

# Arquitectura de las FPGA

# Contenido

- Descripción de los distintos bloques internos en las FPGAs.
- Jerarquía de memorias y sus recursos asociados.
- Recursos de I/O disponibles
- Recursos específicos de hardware disponibles.
- Arquitectura básica de un SoC (Procesador + Lógica Programable)

# Temario

- **Introducción a FPGA**
- Recursos de Lógica
- Recursos de I/O
- Recursos de Memoria y DSP48
- Conversor ADC
- Recursos de Temporización
- Introducción a SoC
- Resumen

# Introducción

## ➤ Las FPGAs contienen los siguientes bloques básicos

### – Recursos de Lógica

- Slices (agrupados en “Configurable Logic Blocks” (CLB))
  - Contienen lógica combinacional y registros (FF)
- Memoria
- Multiplicadores por hardware

### – Recursos de Interconexión

- Interconexión entre CLBs
- Bloques IOBs
  - Interface entre la FPGA y el mundo exterior

### – Otros Recursos

- Global clock buffers
- Boundary scan logic (JTAG)

## Ejemplos de distintas familias de FPGAs

ARTIX<sup>7</sup>

KINTEX<sup>7</sup>

VIRTEX<sup>7</sup>

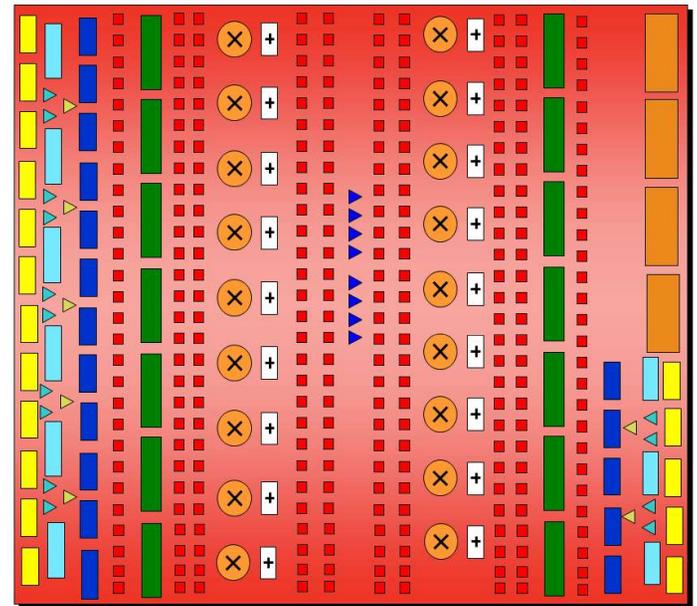
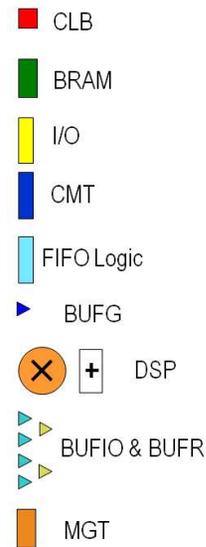
ZYNQ<sup>™</sup>

### Maximum Capability

	Lowest Power and Cost	Industry's Best Price/Performance	Industry's Highest Performance	All Programmable SoC
Logic Cells in K	33 – 215	66 – 478	583 – 1,139	28 – 444
Block RAM in Mb	2 – 12	4 – 34	28 – 68	2 - 27
DSP Slices	90 – 740	240 – 1,920	1,260 – 3,360	80–2,020
Peak DSP Perf. (GMACs)	929	2,845	5,335	2,622
Transceivers	Up to 16	Up to 32	Up to 88	Up to 16
Transceiver Perf. (Gbps)	6.6	12.5	12.5, 13.1 and 28	6.6, 12.5
Memory Perf. (Mbps)	1066	1866	1866	1333
User I/O Pins	106 – 500	285 – 500	350 – 1,100	54 – 400
I/O Voltages	3.3V and below	3.3V and below 1.8V and below	3.3V and below 1.8V and below	3.3V and below 1.8V and below

# Arquitectura de FPGAs

➤ Las distintas familias de FPGAs tienen distintas combinaciones de recursos internos, esto permite escalar los diseños en una relación entre el costo y el desempeño del sistema



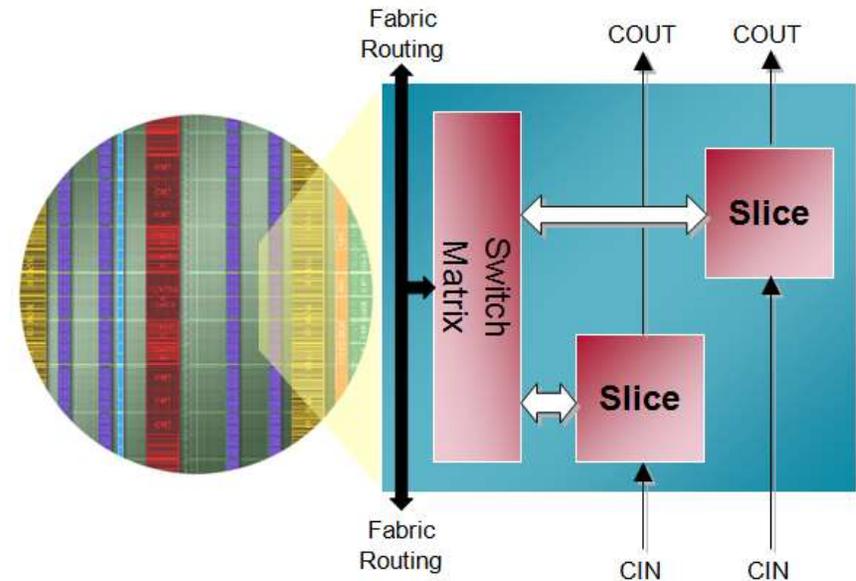
Ejemplo de arquitectura (Artix 7)

# Temario

- Introducción a FPGA
- **Recursos de Lógica**
- Recursos de I/O
- Recursos de Memoria y DSP48
- Conversor ADC
- Recursos de Temporización
- Introducción a SoC
- Resumen

# Bloques Configurable Logic Block (CLB)

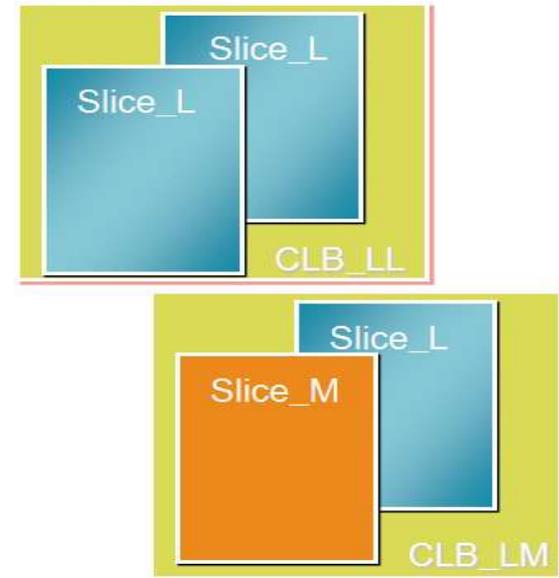
- **Es el recurso principal para implementar sistemas en las FPGAs**
  - Funciones combinacionales
  - Flip-flops
- **Cada CLB contiene dos slices**
- **Se conectan a una matriz de conexiones para rutear las señales a otras partes de la FPGA**
  - Las Carry chain se rutean verticalmente en columnas de un slice a otro
  - Las señales de datos se rutean horizontalmente en filas de un CLB a otro



# Los CLBs pueden tener dos combinaciones de Slices

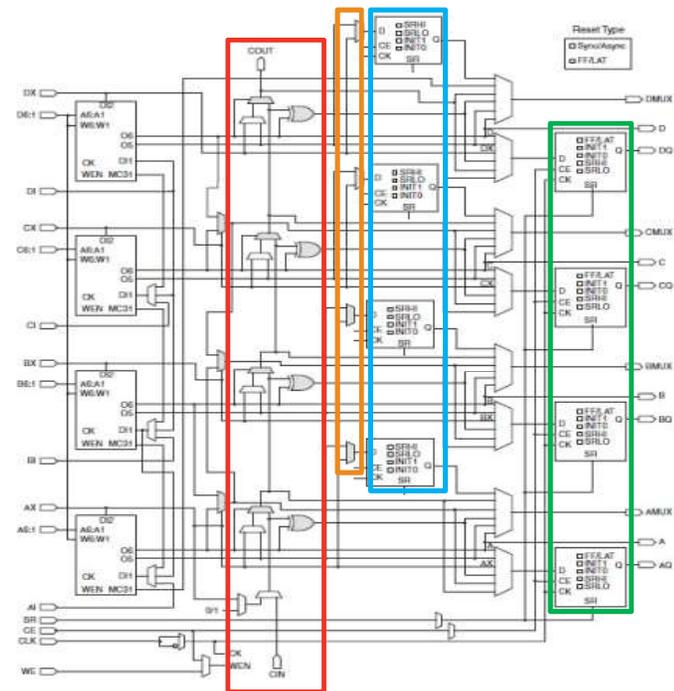
## ► Hay dos tipos de Slices

- SLICEM: slice completo
  - Su LUT puede usarse para lógica y memoria/SRL
  - Tiene varios multiplexores y Carry chain
- SLICEL: para lógica y aritmética solamente
  - Su LUT solo se puede usar para lógica (no memoria)
  - Tiene menos multiplexores y Carry chain



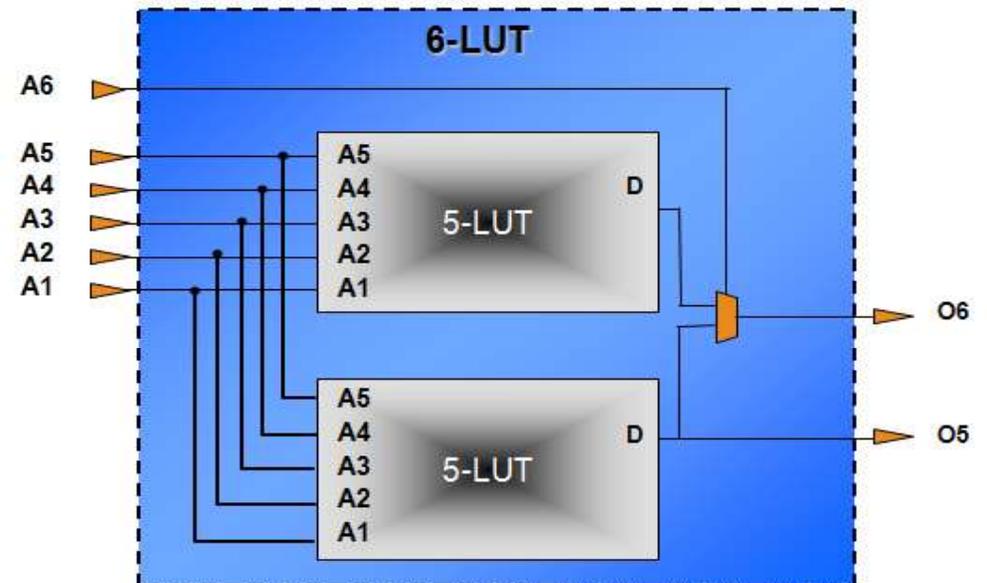
# Recursos de un Slice

- Cuatro Look-Up Tables (LUT) de 6 entradas
- Multiplexores
- Carry chains
- SRL
  - Se pueden poner en cascada para ampliar el tamaño del SRL
- Cuatro flip-flops/latches
  - Cuatro flip-flops adicionales para almacenar resultados



