	A	B	C	D	F1	F2	F3	F4
0	0	0	0	0	1	0	0	1
1	0	0	0	1	1	0	0	0
2	0	0	1	0	0	1	1	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	0	1
5	0	1	0	1	0	1	0	0
6	0	1	1	0	0	0	1	1
7	1	1	1	1	0	0	1	0

Figura 13.18 Tabla de Verdad del sistema

BOOLE permite otros caminos de análisis, bastará con que el alumno introduzca el sistema mediante una técnica de representación y active otra para observar el resultado. De hecho, y visto así, análisis y diseño se distinguen en el punto de vista, y no en los métodos o técnicas de representación utilizadas.

13.4.4 Simplificación de funciones

En el proceso de aprendizaje del alumno un paso costoso para él es aprender a simplificar funciones booleanas. El método en sí no es difícil de explicar, aunque sí lo es su aplicación. Como método heurístico que es, sólo se aprende practicando: dada una función, el alumno la simplifica y comprueba su acierto.

Visto desde el punto de vista de la clase, el profesor explica los pasos, completa algún ejercicio y deja el resto del trabajo al alumno. En este escenario BOOLE es de gran ayuda tanto para el profesor como para el alumno.

Si el alumno eligiera en la pantalla de sistema combinacionales la opción Modo Aprendizaje (una vez dado nombre y declarado el número de variables de entrada y salida), vería que aparecen el V-K del sistema cargado y una línea para que escriba la expresión que él cree mínima. BOOLE analiza lo introducido, indicando al alumno:

- ◆ la expresión es incorrecta (no se corresponde con el V-K),
- ◆ la expresión es correcta pero no mínima (se corresponde con el V-K pero no es todo lo mínima que se puede)
- ◆ la expresión es correcta y mínima (acierto).

Las figuras 13.19, 13.20 y 13.21 muestran un V-K con tres respuestas distintas. Para escribir las expresiones el alumno deberá seguir los consejos de la **Ayuda** y pulsar **Evaluar**.

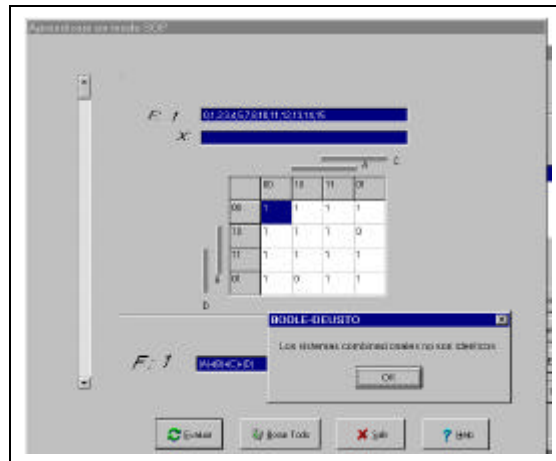


Figura 13.19 Modo Aprendizaje

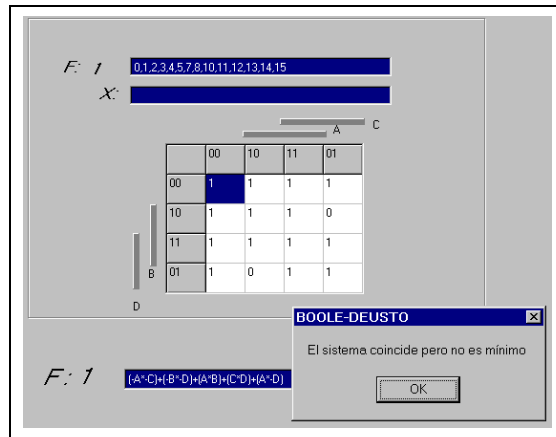


Figura 13.20 Modo Aprendizaje

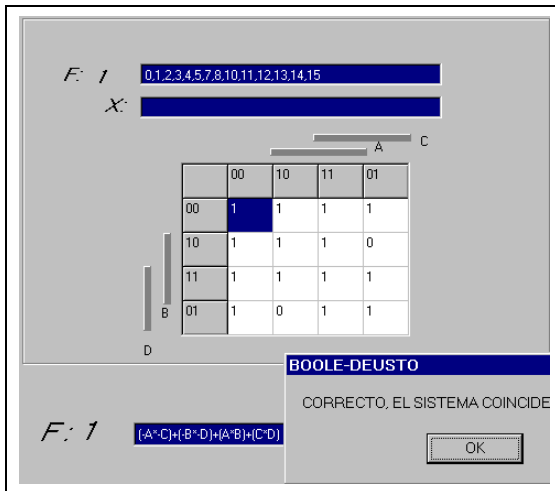


Figura 13.21 Modo Aprendizaje

Como se puede ver en las figuras, BOOLE no muestra el resultado correcto, sólo informa al alumno de lo correcto de su opción. Si quisiera ver el resultado, debería volver a la pantalla principal y activar **Expr. SOP Simplificada**.

13.4.5 Metodología booleana

A la hora de diseñar o analizar se aplican métodos para pasar de una representación a otra, y así diseñar es proceder de una forma ordenada en estas transformaciones. Sin embargo, muchas veces antes de que el alumno dé estos pasos ordenados los practica por separado. Por ejemplo:

- ◆ Obtener la forma normal de una tabla de verdad, y viceversa.
- ◆ Obtener el diagrama de V-K de una tabla de verdad, y viceversa.
- ◆ Obtener el diagrama de V-K de una forma normal, y viceversa.
- ◆ Obtener la suma de minitérminos (o producto de maxitérminos) de una tabla de verdad, y viceversa.
- ◆ Obtener la expresión NAND/NOR de una expresión SOP o POS.

