

Propuesta de una metodología para una rápida enseñanza de circuitos lógicos y de su integración en una UCP en carreras de Informática

Mario Carlos Ginzburg
Profesor Titular de "Sistemas de Computación I y II"
en la Universidad Abierta Interamericana
mario.ginzburg@uai.edu.ar

Resumen. Los programas de Arquitectura de Computadoras y similares de carreras de Informática son extensos, y en general presentan una sola unidad dedicada a circuitos lógicos, debiéndose formar en limitadas clases a los alumnos en los conocimientos básicos. Esta problemática trae aparejado un debate acerca de los contenidos mínimos a enseñar acorde con la formación buscada, que en esta propuesta se plantean y desarrollan. Opcionalmente, conforme al perfil y nivel de cada carrera, los circuitos de la UCP tratados aisladamente pueden integrarse en un modelo de UCP didáctico, también desarrollable en pocas clases. Esta plataforma no requiere conocimientos de electrónica, y también puede emplearse en estudios terciarios de Informática y en carreras de Electrónica como introducción a los circuitos digitales. La presente metodología didáctica se ha concretado con excelentes resultados en ingenierías informáticas de la FIBA (3er. año) y Facultad de Tecnología de la UAI (1er. año) desde el 2002 al presente.

Palabras claves: circuitos lógicos, enseñanza didáctica rápida

1. Introducción

Cuando alumnos de carreras de Informática de distintas partes del país solicitan equivalencias de estudios, se observa al leer los programas de Arquitectura de Computadoras o asignatura similares, que en general se dispone de pocas clases para desarrollar desde “cero” los contenidos dedicados a los circuitos básicos que conforman la UCP (compuertas, decodificadores, UAL, multiplexores, flip flops, registros y buses). Estos circuitos en general se enseñan desconectados entre sí, sin conformar una UCP. En las carreras de Electrónica se tiene una asignatura como Técnicas Digitales dedicada por completo a los circuitos combinatoriales y secuenciales.

Por las limitaciones de tiempo señaladas se requiere que la enseñanza de esos circuitos sea sintética y conceptual, a la par que didáctica. Para lograr esto es necesario discutir primero qué contenidos son propios de Informática, como se plantea en el ítem 2 de este trabajo

Si además, por el perfil y exigencias del nivel de estudios de una carrera de Informática, dichos circuitos digitales deben integrarse en un modelo didáctico simple de UCP, su tiempo de enseñanza se vuelve doblemente crítico si se quiere desarrollar normalmente los restantes contenidos de Arquitectura de Computadoras.

Tal modelo de UCP construido con compuertas y flip flops que forman parte de sus bloques funcionales, debe permitir, sin usar electrónica, visualizar y comprender en profundidad en cada ciclo del pedido y ejecución de instrucciones, entre otras cosas, el papel temporal y circuital de los MHz y de las líneas de control sobre: los registros, la UAL, la memoria y los caminos de datos (“data paths”). Asimismo debe poder ilustrar acerca de cómo interactúan hardware y software, y cómo es que en la decodificación el código de operación de una instrucción determina el valor que tendrán las líneas de control para que el hardware la ejecute y pida la siguiente. Además debe ser factible en dicho modelo ejecutar instrucciones de salto en las que intervienen los flags, para comprender cómo la máquina decide entre dos alternativas, y poner en claro que la UCP no tiene inteligencia propia para concretar en cada ciclo sus acciones. Igualmente debe ser útil para establecer diferencias entre CISC y RISC.

A fin de poder desarrollar con rapidez este modelo, es indispensable que previamente los alumnos hayan sistematizado en un modelo sin circuitos lógicos, sólo con “cajas negras” de la UC, la UAL y los registros interconectadas por buses, los movimientos y cambios que ocurren en un computador cuando se piden y ejecutan las instrucciones. De esta forma se sigue el orden de lo general a lo particular, para luego volver a lo general.

2. Problemática abordada

Conforme al objetivo buscado de que en las carreras de Informática la enseñanza de los circuitos lógicos sea sintética, conceptual, didáctica, y sean mínimos sus tiempos de enseñanza, primero es indispensable definir qué aspectos de la enseñanza de circuitos son necesarios y suficientes en estas carreras, para luego desarrollar un proceso de enseñanza-aprendizaje apropiado y didáctico que cualquier docente puede aplicar.

La presente metodología es fruto de sucesivos perfeccionamientos didácticos practicados durante más de diez años de enseñanza de los contenidos aquí planteados. En tal sentido, el autor como director de cátedras, llevó a cabo observaciones empíricas siguiendo la evolución en el aprendizaje a través de cuestionarios, exámenes parciales y finales de centenares de alumnos de las asignaturas “Estructura del Computador”, de 3er. año de Ingeniería Informática de la FIBA (2000-2006), y Sistemas de Computación II de 1er. año la Facultad de Tecnología Informática de la UAI (del 2004 al presente). Debe consignarse que la mayoría de los alumnos provienen de establecimientos secundarios no técnicos, sin conocimientos de electrónica o electricidad.

