

# Opciones de implementación tecnológica

## Diseño Full Custom

↳ justificado su uso desde un punto de vista económico Si:

- \* Gran volumen de producción
- \* Puede ser reutilizado
- \* El costo no es primordial en el diseño
- \* Características críticas (velocidad, consumo, área)

↳ Herramientas utilizadas:

- \* Editor de Layout
- \* Chequeador de Reglas de Diseño (DRC)
- \* Extractor del circuito

↳ Editor de Layout: genera la representación física del diseño dada una topología

- \* layout simbólico (stick diagram)
- \* las reglas de diseño imponen restricciones
- \* Herramienta MAGIC (U. Berkeley)

↳ Chequeador de reglas de diseño

- \* Verifica que el layout físico del cto. cumple c/ las reglas de diseño de la tecnología.
- \* Formatos: Online y batch (Post diseño)
- \* Consume gran cantidad de tiempo
- \* Preserva el diseño jerárquico en el nivel físico.

↳ Extractor del circuito: Crea un esquemático del layout físico.

\* Reconstruye la red de transistores incluyendo sus tamaños y las interconexiones entre los pines

\* Reconstruye las capacidades con respecto a GND o a otros nodos de la red.

Diseño semi-custom basado en celdas estándar.

↳ Se tiene una reducción de la densidad de integración y/o prestaciones

↳ Reutilización de librerías de Células.

↳ Se verifican las celdas una única vez por tecnología

↳ Los fabricantes ofrecen librerías de células optimizadas y garantizadas

↳ Niveles de Células

\* Células estándar (lógica)

\* Macrocelulas: RAM, ROM, Multiplicadores

\* Bloques funcionales

↳ CPU RISC

↳ Microcontroladores

↳ UARTS

\* Células analógicas

↳ Amp. Operacionales

↳ Conversores A/D y D/A

⊖ Reduce la posibilidad de un diseño fino

↳ Estandarización a nivel de compuertas:

\* Ineficiente para estructuras regulares

↳ Registros

↳ Sumadores

↳ Multiplicadores

↳ Memorias

↳ Operaciones aritméticas, lógicas y de almacenamiento

\* El uso de celdas puede ser ineficiente.

↳ Aproximaciones bit-slice

↳ Regularidad en una dimensión

## Dispositivos lógicos programables (PLD)

Eliminan el proceso de routing. Los caminos de conexión están realizados y únicamente se controlan las uniones entre ellos.

→ AND fija OR programable

- PROM: No tienen salidas programable ni realimentadas.

Las funciones se implementan directamente a partir de la tabla de la verdad, sin necesidad de minimizar

→ AND programable OR fija

- PAL: Número limitado de términos producto por salida

Las funciones deben minimizarse en forma individual

no es posible compartir términos.

→ Ambas programables

- PLA: Matriz AND de 8 sumo 7 u 8 implicantes primos

esenciales, no más de 8 compuertas AND por salida.

Matriz OR solo para los 8 Implicantes

Tamaño de chip menor, disminución capacidades parásitas

Los anteriores solo se pueden programar una vez  
→ AND programable OR fija

- GAL: Tiene configuraciones de salida reprogramables  
usan tecnología E2CMOS (Electrically Erasable CMOS)

- CPLD: Incluyen varias matrices PAL en un mismo chip.

- Reducción al mínimo del consumo.

- Macroceldas agrupadas en bloques de función con estructura PLA. Flip Flop configurable, multiplexores, polaridad configurable, suma de productos, hasta 40 entradas y 56 terminales.

- Las F.B. se conectan mediante una matriz de interconexión avanzada.

- Incorpora bloque de Programación en sistema (ISP) y control de Boundary Scan (BSC)

- FPGA: Conjunto de unidades de procesamiento configurables insertadas en una red de interconexión también configurable

- Granularidad, procesamiento independiente a nivel de bit

- Control de instrucciones, 1 instrucción por elemento proceso

- Interconexión estática, conmutaciones configurables.

- Consta de tres tipos de elementos programables:

1. Configurable Logic Block, CLB

2. Switching Matrix, SM

3. Input/Output Blocks, IOB

- Presenta líneas de interconexión agrupadas en canales verticales y horizontales. Mayor regularidad q CPLD

- La memoria de configuración almacena la info necesaria para programar los elementos disponibles. Son inicializadas en el proceso de carga del programa de configuración

# Flujo de Diseño en FPGAs

- El proceso de diseño de un módulo hardware se divide en tres fases: creación del modelo, síntesis e implementación
- Al finalizar cada una de las fases puede comprobarse la validez del diseño mediante simulación.
  - Simulación funcional RTL, valida el modelo VHDL creado
  - Simulación funcional post-síntesis a nivel de puertas para validar el modelo sintetizado
  - Simulación temporal post-implementación para validar la implementación del modelo usando la tecnología de una FPGA concreta.
- El diseño descrito en lenguaje de alto nivel VHDL puede contener:
  - Código de modelos genéricos independientes de la tecnología destino.
  - Componentes instanciados propios de la arquitectura del dispositivo en el que se va a implementar
    - ↳ Componente DCM Xilinx
    - ↳ Componentes COREGEN
- Para realizar simulaciones temporales se necesitan modelos VHDL con información temporal
  - ↳ Los retardos internos están caracterizados en ficheros SDF
  - ↳ VITAL es una librería para realizar simulaciones temporales

# Conclusiones

- El diseño de circuitos específicos tiene mayor importancia actual frente a soluciones programadas
- Circuitos ASIC
  - ↳ Mayor esfuerzo del diseñador
  - ↳ Suministrar al fabricante todos los datos del mismo
  - ↳ Tecnologías caras en comparación a los disp. programables
  - ↳ El precio disminuye con grandes producciones
- Dispositivos Configurables
  - ↳ Simplifican y acortan el proceso de diseño
  - ↳ Se posicionan entre las soluciones programadas y los ASICs en términos de velocidad
  - ↳ No existe relación con el proceso de fab.
  - ↳ Disminuye tiempo al mercado.
  - ↳ Útil para prototipado rápido.