- ◆ La obtención del circuito lógico de una expresión booleana.

Todas las anteriores transformaciones (y otras) son contempladas por BOOLE, sin más que introducir el sistema y pulsar la opción deseada. Es bajo este uso cuando toma sentido la idea de Calculadora Booleana asociada a BOOLE (además de ser una herramienta de análisis y diseño).

Por ejemplo, la figura 13.22 muestra la obtención de las expresiones NAND/NOR de una suma de productos.

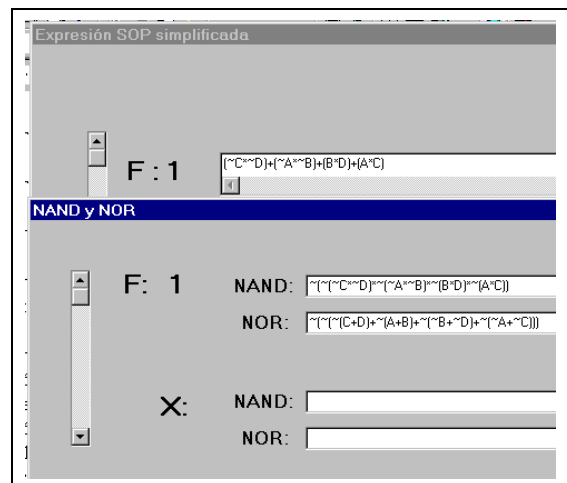


Figura 13.22 Expresiones NAND/NOR

Para obtener el circuito lógico de una expresión, algo tan sencillo como incómodo de hacer en clase, basta con que el alumno llegue a la pantalla de visualización de circuitos y escriba en la parte inferior la ecuación a visualizar. Al activar **Visualizar**, BOOLE le mostrará al alumno el circuito lógico de la expresión introducida. La figura 13.23 muestra el circuito lógico de una expresión cualquiera.

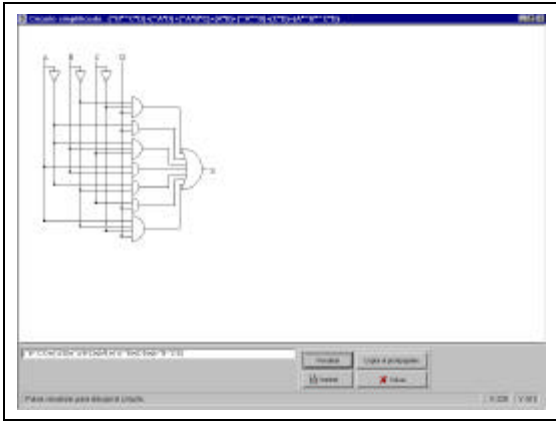


Figura 13.23 Circuito lógico de una ecuación

Con estos sencillos ejemplos queda claro que BOOLE es un excelente compañero del alumno en sus primeros pasos metodológicos, cuando se acerca por primera vez al álgebra de Boole y las representaciones booleanas. Es probable que este interés del alumno sea poco duradero en el tiempo (se aprende rápido), pero intenso.

13.5 Sistemas secuenciales con BOOLE

BOOLE permite analizar y diseñar sistemas secuenciales a nivel de bit, llamados también autómatas. El comportamiento de BOOLE con autómatas es distinto al visto para combinacionales. En este caso BOOLE no permite modificar el sistema en cualquiera de las fases, ya que esto no tiene sentido y no aporta nada didácticamente al entorno.

Las fases de diseño de un autómata son:

- ◆ Leer y entender el enunciado.
- ◆ Determinar las variables de entrada y de salida.
- ◆ Obtener el Diagrama de Transición de Estados (DTE), ya sea Moore o Mealy, indistintamente.

- ◆ Obtener la Tabla de Transición de Estados y Salidas.
- ◆ Obtener la Tabla de Codificación de Estados.
- ◆ Obtener la Tabla Codificada de Transición de Estados y Salidas.
- ◆ Obtener la Tabla de Excitación de Biestables D o J-K, indistintamente.
- ◆ Obtener los diagramas de V-K de cada J-K o D y de cada salida.
- ◆ Obtener las expresiones mínimas de las entradas de los biestables (J-K o D) y de las salidas.
- ◆ Obtener el circuito lógico con biestables J-K o D.
- ◆ Implementar el circuito digital correspondiente con circuitos integrados.
- ◆ Probar el circuito implementado.

Al igual que para sistemas combinacionales, los dos primeros pasos y los dos últimos no son contemplados por BOOLE, el resto sí.

13.5.1 Ejemplo 1 de sistema secuencial

El circuito a diseñar debe comportarse como un sumador serie. El sistema recibe en serie dos entradas A y B y ofrece también en serie en la salida la suma de dichas entradas.

Lo primero que el alumno debe hacer es elegir **Sistema Secuencial** (ver figura 13.24).



Figura 13.24 Pantalla de inicio de BOOLE

La imagen que el alumno verá es la de la figura 13.25. Quizá, el cuadradito con las flechas y el círculo quede fuera de la *mesa de trabajo*, pero el alumno puede seleccionarlo y arrastrarlo hasta ella.

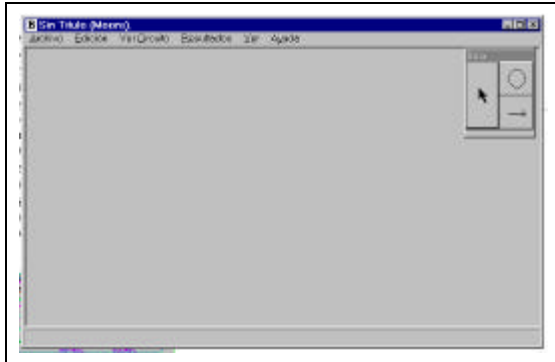


Figura 13.25 Mesa de trabajo de autómatas

Seguidamente el alumno debe declarar el autómata a diseñar. Para ello desplegará el menú **Archi vo**, y en él elegirá la opción **Nuevo**, apareciendo la pantalla de la figura 13.26. El autómata a diseñar tiene dos entradas y una salida, y será descrito como Moore. Pulsando **Aceptar**, BOOLE vuelve a mostrar la mesa de trabajo.

