

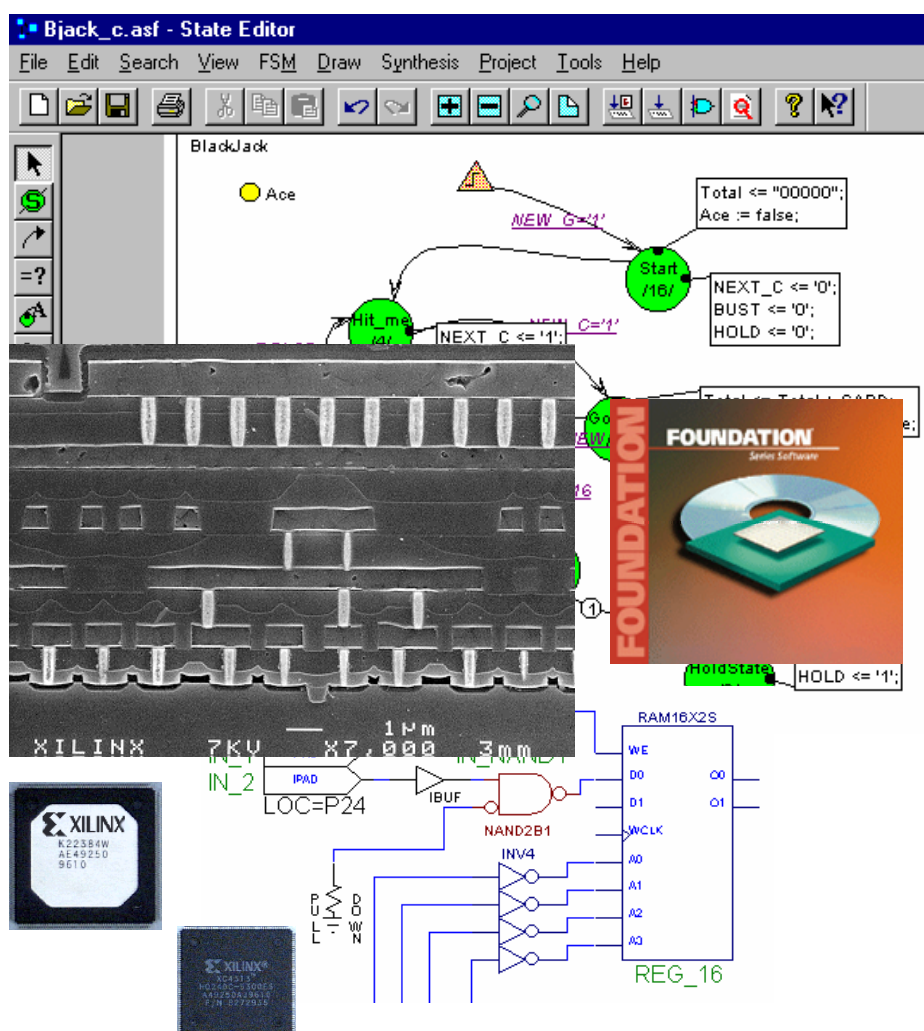


Universidad
de Huelva

Escuela Politécnica Superior
Universidad de Huelva

SISTEMAS SECUENCIALES SÍNCRONOS

TERCER CURSO. ELECTRÓNICA DIGITAL



Manuel Sánchez Raya
Versión 1.1
16 de Octubre de 2000

ÍNDICE

1.- Introducción.....	1
1.1.- Clasificación de los sistemas secuenciales.....	2
1.2.- Tipos fundamentales de autómatas finitos.....	2
2.- Elementos básicos de memoria: Biestables.....	3
2.1.- Flip-Flop R-S.....	4
2.2.- Biestable D.....	5
2.3.- Flip-Flop J-K.....	6
2.4.- Flip-Flop T.....	6
Características Dinámicas de los biestables.....	7
3.- Análisis y síntesis de sistemas síncronos basados en biestables.....	7
3.1.- Análisis.....	7
3.2.- Fases de la síntesis de circuitos síncronos.....	8
3.3.- Reducción de Tablas de Estado.....	8
3.3.1.- Método de la carta de implicación para tablas completamente especificadas.....	9
3.3.2.- Método de la carta de implicación tablas incompletamente especificadas.....	10
3.2.3.- Asignación de Estados.....	14
3.4.- Problemas estructurales en los circuitos secuenciales.....	17
4.- Sistemas secuenciales síncronos integrados de aplicación general.....	19
4.1.- Registros.....	19
4.2.- Contadores.....	21
4.2.1.- Contadores Asíncronos.....	21
4.2.2.- Contadores síncronos.....	22
4.2.3.- Asociación de contadores.....	24
5.- Síntesis de sistemas secuenciales microprogramables.....	25
5.1.- Completos.....	25
5.2.- Incompletos.....	26
6.- Sistemas secuenciales programables monochip (PLD's).....	26
6.1.- Arquitectura AND programable OR programable. (PLS).....	26
6.2.- Arquitectura AND programable OR fija. (PAL).....	30
6.2.1.- Matrices lógicas AND programables (PAL's) con capacidad secuencial.....	30
6.2.2.- Dispositivos lógicos programables y borrables (EPLD's) basados en macroceldas.....	31
6.3.- Fases del diseño asistido por ordenador de sistemas secuenciales basados en PLD's.....	34
7.- Dispositivos programables avanzados CPLD.....	38
7.1.- Dispositivo EEPL de AMD.....	40
7.2.- Dispositivo EPLD de ALTERA.....	41
7.3.- Dispositivo EEPLD de LATTICE.....	44
7.4.- Dispositivos CPLD de XILINX.....	46
8.- Dispositivos FPGA de XILINX.....	47

SISTEMAS SECUENCIALES SÍNCRONOS

1.- Introducción.

Recibe el nombre genérico de sistema de conmutación al sistema lógico que genera unas señales de salida en función de las señales de entrada. Esta dependencia entre señales da lugar a dos grandes tipos de sistemas de conmutación:

1. Combinacionales
2. Secuenciales

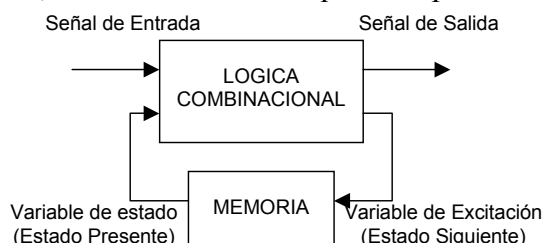
Un sistema combinacional es aquél cuyas salidas son función exclusiva del valor de las entradas. Es decir, cuando el valor de las señales de entrada varía a lo largo del tiempo, las salidas también lo hacen de acuerdo con dichas variaciones.

Un ejemplo de sistema combinacional es el “sumador completo de dos bits”. El valor de la suma y del acarreo final es función, únicamente, de los valores de los operandos (a y b) y del acarreo inicial. Un sistema combinacional puede quedar definido completamente mediante una tabla de verdad. De ésta se deducen directamente las ecuaciones booleanas a las que corresponden las salidas. Una vez simplificadas las ecuaciones, la implementación se reduce a interconectar los módulos lógicos adecuados.

En cambio, en los sistemas secuenciales, el valor de las salidas no sólo es función del valor de las entradas, sino también de la “historia” o “secuencia” previa por la que han atravesado dichas entradas.

Por ejemplo, supóngase una máquina cuyo funcionamiento se controla por medio de un pulsador X de marcha/parada. Si la máquina está detenida y se pulsa X, se pone en marcha. Una vez puesta en marcha, aunque se deje de pulsar X, la máquina continúa funcionando. Para detenerla basta volver a pulsar X; luego, una vez parada, aunque se deje de pulsar X la máquina permanece parada. En un momento dado la máquina tiene que almacenar información de su estado (parada o arrancada). El sistema que realice el control de la máquina deberá considerar el **estado anterior** para deducir el **estado presente**.

El controlador de la máquina solo tiene dos estados posibles; PARO y MARCHA; por lo tanto, con una sola línea que transporte información binaria se puede codificar su estado. La memoria informa sobre el estado presente mediante la línea “variable de estado”. La lógica combinacional obtiene la salida en función de ésta y la señal de entrada, proporcionando además, la variable de excitación a la memoria que informa del estado siguiente. La memoria almacena los estados por los que pasa el sistema.



En un circuito secuencial se cumple lo siguiente:

- El valor de las variables de salida en un instante dependen del valor de las entradas y de la secuencia de estados por los que ha pasado el sistema anteriormente, los cuales quedan definidos por el estado que tuvo en el instante anterior que guarda la memoria.
- Para una misma combinación de los valores de las variables de entrada, pueden corresponder diferentes valores de las variables de salida.

1.1.- Clasificación de los sistemas secuenciales.

Sistemas secuenciales asíncronos.

En este grupo, la progresión de los estados internos es automática y no requiere señales externas que la controlen. El valor de las señales de salida sigue los cambios de las señales de entrada, hasta que el sistema alcanza un estado estable para una determinada combinación de las entradas. Se llama estado estable, aquél que se alcanza con una combinación concreta de las señales de entrada y que tiene la propiedad que el siguiente estado coincide con él.

Sistemas secuenciales síncronos.

La evolución de los estados internos es gobernada por los impulsos de un generador externo. Los impulsos producidos por un generador de reloj carecen de información y sólo sirven para indicar el momento en el que se realiza el cambio del estado. Deben cumplir dos condiciones:

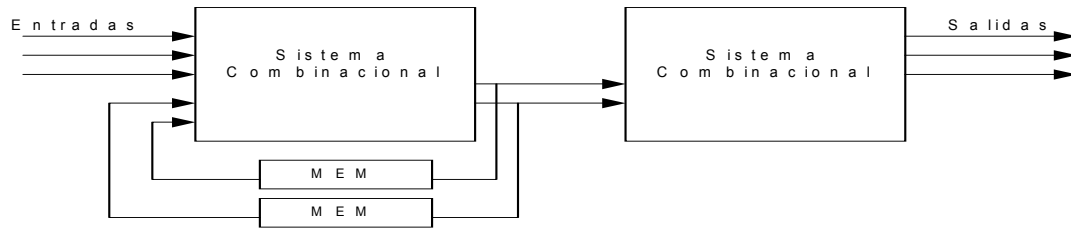
- Su frecuencia debe permitir que entre cada dos impulsos se establezca todo el sistema.
- Cada impulso de reloj solo puede originar un cambio de estado.

$$\text{Sistemas secuenciales} \left\{ \begin{array}{l} \text{Asíncronos} \left\{ \begin{array}{l} - \text{Realimentación directa.} \\ - \text{Células asíncronas activas por nivel.} \end{array} \right. \\ \text{Síncronos} \left\{ \begin{array}{l} - \text{Células síncronas activas por nivel.} \\ - \text{Células síncronas activas por flanco.} \end{array} \right. \end{array} \right.$$

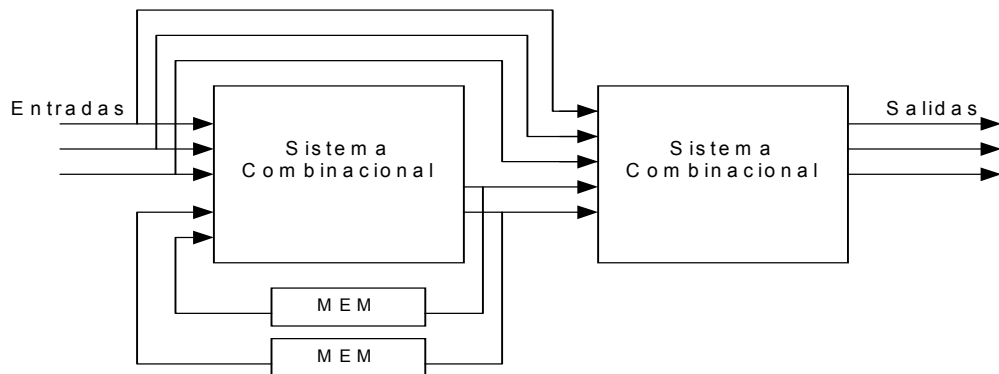
1.2.- Tipos fundamentales de autómatas finitos.

Los autómatas son máquinas secuenciales que funcionan de acuerdo con una sucesión de estados internos, en número finito, que codifican la historia de las señales de entrada. A los autómatas se les aplica una sucesión de señales de entrada distribuida en el tiempo, produciendo una señal de salida función de la última combinación de las entradas y de su estado interno. Los autómatas se clasifican en dos tipos básicos: *autómatas de Moore* y *autómatas de Mealy*.

Un **autómata de Moore** es aquél cuyas salidas toman un valor constante, durante el transcurso de un estado interno completo. Es decir, las salidas tienen el mismo valor en cada estado del autómata, puesto que solo dependen de él.



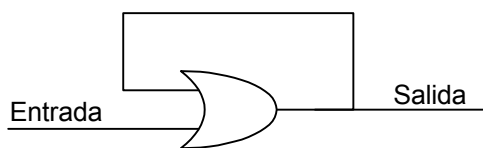
En un **autómata de Mealy**, una salida no está definida durante la duración de todo un estado interno, sino sólo en una parte del mismo, en la que se cumplen las condiciones exigidas a las señales de entrada.



En general, cualquier sistema secuencial puede definirse mediante un modelo de Moore o de Mealy, y hay normas concretas para traducir un tipo de autómata a otro. Normalmente el modelo de Moore emplea más estados internos y más memoria (flip-flop) que el de Mealy.

2.- Elementos básicos de memoria: Biestables.

Las mismas puertas AND y OR pueden interconectarse para formar elementos de memoria y tienen la capacidad de recordar si a sus entradas se les ha aplicado o no un nivel 1 con anterioridad. Por ejemplo, con una simple puerta OR podemos hacer una memoria de capacidad muy limitada.

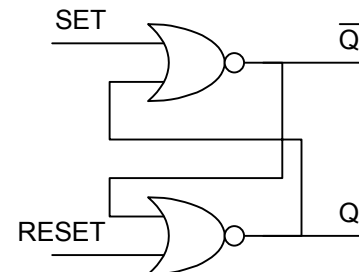
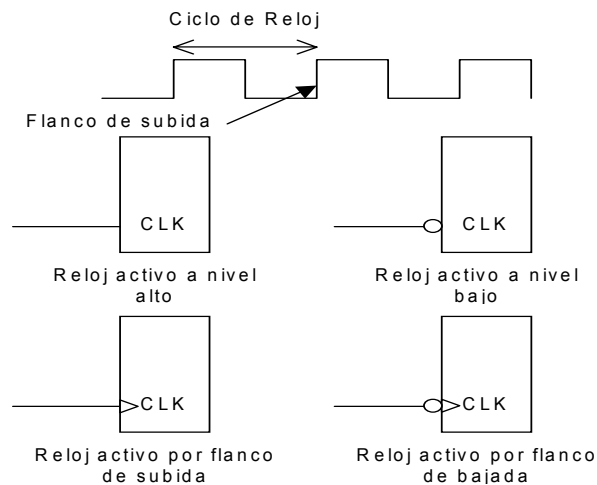


Inicialmente, la salida Q y la entrada están a nivel lógico 0. Si la entrada pasa a nivel 1, la salida pasará a 1 y aunque posteriormente A vuelva a 0, la salida permanecerá en 1. La única forma de borrar la memoria para ponerla en su estado inicial será desconectar la salida Q de la entrada B y al quedar ambas entradas a 0, la salida pasará a 0.

Los elementos básicos de memoria son los biestables denominados “basculas” (latches: cerrojos), a partir de los cuales y mediante modificaciones se consiguen elementos de memoria más complejos. Según la lógica que se utilice y el tipo de disparo, los biestables fundamentales pueden clasificarse en 4 clases:

- R – S (Reset – Set)
- J – K (Maestro – Esclavo)
- D (Data)
- T (Toggle)

El disparo de estos biestables puede ser de dos tipos, bien como consecuencia de alcanzar un determinado nivel de tensión, bien por la aplicación de un determinado tipo de flanco en un impulso de tensión.



2.1.- Flip-Flop R-S

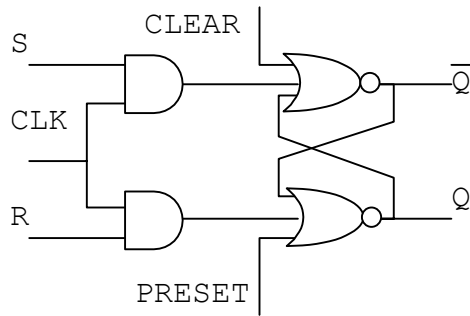
El circuito de la figura de la derecha se le denomina flip-flop R-S (Reset-Set) o báscula R-S y es el más básico de todos los flip-flop. Las salidas del circuito están conectadas a las entradas, las señales aplicadas a la entrada lo atraviesan y retornan a las entradas, de forma que la señal de entrada produce múltiples efectos. A la conexión de la salida con las entradas se le llama “feedback” o realimentación, y es esencial en el comportamiento de los circuitos lógicos con memoria.

Si introducimos un 1 lógico por la entrada SET, mientras RESET está a 0, la salida \overline{Q} pasará al estado lógico 0 y como esta salida está conectada con la otra puerta NOR, la salida Q quedará a 1. El nivel 1 de Q realimenta la entrada de la puerta superior, por lo que el nivel 1 de la entrada SET puede pasar a 0 permaneciendo las salidas en el mismo estado. Para reiniciar el circuito, se pone la entrada RESET a 1, con lo que Q pasa a nivel 0 y \overline{Q} a 1. En este estado se dice que la bascula se ha puesto a cero y borrada su información. En la báscula R-S, si la entrada S tiene un valor 1, la salida $Q=1$, y si la R toma valor 1, la salida $Q=0$, estando las salidas Q siempre inversas a las de \overline{Q} .

Q_t	S	R	Q_{t+1}
0	0	0	0
0	0	0	0
0	1	0	1
0	1	1	Indeterminado
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	Indeterminado

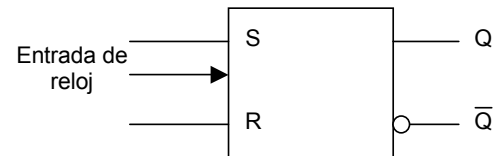
Cuando ambas entradas R y S estén altas simultáneamente, ambas salidas quedarán a nivel 0. Este estado especial se debe evitar, puesto que crea indeterminación en las salidas. Cuando desaparece la condición de indeterminación ($R=S=1$) no se conoce algebraicamente el estado que aparecerá en el flip-flop. La tabla de verdad de este flip-flop, representando Q_{t+1} , el estado siguiente de la salida al momento que se considera será:

Hay tres señales básicas, comunes a la mayoría de los flip-flop. Estas señales son las de “reloj” (clock), “puesta a 1” (preset) y borrado o “puesta a 0” (clear). Para ello ampliamos nuestra báscula con más circuitería.



Dos puertas AND conectadas a las entradas de una b scula R-S conectan una se al de reloj, de forma que se puede abrir (enable) o cerrar (disable) el paso de las se ales R y S, es necesario que la se al de reloj est  a nivel 1 para permitir el paso de las se ales R y S. La se al de reloj sirve para bloquear y desbloquear los datos de los flip-flop adem s de permitir sincronizar los datos de entrada al biestable.

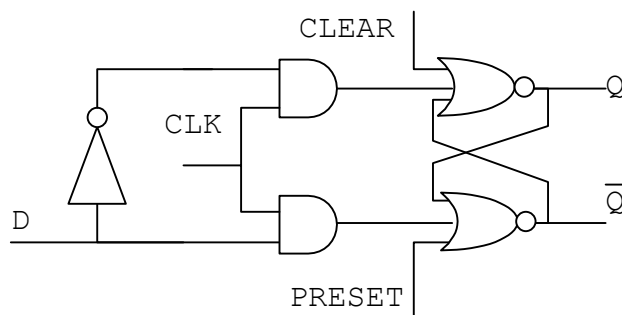
Las se ales de preset y clear (puesta a 1 y a 0, respectivamente) son entradas que se emplean para poner a 1 o a 0 el flip-flop, sin involucrar para nada las se ales de reloj o las entradas. Es decir, se emplean para poner a 1 o a cero el biestable cuando la se al de reloj est  baja o alta, por eso se dice que se pone a 1 o a 0 el flip-flop as ncronamente (en desincron a con la se al de reloj).



Cuando por primera vez se le aplica alimentaci n al flip-flop no se sabe si quedar  en estado 1 o 0. Las se ales de preset y clear, se usan en este caso para iniciar cada flip-flop con un estado conocido.

2.2.- Biestable D.

En esta b scula solo existe una entrada marcada como D y, debido al uso de un inversor, las entradas R y S siempre ser n opuestas y nunca se podr  producir en ellas la condici n de indeterminaci n.



Se aplica a la entrada de datos D la se al alta o baja que se desee memorizar y, antes de que cambie, se pone a cero la se al de reloj. Tan pronto como pasa a cero la se al de reloj, aunque la l nea de entrada de datos cambie de estado, la salida no se ve afectada y se mantiene. Esta b scula transmite a la salida el estado de la  nica entrada que posee, siempre que

se active la se al de reloj, por lo que se denomina "latch" o cerrojo. Esto se denomina entrada de reloj activa por nivel.

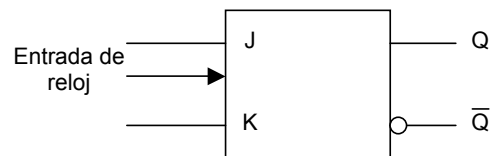
Tambi n podemos disponer de circuitos activos por flanco en lugar de activos por nivel. El estado en el circuito activo por flanco aparece solo en el instante en que la se al de reloj pasa del nivel l gico bajo al alto, y en ning n otro caso, es decir, en el flanco ascendente del impulso de reloj. Este es el funcionamiento caracter stico de la mayor parte de los flip-flop y su carga se realiza "por flanco de disparo".

Estos circuitos activos por flanco, constan de dos b sculas interconectadas, una de entrada y otra de salida y a menudo se denominan configuraci n maestro-esclavo. La subida de la se al de reloj ocasiona la entrada a la b scula de salida de la informaci n presente en las entradas de la b scula de entrada.

2.3.- Flip-Flop J-K

El flip-flop J-K es el más usado en general y también el más versátil y sofisticado. Al igual que la báscula R-S, tiene dos entradas de datos. No puede tener una salida indeterminada. La mayoría de las versiones del J-K están controladas por el flanco posterior o descendente de la señal de reloj, en lugar de serlo por el flanco ascendente, que controla al tipo D.

En principio, si una de sus entradas tiene nivel lógico 1 y la otra 0, será puesto a 1 o a 0 por el flanco de la señal de reloj igual que la báscula R-S. Si ambas entradas están a 0 cuando llega el impulso de reloj, recordará simplemente el mismo estado que tenía antes del flanco de dicho impulso. Si sus dos entradas están a nivel 1, el flip-flop cambiará su estado con el impulso de reloj. Esta forma de trabajo se llama de “balanceo” o “toggle”.

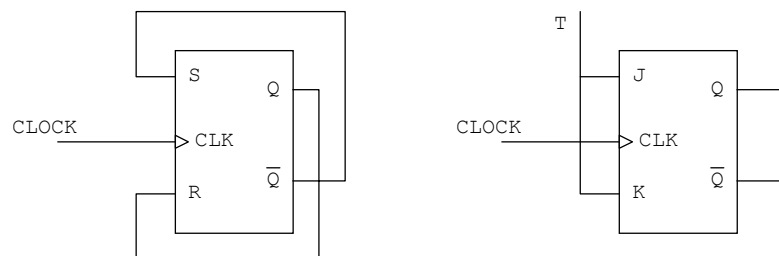


La báscula J-K dispone de dos entradas denominadas J y K, así como una entrada de señal de reloj para sincronización. La tabla de verdad es:

2.4.- Flip-Flop T.

El flip-flop T (toggle) se puede considerar una versión simplificada del JK. Es parecido porque bascula con todos los impulsos de reloj, bien en el flanco ascendente o en el descendente, según los detalles específicos del circuito. Se emplea para circuitos como contadores, divisores, etc. También se puede construir uniendo las patillas J y K de un biestable JK.

Q_t	J	K	Q_{t+1}
0	0	0	0 (Mantiene estado)
0	0	1	0
0	1	0	1
0	1	1	1 (cambia estado)
1	0	0	1 (Mantiene estado)
1	0	1	0
1	1	0	1
1	1	1	0 (cambia estado)



Todos los flip-flop están disponibles en CI, para su empleo no es necesario saber más que el conexionado externo de los pines y la tabla de la verdad o tipo de flip-flop que contiene, no es preciso conocer el circuito en sí.

Las entradas de preset y clear se denominan generalmente S y R respectivamente y su estado se invierte dentro del circuito del flip-flop, por lo que para activar estas entradas debe ser aplicado un nivel lógico 0. Las entradas de preset y clear nunca deben ser activadas al mismo tiempo.

En la familia lógica TTL cualquier entrada que queda sin conectar se considera como “flotante” y responderá como si tuviese un nivel lógico 1. Sin embargo, no se debe dejar ninguna patilla como flotante, pues esta captaría ruido y el podría dar una respuesta aleatoria,

sobre todo si usamos C.I de la familia CMOS, que tienen una impedancia de entrada en sus patillas muy elevada (del orden de Mohmios) a diferencia de los TTL que la tienen más baja (del orden de Kohmios).

Como resumen tenemos la tabla de excitación de biestables:

Q_t	Q_{t+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Q_t	Q_{t+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

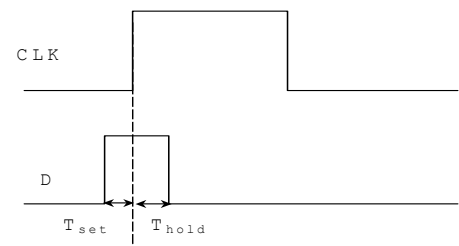
Q_t	Q_{t+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Q_t	Q_{t+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Características Dinámicas de los biestables.

- **Tiempo de set-up:** T_{set} , periodo de tiempo que deben estar los datos estables antes del impulso de reloj.
- **Tiempo de hold:** T_{hold} , periodo de tiempo que debe mantenerse los datos tras el impulso de reloj.

Para un biestable activo por flanco de subida:



3.- Análisis y síntesis de sistemas síncronos basados en biestables.

El análisis de circuitos como hemos visto con los sistemas combinacionales, consiste una vez definido el circuito, esto es, los componentes que lo forman y su interconexión, obtener la respuesta del circuito. En nuestro caso se trata de obtener la evolución del sistema secuencial síncrono en función de sus entradas. Por el contrario, la síntesis de circuitos consiste en el diseño del circuito a partir de unas especificaciones concisas de la función que debe desarrollar el sistema.

3.1.- Análisis.

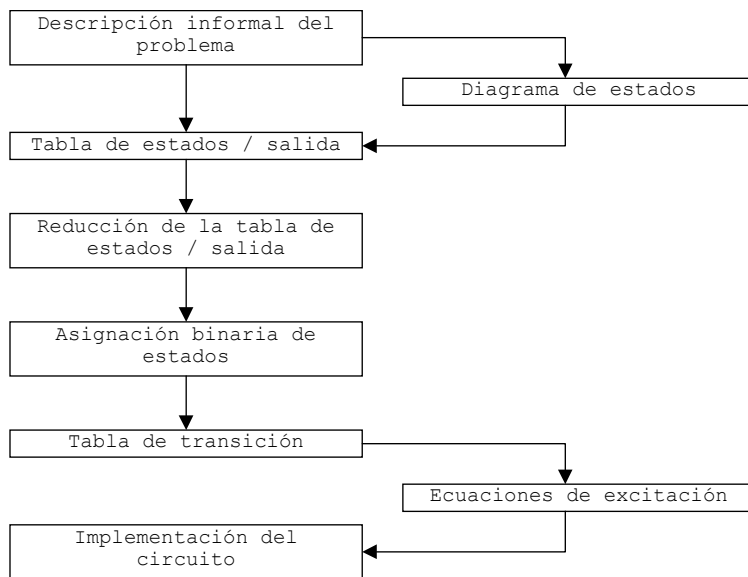
Fases del análisis de circuitos síncronos:

1. Asignación de ***variables de estado*** de forma que el número de estados sea 2^K variables de estado. Por ejemplo, un circuito con dos biestables solo puede tener un máximo de cuatro estados.
2. Obtención de las ***ecuaciones de excitación*** a partir del circuito. Consiste en la obtención de las ecuaciones algebraicas que ponen las entradas de los biestables en función de la entrada del sistema y de las salidas de los biestables que contienen el estado actual.

