Anexo A: FPGA

Introducción

Cuando se requiere del diseño de un sistema electrónico y surge la necesidad de implementar una parte con hardware dedicado son varias las posibilidades que hay. Una es un diseño full-custom, pero resulta de un costo elevado y enorme complejidad de diseño. Otro posibilidad es la implementación software, que es muy barata y flexible.
Las diferentes soluciones para el diseño de circuitos digitales son.

Programa software.
Lógica programable: MPGAs ,FPGAs.
Células estándar.
Full-custom.
Que presentan alta flexibilidad, complejidad y costo.

Entre estas dos opciones se puede elegir la fabricación de un circuito electrónico realizado mediante diseño semi-custom, utilizando células estándar, o recurrir a un circuito ya fabricado que se pueda programar por el usuario, como son las FPGAs. De estas dos opciones la primera proporciona mejores prestaciones, aunque es mas cara y exige un ciclo de diseño relativamente largo. Por otro lado, los dispositivos lógicos programables constituyen una buena oferta para realizar diseños electrónicos digitales con un buen compromiso costo-prestaciones. Y lo que es mejor, permiten obtener una implementación en un tiempo de diseño asombrosamente corto.

Otro aspecto que se debe tener en cuenta para decidirse por este tipo de implementación es que el costo de realización es muy bajo, por lo que suele ser una buena opción para la realización de prototipos.

Evolución de los dispositivos programables

Se entiende por dispositivo programable aquel circuito de propósito general que posee una estructura interna que puede ser modificada por el usuario (o a petición suya, por el fabricante) para implementar una amplia gama de aplicaciones. El primer dispositivo que cumplió estas características es una memoria PROM, que puede realizar un comportamiento de circuito utilizando las líneas de

direcciones como entradas y las de datos como salidas (para implementar una tabla de verdad). Existen dos tipos básicos de PROM: Programables por mascara (en la fabrica), que proporcionan mejores características y son las denominadas de conexiones hardwired. Programables en el campo por en usuario. Son las EPROM y EEPROM que Proporcionan peores caracteristicas, pero son menos costosas para volúmenes pequeños de producción y se pueden programar de manera inmediata.

El PLD, Programmable Logic Device, es una matriz de puertas AND conectada a otra matriz de puertas OR más biestables. Cualquier circuito lógico se puede implementar, por tanto, como suma de productos. La versión más básica del mismo es una PAL, con un plano de puertas AND y otro fijo de puertas OR. Las salidas de estas ultimas se pueden pasar por un biestable en la mayoría de los circuitos comerciales. Ventaja: son bastante eficientes si se implementan circuitos no superiores a unos centenares de puertas.

Desventajas: arquitectura rígida, y está limitado por un número fijo de biestables y entradas/salidas.

La PLA, Programmable Logic Array, es más flexible que la PAL: se pueden programar las conexiones entre los dos planos. Estos dispositivos son muy simples y producen buenos resultados con funcionalidades sencillas (sólo combi nacional). Hace falta algo un poco más sofisticado y general: una matriz de elementos variados que se puedan interconectar libremente. Este es el caso de una MPGA (Mask-Programmable Gate Array), cuyo principal representante está constituido por un conjunto de transistores más circuitería de E/S. Se unen mediante pistas de metal que hay que trazar de forma óptima, siendo esta la máscara que hay que enviar al fabricante.

Las FPGA (Field Programmable Gate Array), introducidas por Xilinx en 1985, son el dispositivo programable por el usuario mas general. También se denominan LCA (Logic Cell Array). Consisten en una matriz bidimensional de bloques configurables que se pueden conectar mediante recursos generales de interconexión. Estos recursos incluyen segmentos de pista de diferentes longitudes, mas unos conmutadores programables para enlazar bloques a pistas o pistas entre sí. En realidad, lo que se programa en un FPGA son los

conmutadores que sirven para realizar las conexiones entre los diferentes bloques, más la configuración de los bloques.

Estructura general de las FPGAs

El proceso de diseño de un circuito digital utilizando una matriz lógica programable puede descomponerse en dos etapas básicas:

Dividir el circuito en bloques básicos, asignándolos a los bloques configurables del dispositivo.
Conectar los bloques de lógica mediante los conmutadores necesarios.

