

HOJA DE RESPUESTAS - PRÁCTICA 3

Nombres y apellidos y de los alumnos:

Francisco Alcudia Diaz

Tomas Higuera Viso

Tabla de resultados (a rellenar durante la ejecución de los diversos ejercicios):

	Ej. 1	Ej. 3a	Ej. 3b	Ej. 3c	Ej. 4a	Ej. 4b
Latencia (retardo entrada-salida), en ciclos [1]	2	2	2	2	3	5
Total Number of LUTs	98	98	141	146	126	130
Slices (Slice-L + Slice-M)	74	76	80	64	76	75
Number of Flip-Flops (total)	146	146	146	146	183	272
Number of Flip-Flops (slice FF)				129		
Number of Flip-Flops (IOB FF)				17		
Number of I/O pins	147	147	147	147	147	147
Period Requirement (ns)	20	12	8	8	8	8
Worst Negative Slack (WNS) (ns)	8.201	0.429	-2.957	-2.856	1.862	4.109
Data Path Delay (ns) [2]	11.808	5.734	5.125	9.908	6.068	3.908
Levels of logic in data path [2]	18	1	1	15	13	7
Time used for logic in data path (%) [2]	43.054	56.211	62.651	45.659	59.266	50.254
Time used for routing in data path (%) [2]	56.946	43.789	37.349	54.341	40.734	49.746
Clock Path Skew (ns) [2]	0.014	-5.302	-5.297	-0.079	-0.064	-0.029

[1] No dar el tiempo en ns, sino el número de flancos activos de reloj que tienen que ocurrir entre el momento en que se presenta una entrada y el momento en que está disponible la salida correspondiente.

[2] En la ruta crítica.

Ejercicio 1:

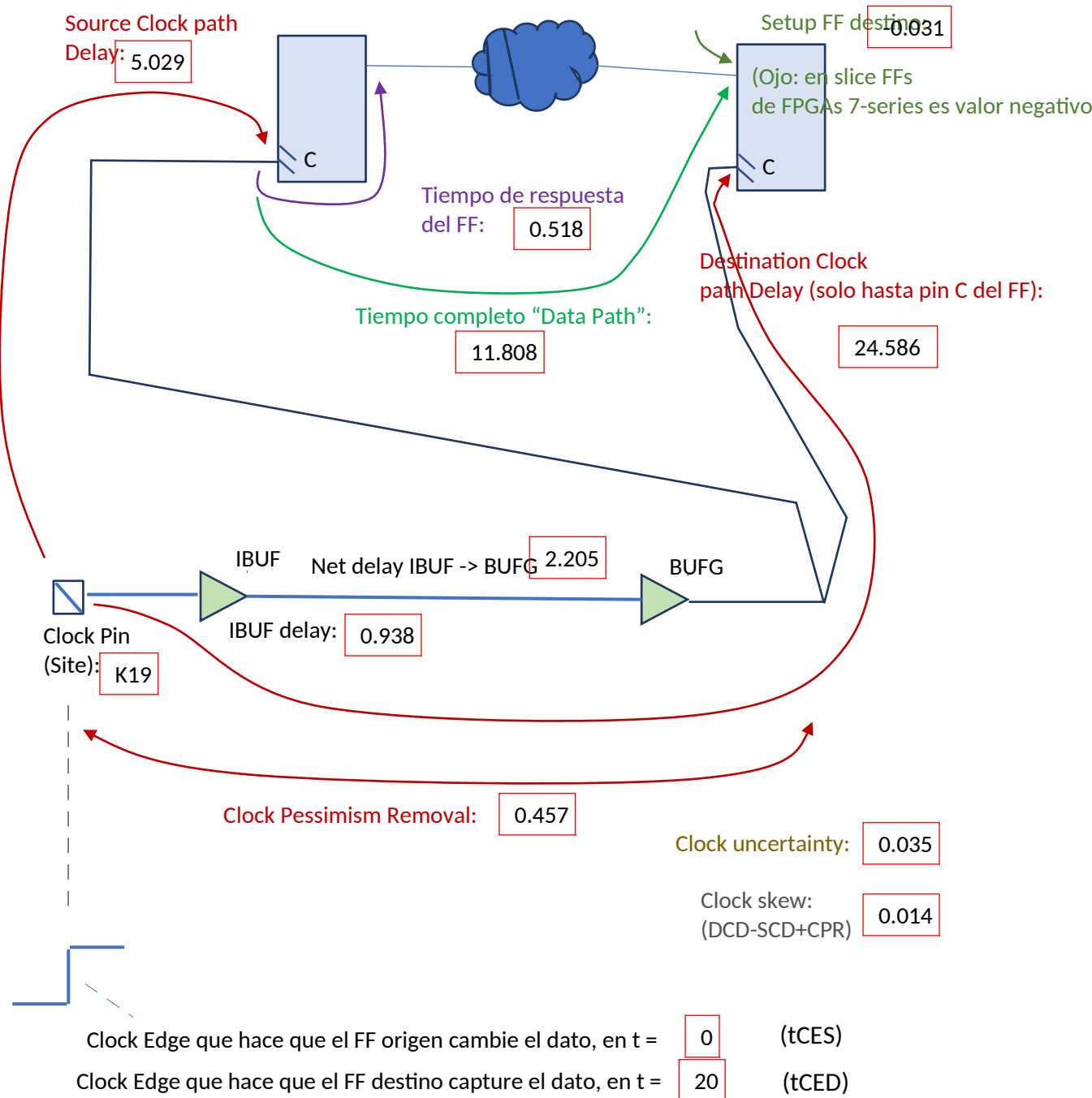
Restricciones temporales iniciales - Rellenar los siguientes dos valores:

Periodo objetivo inicial (ns): 20

Frecuencia objetivo inicial (MHz): 50

Esquema de tiempos en el peor path entre flip-flops - Rellenar los recuadros de este esquema:

(todos los tiempos a rellenar en nanosegundos)



Arrival Time = tCES + Source Clock Path delay + Data Path delay:

16.837

Required Time = tCED + Destination Clock Path delay + Clock Pessimism Removal
- Clock Uncertainty - Setup FF destino:

25.038

Slack = Required Time - Arrival Time: 8.201

Ejercicio 2:

Entradas:

- ¿Se cumple la constraint impuesta? [Sí](#)
- ¿Cuál es el *slack* para esta constraint? [13.731 ns](#)

Salidas:

- ¿Se cumple la constraint impuesta? [Sí](#)
- ¿Cuál es el *slack* para esta constraint? [7.272 ns](#)

Ejercicio 3:

Iteración 1: cambio de la restricción de periodo de reloj a 12 ns:

- ¿Se logran cumplir todos los requerimientos impuestos? [Sí](#)
- En caso contrario, decir qué constraint tiene el peor slack:

Iteración 2: cambio de la restricción de periodo de reloj a 8 ns:

- ¿Se logran cumplir todos los requerimientos impuestos? [No](#)
- En caso contrario, decir qué constraint tiene el peor slack: [La de salidas](#)
- Comparar los resultados con los de la Iteración 1:

[En comparación con la iteración 1, destaca mucho la diferencia de LUTs utilizadas, siendo 98 en la iteración 1 y 141 en la iteración 2.](#)

Iteración 3: Utilización de la constraint *set_property IOB TRUE [all_outputs]*:

- ¿Se logran cumplir todos los requerimientos impuestos? [No](#)
- En caso contrario, decir qué constraint tiene el peor slack: [La de salidas](#)
- Comparar los resultados con los de la Iteración 2:

[Respecto al tamaño, se reduce el número de slices utilizadas puesto que dejamos de usar FFs de slices y pasamos a utilizar los FFs de los bloques OLOGIC. Respecto a las diferencias del camino crítico, puesto que utilizamos estos otros FFs, estamos "alargando" el data path, convirtiendo éste en el camino crítico puesto que ahora los datos recorren más pista.](#)

Ejercicio 4:

a) Pipeline con una etapa de registros

¿Qué impacto tiene la técnica en área?

Aumenta considerablemente el número de FFs.

¿Y en timing?

Los requisitos de timing se cumplen. Aplicando la técnica podemos reducir el periodo del ciclo de reloj y calcular más checksums por segundo que sin aplicar la técnica. Hemos comprobado que con 6 ns de ciclo de reloj los requisitos se cumplen y con 5 ns no.

Calcularemos la mejora con respecto al diseño sin pipeline que funcionaba a 12 ns.

$12 \text{ ns} / 6 \text{ ns} = 2$, esto significa que nuestro sistema es el doble de bueno, o que rinde el doble que el sistema sin pipeline a 12 ns.

De tal forma que conseguimos una aceleración del 100 %.

b) Pipeline con dos o tres etapas de registros ("OPCIONAL").

Número de líneas adicionales de registros añadidas: dos sin contar la línea de registros del apartado anterior. Es decir, creando un pipeline con tres etapas.

Analizar área y velocidad y comentar el resultado:

El área aumenta mucho, concretamente el número de FFs. Al igual que en el apartado A podemos reducir más el periodo del ciclo de reloj. Hemos comprobado que con 4 ns de ciclo de reloj los requisitos se cumplen, y con 3 ns no.

Repetimos el cálculo de mejora:

$12 \text{ ns} / 4 \text{ ns} = 3$

Logramos una aceleración del 200 %.