3. **Tabla de estados** a partir de las ecuaciones de excitación. Puesto que conocemos las ecuaciones de excitación podemos conocer el estado siguiente dado el estado actual mediante la tabla de excitación del biestable que forma el circuito secuencial. Esta información se refleja en una tabla de estados o diagrama de fases por las que pasa el circuito.
4. **Diagrama de estados** a partir de la tabla de estados. Se puede obtener directamente un diagrama de estados a partir de la tabla de fases del sistema secuencial.

3.2.- Fases de la síntesis de circuitos síncronos.

Fases de la síntesis de circuitos síncronos:



1. Descripción informal del problema
2. Diagrama de estados
3. Tabla de estados / salida
4. Reducción de la tabla de estados / salida
5. Asignación binaria de estados
6. Tabla de transición
7. Ecuaciones de excitación
8. Implementación del circuito

3.3.- Reducción de Tablas de Estado.

El primer paso en el diseño de un circuito secuencial después de una descripción verbal del comportamiento deseado, es la construcción de la tabla de estado que describe el comportamiento del circuito. Dicha tabla puede contener más estados de los estrictamente necesarios.

La minimización del número de estados reduce en muchos casos la complejidad y el coste de la realización de una máquina (el número de elementos de memoria requerido en la realización de la máquina, es directamente proporcional al número de estados). Para una máquina de n -estados $k = \log_2(n)$ variables de estado.

Hay que dejar claro que todo diagrama de estados puede construirse de diferentes maneras y así tener varios circuitos lógicos secuenciales asociados, mientras que todo circuito lógico secuencial determinista que analicemos tendrá un solo diagrama de estados que describa su funcionamiento.

Una tabla de estado está **completamente especificada** si para toda entrada y estado presente tiene especificada la salida y el próximo estado. Una tabla de estado está **incompletamente especificada** si existe indeterminación, al menos, en una salida o próximo estado.

DEF 1. Dos estados q_i y q_j de M son **equivalentes** si para toda secuencia de entrada a M produce la misma secuencia de salida, independiente de si q_i o q_j es el estado inicial. Es decir, dos estados q_i y q_j son equivalentes si:

- a) Tienen la misma salida.
- b) Sus próximos estados son equivalentes.

La equivalencia tiene las siguientes propiedades:

- A. **Reflexiva:** q_i es equivalente a q_i .
- B. **Simétrica:** si q_i es equivalente a q_j , también lo es q_j a q_i .
- C. **Transitiva:** si q_i es equivalente a q_j y q_j es equivalente a q_k , entonces q_i es equivalente a q_k .

DEF 2. Dos estados q_i y q_j de M son **incompatibles** si no son equivalentes.

DEF 3. Dos estados q_i y q_j de M son **compatibles** si ambos tienen la misma salida.

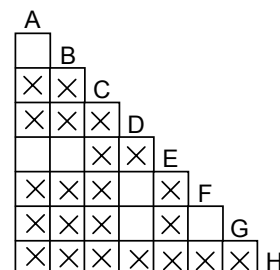
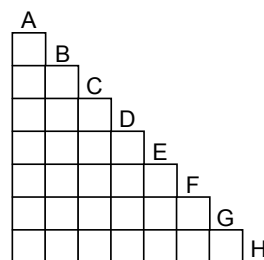
3.3.1.- Método de la carta de implicación para tablas completamente especificadas.

Tomaremos como ejemplo la siguiente tabla de transición de estados:

S	0	1
A	B 1	C 1
B	B 1	C 1
C	D 0	E 1
D	F 0	G 0
E	A 1	E 1
F	F 0	G 0
G	D 0	H 0
H	A 1	H 0

El sistema secuencial posee una entrada S , una salida y ocho estados A..H. Para obtener la tabla de estados mínima debemos comprobar si se pueden reducir o eliminar estados. Para ello seguimos el proceso de reducción de estados, que consta de los siguientes pasos:

1. Formar la **lista de estados con salidas diferentes**, que son estados incompatibles.
2. Construir la **lista de pares compatibles** o de reducción, que es una tabla en forma de escalera que se obtiene asignando un escalón a cada estado como se observa en la siguiente figura.



- En esa tabla, marcar las casillas de los pares de la lista del paso 1, o sea, los **pares incompatibles**.
- Observando la tabla de estados, escribir dentro de cada casilla de la tabla de reducción los estados que deben ser compatibles para que el par que corresponde a dicha casilla lo sea también:

A								
	B							
X	X	C						
X	X	X	D					
A-B C-E	A-B C-E	X	X	E				
X	X	X		X	F			
X	X	X	D-F G-H	X	D-F G-H	G		
X	X	X	X	X	X	X	H	

A								
	B							
X	X	C						
X	X	X	D					
A-B C-E	A-B C-E	X	X	E				
X	X	X		X	F			
X	X	X	D-F G-H	X	D-F G-H	G		
X	X	X	X	X	X	X	H	

- En la tabla de reducción tachar aquellas casillas en las que exista escrito algún par de estados incompatibles. Repetir este paso de forma iterativa hasta que se hallan marcado todos los pares incompatibles.

Al final de este punto, todas las celdas sin tachar corresponden a parejas de estados compatibles (o equivalentes si, como este caso, la tabla de estados salida está completamente especificada).

- Construir la lista de estados compatibles.** Para ello hay que formar una tabla con tres columnas: en la primera anotar un estado (se empieza por el del escalón más bajo); en la segunda se anotan los estados equivalentes del primero (aquellos cuyas casillas no estén marcadas); en la tercera se anotan los estados compatibles. En máquinas completamente especificadas, los compatibles se agrupan por clases de equivalencia, por lo que la lista final de compatibles se forma agrupando todos los estados que sean compatibles entre sí.

A partir de los compatibles se construye la nueva tabla de estados.

S	X=0	X=1
A	B 1	C 1
C	D 0	E 1
D	F 0	G 0
E	A 1	E 1
G	D 0	H 0
H	A 1	H 0

Equivalentes	Compatibles
--	{H}
--	{H,G}
--	{H,G, F}
--	{H,G, F, E}
F	{H,G, DF, E}
--	{H,G, DF, E, C}
--	{H,G, DF, E, C, B}
B	{H,G, DF, E, C, AB}

3.3.2.- Método de la carta de implicación para tablas incompletamente especificadas.

Las **inespecificaciones** se pueden originar por secuencias de entrada que nunca van a aparecer, o bien inespecificaciones en los próximos estados, o bien salidas que no se chequean para unas determinadas entradas.

DEF 1. Dos estados q_i , q_j son **compatibles** si para cualquier secuencia de entrada aplicadas a ambos, las salidas coinciden si están especificadas.

	00	01	10	11
A	, 0	, -	, -	, 1
B	, -	, 1	, -	, 1
C	, 1	, -	, 1	, 1

10 — $A \approx B$ son compatibles
 — $B \approx C$ son compatibles
 — $A \approx C$ no son compatibles

La relación de compatibilidad ahora no cumple la propiedad transitiva.

Teorema: Dos estados q_i, q_j son compatibles si:

- i) Las salidas coinciden, si están especificadas.
- ii) Los próximos estados son compatibles, si están especificados.

DEF 2. Un conjunto de estados $C_i = \{q_1, \dots, q_i\}$ de una tabla Q forma un conjunto compatible si todos sus miembros son compatibles entre sí.

DEF 3. Un conjunto de compatibles es un **Máximo Compatible** si no es subconjunto de ningún otro conjunto compatible.

DEF 4. Todo estado que no sea compatible con ningún otro es un Máximo Compatible.

Para que una máquina secuencial más sencilla N sea equivalente a la máquina secuencial primitiva M y sea la máxima reducción debe cumplir:

- a) **Condición de cubrimiento:** Cada estado de M tiene que estar cubierto por al menos un conjunto compatible.

DEF 5. Un estado p_j “cubre” a un estado q_i ($p_j \supseteq q_i$) si aplicado a ambos estados p_j, q_i , las secuencias de salida son idénticas siempre que la salida esté especificada.

El concepto de cubrimiento, a diferencia de la relación de compatibilidad, posee la propiedad transitiva y es una relación de orden.

DEF 6. Una tabla N “cubre” a la tabla de M si para cada estado q_i existe al menos un estado p_j tal que $p_j \supseteq q_i$. Si un estado p_i cubre a un conjunto de estados C_i estos estados forman un conjunto compatible.

- b) **Condición de cerramiento:** Los próximos estados de cada uno de un Conjunto Compatible (C.C.) deben de estar incluidos en, al menos, en uno de los C.C. seleccionados.

El conjunto de estados de $N = \{p_1..p_r\}$ debe ser tal que cada C.C. de M este cubierto por uno de los estados de N, si y solo si la colección de C.C. de M es cerrada.

- c) **Condición de mínimo:** El número de C.C. seleccionados debe ser el mínimo.

El problema es, pues, encontrar una mínima colección de C.C’s cerrada y que incluya todos los estados de la máquina original.

Determinación de los máximos compatibles (MC's)

Los procedimientos más usuales se basan en la carta de implicación. Sea la tabla de estado y su carta de implicación de la figura siguiente:

Q _i	I1	I2	I3	I4
1	2 0	- 1	3 -	2 0
2	3 0	5 1	2 0	- -
3	3 0	4 1	- -	5 0
4	- -	- -	1 1	- -

1				
2,3	2			
2,3	4,5	3		
2,3	1,5	1,4	4	
1,3	×		1,2	5

, sus Máximos Compatibles se pueden obtener mediante:

- A) **Método algorítmico:** Sea $k = n-1$, $n =$ número de estados
1. Sean los conjuntos $L' = \emptyset$, $L = \{ n\text{-ésimo estado} \}$
 2. Formar $S_k = \{ \text{estados compatibles con el } k\text{-ésimo estado} \}$
Generar $L' = \{ (S_k \cap L) \cup k \}$
Ampliar $L = \{ L \cup L' \}$
Eliminar los elementos redundantes en L .
 3. Decrementar k , $k = k - 1$
 4. Si $k > 0$ volver a 2. Caso contrario finaliza.

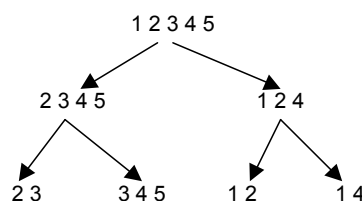
Ejemplo:

k	S _k	L' = { \emptyset }	L={5}
4	5	45	(45)
3	45	345	(345),(45)
2	3	23	(23),(345)
1	24	12,14	(23),(345),(12),(14)
0	final		

M.C.'s = $\{ (12),(345),(12),(14) \}$

B) Método en árbol (basado en el concepto de partición)

1. Iniciar con el conjunto de todos los estados.
2. Aplicar la incompatibilidad existente en la columna 1 de la tabla de implicación, formándose dos subconjuntos.
3. Generar los siguientes subconjuntos aplicando el paso 2 a las restantes columnas (2,...,n) de izquierda a derecha de la tabla de implicación.
4. Eliminar los subconjuntos contenidos en otros.



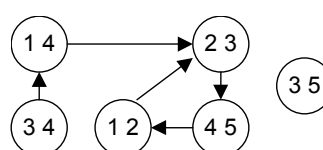
A partir de los Máximos Compatibles se determina una máquina N que cumpla las tres condiciones enunciadas anteriormente. Si N la formamos con todos los M.C.'s obtenidos, cumplirá las condiciones de cubrimiento y cerramiento pero no la mínima.

Diagrama o Grafo de Implicación.

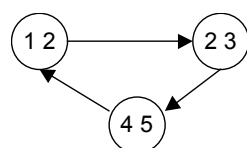
Cada nudo representa a un M.C. o a un par de compatibles y los arcos dirigidos van desde cada par a los nudos implicados por este.

Un conjunto de compatibles $C_i = \{q_1, q_2, \dots\}$ implica a otro conjunto de compatibles $C_j = \{q_1', q_2', \dots\}$ si para alguna entrada I_k , C_j es I_k -sucesor de C_i .

Pares compatibles: (12),(34),(35),(45),(12) y (14)



Cubrimiento cerrado no mínimo. Se observa un cubrimiento cerrado formado por solo tres



Cubrimiento Cerrado Mínimo
 $N = \{ (12), (23), (45) \}$

	I1	I2	I3	I4
A	B, 0	C, 1	B, 0	* A, 0
B	B, 0	C, 1	* B, 0	C, 0
C	-, -	A, 1	A, 1	-, -

* Puede ser A o B. Se elige como próximo estado el que corresponde al estable.

Una vez minimizada la tabla hay que escoger un asignamiento binario adecuado (reducir la complejidad del circuito, evitar carreras y azares).

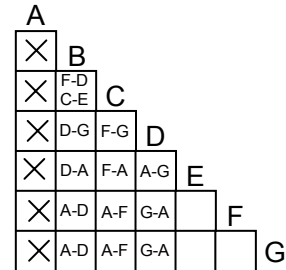
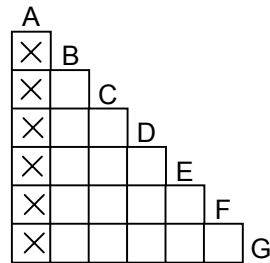
Ejemplo: Tomaremos como ejemplo la siguiente tabla de transición de estados de un autómata de Mealy:

S	X=0	X=1
A	A 0	B 0
B	D 1	C 0
C	F 1	E 0
D	G 1	-
E	A 1	-
F	A 1	-
G	A 1	-

NS Z

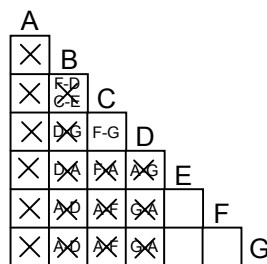
El sistema secuencial posee una entrada(X), una salida (Z) y siete estados A..G.

Paso 1-3: Marcar estados incompatibles.



Paso 4: Escribir estados que deben ser compatibles para que el par correspondiente a dicha casilla lo sea también:

Paso 5: Tachar casillas de pares incompatibles:



COMPATIBLES

{A,B,CD,EFG}

↓ ↓ ↓ ↓
{A,B,C, D}

S \ X	0	1
A	A , 0	B , 0
B	C , 1	C , 0
C	D , 1	D , 0
D	A , 1	- , -

3.2.3.- Asignación de Estados.

Correspondencia entre los estados y una codificación binaria, pasando de la Tabla de Estado a la Tabla de Transición, con el mínimo número de variables internas (y_1, y_2, \dots, y_r).

$$n = \text{número de estados. } r = \log_2(n)$$

Dependiendo del asignamiento elegido se reducirá el número de conexiones y puertas lógicas en los circuitos secuenciales síncronos.

Método exhaustivo. Consiste en probar todas las asignaciones posibles.

Para $r = \text{número de variables de estados}$, se tiene $2^{r-1} < n \leq 2^r$ codificaciones posibles distintas. El número de asignaciones distintas es:

$$N = \frac{(2^r - 1)!}{(2^r - n)! r!}$$

, donde no se consideran asignaciones distintas: a) por permutación de las variables, b) complementación de las columnas; ya que no se altera el diseño del circuito secuencial.

Ejemplo: Para $n = 4$ se necesitan dos variables internas y_1, y_2 . El número de codificaciones distintas es $2^r = 4$: (00,01,10,11).

El número de asignaciones diferentes es: $N = \frac{(4-1)!}{1! 2!} = 3$

La complejidad del circuito combinacional obtenido para la máquina secuencial depende de la asignación binaria elegida, para lo que, aún, no existe un método efectivo. Si el número de estados es reducido se puede aplicar con óptimo resultado el Álgebra de Partición o el Sistema de Conjuntos.

Regla básica: Agrupar los 1's de excitación a los biestables lo máximo posible (reducción de coste).

Dos códigos se dicen adyacentes si difieren en una sola variable de estado.

Regla I: a) Comprobar los estados que tienen idénticos próximos estados en cada columna. A tales estados se les da asignaciones adjuntas.

- b) Comprobar los estados que tienen los mismos próximos estados pero en diferentes columnas. Tendrán asignaciones adjuntas si los próximos estados pueden tener asignaciones adjuntas.
- c) Estados con idénticos próximos estados en algunas columnas tendrán asignaciones adjuntas.

Regla II: Los próximos estados de una fila dada deben estar con asignaciones adjuntas.

Regla III: Los estados que tengan la misma salida para una cierta entrada deben tener asignaciones adjuntas.

Estas reglas se aplican en el orden descendente expuesto. Ejemplo:

Diseñar un comprobador de paridad para caracteres de 4 bits. Arrancando desde reset, el circuito recibe los caracteres de 4 bits en serie sobre una sola línea X de entrada, al tiempo del 4º bit, la salida Z pasa a 1 si el número total de 1's es par. Para cualquier otro caso $Z = 0$. Recibido el 4º bit resetea para volver a su posición inicial esperando otro carácter de 4 bits.

Diagrama de estado:

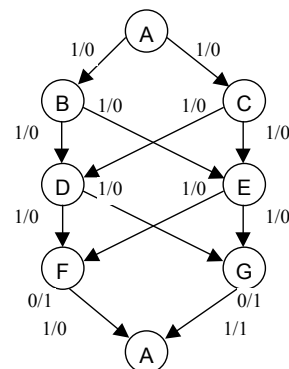
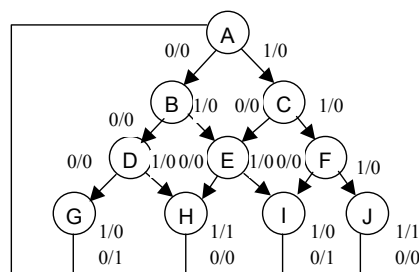


Tabla Estado/Salida

	0	1
A	B, 0	C, 0
B	D, 0	E, 0
C	E, 0	D, 0
D	F, 0	G, 0
E	G, 0	F, 0
F	A, 1	A, 0
G	A, 0	A, 1

Reducción de estados

$P0 = \{ A B C D E F G \}$
 $P1 = \{ ABCDE, F, G \}$
 $P2 = \{ ABC, D, E, F, G \}$
 $P3 = \{ A, B, C, D, E, F, G \}$

La tabla corresponde al diagrama reducido. En la reducción de estados se observa que es la tabla mínima. Asignación de estados:

Regla 1a: F y G tienen que ser adyacentes.
 Regla 1b: B y C tendrán que ser adyacentes si lo son D y E.
 D y E tendrán que ser adyacentes si lo son F y G
 Regla 2: A debe ser adyacente con B y C.

$y_0 \setminus y_2 y_1$	00	01	11	10
0	A	F	D	B
1		G	E	C

Mediante una correcta asignación reducimos los circuitos combinacionales de la máquina de estados.

Tabla de transición – salida.

$y_2 y_1 y_0 \setminus x$		0	1
000	A	100,0	101,0
100	B	110,0	111,0
101	C	111,0	110,0
110	D	010,0	011,0
111	E	011,0	010,0
010	F	000,1	000,0
011	G	000,0	000,1
001	.	---,-	---,-

Calculada la tabla de transición, se pueden hallar las ecuaciones de excitación de los biestables.

Utilizando biestables J-K.

$y_2 y_1 y_0 \setminus x$	0	1
000	1 X	1 X
001	X X	X X
010	0 X	0 X
011	0 X	0 X
100	X 0	X 0
101	X 0	X 0
110	X 1	X 1
111	X 1	X 1
J2, K2		

$y_2 y_1 y_0 \setminus x$	0	1
000	0 X	0 X
001	X X	X X
010	X 1	X 1
011	X 1	X 1
100	1 X	1 X
101	1 X	1 X
110	X 0	X 0
111	X 0	X 0
J1, K1		

$y_2 y_1 y_0 \setminus x$	0	1
000	0 X	1 X
001	X X	X X
010	0 X	0 X
011	X 1	X 1
100	0 X	1 X
101	X 0	X 1
110	0 X	1 X
111	X 0	X 1
J0, K0		

3.4.- Problemas estructurales en los circuitos secuenciales.

Vamos a analizar en este apartado la bondad del modelado mediante una máquina secuencial de un circuito secuencial de conmutación. Habrá que puntualizar que los problemas estructurales se suelen poner de manifiesto solo cuando se da alguna de estas condiciones: a) la señal de reloj que alimenta al circuito síncrono tiene una frecuencia muy alta (circuitos secuenciales de alta o muy alta velocidad $> 20\text{MHz}$), b) los impulsos de la señal de reloj son muy breves o tienen unos flancos de subida o bajada abruptos.

- i) Una primera cuestión es la discretización del tiempo. Hemos considerado los retrasos en los lazos de realimentación, lo que es una idealización dado que los retrasos estarán distribuidos entre las diversas puertas que constituyen el circuito. Para ser determinista el modelo, no se cambiarán las entradas al sistema hasta que este no ha alcanzado un estado estable”.
- ii) Consecuencia de lo anterior es que si para determinada entrada no existe ningún estado estable, la máquina no es determinista (no da una salida definida) respecto al comportamiento del circuito.

Ejemplo: Sea la máquina secuencial de la figura. Si del estado 00 con entrada I0 pasamos a la entrada I1, el sistema comenzará una evolución $00 \rightarrow 01 \rightarrow 11 \rightarrow 10 \rightarrow 00 \rightarrow \dots$ que no alcanza ningún estado estable, y al no ser homogéneos los distintos tiempos de transición difícilmente sabremos cuando se produzca un cambio en la entrada en que estado se encuentra.

- iii) Otro problema adicional es el de las *carreras*, “situaciones no deterministas consecuencia del cambio simultáneo de varias variables de estado”. Físicamente la simultaneidad en el cambio de dos o más señales va a ser muy poco probable, lo cual llevará a pasar por situaciones intermedias donde solo cambie una variable, estas evoluciones podrán coincidir con el estado terminal estable que predice la máquina secuencial, en cuyo caso se dice que son *carreras no críticas*, o bien puede que exista la posibilidad de evolución por un camino que no finalice en el estado estable predicho por la máquina secuencial, en cuyo caso se dice que la *carrera es crítica*. Las carreras críticas hacen que el circuito no se comporte como esperábamos. Debe ser algo a evitar en un buen diseño.

	I0	I1
00	(00)	01
01		11
11		10
10		00

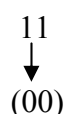
y1,y0	00	01	11	10
00	(00)	01	(00)	01
01	00	(01)	00	11
11	00 *	(11)	00 %	(11)
10	00	01 *	11	(10)

Y1, Y0

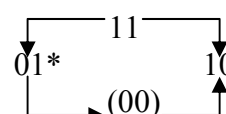
las casillas marcadas corresponden a situaciones de carreras:

$$(x1,x0,y1,y0) = (0,0,1,1) \rightarrow (Y1,Y0) = (0,0)$$

evolución máq. secuencial



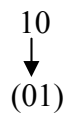
posibles evoluciones físicas



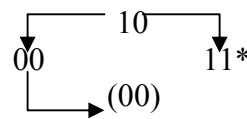
[*] corresponde a una carrera no crítica

$$(x_1, x_0, y_1, y_0) = (0, 1, 1, 0) \rightarrow (Y_1, Y_0) = (0, 1)$$

evolución máq. secuencial



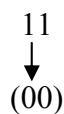
posibles evoluciones físicas



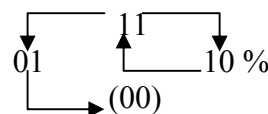
[*] corresponde a una carrera crítica

$$(x_1, x_0, y_1, y_0) = (0, 0, 1, 1) \rightarrow (Y_1, Y_0) = (1, 1)$$

evolución máq. secuencial



posibles evoluciones físicas



[%] corresponde a un ciclo (situación crítica)

Es obvio que si existen más de un estado estable para una misma entrada, las carreras que se produzcan en esa columna pueden ser críticas.

- iv) Un último problema que puede surgir en los circuitos secuenciales son los *riesgos o azares* consecuencia de cambios múltiples en las variables de entrada, el problema es similar al de las carreras. Si la entrada del sistema cambia de forma que exige un cambio múltiple de las señales, lo más probable es que estas no lo hagan simultáneamente y por ello no alcancen el estado predicho por la máquina secuencial.

$$\text{Ejemplo: } (x_1, x_0, y_1, y_0) = (0, 0, 0, 0) \rightarrow (x_1, x_0, y_1, y_0) = (1, 1, ?, ?)$$

y ₁ ,y ₀	00	01	11	10
00	(00)	01	(00)	01
01		(01)	00	11
11			(11)	10
10			11	(10)

Y₁,Y₀
x₀,x₁: 00→01→11

y ₁ ,y ₀	00	01	11	10
00	(00)	01	(00)	01
01		(01)	00	11
11			(11)	10
10			11	(10)

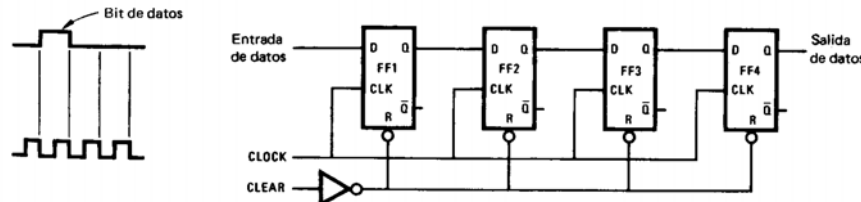
Y₁,Y₀
x₀,x₁: 00→10→11

Los problemas, el (i) y (iv) afectan al modo de operación del circuito y por tanto solucionables si operamos en Modo Fundamental y cambios simples en las entradas. Los problemas (ii) y (iii) son estructurales y por tanto su solución pasa por cambiar el circuito.

4.- Sistemas secuenciales síncronos integrados de aplicación general.

4.1.- Registros.

Un registro de desplazamiento consiste en un conjunto de flip-flops interconectados de diversas formas, en el cual la salida de un flip-flop es la entrada de otro.

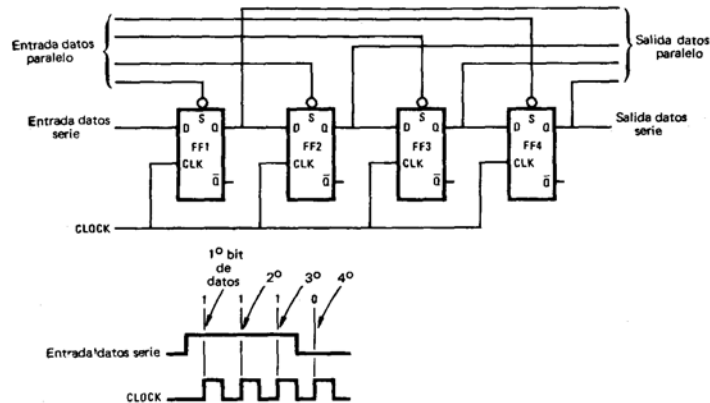


En los registros de desplazamiento todos los flip-flop tienen una señal común y se activan y desactivan de forma sincronizada (al mismo tiempo). Supongamos que suministramos a la entrada del registro un nivel lógico 1; con la llegada del impulso de reloj, el nivel 1 se almacena en FF1 y aparece en su salida, transcurrido un corto tiempo tras el flanco de la señal de reloj. En el próximo impulso de reloj FF2 recibe el nivel 1 desde la salida de FF1. Mientras tanto FF1 podrá recibir un nuevo dato a su entrada. El primer bit se desplaza a través de los flip-flop al ritmo de la señal de reloj.