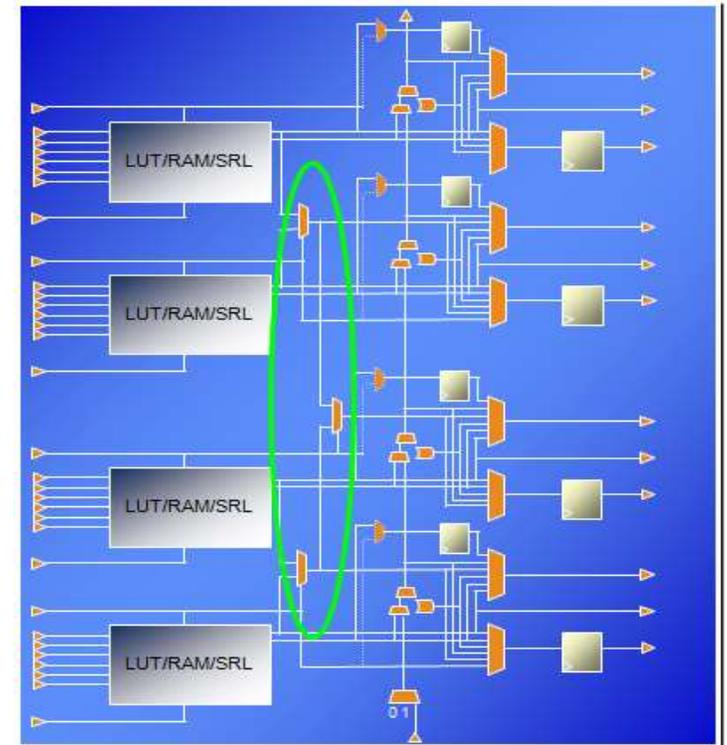
## LUT de 6 entradas con salida Dual

- Las LUTs se pueden descomponer en 2 LUTs de 5 entradas con la entrada en común
  - No hay pérdida de velocidad respecto a una LUT de 6 entradas
- Esto permite implementar una función de 6 variables o 2 funciones de 5 variables



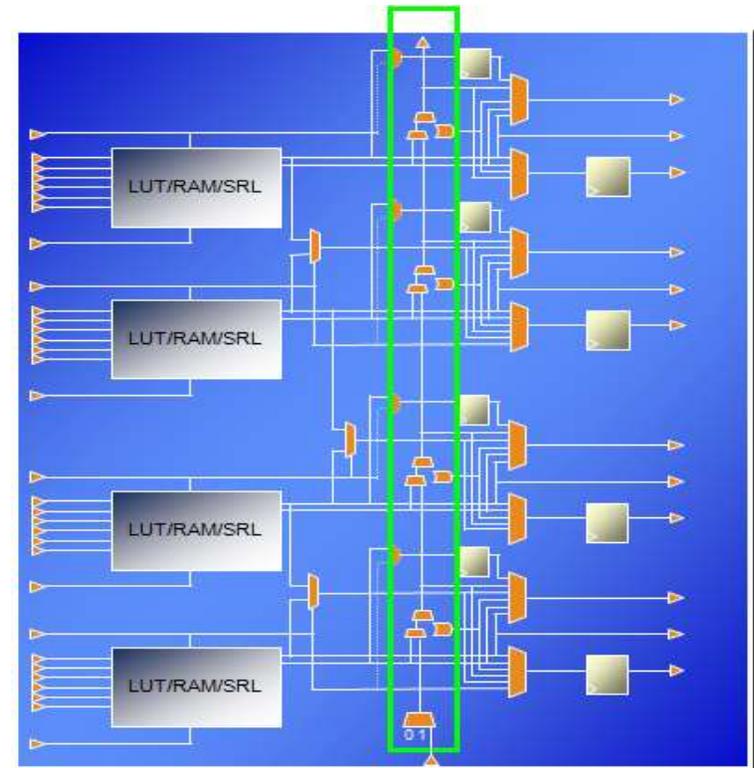
# Multiplexores

- **Cada F7MUX combina las salidas de dos LUTs**
  - Se puede implementar una función de 7 entradas
  - Se puede implementar un multiplexor 8 a 1
- **Cada F8MUX combina las salidas de dos F7MUXs**
  - Se puede implementar una función de 8 entradas
  - Se puede implementar un multiplexor de 16 a 1
- **Estos MUXes son controlados por entradas en el slice**
- **La salida de estos MUXes pueden ser entrada de circuitos combinacionales o flip-flop/latch**



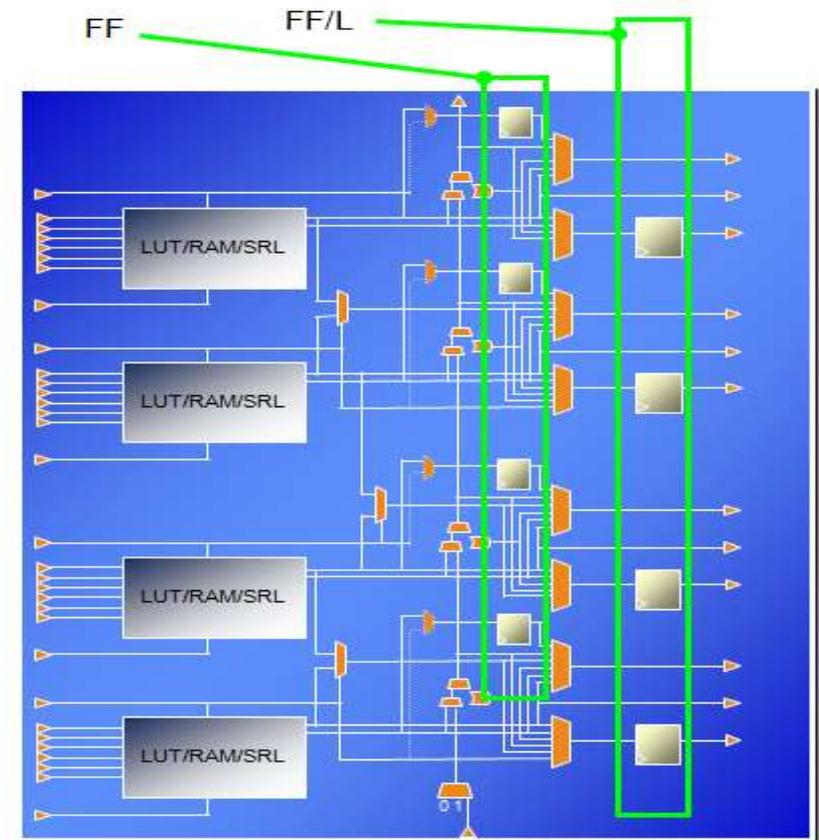
# Carry Chain

- **La lógica de Carry chain permite implementar circuitos muy rápidos de suma y resta**
  - El Carry out se propaga verticalmente a través de los 4 LUTs del slice.
  - La lógica de Carry chain se propaga de un slice al siguiente de la misma columna en el CLB de arriba.
- **Lógica de Carry look-ahead**
  - La lógica combinacional de Carry look-ahead se divide en los cuatro LUTs del slice
  - Esto permite implementar lógica de Carry en cascada entre los slices



## Flip-Flops de Slice y Flip-Flop/Latches para registro de resultados

- **Cada slice tiene 4 flip-flop/latches (FF/L)**
  - Pueden ser configurados como flip-flops o como latches
  - La entrada D puede venir de la salida O6 de la LUT, de la Carry chain, del multiplexor, o de las salidas de slice AX/BX/CX/DX
- **Cada slice también tiene 4 flip-flops (FF)**
  - La entrada D puede venir de la salida O5 o de la entrada AX/BX/CX/DX
    - Estos FF no tienen acceso a la Carry chain, los multiplexores o la entrada del slice
- **Si uno de los FF/L se configura como latch, los otros sólo pueden configurarse también como latches, no como FF**



## Características de los Flip-Flop del Slice

### ➤ Entradas tipo D

- Con salida Q solamente

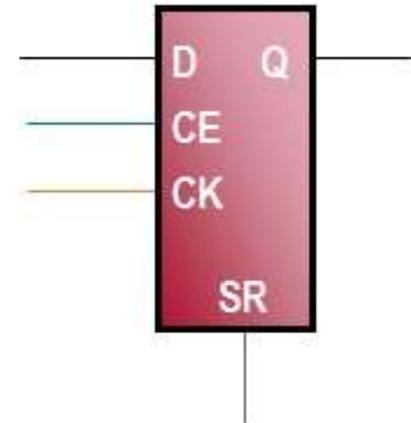
### ➤ Entrada de reloj (CK)

- El reloj puede invertirse, pero esto afecta a los 4 FF (todos los FF del slice deben usar el CK invertido).