A fin de no perder tiempo en dibujos por parte del docente o de los alumnos, esta metodología requiere proyectar imágenes, que pueden formar parte de un texto, o un apunte con un atlas de figuras, al cual los alumnos pueden remitirse para sistematizar y repasar todos los temas tratados en clase.

2.1 Enseñanza tradicional con aspectos relacionados con carreras de electrónica

Habitualmente, en parte por que los primeros profesores de materias del tipo Arquitectura de Computadoras en facultades de Informática fueron ingenieros electrónicos, el aprendizaje de los circuitos lógicos en general ha sido muy semejante al que tiene lugar en los primeros tramos de la asignatura Técnicas Digitales de las carreras de Ingeniería Electrónica. Un objetivo propio en estas carreras es que el futuro ingeniero esté capacitado para proyectar subsistemas electrónicos digitales.

Asimismo, aún hoy, con la influencia de Técnicas Digitales, además de métodos algebraicos de minimización y transformaciones circuitales, se enseñan detalladamente un conjunto de flip flops asincrónicos y sincrónicos, en sus distintos funcionamientos (R-S, J-K, T, etc.), siendo que, como se desarrolla en el ítem 3, es factible a partir de un multiplexor de 2 entradas construir de manera simple un flip flop “D” sincrónico, con el cual se pueden implementar los registros y secuenciadores que necesita un modelo simple de UCP.

Esta concepción en la enseñanza de los temas básicos también se apoya en la bibliografía clásica para Arquitectura de Computadoras, con textos que contemplan tanto el funcionamiento como el diseño de computadores, objetivo que en principio no está presente en el perfil del ingeniero en Informática o Sistemas.

En Stallings [1] se enseñan los temas antes mencionados, y además se agrega simplificación por el método de Quine-Mc Cluskey y contadores sincrónicos. En Tanenbaum [2] si bien aparece la enseñanza tradicional de este tema, no se dan métodos de simplificación. En Murdocca - Heuring [3], en Hamacher - Vranesic - Zaky [4], y en Alcalde - Portillo - García Merayo [5], por citar algunos, contienen bien detalladas metodologías apropiadas para carreras de ingeniería electrónica. Incluso en algunos textos aparecen transistores y hojas con datos eléctricos de compuertas.

Una limitación que presenta en general esta enseñanza, es que si bien los alumnos tienen un primer conocimiento del funcionamiento de circuitos lógicos tratados aisladamente, no ven su funcionalidad formando parte de una UCP.

2.2 Objetivos propios, planteo didáctico y herramientas para la enseñanza de circuitos lógicos en carreras de Informática

En la formación del ingeniero en Informática es secundario el proyecto de circuitos lógicos, ya sea con mínima cantidad de compuertas, o con compuertas de un solo tipo, o para integrarlos en un chip de un computador. Esto es propio de un ingeniero electrónico, que necesita una enseñanza intensa del álgebra de Boole y de la síntesis con chips de muy alta integración circuital, amén de conocimientos profundos sobre física electrónica, electricidad y electrónica.

En las carreras de informática interesa en primer lugar *el funcionamiento* de los circuitos citados en el ítem 1 que componen una UCP, y si se pretende un nivel de enseñanza superior, dichos circuitos pueden interconectarse a fin de conformar una UCP, sin apelar a la electrónica.

El Álgebra de Boole debe ser una herramienta formativa en Informática, en el sentido de que los alumnos en primer lugar puedan entrever isomorfismos entre las funciones Or, And y negación y estructuras lógicas del lenguaje Assembler y de lenguajes de alto nivel. Así, existen a nivel de máquina instrucciones en bajo nivel que ordenan operaciones And, Or, negación, X-Or usadas en programas; e inversamente, mediante una estructura IF puede describirse la tabla de funcionamiento de una compuerta.

Didácticamente la enseñanza se simplifica, es más comprensible y se reducen tiempos, si los circuitos se construyen sólo con compuertas And, Or e inversores, como ocurre en la metodología planteada.

Además es necesario conocer los símbolos de la lógica clásica para comprender formalmente las operaciones lógicas que realiza la UAL, o escribir en forma compacta mediante sumas de productos un comportamiento circuital And-Or, con el que puede expresarse verbalmente cualquier tabla de funcionamiento mediante las conectivas “o” e “y”. Resulta así un correlato conceptual y didáctico entre el álgebra de Boole y el álgebra de proposiciones.

Esto es la base para que los alumnos de Informática puedan construir un circuito And-Or suma de minterminos a partir de una tabla de funcionamiento que debe cumplir, desarrollando de esta forma en ellos aptitudes generales para

proyectar. Así, a partir de la tabla de un sumador completo, se sintetiza un circuito que cumpla con ella, para poder construir luego con 4 u 8 de éstos el sumador/restador de una UAL y los flags que ella genera.