Si el registro de desplazamiento recibe más de 4 impulsos de reloj, el 1 lógico será desplazado fuera de FF4 y todos los flip-flop pasarán a contener 0. Un registro de desplazamiento se emplea para almacenar un dato o varios bits. Para introducir en un registro de 4 flip-flop un número de 4 bits deben suministrarse 4 impulsos de reloj, eso se conoce como carga de datos en serie y al registro usado “registro de desplazamiento de 4 bits con carga serie”.

Hay otra alternativa para cargar los registros en “paralelo”, caso en el cual una línea independiente está conectada a la entrada de cada uno de los 4 flip-flop y todos los dígitos se cargan simultáneamente. Esta carga puede tener lugar en el mismo momento de activar la entrada de carga o bien una vez activa la entrada de carga, en el siguiente impulso de reloj. En el primer caso se trata de **carga asíncrona** porque tiene lugar en cualquier momento independientemente de la señal de reloj, en el segundo caso se denomina **carga síncrona** porque tiene lugar en sincronía con la señal de reloj.

También los datos almacenados se pueden extraer en serie o en paralelo. La lectura de un registro en serie se efectúa desplazando los datos a través de los flip-flop y sacándolos en la salida del último, uno a uno. En la lectura en paralelo se sacan todos los datos simultáneamente.

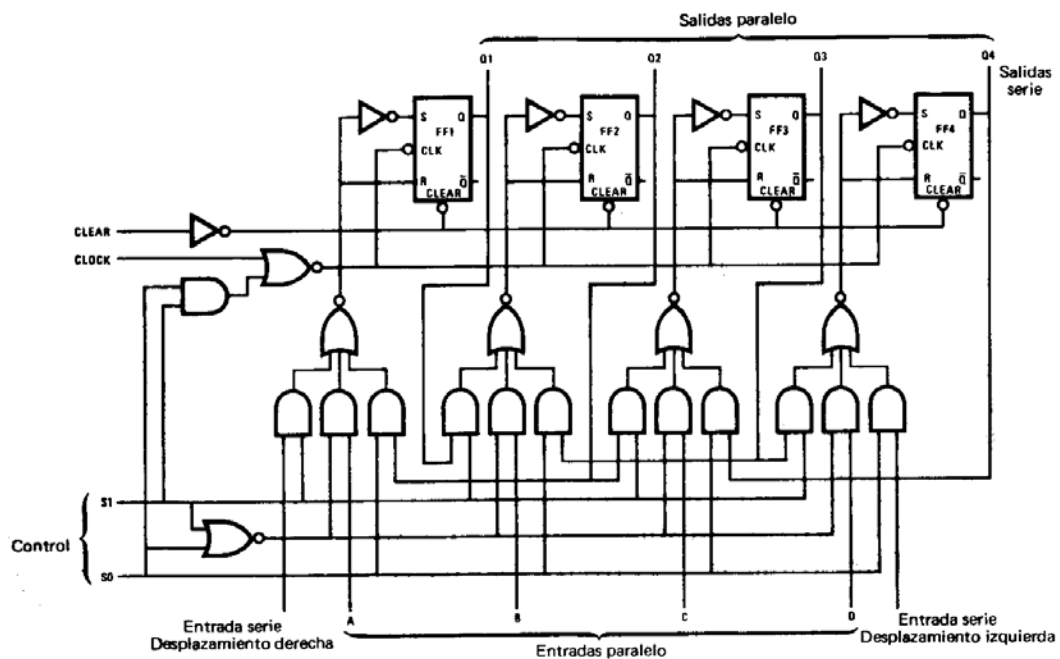


Si el dato es cargado en serie y leído en paralelo se dice que el registro de desplazamiento está funcionando como convertor serie-paralelo; si el dato se carga en paralelo y se lee en serie, funciona como convertor paralelo-serie.

Tipos de Registros

- **Paralelo – Paralelo:** la entrada y salida de datos se realiza en paralelo.
- **Serie – Paralelo:** la entrada es en serie y la salida en paralelo.
- **Paralelo – Serie:** la entrada se realiza en paralelo y la salida en serie.
- **Serie – Serie:** entrada y salida se realiza en serie.
- **Universal:** dispone de cualquier posibilidad de entrada / salida.

Algunos registros están configurados para permitir cualquier posibilidad de entrada/salida además del desplazamiento de los datos tanto a la derecha como a la izquierda. A estos registros se les denomina “universales”, ya que pueden desplazar la información en cualquier dirección, cargarla bien sea en serie o en paralelo, así como extraer los datos en serie o en paralelo.



Registro universal de cuatro bits.

4.2.- Contadores.

Otra aplicación clásica de los flip-flop son los contadores binarios, que se utilizan para contar el número de veces que sucede un cierto acontecimiento. Los circuitos contadores binarios difieren de los registros de desplazamiento en que sus flip-flop están conectados entre sí de forma diferente. Se pretende que el contador disponga del número máximo de estados.

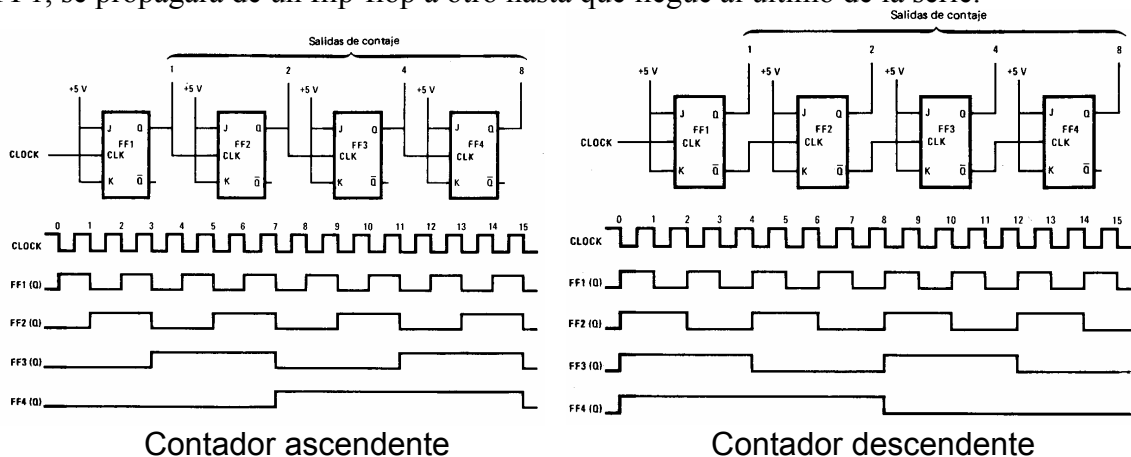
Los contadores se utilizan como circuitos básicos en otros circuitos lógicos. Se emplean en cómputo, como secuenciadores de equipos u operaciones de proceso, en medición y división de frecuencia, manipulación aritmética, medición de intervalos de tiempo y otros muchos fines. Se pueden clasificar en dos grupos fundamentales:

- **Asíncronos**, conocidos también como contadores serie.
- **Síncronos**, a los que se llama contadores paralelo.

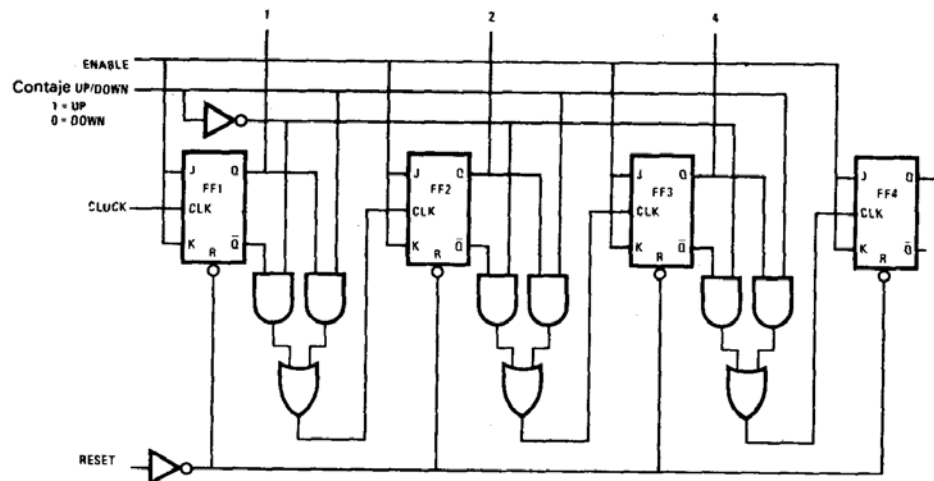
En los contadores síncronos todos los flip-flop cambian de estado simultáneamente, en tanto que en los síncronos cambia de estado un flip-flop y este cambio activa un segundo flip-flop, el cual puede después activar un tercero y así sucesivamente. El número de estados sucesivos a través de los cuales un contador realiza una secuencia antes de que se repita se denomina “módulo”. Los contadores de módulo potencia de 2 son los más fáciles de construir. Un contador es un circuito que realiza una secuencia a través de M estados diferentes en un orden especial, siendo M el módulo del contador.

4.2.1.- Contadores Asíncronos.

Un contador de propagación binaria es el más básico de todos, contará hacia arriba o incrementando en el código 8421, o bien decrementando o hacia abajo (los biestables e activan con el flanco posterior del impulso). El efecto de un impulso de reloj introducido en la entrada de FF1, se propagará de un flip-flop a otro hasta que llegue al último de la serie.



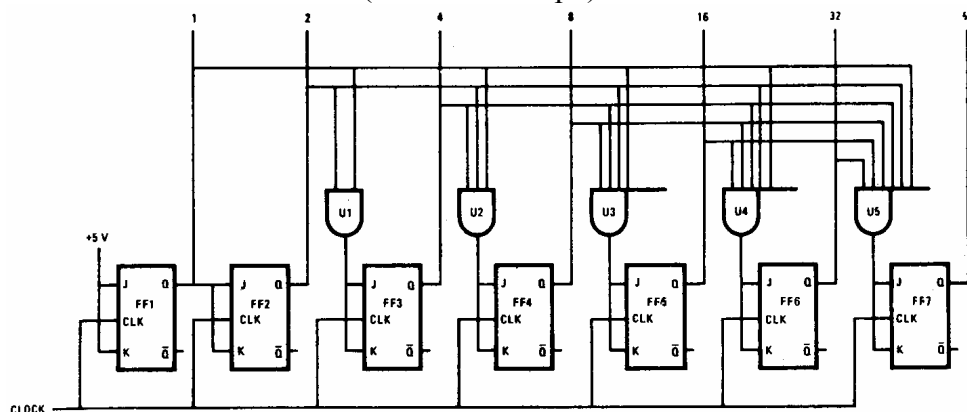
Una característica importante de cualquier contador es la velocidad con que puede funcionar. Si cada flip-flop tiene un retardo de propagación de 25 ns, el retardo total desde el momento en que se aplique a FF1, el flanco posterior de un impulso de reloj, hasta que FF4 haya completado su cambio de estado es de 100 ns. Por tanto, el siguiente flanco de impulso de reloj no puede tener lugar hasta 100 ns más tarde. Durante este tiempo los flip-flop cambian de estado y la salida del contador será incorrecta. Esta limitación es el principal inconveniente del contador asíncrono. Un retardo de 100 ns entre los flancos posteriores de dos impulsos de reloj permite una frecuencia máxima de reloj de 10 MHz (que es el inverso de 100 ns).



Contador up/down asincrónico con señales de habilitación y reset.

4.2.2.- Contadores síncronos.

Los contadores síncronos se basan en el mismo principio que los asíncronos, exceptuando que en este caso todos los flip-flop son activados mediante una señal de reloj común y todos cambian de estado de forma síncrona (al mismo tiempo).



Como todos los registros reciben el impulso de reloj y cambian de estado al mismo tiempo, el retardo total es independiente del número de registros y es exactamente el de un registro. Si el tiempo total de retardo de propagación de un registro JK y de la puerta AND asociada es de 35 ns (25 + 10 ns), los impulsos de reloj pueden producirse con una frecuencia máxima de 30 MHz en un contador síncrono.

Todas las líneas de un contador síncrono cambian simultáneamente, no hay estados intermedios con salidas de contador incorrectas. Como inconveniente, precisa más puertas lógicas que los asíncronos, y por tanto es más complejo y costoso. Además, si se tuviese que ampliar el contador, el número de entradas de la puerta AND de la última etapa sería muy elevado y constituiría una limitación de tipo práctico.

Contador BCD.

Un contador BCD cuenta hasta diez y después se pone a 0 o reset y comienza a contar nuevamente. El contador BCD funciona igual que cualquier otro contador. Un contador binario

o 8421 con más de 10 estados avanza a 1010 en el impulso siguiente al 9, mientras que el contador BCD retorna por sí solo a 0000.

Contador de módulo N

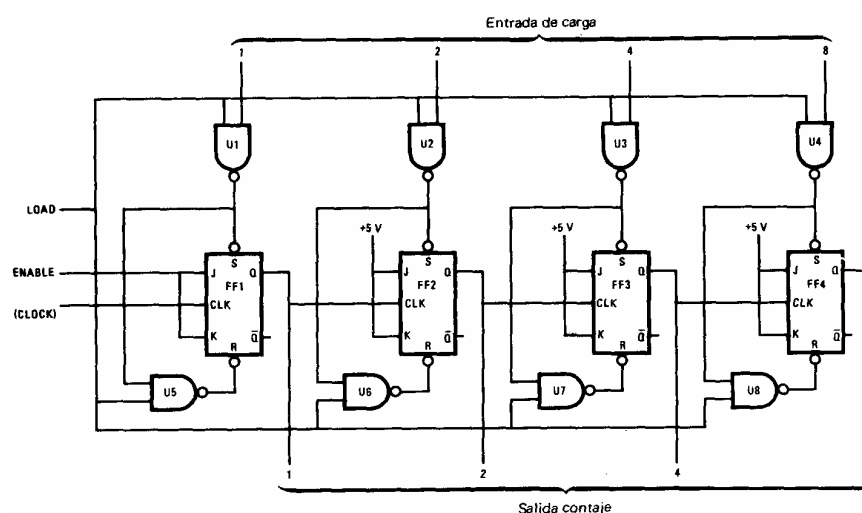
Un contador de módulo N tiene N estados diferentes. Por lo tanto, puede ser asíncrono o síncrono, con tal que contenga los circuitos precisos para controlar el número de estados que puede tener. Por ejemplo, un contador BCD es el que puede contar hasta 16, pero cuyo módulo está limitado a 10 mediante puertas que detectan la cuenta 10 y ponen a cero el contador cuando ésta se produce.

Para conseguir un factor de división o módulo mayor conectamos en serie varios contadores de menor módulo. Para determinar el módulo de la combinación de contadores en serie se multiplican los módulos particulares de cada uno.

Contador programable.

Contador programable es cualquiera cuyo módulo o diagrama de cómputo pueda modificarse de alguna forma mediante una señal de control, en lugar de realizar modificaciones en el circuito. Debe disponer de una señal de control que reajusta el contador a un determinado número o que lo pone en reset (puesta a 0) para comenzar otra vez el cómputo.

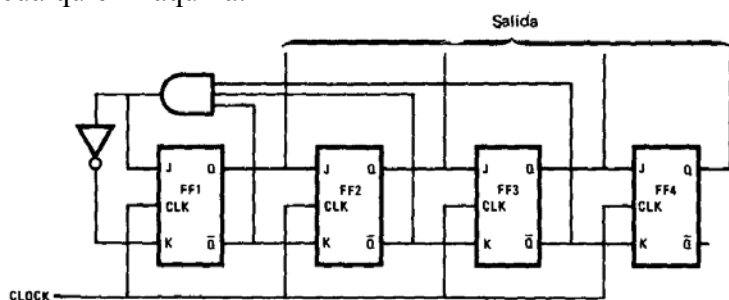
Un contador programable puede realizarse con un contador asíncrono de 4 bit que puede preajustarse a cualquier número entre el 0 y el 15, mediante las puertas U1 a U4, activando la señal de carga (colocando un nivel lógico 0). Si se va a usar como contador de módulo 7, se preajusta a un cómputo 8 y se deja que cuente desde 8 hasta 15. Al llegar a 15, el contador debe volver a 8 nuevamente (antes del siguiente impulso de reloj), si se quiere repetir el ciclo de cómputo de la misma forma. Con este sistema lo que se determina es el número en que empieza el cómputo.



Contador en anillo.

Un contador anular o en anillo es simplemente un registro de desplazamiento de circulación o rotación, es decir un registro cuya salida está conectada a la entrada J y K del FF1, cerrando el círculo. Este se carga con un bit lógico en el primer flip-flop y ceros en los demás, de forma que este 1 circula repetidamente por el registro. Como consecuencia, el

contador saca una serie de estados únicos, que se pueden usar para regular la secuencia de cualquier máquina.

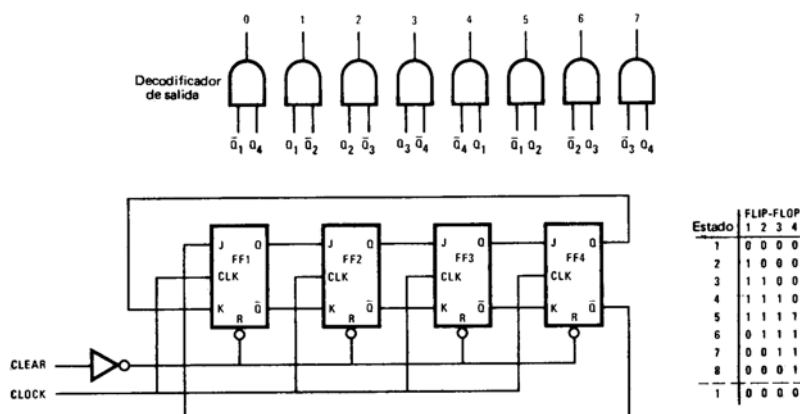


Ventaja: No es necesario decodificación.

Desventaja: No se aprovechan al máximo los biestables.

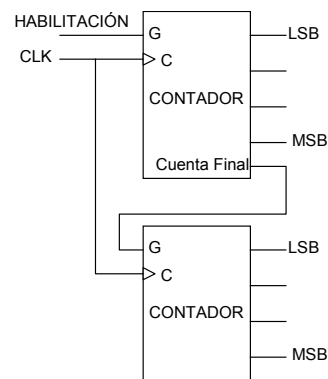
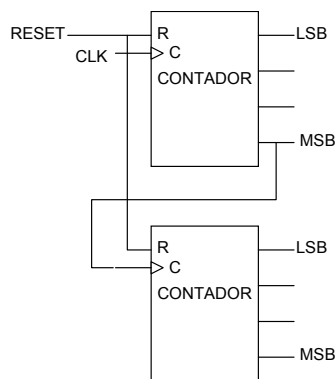
Contador Johnson.

El contador Johnson (denominado también contador anular cruzado o invertido) es una ligera modificación del anterior, la salida Q del último flip-flop se vuelve a conectar a la entrada J del primero. El contador pasa por $2N$ estados diferentes, siendo N el número de flip-flop. Un contador Johnson de 4 bits tiene el doble de estados que uno anular de 4 bits, necesitando un decodificador para dar salida a los estados.



4.2.3.- Asociación de contadores.

Podemos usar varios contadores para ampliar el número de estados de cuenta. Hay dos formas de ampliar el número de bits de contaje asociando dos contadores: de forma asíncrona o de forma síncrona. En el primer caso se conecta la entrada de reloj del contador de bits de mayor peso a la última cuenta (MSB) del contador de menor peso. La ventaja es la sencillez del circuito, la desventaja la velocidad limitada debido al acarreo asíncrono.



En el segundo caso se alimenta con la misma señal de reloj a ambos contadores y se dispone de entrada y salida de habilitación de cuenta en ambos contadores (G). Esta señal se usa para habilitar la cuenta del contador de los bits de mayor peso cuando se rebase la capacidad del contador de bits de menor peso. Es decir, cuando todas las salidas del primer contador pasan de 1111 a 0000 se habilita el segundo contador y se incrementa en uno la cuenta de los bits de mayor peso. Este contador tiene todas las ventajas del contador síncrono y también sus desventajas.

5.- Síntesis de sistemas secuenciales microprogramables.

Son aquellos en los que la parte combinacional es programable y ha sido realizada mediante alguna de las formas estudiadas. La característica más importante de estos sistemas es que se puede cambiar su diagrama de flujo sin necesidad de modificar sus conexiones.

5.1.- Completos.

El bloque combinacional está formado por una memoria ROM. La memoria que almacena el estado está formada por registros, o bien se puede usar un contador síncrono.

Este método está basado en el empleo de una memoria PROM en la que se graba toda la tabla de decisiones del autómata. La evolución de la máquina secuencial a lo largo del tiempo, se realiza durante una secuencia de lecturas que se efectúan sobre la memoria. Este procedimiento proporciona bastante flexibilidad, puesto que cualquier cambio del algoritmo de control sólo afecta al contenido de la PROM, manteniendo constante la circuitería auxiliar.

El sistema se basa en enviar a la memoria una dirección, que contiene una determinada información codificada, para proceder a realizar un ciclo de lectura en el que se obtiene otra información, que es procesada para deducir el valor de las salidas y el estado siguiente de los flip-flop D, encargados de la codificación de los estados del autómata.

La dirección que se entrega a la memoria PROM, consta de dos campos:

1. Información sobre el estado presente, codificado por los flip-flop ($Q_1 - Q_N$)
2. Un bit por señal de entrada.

La información que se obtiene de la lectura de la dirección que se ha proporcionado a la memoria, también consta de dos campos:

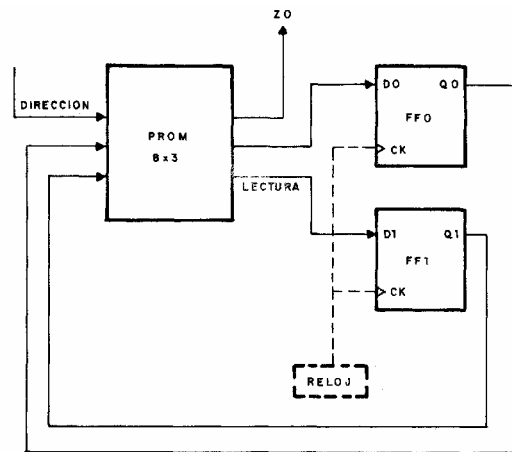
1. Información sobre el estado siguiente del autómata ($Q_1 - Q_N$)
2. Valor de las señales de salida.

Se construye una tabla que contenga todas las direcciones de la memoria y en cada una de ellas la información que debe tener grabada.

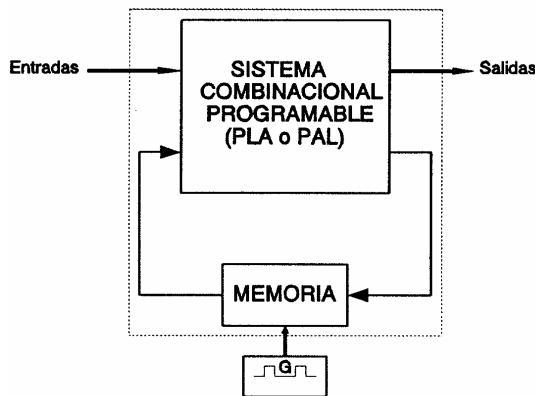
Para cada estado presente hay que grabar en la memoria todas las combinaciones que admiten las variables de entrada, aunque no se empleen. Este tipo de autómata da lugar a una memoria de mucha capacidad, puesto que analiza todas las posibles combinaciones de las variables de entrada, aunque no afecten al estado. Por otra parte, producen muchas palabras con el mismo contenido.

Para la siguiente tabla de transición de estados, podríamos tener el siguiente circuito:

Dirección		Dato		ROM
Entrada X	Estado Act.	Estado Sig.	Salida Z0	Valor HEX
0	0 0	0 1	0	02H
0	0 1	1 0	0	04H
0	1 0	1 1	1	07H
0	1 1	0 0	1	01H
1	0 0	0 0	0	00H
1	0 1	0 0	0	00H
1	1 0	0 0	1	01H
1	1 1	0 0	1	01H



5.2.- Incompletos.



con una señal de reloj común constituye un secuenciador.

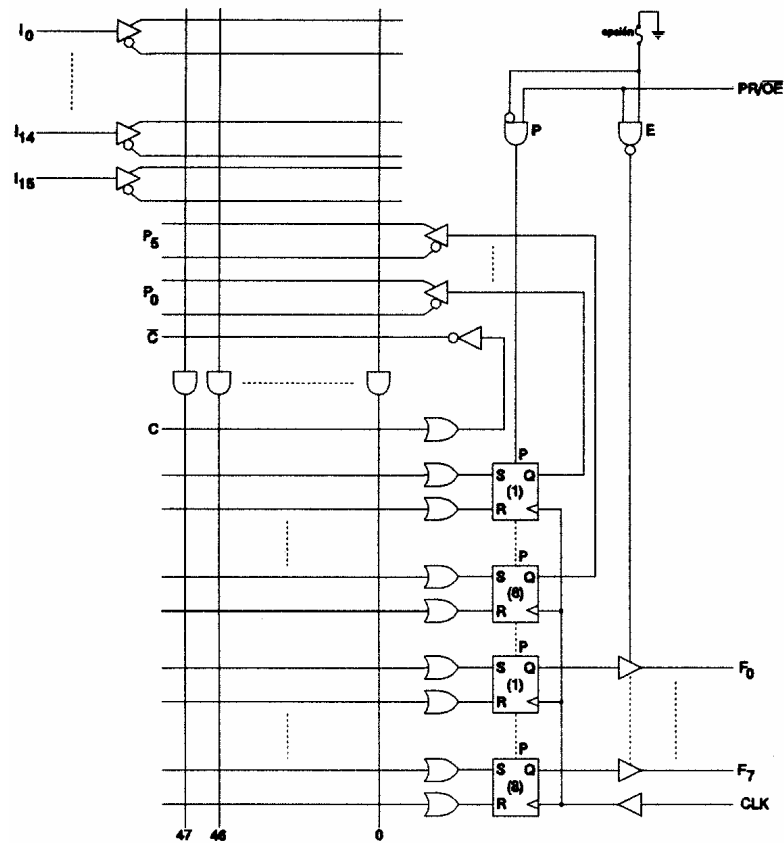
Los sistemas digitales contienen bloques secuenciales síncronos cuya lógica combinacional no suele tener una estructura regular y por tanto son más indicados para implementarse con dispositivos PLA en lugar de ROM como sistemas completos. Los controladores secuenciales y secuenciadores recaen en esta categoría. Las máquinas secuenciales síncronas se pueden clasificar en las variantes síncronas y asíncronas de las formas de Mealy y Moore. En un diseño se pueden encontrar ambos tipos. Un PLD conteniendo una PLA y un conjunto de flip-flops

6.- Sistemas secuenciales programables monochip (PLD's).

6.1.- Arquitectura AND programable OR programable. (PLS)

Las tres partes principales de un secuenciador basado en PLA son las dos matrices lógicas, la matriz AND y la matriz OR (que constituyen la PLA), y la fila de biestables con señal de reloj. Las señales de salida producidas por la matriz OR, alimentan los pines de salida o las entradas de los biestables. Las salidas de los biestables se realimentan en la matriz AND o se dirigen a la salida. No todos los secuenciadores poseen todos estos caminos en su estructura. Los secuenciadores vienen caracterizados por el número de entradas, número de términos producto, número de biestables y número de salidas.

