



Universidad Nacional  
de San Juan

## Trabajo Final de Carrera Ingeniería Electrónica



Facultad de Ingeniería  
Universidad Nacional de San Juan

Desarrollo e implementación de un puerto USB 2.0 de alta velocidad para FPGA a través de un puente específico

**Edwin Barragán**

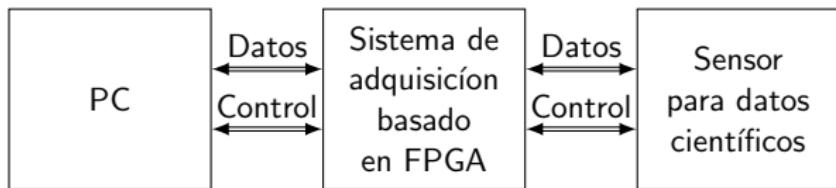
Autór

**Mag. Ing. Cristian Sisterna**

**Mgtr. Ing. Martín Perez**

**Dr. Ing Marcelo Segura**  
Asesores

# Comunicación USB 2.0 para sistema científicos implementados en FPGA



- Los avances tecnológicos permiten el desarrollo de nuevos sensores.
- Estos sensores producen una gran cantidad de datos.
- Las ventajas que presentan los FPGA los convierte en dispositivos interesantes para la adquisición de los datos.
- La PC posibilita el procesamiento y análisis de datos.
- Se hace necesaria una comunicación veloz y eficaz entre la PC y el FPGA.

# Selección del protocolo de comunicación

Ethernet



Wi-Fi



USB



- Gran ancho de banda (1 Gbps)
- Destinado a redes de PC
- Requiere infraestructura adicional

- Buen ancho de banda (600 Mbps)
- Redes inalámbricas
- Comunicación inestable

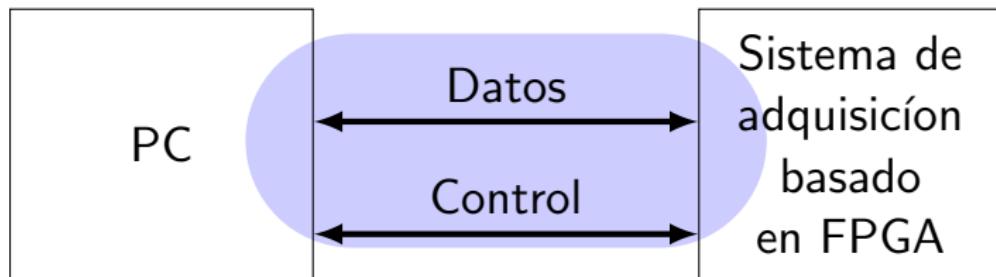
- Buen ancho de banda (480 Mbps)
- Destinado a periféricos

## Selección del protocolo de comunicación



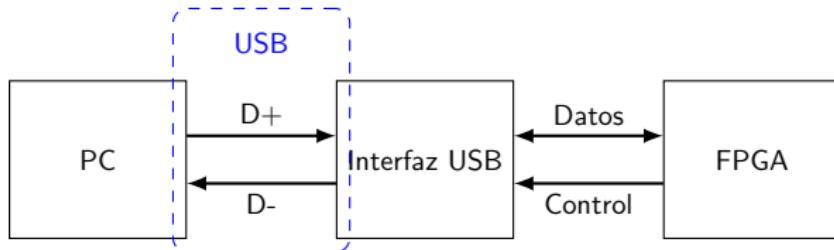
USB 2.0 resulta la mejor alternativa para comunicar periféricos a una PC sin comprometer funcionalidades importantes.

# Objetivos



- Objetivo General
  - ▶ Realizar una comunicación entre un FPGA y una PC mediante USB 2.0.

# Objetivos



- Objetivos Particulares

- ▶ Seleccionar el dispositivo para realizar la interfaz USB.
- ▶ Realizar la configuración de la interfaz USB.
- ▶ Sintetizar un circuito en VHDL que sea capaz de interactuar con la interfaz USB, minimizando la utilización de recursos programables.
- ▶ Desarrollar un sistema de pruebas.
- ▶ Validar el funcionamiento.

# Agenda

## 1 Introducción

- Bus Serial Universal

## 2 Implementación

- Arquitectura del sistema
- Selección de los componentes del sistema
- Configuración de la Interfaz USB
- Desarrollo de la Máquina de Estados Finitos

## 3 Evaluación y validación

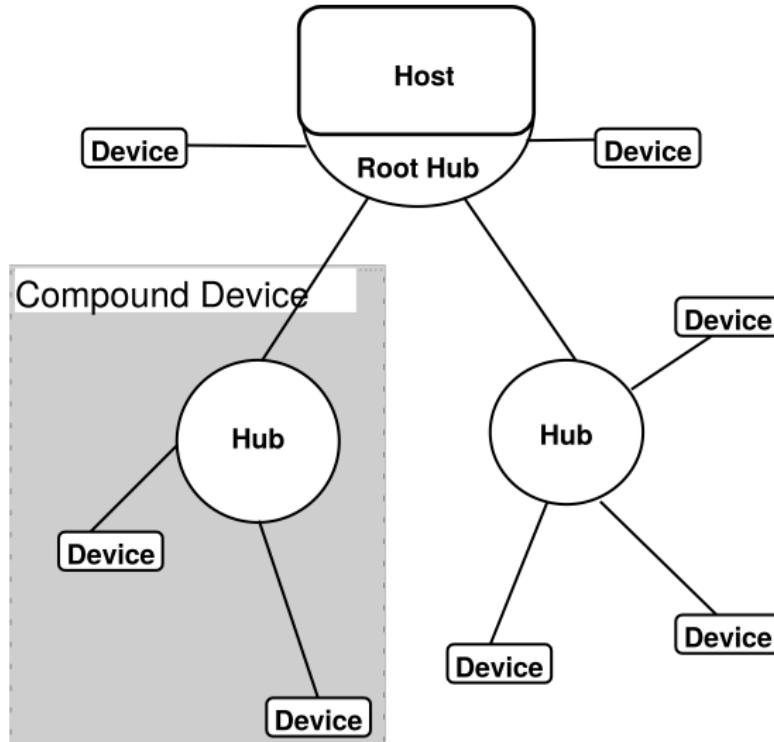
## 4 Resultados y Trabajos Futuros

# Agenda

## 1 Introducción

- Bus Serial Universal

# USB - Bus Serial Universal



Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC and Philips, *Universal Serial Bus Specification*, Revision 2.0.

## USB 2.0 - Velocidades de operación



Low-Speed: 1.5 Mbps



Full-Speed: 12 Mbps



High-Speed: 480 Mbps

---

Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC and Philips, *Universal Serial Bus Specification*, Revision 2.0.

## USB - Tipo de transferencias

- Control: Destinada a comunicaciones de control y estado del bus entre el Host y los Dispositivos.
- Interrupción: Posee latencia mínima. Destinada a comunicaciones no periódicas.
- Isócronas: Acceso periódico al bus. Sin retransmisión de mensajes. Destinada a la comunicación que caduca con demoras en la entrega, como imágenes.
- en Masa: Destinada a maximizar la ocupación del bus.

---

Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC and Philips, *Universal Serial Bus Specification*, Revision 2.0.

# USB - Tipo de transferencias

- Control: Destinada a comunicaciones de control y estado del bus entre el Host y los Dispositivos.
- Interrupción: Posee latencia mínima. Destinada a comunicaciones no periódicas.
- **Isócronas:** Acceso periódico al bus. Sin retransmisión de mensajes. Destinada a la comunicación que caduca con demoras en la entrega, como imágenes.
- en Masa: Destinada a maximizar la ocupación del bus.



---

Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC and Philips, *Universal Serial Bus Specification*, Revision 2.0.

## USB - Tipo de transferencias

- Control: Destinada a comunicaciones de control y estado del bus entre el Host y los Dispositivos.
- Interrupción: Posee latencia mínima. Destinada a comunicaciones no periódicas.
- Isócronas: Acceso periódico al bus. Sin retransmisión de mensajes. Destinada a la comunicación que caduca con demoras en la entrega, como imágenes.
- **en Masa:** Destinada a maximizar la ocupación del bus.



---

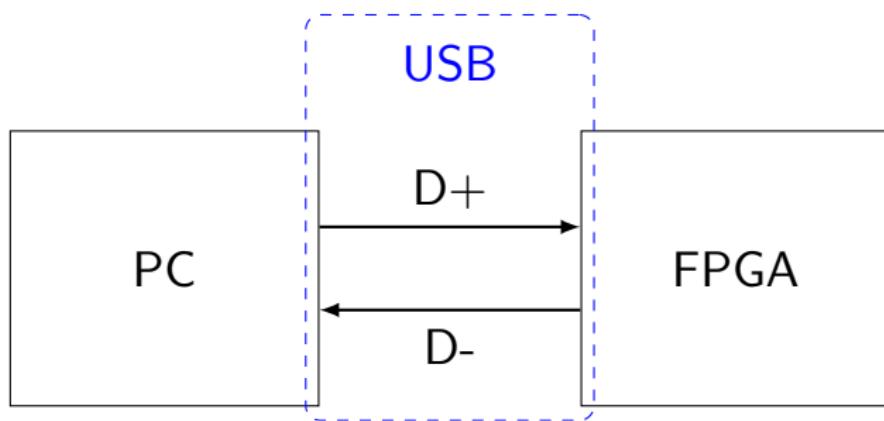
Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC and Philips, *Universal Serial Bus Specification*, Revision 2.0.

# Agenda

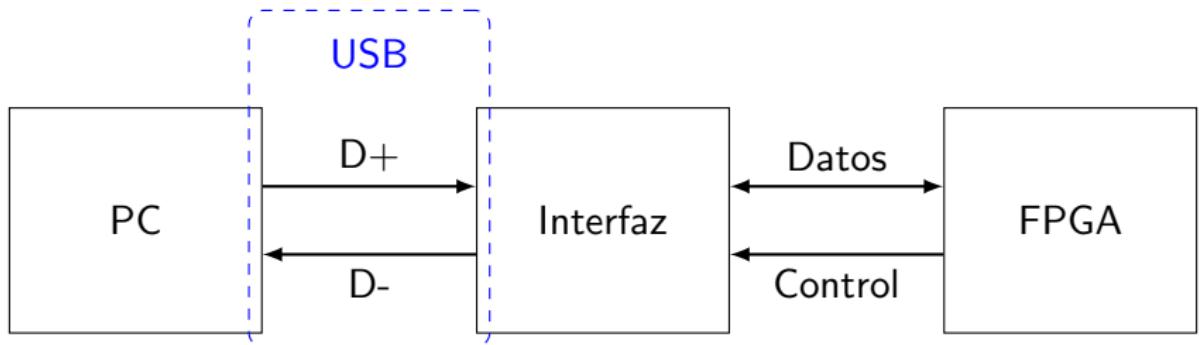
## 2 Implementación

- Arquitectura del sistema
- Selección de los componentes del sistema
- Configuración de la Interfaz USB
- Desarrollo de la Máquina de Estados Finitos

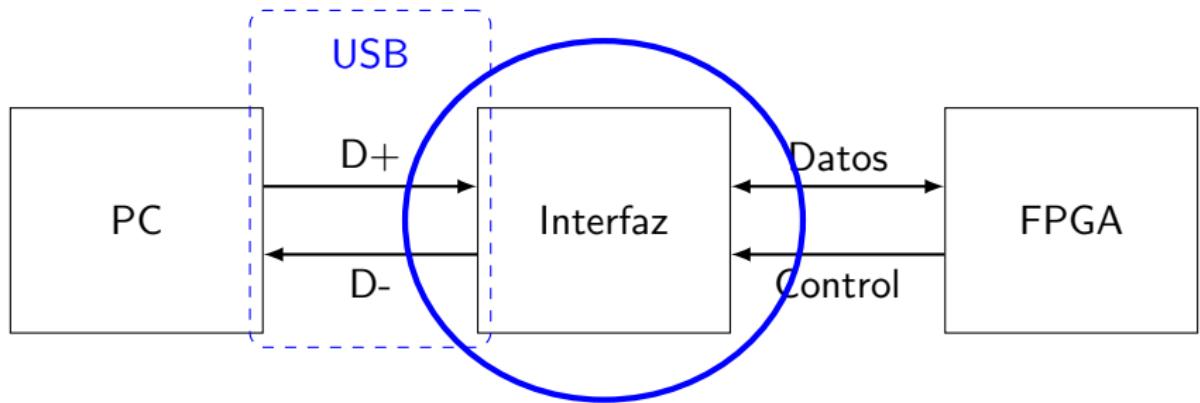
# Arquitectura del sistema realizado



# Arquitectura del sistema realizado

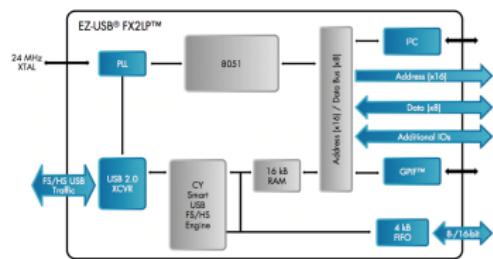


# Arquitectura del sistema realizado

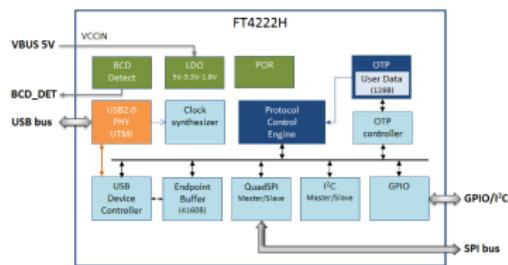


# Selección de la Interfaz USB

## Cypress FX2LP



## FTDI FT4222H



- Pros

- ▶ Comunicación paralela de 16 bits
- ▶ Microcontrolador 8051

- Contras

- ▶ Requiere esfuerzo en la configuración y software dedicado

- Pros

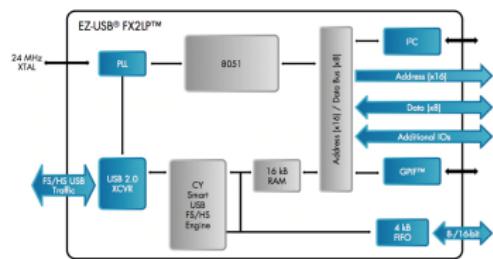
- ▶ No requiere programación
- ▶ Económico

- Contras

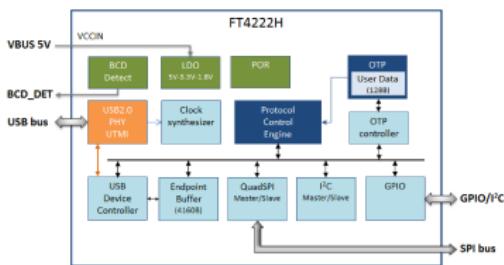
- ▶ Solo permite Transferencias en Masa
- ▶ Comunicación SPI cuádruple

# Selección de la Interfaz USB

## Cypress FX2LP



## FTDI FT4222H



### Pros

- ▶ Comunicación paralela de 16 bits
- ▶ Microcontrolador 8051

### Contras

- ▶ Requiere esfuerzo en la configuración y software dedicado

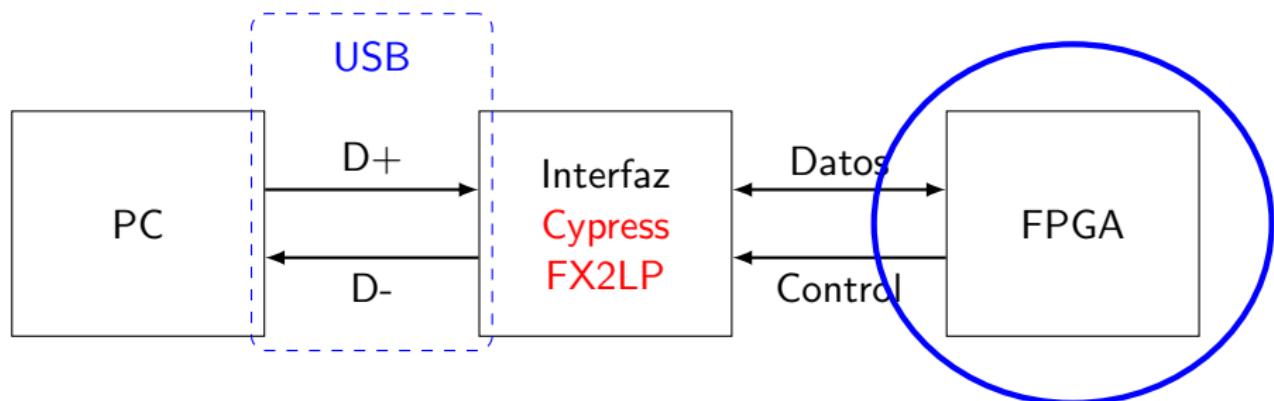
### Pros

- ▶ No requiere programación
- ▶ Económico

### Contras

- ▶ Solo permite Transferencias en Masa
- ▶ Comunicación SPI cuádruple

# Selección del FPGA



# Selección del FPGA



Spartan-3E Starter

- FPGA Spartan 3E de Xilinx
- Gran cantidad de periféricos



Nexys 3

- FPGA Spartan 6 de Xilinx
- Gran cantidad de periféricos



Mojo v3

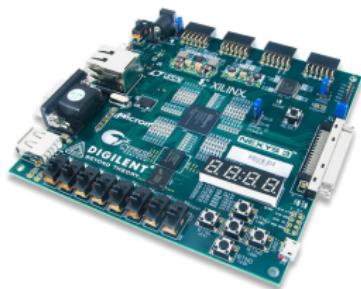
- FPGA Spartan 6 de Xilinx
- Muy económica

# Selección del FPGA



Spartan-3E Starter

- FPGA Spartan 3E de Xilinx
- Gran cantidad de periféricos



Nexys 3

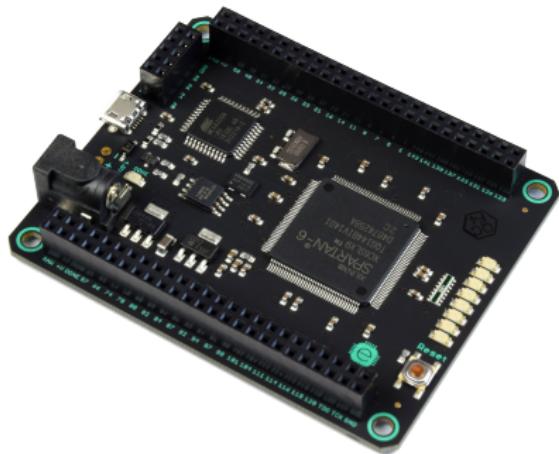
- FPGA Spartan 6 de Xilinx
- Gran cantidad de periféricos



Mojo v3

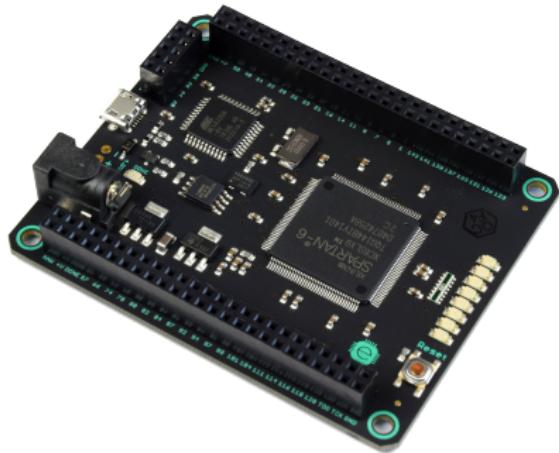
- FPGA Spartan 6 de Xilinx
- Muy económica

# La placa de desarrollo MOJO v3



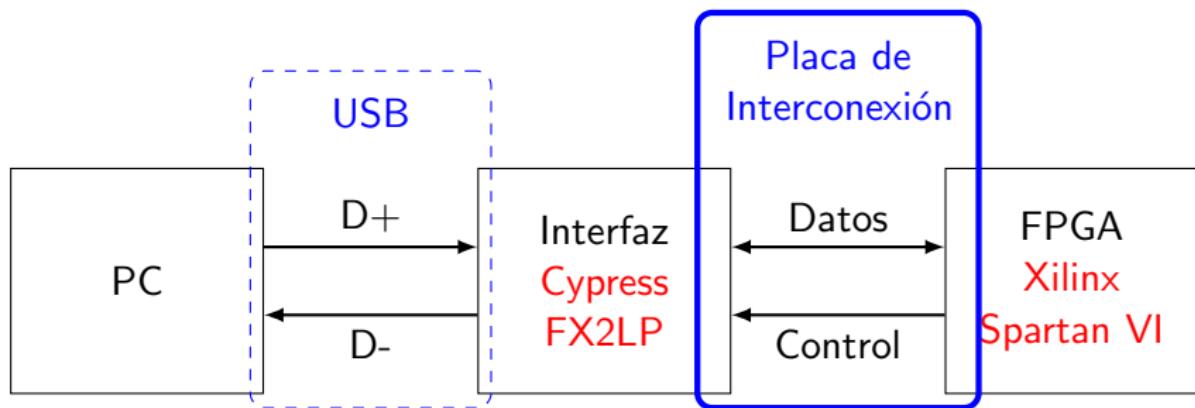
- FPGA Spartan 6 XC6SLX9 de Xilinx
- 84 pines IO digitales
- 8 entradas analógicas
- 8 LEDs de propósito general
- 1 pulsador de propósito general
- ATmega32U4 para configurar la FPGA y leer los pines analógicos.
- Memoria flash para almacenar la configuración de la FPGA.
- USB para programación y comunicación.

