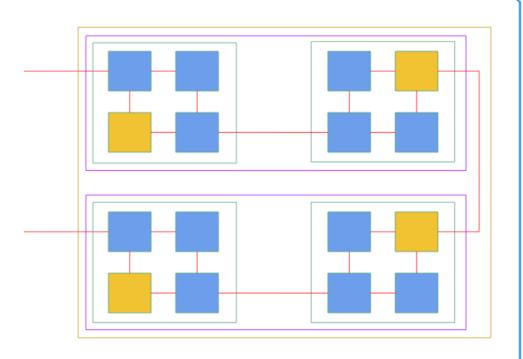


Diseño Bottom-Up



Instanciando componentes

- A medida que desarrollamos y verificamos los bloques básicos de nuestro diseño, podemos instanciarlos y conectarlos para formar arquitecturas más grandes.
- Esto se conoce como diseño "bottom-up"

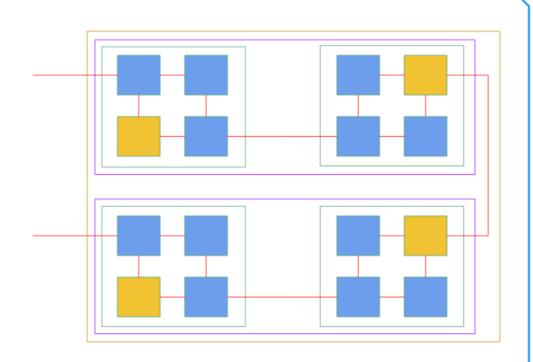






Instanciando componentes

- Podemos instanciar dos tipos de submódulos en nuestro diseño:
 - MÓDULOS en Verilog
 - Desarrollo propio
 - Librería externa (IP)
 - PRIMITIVAS propias del hardware que usemos









Instanciando módulos

```
input wire d,
input wire clk,
input wire rst,
input wire rst,
output reg q
);

FFD.v

// Inluimos el codigo a usar
include "FFD.v"

wire clock;
wire reset;
wire data;
reg out;
//...
FFD mi_flip_flop (.clk(clock),.rst(reset),.d(data),.q(out));
```

defparam NOMBRE_INSTANCIA.PARAMETRO = VALOR





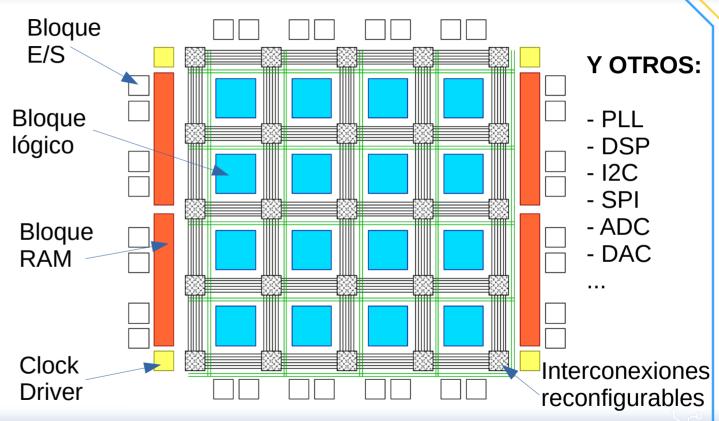
Hardware y Primitivas





Primitivas

Son los elementos de hardware que se encuentran en la FPGA y cuya interconexión da lugar a nuestros circuitos.









Primitivas

Siempre es importante leer la **hoja de datos** de nuestra FPGA para saber de qué recursos de hardware dispondremos.

¿Vale la pena elegir una FPGA con mucha variedad y cantidad de primitivas?¿Cuáles serían las ventajas y desventajas?

VENTAJAS	DESVENTAJAS
Habilitan nuevas funcionalidades en nuestro diseño (DAC/ADC)	Portabilidad
Reducen el consumo de celdas lógicas (memorias, puertos, aritmética)	\$