SECUENCIADOR LÓGICO PROGRAMABLE PLS105



El PLS105 posee una matriz AND con 48 términos producto máximo de entradas del dispositivo y sus complementos y de las variables realimentadas y sus complementos. Los términos productos se introducen en la matriz OR que produce las entradas de excitación de los biestables.

Un añadido importante a la PLA es la **matriz de complemento**. Esta línea OR adicional se complementa primero y luego se introduce en la matriz AND donde se comporta como una entrada adicional. Como veremos, el uso de esta matriz puede reducir términos producto.

Los biestables tienen una línea de reloj común. En este dispositivo, todos son de tipo R-S, lo cual tiene ciertas ventajas en el diseño de secuenciadores. Los biestables están conectados en una de las tres formas siguientes:

- Sus salidas se realimentan internamente a la matriz AND, y por tanto no se tiene acceso a ellas mediante patillas externas (“buried”).
- Sus salidas se realimentan internamente y además están accesibles externamente.
- Sus salidas están conectadas directamente al exterior.

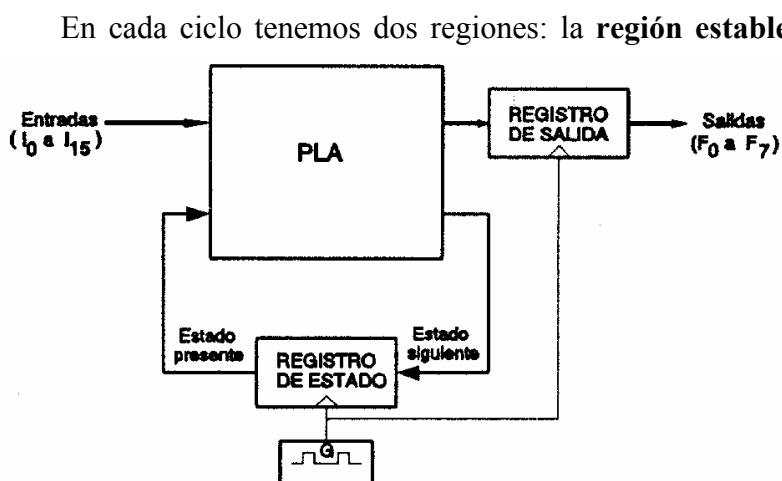
Por último, en este dispositivo en particular, hay una patilla (16) que se puede programar para ser usada de dos formas. En modo “*output enable*” la patilla se usa para controlar el tercer estado de las salidas, y en modo “*preset*” se usa para hacer un preset asíncrono a los biestables. El primer modo es útil cuando las salidas están conectadas a un bus, y el segundo permite seleccionar de forma incondicional un determinado estado de la secuencia. Una propiedad del

dispositivo es que pone a cero los biestables al conectar la alimentación y asegura que el secuenciador comienza en un estado definido.

En el diseño con PLA se suele usar una tabla de estados compacta. La siguiente tabla muestra la tabla PLS para un sumador serie:

Entradas		Estado Presente	Estado siguiente	Salida
a_i	b_i	c_i	c_{i+1}	s_i
0	0	0	0	0
0	1	0	0	1
1	0	0	0	1
1	1	0	1	0
0	0	1	0	1
0	1	1	1	0
1	0	1	1	0
1	1	1	1	1

Tenemos solo una columna para cada entrada en lugar de una para cada combinación de entrada. Estas columnas junto con la columna de “estado presente” definen las entradas a la lógica combinacional cuyas salidas definen el próximo estado y la salida de la máquina de estados. Esta tabla define una máquina de Mealy, porque las salidas dependen del estado y de las entradas.



comienzan a cambiar a sus nuevos valores. Ignorando los cambios a sus entradas por el momento, el cambio del registro de estado provoca que la lógica combinacional comience a evaluar las nuevas señales que forman las salidas asíncronas del dispositivo y las entradas de los biestables. Si el retardo de propagación de la matriz lógica es T_L , el periodo estable comenzará en un tiempo igual a la suma de los valores máximos de T_{CLK} y T_L .

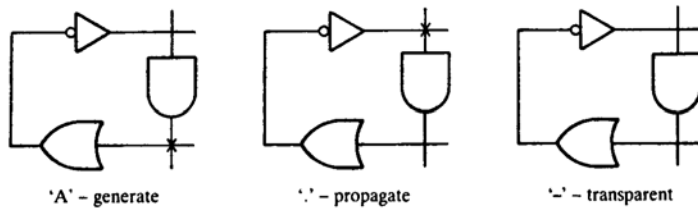
Para que el circuito funcione de forma correcta, todas las entradas en los flip-flop deben permanecer estables el tiempo de set-up mínimo del flip-flop, T_{SU} , antes del siguiente flanco activo de reloj. Si una de las entradas cambia tras este umbral de tiempo, el próximo estado o salida síncrona puede ser almacenada incorrectamente y el circuito falla. Esto se denomina “*carrera de transición*”. Para evitarlo, el periodo de reloj, T_P debe ser mayor que la suma de T_{SU} , T_{CLK} y T_L . Esto determina el periodo mínimo de reloj y por tanto, la máxima frecuencia de reloj del circuito.

Las salidas asíncronas también tendrán un periodo inestable en cada ciclo, como resultado de los distintos retardos de propagación de los biestables de estado y de la lógica. Incluso si un bit particular de salida no debe cambiar en una transición de estado, puede ocurrir

un “glitch” debido a un azar en la lógica. Si las salidas no deben tener glitches ni transiciones múltiples, la única solución es resincronizarla con un registro de salida, en otras palabras, convertirlas en salidas síncronas, como vimos en el tema 1.

Matriz Complemento

La matriz complemento es una característica única de los secuenciadores basados en PLA. Es una línea OR extra que puede tener entradas de cualquier conjunto de productos, que es complementada y realimentada como una línea de entrada adicional a los productos. Esta línea de valdrá uno lógico si ninguno de los términos producto conectados en la línea OR es uno. Tiene tres posibles configuraciones para cada línea de término producto. Esta información se añade para cada línea al final de la tabla de programación.



Para ver como se usa, veamos algunos ejemplos. Cada diseño de máquina de estados debe tener alguna forma de escapar de **estados ilegales**. Estos estados vienen definidos por combinaciones no usadas de los bits de estado. Si por

cualquier razón el secuenciador entra en uno de estos estados debe existir una transición a un estado conocido. Esto lo realizamos con la matriz complemento. Cuando lleguemos a este estado no habrá ningún producto activo, porque será un estado sin salida. Esta condición se detecta conectando cada término producto usado a la línea OR de la matriz complemento y realimentando el complemento a un nuevo término producto que defina la transición al estado conocido.

La penalización por usar la matriz complementario es un retardo de propagación mayor y reducir de esta forma la máxima frecuencia de reloj útil. La razón es que se necesitan dos pasos a través de la matriz antes de que los productos con entradas en la matriz complemento se hagan estables.

El comportamiento de una máquina síncrona se puede especificar mediante una tabla de definición de PLA (o PLS), que constituye también una definición directa de un secuenciador. Los parámetros de temporización de la PLA de un secuenciador y sus biestables definen la máxima frecuencia de reloj de trabajo. Los diagramas de flujo de la máquina de estados o sus equivalentes constituyen una forma muy conveniente de especificar el comportamiento de un secuenciador. El uso de biestables S-R o J-K permiten ahorrar términos producto. Las entradas asíncronas provocan problemas que se pueden solucionar parcialmente mediante asignaciones de estado. La mejor forma de solucionar estos problemas consiste en sincronizar la entrada con la señal de reloj, colocando entradas registradas.

Cuando el número de productos excede la capacidad del PLD, se debe realizar minimización manual mediante combinación de términos producto. La matriz complemento, presente en algunos secuenciadores, puede reducir el número de términos producto disminuyendo como contrapartida la máxima frecuencia útil de funcionamiento.

6.2.- Arquitectura AND programable OR fija. (PAL)

Aunque los secuenciadores basados en PLA son los dispositivos más versátiles para usar en aplicaciones de máquinas de estados, hay otras circunstancias que hacen que la arquitectura PAL no sea tan difícil de usar y constituya la solución más económica. Por ejemplo, se pueden implementar de forma económica con PAL pequeñas máquinas de estado que necesiten funcionar a frecuencias de reloj muy altas. También, la mayoría de los dispositivos PLD de mayor densidad tienen arquitectura PAL, disponiendo de términos producto dedicados a salidas particulares o a registros internos.

Hay dos tipos de dispositivos PAL: los de **salida registrada**, que disponen de matriz AND y salidas registradas, donde cada salida de la PAL se conecta a la entrada de un registro. Tienen la nomenclatura “número de entradas R número de salidas” Ej: 20R4.

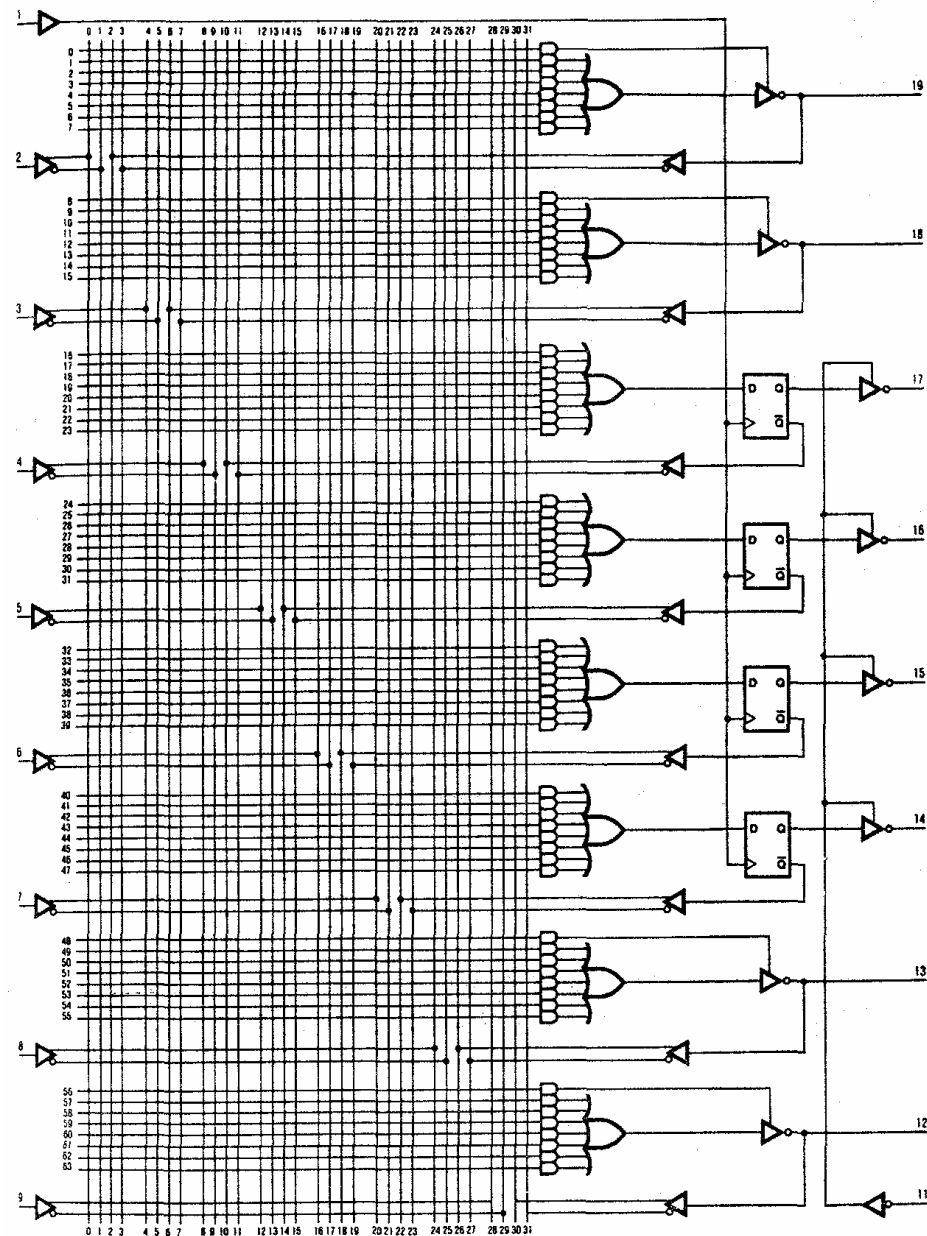
Por otro lado están los **versátiles** que disponen de macroceldas, en estos un grupo de lógica junto con uno o varios biestables forman un bloque denominado macrocelda, con una entrada y una salida, tienen la nomenclatura “número de entradas V número de salidas” Ej: 20V8.

6.2.1.- Matrices lógicas AND programables (PAL's) con capacidad secuencial.

Con dispositivos PAL, la transformación simple de diagrama de flujo a tabla PLS ya no funciona, normalmente se hace necesaria una etapa intermedia de diseño lógico. Esto se debe a que los términos producto para cada salida deben ser designados de forma separada, puesto que no se pueden compartir como vimos en el tema 2.

Generalmente se recurre a sistemas de programación de PLD, que a partir del diagrama de estados, realizan la simplificación de estados, la asignación de estados y la reducción de las tablas, obteniendo a su salida la tabla de programación del dispositivo. Estas herramientas están ampliamente extendidas, como ejemplos tenemos el PALASM, ORCAD PLD, ABEL, CU-PL, AHDL, etc.

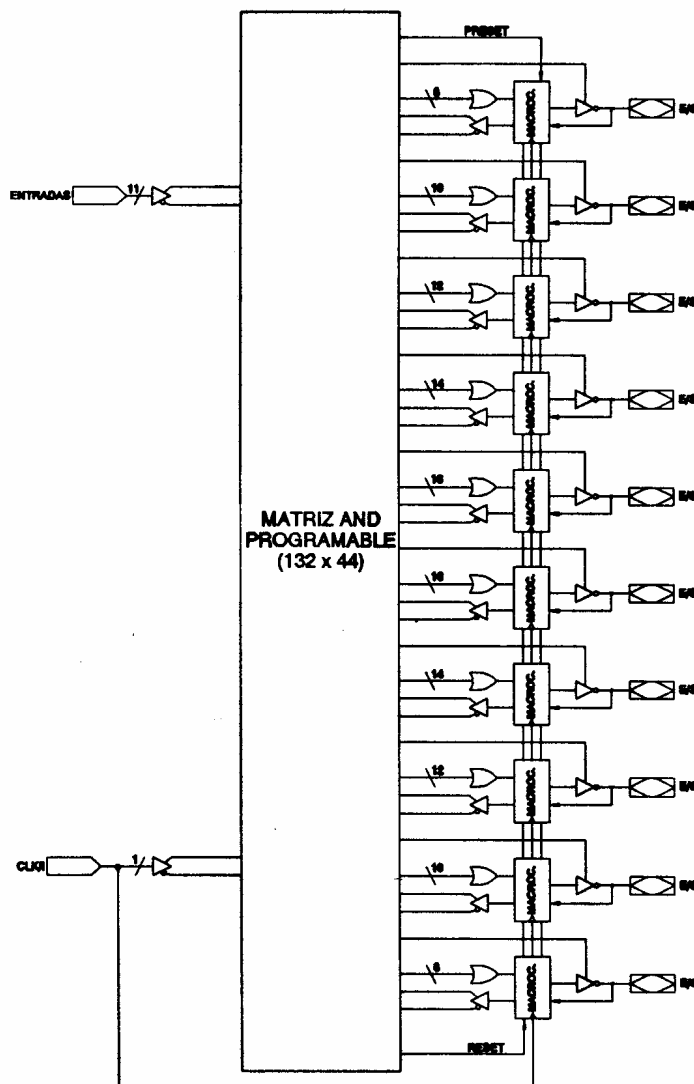
Como ejemplo tenemos la PAL 16R4, que dispone de 16 entradas, 4 salidas combinacionales y 4 registradas.



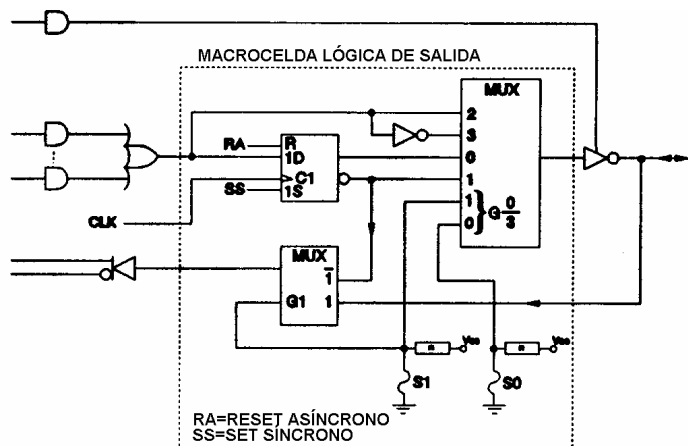
6.2.2.- Dispositivos lógicos programables y borrables (EPLD's) basados en macroceldas.

La arquitectura de una matriz lógica genérica (GAL) se caracteriza, al igual que la de las PALs, por la utilización de una matriz AND programable que alimenta una matriz OR fija, pero en cada una de sus patillas de salida se encuentra una macrocélula lógica que puede ser configurada por el diseñador para implementar la función deseada.

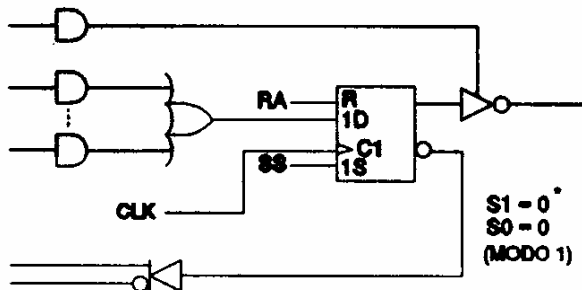
Este tipo de PLD se construye usando tecnología EECMOS y entre sus principales características podemos citar la rapidez (entre 5 y 15 ns) y bajo consumo (30 a 60 mA), siendo su característica más sobresaliente su capacidad de reprogramación tras haber sido borrada eléctricamente, garantizando el fabricante un mínimo de 100 ciclos de borrado/escritura con una retención de la información por encima de los 20 años. Fue AMD en 1983 quien introdujo la GAL más popular: la **22V10** que equivale a 300 o 400 puertas simples y es capaz de sustituir 10 C.I. de lógica discreta. Posteriormente se han mejorado sus características hasta conseguir muy bajo consumo y una frecuencia máxima de reloj de 150 MHz.



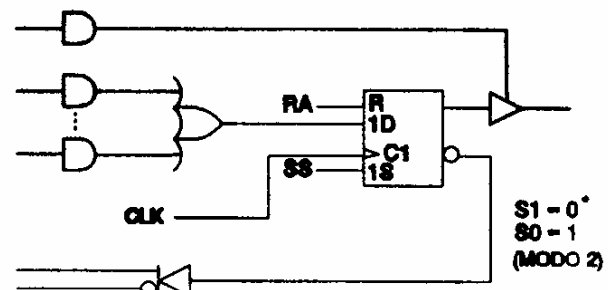
La GAL 22V10 dispone de 12 entradas y 10 macroceldas de E/S. La macrocelda permite una de cuatro configuraciones de salida potenciales: salida registrada o combinacional, activa a nivel alto o activa a nivel bajo. La configuración de salida se determina mediante dos bits que controlan dos multiplexores en cada macrocelda. Cada salida de macrocelda puede ser una patilla de entrada o de salida.



- **Configuración con salida registrada.** Cada macrocelda incluye un biestable de tipo D para almacenamiento y sincronización. El biestable se carga en la transición bajo a alto de la señal de reloj. En la configuración registrada ($S1=0$), la realimentación hacia la matriz AND se obtiene de la salida negada del flip-flop.

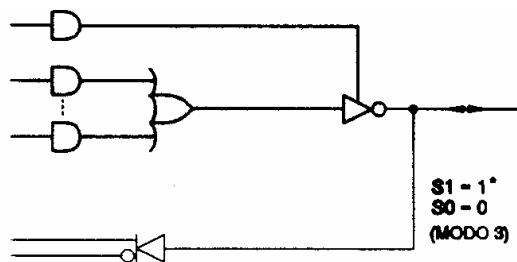


SALIDA REGISTRADA ACTIVA A NIVEL BAJO
REALIMENTACIÓN REGISTRADA

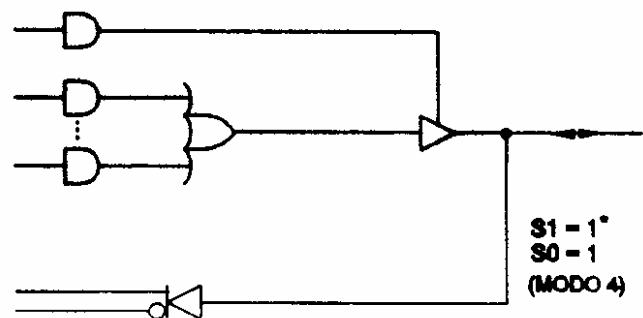


SALIDA REGISTRADA ACTIVA A NIVEL ALTO
REALIMENTACIÓN REGISTRADA

- **Configuración de entrada/salida combinacional.** Cualquier macrocelda se puede configurar como combinacional seleccionando el camino del multiplexor que evita el biestable ($S1=1$). En esta configuración la realimentación se obtiene de la patilla de salida.



SALIDA COMBINACIONAL ACTIVA A NIVEL BAJO
REALIMENTACIÓN DESDE E/S



SALIDA COMBINACIONAL ACTIVA A NIVEL ALTO
REALIMENTACIÓN DESDE E/S

• 0 = FUSIBLE INTACTO
1 = FUSIBLE ELIMINADO

Cada salida dispone de un buffer con salida triestado con una señal de control del triestado. Un término producto controla el buffer, permitiendo que la habilitación sea función cualquier producto de entradas del dispositivo o señales de realimentación. Las salidas combinacionales proporcionan una patilla de entrada/salida bidireccional, que puede configurarse como entrada dedicada si el buffer se encuentra siempre deshabilitado.

La polaridad de cada macrocelda de salida puede ser activa a nivel bajo o a nivel alto, la que se necesite en la salida o la más adecuada para reducir términos producto. La polaridad programable permite que las expresiones booleanas se escriban en su forma más compacta (real o invertida), y la salida puede ser todavía de la polaridad deseada. La selección se controla mediante un bit programable, en la salida y afecta a las salidas combinacional y registrada.

Para la inicialización, este dispositivo dispone de términos producto adicionales de preset y de reset. Estos términos están conectados a todas las salidas registradas. Cuando el término producto SP (Preset síncrono) pasa a nivel alto, los registros de salida se cargan con un nivel alto en la próxima transición de la señal de reloj. Cuando el término producto AR (Reset asíncrono) pasa a nivel alto, los registros de salida se cargan inmediatamente con un 0 independientemente de la señal de reloj. Curiosamente una vez conectada la alimentación al circuito todos los biestables cambian sus salidas a 0 para conseguir un estado predecible de

comienzo. Sin embargo, las salidas en las patillas dependerán de la polaridad de salida programada.

6.3.- Fases del diseño asistido por ordenador de sistemas secuenciales basados en PLD's.

Las fases son muy similares al diseño de sistemas combinacionales estudiado anteriormente. La especificación de máquinas secuenciales en ORCAD PLD se puede realizar de tres formas:

- Mediante una **tabla de estados** que codifica la información obtenida a partir del diagrama de estados.
- Con la descripción de la secuencia de estados o **cuentas** por las que pasa el dispositivo y cuando tiene lugar en función de las entradas.
- Mediante la especificación de la **secuencia** que debe seguir la salida. Se emplea para especificar generadores de secuencia.

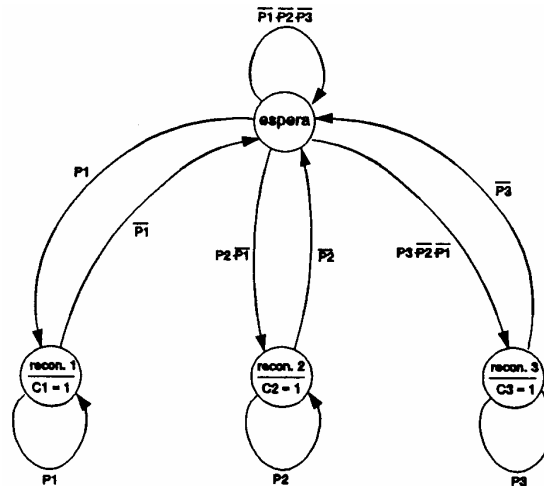
Realizaremos un ejemplo de sistema síncrono para ver como se implementa. Un arbitrador de recursos se encarga de conceder un determinado recurso en función de las peticiones que tenga del recurso. El funcionamiento será el siguiente:

1. Esperar a que una de las líneas de petición (P1, P2, P3) sea activa.
2. Si la línea de petición P1 está activa, activar la línea de concesión C1 mientras P1 permanezca activa. Entonces volver al paso 1.
3. Si la línea de petición P2 está activa y P1 inactiva, activar la línea de concesión C2 mientras P2 permanezca activa. Entonces volver al paso 1.
4. Si la línea de petición P3 está activa y P1 y P2 inactivas, activar la línea de concesión C3 mientras P3 permanezca activa. Entonces volver al paso 1.