Si bien una expresión booleana tiene la relevancia cognitiva y formativa de expresar formalmente con pocos símbolos una tabla o un comportamiento circuital, didácticamente es discutible si al inicio de la enseñanza de los circuitos lógicos se deben desarrollar forzosamente métodos algebraicos para simplificar usando Karnaugh, o para pasar de un circuito a otro equivalente mediante De Morgan. Estos desarrollos producen discontinuidades didácticas en el proceso de enseñanza-aprendizaje del funcionamiento de los circuitos lógicos. Esto se manifiesta en que al abordar el tema de la UAL, o siguientes, muchas veces los alumnos, por el tiempo transcurrido, no tienen presentes temas básicos vistos al comienzo, por lo que hay que volver a repasarlos.

Por otra parte, una enseñanza analítica de circuitos basada exclusivamente en expresiones booleanas puede tornarse demasiado abstracta o árida para muchos alumnos.

En este sentido, si se quiere verificar algebraicamente que una expresión booleana se corresponde con una tabla, el alumno debe reemplazar en forma abstracta valores binarios en esa expresión. Este procedimiento puede volverse algo mecánico y atemporal, alejando al alumno de una primera apreciación empírica y directa del funcionamiento de un circuito And-Or corriente.

En cambio, como se ejemplificará, una simple inspección visual inmediata de cada And con inversores que conforma dicho circuito And-Or, determinará directamente qué combinación binaria hace valer uno la salida de la And, y en consecuencia también la salida de la Or. De este modo se va construyendo conceptualmente la tabla buscada.

Si en lugar de una expresión booleana, se parte de una tabla de funcionamiento, siempre se tiene una visión totalizante del comportamiento que debe proveer cualquier implementación circuital que se deduzca de ella, y de hecho proporciona más información que un plano con compuertas que simboliza un circuito digital que cumple con esa tabla, o que la expresión algebraica que sintetiza su respuesta.

Consecuentemente, a los fines didácticos, en la metodología presentada se dará más preeminencia a las tablas que a sus expresiones algebraicas.

Por otra parte en la asignatura Matemática Discreta de la carrera de Informática es corriente enseñar el álgebra de Boole, incluidas las formas normales y la minimización, ocurriendo a veces superposiciones en la enseñanza de temas.

En relación con la enseñanza de flip flops, no es imprescindible partir de los asincrónicos, y es suficiente desarrollar el "D" sincrónico, con sus dos estados fácilmente recordables: uno cuando la salida copia la entrada (con $Ck=1$) y el otro cuando retiene (con $Ck=0$). Con este flip flop se construirán los registros de la UCP y las celdas de memoria.

De esta forma, si con cuidado se prescinde de contenidos propios de Ingeniería Electrónica, es factible reducir las horas de clases que ello implica, y se pueden desarrollar en un par de clases los circuitos de la UCP y memoria.

En el corto tiempo que insume este aprendizaje tipo "mecano", se van construyendo uno tras otro, circuitos combinacionales y secuenciales de complejidad progresiva, usando sólo compuertas And, Or e inversores. Ello redundará en que sin saltos temáticos los alumnos puedan integrar y sistematizar más fácilmente los aspectos comunes y diferentes del funcionamiento de los mismos.

A posteriori es factible integrar estos circuitos para conformar una UCP básica completa, que también puede desarrollarse en un reducido número de clases.

3. Esquema con aspectos básicos de la plataforma propuesta

En el inicio del proceso de enseñanza-aprendizaje presentado se procura que el alumno pueda establecer correspondencias y equivalencias conceptuales y visuales de significados simbólicos entre una tabla, un circuito And-Or que la verifica y una suma de productos miniterminos que lo expresa.

Al respecto, a partir de las tablas de las compuertas And y Or se verifica y pone de relieve que su denominación está relacionada con el hecho de que la relación entre entradas y salida se puede enunciar con proposiciones lógicas con las conectivas "o" e "y", a la par que se definen la simbología circuital de cada compuerta y su expresión algebraica, poniendo de relieve que son tres formas distintas de simbolizar lo mismo.

A continuación en cada una de las entradas de una And se conecta o no un inversor (conexión que abreviaremos Inv-And), se determina la tabla de este conjunto y se verifica visualmente en el conexionado que la salida de la And vale 1 para una sola combinación presente en las entradas del conjunto Inv-And. Didácticamente conviene poner de relieve por simple examen visual, que para un conjunto Inv-And existe una única combinación que puede hacer que todas las entradas de la And valgan 1 para que su salida valga 1. Como ilustra el Inv-And más alto de la figura 1 su salida vale 1 sólo si en sus entradas se tiene $A_i = 0$ "y" $B_i = 1$ "y" $C_{i-1} = 1$ que producirán 1 "y" 1 "y" 1 en las entradas de la And.

Los alumnos así internalizan la correspondencia: combinación binaria \iff Inv-And que la detecta unívocamente.

En lo que sigue, didácticamente usaremos en forma modular y visual conjuntos Inv-And para detectar combinaciones con dos usos complementarios: a) dado un Inv-And construido, determinar visualmente qué combinación da valor 1 a su

salida; y b) dada una combinación a detectar, construir por simple inspección visual un Inv-And que la identifique mediante el valor 1 de su salida.