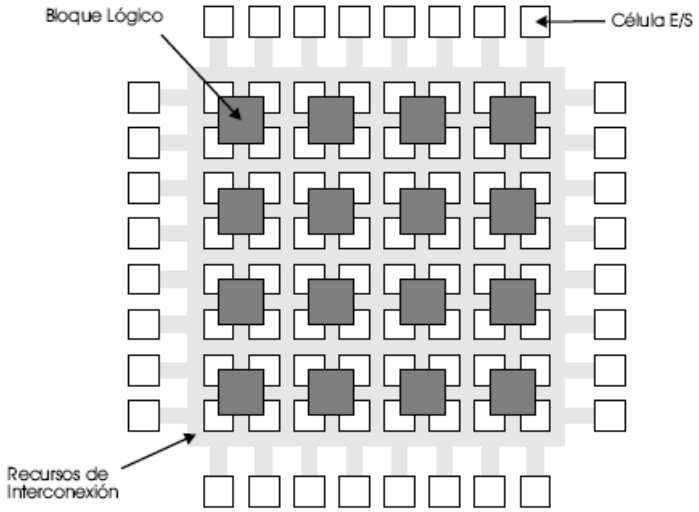

Fig.A1.-Estructura general de un FPGA(Xilinx)

Para ello el fabricante proporciona las herramientas de diseño adecuadas. Los elementos básicos constituyentes de una FPGA como las de Xilinx se pueden ver en la fig.A1 y son los siguientes:

1. Bloques lógicos, cuya estructura y contenido se denomina arquitectura. Hay muchos tipos de arquitecturas, que varían principalmente en complejidad (desde una simple puerta hasta módulos mas complejos o estructuras tipo PLD). Suelen incluir biestables para facilitar la implementación de circuitos secuenciales. Otros módulos de importancia son los bloques de Entrada/Salida.

2. Recursos de interconexión, cuya estructura y contenido se denomina arquitectura de rutado.
3. Memoria RAM, que se carga durante el RESET para configurar bloques y conectarlos.

Entre las numerosas ventajas que proporciona el uso de FPGAs dos destacan principalmente: el bajo costo y el corto tiempo de producción. Entre los inconvenientes de su utilización están su baja velocidad de operación y baja densidad lógica (poca lógica implementable en un solo chip) pero para la aplicación en circuitos de control es suficiente (200 mhz aprox). Su baja conexión se debe a los retardos introducidos por los conmutadores y las largas pistas de conexión.

Por supuesto, no todas las FPGA son iguales. Dependiendo del fabricante nos podemos encontrar con diferentes soluciones. Las FPGAs que existen en la actualidad en el mercado se pueden clasificar como pertenecientes a cuatro grandes familias, dependiendo de la estructura Que adoptan los bloques lógicos que tengan definidos. Las cuatro estructuras se pueden ver en la fig.A2, sin que aparezcan en la misma los bloques de entrada/salida.

1. Matriz simétrica, como son las de XILINX
2. Basada en canales, ACTEL
3. Mar de puertas, ORCA
4. PLD jerárquica, ALTERA o CPLDs de XILINX.

En concreto, para explicar el funcionamiento y la estructura básica de este tipo de dispositivos programables solo se consideran las familias de XILINX.

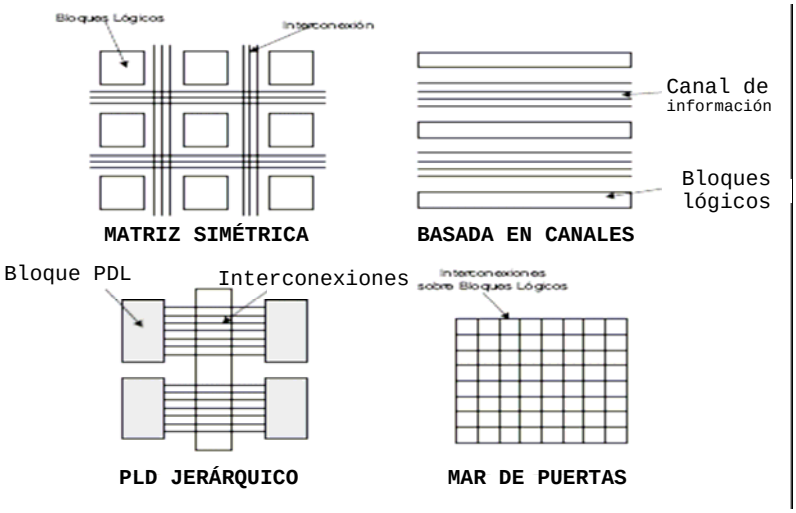

Fig.A2.-Tipos de FPGAs

Arquitectura de las FPGA de Xilinx.

Tecnología de programación

Antes de continuar con conocimientos mas avanzados acerca de FPGAs (de XILINX en concreto), hay que aclarar como se realiza el proceso de programación (ie., las conexiones necesarias entre bloques y pistas). En primer lugar, si se piensa que el número de dispositivos de conexión que hay en una FPGA es muy grande (típicamente superior a 100000), es necesario que cumplan las siguientes propiedades:

Ser lo más pequeños que posible.

Tener la resistencia ON lo mas baja posible, mientras la OFF ha de ser lo mas alta posible (para que funcione como conmutador).