El diagrama de bloques será el siguiente:



El diagrama de estados será el siguiente:



El fichero fuente para crear el arbitrador usando PROTEL CUPL mediante tabla de estado será el siguiente:

```

Name                      Arbitrador;
Partno                    P9000183;
Date                      06/11/99;
Revision                  01;
Designer                  Manuel Sanchez;
Company                   UHU;
Assembly                  PC Memory;
Location                  U106;

/** Inputs **/
Pin [4..6] = [P1..3];    /* Entradas de peticion */
Pin 9      = reset;      /* Reset */

/** Outputs **/
Pin [23..27] = [C1..3,Q0..1]; /* Salidas de concesion */
/* Codificacion de estados */
$define ESPERA 0
$define RECON_1 1
$define RECON_2 2
$define RECON_3 3

SEQUENCE [Q0..1] {

/* Espera peticion P1, P2 o P3 en este orden de prioridad */
PRESENT ESPERA IF P1 NEXT RECON_1 OUT C1;
      IF P2 NEXT RECON_2 OUT C2;
      IF P3 NEXT RECON_3 OUT C3;
      NEXT ESPERA OUT !C1 OUT !C2 OUT !C3;
/* Concede recurso a peticion 1 (P1) y lo mantiene asi hasta que
P1 deja de solicitarlo */
PRESENT RECON_1 IF P1 NEXT RECON_1 OUT C1;
      NEXT ESPERA OUT !C1 OUT !C2 OUT !C3;
/* Concede recurso a peticion 2 (P2) y lo mantiene asi hasta que
P2 deja de solicitarlo */
PRESENT RECON_2 IF P2 NEXT RECON_2 OUT C2;
      NEXT ESPERA OUT !C1 OUT !C2 OUT !C3;
/* Concede recurso a peticion 3 (P3) y lo mantiene asi hasta que
P3 deja de solicitarlo */
PRESENT RECON_3 IF P3 NEXT RECON_3 OUT C3;
      NEXT ESPERA OUT !C1 OUT !C2 OUT !C3;
}

```

Una vez compilado el fichero, el compilador realiza la reducción de las ecuaciones de excitación de las salidas registradas del dispositivo y nos lo ofrece en un completo informe.

```
*****
                                Arbitrador
*****
ADVANCED PLD      4.0 Serial# MW-67999999
Device            g22v10lcc  Library DLIB-h-36-2
Created           sáb nov 06 18.57.42 1999
Name              Arbitrador
Partno            P9000183
Revision          01
Date              06/11/99
Designer          Manuel Sanchez
Company           UHU
Assembly          PC Memory
Location          U106
=====
                                Expanded Product Terms
=====
C1.d  =>
      P1 & !Q1
C2.d  =>
      P2 & !Q0
C3.d  =>
      P3 & !Q0 & !Q1
      # P3 & Q0 & Q1
Q0.d  =>
      P1 & !Q1
      # P3 & Q0 & Q1
      # P3 & !Q0 & !Q1
Q1.d  =>
      P3 & Q0 & Q1
      # P2 & !Q0
      # P3 & !Q0 & !Q1
C1.oe =>
      1
C2.oe =>
      1
C3.oe =>
      1
Q0.oe =>
      1
Q1.oe =>
      1
=====
                                Symbol Table
=====
```

Pin	Variable	Ext	Pin	Type	Pterms Used	Max Pterms	Min Level
Pol	Name	---	---	----	-----	-----	-----
	C1		23	V	-	-	-
	C1	d	23	X	1	16	1
	C2		24	V	-	-	-
	C2	d	24	X	1	14	1
	C3		25	V	-	-	-
	C3	d	25	X	2	12	1
	P1		4	V	-	-	-
	P2		5	V	-	-	-
	P3		6	V	-	-	-

Q0		26	V	-	-	-
Q0	d	26	X	3	10	1
Q1		27	V	-	-	-
Q1	d	27	X	3	8	1
reset		9	V	-	-	-
C1	oe	23	D	1	1	0
C2	oe	24	D	1	1	0
C3	oe	25	D	1	1	0
Q0	oe	26	D	1	1	0
Q1	oe	27	D	1	1	0

LEGEND D : default variable F : field G : group
 I : intermediate variable N : node M : extended node
 U : undefined V : variable X : extended variable
 T : function

=====

Fuse Plot

=====

SP

05764 xxx

AR

00000 xxx

Pin #27 05808 Mode -x

00044 -----

00088 ---x---x-----x-----

00132 -----x-----x-----

00176 --x---x-----x-----

00220 xxx

00264 xxx

00308 xxx

00352 xxx

00396 xxx

00440 -----

00484 --x-----x-----

00528 ---x---x-----x-----

00572 --x---x-----x-----

00616 xxx

00660 xxx

00704 xxx

00748 xxx

Pin #26 05810 Mode -x

00792 xxx

00836 xxx

00880 xxx

00924 -----

00968 --x---x-----x-----

01012 ---x---x-----x-----

01056 xxx

01100 xxx

01144 xxx

01188 xxx

01232 xxx

01276 xxx

01320 xxx

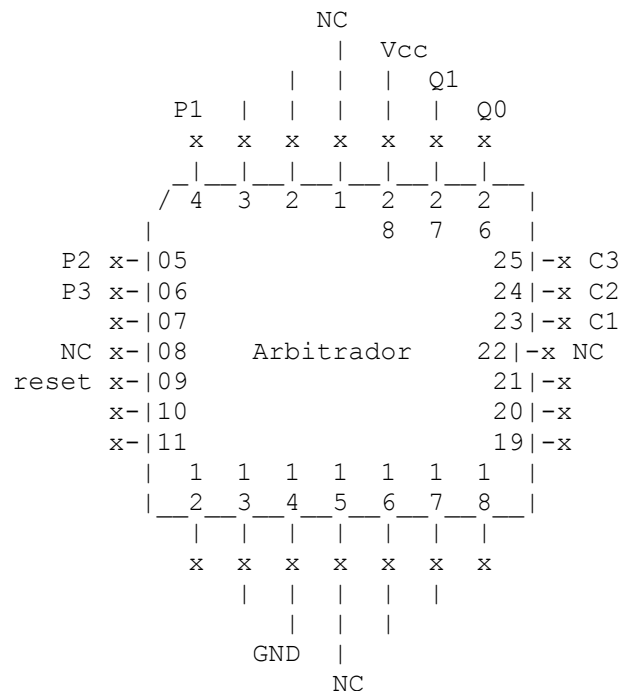
01364 xxx

01408 xxx

LEGEND X : fuse not blown

- : fuse blown

Chip Diagram



7.- Dispositivos programables avanzados CPLD.

A finales de la década de los 80, las macroceldas fueron complicando sus funciones y los CIs aumentaron su número de patillas, siendo capaces de implementar hasta 2 o 3 PALs del tipo 22V10. Nacieron los CPLDs (Complex Programmable Logic Device), dispositivos que cubren el vacío existente entre la integración a nivel de puertas conseguida por la lógica programable simple (alrededor de 250 puertas) y la integración que es capaz de conseguirse mediante una FPGA (Field Programmable Gate Array) que es de alrededor de 5000 como mínimo hasta 500.000 puertas equivalentes.

La razón de ser de estos componentes está en la complejidad de los diseños digitales actuales, que supone la sustitución de numerosos componentes de lógica discreta, tarea que no puede ser realizada por la lógica programable simple. Básicamente, la arquitectura interna de un CPLD está formada por bloques de control de entrada y salida conectados a bloques de macroceldas (formados internamente por una red de puertas AND y, al menos, un registro programable con funciones de CLEAR y RESET) que adoptan distintas denominaciones, según el fabricante:

- Bloques de redes lógicas o LAB (Logic Array Blocks), para Altera.
- Bloques de PAL o PAL Blocks, para AMD.
- Bloques globales de lógica o GLB (Global Logic Blocks), para el fabricante Lattice.

A su vez, estos bloques de macroceldas están conectados entre sí a través de una matriz de interconexión programable que, al igual que los bloques de macroceldas, adopta un nombre diferente, según el fabricante:

- PIA (Programmable Interconnect Array), para el fabricante Altera.
- SM (Switch Matrix) y CSM (Central Switch Matrix), para AMD.
- GRP (Global Routing Pool) y ORP (Output Routing Pool) para el fabricante Lattice.

A comienzos de los 90, AMD lanzó al mercado su familia MACH (Macro Array CMOS High-speed/density) compuesta por una serie de dispositivos que combinan tiempos de propagación que varían entre los 10 y 20 ns, con un nivel de integración que oscila entre la 900 puertas y 32 macroceldas de la MACH110 a las 10000 puertas y 256 macroceldas que ofrece la actual MACH465. Y todo ello, a un coste incluso inferior al número equivalente de 22V10 a las que son capaces de sustituir.

Pero es Altera, fabricante pionero de EPLDs, el que ha lanzado al mercado los CPLDs más populares agrupados en tres familias:

- La **familia MAX5000**, con capacidad de hasta 3.800 puertas, 100 patillas como máximo y hasta 12 ns y 50-70 MHz.
- La **familia MAX7000**, con capacidad de hasta 5.000 puertas, frecuencias de 125 MHz y retardos de 7,5 ns. Los CIs pertenecientes a esta familia utilizan una matriz de interconexión mejorada (EPIA).
- La **familia FLEX8000**, con capacidad de hasta 24.000 puertas, formada por dispositivos que combinan el alto número de registros presentes en las FPGA con la rapidez y arquitectura de vías predefinidas de los CPLDs.

Lattice también fabrica CPLDs agrupadas en tres familias: la **pLS/ispLSI1000** capaz de integrar un máximo de unas 4000 puertas, la pLSI/ispLSI2000 con una frecuencia de funcionamiento interno de 135 MHz y 7,5 ns de tiempo de propagación y la pLSI/ispLSI3000 capaz de integrar hasta 14.000 puertas.

En cuanto a las herramientas de programación, para los CPLDs y FPGAs, el software es dependiente del fabricante (se dice que es propietario del fabricante), es distinto y suele ser costoso. Además, debido al elevado y variable número de patillas de los dispositivos, los programadores sólo sirven para un número determinado de dichos dispositivos.

Actualmente, los diseñadores de lógica basada en CPLDs y FPGAs están de acuerdo en que la variedad de herramientas propietarias de cada fabricante constituyen un impedimento frente a las de tipo universal, capaces de integrar adaptadores o módulos específicos para cada arquitectura. De estas últimas existen pocas, aunque cada vez aparecen más aunque con la oposición de los grandes fabricantes.

Dispositivos clásicos frente a nuevos dispositivos

Los dispositivos clásicos suelen utilizar tecnología bipolar (TTL).

Ventajas :

- Son rápidos.
- Son económicos.

- Fáciles de programar.

Inconvenientes :

- Gran consumo de corriente.
- Una vez programados, no puede alterarse su programación.
- Los fusibles ocupan mucha área de silicio.

En los nuevos dispositivos se emplea principalmente tecnología CMOS con celdas EPROM (EPLD: Erasable Programmable Logic Devices) y EEPROM (dispositivos EEPLD = Electrically Erasable Programmable Logic Devices)

Ventajas:

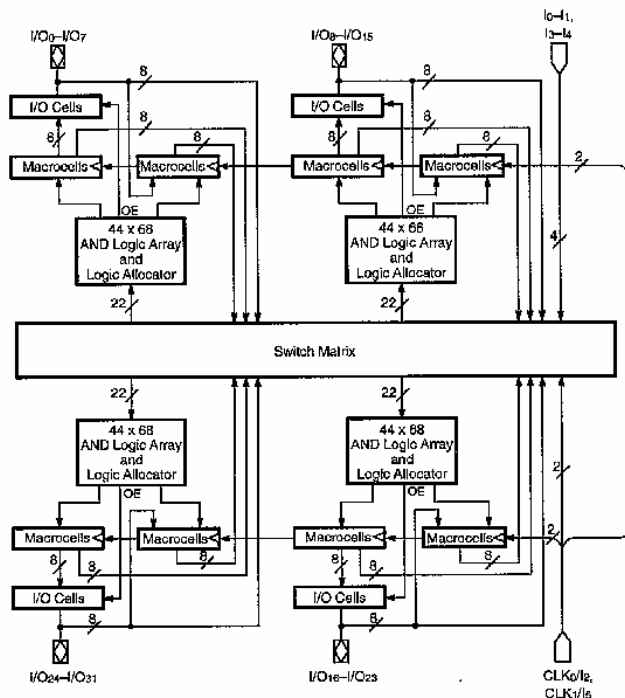
- Menor consumo.
- Las celdas EPROM y EEPROM ocupan menos área de silicio que los fusibles bipolares, pudiendo aumentarse el nivel de integración de los dispositivos.
- Los dispositivos son reprogramables.
- Los dispositivos pueden ser comprobados y testeados más fácilmente por el fabricante.

Inconvenientes:

- Mas caros.
- Más lentos.
- Usan herramientas de programación propietarias del fabricante

7.1.- Dispositivo EEPL de AMD.

Como ejemplo tenemos el dispositivo MACH 210 de la familia, este tiene aproximadamente seis veces la capacidad de macroceldas que la 22V10. Consiste en cuatro



bloques PAL interconectados por una matriz de conmutación. Los cuatro bloques PAL son esencialmente estructuras 22V16 con matrices de términos productos y macroceldas programables, incluyendo macroceldas sin salida al exterior. La matriz de interconexión conecta los bloques PAL entre ellos y a las patillas de entrada. Este dispositivo tiene dos tipos de macroceldas, de salida y sin salida externa ("buried"). Las macroceldas dedicadas sin salida externa proporcionan registros de entrada para su uso en sincronización de señales de entrada.

Cada bloque PAL en la MACH 210, contiene una matriz lógica de 64

términos producto, un planificador lógico, 8 macroceldas de salida, 8 macroceldas “buried” y 8 celdas de E/S. La matriz de conmutación alimenta cada bloque de la PAL con 22 entradas. Además de estos términos producto, dos términos producto controlan la habilitación de salida, un término el reset asíncrono y otro el preset asíncrono.

La matriz de conmutación se alimenta de las entradas y de las señales de realimentación de los bloques PAL. Cada bloque PAL proporciona 16 señales de realimentación internas y 8 señales de realimentación de E/S. La matriz de conmutación distribuye estas señales a los bloques PAL de una manera eficiente. El software de diseño configura de forma automática la matriz de conmutación cuanto se adapta el diseño al dispositivo.

El planificador lógico toma 64 términos producto y los organiza en las 16 macroceldas como sea conveniente. Cada macrocelda puede ser alimentada hasta con 16 términos productos. Esto también lo controla el software de diseño.

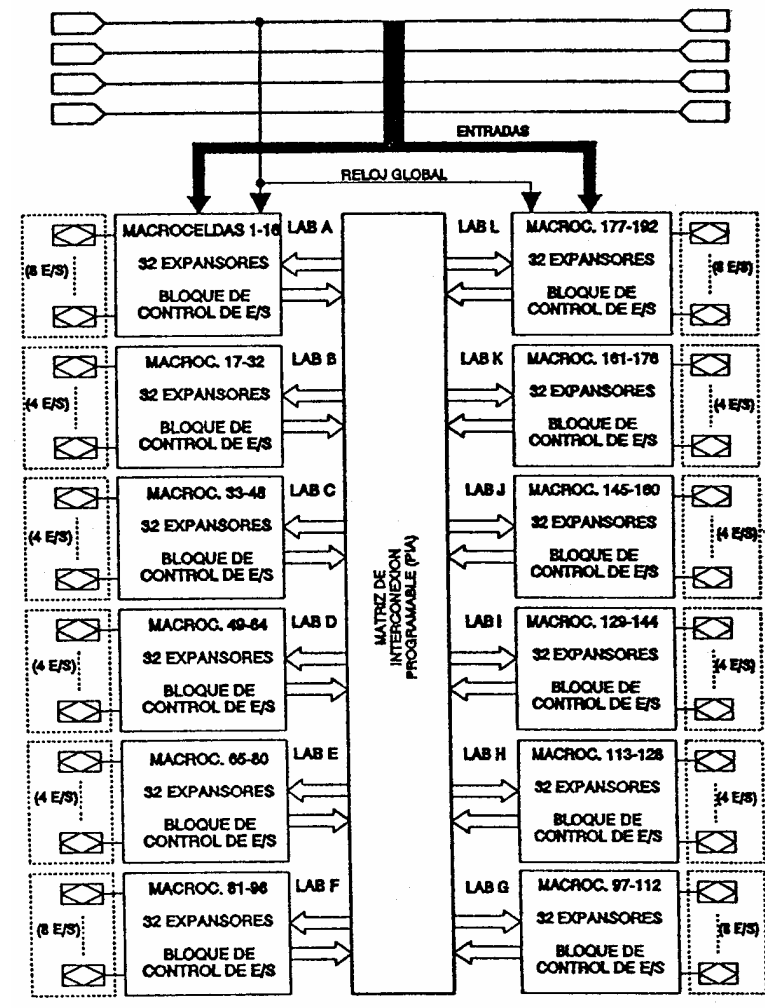
7.2.- Dispositivo EPLD de ALTERA.

Altera dispone de tres familias, la MAX5000, la MAX7000 y la MAX9000. La familia MAX 5000 son EPLDs con densidades de 32 a 192 macroceldas (600 a 3750 puertas útiles) y retardos de patilla a patilla de hasta 15 ns, y frecuencias de conteo de hasta 80 MHz. Pueden usarse para diseños desde decodificadores de direcciones de 28 patillas a periféricos a medida de más de 100 patillas. Por ejemplo, el dispositivo EPM5192 puede reemplazar unos 100 integrados de la serie TTL, pudiendo integrar subsistemas completos en un solo encapsulado, reduciendo el área de placa de circuito y el consumo de corriente.

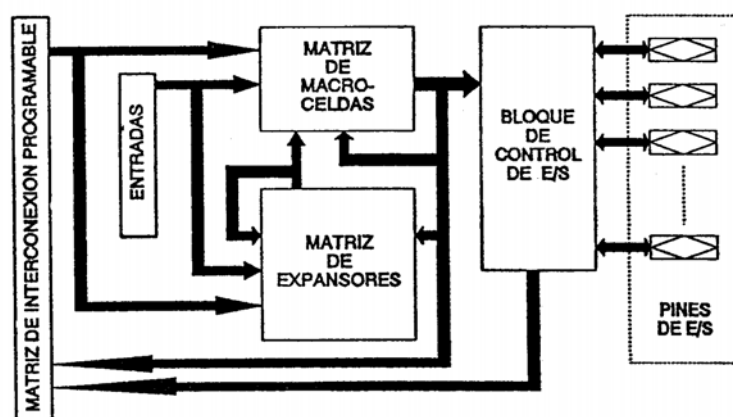
La familia MAX5000 dispone de entre 32 y 192 macroceldas que se combinan en grupos denominados bloques de lógica matricial (LABs). Cada macrocelda dispone de una matriz programable AND, fija OR y un registro configurable que funciona como D, T, JK o SR con funciones programables independientes de reloj, puesta a cero y puesta a uno. Para construir funciones lógicas complejas, a cada macrocelda se la puede ampliar con un expansor comunitario de términos productos (“shareable expanders”) para proporcionar más de 32 términos producto por macrocelda.

Las macroceldas constituyen el principal recurso para la implementación lógica. De los expansores se extrae capacidad lógica adicional, que se pueden usar para aumentar los términos producto de entrada de cualquier macrocelda. La matriz de expansión de términos producto consiste en un grupo de términos producto invertidos que pueden usarse y compartirse por todas las macroceldas del LAB para crear lógica combinacional o registrada.

Estas macroceldas flexibles y expansores compartidos facilitan los diseños con términos producto variables frente a las arquitecturas con términos producto fijos. Todas las salidas de las macroceldas se han rutado globalmente hacia un LAB mediante interconexiones de LAB. Las salidas también alimentan el bloque de control de entrada/salida que consiste en grupos de buffers triestado programables y patillas de E/S. Varios LABs se encuentran conectados mediante un PIA (matriz de interconexión programable). Todas las macroceldas alimentan el PIA para proporcionar rutado a diseños con alto fan-in.

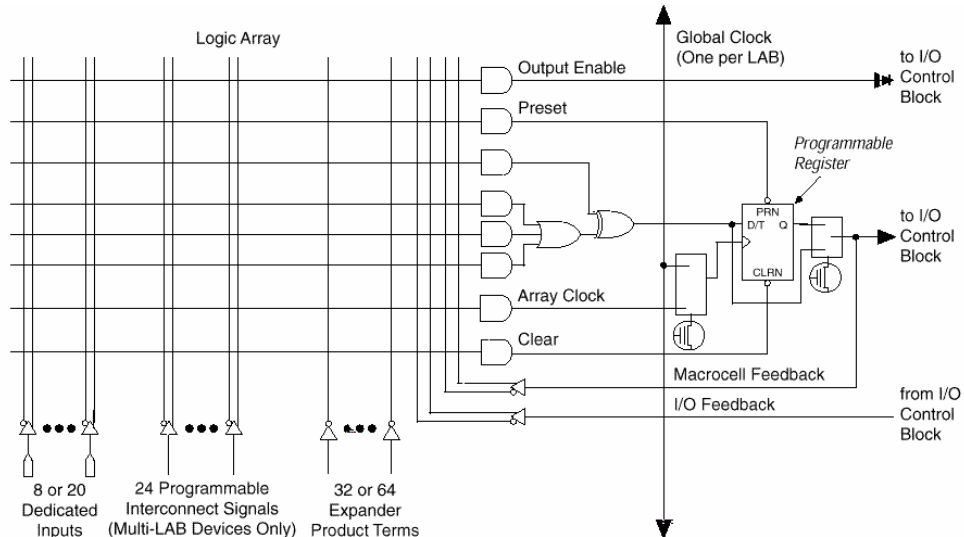


La EPM5192 contiene 5 LABs. Cada LAB consiste en una matriz de macroceldas y una matriz de expansores de términos productos.



La macrocelda de la familia MAX5000 consiste en una matriz lógica programable y un registro configurable de forma independiente. Este puede programarse para emular funcionamiento como D, T, JK o SR, como cerrojo transparente o puentado por una operación combinacional.

La lógica combinacional se implementa mediante una matriz lógica programable, en la que tres términos producto que se agrupan por una puerta OR alimenta la entrada de una puerta XOR. La segunda entrada de la puerta XOR se usa para funciones de lógica aritmética XOR compleja y para inversión de funciones “De Morgan”. La salida de la puerta XOR alimenta el registro programable o se puentea cuando interesa operación combinacional.

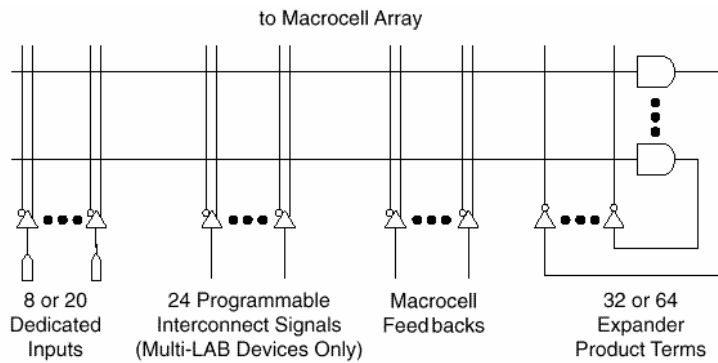


Disponemos de términos producto adicionales (denominados “secundarios”) que se emplean para controlar las señales output enable, preset, clear y reloj. Los términos producto de preset y clear generan el preset asíncrono activo a nivel bajo y el clear asíncrono del flip-flop. El término producto de reloj permite que cada registro tenga un reloj independiente. Las macroceldas que generan un pin de salida pueden usar el término producto de habilitación de salida para controlar el buffer triestado activo a nivel alto en el bloque de control de E/S.

Cada LAB soporta **reloj global** o **reloj de matriz**. El reloj global lo proporciona una señal de reloj dedicada (CLK) que ofrece tiempos de retardo muy cortos. Debido a que cada LAB dispone de un reloj global, todos los relojes de los biestables dentro del LAB pueden ser activos por flanco positivo a partir de la patilla CLK. Si esta patilla no se usa como reloj global, puede usarse como una entrada dedicada de alta velocidad.

En el modo de reloj de matriz, cada flip-flop obtiene la señal de reloj mediante un término producto. Cualquier patilla de entrada o lógica interna se puede usar como fuente de reloj. Esto permite configurar cada flip-flop como activo por flanco positivo o negativo, aumentando la flexibilidad. De esta forma se pueden integrar sistemas que precisen múltiples señales de reloj. Cada flip-flop en un LAB puede disponer de una señal de reloj generada de forma diferente, sin embargo, no se pueden mezclar estos modos de relojado en el mismo LAB.

Aunque la mayoría de las funciones lógicas se pueden implementar con los términos producto disponibles en cada macrocelda, algunas funciones lógicas son más complejas y requieren términos producto adicionales. Si bien, se pueden usar macroceldas adicionales para añadir los recursos necesarios, la arquitectura del MAX5000 permite usar el expansor compartido para proporcionar términos producto adicionales a cualquier macrocelda del mismo LAB. Cada LAB dispone de 32 expansores compartidos que proporcionan la flexibilidad para implementar diseños que necesiten gran cantidad de términos producto.

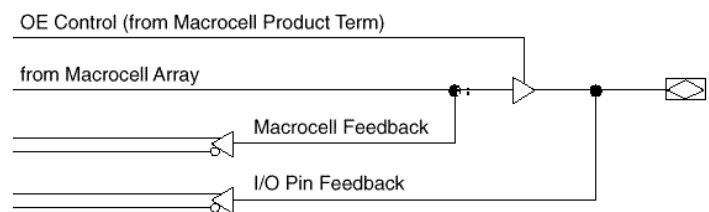


Los expansores están alimentados por todas las señales del LAB. Un expansor puede alimentar todas las macroceldas del LAB o múltiples términos producto en la misma macrocelda. Puesto que los expansores también alimentan los términos productos “secundarios” de cada macrocelda, se pueden implementar funciones lógicas complejas sin usar macroceldas adicionales. Se incurre en un pequeño retardo (t_{SEXP}) cuando se usan los expansores compartidos.

Estos dispositivos usan una matriz de interconexión programable (PIA) para rutear señales entre varios LABs. El PIA, que viene alimentado por todas las macroceldas y patillas de E/S, conecta solo las señales necesarias para implementar lógica dentro del LAB. Mientras los retardos de esquemas de rutado segmentado en FPGAs son acumulativos, variables y dependientes del camino seguido, el PIA tiene un retardo fijo. El PIA elimina de esta forma desplazamientos entre señales y hace fácil la predicción de temporizaciones dentro del dispositivo.

Cada LAB dispone de un bloque de control de E/S que permite que cada patilla se pueda configurar de forma individual como entrada, salida, o bidireccional.

La arquitectura de la familia MAX5000 proporciona doble realimentación de E/S en la cual la realimentación de la macrocelda y de la patilla de E/S son independientes, aumentando la flexibilidad. Cuando la patilla de E/S se configura como entrada, la macrocelda asociada puede usarse como “buried”.

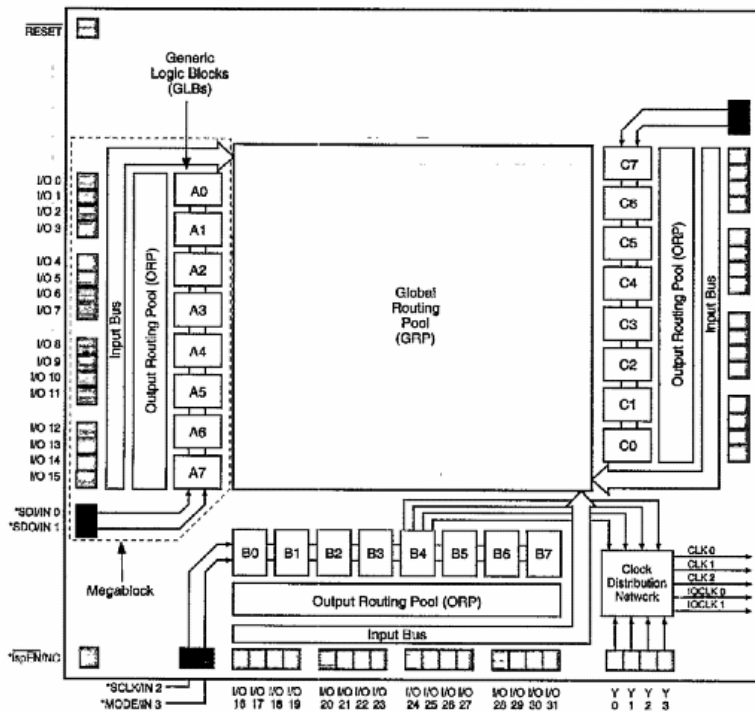


7.3.- Dispositivo EEPLD de LATTICE.

Los dispositivos lógicos programables pLSI y ispLSI 1024 son dispositivos lógicos programables de alta densidad que contienen 144 registros, 48 patillas de E/S universales, seis patillas de entrada dedicadas, cuatro entradas de reloj dedicadas y un bloque de rutado global (Global Routing Pool GRP). El GRP proporciona interconexión completa entre estos elementos. El ispLSI 1024 dispone de capacidad de programación en sistema con 5 voltios y de capacidades de diagnóstico. Es el primer dispositivo que ofrece reprogramabilidad de lógica no volátil en sistema, así como flexibilidad de interconexión para proporcionar sistemas realmente reconfigurables.

La unidad básica de lógica de estos dispositivos es el GLB (Generic Logic Block). Los GLBs se etiquetan como A0, A1, ..., C7. Hay un total de 24 GLBs en este dispositivo. Cada GLB tiene 18 entradas, una matriz programable AD/OR/XOR y cuatro salidas que pueden configurarse como combinacionales o registradas. Las entradas al GLB provienen de los GRP y

de las entradas dedicadas. Todas las salidas de los GLB se realimentan en el GRP para que puedan ser conectadas a las entradas de cualquier otro GLB del dispositivo.