Tabla 1

A _i	B _i	C _{i-1}	C _i	S _i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

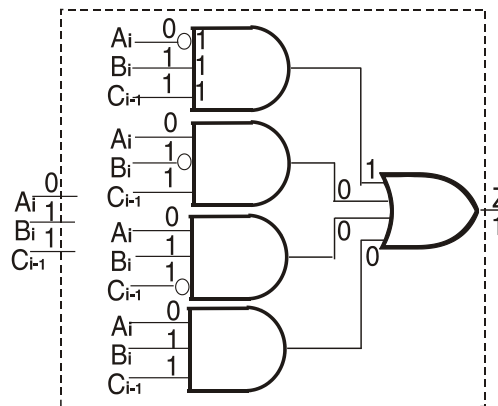


figura 1

Luego de las verificaciones con Inv-And, en las que fundamentalmente interviene la vista, ya pueden desarrollarse circuitos decodificadores de 2 y 3 entradas, basados en estos módulos Inv-And detectores de combinaciones, lo cual también sirve para que los alumnos afiancen conocimientos.

Para el decodificador de 3 entradas los alumnos deberán construir 8 módulos Inv-And para identificar del 000 al 111, y hacer el cableado necesario para que todos reciban juntos la combinación presente en las entradas del decodificador. Además verificarán que sólo tendrá valor 1 la salida del decodificador correspondiente al Inv-And construido para detectar la combinación presente en las entradas del decodificador.

En Informática es importante vincular el funcionamiento de un decodificador con el carácter random de cualquier memoria electrónica, planteando que en éstas el número binario de n bits que recibe el decodificador es la dirección de la celda a acceder, y que cada una de sus 2ⁿ salidas termina en una celda, para permitir el acceso a ella cuando es direccionada.

Seguidamente puede analizarse un circuito Inv-And-Or de 3 entradas, en el que varios Inv-And de igual número de entradas concurren a una Or como en la figura 1 con el objeto de hallar por inspección visual la tabla que el circuito verifica, supuesta desconocida.

A tal fin primero los alumnos deben reconocer que algunos de los Inv-And son los mismos que se han tratado en el decodificador. O sea que los 4 Inv-And del circuito dado conforman un decodificador que no detecta 8 sino sólo 4 combinaciones distintas (011, 101, 110, 111), determinables visualmente por la ubicación de los inversores en cada Inv-And.

La salida de la Or toma valor 1 si cualquier salida de un Inv-And vale 1. Ello ocurrirá si la salida del Inv-And superior detecta, como muestra la figura 1 (por la ubicación de su inversor) que en las entradas del circuito se ha recibido 011 (Ai=0 “y” Bi=1 “y” Ci-1=1); “o” si el Inv-And siguiente detecta que se ha recibido 101 (Ai=1 “y” Bi=0 “y” Ci-1=1); “o” si el anteúltimo Inv-And detecta que se ha recibido 110; “o” si el último Inv-And detecta que se ha recibido 111.

Así se obtienen las 4 combinaciones 011, 101, 110 y 111 a las que les corresponden los 4 “unos” de la tabla 1. Como cada una de las restantes combinaciones no puede generar salida 1 en ninguno de los 4 Inv-And, ellas producirán 4 ceros en las entradas de la Or, o sea salida cero en ella.

Con esta técnica, partiendo de un circuito Inv-And-Or, con igual número de entradas en cada And, con sólo examinarlo visualmente, se puede determinar la tabla que cumple.

A cualquier circuito Inv-And-Or como el analizado se le puede hacer corresponder un enunciado del álgebra proposicional como el anterior usando las conectivas “o” e “y”, lo cual permite al alumno tener una comprensión más clara de su funcionamiento lógico.

Este tipo de enunciados se pueden formalizar en el álgebra de Boole mediante sumas de productos minitérminos, que se corresponden circuitalmente a estructura Inv-And-Or con compuertas And de igual número de entradas.

Si se quiere también puede hallarse la suma de productos minitérminos propia del circuito con las correspondencias simbólicas “+” y “.” para las conectivas “o” e “y” respectivamente, y con las convenciones simbólicas que se aplican para escribir minitérminos. Así el funcionamiento anterior puede expresarse:

$$C_i = \bar{A}_i \cdot B_i \cdot C_{i-1} + A_i \cdot \bar{B}_i \cdot C_{i-1} + A_i \cdot B_i \cdot \bar{C}_{i-1} + A_i \cdot B_i \cdot C_{i-1}$$

No es obligatorio tratar este tema al comienzo de la enseñanza. Tampoco se plantea hallar la tabla buscada reemplazando en esa suma de minitérminos cada combinación, de 000 a 111 en este caso, siendo que ella se ha obtenido concretamente observando directa y simplemente las entradas de los Inv-And, para así determinar las combinaciones (tantas como Inv-And distintos existan) para las cuales la columna de salida de la tabla vale 1.

Habiendo los alumnos asimilado la metodología visual de análisis del comportamiento de un circuito, puede plantearse el procedimiento inverso: a partir de una tabla como la tabla 1 (que en este ejemplo debe cumplir la salida C_i de un semisumador), construir paso a paso el circuito Inv-And-Or (suma de minterminos) que la verifica, el cual aparece terminado en la figura 1.