### ➤ Entrada de habilitación (CE)

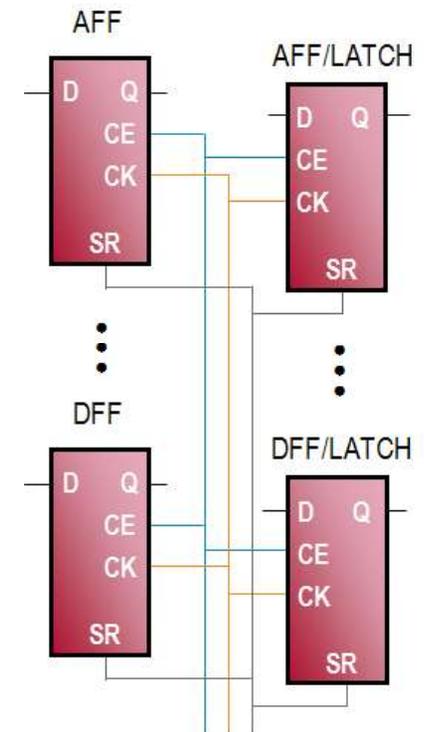
### ➤ Entradas de set/reset

- Estas entradas pueden ser sincrónicas o asincrónicas
- Configuran la salida del flip-flop a un valor predeterminado



# Conjuntos de Control

- **Tanto los flip-flops como los flip-flop/latches comparten las mismas señales CK, SR, y CE**
  - A estas señales se las denomina el “conjunto de control” de los flip-flops
  - Las señales CE y SR son activas en alto
  - La señal CK puede invertirse a la entrada del slice
- **Si uno de los flip-flop usa CE, todos los otros FF usan el mismo CE**
  - La señal CE habilita la señal de reloj a la entrada del slice
  - Esto reduce el consumo
- **Si uno de los flip-flop usa las señales SR, los otros FF también usan el mismo SR**
  - El valor de reset para cada flip-flop se puede configurar individualmente



## SLICEM Usado como memoria RAM

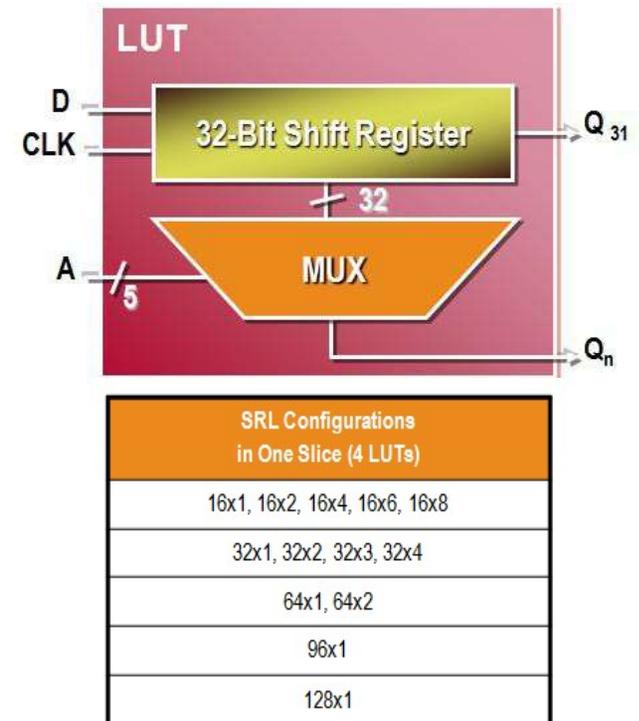
- **Los slices del tipo SLICEM se pueden usar como elementos de memoria**
- **La escritura es sincrónica, la lectura es asincrónica**
  - Se puede convertir en sincrónica utilizando los FF disponibles en el slice
- **Distintas configuraciones de memoria:**
  - Single port
    - Una LUT6 = 64x1 o 32x2 RAM
    - En cascada hasta 256x1 RAM
  - Dual port (D)
    - 1 puerto de lectura/escritura + 1 puerto de solo lectura
  - Dual port simple (SDP)
    - 1 puerto de solo escritura + 1 puerto de solo lectura
  - Quad-port (Q)
    - 1 puerto de lectura / escritura + 3 puertos de lectura solamente

Single Port	Dual Port	Simple Dual Port	Quad Port
32x2	32x2D	32x6SDP	32x2Q
32x4	32x4D	64x3SDP	64x1Q
32x6	64x1D		
32x8	64x2D		
64x1	128x1D		
64x2			
64x3			
64x4			
128x1			
128x2			
256x1			

**Each Port Has Independent Address Inputs**

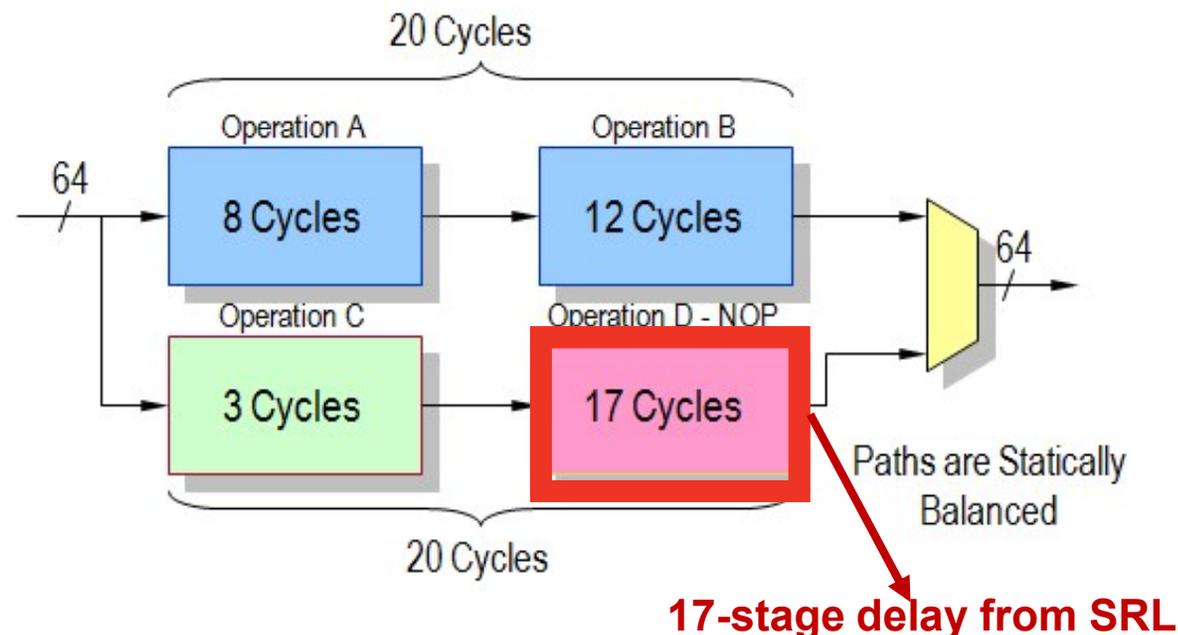
# SLICEM Usado como registro de desplazamiento de 32-bit

- **SRL = Shift Register Lut**
- **Distintas configuraciones**
  - Registros de tamaño variable
  - FIFOs sincrónicas
  - Memoria tipo Content-Addressable Memory (CAM)
  - Generador de patrones
  - Compensador por retardos o latencias
- **Se pueden poner en cascada hasta un registro de 128x1 en un slice**



## Ejemplo de LUT como registro de desplazamiento

- Se realizan dos operaciones en paralelo de 20 ciclos de duración, una de ellas se descompone en dos suboperaciones (A y B), y la otra es una sola operación (C)
- Es necesario agregar una operación D (NOP), lo que implica agregar 17 etapas de pipeline (17 ciclos de ejecución) de 64 bits cada una
  - Son 1,088 flip-flops (o sea 136 slices) o 64 SRLs (o sea 16 slices)



# Temario

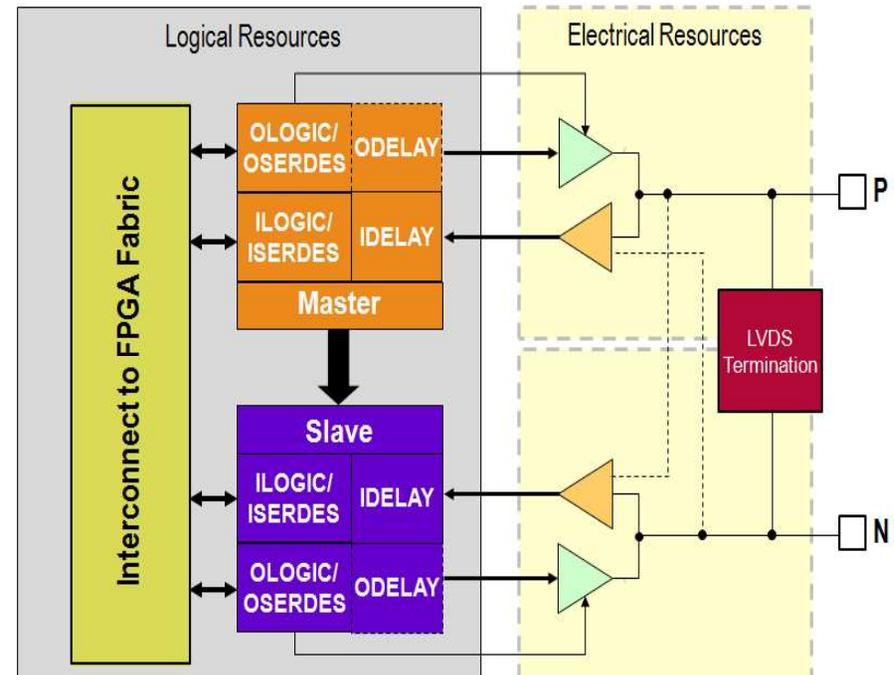
- Introducción a FPGA
- Recursos de Lógica
- **Recursos de I/O**
- Recursos de Memoria y DSP48
- Conversor ADC
- Recursos de Temporización
- Introducción a SoC
- Resumen

## Requisitos de las interfaces de I/O

- **Velocidades muy altas manteniendo la integridad de la señal**
  - Operación Source-synchronous (clock forwarding)
  - Operación System-synchronous (common systems clock)
  - Terminaciones en las líneas de transmisión para evitar reflexiones de señal
- **Generar y recibir datos en buses paralelos muy anchos**
  - Compensar por bus skew y errores de temporización de reloj
  - Conversión entre datos serie y paralelo
  - Conseguir bit rates muy altos (> 1 Gbps)
- **Interfaces Single Data Rate (SDR) o Double Data Rate (DDR)**
- **Interfaces a distintos estándares**
  - Diferentes voltajes, capacidades de bus y protocolos

# FPGA I/O

- **Amplio rango de voltajes**
  - 1.2V a 3.3V
- **Soporte a distintos estándares de I/O**
  - Single ended y diferencial
  - Entradas referenciadas
  - Capacidad 3-state
- **Alta performance**
  - Hasta 1600 Mbps LVDS
  - Hasta 1866 Mbps single-ended para DDR3
- **Interface a memorias**
  - Soporte en Hardware para QDRII+ y DDR3
- **Impedancia controlada digitalmente**
- **Opciones para reducción de consumo**



# Distintos tipos de I/O

## ➤ Dos tipos de I/O:

### – High Range (HR)

- Soporta estándares de I/O con voltajes Vcco hasta 3.3V

### – High Performance (HP)

- Soporta estándares de I/O con voltajes Vcco hasta 1.8V solamente

I/O Types	Artix-7 Family	Kintex-7 Family	Virtex-7 Family	Virtex-7 XT/HT Family
High Range	All	Most	Some	
High Performance		Some	Most	All