El dispositivo también dispone de 48 celdas de E/S, cada una de las cuales se encuentra conectada directamente a una patilla de E/S. Cada celda de E/S puede programarse individualmente como entrada combinacional, entrada registrada, entrada con cerrojo transparente, salida o patilla bidireccional con control triestado. Además, todas las salidas disponen de polaridad programable. Los niveles de señal son voltajes compatibles TTL.

Se encuentran conectados ocho GLBs, 16 celdas de E/S, dos entradas dedicadas y un ORP (Output Routing Pool) conjuntamente para formar un “megabloque”. Las salidas de los ocho GLBs se conectan a un conjunto de 16 celdas universales de E/S por el GRP. Las celdas de E/S en el megabloque también comparten una señal común de habilitación de salida. Este dispositivo contiene tres **megabloques**.

El GRP tiene como entradas las salidas de todos los GLBs y todas las entradas de las celdas de E/S bidireccionales. Todas estas señales se encuentran disponibles a las entradas de los GLBs. El reloj en estos dispositivos se selecciona mediante la red de distribución de reloj. Cuatro patillas de reloj dedicadas (Y0, Y1, Y2 e Y3) se introducen en la red de distribución, y se proporcionan cinco salidas de reloj (CLK0, CLK1, CLK2, IOCLK0 e IOCLK1) para rutar la señal de reloj a los GLBs y a las celdas de E/S. La red de distribución de reloj también se puede usar con un reloj especial. La lógica del GLB permite al usuario crear un reloj interno a partir de una combinación de señales internas en el dispositivo.

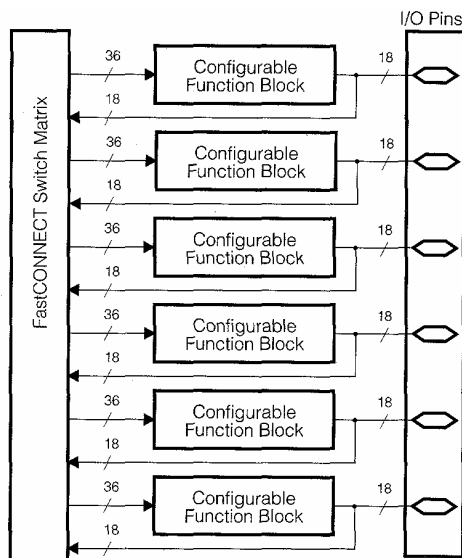
Estos dispositivos tienen la **capacidad ISP** (In System Programmable) para poder programar el dispositivo a través de tres líneas de datos en formato serie de forma rápida (del orden de milisegundos) independientemente de la conexión en el sistema en que se encuentre, permitiendo modificar su función incluso una vez diseñada la placa de circuito impreso y todo el sistema electrónico periférico al dispositivo. Los sistemas ISP reúnen las siguientes ventajas:

- Permiten diseñar el contenido lógico del dispositivo sobre la marcha y poder probarlo inmediatamente.
- Hacen posible optimizar y depurar los diseños realizados sin tener que quitar el dispositivo de su lugar en la placa.
- Posibilitan que el dispositivo pueda realizar varias funciones diferentes en un momento determinado, sin más que cambiar su programación interna.

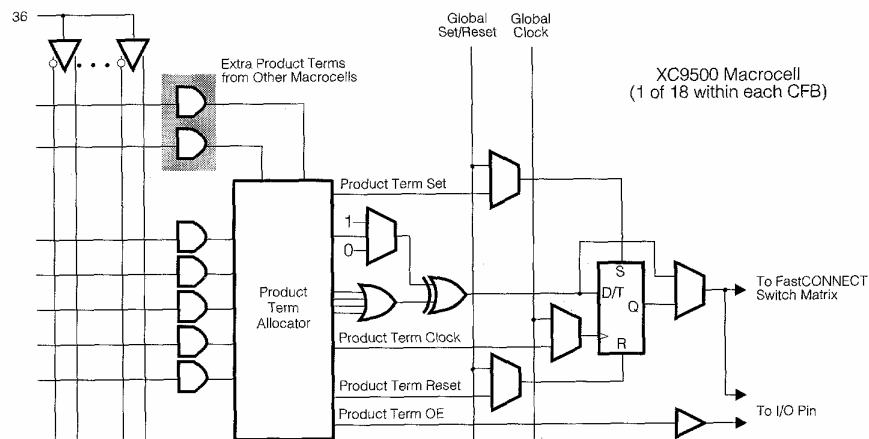
7.4.- Dispositivos CPLD de XILINX.

Xilinx dispone de las series XC7200 y XC7300, se trata de dispositivos OTP (One Time Programming, programables solo una vez) con una densidad de puertas baja-media y una estructura basada en tres megabloques junto con un interfaz de rutado. Solo se emplea en diseños de baja complejidad.

Pero la serie que más éxito tiene para este fabricante es la XC95000, constituida con tecnología Flash, permite ser regrabada directamente en la aplicación (ISP) cuantas veces sea necesario sin necesidad de programador especial. Por ejemplo, el XC95108 contiene seis bloques de funciones configurables (CFBs) equivalentes a una PAL de 18 macroceldas con 36 entradas y 18 salidas. Las salidas de las macroceldas se conectan a patillas de E/S, pero también se conectan a una matriz de interconexión global (XILINX la llama "Fast Connect Switch Matrix"). Cada CFB acepta 36 entradas de las $6 \times 18 = 108$ señales de realimentación que entran en la matriz de conmutación. Se pueden construir funciones lógicas multinivel muy complejas programando las funciones lógicas individuales de cada macrocelda de cada CFB y luego conectandolas a través de la matriz de conmutación. El resultado es un diseño donde cada patilla del CPLD proviene de una macrocelda que implementa una función lógica ampliada de una combinación de varias entradas.

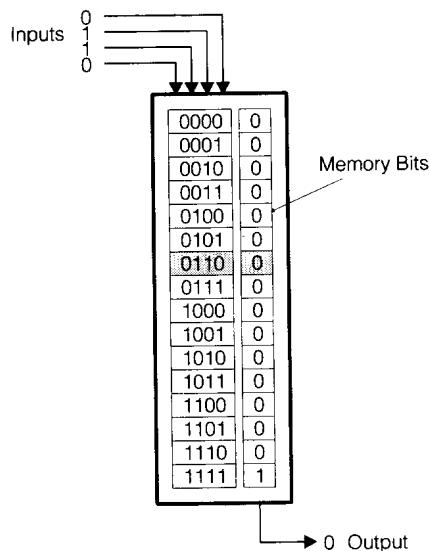


Cada conmutador de programación de estos dispositivos está controlado por un elemento de almacenamiento que guarda la configuración del conmutador asociado. Cambiando los valores de estos elementos de almacenamiento cambia el estado de los conmutadores y altera las funciones del dispositivo. Estos conmutadores pueden reprogramarse muchas veces para implementar nuevos diseños (o reparar diseños defectuosos), eliminando la necesidad de comprar un nuevo dispositivo para cada modificación del diseño. Las CPLDs XC9500 emplean celdas de almacenamiento basadas en Flash, por lo que el dispositivo mantiene su programación incluso si este se apaga.



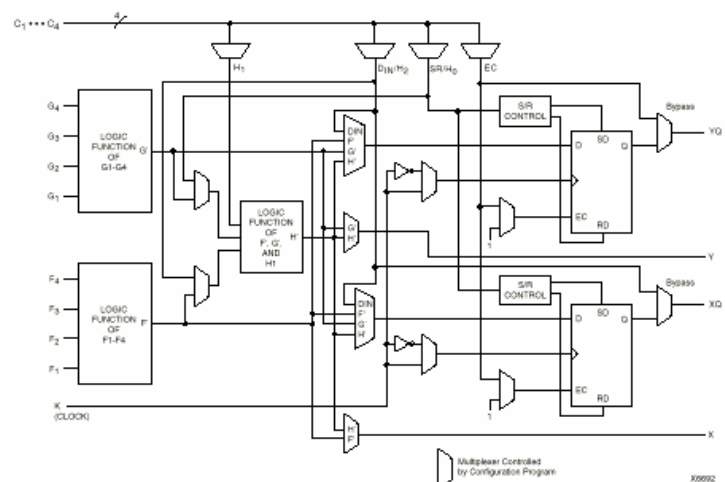
8.- Dispositivos FPGA de XILINX.

Las FPGA que usa este fabricante, como la mayoría disponibles en el mercado se basan en celdas de memoria RAM estática (biestables) para almacenar la información de configuración, por lo que si se interrumpe la alimentación pierden su programación. Xilinx dispone de varias familias de dispositivos FPGA, la más antigua es la XC3000 ahora ya en desuso y descatalogada. La familia más usada y experimentada de FPGAs de este fabricante la constituye la serie XC4000, que ha tenido varias actualizaciones: XC4000A, XC4000E, XC4000XL, etc. La serie XC5200 se impone como una versión barata de la 4000, siendo compatible con ésta pero contando con menor número de recursos en términos de puertas lógicas equivalentes. Por último los desarrollos más avanzados y modernos de este fabricante lo constituyen las series Spartan XCS40 y Virtex V3000 pudiendo estas últimas abarcar densidades de 300K a 1M puertas equivalentes, colocándose a la cabeza de máxima densidad disponible hoy día para estos dispositivos.



Estos dispositivos tienen una arquitectura alternativa a las CPLD. El bloque de construcción básico para las FPGA es la tabla de entradas (LUT o Lookup table). Una tabla LUT tiene solo cuatro entradas y una pequeña memoria que contiene 16 bits. Aplicando una combinación binaria determinada a las entradas (como 0110) entrara la dirección de un bit particular de la memoria y pondrá a la salida su valor. Se puede realizar cualquier función lógica de cuatro entradas programando la memoria de la tabla LUT con los bits apropiados. Por ejemplo, una puerta AND de cuatro entradas se construye cargando la memoria entera con bits a 0 excepto por un bit a 1 colocado en la celda que se activa cuando todas las entradas valen 1.

En las FPGAs de la serie XC4000, se combinan tres LUTs con dos flip-flops y alguna circuitería adicional de selección para formar un bloque de lógica configurable (CLB), como muestra la figura. Los CLBs se disponen en un array con matrices de conmutación programables (PSMs) entre los CLBs. Los PSM también se conocen con el nombre de cajas mágicas y son análogos a la matriz de conmutación FastCONNECT de la serie XC95000 y se usan para dirigir las salidas de CLBs vecinos a las entradas de otro CLB. Las patillas de E/S de la FPGA se pueden conectar a los PSMs o CLBs o se pueden conectar a su propia matriz de rutado. La mayoría de las FPGA disponen de más CLBs que patillas de E/S, por lo que cada CLB no puede tener una conexión directa al mundo exterior como las macroceldas de un CPLD. Por ejemplo, la FPGA



de XILINX XC4005XL tiene 196 CLBs en una matriz de 14 x 14, pero solo dispone de 64 patillas cuando se tiene en un encapsulado de plástico portador de chip (PLCC) de 84 patillas.

Anexo 1. Hojas de datos (Data Sheets) de componentes.

Algunos de los dispositivos secuenciales más usados en tecnología TTL:

a) Biestables:

7473: Doble flip-flop, JK (Maestro-Esclavo)
7474: Doble flip-flop, tipo D
74174: Séxtuple flip-flop tipo D

b) Registros:

7494, 74194, 74195: Registro de desplazamiento de 4 bits.
7491: Registro de desplazamiento de 8 bits.

c) Contadores:

7490: Contador decimal de 4 bits.
7493: Contador binario de 4 bits.
7492: Contador divisor por 12
74192: Contador bidireccional de 4 bits síncrono y programable.

d) Lógica secuencial programable:

GAL 22V10
MACH 210
MAX5000
ISP 1024
XC4000 Series

Flip-flop dual JK maestro/auxiliar con reloj y borrado separados

1 - SN 7473 N 4 - MC 7473 P 7 - ZN 7473 E 10 - FJJ 121 13 - TL 7473 N	2 - F 7473 PC 5 - DM 7473 N 8 - N 7473 A 11 - FLJ 121 14 - SF.C 473 E	3 - F 9 N 73 PC 6 - DM 8501 N 9 - T 7473 B1 12 - MIC 7473 N 15 - SW 7473 N
---	---	--

Diagrama lógico

(cada flip-flop)

RELOJ (CPI)

BORRADO (Rp)

Q

Q-bar

TABLA DE VERDAD

t_n		t_{n+1}	
J	K	Q	Q-bar
L	L	L	L
L	H	L	L
H	L	L	H
H	H	H	H

Notas:
 t_n = Instante antes del impulso de reloj
 t_{n+1} = Instante después del impulso de reloj

Lógica positiva:
 Entr. BAJA para clear de Q a nivel ALTO
 El borrado es independiente del reloj.

DESCRIPCION - Este dispositivo es un flip-flop dual JK maestro/auxiliar con entradas separadas de borrado y reloj en cada flip-flop. Las entradas a la sección maestro están controladas por el impulso de reloj. El impulso de reloj regula también el estado de los transistores de acoplamiento que conectan las secciones maestro y auxiliar. La secuencia de funcionamiento es como sigue:

- 1 - Aislar auxiliar de maestro.
- 2 - Entrar la información de las entradas J y K al maestro.
- 3 - Incapacitar las entradas J y K.
- 4 - Transferir la información del maestro al auxiliar.

FORMA DE ONDA DEL RELOJ

CONDICIONES DE FUNCIONAMIENTO RECOMENDADAS

PARAMETRO	MIN.	TIP.	MAX.	UNIDADES
Tensión de alimentación V_{CC}	4.75	5.0	5.25	Volts
Margen de temperatura ambiente	0	25	70	$^{\circ}C$
Cargab. de salida norm. de cada salida, N			10	U. L.
Anchura del impulso de reloj $t_p(\text{clock})$	20			ns
Anchura del impulso de borrado $t_p(\text{clear})$	25			ns
Tiempo de estab. (Setup) de entrada t_{setup}	$t_p(\text{clock})$			
Tiempo de manten. (Hold) de entrada t_{hold}	0			

CARACTERISTICAS ELECTRICAS EN EL MARGEN DE TEMPERATURA DE FUNCIONAMIENTO
 (si no se especifica otra cosa)

SÍMBOLO	PARAMETRO	MIN.	TIP. (3)	MAX.	UNIDADES	COND. PRUEBA (1)
V_{IH}	Tensión de entrada ALTA	2.0			Volts	Entrada ALTA Garantizada
V_{IL}	Tensión de entrada BAJA			0.8	Volts	Entrada BAJA Garantizada
V_{OH}	Tensión de salida ALTA	2.4			Volts	$V_{CC} = \text{MIN.}, I_{OH} = -0.4 \text{ mA}$
V_{OL}	Tensión de salida BAJA		0.22	0.4	Volts	$V_{CC} = \text{MIN.}, I_{OL} = 16 \text{ mA}$
	Corr. de entr. ALTA en J o K			40	μA	$V_{CC} = \text{MAX.}, V_{IN} = 2.4 \text{ V}$
				1.0	mA	$V_{CC} = \text{MAX.}, V_{IN} = 5.5 \text{ V}$
I_{IH}	Corr. de entr. ALTA en borrado, o reloj			80	μA	$V_{CC} = \text{MAX.}, V_{IN} = 2.4 \text{ V}$
				1.0	mA	$V_{CC} = \text{MAX.}, V_{IN} = 5.5 \text{ V}$
I_{IL}	Corr. de entr. BAJA en J o K			-1.6	mA	$V_{CC} = \text{MAX.}, V_{IN} = 0.4 \text{ V}$
	Entrada de corriente BAJA en borrado, o reloj			-3.2	mA	$V_{CC} = \text{MAX.}, V_{IN} = 0.4 \text{ V}$
I_{OS}	Corr. de sal. en c. (3)	-18		-57	mA	$V_{CC} = \text{MAX.}, V_{IN} = 0 \text{ V}$
I_{CC}	Corriente de alimentación		20	40	mA	$V_{CC} = \text{MAX.}$

Flip-flop de disparo por flanco tipo D dual																								
1 - SN 7474 N 4 - MC 7479 P 7 - ZN 7474 E 10 - FJJ 131 13 - TL 7474 N		2 - F 7474 PC 5 - DM 7474 N 8 - N 7474 A 11 - FLJ 141 14 - SF.C 474 E		3 - F 9 N 74 PC 6 - DM 8510 N 9 - T 7474 B 1 12 - MIC 7474 N 15 - SW 7474 N																				
Diagrama lógico (cada flip-flop)																								
				<table><tr><th colspan="3">TABLA DE VERDAD (cada flip-flop)</th></tr><tr><th>t_n</th><th colspan="2">t_{n+1}</th></tr><tr><th>ENTR</th><th>SALIDA</th><th>SALIDA</th></tr><tr><th>D</th><th>Q</th><th>\bar{Q}</th></tr><tr><td>L</td><td>L</td><td>H</td></tr><tr><td>H</td><td>H</td><td>L</td></tr></table> <p>Notas: t_n = instante antes del impulso de reloj t_{n+1} = instante después del impulso de reloj Lógica positiva: Entr. BAJA preset de Q a nivel ALTO Entr. BAJA clear de Q a nivel BAJO Preset y borrado (clear) son independientes del reloj.</p>			TABLA DE VERDAD (cada flip-flop)			t_n	t_{n+1}		ENTR	SALIDA	SALIDA	D	Q	\bar{Q}	L	L	H	H	H	L
TABLA DE VERDAD (cada flip-flop)																								
t_n	t_{n+1}																							
ENTR	SALIDA	SALIDA																						
D	Q	\bar{Q}																						
L	L	H																						
H	H	L																						
<p>DESCRIPCION - Este dispositivo es un flip-flop tipo D dual por disparo de flanco con entradas directas de borrado y preset así como salidas Q y Q. La información en la entrada es transferida a las salidas en el flanco positivo del impulso de reloj. Está diseñado para utilizarse en aplicaciones de media a alta velocidad.</p> <p>Los disparos del reloj ocurren a un nivel de tensión del impulso de reloj y no está directamente relacionado con el tiempo de transición del impulso hacia positivo. Una vez que se ha sobrepasado la tensión de umbral de la entrada de reloj, la entrada de datos (D) queda bloqueada y la información presente no será transferida a la salida.</p>																								
CONDICIONES DE FUNCIONAMIENTO RECOMENDADAS																								
PARAMETRO	MIN.	TIP.	MAX.	UNIDADES																				
Tensión de alimentación V_{CC}	4.75	5.0	5.25	Volts																				
Temperatura ambiente de funcionamiento	0	25	70	$^{\circ}C$																				
Cargab. de salida norm. de cada salida, N			10	U. L.																				
Anchura impulso de reloj $t_p(\text{clock})$	30			ns																				
Anchura del impulso de preset, $t_p(\text{preset})$	30			ns																				
Anchura del impulso de borrado, $t_p(\text{clear})$	30			ns																				
CARACTERISTICAS ELECTRICAS EN EL MARGEN DE TEMPERATURA DE FUNCIONAMIENTO (si no se especifica otra cosa)																								
SÍMBOLO	PARAMETRO	MIN.	TIP. (2)	MAX.	UNIDADES	COND. DE PRUEBA (1)																		
V_{IH}	Tensión de entrada ALTA	2.0			Volts	Entrada ALTA garantizada																		
V_{IL}	Tensión de entrada BAJA			0.8	Volts	Entrada BAJA garantizada																		
V_{OH}	Tensión de salida ALTA	2.4	2.5		Volts	$V_{CC} = \text{MIN.}, I_{OH} = -0.4 \text{ mA}$																		
V_{OL}	Tensión de salida BAJA		0.22	0.4	Volts	$V_{CC} = \text{MIN.}, I_{OL} = 16 \text{ mA}$																		
	Corriente de entr. ALTA en D			40	μA	$V_{CC} = \text{MAX.}, V_{IN} = 2.4 \text{ V}$																		
				1.0	mA	$V_{CC} = \text{MAX.}, V_{IN} = 5.5 \text{ V}$																		
I_{IH}	Corriente de entrada ALTA en preset o reloj			80	μA	$V_{CC} = \text{MAX.}, V_{IN} = 2.4 \text{ V}$																		
				1.0	mA	$V_{CC} = \text{MAX.}, V_{IN} = 5.5 \text{ V}$																		
	Corriente de entrada ALTA en borrador			120	μA	$V_{CC} = \text{MAX.}, V_{IN} = 2.4 \text{ V}$																		
				1.0	mA	$V_{CC} = \text{MAX.}, V_{IN} = 5.5 \text{ V}$																		
I_{IL}	Corr. de entr. BAJA preset o D			- 1.6	mA	$V_{CC} = \text{MAX.}, V_{IN} = 0.4 \text{ V}$																		
	Corriente de entrada BAJA en borrado o reloj			- 3.2	mA	$V_{CC} = \text{MAX.}, V_{IN} = 0.4 \text{ V}$																		
I_{OS}	Corr. de sal. en cortocircuito (3)	-18		- 57	mA	$V_{CC} = \text{MAX.}, V_{IN} = 0 \text{ V}$																		
I_{CC}	Corriente de alimentación		17	30	mA	$V_{CC} = \text{MAX.}$																		

Flip-flop séxtuple tipo D con borrado (clear)

1 - SN 74174 N 4 - 7 - ZN 74174 E 10 - 13 -	2 - 5 - DM 74174 N 8 - 11 - FLJ 531 14 -	3 - 6 - 9 - 12 - 15 - SW 74174 N
---	--	--

diagrama lógico

diagrama de conexión

símbolo lógico

DESCRIPCION. Estos flip-flops monolíticos, con disparo por flanco positivo, utilizan circuitería TTL para implementar lógica flip-flop tipo D. Todos tienen una entrada directa de borrado.

La información en las entradas D que cumpla los requisitos en cuanto a tiempo de formación (seput time) es transferida a las salidas Q en el flanco positivo del impulso de reloj. El disparo de reloj ocurre para un nivel de tensión particular y no está directamente relacionado con el tiempo de transición del paso a positivo del impulso. Cuando la entrada de reloj está en nivel alto o bajo, la señal de entrada D no tiene efecto sobre la salida.

Estos circuitos son perfectamente compatibles con la mayoría de los circuitos TTL o DTL.

CONDICIONES DE FUNCIONAMIENTO RECOMENDADAS

PARAMETRO		MIN.	TIP.	MAX.	UNIDADES
Tensión de alimentación, V_{CC}		4.75	5	5.25	Volts
Corriente de salida nivel ALTO, I_{OH}				-800	μA
Corriente de salida nivel BAJO, I_{OL}				16	mA
Frecuencia de reloj, f_{clock}		0		25	MHz
Anchura del impulso de reloj o borrado, t_w		20			ns
Tiempo de formación, t_{setup}	Entrada de datos	20			ns
Tiempo de mantenimiento de datos, t_{hold}	Borrado estado inactivo	25			ns
		5			ns
Temperatura ambiente de funcionamiento, T_A		0		70	$^{\circ}C$

CARACTERISTICAS ELÉCTRICAS EN EL MARGEN DE TEMPERATURA RECOMENDADO (a menos que se indique otra cosa)

SÍMBOLO	PARAMETRO	MIN.	TIP. (2)	MAX.	UNIDADES	CONDICIONES DE PRUEBA (1)
V_{IH}	Tensión de entrada nivel ALTO	2			Volts	$V_{CC} = MIN., I_I = -12 mA$
V_{IL}	Tensión de entrada nivel BAJO			0.8	Volts	
V_I	Tensión limitadora de entrada			1.5	Volts	
V_{OH}	Tensión de salida nivel ALTO	2.4	2.4		Volts	$V_{CC} = MIN., V_{IH} = 2 V$ $V_{IL} = 0.8 V, I_{OH} = -800 \mu A$
V_{OL}	Tensión de salida nivel BAJO		0.2	0.4	Volts	$V_{CC} = MIN., V_{IH} = 2 V$ $V_{IL} = 0.8 V, I_{OL} = 16 mA$
I_I	Corriente de entrada para tensión de entrada máxima			1	mA	$V_{CC} = MAX., V_I = 5.5 V$
I_{IH}	Corriente de entrada nivel ALTO			40	μA	$V_{CC} = MAX., V_I = 2.4 V$
I_{IL}	Corriente de entrada nivel BAJO			1.6	mA	$V_{CC} = MAX., V_I = 0.4 V$
I_{OS}	Corriente de salida en cortocircuito (3)	- 18		87	mA	$V_{CC} = MAX.$
I_{CC}	Corriente de alimentación (25)		45	65	mA	$V_{CC} = MAX.$

registro de desplazamiento de 8 bits

1 - SN 7491 AN 4 - MC 7491 AP 7 - ZN 7491 AE 10 - FJJ 151 13 -	2 - F 7491 PC 5 - DM 7491 AN 8 - N 7491 A 11 - FLJ 221 14 -	3 - F 9391 PC 6 - 9 - 12 - MIC 7491 AN 15 - SW 7491 N
--	---	---

diagrama lógico

simbolo lógico

DESCRIPCION. -Este dispositivo es un registro de desplazamiento de 8 bits de entrada serie, salida serie que utiliza tecnología TTL. Está compuesto de ocho flip-flops RS maestro/auxiliar, puerta de entrada y un excitador de reloj. El registro es capaz de almacenar y transferir datos de acuerdo con un reloj hasta 18 MHz manteniendo un nivel de inmunidad de ruido típico de 1 V. La disipación de potencia es típicamente 175 mW; se dispone en las salidas de una cargabilidad máxima de 10.

Los datos de un solo carril y el control de entrada pasan por una puerta a través de las entradas A y B y un inversor interno para formar las entradas complementarias al primer bit del registro de desplazamiento. La excitación para la línea de reloj interna común está proporcionada por un excitador de reloj inversor. Cada una de las entradas (A, B y CP) aparecen como únicamente una carga de entrada TTL.

El inversor/excitador de impulso de reloj hace que se desplace la información a la salida en el flanco positivo de un impulso de reloj de entrada, haciendo por tanto al registro de desplazamiento perfectamente compatible con otras funciones síncronas de disparo por flanco.

CONDICIONES DE FUNCIONAMIENTO RECOMENDADAS

PARAMETRO	MIN.	TIP.	MAX.	UNIDADES
Tensión de alimentación V_{CC} (Nota 10)	4.75	5.0	5.25	Volts
Margen de temper. ambiente de funcionam.	0	25	70	$^{\circ}\text{C}$
Carg. de salida normal. de las salidas			10	U. L.
Anchura de impulso de reloj, $t_{p(\text{clock})}$	25			ns
Tiempo de estab. (setup) de entr., t_{setup}	25			ns
Tiempo de manten. de entr., t_{hold}	0			

CARACTERÍSTICAS ELÉCTRICAS EN EL MARGEN DE TEMPERATURA Y FUNCIONAMIENTO
(si no se especifica otra cosa)

SÍMBOLO	PARAMETRO	MIN.	TIP. (3)	MAX.	UNIDADES	COND. DE PRUEBA (1)
V_{IH}	Tensión de entrada ALTA	2.0			Volts	V_{IH} garantizada
V_{IL}	Tensión de entrada BAJA			0.8	Volts	V_{IL} garantizada
V_{OH}	Tensión de salida ALTA	2.4	3.5		Volts	$V_{CC} = \text{MIN.}, I_{OH} = -0.4 \text{ mA}$
V_{OL}	Tensión de salida BAJA		0.22	0.4	Volts	$V_{CC} = \text{MIN.}, I_{OL} = 16 \text{ mA}$
I_{IH}	Corr. de entrada ALTA			40	μA	$V_{CC} = \text{MAX.}, V_{IN} = 2.4 \text{ V}$
I_{IL}	Corr. de entrada BAJA			1.0	mA	$V_{CC} = \text{MAX.}, V_{IN} = 5.5 \text{ V}$
I_{OS}	Corr. de salida en cort. (3)	10		87	mA	$V_{CC} = \text{MAX.}, V_{IN} = 0.4 \text{ V}$
I_{CC}	Corriente de alimentación		35	50	mA	$V_{OUT} = 0 \text{ V}$ $V_{CC} = \text{MAX.}$ $V_{IN} = 4.5 \text{ V}$

registro de desplazamiento de 4 bits a derecha e izquierda

1 - SN 7495 AN 4 - MC 7495 P 7 - ZN 7495 AE 10 - FJJ 231 13 -	2 - F 7495 PC 5 - DM 7495 N 8 - N 7495 A 11 - FLJ 191 14 - SF.C 495 E	3 - F 9395 PC 6 - DM 8580 N 9 - 12 - MIC 7495 AN 15 - SW 7495 N
---	---	---

diagrama lógico

símbolo lógico

DESCRIPCION. El Registro de Desplazamiento está compuesto por cuatro flip-flops RS maestro/auxiliar, cuatro puertas AND-OR-INVERT, una puerta AND-OR y seis excitadores-inversores. Las interconexiones interna de estas funciones proporcionan un registro versátil que puede efectuar operaciones de desplazamiento a derecha o desplazamiento a izquierda dependiendo del nivel de entrada lógico al control de modo. Pueden concertarse varios de estos registros en serie para formar un registro de n-bits de desplazamiento a derecha o desplazamiento a izquierda. Este registro puede utilizarse también como registro de almacenamiento de entrada paralelo, salida paralelo con control de puerta (modo).