El tipo de circuito buscado ya fue analizado por los alumnos, y al igual que cualquiera a sintetizar, será del tipo Inv-And-Or, de estructura y funcionamiento lógico que se corresponde con un enunciado que emplea las conectivas “o” e “y”.

Procediendo de forma inversa al análisis, a partir de la tabla se determina que en este caso son 4 la cantidad de combinaciones a detectar que deben generar el valor 1 en la salida de la Or final. Este número también establece que serán necesarios 4 módulos detectores Inv-And, cuyas salidas irán a una Or que así tendrá 4 entradas (figura 1). La ubicación de los inversores en cada uno de los 4 Inv-And sigue la misma técnica usada en la construcción del decodificador, o sea depende de la posición de los ceros de cada combinación a detectar que debe hacer valer 1 a la salida.

Así se llega al circuito de la figura 1, y luego con 4 semisumadores se puede construir un sumador/restador de 4 bits, que forma parte de una UAL.

Empleando esta metodología de síntesis, a partir de la tabla de funcionamiento de que se trate, los alumnos están capacitados para construir el correspondiente circuito combinacional Inv-And-Or.

Con la presente propuesta, desde el inicio de la enseñanza de circuitos lógicos es factible ahorrar muchas horas de clase sin desviar la atención del alumno en temas como métodos algebraicos de simplificación de funciones, transformación de expresiones booleanas en Nand-Nand, u otros.

La enseñanza de flip flops parte de un circuito selector (multiplexor) de 2 entradas (figura 2) cuya entrada de selección S oficiará de reloj (Ck) en el flip flop sincrónico “D” que resulta simplemente de realimentar la salida Q en la entrada inferior del selector (figura 3).

Los alumnos comprenden en primer lugar que de este modo el cable de realimentación obliga a que el valor que tiene la salida Q del selector sea siempre el mismo que presenta la entrada inferior, o sea que el propio circuito da valor a una de sus entradas, por lo que así ésta no recibe valor 0/1 del exterior.

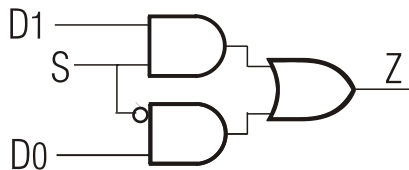


Figura 2

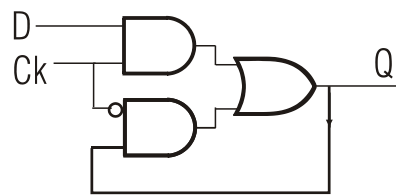


figura 3

Los alumnos comprenden en primer lugar que de este modo el cable de realimentación obliga a que el valor que tiene la salida Q del selector sea siempre el mismo que presenta la entrada inferior, o sea que el propio circuito da valor a una de sus entradas, por lo que así ella no recibe valor 0/1 del exterior.

A continuación resulta claro, con los valores indicados (figura 4), que mientras sea $Ck = 1$ se selecciona la entrada D vinculada al exterior, por lo que la salida Q “copiará” el valor 0/1 que D recibe: si D mantiene su valor así lo hará Q , pero si D cambia también lo hará Q , tantas veces como lo haga D , sin posibilidad de mantener su valor.

Esta situación (como ilustra el dibujo superior de la figura 4) se simboliza con fines didácticos y mnemónicos con la caja del flip flop en la cual un cable virtual une la salida Q con la entrada D , siendo que en un cable conductor un extremo tiene el mismo valor (voltaje) que el otro.

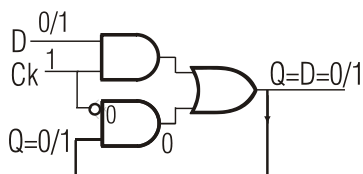
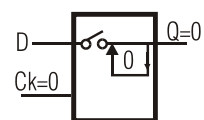
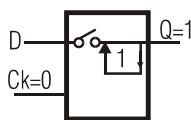
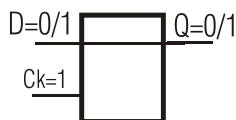


figura 4

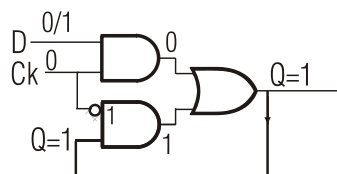


figura 5

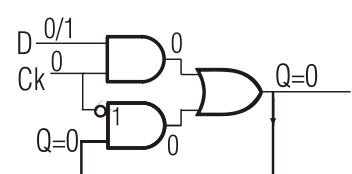


figura 6

Si luego se hace $Ck = 0$ (figuras 5 y 6) se selecciona la entrada inferior, que continuamente tiene el valor 0/1 de la salida, por estar conectadas ambas por el cable que las une. Entonces, el valor que tenga Q en el instante en que es $Ck = 0$ (que es

el mismo que tenía la entrada D en ese momento), será el de la entrada inferior. Conforme al funcionamiento del selector este valor debe aparecer en la salida Q, por lo que el valor que tenía Q en el instante en que $C_k = 0$ no podrá modificarse mientras sea $C_k = 0$, dado que en la entrada inferior permanentemente está ese mismo valor de Q (figuras 5 y 6).