# La placa de desarrollo MOJO v3



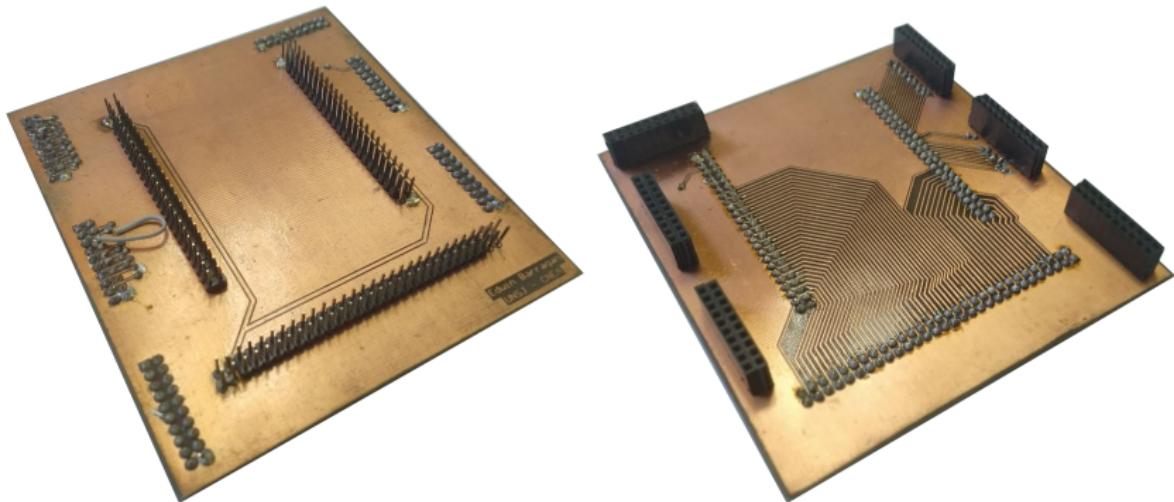
- USB para programación y comunicación.
  - ▶ USB 2.0 Full-Speed de 12 Mbps
  - ▶ Implementado a través de controlador ATmega32U4
  - ▶ Interfaz FPGA-ATmega32U4 via SPI de 8 Mbps.

# Circuito de interconexión



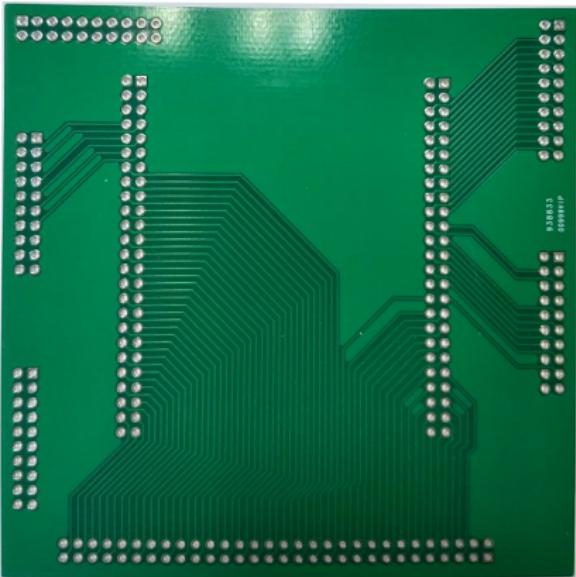
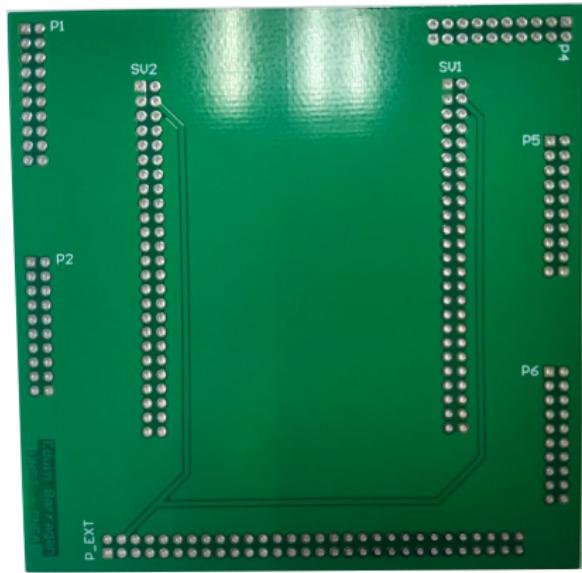
# Circuito de interconexión

- Versión 1:



# Circuito de interconexión

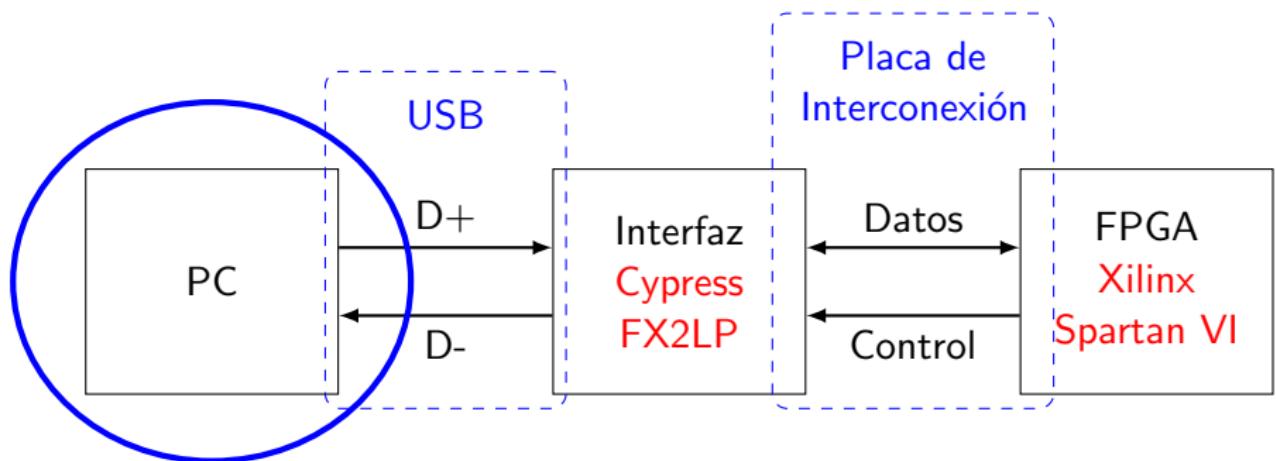
- Versión 2:



# Componentes conectados



# Acceso al dispositivo desde la PC

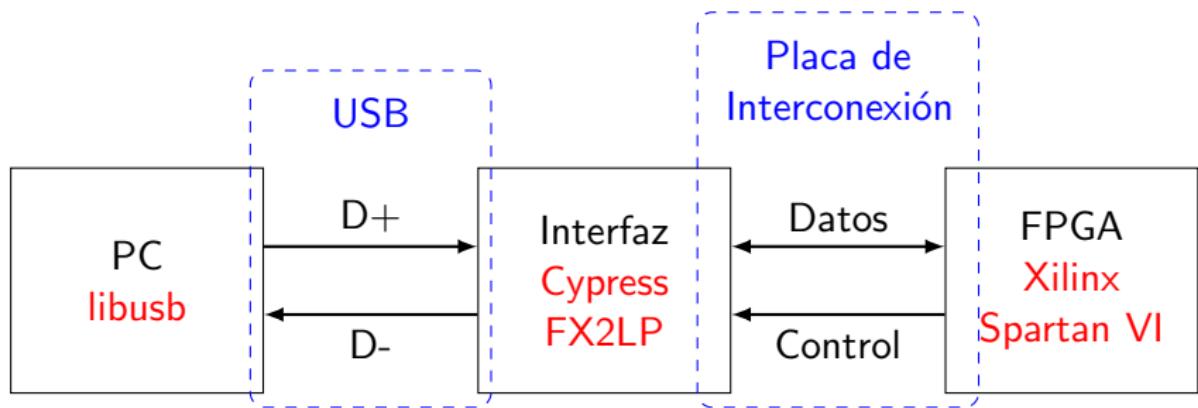


# Acceso al dispositivo desde la PC

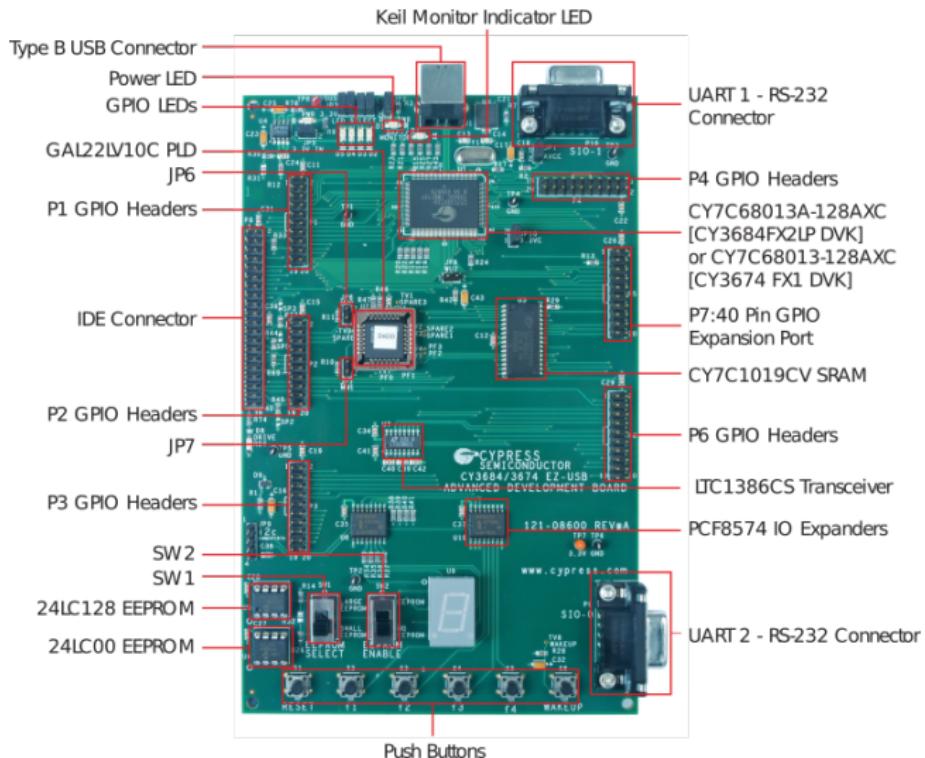
Para acceder al puerto USB desde la PC se utilizó la biblioteca libusb-1.0:

- Software libre: Puede ser utilizado sin necesidad de comprar una licencia.
- API portable: Permite escribir código que puede ser compilado en múltiples Sistemas Operativos.
- Abundante documentación: Se encuentran en Internet manuales y tutoriales muy completos sobre su funcionamiento.

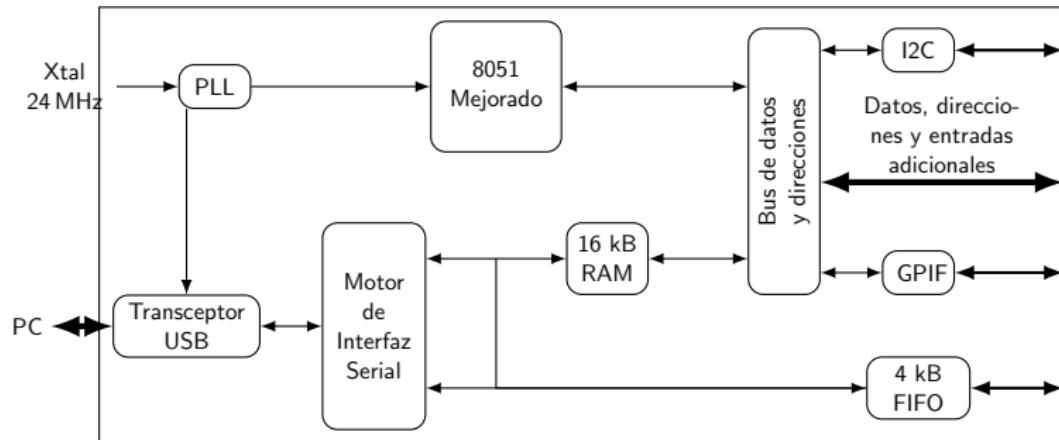
# Componentes del sistema



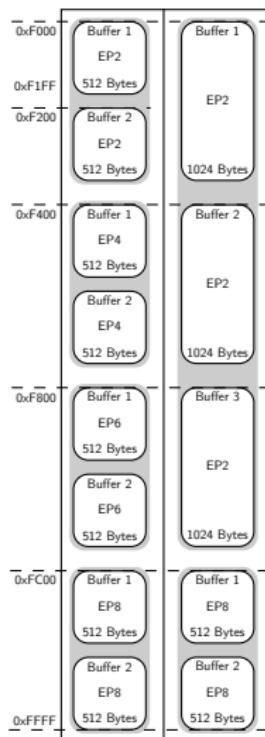
# Placa de desarrollo utilizada para la interfaz



# El circuito integrado FX2LP



# Configuración del dispositivo USB

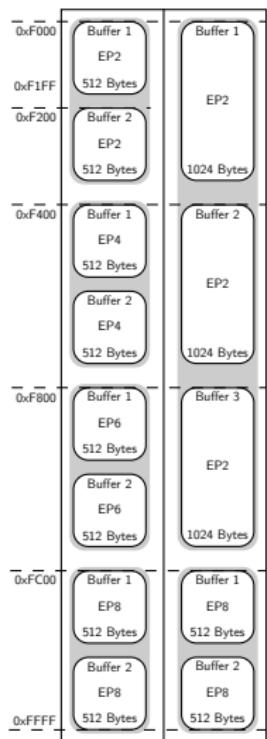


La configuración de las tuberías de comunicación se definieron con la finalidad de obtener el mayor ancho de banda posible a la entrada.

- Entrada:

- ▶ Extremo EP2
- ▶ Transferencias Isocrónicas
- ▶ 1024 Bytes máximo por transferencia
- ▶ 3 Buffers

# Configuración del dispositivo USB

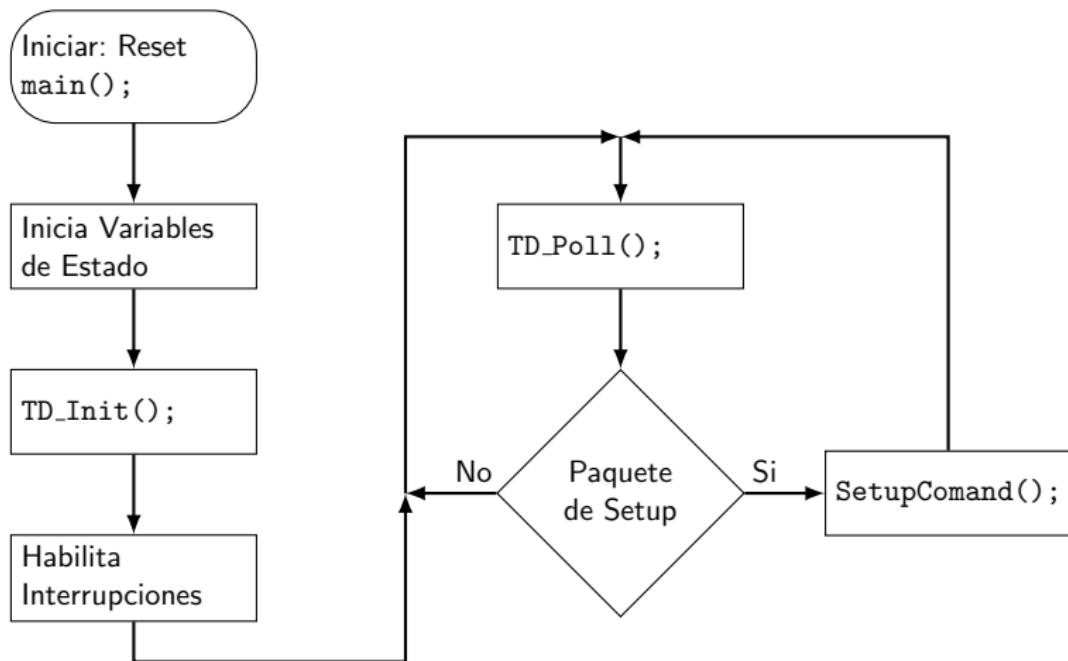


La configuración de las tuberías de comunicación se definieron con la finalidad de obtener el mayor ancho de banda posible a la entrada.