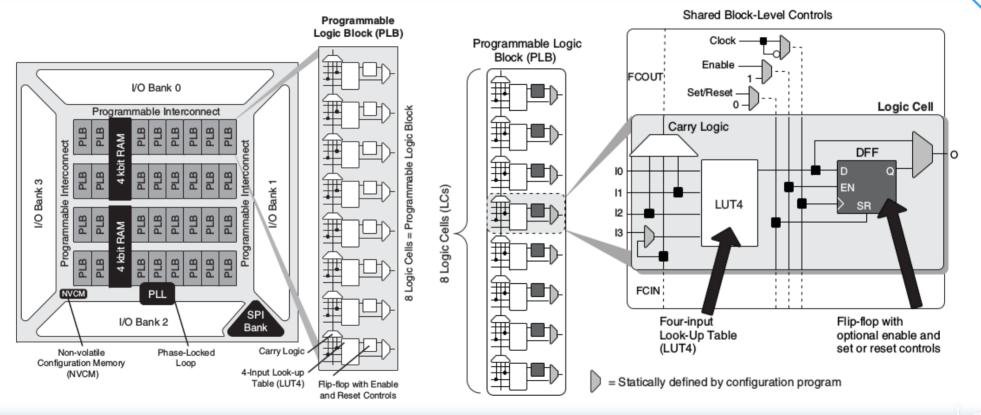
ICE40-HX8K

Es el modelo usado en la EDU-CIAA-FPGA

Table 1-1. iCE40 Family Selection Guide

Part Number	LP384	LP640	LP1K	LP4K	LP8K	HX1K	HX4K	HX8K
Logic Cells (LUT + Flip-Flop)	384	640	1,280	3,520	7,680	1,280	3,520	7,680
RAM4K Memory Blocks	0	8	16	20	32	16	20	32
RAM4K RAM bits	0	32K	64K	80K	128K	64K	80K	128K
Phase-Locked Loops (PLLs)	0	0	1 ¹	2 ²	2 ²	1 ¹	2	2
Maximum Programmable I/O Pins	63	25	95	167	178	95	95	206
Maximum Differential Input Pairs	8	3	12	20	23	11	12	26
High Current LED Drivers	0	3	3	0	0	0	0	0

Floorplan y celdas lógicas

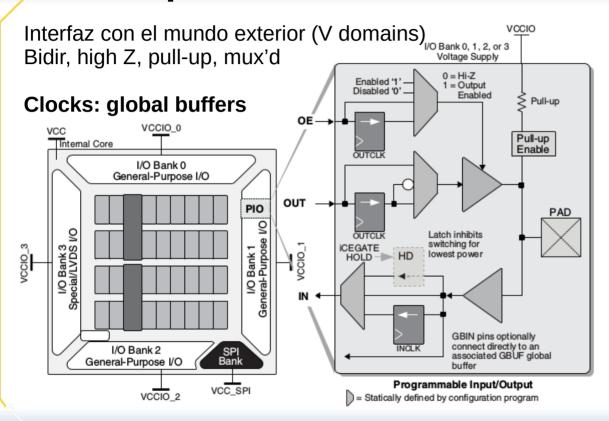


UTN **X** HAEDO

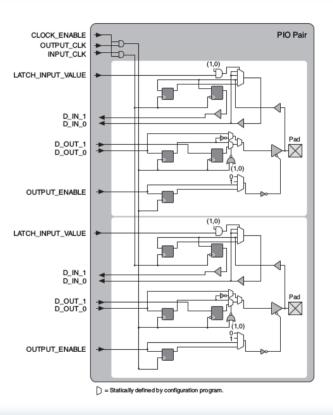




Bloques E/S



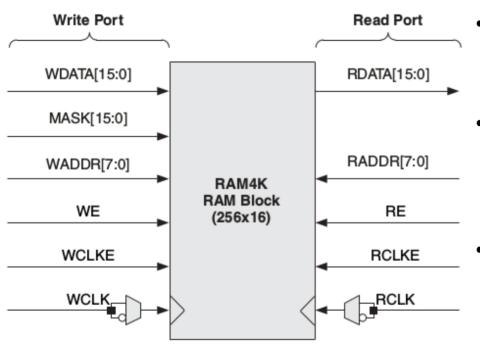
UTN **X** HAEDO







Bloques de memoria RAM



- Almacenar "grandes" cantidades de información en el diseño implicaría consumir una gran cantidad de celdas lógicas.
- Los bloques de memoria en el circuito ofrecen almacenamiento de mayor densidad (menor costo) que usando celdas (más valiosas por su versatilidad).