Así el valor de Q se autorepite, se automantiene, quedando retenido, memorizado, aunque el valor de la entrada D cambie al valor contrario, dado que por no estar D seleccionada, su valor no puede cambiar la salida Q.

Como ilustran los dibujos superiores de las figuras 5 y 6 se simboliza mnemóticamente en cada caso esta situación de retención de un cero o uno mientras sea $C_k = 0$, con la salida Q autoinyectándose el valor que tiene, y la entrada D aislada de la salida, sin poder intervenir en el valor de Q.

Para cambiar el valor que mantiene la salida Q, se debe hacer $C_k = 1$, con lo cual se volverá a seleccionar la entrada D para que su valor pase a la salida, al mismo tiempo que se deja de seleccionar la otra entrada, para que la salida no se copie a sí misma. Si bien el valor realimentado de la salida Q está siempre presente en la entrada inferior debido al cable que las une, sólo se repite en Q cuando esta entrada está seleccionada por ser $C_k = 0$.

Con este tipo de flip flop sincrónico resulta más sencillo de entender a los alumnos cómo un circuito puede retener un bit. Asimismo es el único que necesitan conocer en la presente plataforma de enseñanza. Su funcionamiento se fija y sistematiza mediante los diagramas temporales correspondientes.

No es imprescindible desarrollar los flip flops asincrónicos como el R-S, ni partir de éste para enseñar flip flops.

Mediante flip flops “D” sincrónicos puede conformarse un registro, como se indica en la figura 7.

Conforme a la experiencia docente acumulada al respecto, puede estimarse que los temas hasta acá planteados pueden desarrollarse en 2 clases de 5 hs. cátedra.

4. Aplicación del método propuesto a la enseñanza de una porción de la UCP

El ejemplo siguiente apunta por un lado a poner de manifiesto las ventajas didácticas en la enseñanza que resultan de visualizar los flip flops (M-E) que conforman los registros mediante los mnemónicos de sus 2 estados, con $C_k = 1$ y $C_k = 0$ antes simbolizados. Por otro lado permite apreciar cómo los alumnos pueden comenzar a incursionar progresivamente en un modelo de UCP, para ver de qué forma concreta los MHz generados por un cristal actúan en el hardware configurando ciclos, y cómo las líneas de control de la UC determinan el ciclo en que un registro debe cambiar.

Si bien el ejemplo de la figura 7 ilustra una porción limitada de una UCP, con la presente metodología didáctica puede construirse una UCP básica completa, que contenga: la UC con líneas de control cuyos valores ella genera en cada ciclo, la UAL, los caminos de datos (“data paths”), los registros básicos, el decodificador y una matriz de conexionado. Un modelo pedagógico de esta UCP que data del año 2007 se desarrolla en un cuadernillo del autor [6] destinado a los alumnos, que debe actualizarse en función de nuevos planteos didácticos planteados en este trabajo.

El registro típico de la figura 7 está conformado por flip flops sincrónicos D “maestro-esclavo” cuyos instantes de cambio están sincronizados por los pulsos (designados C_k) recibidos por todos los flip flops simultáneamente, siendo que estos pulsos llegan sin invertir a las entradas C_k de todos los esclavos (E), e invertidos a las entradas C_k de los maestros (M).

En el conjunto dibujado el registro es representativo del Puntero de Instrucciones de la UCP, al cual en uno de los ciclos de los MHz al valor (0110) que contienen sus E se le quiere sumar 0010, para que en ese ciclo los E pasen a guardar la dirección (1000) de la próxima instrucción a localizar en memoria.

La estructura y comportamiento de este registro son iguales a los de cualquier otro registro de la UCP vinculado a los caminos de datos (“data paths”). En el inicio del ciclo que corresponda se debe poder aportar una copia del contenido del registro involucrado a donde ordene la instrucción en curso, y si es necesario, en ese mismo ciclo se debe poder reemplazar dicho contenido por otro nuevo, cuando el cristal genera el pulso con que termina ese ciclo. Esto sólo es factible si los flip flops “D” sincrónicos son “maestro-esclavo” (M-E), o sea compuestos por dos de estos flip flops simples.

Como aparece en la figura 7, en un ciclo determinado una compuerta And genera un pulso reloj que permitirá, en su flanco ascendente, que los M retengan el nuevo contenido (1000), que será copiado por los E, que lo guardarán en el otro flanco. Para ello una entrada de dicha And recibe continuamente los pulsos de los MHz que genera un cristal, y la otra está conectada a una línea de control (LC) de la UC, cuya circuitería hace que esta LC tenga valor 1 durante todo este ciclo en que supuestamente debe cambiar el presente registro. Por lo tanto durante ese ciclo la salida de la And generará una réplica de la forma de onda que produce el cristal de los MHz, la cual así llegará a los flip flops del registro como señal C_k . Durante los ciclos en que la LC tenga valor 0, la salida de la And valdrá 0, por lo que también será $C_k = 0$ en los E, que así estarán en estado de retención, permitiendo que el registro almacene.