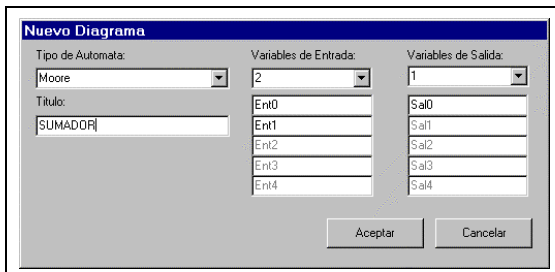


Figura 13.26 Declaración del autómata

Una vez en la mesa de trabajo el alumno puede introducir el DTE gráficamente. En este aspecto BOOLE se muestra muy ágil y cómodo, de manera que cargar el DTE es tan fácil como vistoso.

Recordemos que un autómata tiene estados, transiciones, entradas y salidas, así BOOLE ofrece el círculo (estado) y la flecha

pequeña (transiciones). La flecha grande sirve para seleccionar un estado o una transición. Veamos la captura del DTE del autómata de Moore de un sumador serie.

Por ejemplo, lo primero es situar los cuatro estados del autómata, seleccionando el círculo y haciendo simplemente clic con el ratón en la posición deseada. Así se obtiene la figura 13.27.

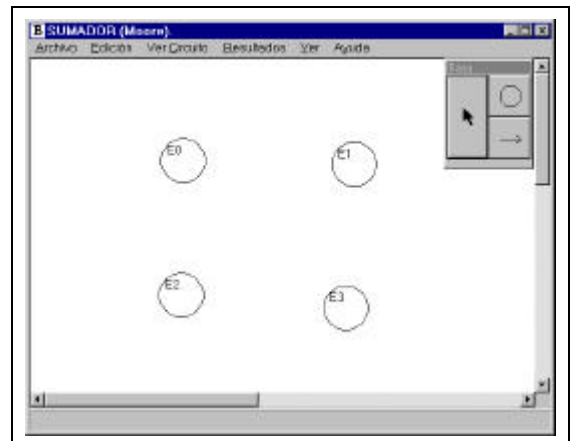


Figura 13.27 Situación de los estados

Antes de seguir digamos cómo pasar de flecha grande a pequeña o círculo. Hay dos formas:

- ♦ Normal: basta con ir al cuadrado y seleccionar con el ratón.
- ♦ Rápido: al pulsar el botón derecho tras colocar un estado o una transición se pasa a la flecha grande.

De los dos el más recomendable es el primero, aunque con el tiempo uno se acaba acostumbrando al segundo.

Si al situar un estado quisiéramos moverlo bastaría con seleccionarlo en modo flecha grande (las dos opciones de antes) y moverlo a la nueva posición.

Para borrar un estado habrá que seleccionarlo en modo flecha grande y luego pulsar el botón derecho.

Aunque el orden de los pasos dependerá del diseñador y del ejercicio en particular, parece lógico seguir con la signación de las

salidas a cada estado (por ser un autómata de Moore). Los estados son cuatro S0C0, S1C0, S0C1 y S1C1, el primero es cuando la suma es 0 y el acarreo de salida también, el resto de estados se entienden del mismo modo. Para asignar la salida al estado habrá que seleccionarlo en modo flecha grande y hacer doble clic sobre el estado. Al hacer esto aparece una pantalla en la que se escribirá el nombre del estado y las salidas asignadas (ver figura 13.28). La figura 13.29 muestra al autómata tras la asignación de las salidas a los estados.

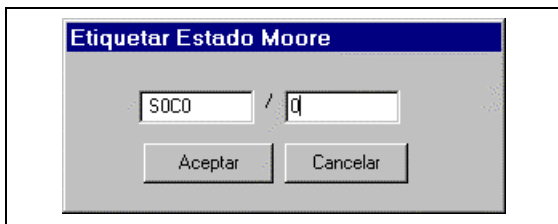


Figura 13.28 Etiquetado de un estado Moore

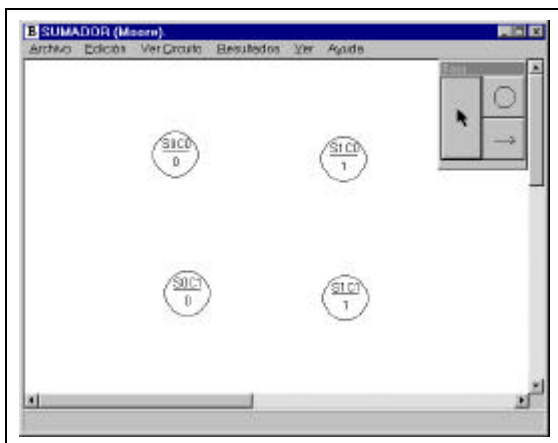


Figura 13.29 Estados de Moore etiquetados

Situados los estados con su nombre y salida, el alumno debe dibujar las transiciones. Simplemente basta con seleccionar la flecha pequeña, hacer clic en el estado origen y clic en el estado destino; dibujando BOOLE la transición. Para hacer un autolazo basta con hacer los dos clics en el mismo estado.

Acabado este paso el autómata será el de la figura 13.30.