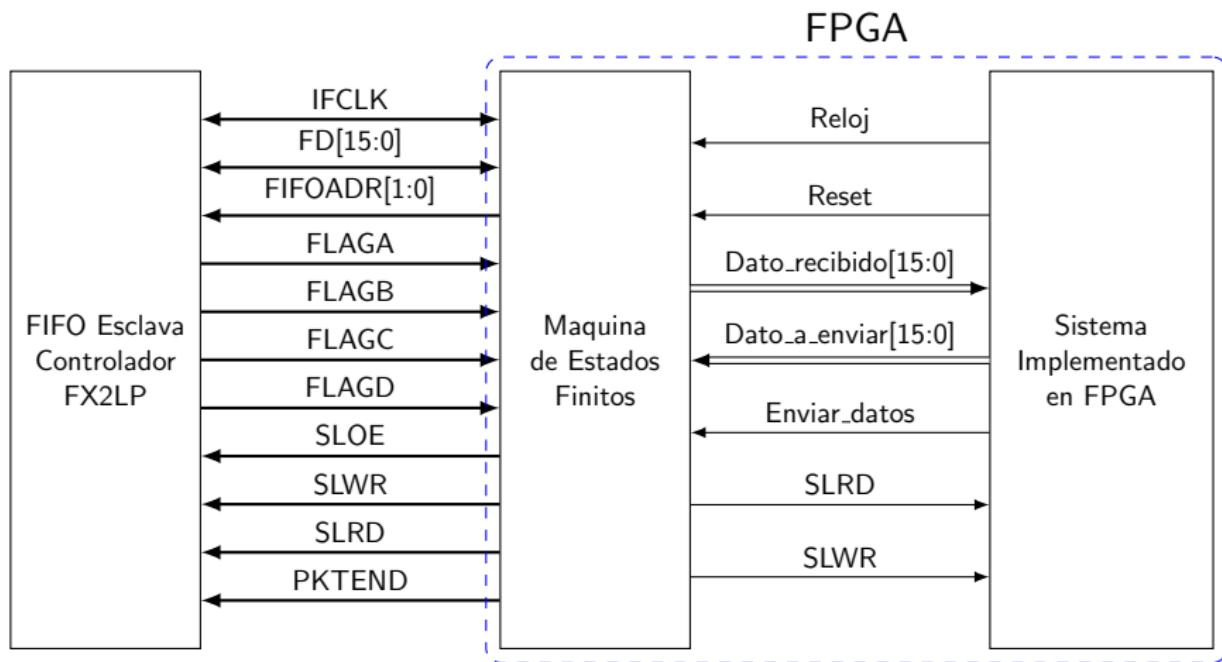
- Salida:

- ▶ Extremo EP8
- ▶ Transferencia por bultos
- ▶ 512 Bytes por transferencia
- ▶ 2 Buffers

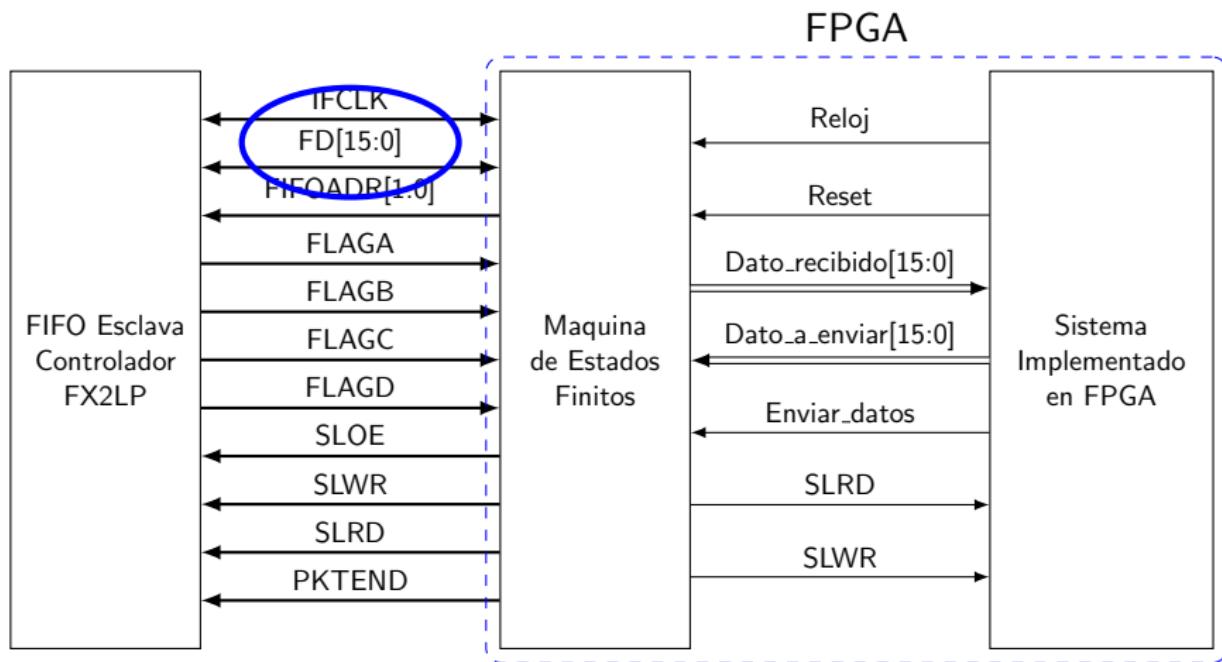
# Programación de la Interfaz USB



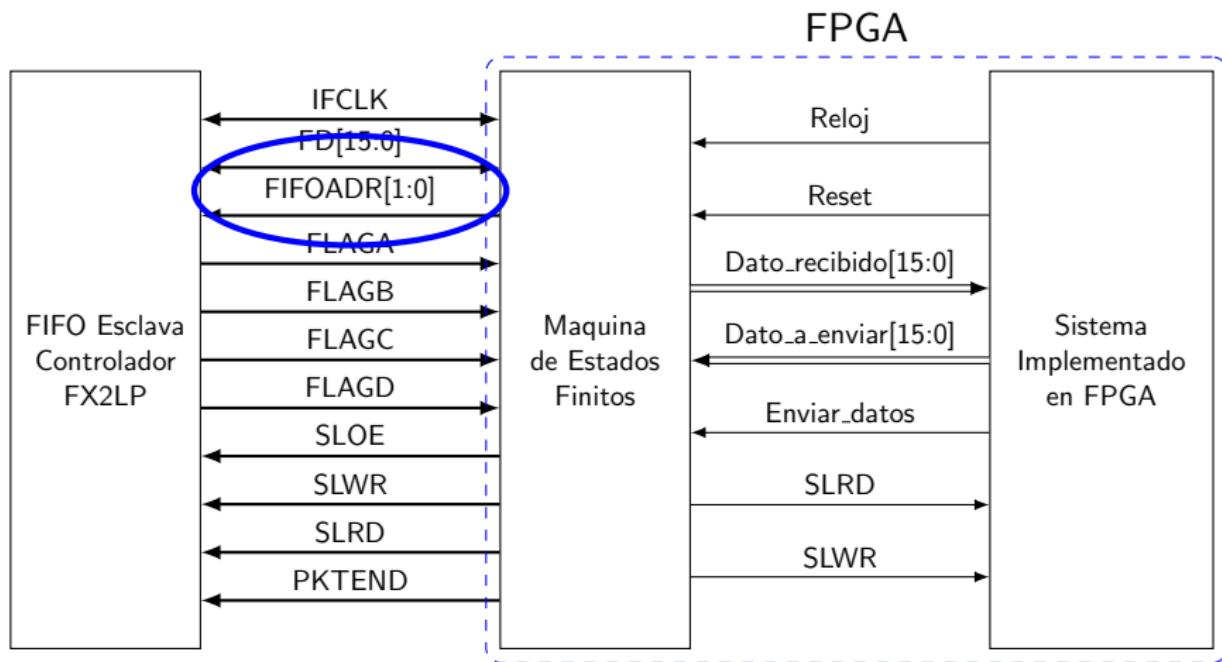
# Diagrama de señales entre la Interfaz USB y el FPGA



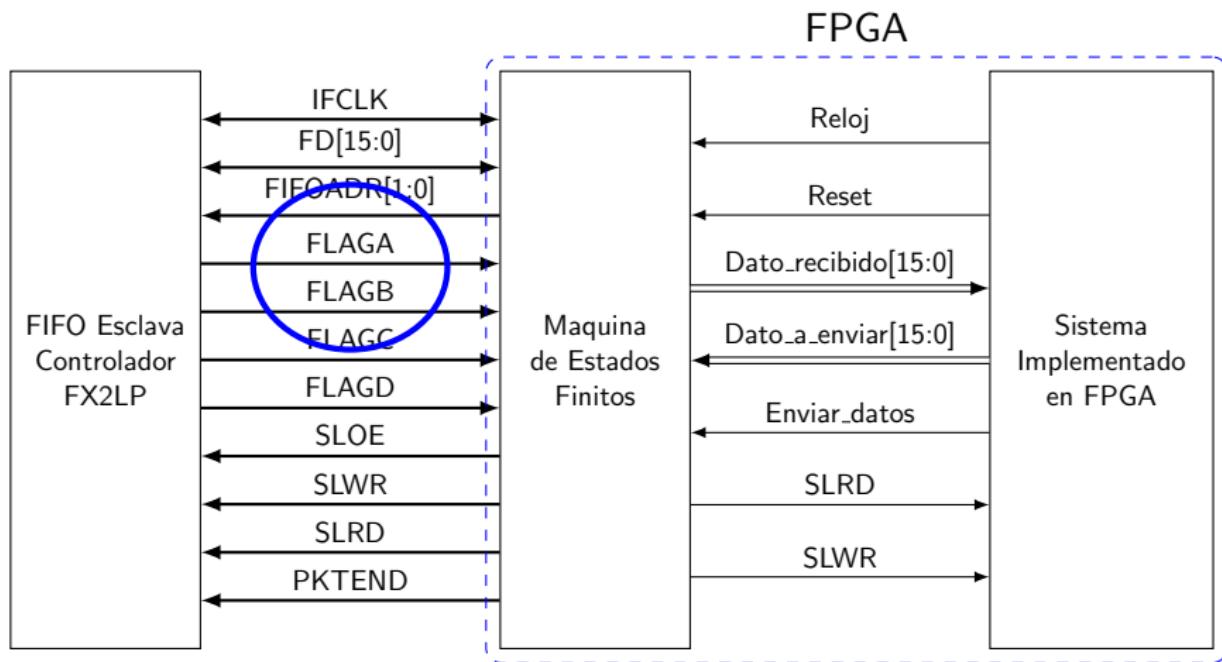
# Diagrama de señales entre la Interfaz USB y el FPGA



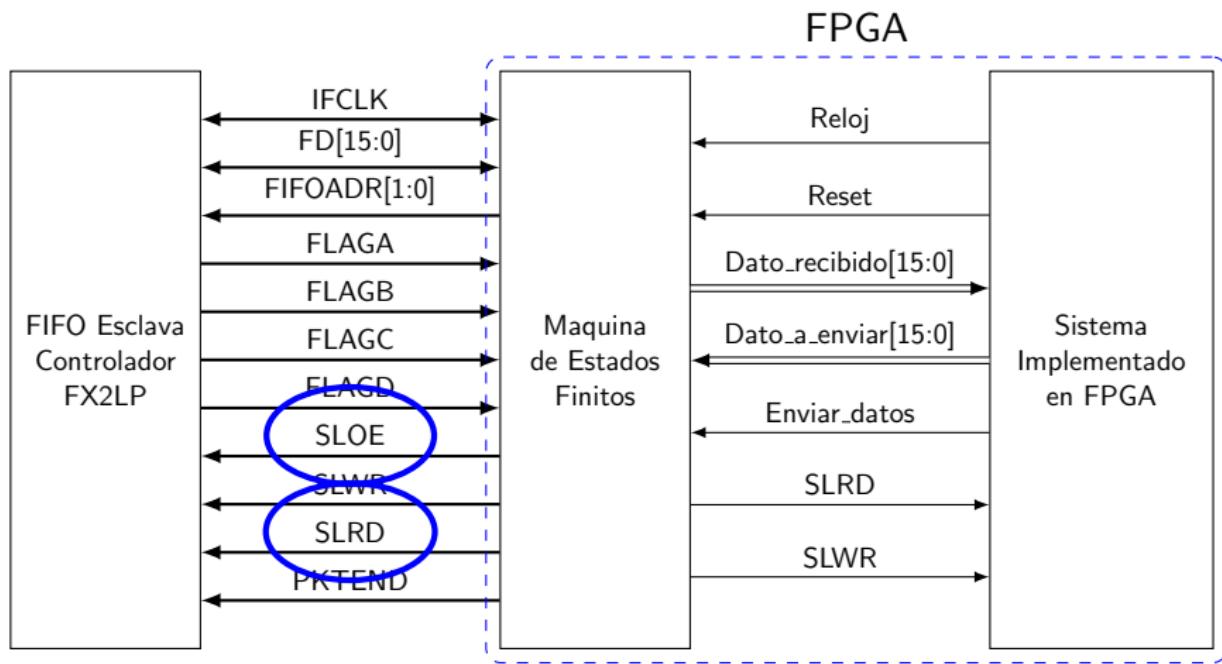
# Diagrama de señales entre la Interfaz USB y el FPGA



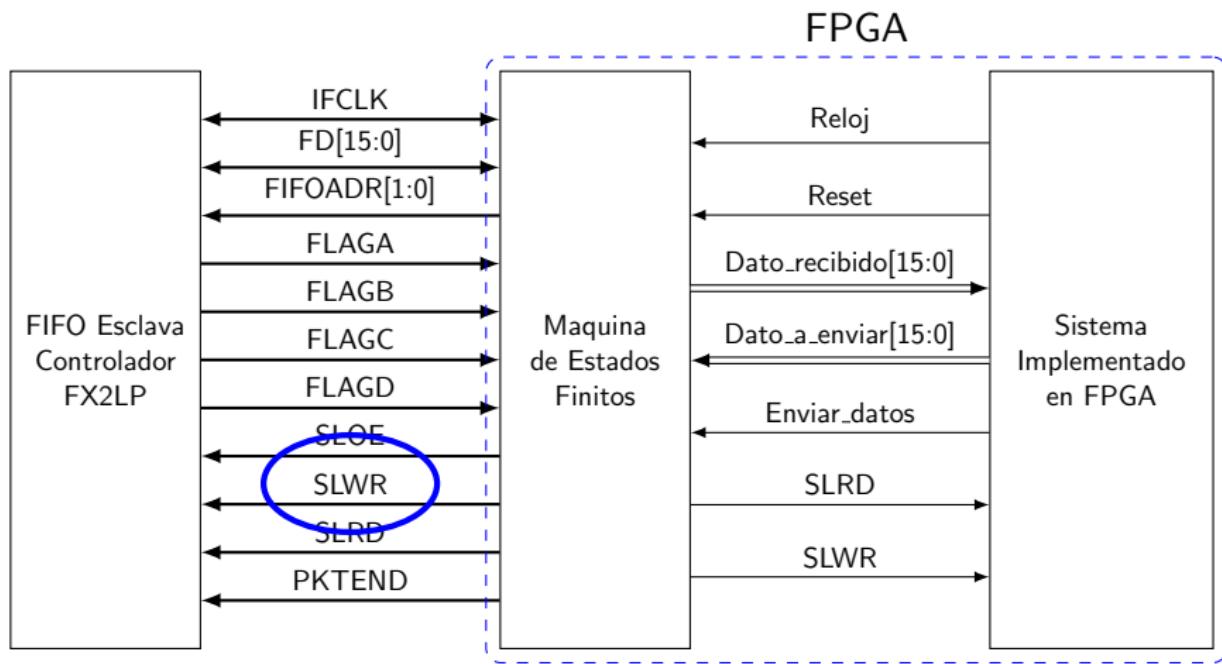
# Diagrama de señales entre la Interfaz USB y el FPGA



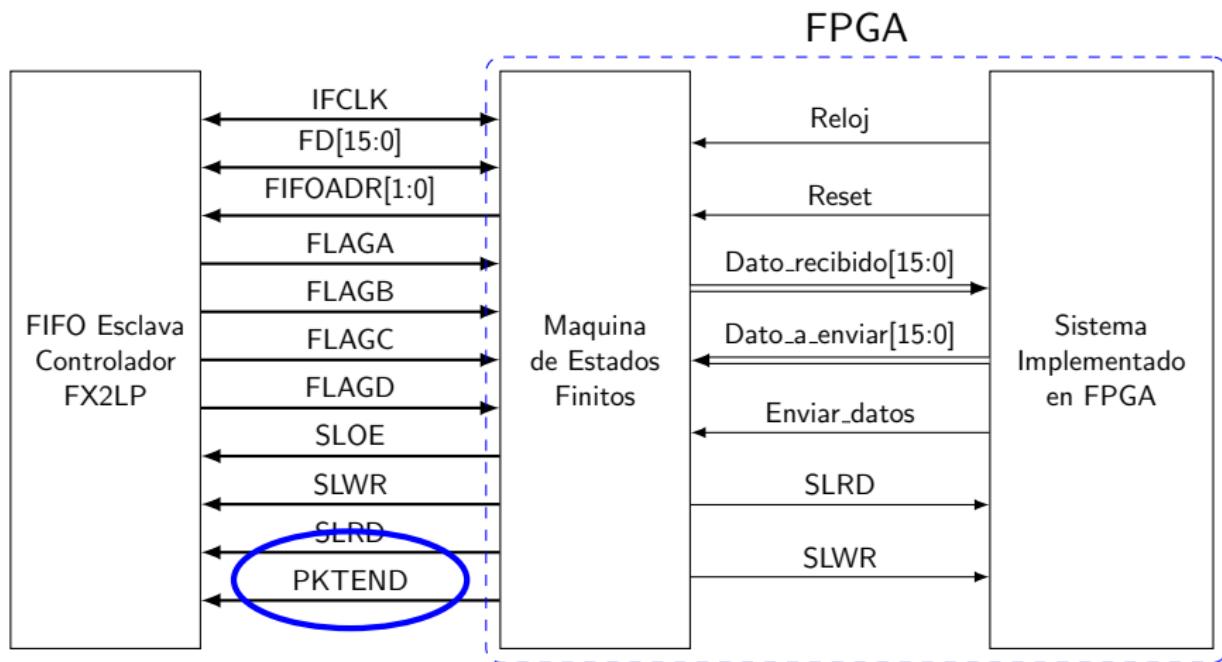
# Diagrama de señales entre la Interfaz USB y el FPGA



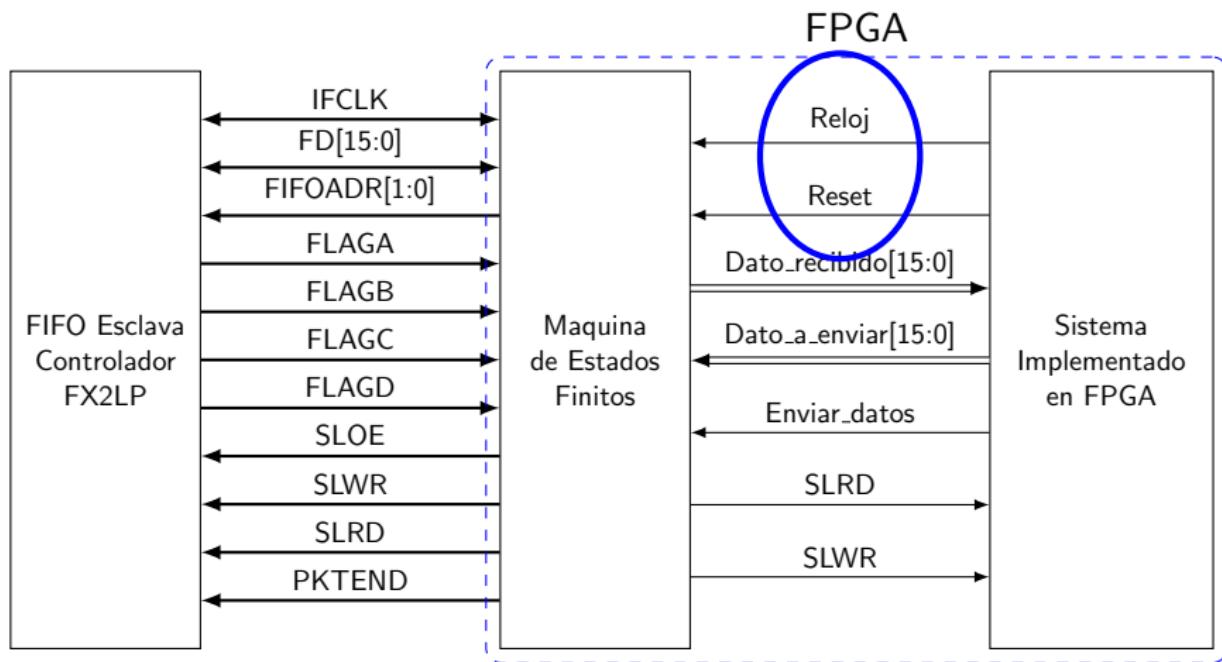
# Diagrama de señales entre la Interfaz USB y el FPGA