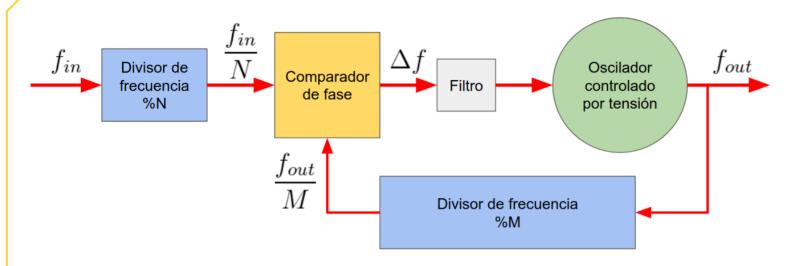
Aplicaciones:

- Mapeo de periféricos a memoria
- Memoria de datos y/o instrucciones para un procesador o softcore en la FPGA





Phase-Locking Loop (PLL)

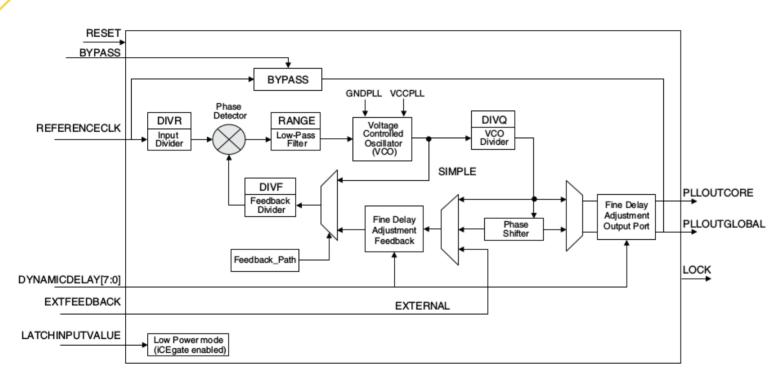


Generan un reloj de salida cuya frecuencia es menor o mayor a la de un reloj de entrada de acuerdo a la configuración de sus divisores de frecuencia internos

$$\Delta f = \frac{f_{in}}{N} - \frac{f_{out}}{M} = 0 \longrightarrow f_{out} = \frac{M}{N} f_{in}$$



Phase-Locking Loop (PLL)



Generan un reloj de salida cuya frecuencia es menor o mayor a la de un reloj de entrada de acuerdo a la configuración de sus divisores de frecuencia internos

iCE40 UltraPlus

Table 2.1. iCE40 UltraPlus Family Selection Guide

Part Number	iCE40UP3K	iCE40UP5K		
Logic Cells (LUT + Flip-Flop)	2800	5280		
EBR Memory Blocks	20	30		
EBR Memory Bits (Kbits)	80	120		
SPRAM Memory Blocks	4	4		
SPRAM Memory Bits (Kbits)	1024	1024		
NVCM	Yes	Yes		
PLL	1	1		
DSP Blocks (MULT16 with 32-bit Accumulator	4	8		
Hardened I ² C, SPI	2,2	2, 2		
HF Oscillator (48 MHz)	1	1		
LF Oscillator (10 KHz)	1	1		
24 mA LED Sink	3	3		
PWM IP Block	Yes	Yes		
Packages, ball pitch, dimension	Total User I/O Count			
30-ball WLCSP, 0.4 mm, 2.15 mm x 2.55 mm	21	21		
48-ball QFN, 0.5 mm, 7.0 mm x 7.0 mm	-	39		





DSP Blocks

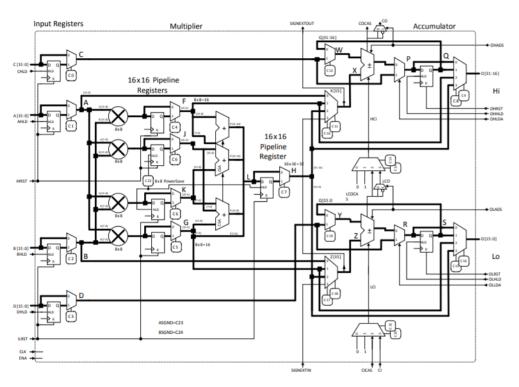


Figure 3.6. sysDSP Functional Block Diagram (16-bit x 16-bit Multiply-Accumulate)

- La síntesis de operaciones matemáticas suele consumir muchas celdas lógicas en función del tipo de aritmética implementada
- Los bloques dedicados a Digital Signal Processing (DSP) alivianan mucho el consumo de recursos.