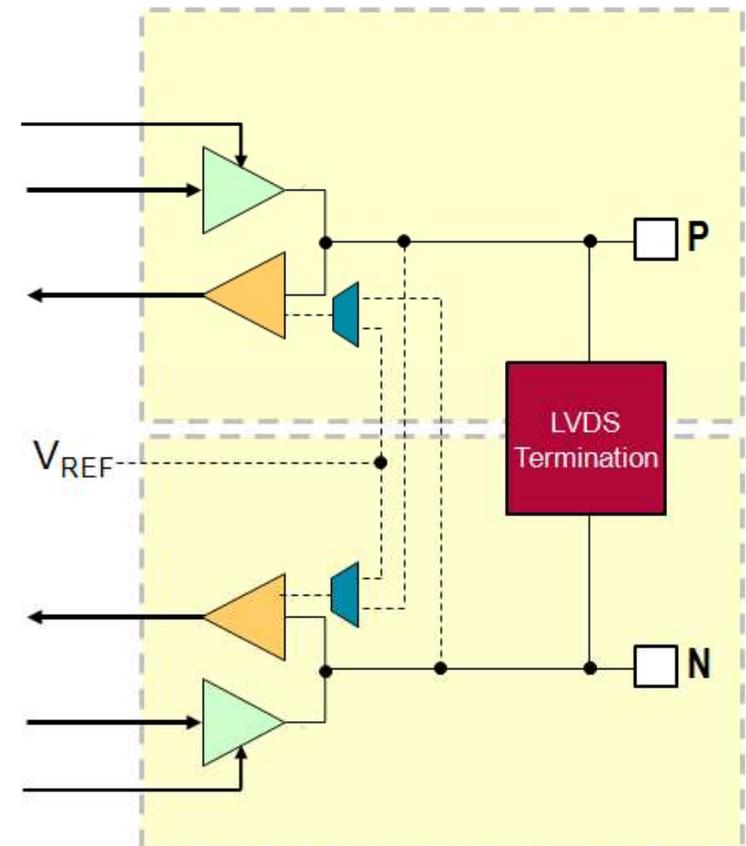
## Características eléctricas de la interface I/O

### ➤ Los pines P y N se pueden configurar como:

- Señales single-ended individuales o
- Pares diferenciales

### ➤ Los receptores pueden ser CMOS estándar o comparadores de voltaje

- Cuando son CMOS estándar
  - 0 lógico cuando esta cerca de tierra
  - 1 lógico cuando esta cerca de  $V_{CC0}$
- Referenciado a  $V_{REF}$ 
  - 0 lógico cuando esta por debajo de  $V_{REF}$
  - 1 lógico cuando esta por encima de  $V_{REF}$
- Diferencial
  - 0 lógico cuando  $V_P < V_N$
  - 1 lógico cuando  $V_P > V_N$



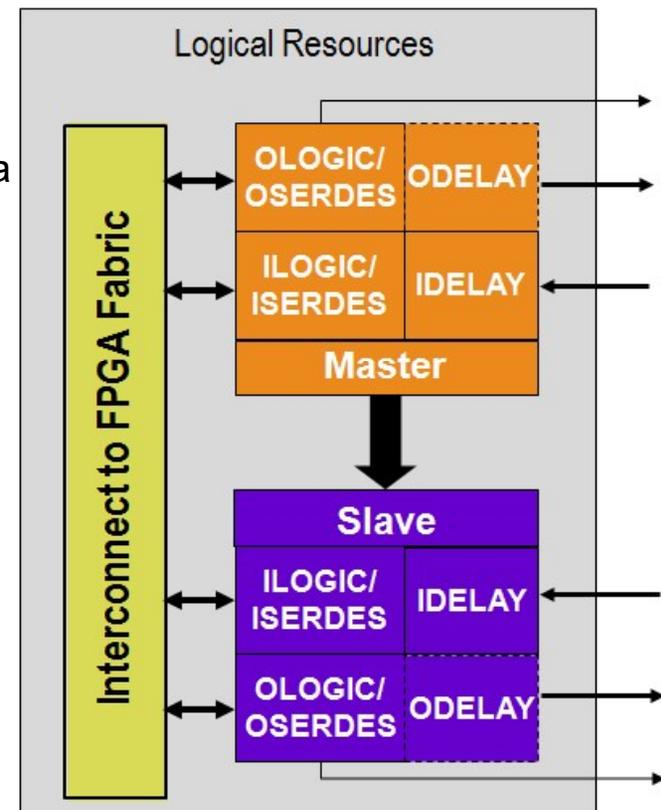
## Recursos asociados a la interface de I/O

### ➤ Dos bloques de lógica por cada par I/O

- Master y slave
- Pueden operar en forma independiente o concatenada

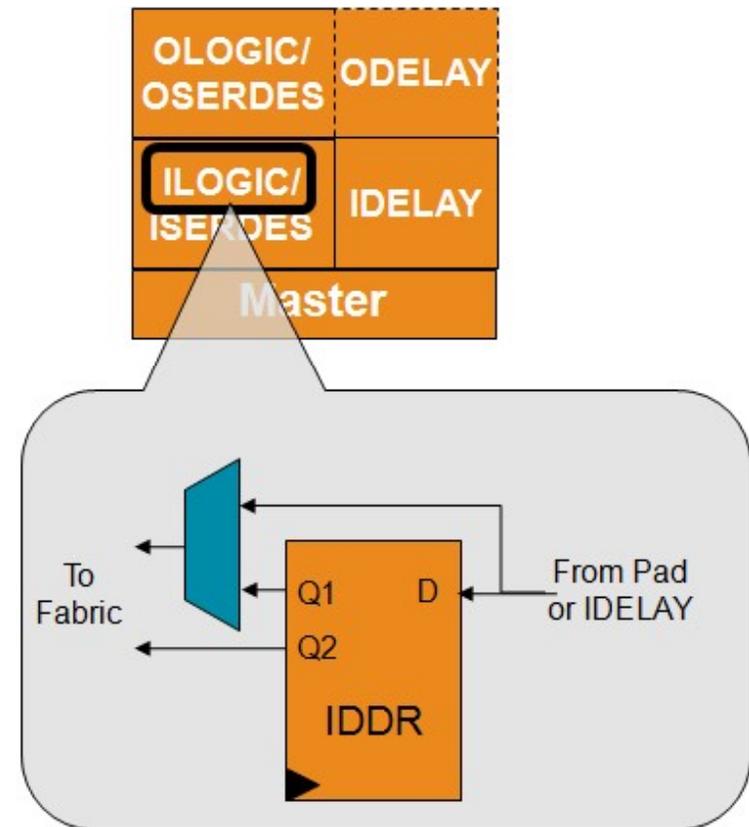
### ➤ Cada bloque contiene:

- ILOGIC/ISERDES (input logic/input serial)
  - SDR, DDR, o lógica de entrada de alta velocidad
- OLOGIC/OSERDES (output logic/output serial)
  - SDR, DDR, o lógica de salida de alta velocidad
- IDELAY
  - Retardo de entrada configurable
- ODELAY
  - Retardo de salida configurable
- Estos últimos disponibles solo en interfaces HP



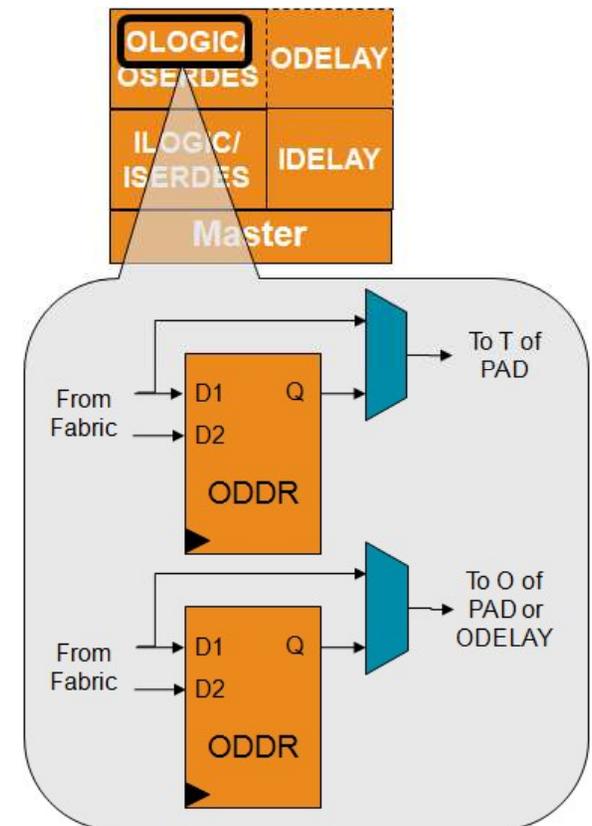
## ILOGIC: Entrada SDR y lógica DDR

- **Dos tipos de bloques ILOGIC**
  - ILOGICE2 para bancos High Performance
  - ILOGICE3 para bancos High Range
- **Las entradas al bloque ILOGIC provienen del receptor**
  - Directamente o a través del bloque IDELAY
- **Las salidas se conectan directamente a la lógica interna de la FPGA (FPGA fabric)**
  - Directamente (lógica sin temporización) o
  - A través del IDDR
    - En modo SDR en el flanco ascendente o descendente del reloj
    - En modo DDR en ambos flancos de reloj
      - También se pueden usar dos relojes con 180° de desfase



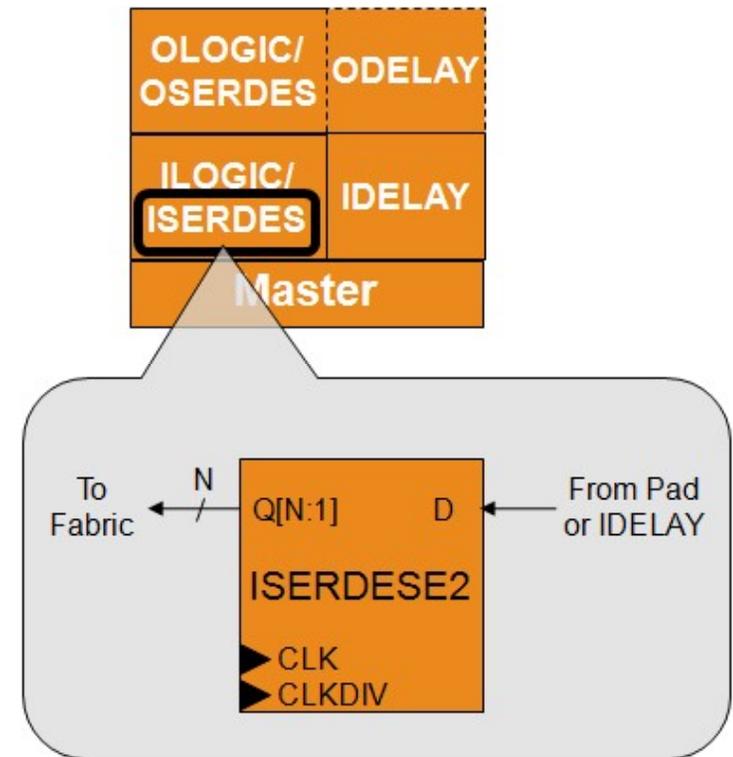
## OLOGIC: Salida SDR y lógica DDR

- **OLOGICE2** para bancos HP, **OLOGICE3** para bancos HR
- La salida del bloque **OLOGIC** se conecta directamente a la etapa de salida o a través del bloque **ODELAY**
  - El bloque **ODELAY** solo esta disponible en los bancos HP
- La salida es controlada directamente desde la lógica interna de la FPGA
  - Directamente, a través de un flip-flop SDR o a través del **ODDR** usando ambos flancos de reloj
- **Cada bloque OLOGIC tiene dos ODDR**
  - Uno para controlar los datos a la etapa de salida
  - Otro para controlar la habilitación de la lógica 3-state
  - Ambos **ODDR** se manejan con el mismo reloj y reset
  - El reloj puede ser en el mismo flanco o en flancos opuestos