Más en detalle, en el inicio del ciclo en que $LC = 1$ y hasta el flanco ascendente del pulso de ese ciclo, como el cristal de los MHz genera 0 volts, la salida de la And valdrá 0, lo cual determina que en los E sea $C_k=0$, y en los M sea $C_k=1$. Entonces, conforme al funcionamiento de estos flip flops, en ese lapso los E seguirán reteniendo el contenido (0110) que tenían, como en sus cajas simboliza cada salida reinyectando su valor, y estos cables transmitirán 0110 al sumador, que le suma 0010 resultando 1000 en sus salidas. Éstas van a las entradas D de los maestros, que por recibir $C_k=1$ se comportan como cables que copian 1000 a sus salidas Q que llegan a las entradas D de los esclavos.

Esta situación persiste hasta que la salida de la And valga 1 por ser $LC=1$ y por generar valor 1 el cristal de los MHz. Entonces, en los M será $/Ck=0$, por lo que pasarán a retener 1000 que era el último valor que tenían sus salidas Q en ese instante, como se simboliza ahora con sus salidas reinyectando su valor. Dado que los E reciben $Ck=1$, se comportarán como cables cuyas salidas Q copian el valor 1000 que reciben de las salidas de los M que están reteniendo. Por lo tanto, las salidas de los E, que son las salidas del registro, enviarán al sumador el valor 1000 que si bien no está retenido en los E, permanece invariable mientras los E retengan. El sumador adicionará 0010 y generará 1010, pero este nuevo valor no pasará a las salidas del registro, pues los M están reteniendo 1000, por lo que no permiten que 1010 pase a los E. Al inicio del siguiente ciclo y en los sucesivos se supone que $LC=0$, por lo que la salida de la And será 0 y nuevamente en los E será $Ck=0$ y $/Ck=1$ en los M, repitiéndose la situación esquematizada en la figura 7 superior. Por lo tanto en dicho inicio los E pasarán a retener 1000, que era el último valor que tenían sus salidas Q en ese instante del inicio del siguiente ciclo. A su vez en los E será $/Ck=1$, por lo que sus salidas copiarán nuevamente el nuevo valor 1010 de las salidas del sumador, que ahora no puede pasar a los E dado que están en estado de retención hasta que no sea otra vez $LC=1$.

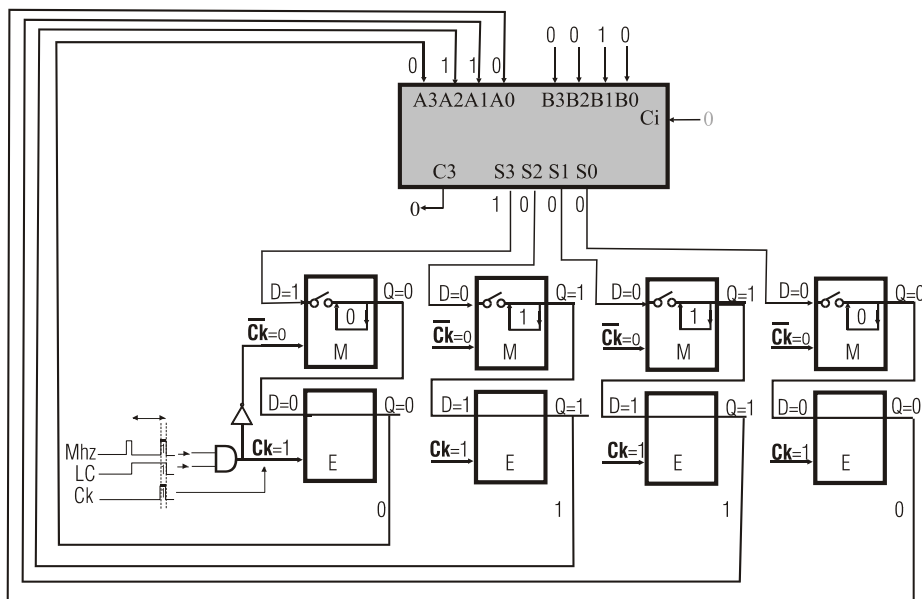
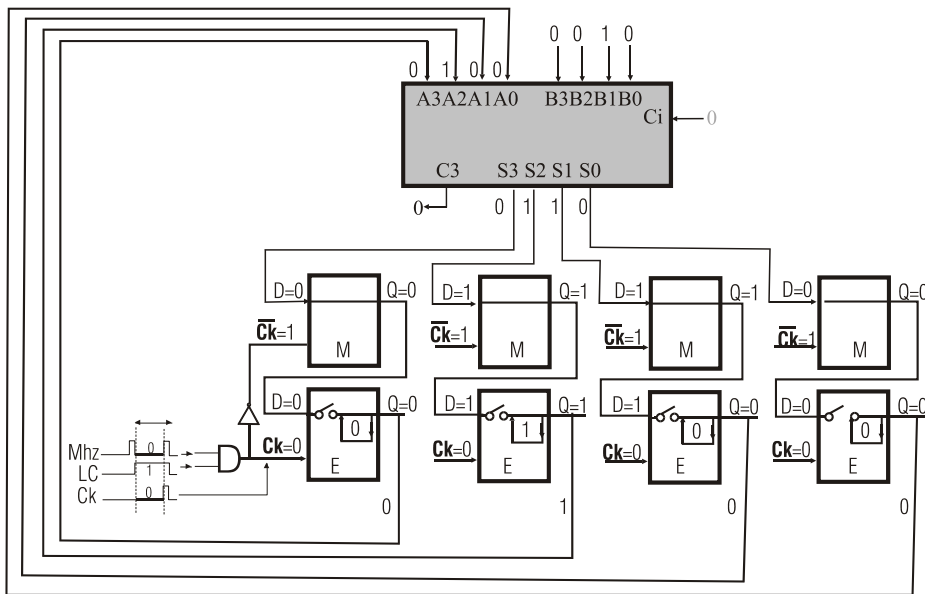


