CCEDAF: Codificador Combinatorio de Estados para el Diseño de Autómatas Finitos

Juan Fº. Sanjuan Estrada, I. García Fernández, J.A. Alvarez Bermejo
Dpto. de Arquitectura de Computadores y Electrónica
Universidad de Almería
04120 Almería
e-mail: jsanjuan@ual.es, {inma, jaberme}@ace.ual.es

Resumen

En esta demostración se presenta el funcionamiento de la herramienta software CCEDAF [1], orientado a alumnos de Laboratorio de Estructura y Tecnología de Computadores, para facilitarles el diseño de autómatas finitos con un menor número de puertas lógicas.

La aplicación CCEDAF (Codificador Combinatorio de Estados para el Diseño de Autómatas Finitos) a partir de la tabla de transiciones del sistema digital, asigna todas las posibles combinaciones de codificación de estados para generar diferentes tablas de excitación, y obtener aquellas combinaciones que requieran un menor número de puertas lógicas.

1. Motivación

Generalmente, los alumnos de primer curso de Ingeniería Técnica de Informática consumen mucho tiempo en los cálculos de diseño de circuitos secuenciales sincronos, también conocidos como autómatas finitos, lo que repercute en disponer de menos tiempo para la simulación y/o implementación del sistema durante las prácticas de laboratorio.

La principal finalidad de las prácticas de laboratorio es que el alumno se familiarice por un lado con las herramientas software, disponibles en el mercado, destinadas al diseño y simulación de sistemas digitales, y por otro lado, el manejo de chips comerciales para la implementación de diferentes sistemas digitales.

Se deja para las clases teóricas y de ejercicios que el alumno aplique el método de diseño de sistemas secuenciales para la resolución de problemas.

Entre las ventajas de un circuito digital con un número reducido de puertas, destacamos la facilidad de simulación, la posibilidad de implementación, e incluso la rapidez en la detección de errores tanto por el alumno como por el profesor.

La asignación de estados es un grave problema computacional que surge, junto con el problema de minimización de estados, en el diseño de sistemas secuenciales [1]. Este problema combinatorio aumenta al incrementarse el número de estados requeridos por el sistema digital. Sin embargo, los diseños que normalmente realizan los alumnos en el laboratorio, suelen tener un máximo de ocho estados, siempre que minimicen correctamente la tabla de transiciones, lo que obliga a analizar un máximo de 40.320 combinaciones diferentes [3]. El tiempo total de computación suele rondar de 4 a 10 minutos, reduciéndose significativamente (un par de minutos) si se aplican las adyacencias entre estados [4].

2. Herramienta CCEDAF

CCEDAF es una aplicación diseñada en lenguaje C++, que permite ejecutarse tanto en sistemas operativos Linux, como en Windows. Para la ejecución de la aplicación se debe introducir en la línea de comandos, el nombre de la aplicación seguido de las siguientes opciones de entrada:

ccedaf fichero.txt tipo_biestable num_resultados

donde *fichero.txt* corresponde al nombre del fichero de texto de entrada encargado de describir la tabla de transiciones de estados minimizada. Con relación al *tipo_biestable* se puede elegir entre los cuatro tipos de biestables disponibles: D, T, RS, o JK. Finalmente, con el parámetro *num_resultados* se especifica el número total de los mejores resultados, es decir, aquellas combinaciones que ofrezcan el menor número de puertas posibles.

2.1. Fichero de entrada

En el fichero de entrada, el alumno debe de especificar la tabla de transiciones del sistema secuencial sincrono que desea diseñar. La tabla de transiciones descrita debe contener un número de estados minimizados, para lo cual, el alumno se puede servir de otras aplicaciones, tales como BOOLEDEUSTO [5]. La minimización de la tabla de transiciones es un aspecto que no se ha tratado en esta primera versión de CCEDAF, sin embargo, no se descarta incluirla en futuras versiones.