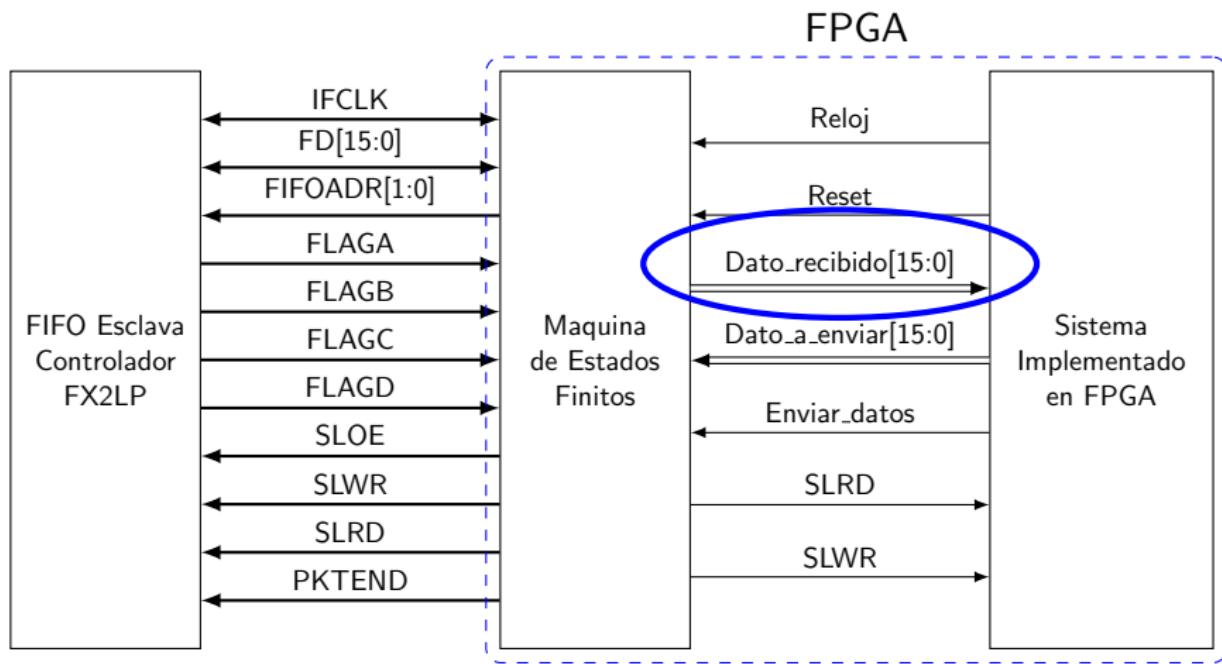
# Diagrama de señales entre la Interfaz USB y el FPGA



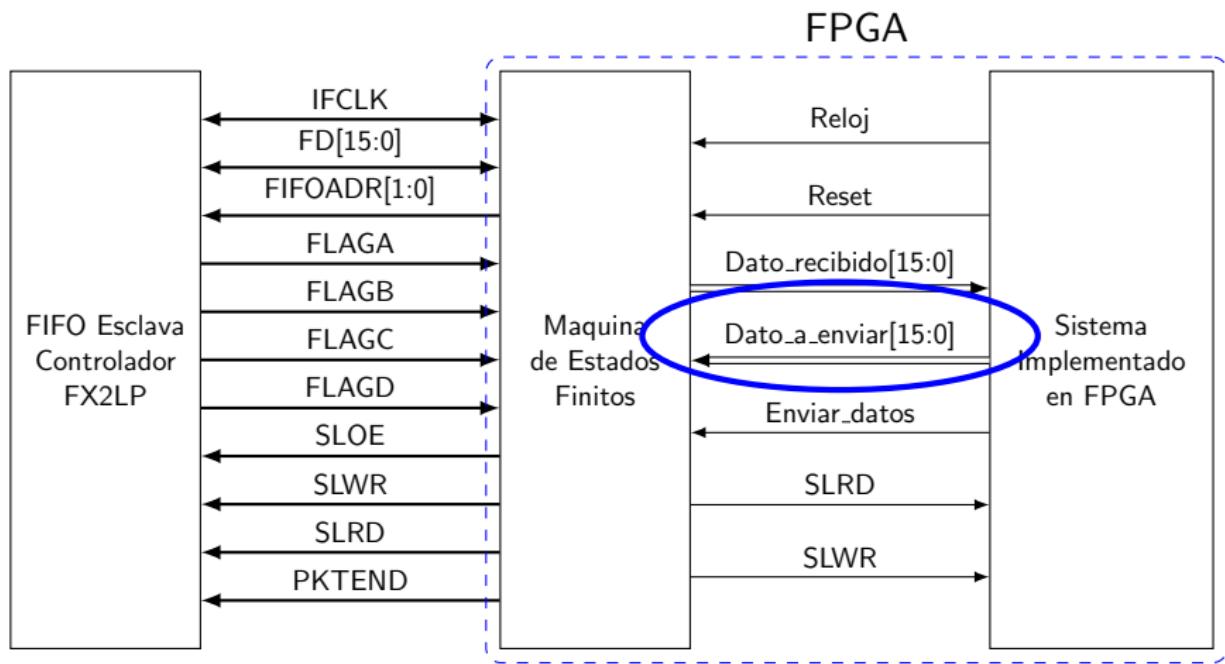
# Diagrama de señales entre la Interfaz USB y el FPGA



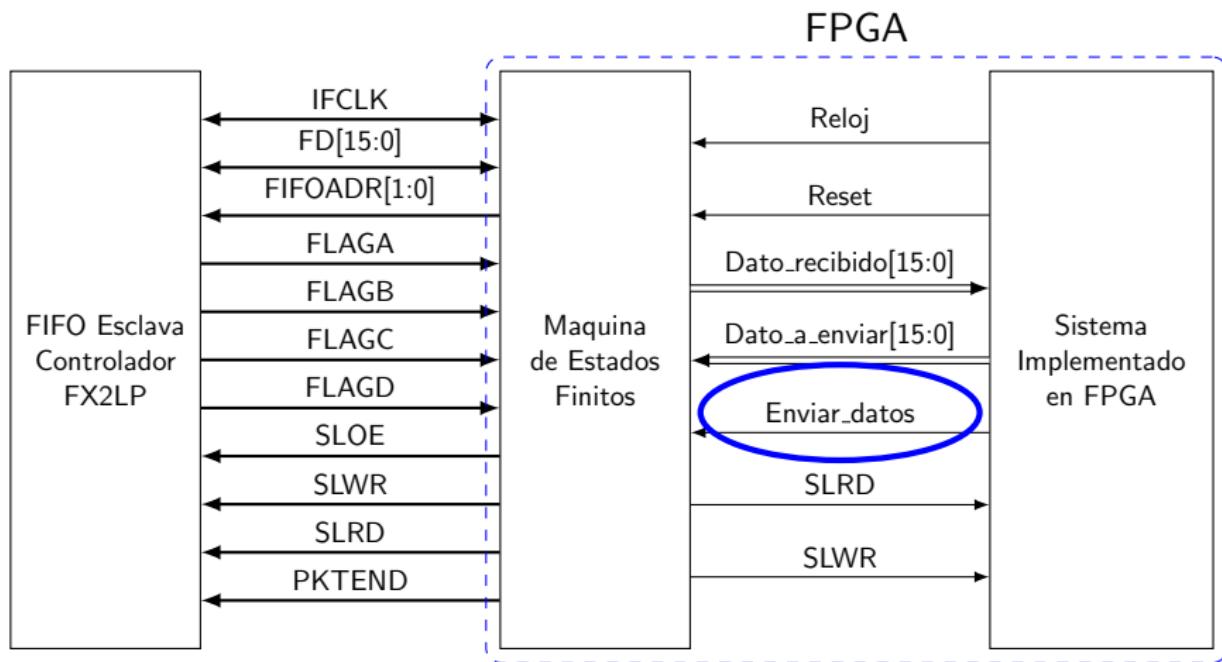
# Diagrama de señales entre la Interfaz USB y el FPGA



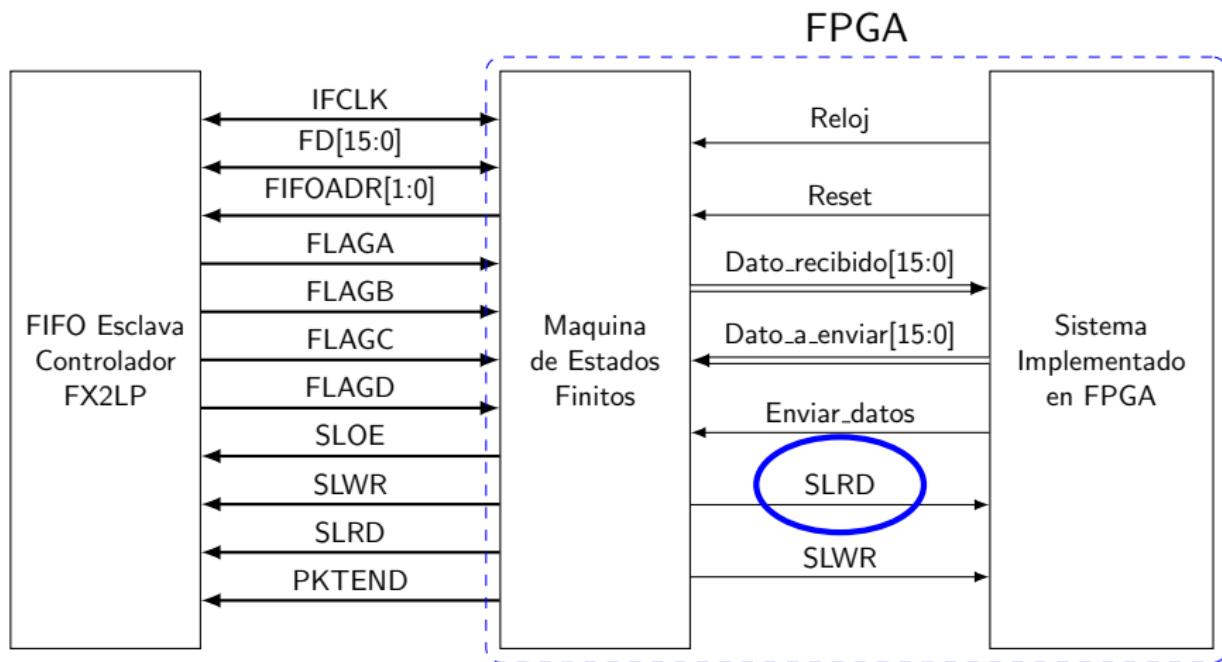
# Diagrama de señales entre la Interfaz USB y el FPGA



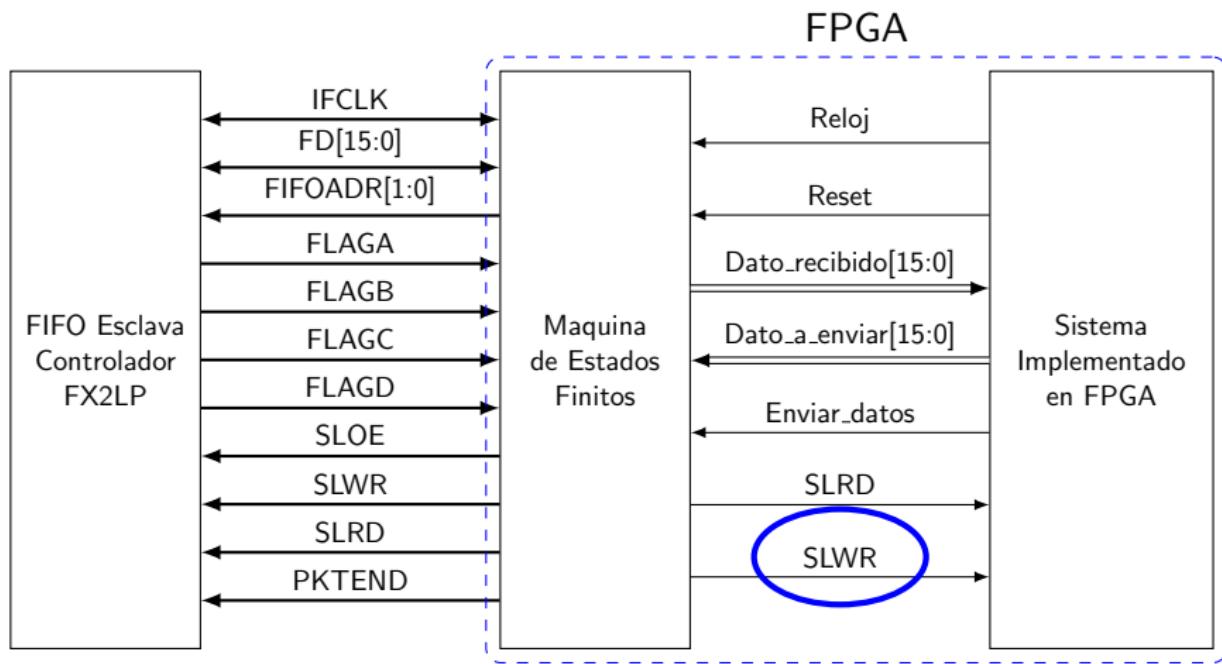
# Diagrama de señales entre la Interfaz USB y el FPGA