Se deben poder incorporar al proceso de fabricación de los FPGA.

El proceso de programación no es único, sino que se puede realizar mediante diferentes tecnologías, como son células RAM estáticas, transistores EPROM y EEPROM, etc. En el caso de las FPGAs de XILINX los elementos de programación se basan en células de memoria RAM que controlan transistores de paso, puertas de transmisión o multiplexores. En la fig-A3 se puede ver esquemáticamente como son. Dependiendo del tipo de conexión requerida se elegirá un modelo u otro.

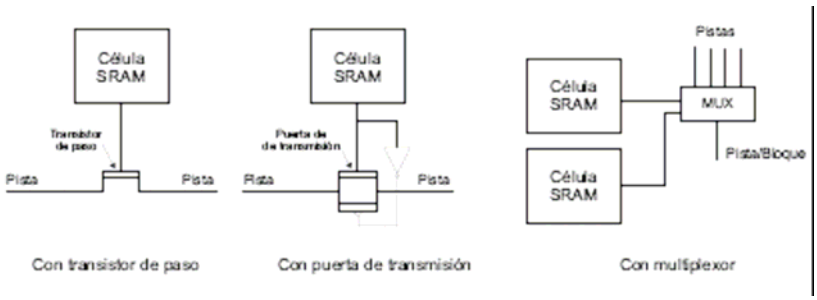

Fig. A3.-Tipos de conectores utilizados por xilinx.

Es importante destacar que si se utilizan células SRAM la configuración de la FPGA seria valida únicamente mientras este conectada la alimentación, pues es memoria volátil. En los sistemas finales esta claro que hace falta algún mecanismo de almacenamiento no volátil que cargue las células de RAM. Esto se puede conseguir mediante EPROMs o disco.

Este elemento de programación es relativamente grande (necesita por lo menos 5 transistores), pero se puede implementar en el proceso normal de fabricación del circuito (CMOS).

Además, permite reconfigurar la FPGA de una forma muy rápida.

Descripción de las principales familias

Hay múltiples familias lógicas dentro de XILINX. Las primeras que surgieron son: XC2000 (descatalogada en el año 1999), XC3000 y XC4000, correspondientes respectivamente a la primera, segunda y tercera generación de dispositivos, que se distinguen por el tipo de bloque lógico configurable (CLB) que contienen. En la actualidad existen también las familias de FPGA SpartanII, SpartanIII, Virtex, VirtexII y VirtexPro.

La Tabla A1 muestra la cantidad de CLBs que puede haber en cada FPGA de las familias base y ese mismo valor expresado en puertas equivalentes.

La Tabla A1. Familias del fabricante XILINX

SERIE	Tipo CLB	No de CLBs	Puertas Equivalentes
XC2000	1 LUT,1 FF	64-100	1.200-1.800
XC3000	1 LUT,2FF	64-484	1.500-7.500
XC4000XL	3 LUT,2FF	64-3136	1.600-180.000

El bloque lógico ha de ser capaz de proporcionar una función lógica en general y reprogramable.

La mejor forma de realizar esto es mediante una tabla de valores preasignados" o tablas de look-up". Básicamente, una tabla de look-up (LUTs en adelante) es una memoria, con un circuito de control que se encarga de cargar los datos.

Cuando se aplica en una dirección las entradas de la función booleana la memoria devuelve un dato, lo que se puede hacer corresponder con la salida requerida. Falta añadir los componentes necesarios para desempeñar funciones no posibles de implementar con una memoria, tales como una batería de registros, multiplexores, buffers etc. Estos componentes están en posiciones fijas del dispositivo.

La ventaja de la utilización de este tipo de tablas es su gran flexibilidad: una LUT de k entradas puede implementar

cualquier función booleana de k variables, y hay 22k funciones posibles. El inconveniente es obvio: ocupan mucho espacio y no son muy aprovechables fig.A4.