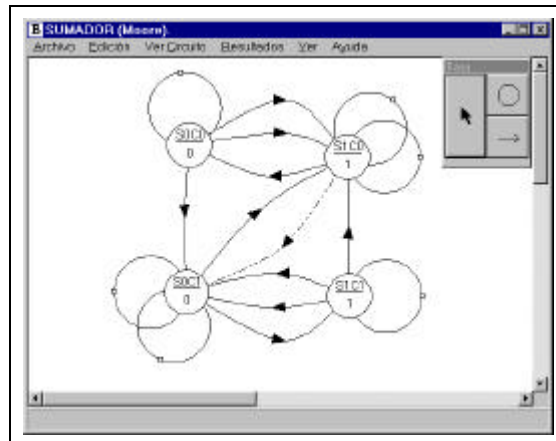


Figura 13.30 Autómata con transiciones

Al situar las transiciones pueden darse dos situaciones incómodas: que una transición quede encima de otra o que no se vea la punta de flecha de la transición, en ambos casos la solución pasa por mover las transiciones.

En el primer caso, basta con estar en modo flecha grande, seleccionar agarrar la punta de flecha (la flecha del autolazo es un cuadrado) y arrastrarla.

En el segundo caso (¿dónde está la punta de flecha?) habrá que pasar a modo flecha grande, seleccionar el estado destino y moverlo, pues la punta de flecha estará debajo de él. Una vez que la flecha esté a la vista, se procede como en el párrafo anterior.

Todo lo dicho puede parecer complicado, pero es muy sencillo acostumbrarse al entorno de dibujo.

En este momento sólo resta asociar las entradas a las transiciones. Al igual que para los estado basta con seleccionar la flecha, hacer doble clic y escribir la entrada de la transición (ver figura 13.31). Al acabar el autómata tendrá el aspecto de la figura 13.32.

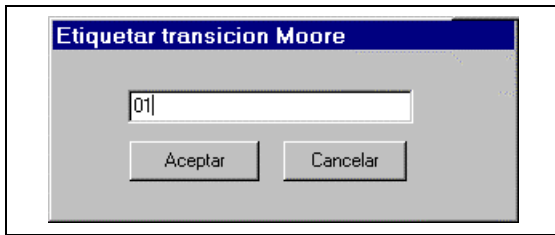


Figura 13.31 Etiquetado de una transición Moore

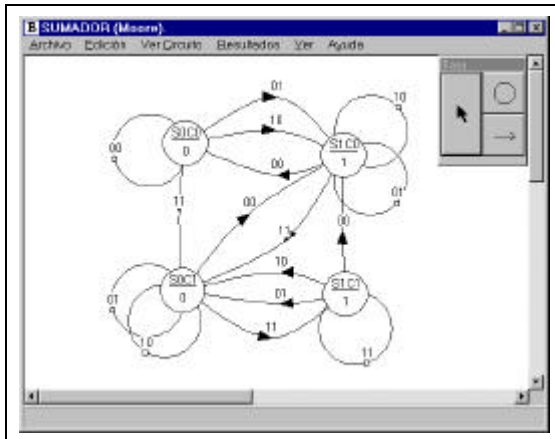


Figura 13.32 Automata de un sumador serie

Aunque no es el caso, el alumno puede asociar a una transición condiciones libres en la entrada, como por ejemplo se muestra en la figura 13.33.

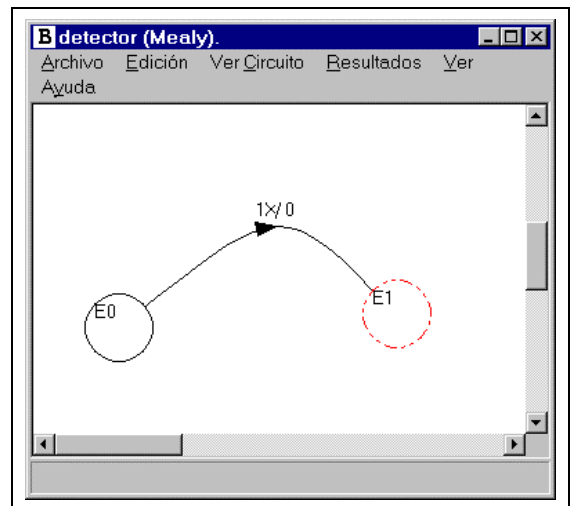


Figura 13.33 Uso de condiciones libres

En este momento el autómata de Moore ya está descrito mediante su DTE, y el alumno ya puede pasar a ver los pasos del diseño. Antes de diseñar se pueden activar dentro del menú **Resultados** las opciones **Verificar Corrección** y **Verificar Completitud**, que nos indican si hemos olvidado alguna transición, o si alguna de ellas está duplicada. La figura 13.34 nos muestra el resultado de dichas verificaciones.

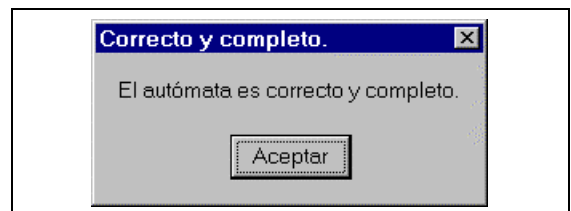


Figura 13.34 Verificación del autómata

En el menú de **Resultados** podemos ver las opciones del diseño: **simular** el autómata, intentar **reducir** el autómata, obtener el autómata de **Mealy equivalente** y **diseñar** el autómata. Dejemos las tres primeras opciones para más adelante y activemos **Diseño**. En este caso BOOLE ofrece todas las tablas del diseño y las expresiones mínimas obtenidas

en una sola pantalla, la que aparece en la figura 13.35.