# Diagrama de señales entre la Interfaz USB y el FPGA

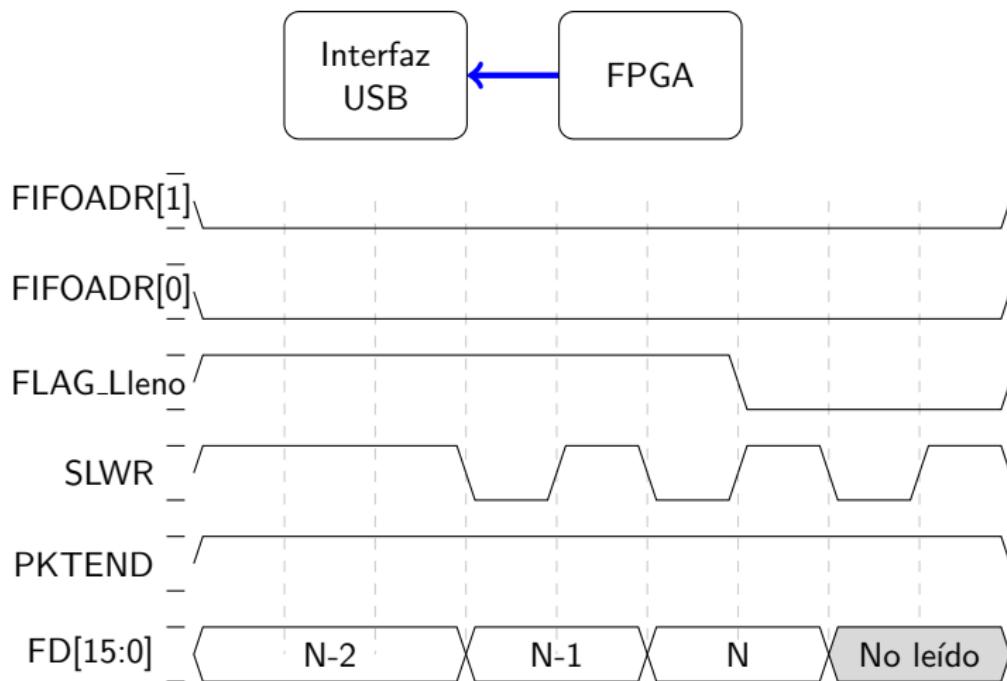


# Diagrama de señales entre la Interfaz USB y el FPGA



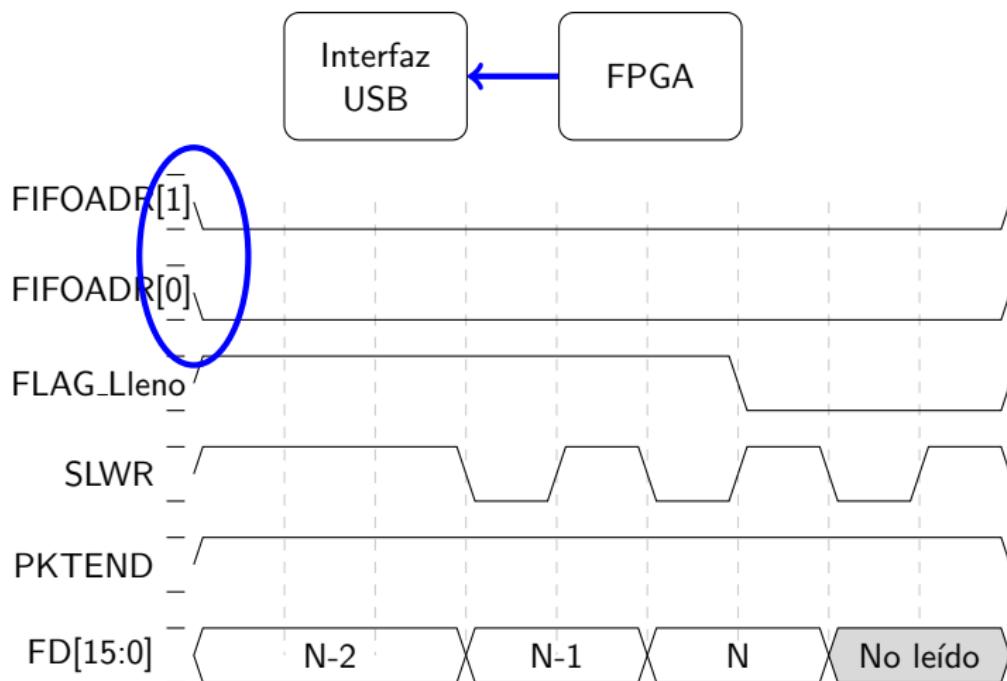
# Operaciones en la FIFO

## Escritura Asíncrona



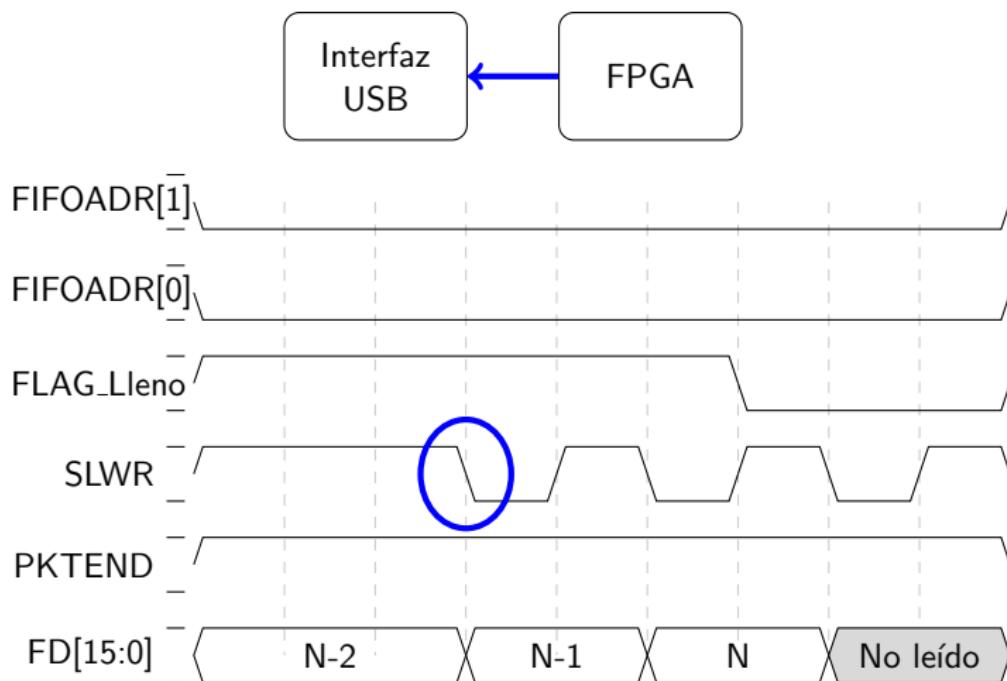
# Operaciones en la FIFO

## Escritura Asíncrona



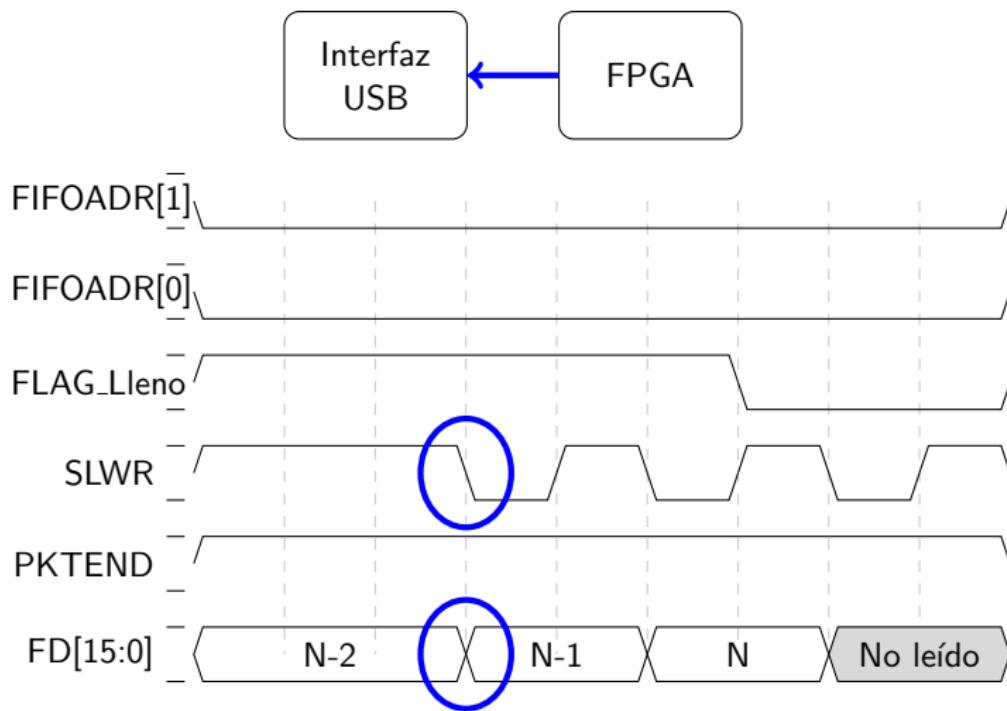
# Operaciones en la FIFO

## Escritura Asíncrona



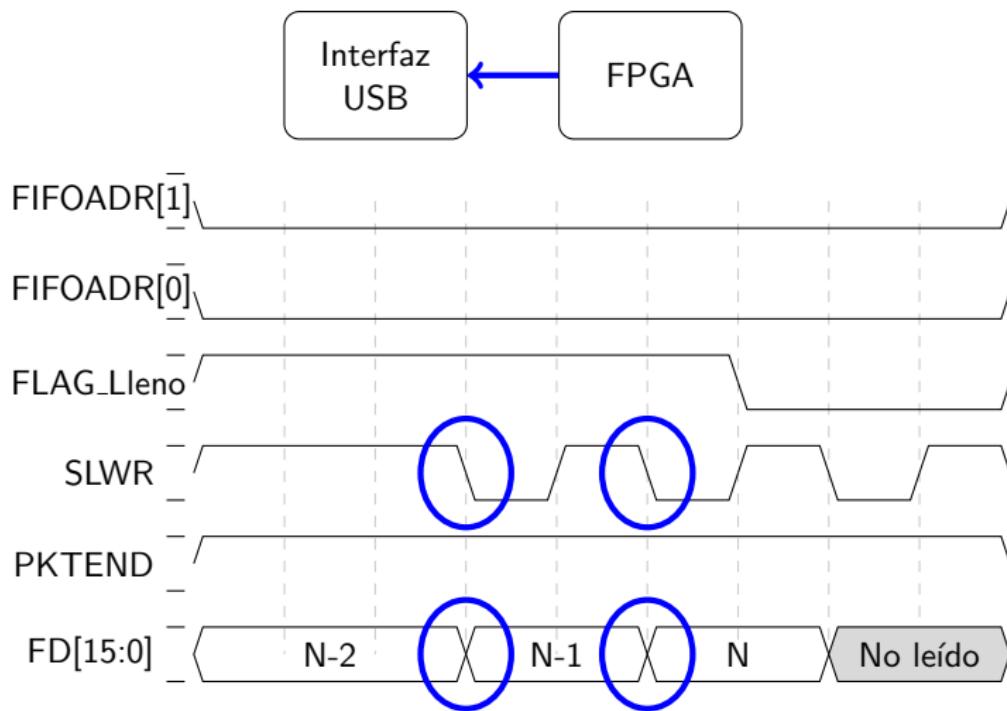
# Operaciones en la FIFO

## Escritura Asíncrona



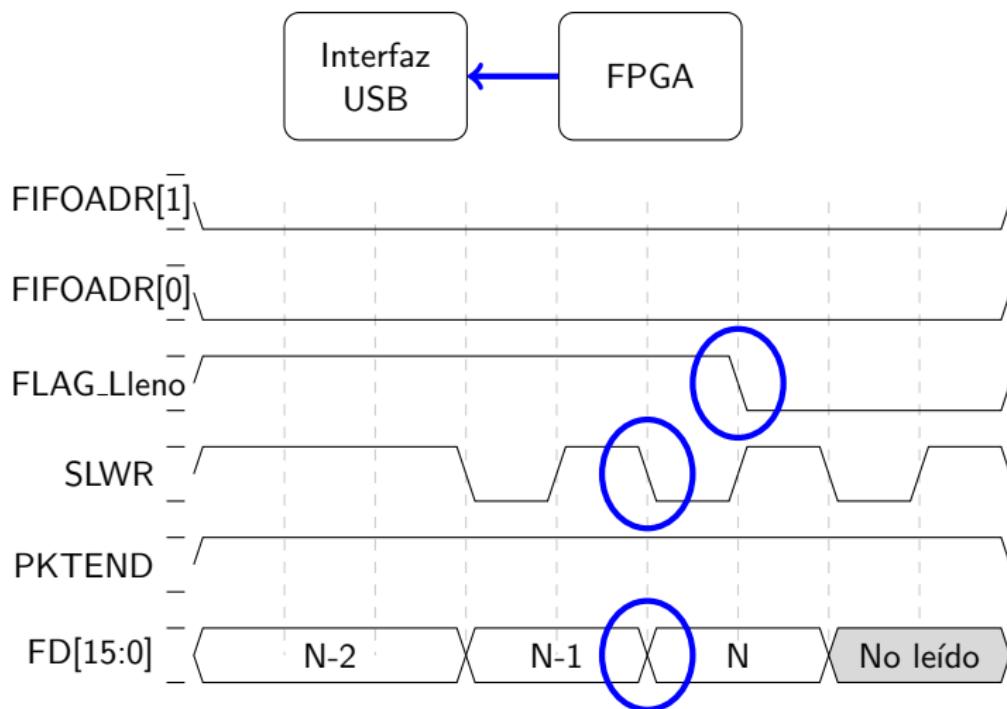
# Operaciones en la FIFO

## Escritura Asíncrona



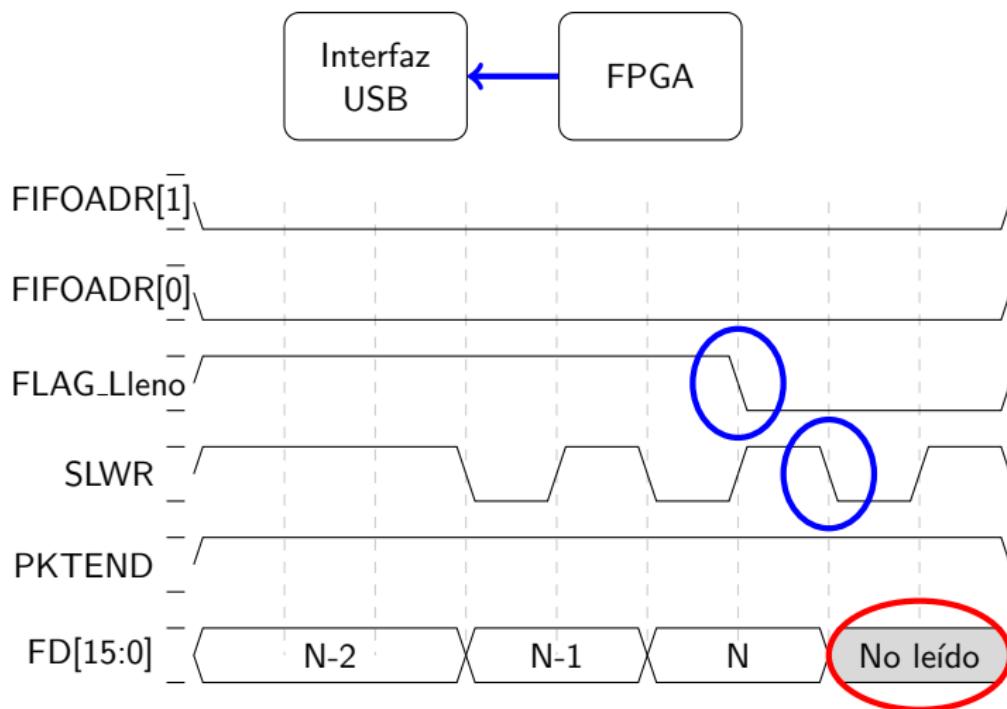
# Operaciones en la FIFO

## Escritura Asíncrona



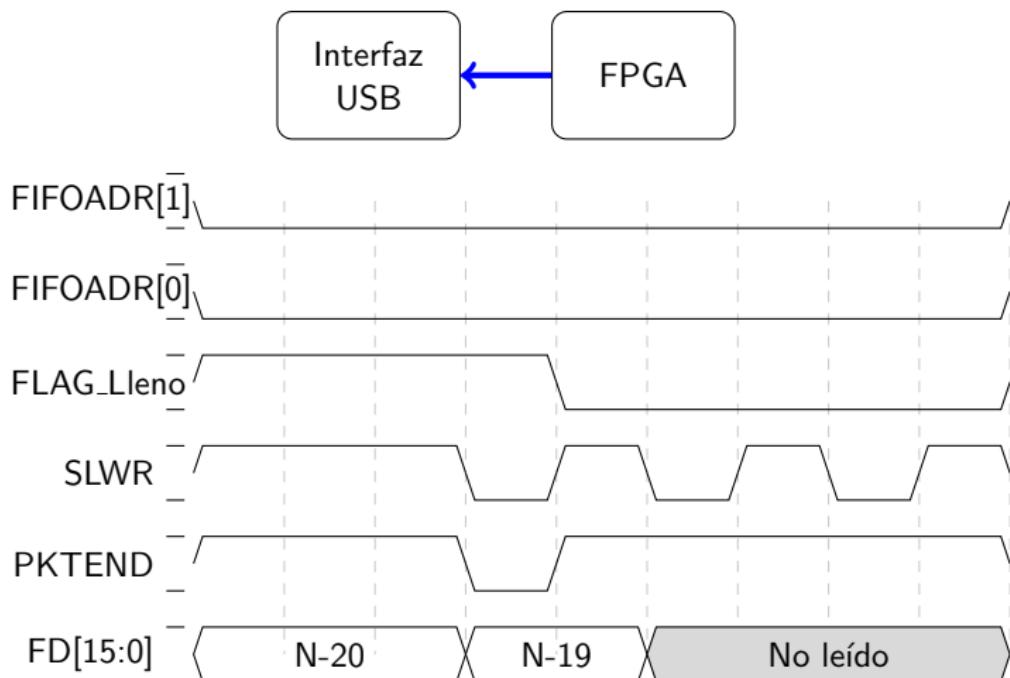
# Operaciones en la FIFO

## Escritura Asíncrona



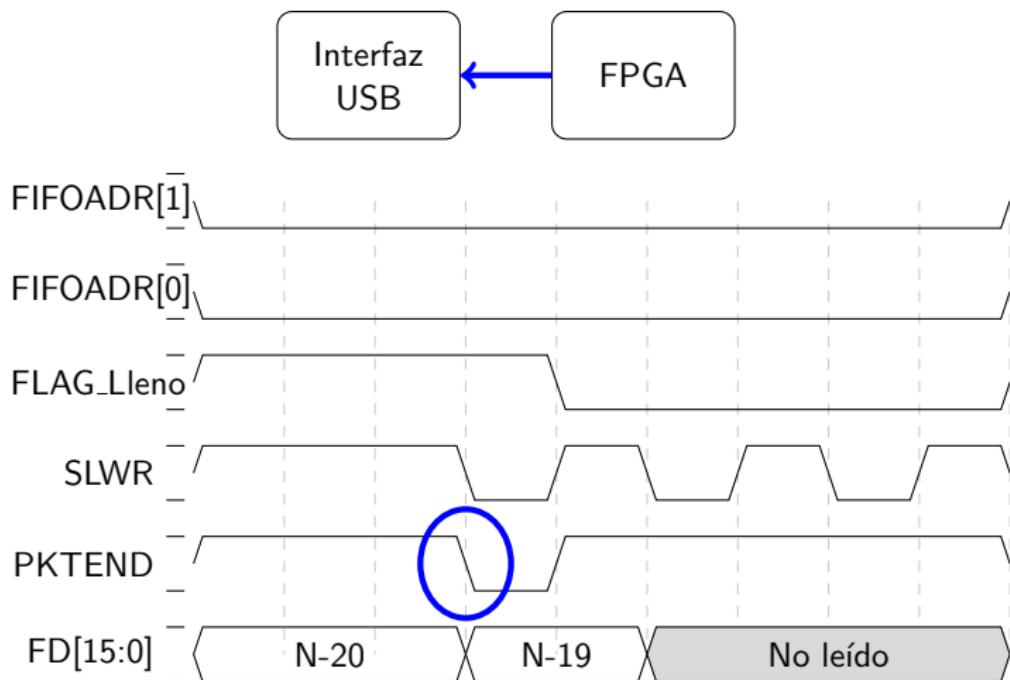
# Operaciones en la FIFO

## Escritura Asíncrona



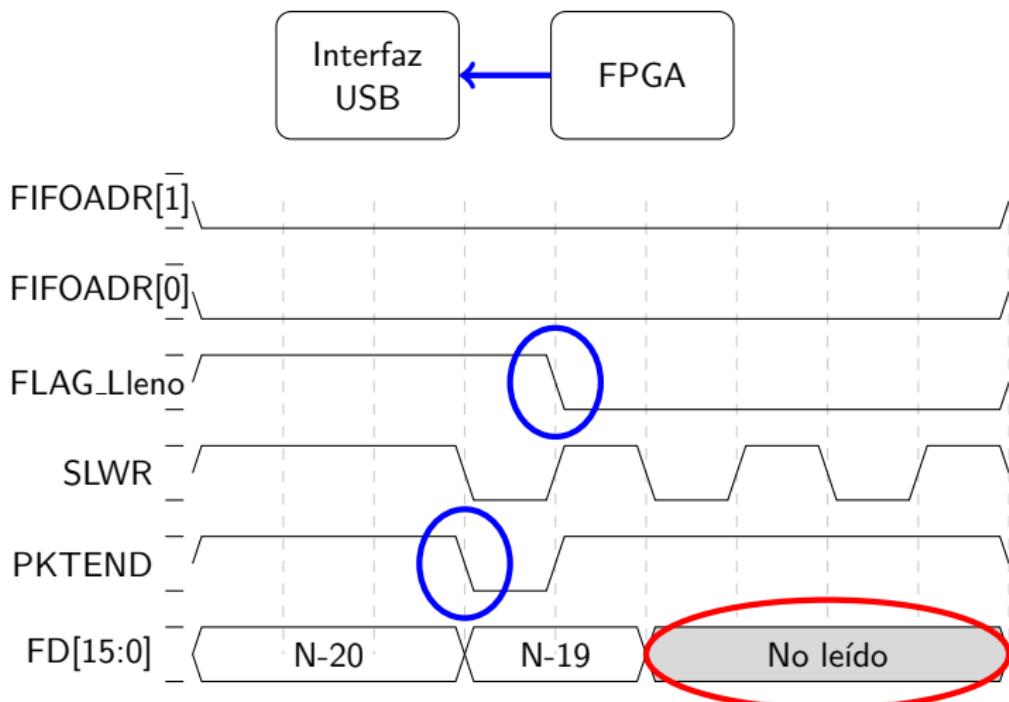