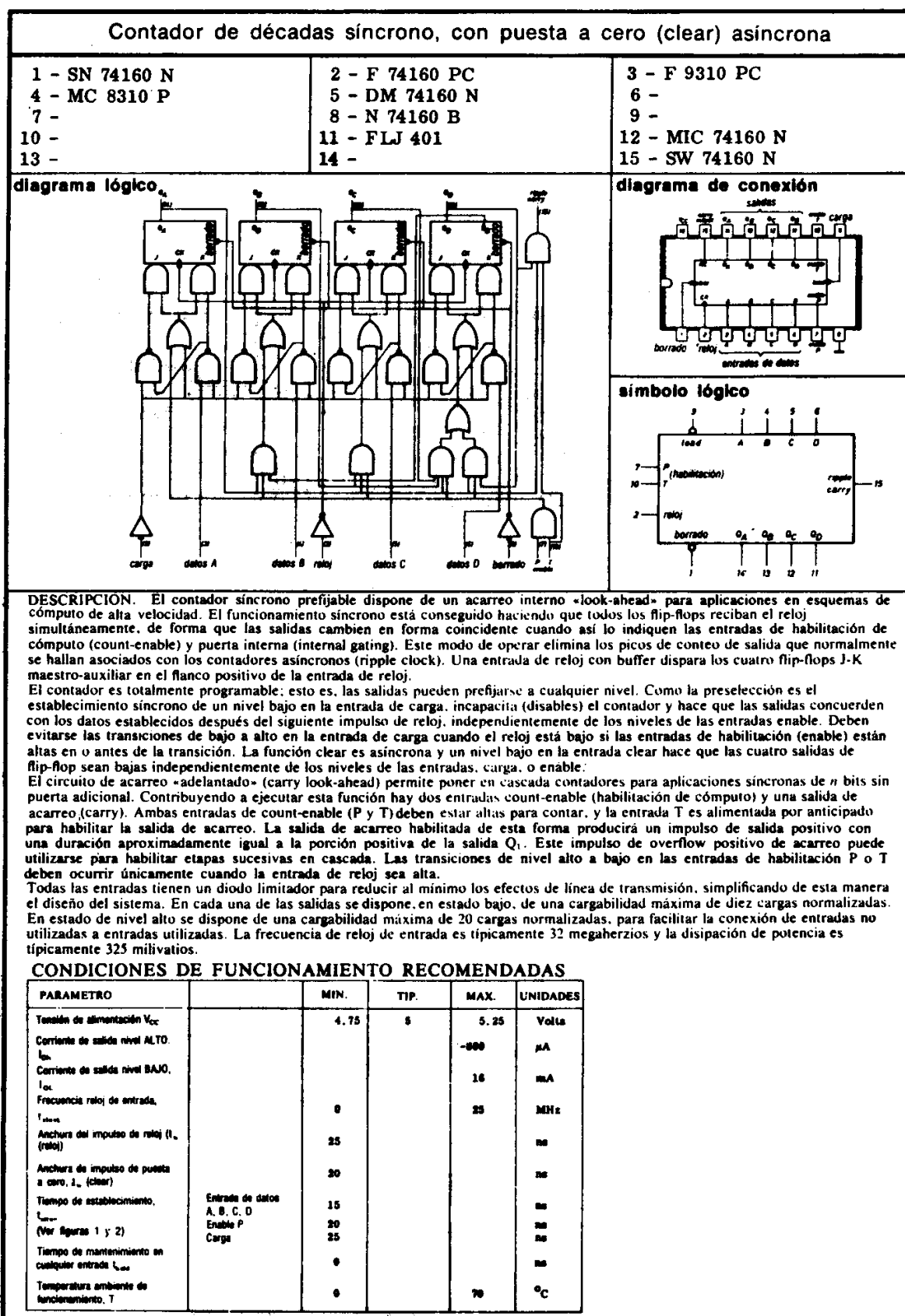
Cuando se aplica un nivel BAJO a la entrada de control de modo, las puertas AND número 1 quedan habilitadas y las puertas AND número 2 quedan inhibidas. En este modo la salida de cada flip-flop queda acoplada a las entradas RS de los flip-flops siguientes ejecutándose la operación de desplazamiento a derecha cuando se producen los impulsos de reloj en la entrada 1 de reloj. En este modo son admitidos datos en serie por la entrada serie. El reloj 2 y las entradas paralelo A hasta D quedan inhibidas por las puertas AND número 2.

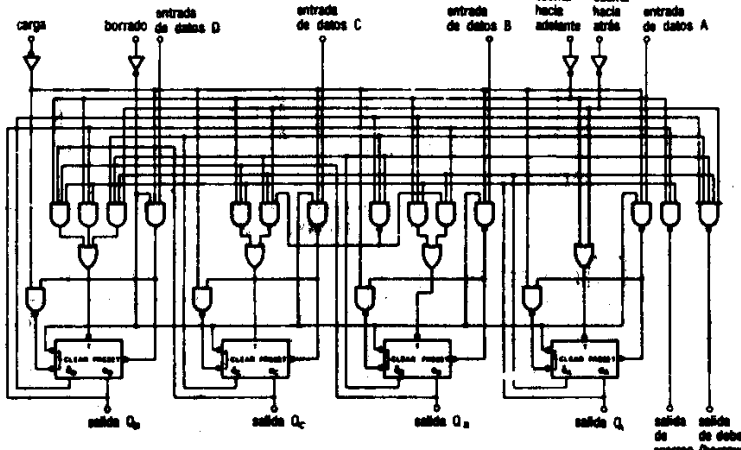
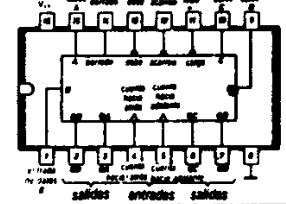
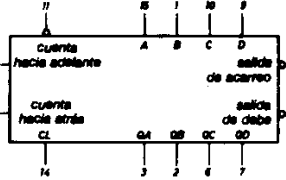
Cuando se aplica a la entrada de control de modo un nivel ALTO, las puertas AND número 1 quedan inhibidas (desacoplando las salidas de las entradas RS sucesivas para evitar el desplazamiento a la derecha) y las puertas AND número 2 quedan habilitadas para permitir la entrada de datos a través de las entradas paralelo A hasta B y reloj 2. Este modo permite la carga paralelo del registro, o mediante interconexión interna, funcionamiento de desplazamiento a la izquierda. En este modo, puede conseguirse el desplazamiento a la izquierda conectando la salida de cada flip-flop a la entrada paralelo de flip-flop anterior (Q_D a entrada C, y así sucesivamente), dirigiendo los datos en serie a la entrada D.

La actuación del reloj (clocking) en el registro de desplazamiento se efectúa a través de la puerta AND-OR E la cual permite fuentes separadas de reloj para utilizarse en los modos de desplazamiento a derecha y desplazamiento a izquierda. Si ambos modos pueden controlarse con un mismo reloj, la entrada de reloj puede aplicarse en forma común a reloj 1 y reloj 2. La información debe estar presente en las entradas RS de los flip-flops maestro-auxiliar antes de la aplicación del impulso de reloj. La transferencia de la información a las patillas de salida se produce cuando la entrada de reloj pasa de nivel ALTO a BAJO.

CONDICIONES DE FUNCIONAMIENTO RECOMENDADAS

PARAMETRO	MIN.	TIP.	MAX.	UNIDADES
Tensión de alimentación V _{CC} (Ver Nota 10)	4.75	5.0	5.25	Volts
Temperatura ambiente de funcionamiento	0	25	70	°C
Cargabilidad de salida de cada salida { nivel ALTO nivel BAJO			20 10	U. I.
Anchura del impulso de reloj, t _p (clock)	15	10		ns
Tiempo de establecimiento requerido en las entradas serie A, B, C, o D, t _{setup} (Ver Fig. A)	10			ns
Tiempo de mantenimiento requerido en las entradas serie A, B, C, o D, t _{hold} (Ver Fig. A)	0			ns
Tiempo de establecimiento de nivel BAJO requerido en el control de modo (t ₁ en Fig. B) (Con respecto a entrada de reloj 1)	15			ns
Tiempo de establecimiento de nivel ALTO requerido en el control de modo (t ₂ en Fig. B) (Con respecto a entrada de reloj 2)	15			ns
Tiempo de establecimiento de nivel BAJO requerido en el control de modo (t ₃ en Fig. B) (Con respecto a la entrada de reloj 2)	5.0			ns
Tiempo de establecimiento de nivel ALTO requerido en control de modo (t ₄ en Fig. B) (Con respecto a entrada de reloj 1)	5.0			ns



Contador de décadas reversible																																
1 - SN 74192 N 4 - MC 74192 P 7 - ZN 74192 E 10 - 13 - TL 74192 N	2 - F 74192 PC 5 - DM 74192 N 8 - N 74192 B 11 - FLJ 241 14 - SF.C 74192 E	3 - F 9360 PC 6 - DM 8560 N 9 - 12 - MIC 74192 N 15 - SW 74192 N																														
diagrama lógico 		diagrama de conexión 																														
DESCRIPCION. El dispositivo puede ponerse a cero (reset), prefijarse (preset) y contar hacia adelante (count up) o hacia atrás (count down). Estos modos de operación tabulados en la tabla de selección de modo. <p>La cuenta es síncrona, cambiando de estado las salidas después de la transición de bajo a alto del reloj de cuenta adelante o de cuenta atrás. La dirección de la cuenta queda determinada por aquella entrada de reloj que percibe un impulso, mientras la otra entrada de reloj está alta. (Se producirá una cuenta incorrecta si ambas entradas de reloj de cuenta adelante y cuenta atrás están bajas simultáneamente.) El contador responderá a un impulso de reloj en cualquiera de las entradas cambiando al siguiente estado apropiado de la secuencia de cuenta. (El diagrama de estado muestra la secuencia regular y además la secuencia de estado si se prefija en el contador un código superior a nueve.) El dispositivo tiene facilidad de carga paralelo (asíncrona) la cual permite la prefijación (preset) del contador. Siempre que la entrada de carga paralelo esté baja y el borrado (clear), bajo, la información presente en las entradas de datos paralelo quedarán cargadas en los contadores y aparecerán en las salidas independientemente de las condiciones de los impulsos de reloj. Cuando la entrada de carga paralelo pasa a alto esta información queda almacenada en el contador y cuando se le aplica el reloj cambian al siguiente estado apropiado en la secuencia de cuenta. Las entradas de datos paralelo quedan inhibidas cuando la carga paralelo está alta y no tiene ningún efecto sobre el contador.</p> <p>Las salidas de cuenta terminal hacia adelante y hacia atrás (acarreo y debe, respectivamente) permiten la operación de cuenta multidécada sin lógica adicional. Los contadores se ponen en cascada, llevando la salida del terminal de cuenta hacia adelante a la entrada de reloj de cuenta hacia adelante y la entrada de reloj de cuenta hacia atrás del terminal de cuenta hacia atrás de los contadores siguientes.</p> <p>La salida de cuenta terminal hacia adelante del dispositivo está baja cuando sus entradas de reloj de cuenta hacia adelante lo están y el contador está en estado nueve. Análogamente, las salidas de cuenta terminal hacia atrás están bajas cuando sus entradas de reloj de cuenta atrás están bajas y ambos contadores están en estado cero. Por tanto, cuando el contador BCD esté en estado nueve y contando hacia adelante, o bien el contador está en estado cero y contando hacia atrás, un impulso de reloj cambiará el estado del contador en el flanco positivo y simultáneamente dará reloj al contador siguiente a través de la salida de cuenta terminal de bajo activo apropiada. Existen dos retardos de puerta por estado cuando los contadores están en cascada.</p> <p>La entrada de borrado asíncrono, cuando está alta, cancela todas las entradas y borra (pone a cero) los contadores. El borrado cancela (overrules) la carga paralelo con lo cual cuando ambas estén activadas los contadores se pondrán a cero (reset). (Evidentemente, la carga paralelo y master reset no deben desactivarse simultáneamente en una operación que puede predecirse.)</p>		símbolo lógico 																														
<table border="0" style="width: 100%;"> <tr> <td style="width: 30%;">PATILLAS</td> <td style="width: 40%;"></td> <td style="width: 30%; text-align: right;">CARGA</td> </tr> <tr> <td>\bar{L}</td> <td>Entrada de carga paralelo (BAJO ACTIVO)</td> <td style="text-align: right;">1 U. L.</td> </tr> <tr> <td>A, B, C, D</td> <td>Entradas de datos paralelo</td> <td style="text-align: right;">1 U. L.</td> </tr> <tr> <td>C_U</td> <td>Entrada de reloj para cuenta hacia adelante</td> <td style="text-align: right;">1 U. L.</td> </tr> <tr> <td>C_D</td> <td>Entrada de reloj para cuenta hacia atrás</td> <td style="text-align: right;">1 U. L.</td> </tr> <tr> <td>CL</td> <td>Entrada de puesta a cero principal (Master Reset)</td> <td style="text-align: right;">1 U. L.</td> </tr> <tr> <td>QA, QB, QC, QD</td> <td>(Clear) (Asíncrona)</td> <td style="text-align: right;">10 U. L.</td> </tr> <tr> <td>\bar{C}_0</td> <td>Salidas del contador</td> <td style="text-align: right;">10 U. L.</td> </tr> <tr> <td>\bar{B}_0</td> <td>Salidas de cuenta hacia adelante del terminal (acarreo) (carry)</td> <td style="text-align: right;">10 U. L.</td> </tr> <tr> <td></td> <td>Salida de cuenta hacia atrás del terminal (debe) borrow</td> <td style="text-align: right;">10 U. L.</td> </tr> </table>			PATILLAS		CARGA	\bar{L}	Entrada de carga paralelo (BAJO ACTIVO)	1 U. L.	A, B, C, D	Entradas de datos paralelo	1 U. L.	C _U	Entrada de reloj para cuenta hacia adelante	1 U. L.	C _D	Entrada de reloj para cuenta hacia atrás	1 U. L.	CL	Entrada de puesta a cero principal (Master Reset)	1 U. L.	QA, QB, QC, QD	(Clear) (Asíncrona)	10 U. L.	\bar{C}_0	Salidas del contador	10 U. L.	\bar{B}_0	Salidas de cuenta hacia adelante del terminal (acarreo) (carry)	10 U. L.		Salida de cuenta hacia atrás del terminal (debe) borrow	10 U. L.
PATILLAS		CARGA																														
\bar{L}	Entrada de carga paralelo (BAJO ACTIVO)	1 U. L.																														
A, B, C, D	Entradas de datos paralelo	1 U. L.																														
C _U	Entrada de reloj para cuenta hacia adelante	1 U. L.																														
C _D	Entrada de reloj para cuenta hacia atrás	1 U. L.																														
CL	Entrada de puesta a cero principal (Master Reset)	1 U. L.																														
QA, QB, QC, QD	(Clear) (Asíncrona)	10 U. L.																														
\bar{C}_0	Salidas del contador	10 U. L.																														
\bar{B}_0	Salidas de cuenta hacia adelante del terminal (acarreo) (carry)	10 U. L.																														
	Salida de cuenta hacia atrás del terminal (debe) borrow	10 U. L.																														
<p>1 Unidad de carga (U. L.) = 40 μA ALTO 1,6 mA BAJO</p> <p>CONDICIONES DE FUNCIONAMIENTO RECOMENDADAS</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>PARAMETRO</th> <th>MIN.</th> <th>TIP.</th> <th>MAX.</th> <th>UNIDADES</th> </tr> </thead> <tbody> <tr> <td>tensión de alimentación V_{CC}</td> <td>4.75</td> <td>5.0</td> <td>5.25</td> <td>Volts</td> </tr> <tr> <td>Margen de temperatura ambiente</td> <td>0</td> <td>25</td> <td>75</td> <td>°C</td> </tr> </tbody> </table>			PARAMETRO	MIN.	TIP.	MAX.	UNIDADES	tensión de alimentación V _{CC}	4.75	5.0	5.25	Volts	Margen de temperatura ambiente	0	25	75	°C															
PARAMETRO	MIN.	TIP.	MAX.	UNIDADES																												
tensión de alimentación V _{CC}	4.75	5.0	5.25	Volts																												
Margen de temperatura ambiente	0	25	75	°C																												

contador de décadas (división por dos y división por cinco)

1 - SN 7490 AN 4 - MC 7490 P 7 - ZN 7490 E 10 - FJJ 141 13 - TL 7490 N	2 - F 7490 PC 5 - DM 7490 N 8 - N 7490 A 11 - FLJ 161 14 - SF.C 490 E	3 - F 9390 PC 6 - DM 8530 N 9 - T 7490 B 1 12 - MIC 7490 N 15 - SW 7490 N
--	---	---

diagrama lógico

simbolo lógico

DESCRIPCION.—El dispositivo es un Contador de Décadas que consta de cuatro rangos duales de flip-flops maestro-auxiliar, interconectados directamente para proporcionar un contador divisor por dos y un contador divisor por cinco. Las entradas de conteo están inhibidas, y todas las salidas puestas a cero lógico o una cuenta binaria codificada decimal (BCD) de 9 a través de líneas de reset directas con puerta. La salida del flip-flop A no está internamente conectada a las etapas siguientes, y por tanto el conteo puede separarse en estos modos independientes:

- Si se utiliza como contador de décadas binario codificado decimal, la entrada \overline{CP}_{BD} debe conectarse externamente a la salida Q_A . La entrada \overline{CP}_A recibe la cuenta de entrada, obteniéndose una secuencia de conteo de acuerdo con la cuenta BCD para la aplicación decimal de complemento a nueve.
- Si se desea una cuenta de división por diez simétrica para sintetizadores de frecuencia u otras aplicaciones que requieran la división de una cuenta binaria por una potencia de diez, la salida Q_D debe conectarse externamente a la entrada de \overline{CP}_A . La cuenta de entrada se aplica entonces a la entrada \overline{CP}_{BD} obteniéndose una onda cuadrada dividida por diez en la salida Q_A .
- Para funcionar como contador divisor por dos y divisor por cinco, no se requiere interconexión interna. El flip-flop A se utiliza como elemento binario para la función división por dos. La entrada \overline{CP}_{BD} se utiliza para obtener una operación binaria de división por cinco en las salidas Q_B , Q_C y Q_D . En este modo los dos contadores operan independientemente; no obstante, todos los cuatro flip-flops se ponen en reset simultáneamente.

PATILLAS		CARGA
R 0	Entradas de reset de cero	1 U. L.
R 9	Entradas de reset de nueve	1 U. L.
\overline{CP}_A	Entrada de reloj	2 U. L.
\overline{CP}_{BD}	Entrada de reloj	4 U. L.
Q_A, Q_B, Q_C, Q_D	Salidas	10 U. L.

1 carga unidad (U.L.) = 40 μ A ALTO/1,6 mA BAJO.

CONDICIONES DE FUNCIONAMIENTO RECOMENDADAS

PARAMETRO	MIN.	TIP.	MAX.	UNIDADES
Tensión de alimentación V_{CC} (Nota 10)	4.75	5.0	5.25	Volts
Margen de temp. ambiente de funcionamiento	4	25	70	$^{\circ}C$
Cargabilidad de salida normalizada de cada salida, N (nota 12)			10	U. L.
Ancho de impulso de cuenta de entr., $t_{p(im)}$	50			ns
Ancho de impulso de reset, $t_{p(reset)}$	50			ns

FINAL

COM'L: -25 IND: -15/25


**Advanced
Micro
Devices**

PALCE22V10Z Family

Zero-Power 24-Pin EE CMOS Versatile PAL Device

DISTINCTIVE CHARACTERISTICS

- **Zero-power CMOS technology**
 - 15 μ A standby current
 - As fast as 15 ns first-access propagation delay and 50 MHz f_{MAX} (external)
- **Unused product term disable for reduced power consumption**
- **Available in Industrial operating range**
 - $T_c = -40^\circ\text{C}$ to $+85^\circ\text{C}$
 - $V_{CC} = +4.5\text{ V}$ to $+5.5\text{ V}$
- **HC- and HCT-compatible inputs and outputs**
- **Electrically-erasable technology provides reconfigurable logic and full testability**
- **10 macrocells programmable as registered or combinatorial, and active high or active low to match application needs**
- **Varied product term distribution allows up to 16 product terms per output for complex functions**
- **Global asynchronous reset and synchronous preset for initialization**
- **Power-up reset for initialization and register preload for testability**
- **Extensive third-party software and programmer support through FusionPLD partners**
- **24-pin SKINNYDIP and 28-pin PLCC packages save space**

GENERAL DESCRIPTION

The PALCE22V10Z is an advanced PAL device built with zero-power, high-speed, electrically-erasable CMOS technology. It provides user-programmable logic for replacing conventional zero-power CMOS SSI/MSI gates and flip-flops at a reduced chip count.

The PALCE22V10Z provides zero standby power and high speed. At 15 μ A maximum standby current, the PALCE22V10Z allows battery powered operation for an extended period.

The ZPAL™ device implements the familiar Boolean logic transfer function, the sum of products. The PAL device is a programmable AND array driving a fixed OR array. The AND array is programmed to create custom product terms, while the OR array sums selected terms at the outputs.

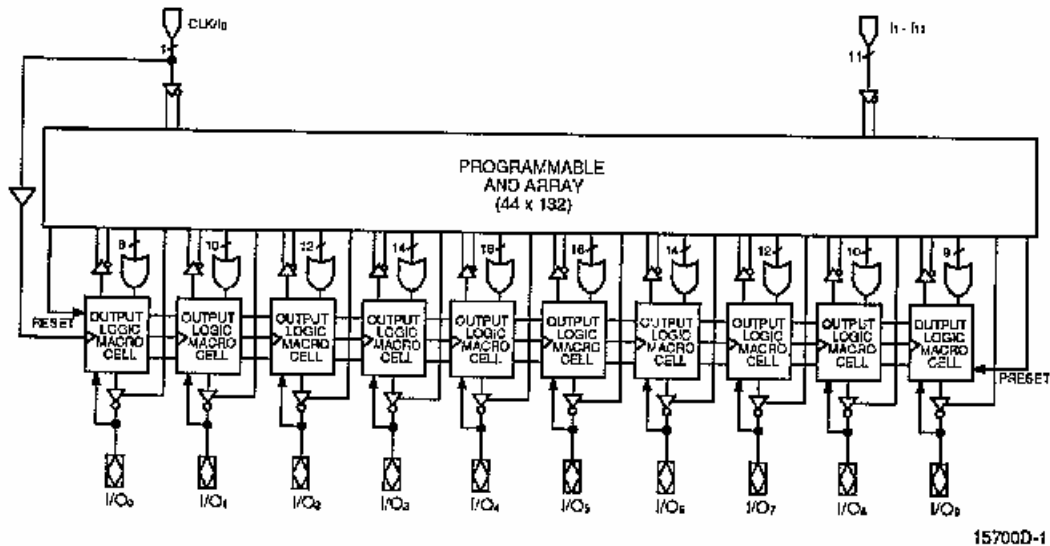
The product terms are connected to the fixed OR array with a varied distribution from 8 to 16 across the outputs (see Block Diagram). The OR sum of the products feeds

the output macrocell. Each macrocell can be programmed as registered or combinatorial, and active high or active low. The output configuration is determined by two bits controlling two multiplexers in each macrocell.

AMD's FusionPLD program allows PALCE22V10Z designs to be implemented using a wide variety of popular industry-standard design tools. By working closely with the FusionPLD partners, AMD certifies that the tools provide accurate, quality support. By ensuring that third-party tools are available, costs are lowered because a designer does not have to buy a complete set of new tools for each device. The FusionPLD program also greatly reduces design time since a designer can use a tool that is already installed and familiar. Please refer to the Software Reference Guide to PLD Compilers for certified development systems, and the Programmer Reference Guide for approved programmers.



BLOCK DIAGRAM

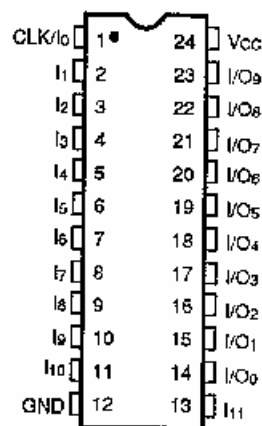


15700D-1

CONNECTION DIAGRAMS

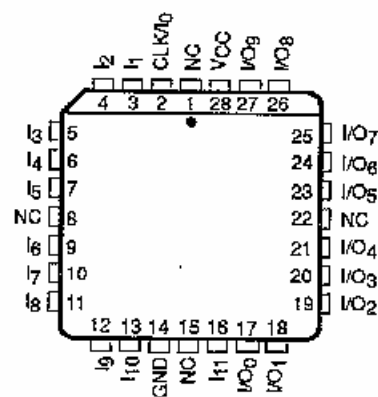
Top View

SKINNYDIP/SOIC



15700D-2

PLCC



15700D-3

Note:

Pin 1 is marked for orientation.

PIN DESCRIPTION

CLK = Clock
 GND = Ground
 I = Input
 I/O = Input/Output
 NC = No Connect
 Vcc = Supply Voltage

FINAL COM'L: -7/10/12/15/20, Q-12/15/20 IND: -12/14/18/24

MACH210A-7/10/12
MACH210-12/15/20
MACH210AQ-12/15/20
High-Density EE CMOS Programmable Logic



DISTINCTIVE CHARACTERISTICS

- 44 Pins
- 64 Macrocells
- 7.5 ns t_{pd} Commercial
12 ns t_{pd} Industrial
- 133 MHz f_{max}
- 38 Inputs; 210A Inputs have built-in pull-up resistors
- Peripheral Component Interconnect (PCI) compliant
- 32 Outputs
- 64 Flip-flops; 2 clock choices
- 4 "PAL22V16" blocks with buried macrocells
- Pin-compatible with MACH110, MACH111, MACH211, and MACH215

GENERAL DESCRIPTION

The MACH210 is a member of AMD's high-performance EE CMOS MACH 2 device family. This device has approximately six times the logic macrocell capability of the popular PAL22V10 without loss of speed.