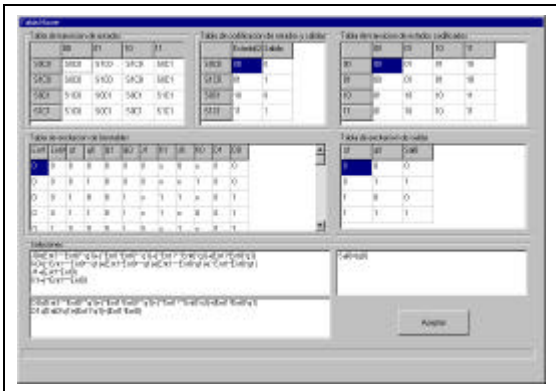


Figura 13.35 Tablas de diseño del autómata

De la anterior imagen cabe aclarar que en la Tabla de Excitación de Biestables la q minúscula representa al estado actual, mientras que Q mayúscula representa al estado futuro, es decir, $Q(t)$ y $Q(t+1)$ ó $Q(t-1)$ y $Q(t)$. La figura 13.36 muestra detallada la zona de las expresiones simplificadas.

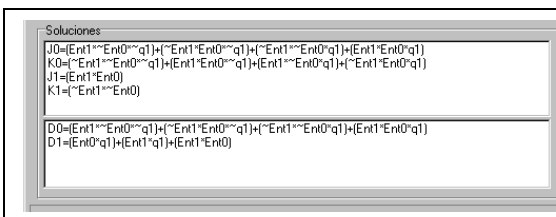


Figura 13.36 Expresiones simplificadas

Para ver el circuito lógico bastará con que el alumno vaya al menú **Ci rcui to** y elija el biestable deseado, por ejemplo **J- K**, obteniéndose la figura 13.37.

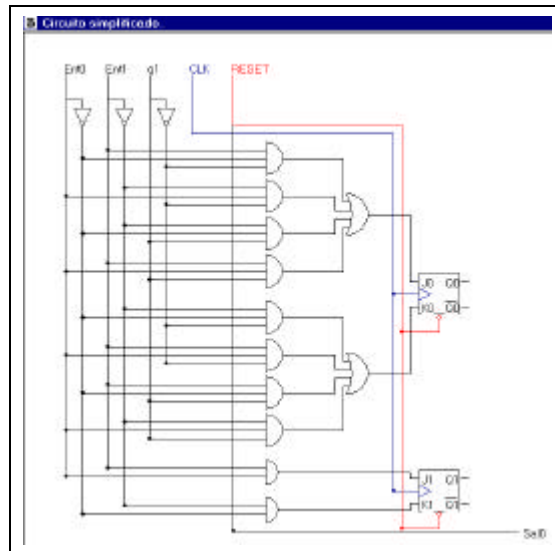


Figura 13.37 Circuito lógico del autómata

Siguiendo estos pasos el alumno habrá podido obtener no sólo el circuito lógico, sino que habrá observado los distintos pasos (y comprobado la bondad de los obtenidos por él). Además el alumno puede guardar el autómata, puede imprimir los resultados, puede copiar el DTE al portapapeles (y de ahí a un documento, con lo que BOOLE vale para documentar ejercicios) y puede obtener el correspondiente programa en OrCAD-PLD.

13.5.2 Ejemplo 2 de sistema secuencial

Planteemos ahora el autómata de Mealy capaz de detectar las secuencias 1010 ó 110 permitiendo solapamientos y cambios entre secuencia. Dispone de una entrada serie y de una salida que indica si se ha detectado una de las dos secuencias.

El DTE correspondiente al anterior enunciado es el de la figura 13.38.

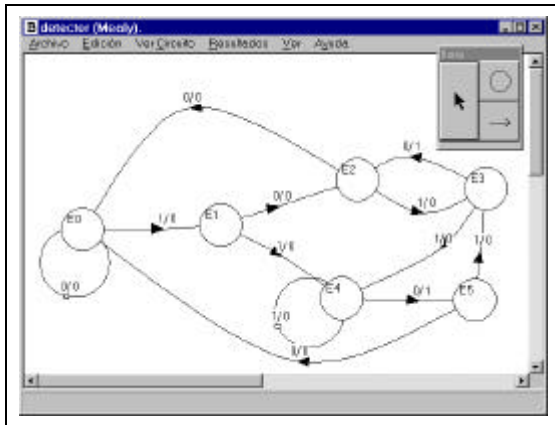


Figura 13.38 Autómata de Mealy del detector

El alumno puede intentar reducir el anterior autómata de 6 estados. Basta con seleccionar **Reducción** en el menú **Resultados**. BOOLE le muestra al alumno una pantalla en la que elegirá si quiere ver los pasos de la reducción (**Ver Pasos**) o sólo ver los resultados (**Simplificar**). Si el alumno optara por la primera opción, deberá guiar el proceso con sucesivos clics de ratón sobre **Continuar**, y así ir viendo cómo se completa la tabla de reducción. Tras varios clics BOOLE ofrece el resultado de la posible reducción, que en este caso deja al autómata en cuatro estados. Si en la figura 13.39, tras la reducción, se optara por **Cambiar**, el propio BOOLE sustituirá el DTE antiguo por el nuevo, quedando éste a la vista. Si hiciera falta el alumno podría reordenar o modificar lo dibujado por BOOLE.

Figura 13.39 Reducción de estados

Tras elegir **Cambiar**, el BOOLE muestra el DTE de la figura 13.40.