Dentro de la descripción de la tabla de transiciones minimizada se debe especificar una serie de parámetros, siguiendo el siguiente orden secuencial: número de entradas (*#Entradas*), número de estados (*#Estados*), número de salidas (*#Salidas*), y número de elementos (*#Elementos*) totales que se incluyen en la tabla de transiciones. Posteriormente, un bloque con todos los elementos de la tabla de transiciones, donde cada uno de los elementos se debe incluir con el siguiente formato:

$$[E_0E_1...E_n\ Q_n\ Q_{n+1}\ S_0S_1...S_m]$$

donde $E_0E_1...E_n$ representa el valor de las variables de entrada (0 ó 1) para un determinado elemento de la tabla de transiciones, Q_n y Q_{n+1} corresponden al actual y próximo estado, respectivamente, del elemento de la tabla, donde cada estado del sistema secuencial síncrono se identifica con una letra del alfabeto (A, B, C, ... Z). Finalmente, la secuencia $S_0S_1...S_m$ representa el valor de las variables de salida (0 ó 1) del sistema digital. A continuación, se muestra como ejemplo la transición del actual estado F al próximo estado G cuando por la entrada (de 2 bits) se introduce 00, obteniéndose a la salida (de 3 bits) el código binario 010, por lo que el elemento de la tabla de transiciones será: *[00 F G 010]*. Los comentarios dentro del fichero de texto deberán ir precedidos por los caracteres *//*.

2.2. Funcionamiento

El alumno puede continuar con el diseño del autómata ejecutando la aplicación CCEDAF para distintos tipos de biestables, escribiendo los siguientes comandos:

ccedaf ejemplo.txt D 10
ccedaf ejemplo.txt T 10
ccedaf ejemplo.txt RS 10
ccedaf ejemplo.txt JK 10

De esta forma, se ejecutan cuatro procesos CCEDAF simultáneamente sobre la misma tabla de transiciones, uno por cada tipo de biestable.

La ejecución de CCEDAF se traduce en la lectura de la tabla de transiciones del fichero de texto y estudio de las adyacencias entre estados, aplicando las reglas de adyacencia, descritas en [2]. Posteriormente, genera códigos binarios que asigna a los estados del sistema digital cumpliendo las reglas de adyacencia anteriores. Finalmente, minimiza tanto las ecuaciones de excitación de los biestables elegidos, como las ecuaciones de las salidas del sistema digital, contabilizando el número de puertas necesarias para la implementación.

El procedimiento anterior, se repite para nuevos códigos de asignación, hasta que se analizan todas los posibles combinaciones, almacenando únicamente aquellas combinaciones que generen un menor número de puertas.

2.3. Fichero de salida

Al finalizar CCEDAF su ejecución, genera un fichero de texto de salida con la siguiente información: las mejores codificaciones de estados con menor número de puertas, sistemas de ecuaciones con las expresiones booleanas, tanto de las entradas a los biestables como las salidas del sistema, el número mínimo de puertas AND, OR y NOT

necesarias para implementar cada posible solución, y finalmente el tiempo total de ejecución requerido.

El tiempo de ejecución de CCEDAF depende considerablemente del número de estados a codificar, de tal forma que cuando excede los 8 estados, el tiempo de computación se dispara exponencialmente.

3. Ejemplo

Realmente, la mejor forma de comprender el funcionamiento de CCEDAF es a través de un ejemplo. A continuación, se muestra el enunciado del sistema secuencial que se desea implementar:

Diseñar un autómata que sea capaz de detectar tres o más unos, o ceros, consecutivos que se introducen secuencialmente por una entrada.

La confección del diagrama de estados (figura 1) a partir del enunciado del problema, es crucial, pues su correcta elaboración (imposible de automatizar) depende directamente de los conocimientos, intuición, y experiencia del alumno en el diseño de sistemas secuenciales.

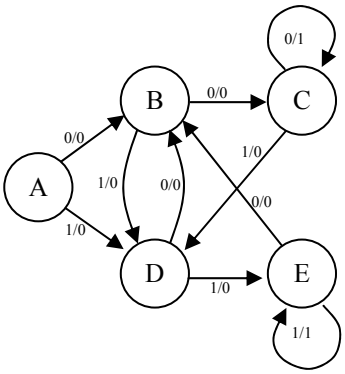

Figura 1. Diagrama de transición de estados

3.1. Tabla de transiciones

A partir del diagrama de estados es relativamente sencillo confeccionar la tabla de transiciones del sistema digital, y tratar de minimizar el número de estados obtenidos, lo que permitirá reducir el número de biestables del circuito final. Una vez obtenida la tabla de transiciones minimizada se escribe el siguiente fichero de entrada:

```
#Entradas = 1; // nº de entradas
#Estados = 5; // nº de estados
#Salidas = 1; // nº de salidas
#Elementos = 10; // nº elementos
// Tabla de transiciones
[0 A B 0]
[0 B C 0]
[0 C C 1]
[0 D B 0]
[0 E B 0]
[1 A D 0]
[1 B D 0]
[1 C D 0]
[1 D E 0]
[1 E E 1]
```

Algoritmo 1. Fichero de entrada (ejemplo.txt)

Donde se puede observar tanto el número de entradas, número de estados, y número de salidas del sistema digital a diseñar, como los elementos de la tabla de transiciones.