figura 7

De esta forma los alumnos, con la ayuda de los dos esquemas mnemónicos de los dos estados del flip flop, pueden visualizar más claramente este proceso que ocurre en un solo ciclo en el cual el registro primero aportó 0110 al sumador, y luego pasó a almacenar el nuevo valor 1000 que el sumador generó en un primer tramo del ciclo. Si bien en lo que resta del ciclo los E transmiten 1000 al sumador, el nuevo valor 1010 que pasa a generar el sumador no puede llegar a los E pues en ese lapso los M retienen 1000. Este ejemplo también sirve para que los alumnos entiendan la necesidad de que cada flip flop sea M-E. De no existir los maestros que al retener 1000 no “dejan pasar” el 1010, el sumador volvería a sumar 0010 n veces mientras sea $C_k = 1$.

5. Implementación de la metodología propuesta y del modelo

Como se indicó, el presente desarrollo didáctico ha sido puesto en práctica con excelentes resultados y perfeccionado continuamente en las asignaturas “Estructura del Computador“, de 3er. año de Ingeniería Informática de la FIBA (2000-2006), y Sistemas de Computación II de 1er. año la Facultad de Tecnología Informática de la UAI (de 2004 al presente). También ha sido implementado sin problemas por los profesores adjuntos de Sistemas de Computación II, Ing. Enrique Douce e Ing. Ricardo Martín, del 2010 al presente.

Alumnos provenientes de escuelas comerciales manifestaron que nunca pensaron que podrían comprender tan en profundidad el funcionamiento de un procesador. También ha permitido avanzar más rápidamente en el dictado de otros temas como control microprogramado, RISC y CISC y el lenguaje Assembler. Asimismo los alumnos en los exámenes finales mostraron una comprensión más cabal del tema.

6. Conclusiones

Es factible en 2 clases de 5 hs. cátedra desarrollar con la metodología propuesta la enseñanza del funcionamiento de los circuitos lógicos básicos, y en 3 clases adicionales construir con ellos un modelo didáctico simple de UCP de características RISC.

Para los circuitos se parte de la capacidad de establecer correspondencias visuales sencillas e intuitivas entre combinaciones binarias y compuertas And con inversores en sus entradas, de modo que la salida de cada And valga 1 sólo para la combinación que se desea detectar con ella.

En forma didácticamente gradual, cada nuevo nivel alcanzado engloba los anteriores y permite pasar al siguiente, hasta llegar a los circuitos concretos básicos de una UCP.

El modelo simple de UCP permite que el alumno deje de tener una comprensión abstracta acerca de muchos detalles importantes que interesan, especialmente en relación con la temporización de las señales internas de la CPU durante la ejecución de las instrucciones, dado que el modelo le proporciona una forma concreta de visualizar en el espacio y tiempo el interior de una UCP. Como todos los modelos de este tipo ayuda entre otras cosas: a entender por qué una máquina no piensa; cómo una instrucción se ejecuta en pasos que ya están predeterminados por el hardware (construido por el hombre) en función del código de operación; a comprender el papel de los pulsos reloj y el de las señales de control en el manejo de los caminos de datos (“data path”); a visualizar los movimientos internos que tienen que ocurrir en dichos pasos; a constatar cómo los circuitos combinatoriales transforman el código de operación en señales de control. De este modo, el alumno se sentirá capacitado para abordar, analizar y comparar modelos con pipe line y modelos de CPU reales.

7 Bibliografía

1. Stallings, W.: Computer Organization and Architecture. 9/E, Prentice Hall (2013).
2. Tanenbaum, A.: Structural Computer Organization, 6/E, Prentice Hall (2012)
3. Murdocca M., Huring V.; Principios de Arquitectura de Computadoras. Prentice Hall –Pearson Education (2002)
4. Alcalde E., Portillo J., Garcia Merayo F.: Arquitectura de Ordenadores, McGraw-Hill (1999)
5. Hamacher V, Vranesic Z, Zaky S. Computer Organization, McGraw-Hill (1996)
6. Ginzburg M.: De la Compuerta al Computador (2007), Biblioteca Técnica Superior (2007)