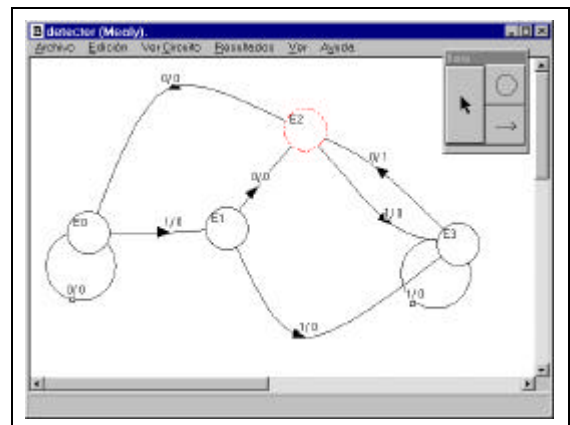


Figura 13.40 Autómata reducido

Sólo quedan por obtener las tablas y el circuito lógico correspondiente. En la figura 13.41 vemos que las tablas son distintas de las del anterior ejercicio, como corresponde a la diferencia entre Moore y Mealy. En la figura 13.42 se ve el circuito lógico basado en biestables tipo D.

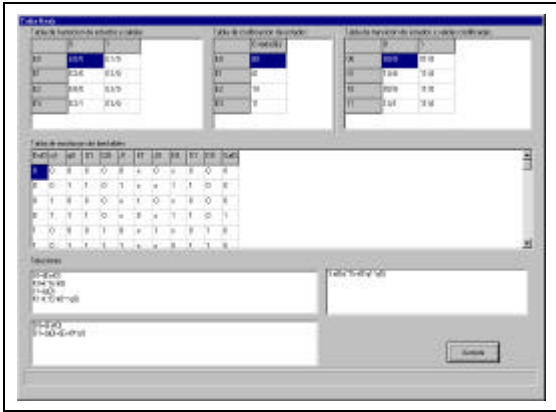


Figura 13.41 Tablas de diseño del autómata

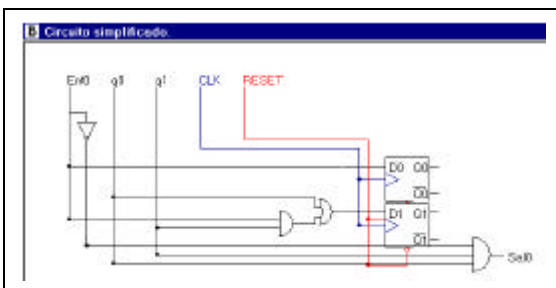


Figura 13.42 Circuito lógico del autómata

Por último, y al igual que en los combinacionales, BOOLE permite obtener el programa en OrCAD-PLD listo para ser compilado, simulado y grabado. La figura 13.43 muestra el programa correspondiente al detector de secuencias.



Figura 13.43 Programa en OrCAD-PLD

Con los dos ejemplos anteriores hemos visto la potencia, didáctica y facilidad de uso del BOOLE para autómatas.

13.5.3 Simulación de autómatas

El diseño de autómatas puede llegar a complicarse, siendo en estos casos muy útil una herramienta de simulación como la que presentamos.

Una vez que el alumno ha introducido el DTE que él cree correcto puede simularlo, y así comprobar su validez. En este caso el alumno deberá preparar un juego de ensayo de forma meticulosa, contemplando el mayor número de situaciones sin llegar a eternizarse. Seguidamente cargará dicho juego de ensayo en el simulador de BOOLE, y observará si los resultados obtenidos coinciden con los esperados. BOOLE no dice si el autómata está bien, es el alumno el responsable de la simulación, enfrentando qué quiere y qué tiene.

13.5.3.1 Simulación batch

Si el alumno dentro del menú **Resultados** eligiera **Simulación Batch** pasaría a ver la figura 13.44. En ella el alumno indicará cuál va a ser el estado inicial y cuántos casos se van a simular (este número puede cambiar). En cada fila el alumno escribirá el valor de las entradas, del reloj y del reset. Por ejemplo, en la primera fila activaremos el reset, en la segunda activaremos el clk con los valores 0 y 1 en las entradas, y así sucesivamente. Destacan en la simulación las filas 5, 6 y 7, pues en ellas cambian las entradas, pero no hay clk.

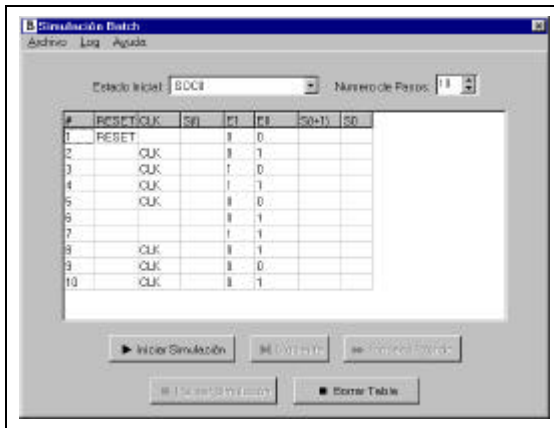


Figura 13.44 Simulación batch

Ahora al pulsar **Inicio Simulación** comienza la misma, y por cada **Si gui ente BOOLE** simula una nueva fila. Completada la simulación se obtiene la figura 13.45, que confirma la validez del autómata introducido.



Figura 13.45 Resultado de la simulación batch

Hay que ser cuidadoso a la hora de simular y leer la simulación, teniendo en cuenta que:

- ♦ Al llegar a una fila (en azul), ésta muestra su estado actual, su salida actual y cuál va a ser su próximo estado (cuando se haga clic y haya clk).