# Operaciones en la FIFO

## Escritura Asíncrona



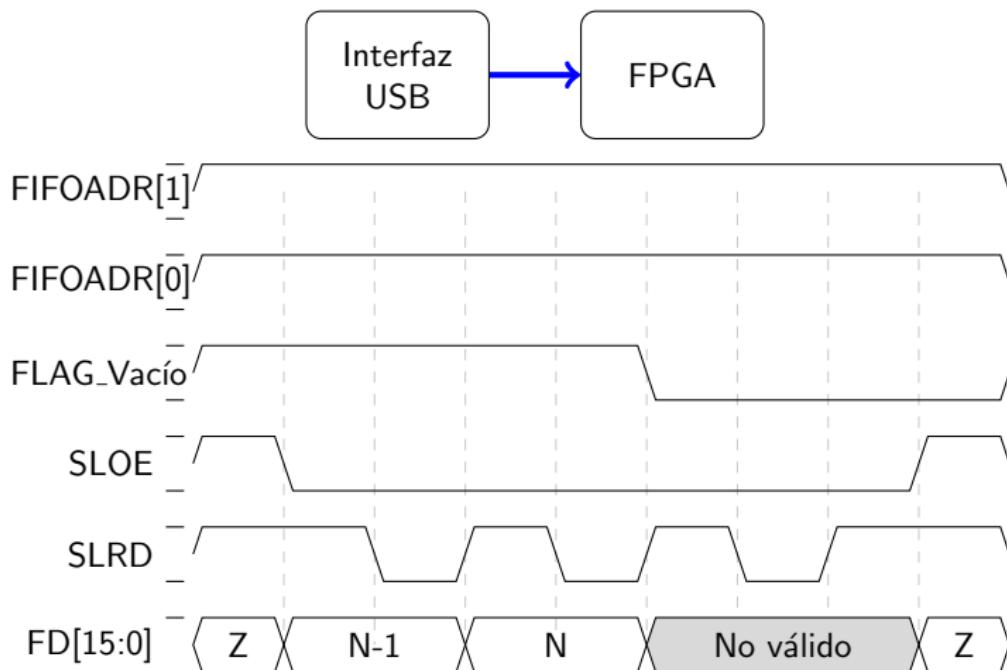
# Operaciones en la FIFO

## Escritura Asíncrona



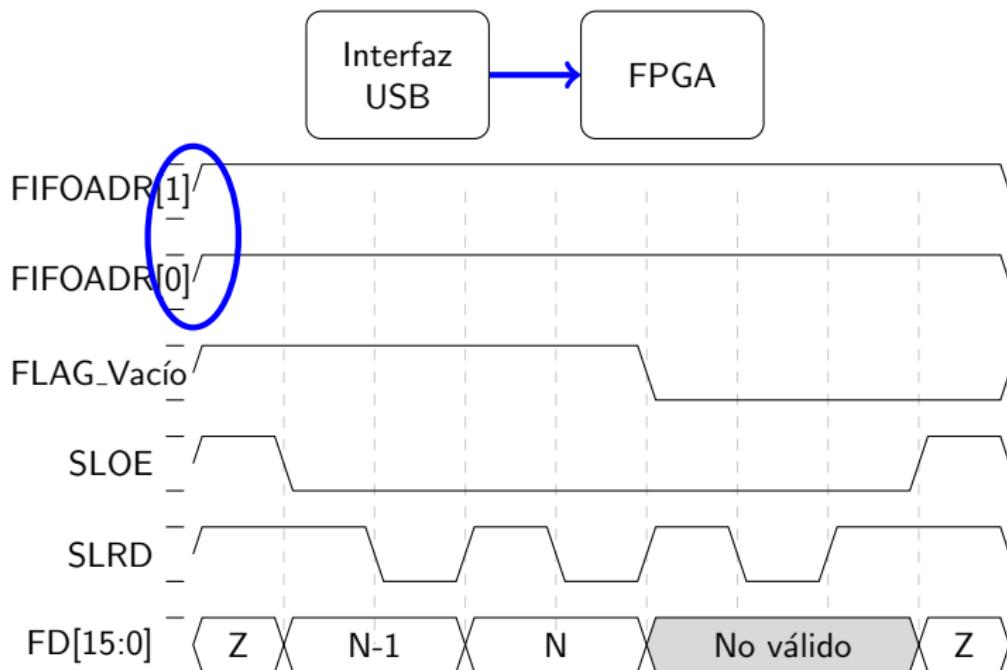
# Operaciones en la FIFO

## Lectura Asíncrona



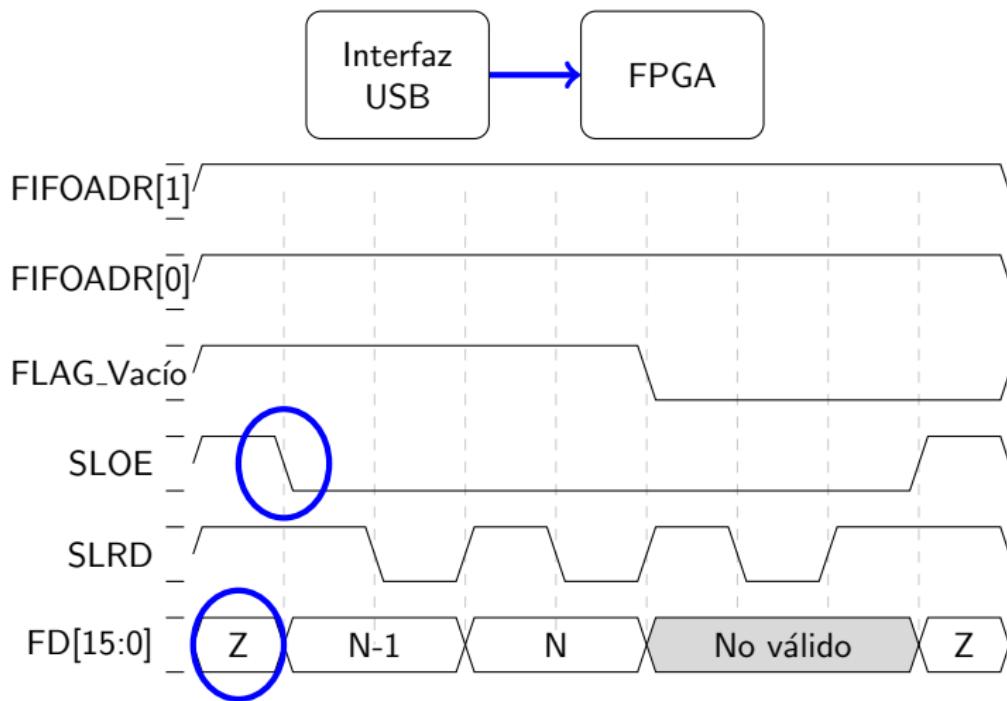
# Operaciones en la FIFO

## Lectura Asíncrona



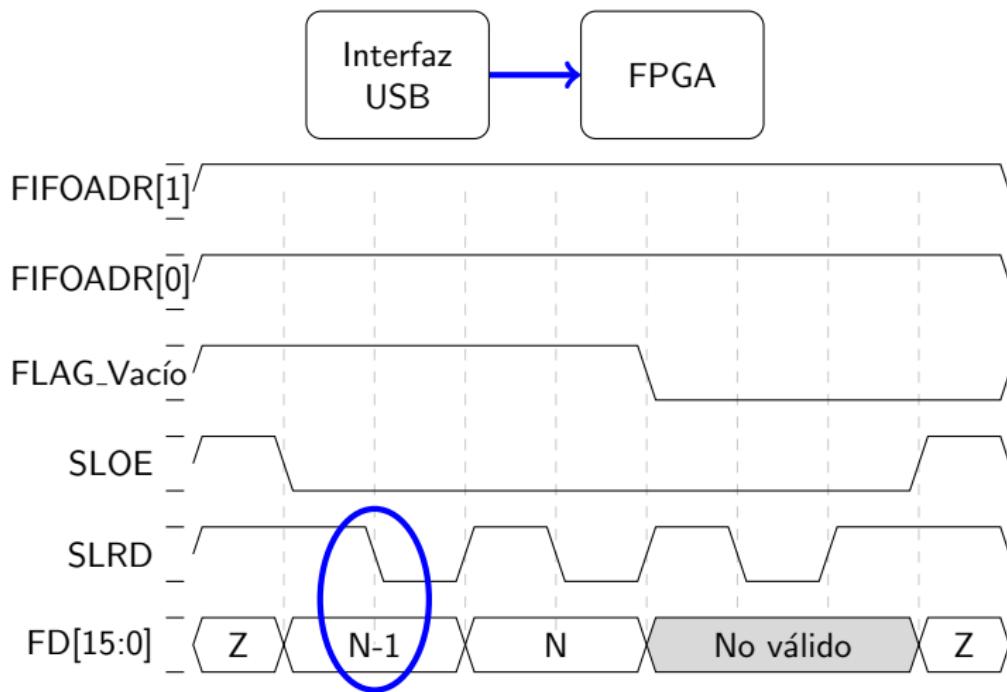
# Operaciones en la FIFO

## Lectura Asíncrona



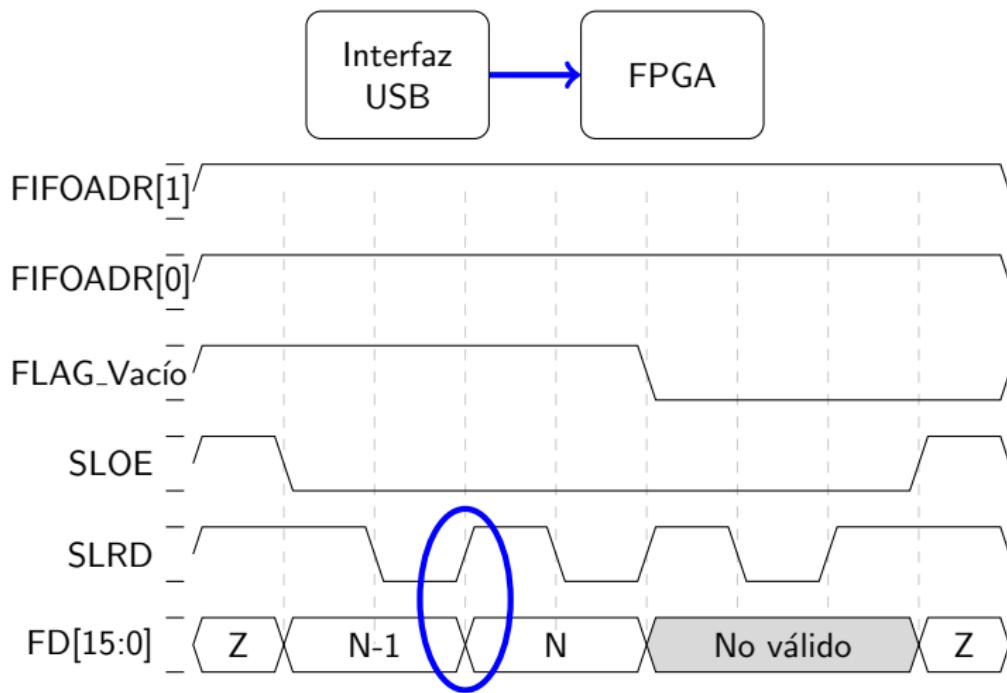
# Operaciones en la FIFO

## Lectura Asíncrona



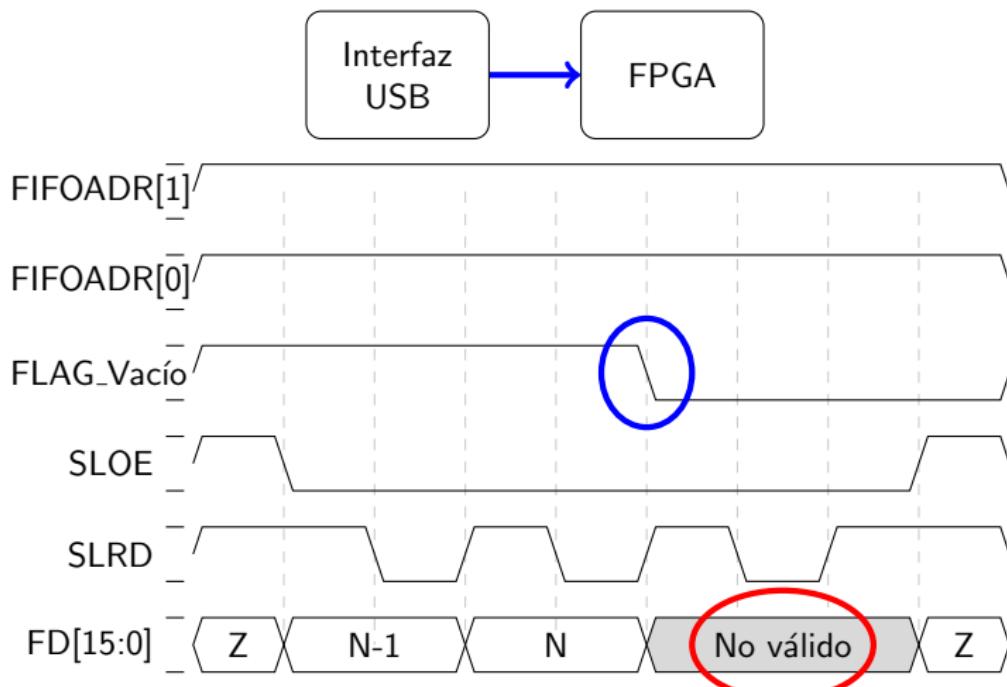
# Operaciones en la FIFO

## Lectura Asíncrona

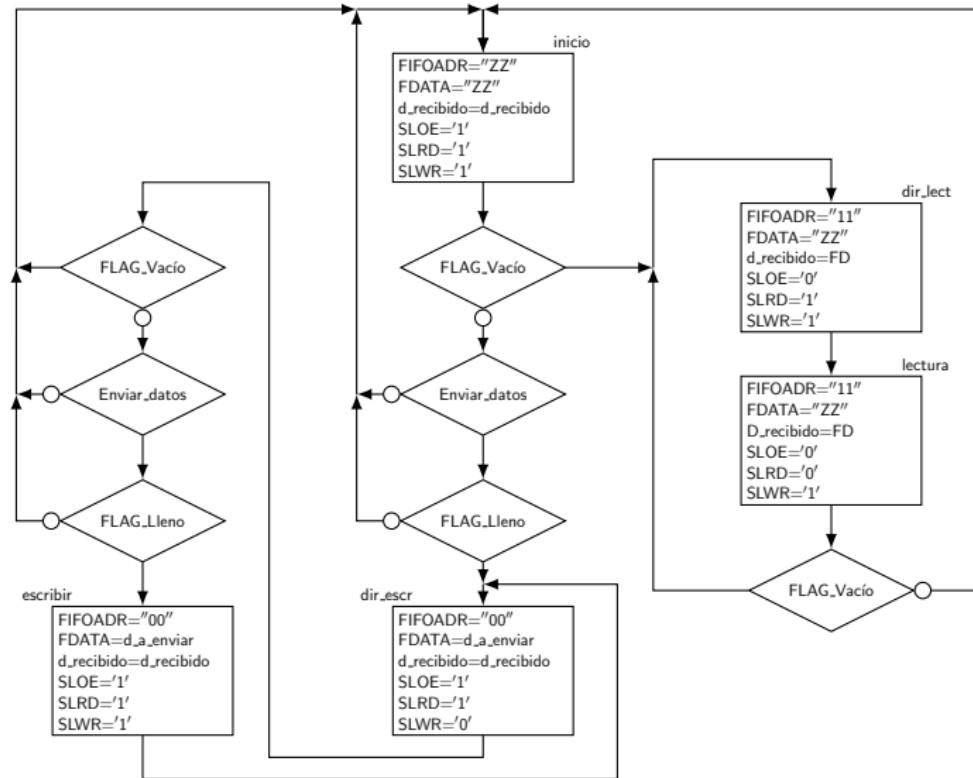


# Operaciones en la FIFO

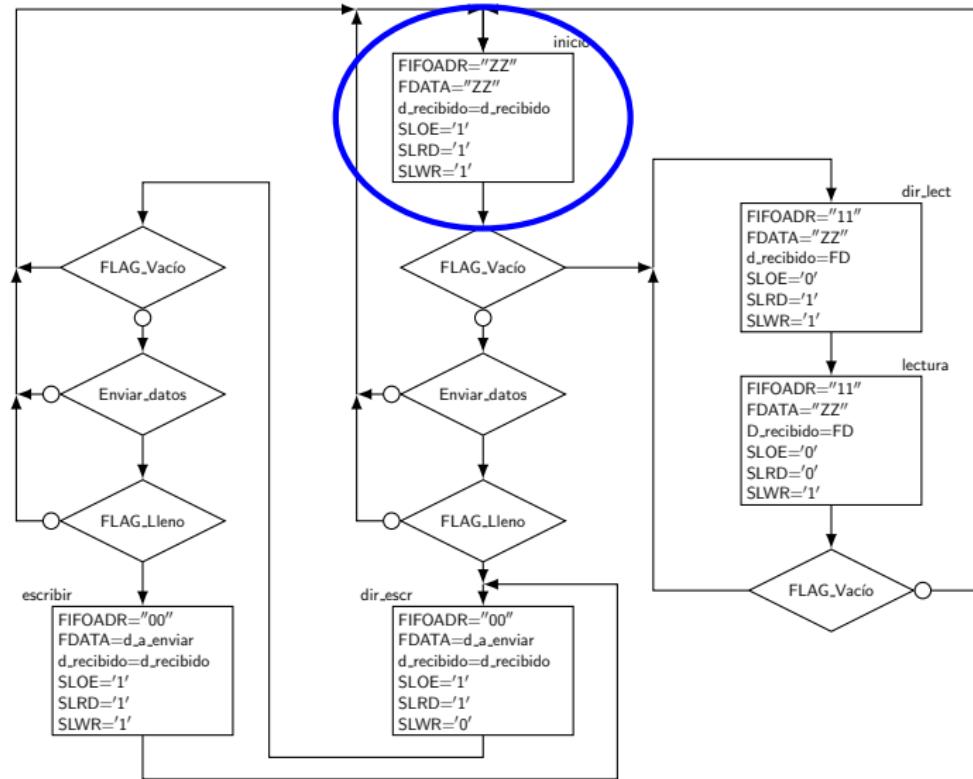
## Lectura Asíncrona



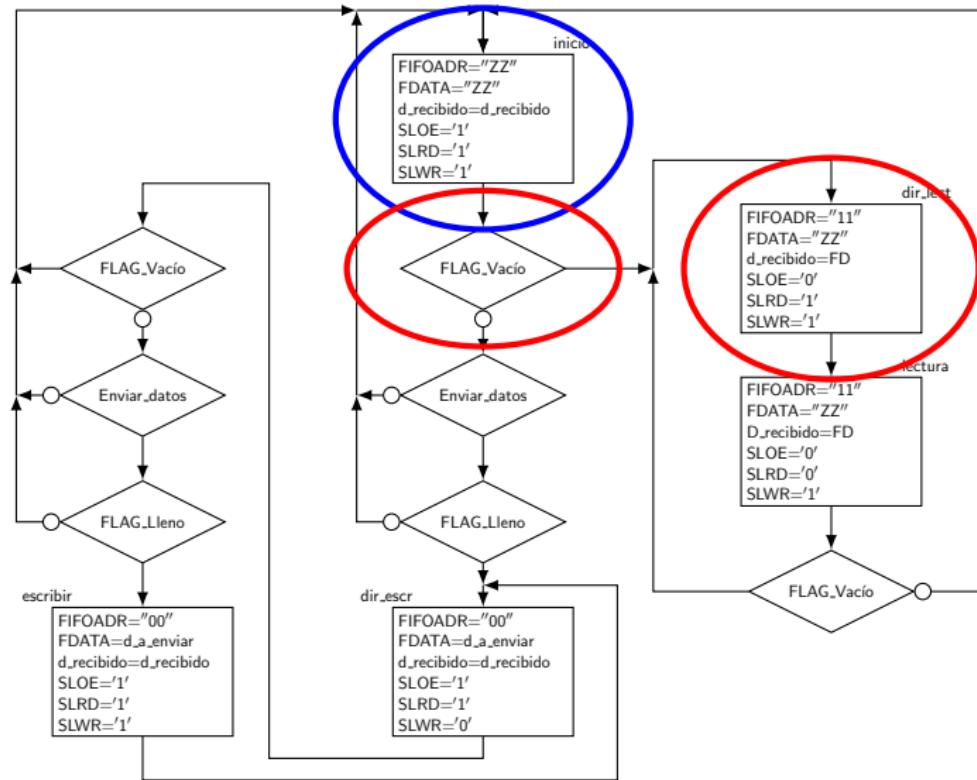
# Máquina de estados algorítmica sintetizada en FPGA



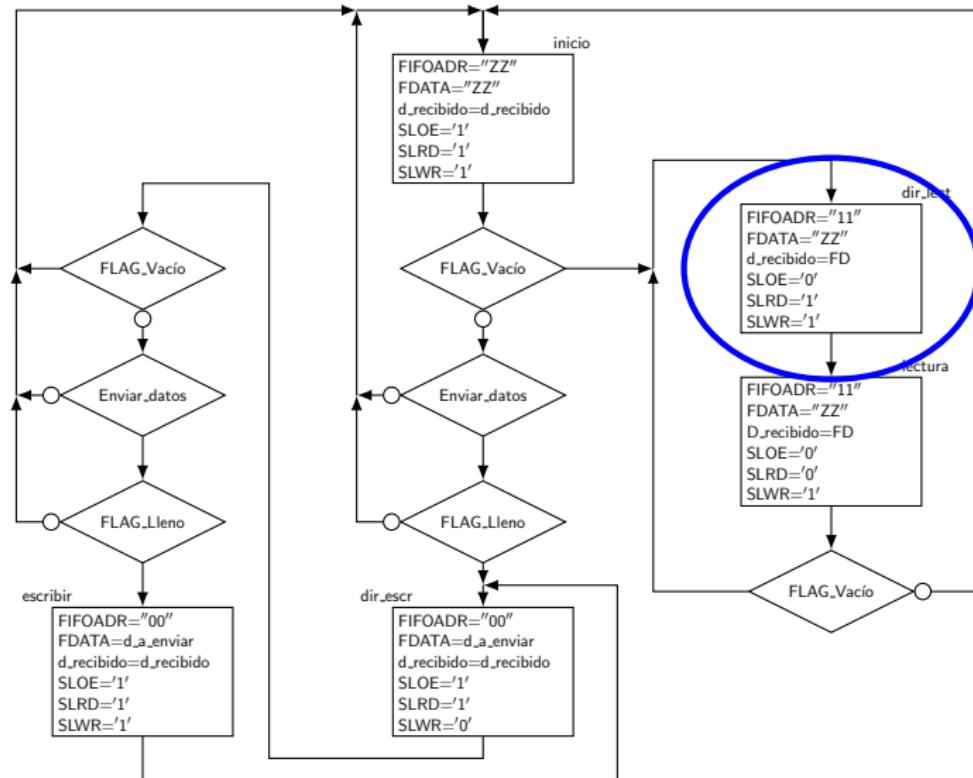
# Máquina de estados algorítmica sintetizada en FPGA