## ISERDES: Conversor Serie a Paralelo de Entrada

- **Reciben datos desde el pad de entrada o el bloque IDELAY**
  - D esta controlada por un reloj de alta velocidad (CLK)
  - Puede ser SDR o DDR
- **Envía datos deserializados a la lógica interna de la FPGA**
  - Q esta controlada por un reloj de baja velocidad (CLKDIV)
- **CLK y CLKDIV deben estar en fase**
- **La deserialización de datos puede ser:**
  - Single data rate: 2, 3, 4, 5, 6, 7, 8
  - Double data rate: 4, 6, 8
- **Se pueden poner en cascada para mayores ratios**
  - Double data rate: 10, 14
- **Tienen lógica BITSLIP para framing**



# OSERDES: Conversor Paralelo a Serie de Salida

## ➤ Serializan los datos de salida al pad o bloque ODELAY

- Q esta controlada por un reloj de alta velocidad (CLK)
- Puede ser SDR o DDR

## ➤ Los datos provienen de la lógica interna

- D esta controlada por un reloj de baja velocidad (CLKDIV)

## ➤ CLK y CLKDIV deben estar en fase

## ➤ Serializa datos:

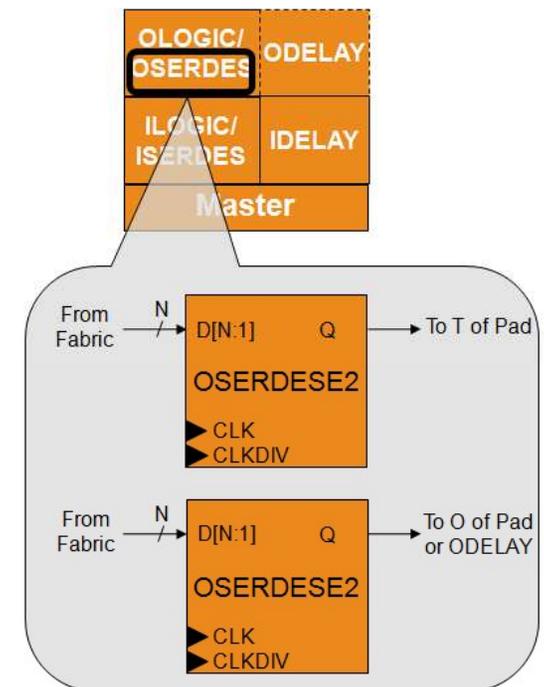
- Single data rate: 2, 3, 4, 5, 6, 7, 8
- Double data rate: 4, 6, 8

## ➤ Se pueden poner en cascada para mayores ratios

- Double data rate: 10, 14

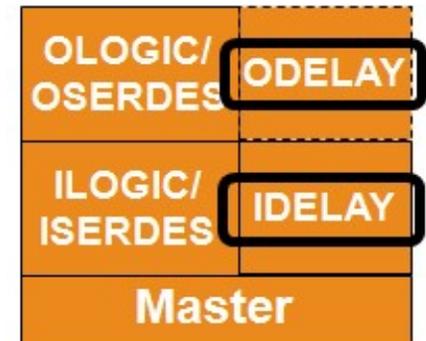
## ➤ Cuando se usa un serializador con 3-state , tanto los datos como la lógica 3-state debe ser de 4 bits

- Los relojes son compartidos por ambos serializadores



## IDELAY y ODELAY

- **Hay líneas de retardo separadas IDELAY y ODELAY**
  - IDELAY esta disponible en los bancos HR y HP
  - ODELAY solo esta disponible en los bancos HP
- **Los elementos de la línea de retardo se controlan mediante el bloque IDELAYCTRL**
  - El retardo es independiente del proceso de fabricación, de la temperatura y del voltaje
- **Los bloques IDELAY y ODELAY son muy similares**
- **La cantidad de etapas se puede acceder desde la lógica de la FPGA**
  - Se puede monitorear, incrementar, decrementar, o fijar . Puede ser de 0 a 31 etapas
- **El reloj de referencia puede ser de 200 MHz en todos los dispositivos; y puede ser de 300 MHz en los dispositivos mas rápidos**
  - Esto genera retardos de 78 ps o 52 ps por etapa

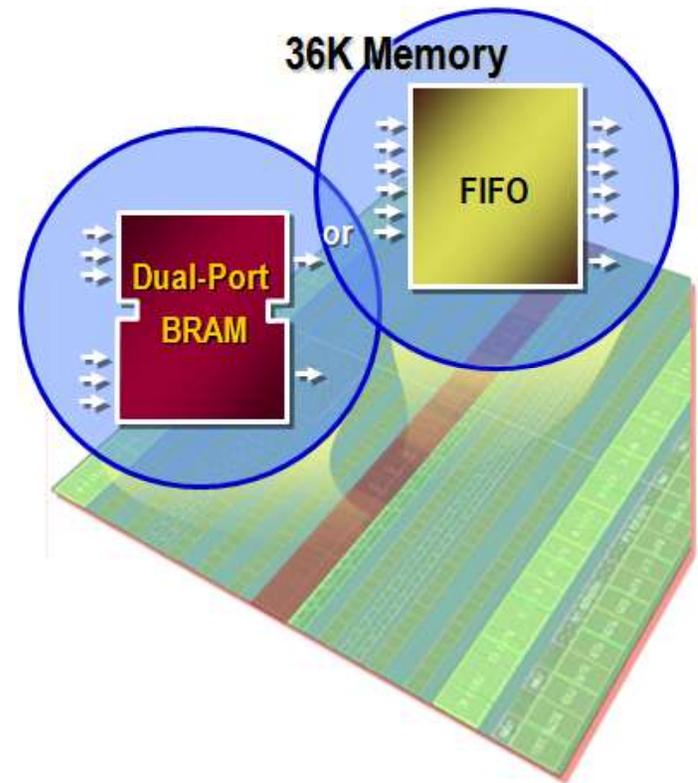


# Temario

- Introducción a FPGA
- Recursos de Lógica
- Recursos de I/O
- **Recursos de Memoria y DSP48**
- Conversor ADC
- Recursos de Temporización
- Introducción a SoC
- Resumen

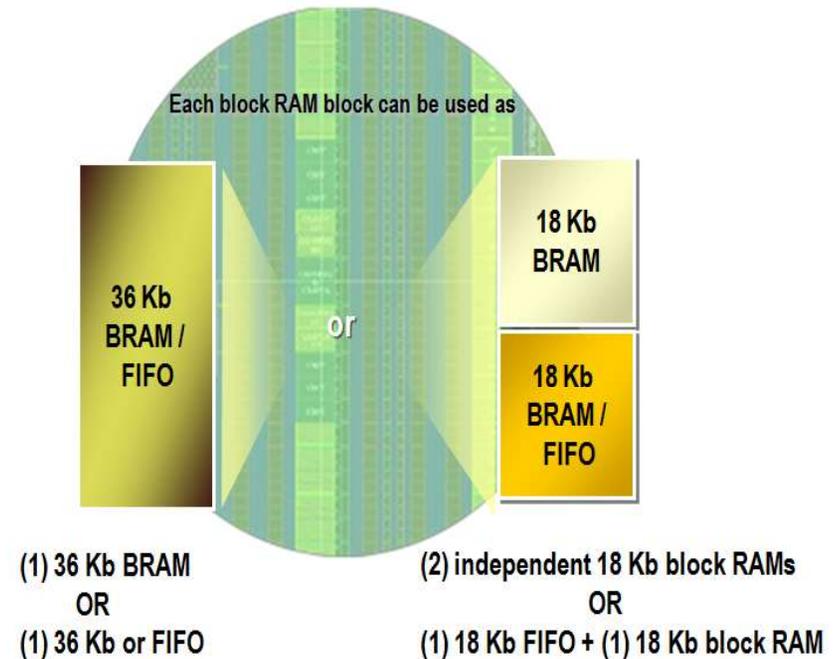
## Bloques RAM y FIFO

- Este bloque es el mismo en los distintos miembros de la familia
- Son de operación completamente sincrónica
  - Todas las salidas tienen registros latch
- Opcionalmente hay un registro interno para pipeline, para lograr mayores frecuencias de operación
- Los datos se pueden acceder a través de dos puertos independientes
  - Las direcciones, reloj, y señales de control son independientes en cada puerto
  - Los tamaños de datos también son independientes por puerto



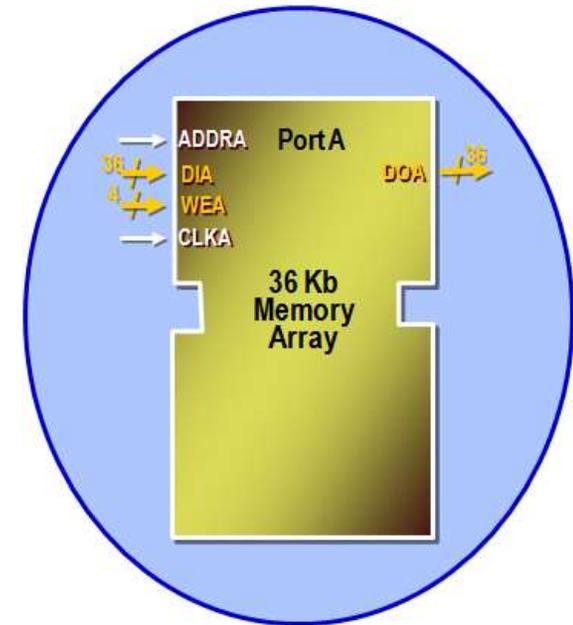
## Bloques RAM and FIFO

- Hay distintas opciones de configuración:
  - True dual-port, simple dual-port, single-port
- Hay lógica integrada para ponerlos en cascada
- Se puede realizar escrituras en tamaño  $B_y$  en buses anchos
- Lógica de control integrada para FIOS rápidas
- Códigos de corrección de Hamming de 64 72-bit por hardware
- Alimentación Vbram separada e independiente