- ♦ Una fila no es procesada hasta no pulsar **Si gui ente**, es decir, la salida que se ve en cada fila es la de la fila anterior.
- ♦ No debe de verse la salida escrita en una fila como el resultado de procesar su entrada, sino como el resultado de procesar la anterior entrada.
- ♦ Esto puede parecer incómodo, pero resulta más real y potente a la hora de plantear simulaciones donde la entrada cambia entre flancos
- ♦ Los cambios en las entradas entre flancos son vistos de distinta forma por Moore y Mealy (Mealy los ve, Moore no).

Para ver mejor lo dicho ahora, el alumno puede tener a la vista simultáneamente la pantalla de simulación y la de captura del DTE. En esta última puede ver gráficamente la simulación, por ejmplo, la figura 13.46 muestra lo dicho para esa segunda fila, viéndose cuál es el estado actual, cuál es la transición que ahora está activa (preparada para cuando se haga clic) y cuál será el próximo estado. Cuando el alumno/profesor practique con la simulación entenderá las ventajas de ésta, aunque resulte algo compleja a primera vista.

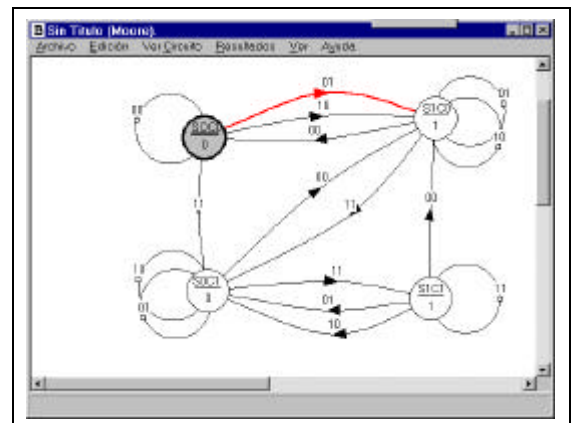


Figura 13.46 Simulación batch gráfica

Si observamos con calma las filas 5, 6 y 7 vemos que sólo es procesada la fila 5 (cuyo resultado aparece en la fila 6).

Además de lo visto, BOOLE permite:

- ◆ Guardar y cargar una simulación. En el menú **Archi vo**.
- ◆ Mostrar la simulación en un fichero de texto para imprimirlo o copiarlo en un documento del tipo que se quiera (muy cómodo para documentar ejercicios). En el menú **Log**.

13.5.3.2 Simulación interactiva

BOOLE permite simular de modo interactivo. En este caso el proceso es más flexible para el alumno; él no tiene por qué preparar antes la secuencia de entrada, sino que la puede ir improvisando sobre la marcha. La figura 13.47 muestra lo que el alumno ve al elegir **Simulación Interactiva** en el menú **Resultados**.

En la pantalla se ven varios elementos que hay que manejar con cuidado. Por ejemplo, lo primero que hará el alumno será pulsar **Reset**, luego escribirá una entrada y pulsará **Aplicar**, con lo que la entrada pasa al DTE (lo que se ve en la parte gráfica de la pantalla) y finalmente hará clic sobre **clk** para que el autómata pase al nuevo estado (si así se desea).

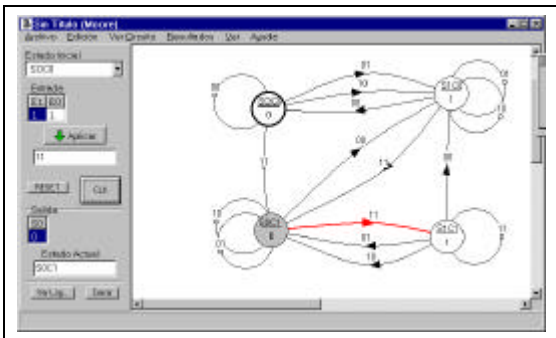


Figura 13.47 Simulación interactiva

El desarrollo de la simulación es interactiva, pero es guardada por BOOLE en un fichero **.log**, de modo que el alumno puede

recuperar y revisar lo simulado bien para estudiarlo, imprimirlo o guardarlo (activando **Ver Log**). Al ver el archivo de texto **.log** sorprende el hecho de que algunas entradas tengan una fila y otras dos. El último caso coincide con la activación del **clk**, así si nos fijamos en las filas 6 de la figura 13.48 resulta que:

- ◆ la primera muestra en qué estado estaba el autómata al recibir la entrada (antes del **clk**),
- ◆ mientras que la segunda fila muestra en qué estado queda el autómata (y su nueva salida) al procesar dicha entrada.

Es decir, y siendo coherente con lo ya dicho, el resultado de un **clk** se ve en la fila siguiente. Esta duplicidad permite un análisis más detallado, teniendo en cuenta que si estorbaran siempre podrán ser borrados en el **.log**.

Paso	RESET	CLK	S(t-)	S(t)	E1	S(t+1)	S0
1				S0C0	0	S0C0	0
2				S0C0	0	S0C0	0
3				S0C0	0	S0C0	0
4				S0C0	0	S1C0	0
5		1		S0C1	1	S0C1	0
6		1		S0C1	1	S1C1	0
7		1		S1C1	1	S1C1	0
8		1		S1C1	0	S0C1	0
9		1		S0C1	0	S0C1	0
9		1		S0C1	1	S0C1	0

Figura 13.48 Fichero **.log** de la simulación

13.5.3.3 Comparación simulación interactiva vs batch

Simular un autómata exige ordenar las entradas respecto del reloj. La pregunta es ¿cuándo cambia la entrada? ¿antes del flanco, después o en el mismo instante?

Muchas veces al simular queremos que la entrada cambie con el flanco, lo que en principio no es real. Lo normal es primero cambiar la entrada y luego generar el flanco,

de esta forma en el momento del flanco se procesa la entrada ya estabilizada.