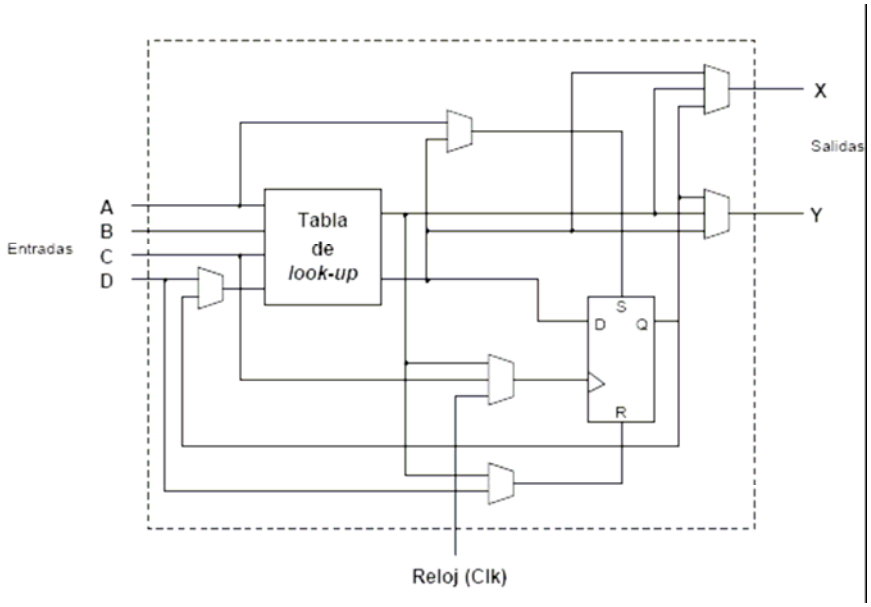

Fig.A4. LUT.

Los bloques lógicos configurables de la familia XC2000 se componen de una tabla de lookup con cuatro entradas y un biestable, con lo que puede generar cualquier función de hasta 4 variables o dos funciones de 3 variables. El de la familia XC3000 es mas complejo: permite implementar una función de 5 variables o dos funciones de 4 variables (limitadas a 5 diferentes entradas, claro). Además contiene dos biestables y cierta lógica. La familia XC4000 es ya mucho mas sofisticada. Tiene tres tablas de look-up dispuestas en dos niveles, llegando a poder implementar funciones de hasta 9 variables.

En general, los recursos de interconexión son de tres tipos:

Conexiones directas, permiten la conexión de las salidas del CLB con sus vecinos mas directos (N, S, E y O).

Interconexiones de propósito general, para distancias superiores a un CLB (mas allá del vecino). Son pistas horizontales y verticales del tamaño de un CLB, pero que se pueden empalmar para crear pistas mas largas.

Líneas de largo recorrido, suelen cubrir lo ancho o largo de la pastilla. Permiten conexiones con un retardo mucho menor que uniendo las anteriores.

El camino crítico de un circuito es el recorrido que, desde una entrada hasta una salida, presenta un retardo máximo.

Descripción detallada de la LUT

Hasta ahora, se ha presentado el CLB de la XC2000 desde el punto de vista de un usuario final. Dado que ahora se va a implementar siguiendo una metodología full-custom, es necesario conocer mas detalles. En la fig.A5 se ve el conjunto de entradas y salidas que tiene realmente la LUT. Como se puede observar, en el símbolo aparecen dos entradas nuevas, las relacionadas con la programación de la LUT:

Las líneas de datos Din, sirven para introducir en la LUT los datos de programación.

La línea de programación Prog, su valor seria diferente en función del modo en que se este (programación/funcionamiento normal).

Como ya se ha explicado previamente, la forma inmediata de implementar una LUT es utilizar una memoria RAM estática. Por ejemplo, considera la tabla de verdad de la función f = ab + c que aparece en la figura. Si esta función lógica se implementa con una LUT de tres entradas, seria necesaria una RAM de $2^3 = 8$ posiciones, y almacenaremos 1 en la posición 000, 0 en la posición 001, y así sucesivamente.

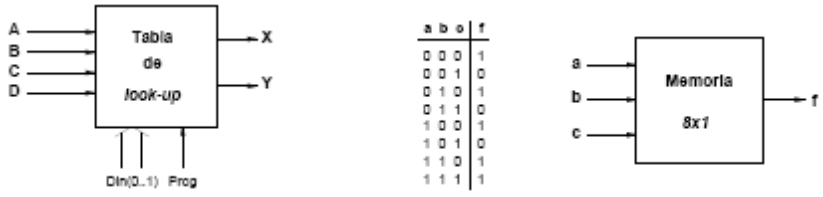

Fig.A5.-Símbolo e implementación de la LUT.

Si queremos realizar una LUT de dos salidas lo podemos hacer con 2 memorias de 2k posiciones, o lo que es mejor, con una memoria de k líneas de dirección y 2 bit de palabra. Por

supuesto, se esta ocupando mucha área, pero esta es la mejor forma de realizar una función booleana cualquiera.

La equivalencia de terminales entre la LUT y la memoria son los siguientes. Las líneas de entrada corresponden al bus de direcciones; las líneas de datos equivalen a bus de datos de entrada; las salidas (X, Y en nuestro caso) corresponden a bus de datos de salida; y la señal de programación seria la línea de lectura/escritura, R/W de la memoria. Dependiendo del tipo de implementación de la memoria se puede necesitar alguna entrada mas (por ejemplo, se puede utilizar alguna señal de reloj para temporizar el funcionamiento de la memoria).

Programación de FPGA.

La herramienta principal para la programación de un FPGA es el lenguaje VHDL, este es un lenguaje orientado a la implementación de hardware mediante programación.