## Block RAM Single-Port

- **Un único puerto de lectura/escritura**
  - Señales Reloj: CLKA, Dirección: ADDRA, Habilitación de escritura: WEA, Escribir datos: DIA, Leer datos: DOA
- **Configuración en bloques de un total de 36-kbit**
  - 32k x 1, 16k x 2, 8k x 4, 4k x 9, 2k x 18, 1k x 36
- **Configuración en bloques de un total de 18-kbit**
  - 16k x 1, 8k x 2, 4k x 4, 2k x 9, 1k x 18, 512 x 36
- **Modo de escritura configurable**
  - WRITE\_FIRST: Los datos escritos en DIA aparecen en DOA
  - READ\_FIRST: El contenido de la RAM en ADDRA se muestra en DOA al momento de escribir en DIA
  - NO\_CHANGE: El valor de DOA se mantiene durante las escrituras
- **Opcionalmente puede haber un registro de salida (DOA\_REG=1)**



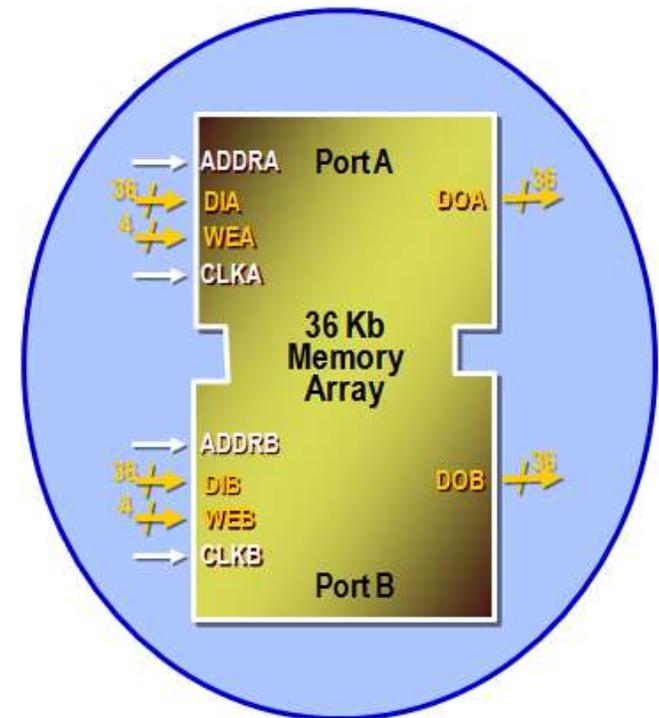
# Bloque RAM Dual-Port

## ➤ Dos puertos separados de lectura/escritura

- Cada puerto tiene señales separadas de reloj, dirección, entrada de datos, salida de datos y habilitación de escritura
  - Los relojes de cada puerto pueden ser asincrónicos entre si.
- Los dos puertos pueden tener diferentes tamaños
  - Están las mismas configuraciones que en Single Port
- Ambos puertos pueden tener distintos modos de escritura

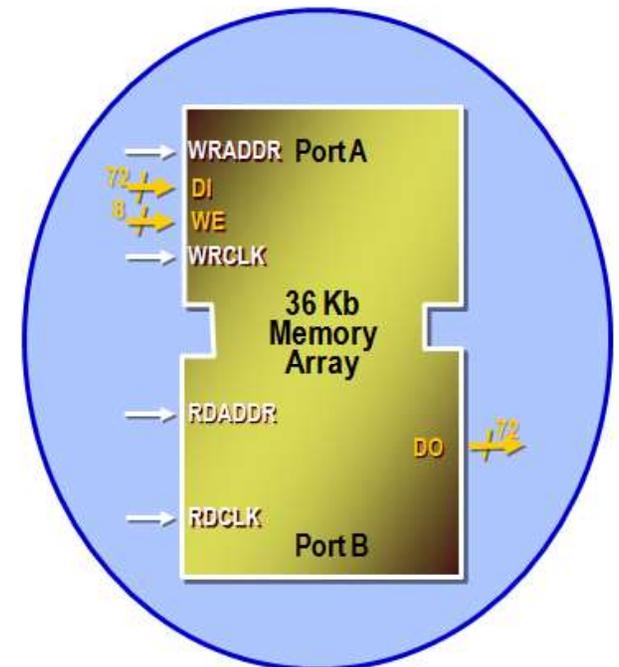
## ➤ No hay contención cuando ambos puertos acceden a la misma dirección

- Excepto que si están temporizados por el mismo reloj, y el puerto que se escribe es READ\_FIRST, entonces el puerto que se lee obtiene los datos anteriores



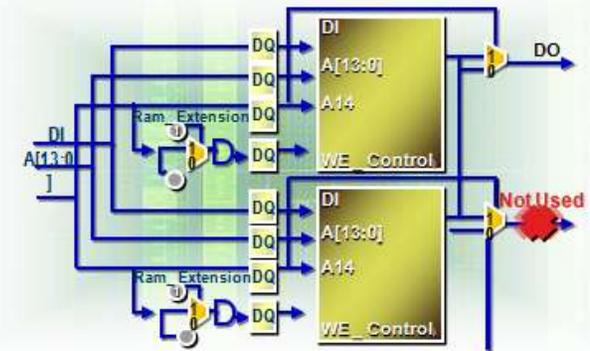
## Bloque RAM Dual-Port Simple

- **Un puerto de lectura y un puerto de escritura**
  - Cada puerto tiene reloj y dirección separados
- **En configuraciones de 36-kbit, uno de los dos puertos debe ser de 72 bits de ancho**
  - El otro puerto puede ser x1, x2, x4, x9, x18, x36, o x72
- **En configuraciones de 18-kbit uno de los dos puertos debe ser de 36 bits de ancho**
  - El otro puerto puede ser x1, x2, x4, x9, x18, o x36



## Bloques RAM en cascada

- Hay lógica integrada para armar bloques de 64Kx1
  - Esto permite poner en cascada dos bloques RAM adyacentes verticalmente sin usar lógica adicional de los CLB
- Se pueden poner en cascada para armar bloques mas grandes:
  - 128Kb, 256Kb, 512Kb...
  - Pero se necesita lógica de los CLB
    - Y es un poco mas lenta que los bloques de 64kbx1
  - Para tamaños de palabra mayores se ponen bloques en paralelo.



Example: Cascade 8 block RAMs to build 256-Kb memory

# Memorias FIFO

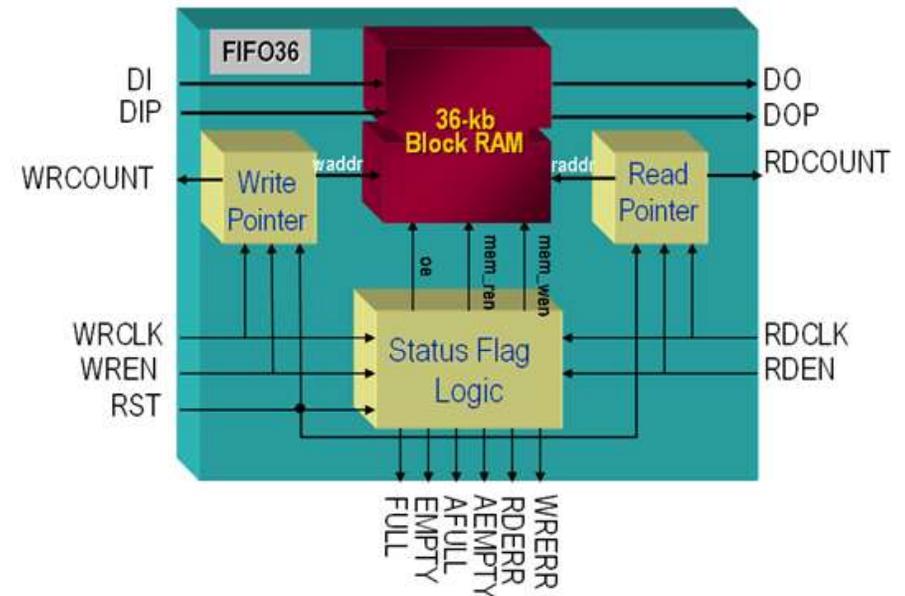
## ➤ Distintas opciones

- Relojes de lectura y escritura sincrónicos o asincrónicos
- Cuatro señales de estado:
  - Lleno, Vacio, casi lleno, casi vacio

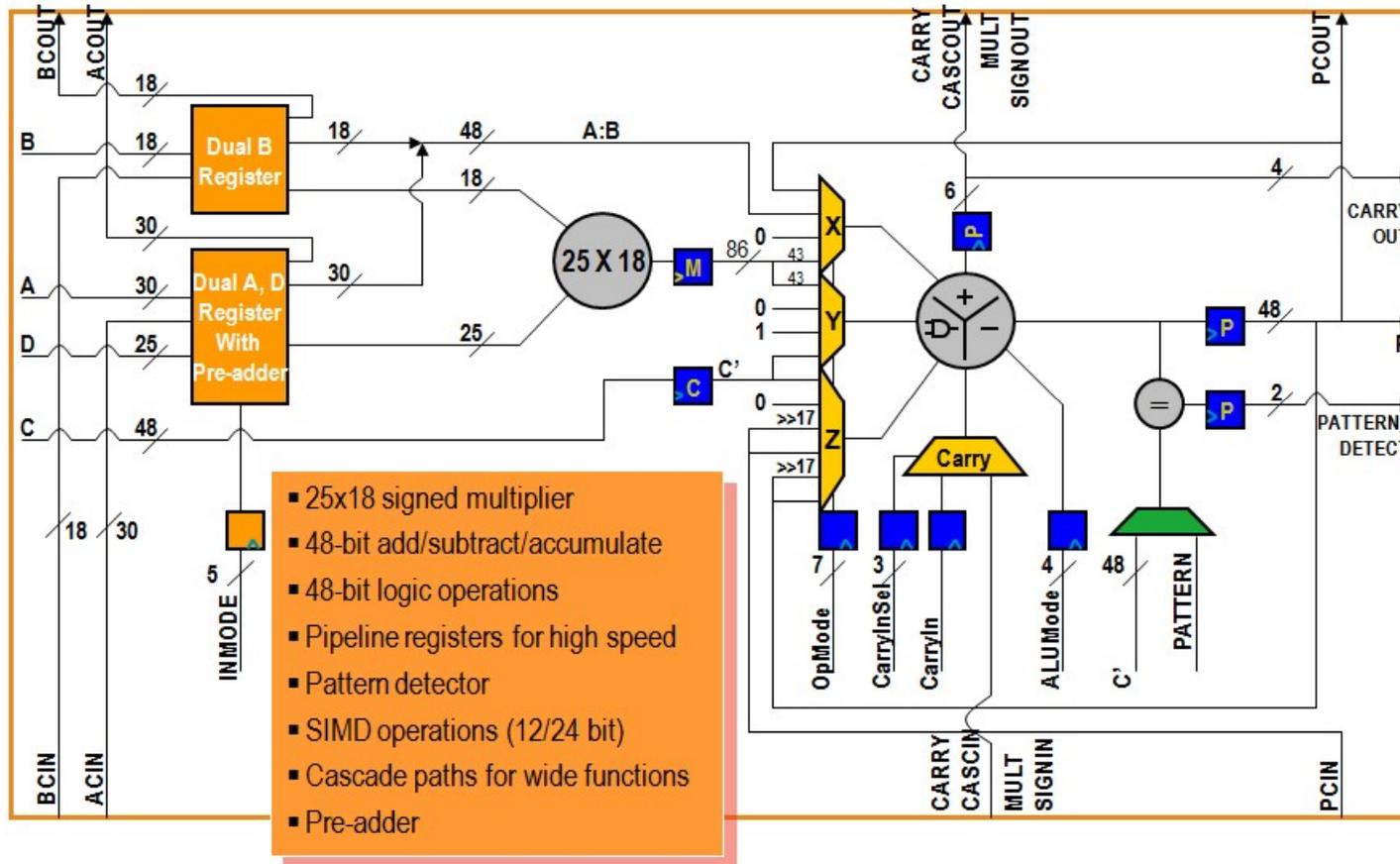
## ➤ Configuraciones:

- Usando block RAM de 36-Kb: 8Kx4, 4Kx9, 2Kx18, 1Kx36, 512x72
- Usando block RAM de 18-Kb: 4Kx4, 2Kx9, 1Kx18, 512x36
- El tamaño de datos de lectura y escritura debe ser el mismo

- Se puede usar corrección de errores integrada cuando el tamaño de palabra es x72



# Bloque DSP48E1



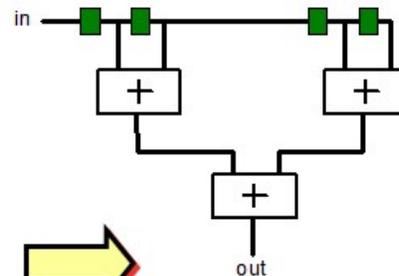
- 25x18 signed multiplier
- 48-bit add/subtract/accumulate
- 48-bit logic operations
- Pipeline registers for high speed
- Pattern detector
- SIMD operations (12/24 bit)
- Cascade paths for wide functions
- Pre-adder

# Uso del bloque DSP48:

**START:** This is the typical adder tree found in many signal processing designs

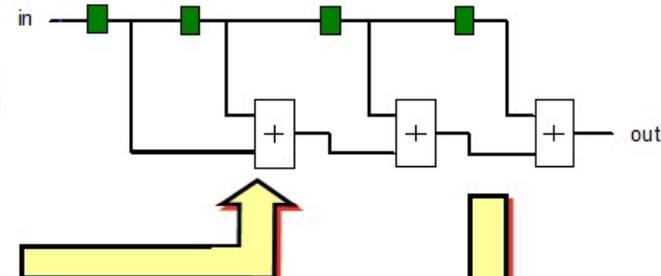
1

**Remove all pipelining** from the tree. This makes it easier to understand and visualize the changes



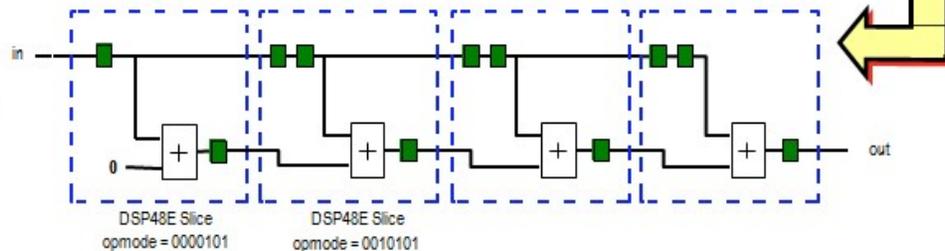
2

**Rearrange the tree.** Notice that functionally has not changed. The diagram has just been redrawn



3

**Pipelining** is required for performance. Adding one in the chain requires one in the data path delay as well. Determining mapping to DSP48E is easy now



# Temario

- Introducción a FPGA
- Recursos de Lógica
- Recursos de I/O
- Recursos de Memoria y DSP48
- **Conversor ADC**
- Recursos de Temporización
- Introducción a SoC
- Resumen

# XADC y AMS

## ➤ XADC es una interface analógica

- ADC dual de 12-bit 1Msps, sensor de temperatura on-chip, 17 entradas, track & hold y acondicionamiento de señal
- Rango de entrada de 1V
- Resolución de conversión de 16-bit
- Ganancia y offset configurable

## ➤ Analog Mixed Signal (AMS)

- Utilizando la lógica de la FPGA se puede configurar el bloque XADC para linealización, calibración, filtrado, y fijación de valor de tensión continua para mejorar el rango dinámico del conversor

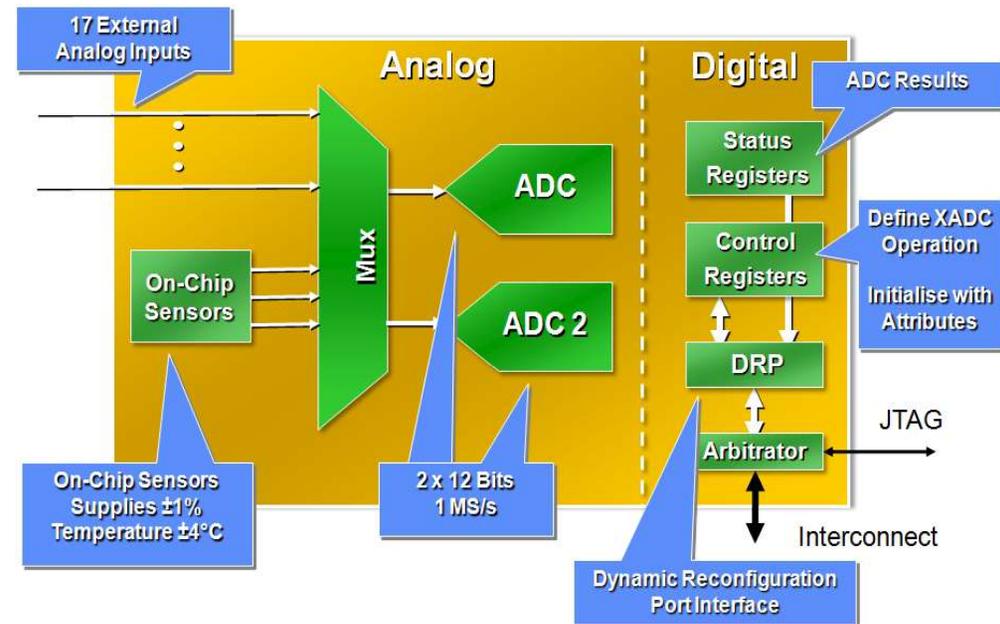
# Bloque XADC

## ➤ Conversión

- Tiempo de conversión 1 uS con sampleo simultaneo en ambos conversores
- Distintos modos de disparo (auto disparo y disparo externo)
- Circuito de track/hold separado para cada ADC. Se puede hacer multiplexado de canales

## ➤ Entradas analógicas

- Entradas diferenciales.
- Soporte para señales unipolares, bipolares y diferenciales



## Bloque XADC

### ➤ Multiplexado y muestreo interno y externo

- Se pueden muestrear las alimentaciones internas y la temperatura del dispositivo
- Se multiplexan los sensores internos y 17 entradas externas
- Se puede controlar un multiplexor analógico externo para reducir el uso de pines

### ➤ Conversión configurable

- El resultado de la conversión se guarda en un registro interno
- Los registros de control configuran la fuente de señal, la frecuencia de muestreo y alarmas
- Los registros están disponibles para la lógica de la FPGA a través del Dynamic Reconfiguration Port (DRP)
- Estos también están disponibles a través de la interface JTAG

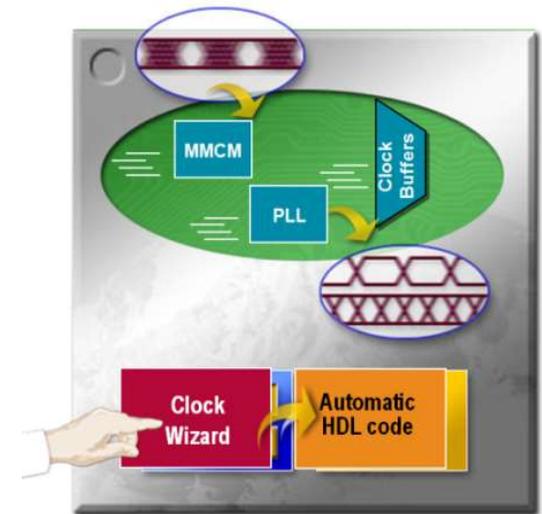
### ➤ Rango de temperatura de $-40^{\circ}\text{C}$ a $+125^{\circ}\text{C}$

# Temario

- Introducción a FPGA
- Recursos de Lógica
- Recursos de I/O
- Recursos de Memoria y DSP48
- Conversor ADC
- **Recursos de Temporización**
- Introducción a SoC
- Resumen

# Gestión del Reloj

- **Bloques Buffer Global**
  - Permiten una distribución del reloj con alto fanout
- **Distribución con bajo skew**
- **Regiones de Reloj**
- **Bloque Clock management tile (CMT)**
  - Un Mixed-Mode Clock Manager (MMCMs) y un Phase Locked Loop (PLL) en cada región de reloj
  - Permiten generar diferentes frecuencias, realizar de-skewing, y filtrado para evitar el jitter
- **Se instancian utilizando el Clocking Wizard**



# Entradas con capacidad de Reloj

- **Todos los diseños sincrónicos requieren una referencia externa de reloj**
- **Hay una entrada con capacidad de reloj en cada banco de I/O**
  - Estas entradas son pines I/O normales con conexiones dedicadas a los recursos de reloj internos.
  - Cada banco de I/O tiene 4 pines con capacidad de reloj
    - 2x Multi-Region Clock Capable (MRCC)
    - 2x Single Region Clock Capable (SRCC)
  - Cada entrada de reloj puede ser usada como una entrada single ended o puede utilizarse junto con la entrada adyacente para formar una entrada diferencial