Vista la situación anterior desde **BOOLE** hay que tener en cuenta que:

- ♦ en la simulación interactiva primero se cambia la entrada y luego se genera el **clk (Aplicar y click)**, lo que asegura que el ritmo de la simulación es controlado por el usuario,
- ♦ en la simulación batch se puede forzar que el cambio de la entrada coincida con el flanco, ahora bien el usuario debe recordar que el efecto de dicha entrada no será visto hasta pulsar **Si guiente**.

Por ejemplo, si queremos simular la secuencia de entrada de la tabla 13.2.

RESET	CLK	A	B	SUMA
RESET		0	0	0
	↑	0	1	1
	↑	1	0	1
	↑	1	1	0
	↑	0	0	1
		0	1	1 (*)
		1	1	1 (*)
	↑	0	1	1
	↑	0	0	0
	↑	0	1	1

(*) no procesadas

Tabla 13.2 Secuencia a simular

Simulada la anterior secuencia mediante el simulador batch se obtiene la tabla 13.3.

Paso	RESET	CLK	S(t)	E0	E1	S(t+1)	S0
1	RESET		SOC0	0	0	SOC0	0
2			SOC0	0	0	SOC0	0
2		CLK	SOC0	0	1	S1C0	0
3		CLK	S1C0	1	0	S1C0	1
4		CLK	S1C0	1	1	S0C1	1
5		CLK	S0C1	0	0	S1C0	0
6			S1C0	0	1	S1C0	1
7			S1C0	1	1	S0C1	1
8		CLK	S1C0	0	1	S1C0	1
9		CLK	S1C0	0	0	SOC0	1
10		CLK	SOC0	0	1	S1C0	0

Tabla 13.3 Resultado de la simulación batch

Se ve que el resultado es el mismo pero retardado un flanco, hubiera hecho falta una

fila más para procesar la entrada 0 1 del último clk.

La simulación interactiva de dicha secuencia es más larga, pues primero debemos cambiar la entrada y luego procesarla, que es lo más *lógico*. El resultado obtenido es la tabla 13.4.

Paso	RESET	CLK	S(t)	E0	E1	S(t+1)	S0
1	RESET		SOC0	0	0	SOC0	0
1			SOC0	0	0	SOC0	0
2			SOC0	0	0	SOC0	0
3		CLK	SOC0	0	0	SOC0	0
3			SOC0	0	0	SOC0	0
4			SOC0	0	1	S1C0	0
5		CLK	S1C0	0	1	S1C0	1
5			S1C0	0	1	S1C0	1
6			S1C0	1	0	S1C0	1
7		CLK	S1C0	1	0	S1C0	1
7			S1C0	1	0	S1C0	1
8			S1C0	1	1	S0C1	1
9		CLK	S0C1	1	1	S0C1	0
9			S0C1	1	1	S1C1	0
10			S0C1	0	0	S1C0	0
11		CLK	S1C0	0	0	S1C0	1
11			S1C0	0	0	SOC0	1
12			S1C0	0	1	S1C0	1
13			S1C0	1	1	S0C1	1
14			S1C0	0	1	S1C0	1
15		CLK	S1C0	0	1	S1C0	1
15			S1C0	0	1	S1C0	1
16			S1C0	0	0	SOC0	1
17		CLK	SOC0	0	0	SOC0	0
17			SOC0	0	0	SOC0	0
18			SOC0	0	1	S1C0	0
19		CLK	S1C0	0	1	S1C0	1
19			S1C0	0	1	S1C0	1

Tabla 13.4 Resultado de la simulación interactiva

Las salidas de la secuencia están marcadas en **negrita** y coinciden con la fila siguiente al **clk**. Dicha fila por cierto está numerada igual que la anterior. Por ejemplo, miremos a las dos filas número 3: la primera indica qué entrada hay en el momento del flanco y la segunda establece la salida tras ese flanco.

Lo dicho puede parecer complicar la simulación, y además hubiera sido más sencillo diseñar una simulación como la de la tabla 13.2. Si hemos optado por esta simulación es para ceder todo el control de la simulación al alumno/profesor, para que ambos puedan simular con libertad y no dirigidos, y poder analizar, por ejemplo, la diferencia entre Moore y Mealy.

13.6 Comentarios

El entorno BOOLE ha sido diseñado y programado durante cinco años por sus autores, Jesús Sanz Martínez y Javier García Zubía, y por alumnos de la Facultad de Informática de la Universidad de Deusto (ESIDE), destacando entre estos últimos Borja Sotomayor. Se han utilizado la tecnología orientada objetos, el lenguaje de programación C++ y la librería STL.

El resultado final tiene en este momento más de 20.000 líneas de código (excluyendo comentarios y líneas en blanco), lo que da una idea de la complejidad del entorno. BOOLE ha sido presentado en cuatro congresos nacionales e internacionales, y ha recibido el Premio al Mejor Equipo Software en el IV Congreso de Tecnologías Aplicadas a la Enseñanza de la Electrónica, TAEE 2000, celebrado en Barcelona en septiembre del año 2000.

El entorno ha sido probado insistentemente por sus autores, sus alumnos (700 alumnos al año de ingeniería informática e industrial), otros profesores (Alfonso Barba y José Antonio Aranguren) y por otros colaboradores externos (gracias especialmente a Mariano Barrón y Javier Martínez), pero esto no es óbice para que no aparezcan nuevos errores y comentarios.

Agradeceríamos que todos aquellos que encuentren un error o tengan cualquier comentario al entorno (positivo o negativo) los enviaran a la dirección: zubia@eside.deusto.es, lo que nos permitirá mejorar BOOLE.

Pensamos que una vez creado el entorno es responsabilidad de todos los que usemos BOOLE el mantenerlo en condiciones, para que todo este esfuerzo no haya sido baldío.