Otras marcas



Intel® Cyclone® 10 LP Device Overview 683879 | 2022.05.27

Intel Cyclone 10 LP Maximum Resources

Table 2. Maximum Resource Counts for Intel Cyclone 10 LP Devices

Resource		Device									
		10CL006	10CL010	10CL016	10CL025	10CL040	10CL055	10CL080	10CL120		
Logic Elements (LE)		6,272	10,320	15,408	24,624	39,600	55,856	81,264	119,088		
м9К	Block	30	46	56	66	126	260	305	432		
Memory	Capacity (Kb)	270	414	504	594	1,134	2,340	2,745	3,888		
18 × 18 M	18 × 18 Multiplier		23	56	66	126	156	244	288		
PLL		2	2	4	4	4	4	4	4		
Clock		20	20	20	20	20	20	20	20		
Maximum I/O		176	176	340	150	325	321	423	525		
Maximum LVDS		65	65	137	52	124	132	178	230		





Otras marcas

Table 1: 7 Series Families Comparison

Max. Capability	Spartan-7	Artix-7	Kintex-7	Virtex-7	
Logic Cells	102K	215K	478K	1,955K	
Block RAM ⁽¹⁾	4.2 Mb	13 Mb	34 Mb	68 Mb	
DSP Slices	160	740	1,920	3,600	
DSP Performance ⁽²⁾	176 GMAC/s	929 GMAC/s	2,845 GMAC/s	5,335 GMAC/s	
MicroBlaze CPU ⁽³⁾	260 DMIPs	303 DMIPs	438 DMIPs	441 DMIPs	
Transceivers	_	16	32	96	
Transceiver Speed	-	6.6 Gb/s	12.5 Gb/s	28.05 Gb/s	
Serial Bandwidth	-	211 Gb/s	800 Gb/s	2,784 Gb/s	
PCIe Interface	-	x4 Gen2	x8 Gen2	x8 Gen3	
Memory Interface	800 Mb/s	1,066 Mb/s	1,866 Mb/s	1,866 Mb/s	
I/O Pins	400	500	500	1,200	
I/O Voltage	1.2V-3.3V	1.2V-3.3V	1.2V-3.3V	1.2V-3.3V	
Package Options	Low-Cost, Wire-Bond	Low-Cost, Wire-Bond, Bare-Die Flip-Chip	Bare-Die Flip-Chip and High- Performance Flip-Chip	Highest Performance Flip-Chip	

Notes:

- 1. Additional memory available in the form of distributed RAM.
- 2. Peak DSP performance numbers are based on symmetrical filter implementation.
- 3. Peak MicroBlaze CPU performance numbers based on microcontroller preset.







Otras marcas

Spartan-7 FPGA Feature Summary

Table 2: Spartan-7 FPGA Feature Summary by Device

Device	Logic Cells	CLB			Block RAM Blocks ⁽³⁾								
		Slices ⁽¹⁾	Max Distributed RAM (Kb)	DSP Slices ⁽²⁾	18 Kb	36 Kb	Max (Kb)	CMTs ⁽⁴⁾	PCle	GT	XADC Blocks	Total I/O Banks ⁽⁵⁾	Max User I/O
XC7S6	6,000	938	70	10	10	5	180	2	0	0	0	2	100
XC7S15	12,800	2,000	150	20	20	10	360	2	0	0	0	2	100
XC7S25	23,360	3,650	313	80	90	45	1,620	3	0	0	1	3	150
XC7S50	52,160	8,150	600	120	150	75	2,700	5	0	0	1	5	250
XC7S75	76,800	12,000	832	140	180	90	3,240	8	0	0	1	8	400
XC7S100	102,400	16,000	1,100	160	240	120	4,320	8	0	0	1	8	400



Síntesis y PnR



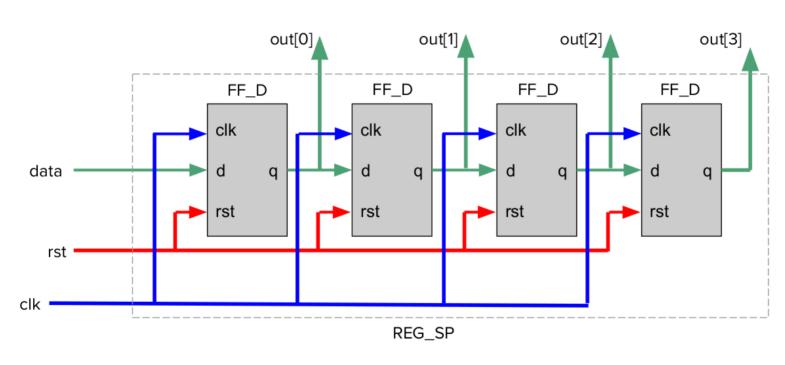
Sintetizador

- Su rol es transformar nuestro diseño RTL en Verilog a un diseño basado en las **primitivas de nuestro dispositivo**
- La transformación de RTL a primitivas se hace por dos caminos:
 - Inferencia: el sintetizador reconoce patrones en el código y los transforma a las primitivas asociadas
 - Instanciación: el diseñador puede instanciar a mano las primitivas del dispositivo para controlar el proceso de síntesis (resultando un código menos portable)