The MACH210 consists of four PAL blocks interconnected by a programmable switch matrix. The four PAL blocks are essentially "PAL22V16" structures complete with product-term arrays and programmable macrocells, including additional buried macrocells. The switch matrix connects the PAL blocks to each other and to all input pins, providing a high degree of connectivity between the fully-connected PAL blocks. This allows designs to be placed and routed efficiently.

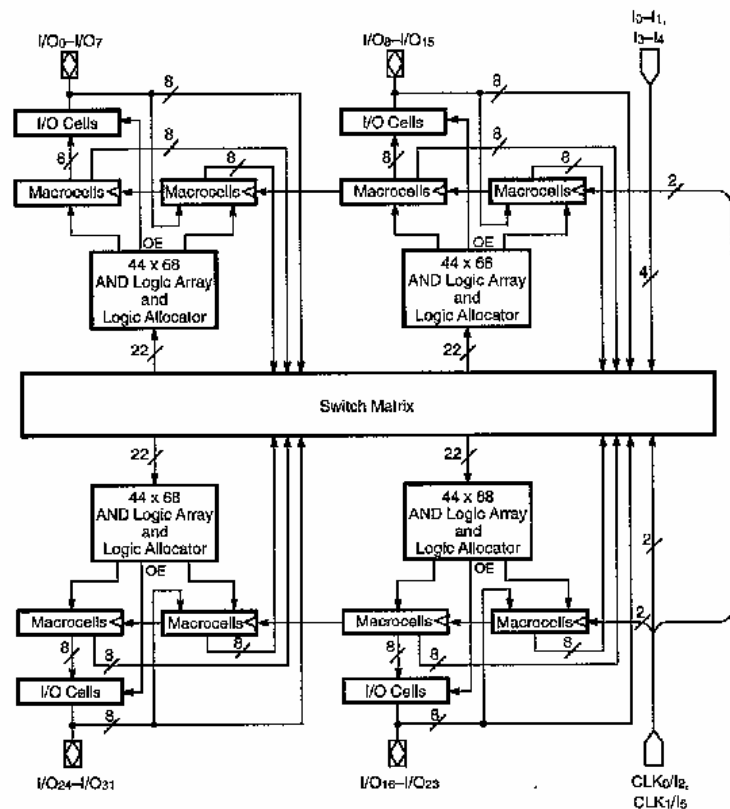
The MACH210 has two kinds of macrocell: output and buried. The MACH210 output macrocell provides regis-

tered, latched, or combinatorial outputs with programmable polarity. If a registered configuration is chosen, the register can be configured as D-type or T-type to help reduce the number of product terms. The register type decision can be made by the designer or by the software. All output macrocells can be connected to an I/O cell. If a buried macrocell is desired, the internal feedback path from the macrocell can be used, which frees up the I/O pin for use as an input.

The MACH210 has dedicated buried macrocells which, in addition to the capabilities of the output macrocell, also provide input registers or latches for use in synchronizing signals and reducing setup time requirements.



BLOCK DIAGRAM



141281-1



FUNCTIONAL DESCRIPTION

The MACH210 consists of four PAL blocks connected by a switch matrix. There are 32 I/O pins and 4 dedicated input pins feeding the switch matrix. These signals are distributed to the four PAL blocks for efficient design implementation. There are two clock pins that can also be used as dedicated inputs.

The MACH210A inputs and I/O pins have built-in pull-up resistors. While it is always a good design practice to tie unused pins high, the 210A pull-up resistors provide design security and stability in the event that unused pins are left disconnected.

The PAL Blocks

Each PAL block in the MACH210 (Figure 1) contains a 64-product-term logic array, a logic allocator, 8 output macrocells, 8 buried macrocells, and 8 I/O cells. The switch matrix feeds each PAL block with 22 inputs. This makes the PAL block look effectively like an independent "PAL22V16" with 8 buried macrocells.

In addition to the logic product terms, two output enable product terms, an asynchronous reset product term, and an asynchronous preset product term are provided. One of the two output enable product terms can be chosen within each I/O cell in the PAL block. All flip-flops within the PAL block are initialized together.

The Switch Matrix

The MACH210 switch matrix is fed by the inputs and feedback signals from the PAL blocks. Each PAL block provides 16 internal feedback signals and 8 I/O feedback signals. The switch matrix distributes these signals back to the PAL blocks in an efficient manner that also provides for high performance. The design software automatically configures the switch matrix when fitting a design into the device.

The Product-term Array

The MACH210 product-term array consists of 64 product terms for logic use, and 4 special-purpose product terms. Two of the special-purpose product terms provide programmable output enable; one provides asynchronous reset, and one provides asynchronous preset.

The Logic Allocator

The logic allocator in the MACH210 takes the 64 logic product terms and allocates them to the 16 macrocells as needed. Each macrocell can be driven by up to 16 product terms. The design software automatically configures the logic allocator when fitting the design into the device.



The I/O Cell

The I/O cell in the MACH210 consists of a three-state output buffer. The three-state buffer can be configured in one of three ways: always enabled, always disabled, or controlled by a product term. If product term control is chosen, one of two product terms may be used to provide the control. The two product terms that are available are common to all I/O cells in a PAL block.

These choices make it possible to use the macrocell as an output, an input, a bidirectional pin, or a three-state output for use in driving a bus.

Table 1 illustrates which product term clusters are available to each macrocell within a PAL block. Refer to Figure 1 for cluster and macrocell numbers.

Table 1. Logic Allocation

Macrocell		Available Clusters
Output	Buried	
M ₀	M ₁	C ₀ , C ₁ , C ₂ C ₀ , C ₁ , C ₂ , C ₃
M ₂	M ₃	C ₁ , C ₂ , C ₃ , C ₄ C ₂ , C ₃ , C ₄ , C ₅
M ₄	M ₅	C ₃ , C ₄ , C ₅ , C ₆ C ₄ , C ₅ , C ₆ , C ₇
M ₆	M ₇	C ₅ , C ₆ , C ₇ , C ₈ C ₆ , C ₇ , C ₈ , C ₉
M ₈	M ₉	C ₇ , C ₈ , C ₉ , C ₁₀ C ₈ , C ₉ , C ₁₀ , C ₁₁
M ₁₀	M ₁₁	C ₉ , C ₁₀ , C ₁₁ , C ₁₂ C ₁₀ , C ₁₁ , C ₁₂ , C ₁₃
M ₁₂	M ₁₃	C ₁₁ , C ₁₂ , C ₁₃ , C ₁₄ C ₁₂ , C ₁₃ , C ₁₄ , C ₁₅
M ₁₄	M ₁₅	C ₁₃ , C ₁₄ , C ₁₅ C ₁₄ , C ₁₅

The Macrocell

The MACH210 has two types of macrocell: output and buried. The output macrocells can be configured as either registered, latched, or combinatorial, with programmable polarity. The macrocell provides internal feedback whether configured with or without the flip-flop. The registers can be configured as D-type or T-type, allowing for product-term optimization.

The flip-flops can individually select one of two clock/gate pins, which are also available as data inputs. The registers are clocked on the LOW-to-HIGH transition of the clock signal. The latch holds its data when the gate input is HIGH, and is transparent when the gate input is LOW. The flip-flops can also be asynchronously initialized with the common asynchronous reset and preset product terms.

The buried macrocells are the same as the output macrocells if they are used for generating logic. In that case, the only thing that distinguishes them from the output macrocells is the fact that there is no I/O cell connection, and the signal is only used internally. The buried macrocell can also be configured as an input register or latch.

PCI Compliance

The MACH210A-7/10 is fully compliant with the *PCI Local Bus Specification* published by the PCI Special Interest Group. The MACH210A-7/10's predictable timing ensures compliance with the PCI AC specifications independent of the design. On the other hand, in CPLD and FPGA architectures without predictable timing, PCI compliance is dependent upon routing and product term distribution.



MAX 5000

Programmable Logic Device Family

January 1998, ver. 4

Data Sheet

Features...

- Advanced Multiple Array MatriX (MAX[®]) 5000 architecture combining speed and ease-of-use of PAL devices with the density of programmable gate arrays
- Complete family of high-performance, erasable CMOS EPROM erasable programmable logic devices (EPLDs) for designs ranging from fast 28-pin address decoders to 100-pin LSI custom peripherals
- 600 to 3,750 usable gates (see [Table 1](#))
- Fast, 15-ns combinatorial delays and 76.9-MHz counter frequencies
- Configurable expander product-term distribution allowing more than 32 product terms in a single macrocell
- 28 to 100 pins available in DIP, J-lead, PGA, and QFP packages
- Programmable registers providing D, T, JK, and SR flipflop functionality with individual clear, preset, and clock controls
- Programmable security bit for protection of proprietary designs
- Software design support featuring the Altera[®] MAX+PLUS[®] II development system on 486- or Pentium-based PCs, and Sun SPARCstation, HP 9000 Series 700/800, and IBM RISC System/6000 workstations

Table 1. MAX 5000 Device Features

Feature	EPM5032	EPM5064	EPM5128	EPM5130	EPM5192
Usable gates	600	1,250	2,500	2,500	3,750
Macrocells	32	64	128	128	192
Logic array blocks (LABs)	1	4	8	8	12
Expanders	64	128	256	256	384
Routing	Global	PIA	PIA	PIA	PIA
Maximum user I/O pins	24	36	60	84	72
t _{PD} (ns)	15	25	25	25	25
t _{ASU} (ns)	4	4	4	4	4
t _{CO} (ns)	10	14	14	14	14
f _{CNT} (MHz)	76.9	50	50	50	50

MAX 5000 Programmable Logic Device Family Data Sheet

...and More
Features

- Programming support with Altera's Master Programming Unit (MPU) or programming hardware from third-party manufacturers
- Additional design entry and simulation support provided by EDIF, library of parameterized modules (LPM), Verilog HDL, VHDL, and other interfaces to popular EDA tools from manufacturers such as Cadence, Exemplar Logic, Mentor Graphics, OrCAD, Synopsys, Synplicity, and Viewlogic

General
Description

The MAX 5000 family combines innovative architecture and advanced process technologies to offer optimum performance, flexibility, and the highest logic-to-pin ratio of any general-purpose programmable logic device (PLD) family. The MAX 5000 family provides 600 to 3,750 usable gates, pin-to-pin delays as fast as 15 ns, and counter frequencies of up to 76.9 MHz. See [Table 2](#).

Table 2. MAX 5000 Device Speed Grades

Device	Speed (t_{PD1})				
	15 ns	20 ns	25 ns	30 ns	35 ns
EPM5032	✓	✓	✓		
EPM5064			✓	✓	✓
EPM5128			✓	✓	✓
EPM5130			✓		✓
EPM5192			✓		✓

The MAX 5000 architecture supports 100% TTL emulation and high-density integration of multiple SSI, MSI, and LSI logic functions. For example, an EPM5192 device can replace over 100 74-series devices; it can integrate complete subsystems into a single package, saving board area and reducing power consumption. MAX 5000 EPLDs are available in a wide range of packages (see [Table 3](#)), including the following:

- Windowed ceramic and plastic dual in-line (CerDIP and PDIP)
- Plastic J-lead chip carrier (PLCC)
- Windowed ceramic pin-grid array (PGA)
- Plastic quad flat pack (PQFP)

MAX 5000 Programmable Logic Device Family Data Sheet

Table 3. MAX 5000 Pin Count & Package Options

Device	Pin Count				
	28	44	68	84	100
EPM5032	CerDIP PDIP PLCC				
EPM5064		PLCC			
EPM5128			PLCC PGA		
EPM5130				PLCC	PGA PQFP
EPM5192				PLCC PGA	

MAX 5000 EPLDs have between 32 and 192 macrocells that are combined into groups called logic array blocks (LABs). Each macrocell has a programmable-AND/fixed-OR array and a configurable register that provides D, T, JK, or SR operation with independent programmable clock, clear, and preset functions. To build complex logic functions, each macrocell can be supplemented with shareable expander product terms ("shared expanders") to provide more than 32 product terms per macrocell.

The MAX 5000 family is supported by Altera's MAX+PLUS II development system, a single, integrated package that offers schematic, text—including the Altera Hardware Description Language (AHDL)—and waveform design entry; compilation and logic synthesis; simulation and timing analysis; and device programming. The MAX+PLUS II system provides EDIF 2 0 0 and 3 0 0, LPM, VHDL, Verilog HDL, and other interfaces for additional design entry and simulation support from other industry-standard PC- and UNIX workstation-based EDA tools. The MAX+PLUS II software runs on 486- and Pentium-based PCs, and Sun SPARCstation, HP 9000 Series 700/800, IBM RISC System/6000 workstations.



For more information, go to the *MAX+PLUS II Programmable Logic Development System & Software Data Sheet* in this data book.



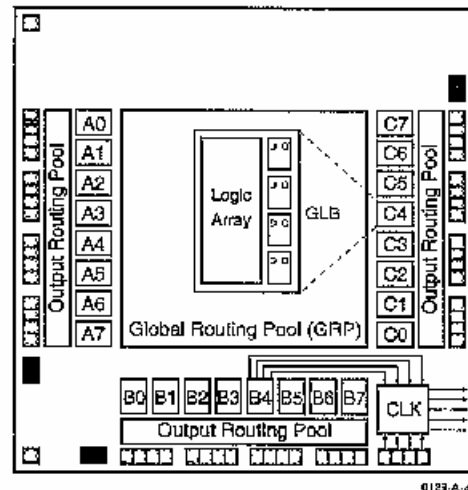
pLSI® and ispLSI™ 1024

High-Density Programmable Logic

Features

- **PROGRAMMABLE AND IN-SYSTEM PROGRAMMABLE HIGH DENSITY LOGIC**
 - High-Speed Global Interconnect
 - 4000 PLD Gates
 - 48 I/O Pins, Six Dedicated Inputs
 - 144 Registers
 - Wide Input Gating for Fast Counters, State Machines, Address Decoders, etc.
 - Small Logic Block Size for Fast Random Logic
 - Security Cell Prevents Unauthorized Copying
- **HIGH PERFORMANCE E²C²MOS™ TECHNOLOGY**
 - f_{max} = 90 MHz Maximum Operating Frequency
 - f_{max} = 60 MHz for Industrial and Military/883 Devices
 - t_{pd} = 12 ns Propagation Delay
 - TTL Compatible Inputs and Outputs
 - Electrically Erasable and Reprogrammable
 - Non-Volatile E²C²MOS Technology
 - 100% Tested
- **ispLSI OFFERS THE FOLLOWING ADDED FEATURES**
 - In-System Programmable 5-Volt Only
 - Change Logic and Interconnects "On-the-Fly" in Seconds
 - Reprogram Soldered Device for Debugging
- **COMBINES EASE OF USE AND THE FAST SYSTEM SPEED OF PLDs WITH THE DENSITY AND FLEXIBILITY OF FIELD PROGRAMMABLE GATE ARRAYS**
 - Complete Programmable Device Can Combine Glue Logic and Structured Designs
 - Four Dedicated Clock Input Pins
 - Synchronous and Asynchronous Clocks
 - Flexible Pin Placement
 - Optimized Global Routing Pool Provides Global Interconnectivity
- **pLSI/ispLSI DEVELOPMENT SYSTEM (pDS™)**
 - pDS Software**
 - Easy to Use PC Windows™ Interface
 - Boolean Logic Compiler
 - Manual Partitioning
 - Automatic Place and Route
 - Static Timing Table
 - pDS+™ Software**
 - Industry Standard, Third Party Design Environments
 - Schematic Capture, State Machine, HDL
 - Automatic Partitioning and Place and Route
 - Comprehensive Logic and Timing Simulation
 - PC and Workstation Platforms

Functional Block Diagram



Description

The Lattice pLSI and ispLSI 1024 are High-Density Programmable Logic Devices containing 144 Registers, 48 Universal I/O pins, six Dedicated Input pins, four Dedicated Clock Input pins and a Global Routing Pool (GRP). The GRP provides complete interconnectivity between all of these elements. The ispLSI 1024 features 5-Volt in-system programmability and In-system diagnostic capabilities. It is the first device which offers non-volatile "on-the-fly" reprogrammability of the logic, as well as the interconnect to provide truly reconfigurable systems. It is architecturally and parametrically compatible to the pLSI 1024 device, but multiplexes four of the dedicated input pins to control In-system programming.

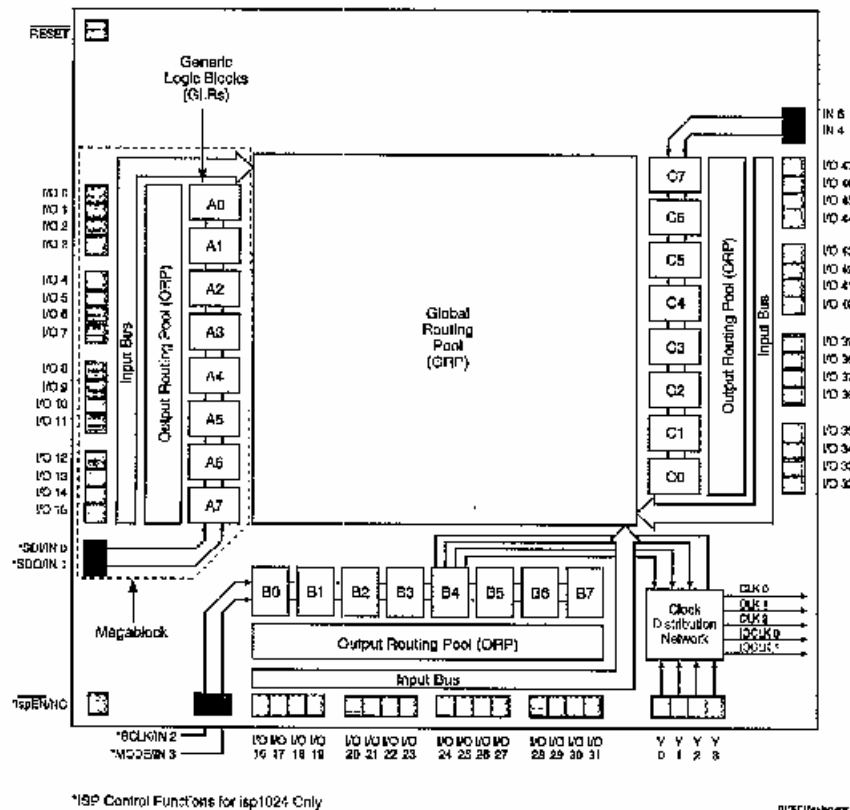
The basic unit of logic on the pLSI and ispLSI 1024 devices is the Generic Logic Block (GLB). The GLBs are labeled A0, A1 .. C7 (see figure 1). There are a total of 24 GLBs in the pLSI and ispLSI 1024 devices. Each GLB has 18 inputs, a programmable AND/OR/XOR array, and four outputs which can be configured to be either combinatorial or registered. Inputs to the GLB come from the GRP and dedicated inputs. All of the GLB outputs are brought back into the GRP so that they can be connected to the inputs of any other GLB on the device.



Specifications *pLSI and ispLSI 1024*

Functional Block Diagram

Figure 1. pLSI and ispLSI 1024 Functional Block Diagram



The devices also have 48 I/O cells, each of which is directly connected to an I/O pin. Each I/O cell can be individually programmed to be a combinatorial input, registered input, latched input, output or bi-directional I/O pin with 3-state control. Additionally, all outputs are polarity selectable, active high or active low. The signal levels are TTL compatible voltages and the output drivers can source 4 mA or sink 8 mA.

Eight GLBs, 16 I/O cells, two dedicated inputs and one ORP are connected together to make a Megablock (see figure 1). The outputs of the eight GLBs are connected to a set of 16 universal I/O cells by the ORP. The I/O cells within the Megablock also share a common Output Enable (OE) signal. The pLSI and ispLSI 1024 devices contain three of these Megablocks.

The GRP has as its inputs the outputs from all of the GLBs and all of the inputs from the bi-directional I/O cells. All of these signals are made available to the inputs of the GLBs. Delays through the GRP have been equalized to minimize timing skew.

Clocks in the pLSI and ispLSI 1024 devices are selected using the Clock Distribution Network. Four dedicated clock pins (Y0, Y1, Y2 and Y3) are brought into the distribution network, and five clock outputs (CLK 0, CLK 1, CLK 2, IOCLK 0 and IOCLK 1) are provided to route clocks to the GLBs and I/O cells. The Clock Distribution Network can also be driven from a special clock GLB (B4 on the pLSI and ispLSI 1024 devices). The logic of this GLB allows the user to create an internal clock from a combination of internal signals within the device.



XC4000E and XC4000X Series Field Programmable Gate Arrays

January 29, 1999 (Version 1.6)

Product Specification

XC4000E and XC4000X Series Features

Note: XC4000 Series devices described in this data sheet include the XC4000E family and XC4000X Series. XC4000X Series devices described in this data sheet include the XC4000EX and XC4000XL families. Separate data sheets are available for two other families in the XC4000X series, the XC4000XLI and XC4000XV. This information does not apply to the older Xilinx families: XC4000, XC4000A, XC4000D, XC4000H, or XC4000L. For information on these devices, see the Xilinx WEOLINX at <http://www.xilinx.com>.

- System featured Field-Programmable Gate Arrays
 - Select RAM[®] memory: on-chip ultra fast RAM with
 - synchronous write option
 - dual-port RAM option
 - Fully PCI compliant (speed grades -2 and faster)
 - Abundant flip-flops
 - Flexible function generators
 - Dedicated high-speed carry logic
 - Wide output buffers on each output
 - Hierarchy of Interconnect Lines
 - Internal 3-state bus capability
 - A global low-skew clock or signal distribution networks
- System Performance beyond 80 MHz
- Flexible Array Architecture
- Low Power Segmented Routing Architecture
- Systems-Oriented Features
 - IEEE 1149.1-compatible boundary scan logic support
 - Individually programmable output slew rate
 - Programmable input pull-up or pull-down resistors
 - 12-mA sink current per XC4000E output
- Configured by Loading Binary File
 - Unlimited reprogrammability
- Readback Capability
 - Program verification
 - Internal node observability
- Backward Compatible with XC4000 Devices
- Development System runs on most common computer platforms
 - Interfaces to popular design environments
 - Fully automatic mapping, placement and routing
 - Interactive design editor for design optimization

Low-Voltage Versions Available

- Low-Voltage Devices Function at 3.0 - 3.6 Volts
- XC4000XL High Performance Low Voltage Versions of XC4000EX devices

Additional XC4000X Series Features

- Highest Performance — 3.0 V XC4000XL
- Highest Capacity — Over 180,000 Usable Gates
- 5V tolerant I/Os on XC4000XL
- 0.35µ SRAM process for XC4000XL
- Additional Routing Over XC4000E
 - almost twice the routing capacity for high-density designs
- Buffered Interconnect for Maximum Speed
- Improved VersaRing[™] I/O Interconnect for Better Fixed Pinout Flexibility
- 12-mA Sink Current Per XC4000X Output
- Flexible New High Speed Clock Network
 - 8 additional Early Buffers for shorter clock delays
 - Virtually unlimited number of clock signals
- Optional Multiplexer or 2 Input Function Generator on Device Outputs
- 4 Additional Address Bits in Master Parallel Configuration Mode
- XC4000XLT devices, optimized for PCI applications, are available.
- The XC4000XV Family offers the highest density with 0.25 micron 2.5 volt technology.

Introduction

XC4000 Series high-performance, high-capacity Field Programmable Gate Arrays (FPGAs) provide the benefits of custom CMOS VLSI while avoiding the initial cost, long development cycle, and inherent risk of a conventional masked gate array.

The result of thirteen years of FPGA design experience and feedback from thousands of customers, these FPGAs combine architectural versatility, on-chip Select-RAM memory with edge-triggered and dual-port masters, increased speed, abundant routing resources, and new, sophisticated software to achieve fully automated implementation of complex, high-density, high-performance designs.

The XC4000E and XC4000X Series currently have 20 members, as shown in Table 1.

January 29, 1999 (Version 1.6)

6 / 7

XC4000E and XC4000X Series Field Programmable Gate Arrays**Table 1: XC4000E and XC4000X Series Field Programmable Gate Arrays**

Device	Logic Cells	Max Logic Gates (No RAM)	Max. RAM Bits (No Logic)	Typical Gate Range (Logic and RAM)*	CLB Matrix	Total CLBs	Number of Flip-Flops	Max. User I/O
XC4002XL	152	1,500	2,148	1,000 - 3,000	8 x 8	64	256	64
XC4003E	238	3,000	3,200	2,000 - 5,000	10 x 10	100	360	80
XC4005E/XL	488	5,000	6,272	3,000 - 9,000	14 x 14	196	616	112
XC4006E	608	6,000	8,162	4,000 - 12,000	16 x 16	256	768	128
XC4008E	770	8,000	10,368	6,000 - 15,000	18 x 18	324	836	144
XC4010E/XL	960	10,000	12,800	7,000 - 20,000	20 x 20	400	1,120	160
XC4012E/XL	1,368	13,000	19,432	10,000 - 30,000	24 x 24	576	1,636	192
XC4020E/XL	1,862	20,000	25,068	13,000 - 40,000	28 x 28	784	2,016	224
XC4025E	2,432	25,000	32,768	15,000 - 45,000	32 x 32	1,024	2,560	256
XC4028E-XL	2,132	28,000	32,768	16,000 - 50,000	32 x 32	1,024	2,560	256
XC4035E-XL	3,078	38,000	41,472	22,000 - 65,000	38 x 38	1,444	3,168	268
XC4044XL	3,600	44,000	51,200	27,000 - 80,000	40 x 40	1,600	3,840	320
XC4052XL	4,590	52,000	61,862	33,000 - 100,000	44 x 44	1,936	4,576	352
XC4062XL	5,472	62,000	73,728	40,000 - 130,000	48 x 48	2,304	5,376	384
XC4085XL	7,448	85,000	100,262	55,000 - 180,000	56 x 56	3,136	7,168	448

* Max values of Typical Gate Range include 20-30% of CLBs used as RAM.

Notes: All functionality in low-voltage families is the same as in the corresponding 5-Volt family, except where numerical references are made to timing or power.

Description

XC4000 Series devices are implemented with a regular, flexible, programmable architecture of Configurable Logic Blocks (CLBs), interconnected by a powerful hierarchy of versatile routing resources, and surrounded by a perimeter of programmable Input/Output Blocks (IOBs). They have generous routing resources to accommodate the most complex interconnect patterns.

The devices are customized by loading configuration data into internal memory cells. The FPGA can either actively read its configuration data from an external serial or byte-parallel PROM (master mode), or the configuration data can be written into the FPGA from an external device (slave and peripheral modes).

XC4000 Series FPGAs are supported by powerful and sophisticated software, covering every aspect of design from schematic or behavioral entry, floorplanning, simulation, automatic block placement and routing of interconnects, to the creation, downloading, and readback of the configuration bit stream.

Because Xilinx FPGAs can be reprogrammed an unlimited number of times, they can be used in innovative designs

where hardware is changed dynamically, or where hardware must be adapted to different user applications. FPGAs are ideal for shortening design and development cycles, and also offer a cost-effective solution for production runs well beyond 5,000 systems per month. For lowest high-volume unit cost, a design can first be implemented in the XC4000E or XC4000X, then migrated to one of Xilinx' compatible HardWire mask-programmed devices.

Taking Advantage of Reconfiguration

FPGA devices can be reconfigured to change logic function while resident in the system. This capability gives the system designer a new degree of freedom not available with any other type of logic.

Hardware can be changed as easily as software. Design updates or modifications are easy, and can be made to products already in the field. An FPGA can even be reconfigured dynamically to perform different functions at different times.

Reconfigurable logic can be used to implement system self diagnostics, create systems capable of being reconfigured for different environments or operations, or implement multi-purpose hardware for a given application. As an added benefit, using reconfigurable FPGA devices simplifies hardware design and debugging and shortens product time-to-market.