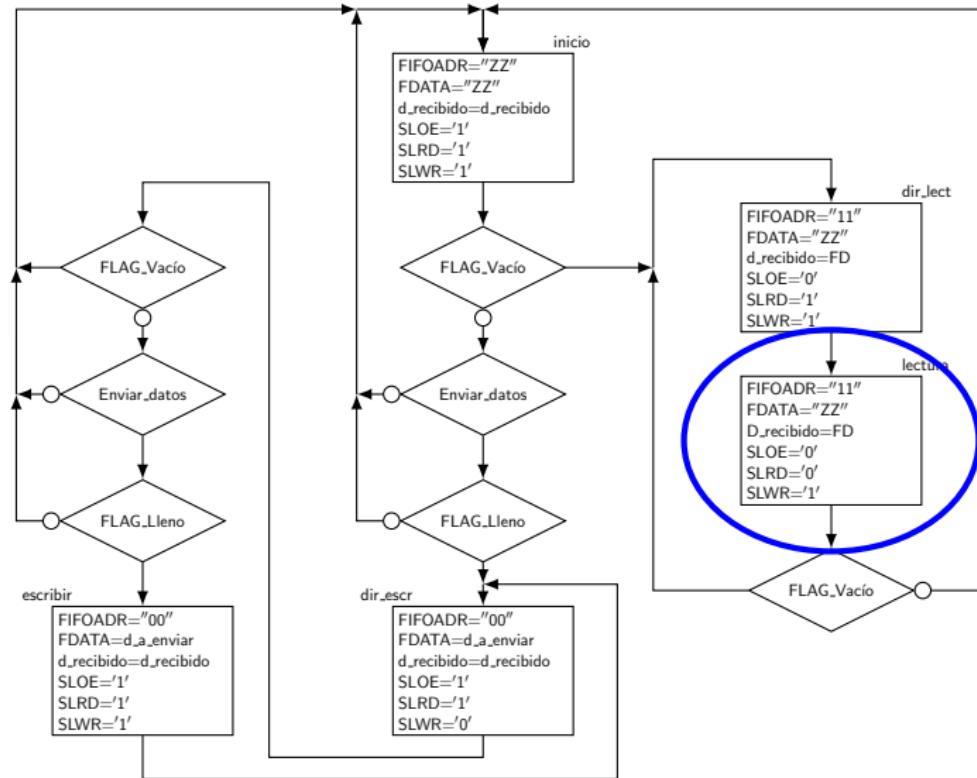
# Máquina de estados algorítmica sintetizada en FPGA



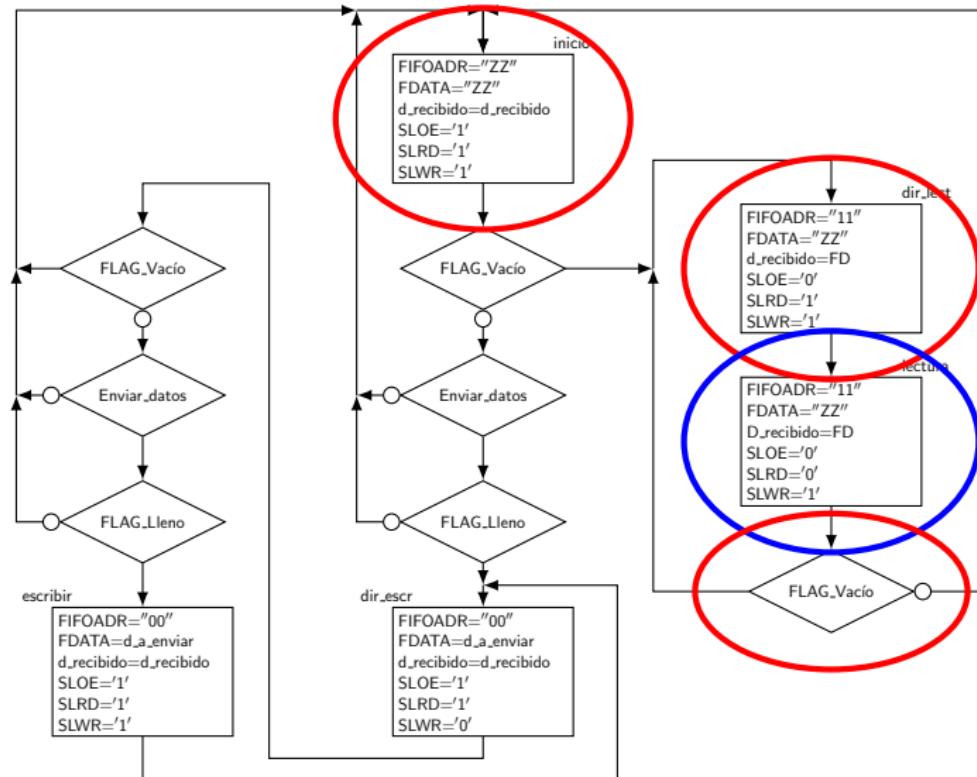
# Máquina de estados algorítmica sintetizada en FPGA



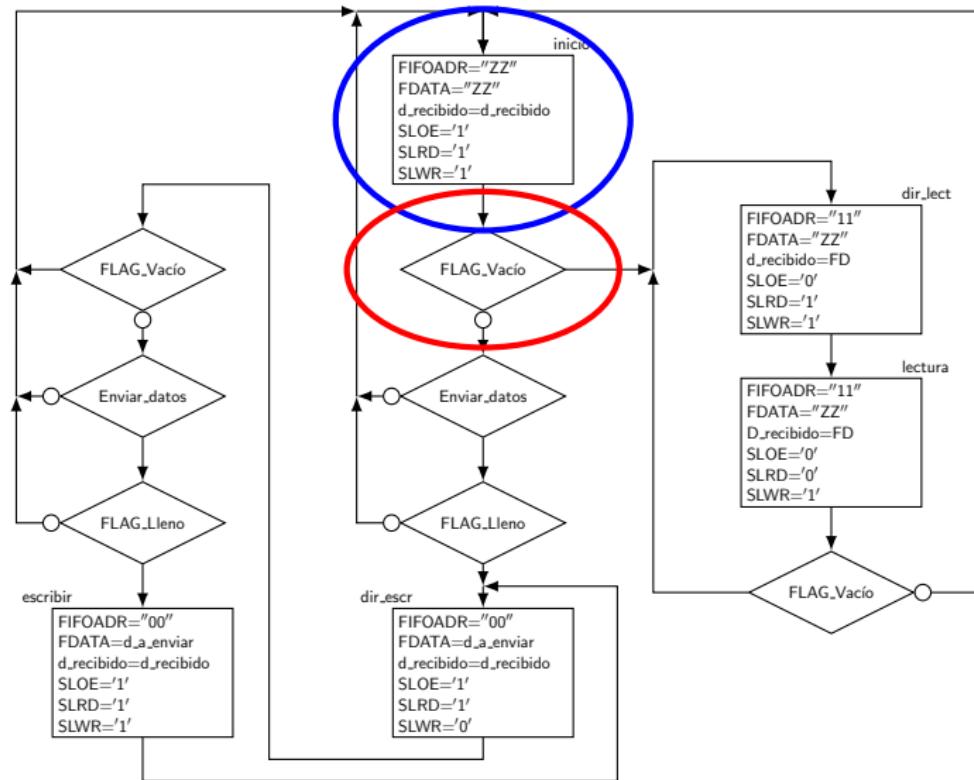
# Máquina de estados algorítmica sintetizada en FPGA



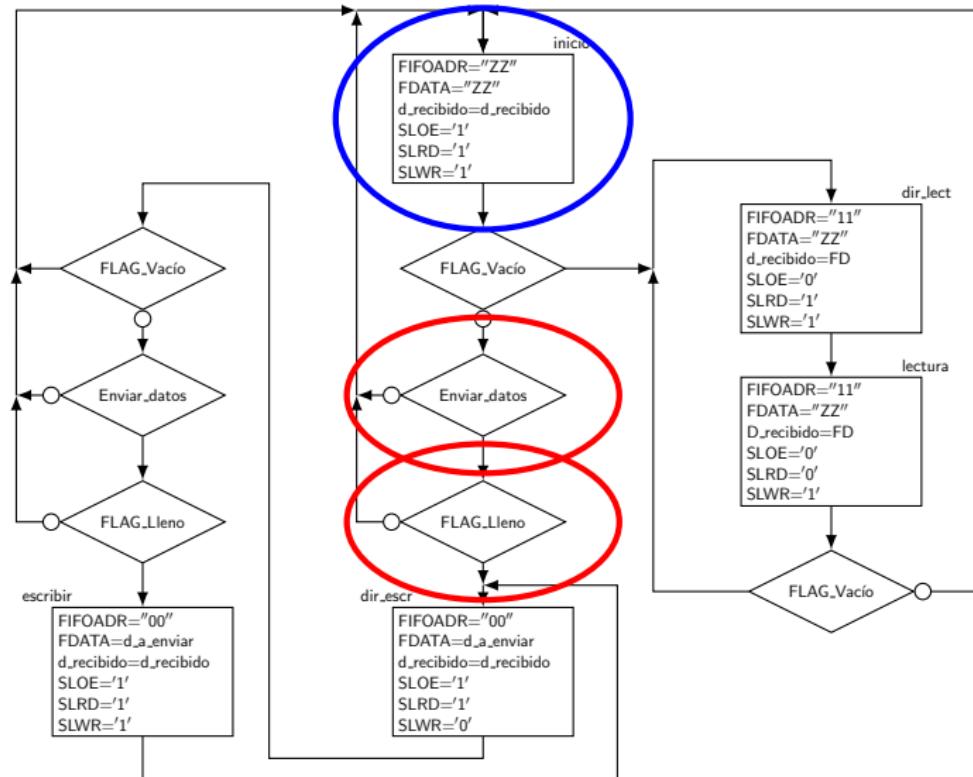
# Máquina de estados algorítmica sintetizada en FPGA



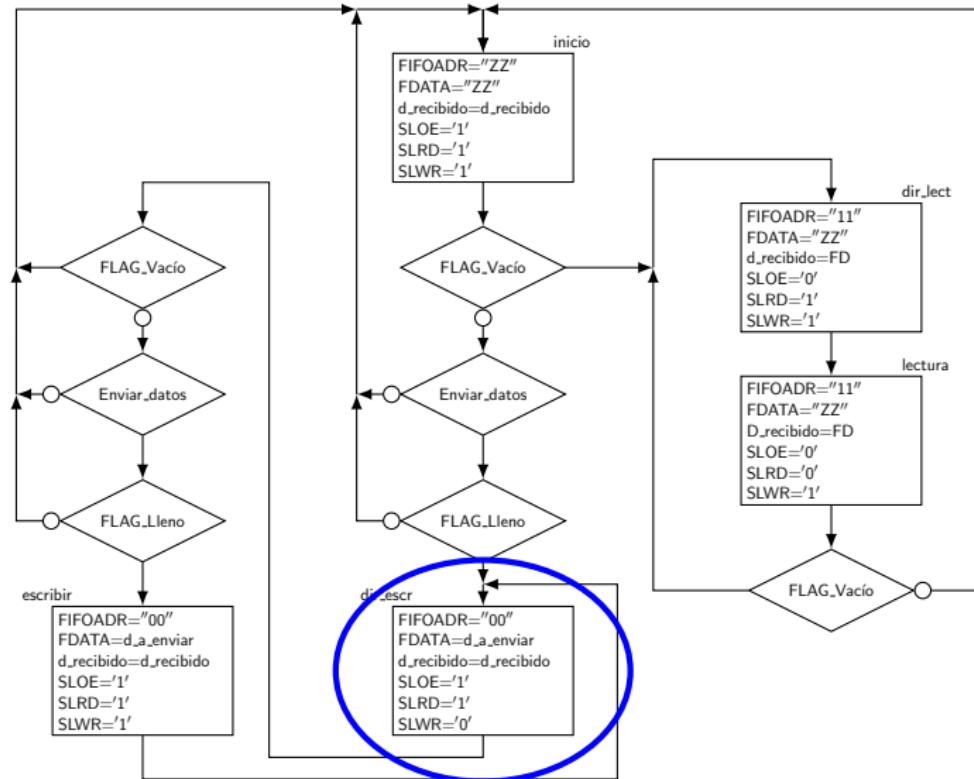
# Máquina de estados algorítmica sintetizada en FPGA



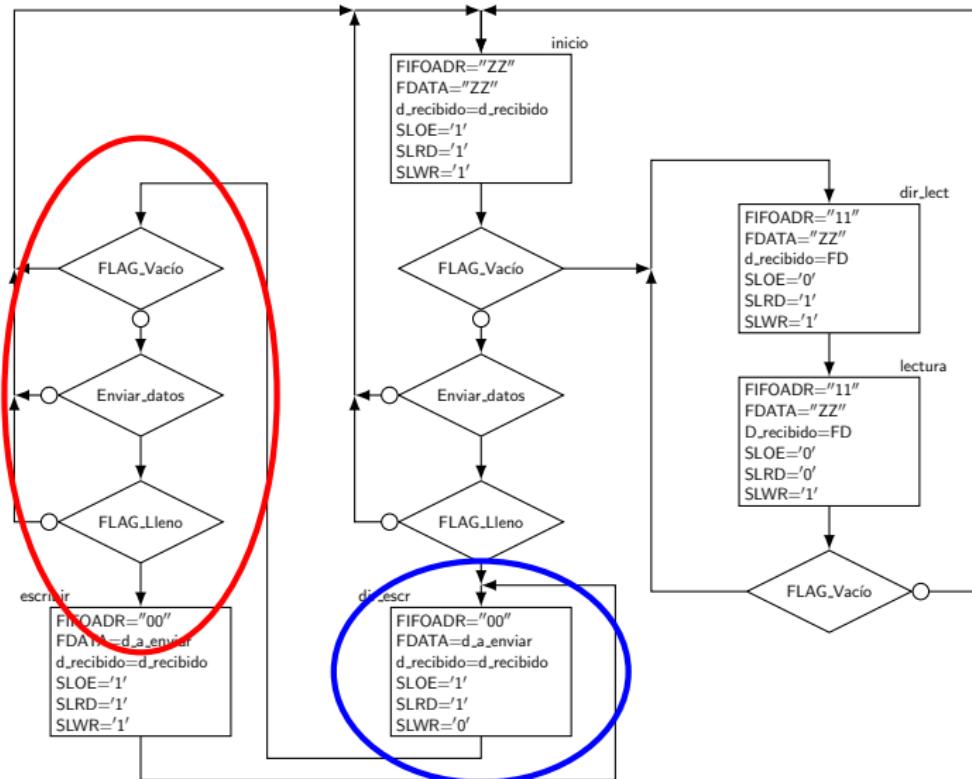
# Máquina de estados algorítmica sintetizada en FPGA



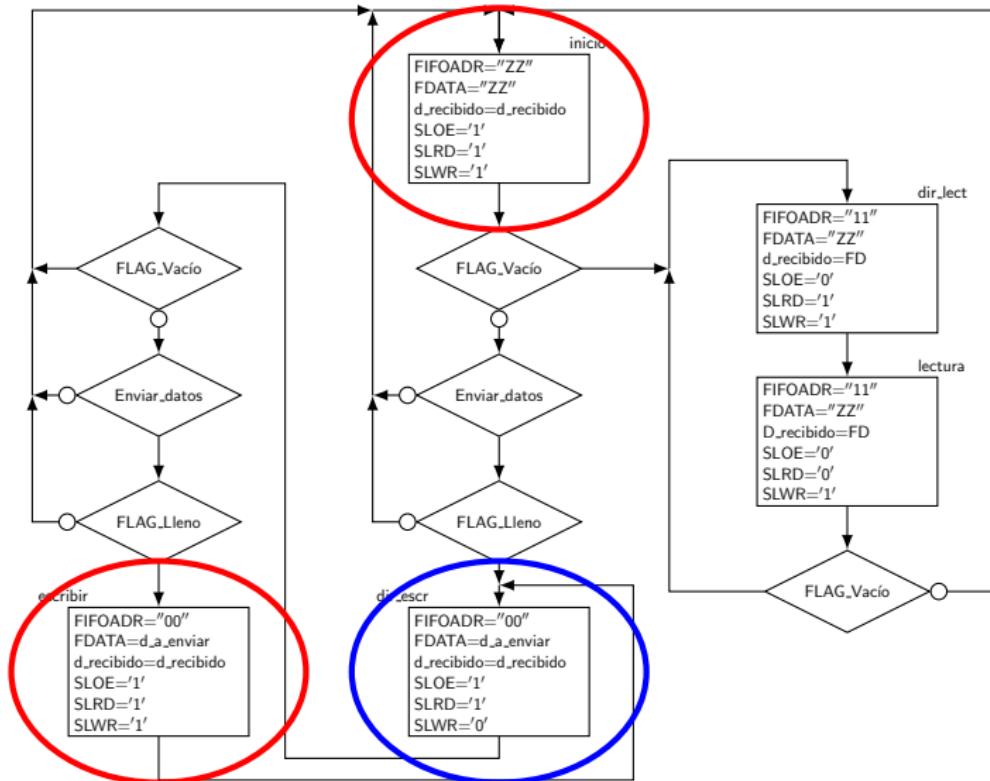
# Máquina de estados algorítmica sintetizada en FPGA



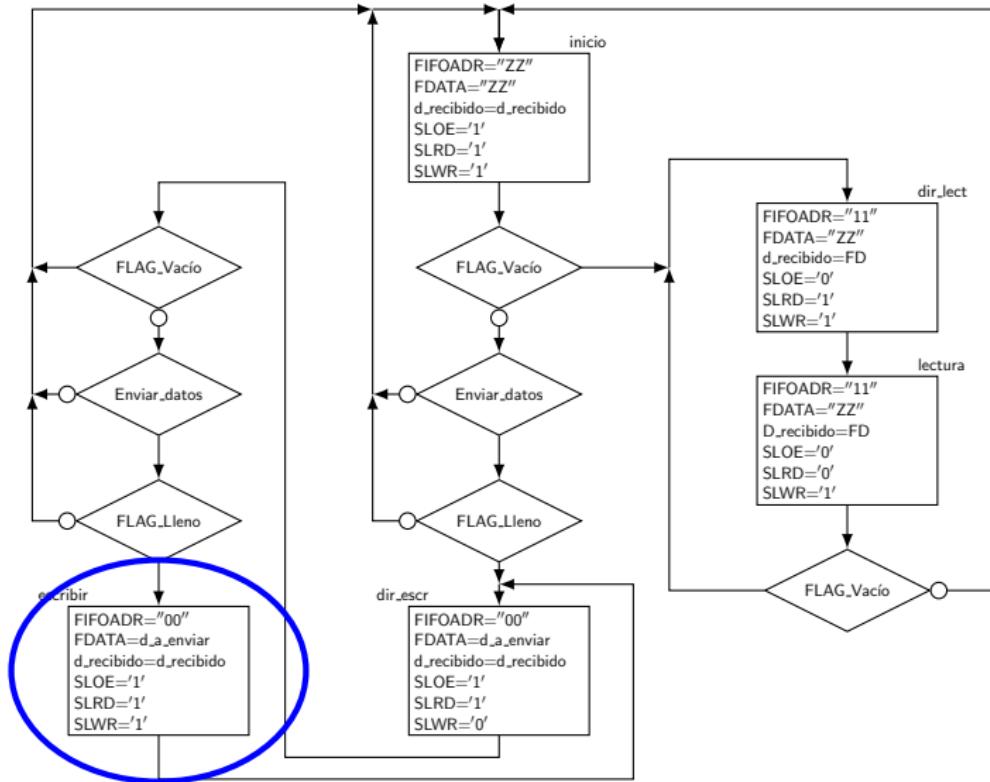
# Máquina de estados algorítmica sintetizada en FPGA