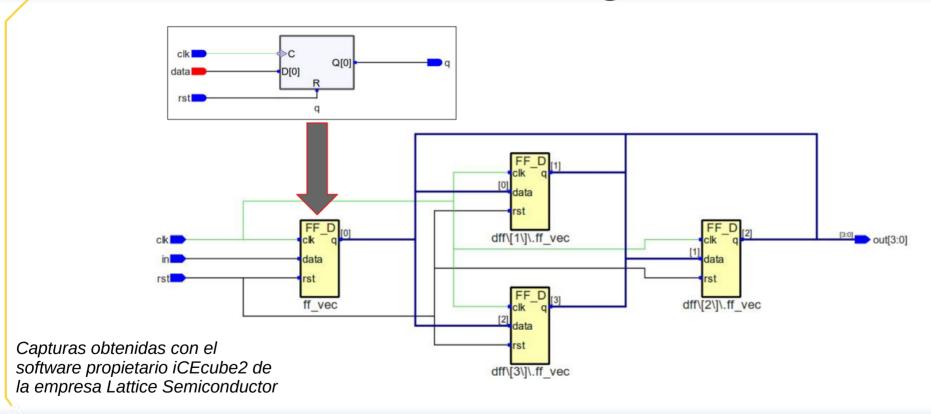


Paso 0: RTL

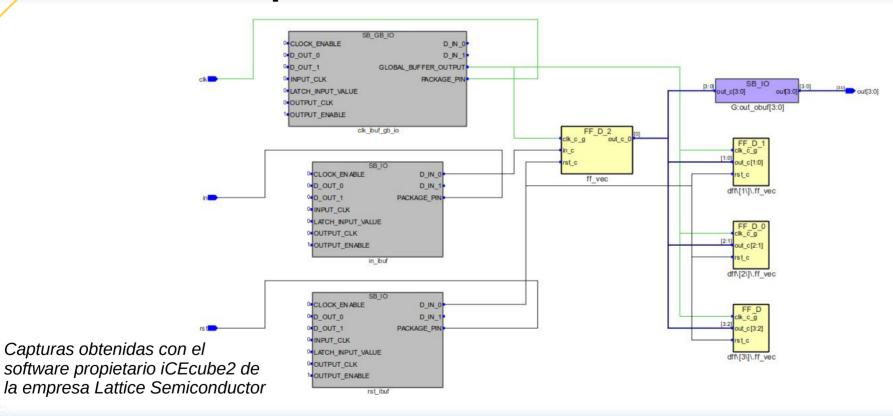


Todo parte de la descripción de nuestro circuito en un HDL

Paso 1: detección lógica



Paso 2: primitivas



Código post-síntesis

```
Block
defparam MEMORY_MEMORY_0_0_physical.WRITE_MODE=0;
                                                                RAM!
defparam MEMORY_MEMORY_0_0_physical.READ_MODE=0;
SB_RAM40_4K MEMORY_MEMORY_0_0_physical (
       .RDATA({dangling_wire_0,dangling_wire_1,dangling_wire_2,dangling_wire_
       .RADDR({dangling_wire_8,dangling_wire_9,dangling_wire_10,N__959,N__941
       .WADDR({dangling_wire_11,dangling_wire_12,dangling_wire_13,N__1127,N__
       .MASK({dangling_wire_14,dangling_wire_15,dangling_wire_16,dangling_wir
       .WDATA({dangling_wire_30,dangling_wire_31,dangling_wire_32,dangling_wi
       .RCLKE(),
       .RCLK(N__968),
       .RE(N_{-}1174),
                          La simulación de este código se denomina
       .WCLKE(N__1157),
                          simulación post-síntesis y
                                                                permite
       .WCLK(N__1136),
                          modelar efectos físicos de las primitivas
       .WE(N_{1175});
                          (consumos, propagación, etc).
```

```
PRE_IO r_clk_ibuf_gb_io_preio (
        .PADOEN(N 1503),
        .PADOUT(N__1502),
        .PADIN(N__1501),
        .CLOCKENABLE(),
        .DINO(),
        .DIN1(),
        .DOUTO(),
        .DOUT1(),
        .INPUTCLK(),
        .LATCHINPUTVALUE(),
        .OUTPUTCLK(),
        .OUTPUTENABLE());
```