3.2. Ejecución

Durante la ejecución se analizan 40.320 combinaciones para los biestables D, T, RS y JK, cuyo tiempo de computación oscila entre cuatro minutos para un tipo D ó T, y diez minutos para un tipo RS ó JK, para un Pentium IV.

El tiempo de computación para sistemas con menos de 8 estados es razonablemente pequeño, incluso aunque no exista adyacencia de estados. Sin embargo, cuando el número de estados aumenta, es muy importante utilizar correctamente la adyacencia entre estados, para reducir el tiempo de computación.

3.3. Resultados

Haciendo una asignación de estados directa: A = 000, B = 001, C = 010, D = 011, y E = 100, (generalmente, es la asignación más sencilla por que no tiene en cuenta ningún tipo de adyacencia entre estados) obtenemos la siguiente simplificación con 10 puertas AND, 4 puertas OR, y 4 puertas NOT.

```
Entradas de los biestables JK:
   J2=x0x2x3
   K2=x0'
   J1=x3+x0x1'
   K1=x3
   J0=x0x1'+x0'x2'
   K0=x0'x2'+x0x2
Salida del circuito:
   S0=x0'x2x3'+x0x1
```

Algoritmo 2. Solución directa con biestables JK.

La ejecución de CCEDAF proporciona diez circuitos con menor número de puertas para cada tipo de biestable (tabla 1), destacando, los biestables tipo D con menor número de puertas.

Tipo	AND	OR	NOT
D	4	2	1
T	8	4	4
	9	3	4
RS	5	2	3
	6	2	2
	6	3	1
JK	5	2	3
	6	2	2
	6	3	1

Tabla 1. Mínimo número de puertas

Entre las distintas posibilidades que nos ofrece, con el mismo número de puertas, el alumno puede seleccionar la combinación que desee. En nuestro caso hemos seleccionado la siguiente, sin ningún motivo especial:

```
Biestable Tipo D
   Combinación: 987
   Asignación de estados: A=001,
B=010, C=011, D=100, E=101.
   Entradas de biestables:
      D0 = x0
      D1 = x0'
      D2 = x0'x2 + x0x1
   Salida del circuito:
      S = x0'x2x3 + x0x1x3
   Nº puertas: 4 AND, 2 OR, 1 NOT.
```

Algoritmo 3. Parte del fichero de salida

Un concepto interesante a destacar es que el número de puertas, que indica CCEDAF, necesarias para implementar el circuito, corresponde al número mínimo de puertas totales. En el ejemplo anterior (algoritmo 2) la salida del circuito utiliza las mismas puertas AND necesarias por la entrada D2 del tercer biestable.

4. Conclusión

Con esta aplicación se ha pretendido que el alumno disponga de una herramienta software que le permita diseñar rápidamente circuitos secuenciales con un menor número de puertas, característica que facilitará la simulación e implementación del autómata de estados finitos.

Esta primera versión de CCEDAF, se verá próximamente mejorada en dos aspectos. Por un lado, permitirá minimizar el número de estados de una tabla de transiciones, y por otro lado, se aumentará la velocidad de ejecución al poder repartir la carga computacional entre varios procesadores.

Referencias

[1] Sanjuán Estrada, J.M., García Fernández, I. y Álvarez Bermejo, J.A. *Automatización del problema de asignación de estados en el diseño de sistemas secuenciales sincronos*. JENUI 2004.

[2] Angulo Usategui, J.M. y García Zubía, J. *Sistemas digitales y tecnología de computadores.* Paraninfo, 2002.

[3] Lloris Ruíz, A., Prieto Espinosa, A., y Parrilla Roure, L. *Sistemas digitales*. McGraw-Hill, 2003.

[4] S. Devadas and R. Newton, A. *Exact algorithms for output encoding, state assignment, and four-level boolean minimization*. IEEE Transactios on computer-aided design. Vol. 10, No. 1. January, 1991.

[5] J. García Zubía, J. Sanz Martínez, y B. Sotomayor. *BOOLE-DEUSTO, la aplicación para sistemas digitales*. VII Jornadas de Enseñanza Universitaria de la Informática (JENUI). 2001.