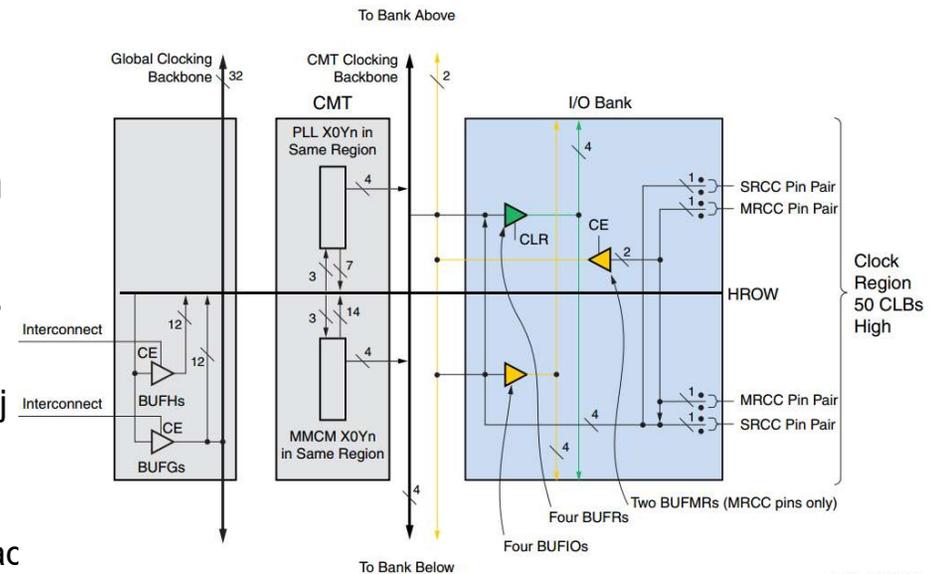
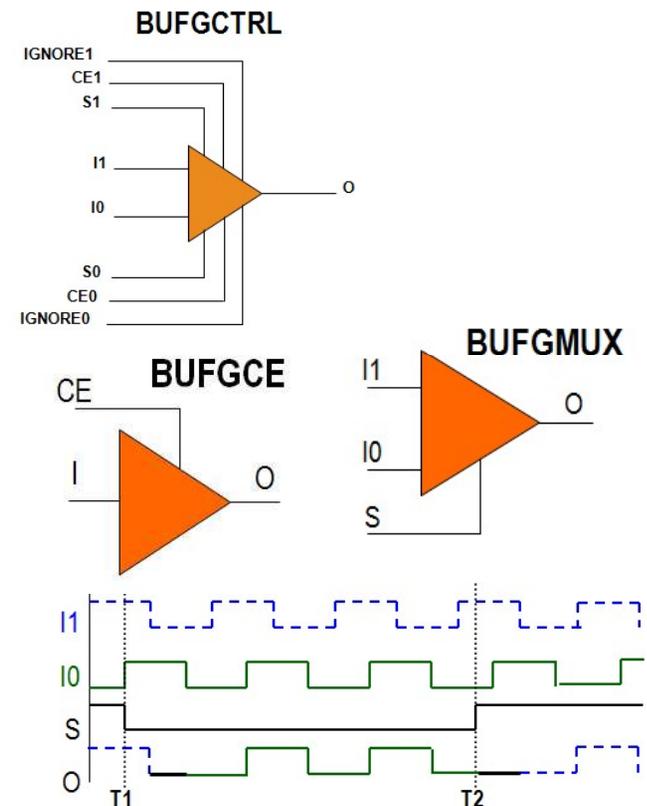


Figure 1-3: Single Clock Region (Right Side of the Device)

# Bloque Global Clock Buffer (BUFGCTRL)

- Los bloques **BUFGCTRL** (o **BUFG**) están en el centro del dispositivo
- Los bloques **BUFGCTRL** se pueden comandar con
  - Una entrada con capacidad de reloj (CCIO)
  - Una salida CMT (clock management tile)
  - El reloj de un transceiver Gigabit
  - Otros bloques BUFG, o BUFR
- Los bloques **BUFGCTRL** permiten:
  - Simple clock buffer (BUFG)
  - Clock buffer con clock switching (BUFGMUX o BUFGMUX\_CTRL)
  - Clock buffer con clock enable (BUFGCE)



# Temario

- Introducción a FPGA
- Recursos de Lógica
- Recursos de I/O
- Recursos de Memoria y DSP48
- Conversor ADC
- Recursos de Temporización
- **Introducción a SoC**
- Resumen

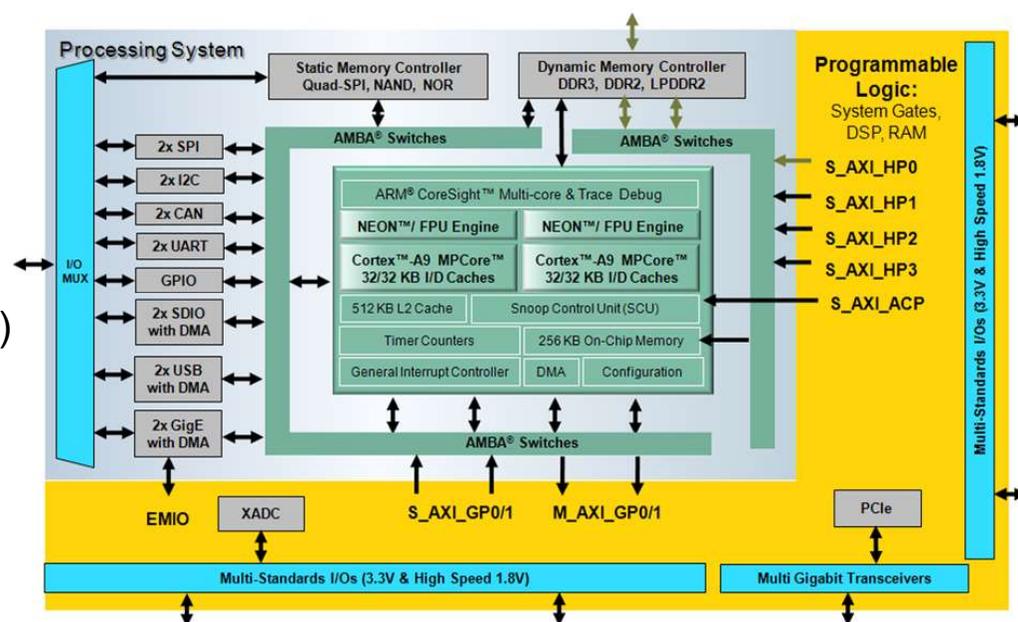
# Zynq-7000

## ➤ Sistema de procesadores basado en tecnología ARM

- Incluye Application Processor Unit (APU)
  - Procesador Dual ARM Cortex™-A9
  - Caches y lógica asociada
- Controladores de memoria integrados
- Periféricos de I/O (serial, I2C, SPI, GPIO, etc)

## ➤ Integración con la lógica interna

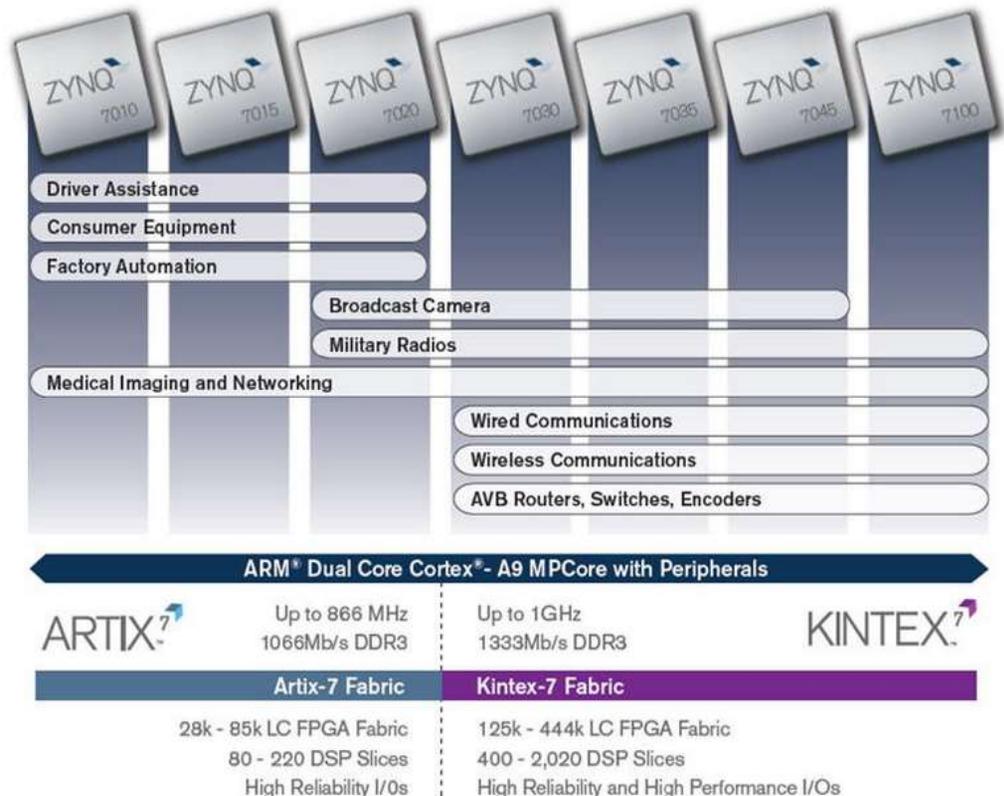
- Permite crear aceleradores por hardware



# Bloques PS (Processor System) y PL (Programmable Logic)

## ➤ Los dispositivos SoC Zynq-7000 tienen dos bloques principales

- PS: Processing system
  - Basado en un procesador Dual ARM Cortex-A9
  - Múltiples periféricos
  - Implementado en silicio (hard core)
- PL: Programmable logic
  - Similar a otros miembros de la familia



# Temario

- Introducción a FPGA
- Recursos de Lógica
- Recursos de I/O
- Recursos de Memoria y DSP48
- Conversor ADC
- Recursos de Temporización
- Introducción a SoC
- **Resumen**

## Resumen

- **Los Slices de la FPGA tienen 4 LUTs de 6 entradas, 8 registros y lógica de Carry**
  - Las LUTs pueden realizar cualquier función combinacional de hasta seis entradas
  - Las LUTs están conectadas a multiplexores dedicados y lógica de Carry
  - Algunas LUTs se pueden configurar como registros de desplazamiento o memorias
  - Los Slices contienen lógica de Carry y los multiplexores MUXF7 y MUXF8
  - Los multiplexores MUXF7 combinan la salida de las LUTs para crear funciones de 7 entradas o multiplexores de 8 entradas
  - Los multiplexores MUXF8 combinan las salidas de los multiplexores MUXF7 para crear funciones de 8 entradas o multiplexores de 16 entradas
  - La lógica de Carry se puede utilizar para implementar funciones rápidas de suma, resta y comparación
- **Los bloques IOBs contienen registros DDR e interfaces SERDES**
- **Los bloques de I/O pueden implementar distintos estándares de señales single ended o diferenciales**

## Resumen

- **Las FPGAs incluyen bloques dedicados de block RAM y Slices DSP**
- **Para la gestión del reloj hay bloques MMCM, PLL, y recursos de ruteo específicos**
- **Se dispone del bloque XADC para señales analógicas**
- **Los dispositivos Zynq-7000 son System on a Chip (SoC) con lógica programable y procesadores ARM**