Clock buffer (primitiva IO)

Tintroducción al Diseño Digital con EDU-CIAA-FPGA



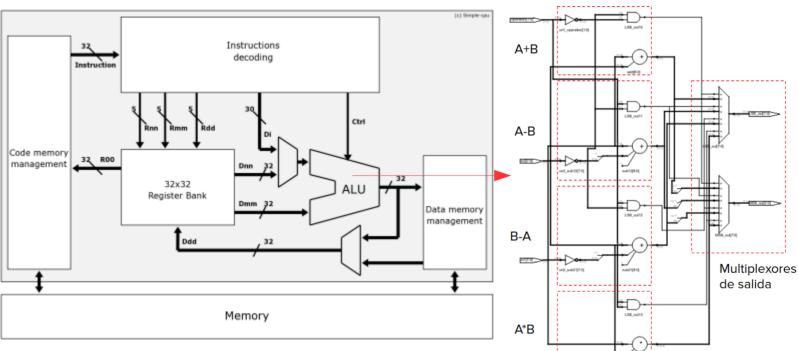
Place and Route

- Placement: selección de cuáles recursos de la FPGA se utilizarán para implementar las primitivas resultantes del proceso de síntesis.
- Routing: configuración de las interconexiones entre los bloques definidos por la etapa de placement.
- La evaluación del routing obtenido puede llevar a cambios en el placement, lo que a su vez deriva en un nuevo routing (iteraciones).
- El resultado final obedece a *constraints* o especificaciones de timing, potencia, ubicación E/S, etc.





Ejemplo: ALU



Fuente: https://www.simple-cpu.com/

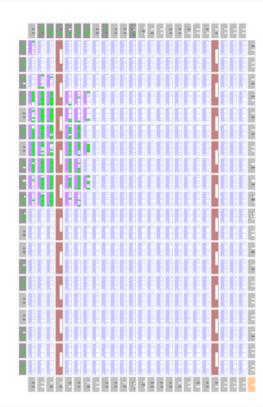
Realiza
operaciones
aritméticas o
lógicas entre
dos operandos
según el estado
de las líneas de
control
asociadas

Vamos a implementarlo en dos FPGA diferentes...





Floorplanning



Logic Resource Utilization:

iCE40 HX 4K

Total Logic Cells: 211/3520

Combinational Logic Cells: 211 out of 3520 5.99432%

Sequential Logic Cells: 0 out of 3520 0%

Logic Tiles: 42 out of 440 9.54545%

Registers:

Logic Registers: 0 out of 3520 0%

IO Registers: 0 out of 880 0
Block RAMs: 0 out of 20 0%
Warm Boots: 0 out of 1 0%

Pins:

Input Pins: 18 out of 107 16.8224% Output Pins: 16 out of 107 14.9533%

UTN **X** HAEDO

InOut Pins: 0 out of 107 0%
Global Buffers: 0 out of 8 0%

PLLs: 0 out of 2 0%

El placement está definido por la cercanía a los pines de E/S

> Capturas obtenidas con el software propietario iCEcube2 de la empresa Lattice Semiconductor

SYNTHESIS TARGET : Lattice iCE40 FPGA



Floorplanning

MAC DSP BLOCK

Final Design Statistics		
Number of LUTs	:	7
Number of DFFs	:	0
Number of DFFs packed to IO	:	0
Number of Carrys	:	2
Number of RAMs	:	C
Number of ROMs	:	0
Number of IOs	:	3
Number of GBIOs	:	0
Number of GBs	:	0
Number of WarmBoot	:	0
Number of PLLs	:	C
Number of I2Cs	:	C
Number of SPIs	:	C
Number of DSPs	:	1
Number of SBIOODs	:	C
Number of LEDDAIPs	:	C
Number of RGBADRVs	:	C
Number of LFOSCs	:	0
Number of HFOSCs	:	0
Number of FILTER_50NSs	:	0
Number of SPRAMs	:	C

iCE40 UltraPlus

Al usar el bloque DSP, el consumo de celdas lógicas cayó de 211 a 74!
Los demás bloques se colocan por cercanía al DSP pero también a las E/S

Capturas obtenidas con el software propietario iCEcube2 de la empresa Lattice Semiconductor



Timing closure





UTN 🕌 HAEDO

Introducción al Diseño Digital con EDU-CIAA-FPGA

Consideraciones prácticas

La principal limitación que presentan los circuitos secuenciales en la práctica es la **frecuencia máxima de operación**. La misma se ve afectada por diversos motivos:

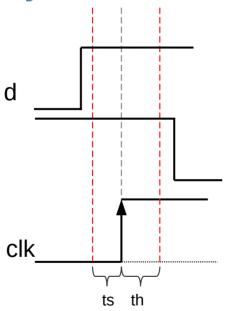
- Tiempos de **setup** y **hold** de los Flip-Flops.
- Delays propios de las celdas lógicas.
- Delays propios del ruteo de señales (tanto datos como clock).
- Jitter (clock)
- Skew (clock)

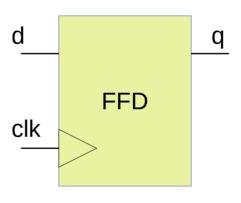
Es importante siempre tener una estimación de la máxima frecuencia del circuito que se está implementando. Esto permitirá determinar si el circuito funcionará correctamente o si deberán emplearse técnicas para mejorar su desempeño temporal.





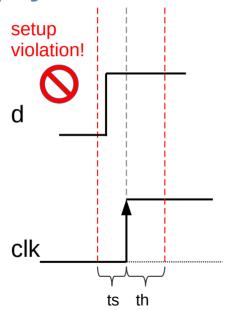
Consideraciones prácticas – setup y hold

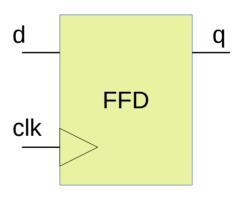




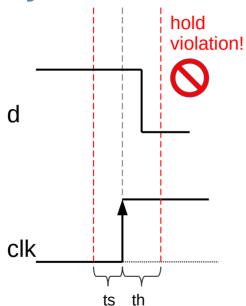
UTN 🕌 HAEDO

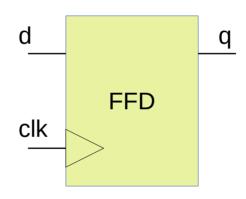
Consideraciones prácticas – setup y hold





Consideraciones prácticas – setup y hold

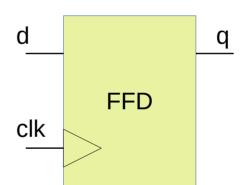




Consideraciones prácticas – setup slack

Setup slack = (required_time) - (arrival_time)

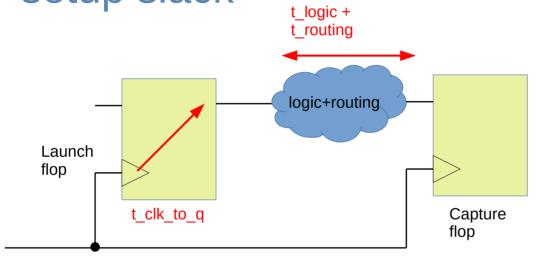
required time =
$$Tclk - ts$$

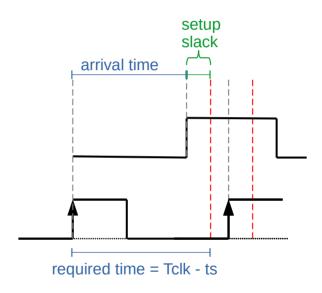


El setup slack debe ser siempre mayor o igual a cero!



Consideraciones prácticas – setup slack





 $setup_slack = (Tclk - ts) - (t_clk_to_q + t_logic + t_routing)$

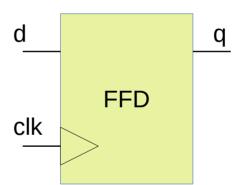


Consideraciones prácticas – hold slack

Hold slack = (arrival_time) - (required_time)

required_time = th

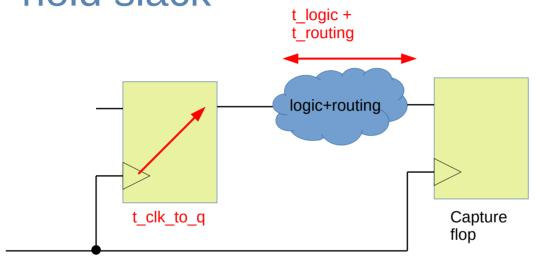
arrival_time = t_clk_to_q + t_logic + t_routing



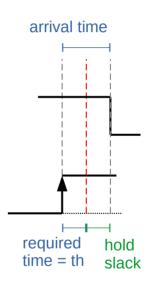
El hold slack debe ser siempre mayor o igual a cero!



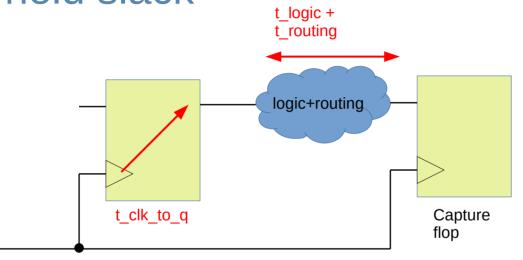
Consideraciones prácticas – hold slack



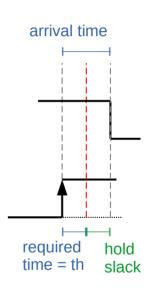
hold_slack = (t_clk_to_q + t_logic + t_routing) - th



Consideraciones prácticas – hold slack



Hold slack no depende del período de clock!









Consideraciones prácticas

Arbiter – frecuencia máxima reportada post PnR:

```
Info: Max frequency for clock 'clk$SB_IO_IN_$glb_clk': 646.41 MHz (PASS at 12.00 MHz)
```

Arbiter Game – frecuencia máxima reportada post PnR:

```
Info: Max frequency for clock 'clk$SB_IO_IN_$glb_clk': 140.88 MHz (PASS at 12.00 MHz)
```



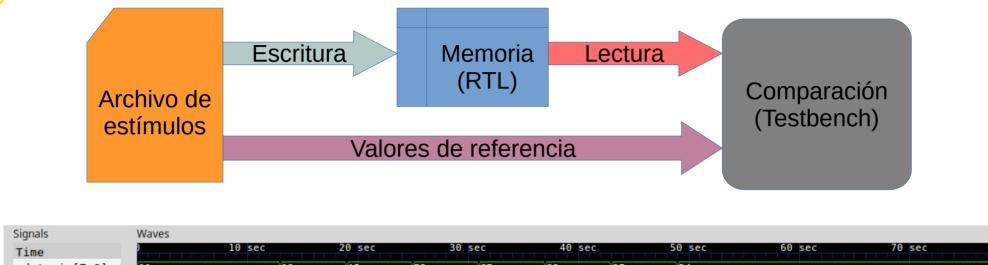
Ejercicios

- 4_1_Memorias
 - Inferencia de un bloque de memoria RAM
 - Simulación con estímulos configurables y análisis de la síntesis
- 4_2_SPWM
 - Instanciación de una primitiva PLL
 - Precarga de memoria RAM con valores iniciales
- Ambos ejemplos son simulables y sintetizables





Ejercicio 4.1: Memoria RAM



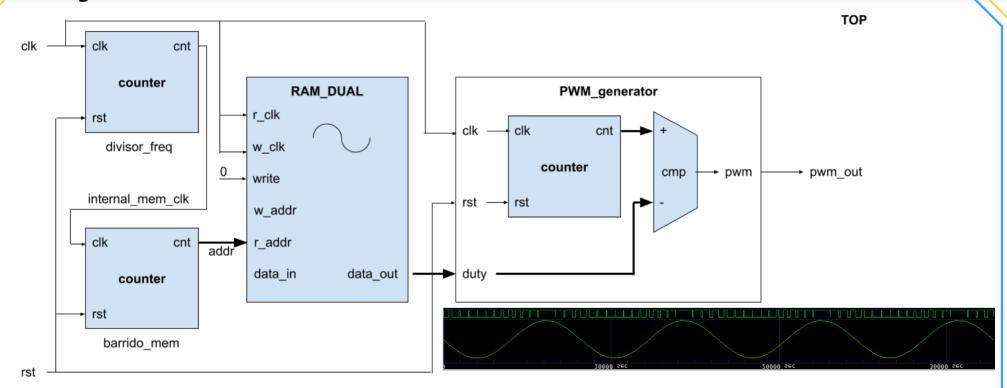




Tintroducción al Diseño Digital con EDU-CIAA-FPGA



Ejercicio 4.2: Modulador SPWM



Hojas de datos consultadas

- ICE40™ LP/HX Family Data Sheet
 - DS1040 Version 3.4, October 2017
- ICE40 Ultra Plus™ Family Data Sheet
 - FPGA-DS-02008 Version 1.4, August 2017
- Intel ® Cyclone ® 10 LP Device Overview
 - C10LP51001 ID 683879 Version 2022.05.27
- Xilinx 7 Series FPGAs Data Sheet: Overview
 - DS180 (v2.6.1) September 8, 2020