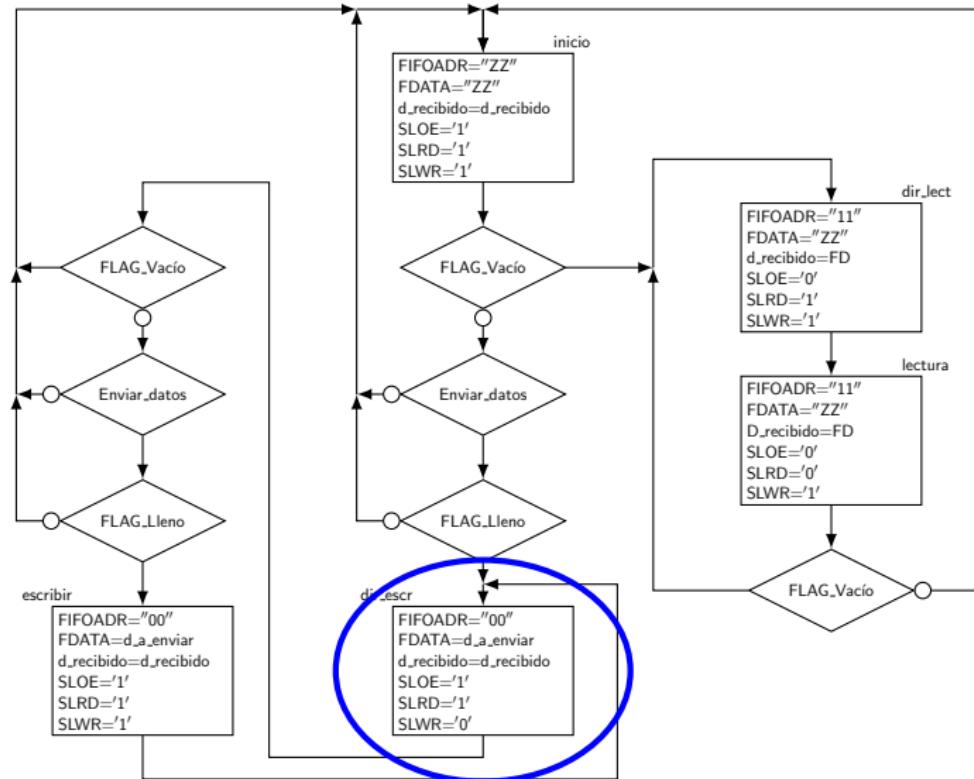
# Máquina de estados algorítmica sintetizada en FPGA



# Máquina de estados algorítmica sintetizada en FPGA

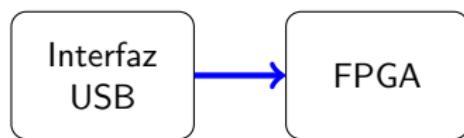


# Máquina de estados algorítmica sintetizada en FPGA



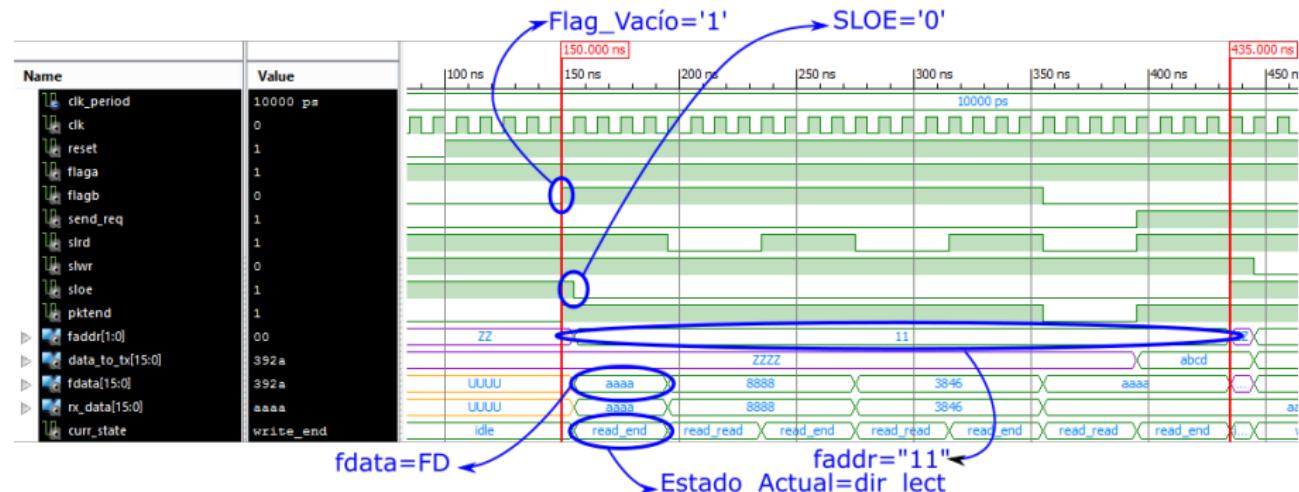
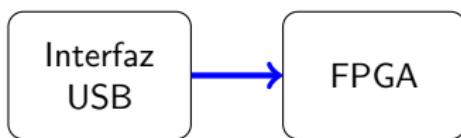
## Verificación Funcional

## Operación de lectura



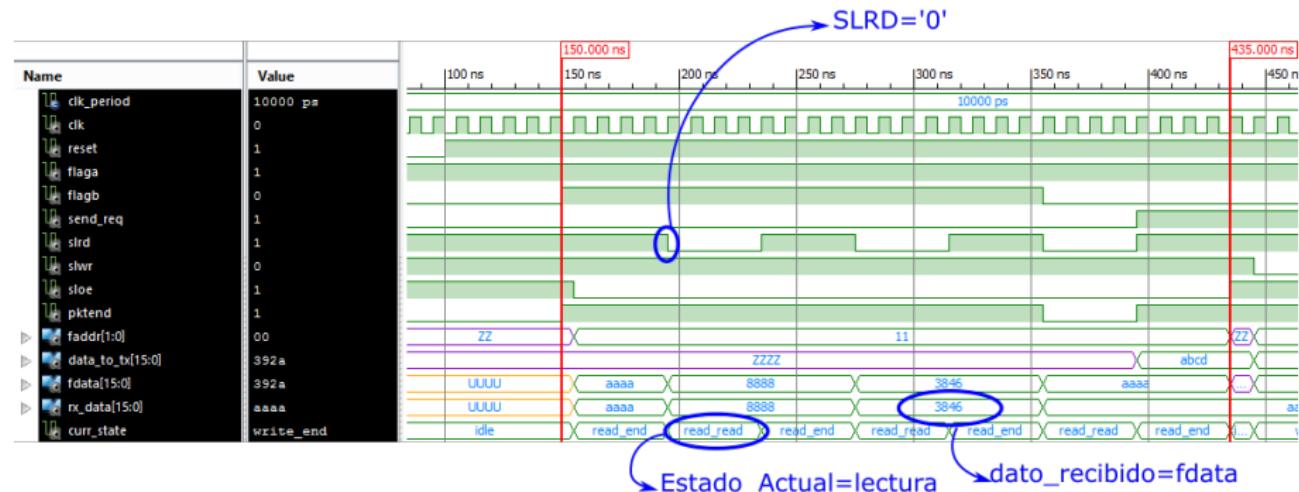
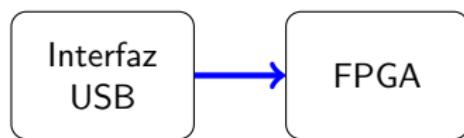
# Verificación Funcional

## Operación de lectura



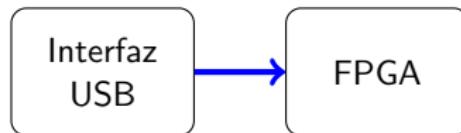
# Verificación Funcional

## Operación de lectura



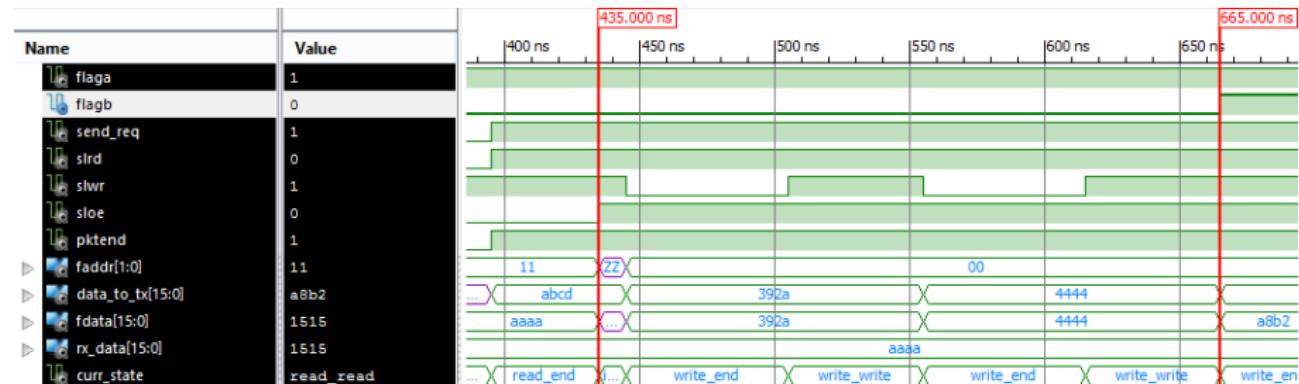
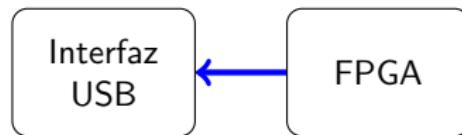
# Verificación Funcional

## Operación de lectura



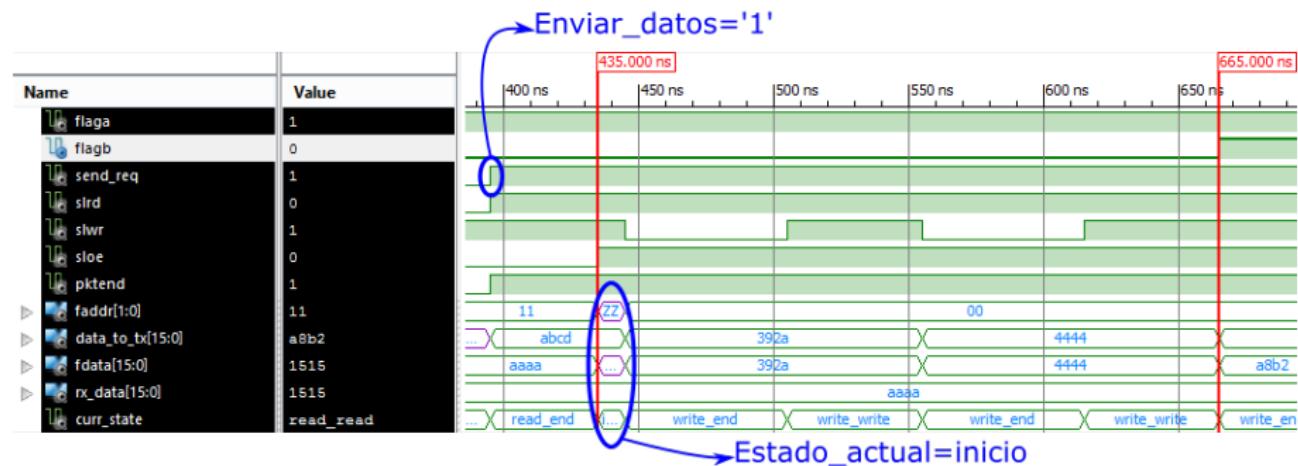
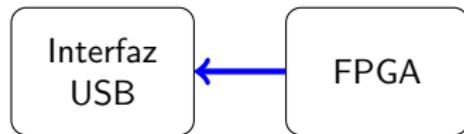
# Verificación Funcional

## Operación de escritura



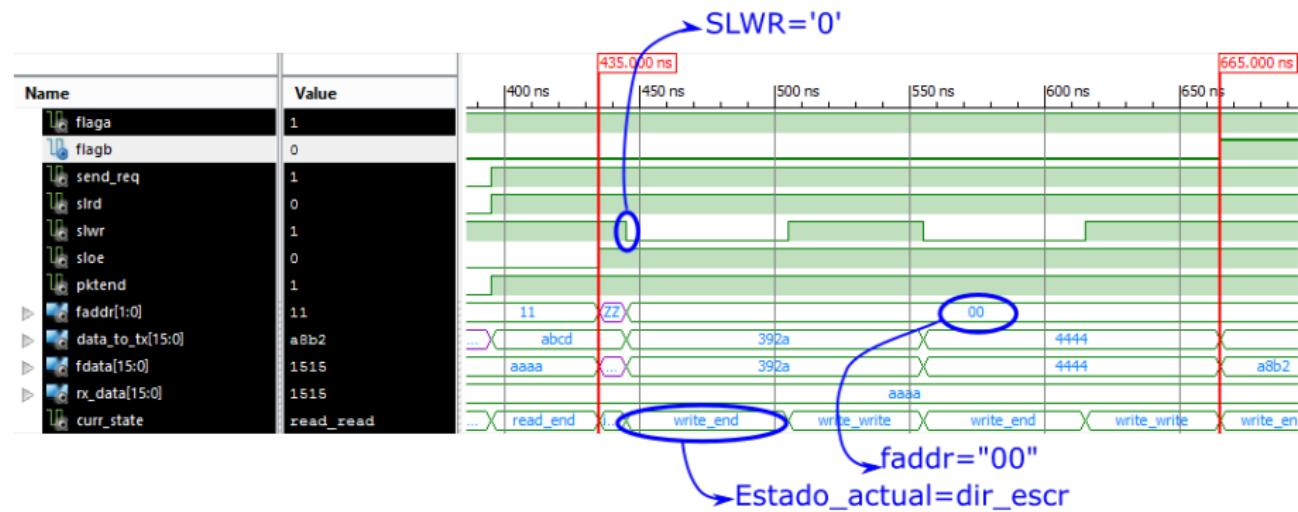
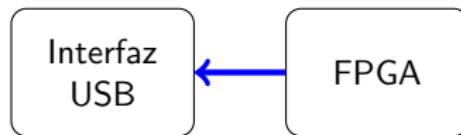
# Verificación Funcional

## Operación de escritura



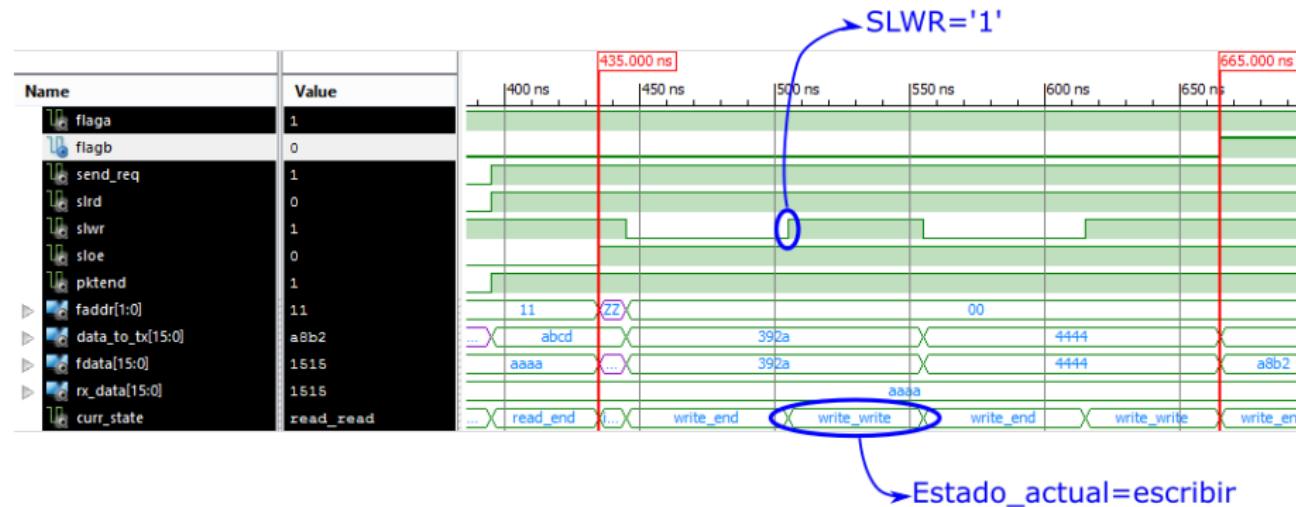
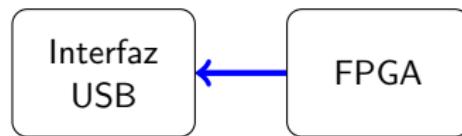
# Verificación Funcional

## Operación de escritura



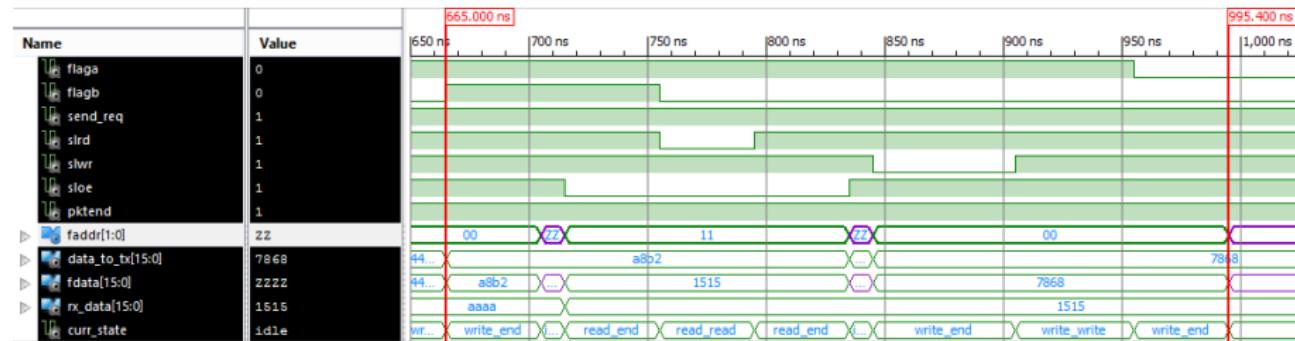
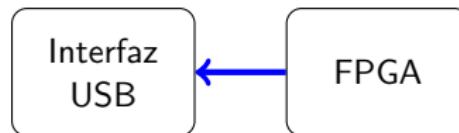
# Verificación Funcional

## Operación de escritura



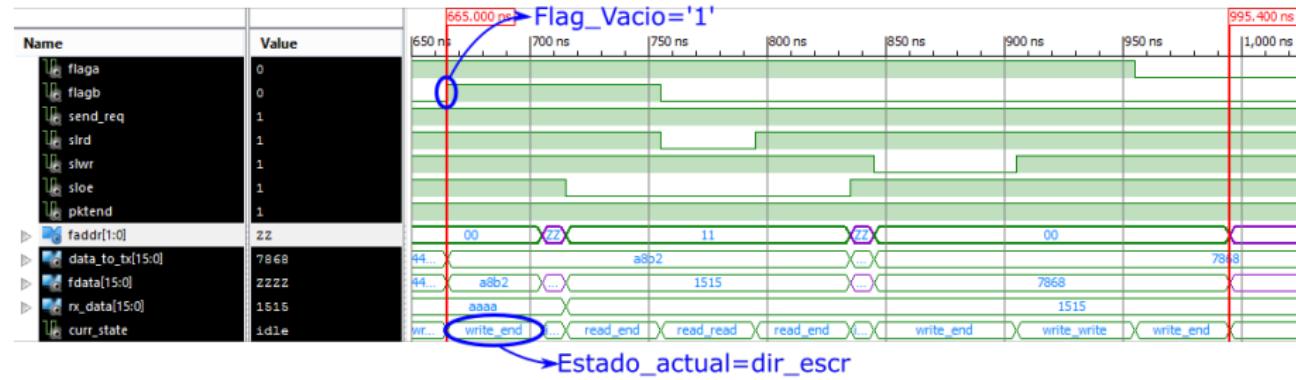
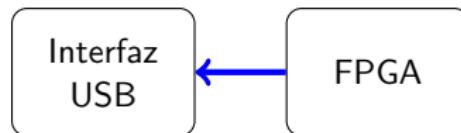
# Verificación Funcional

## Operación de escritura



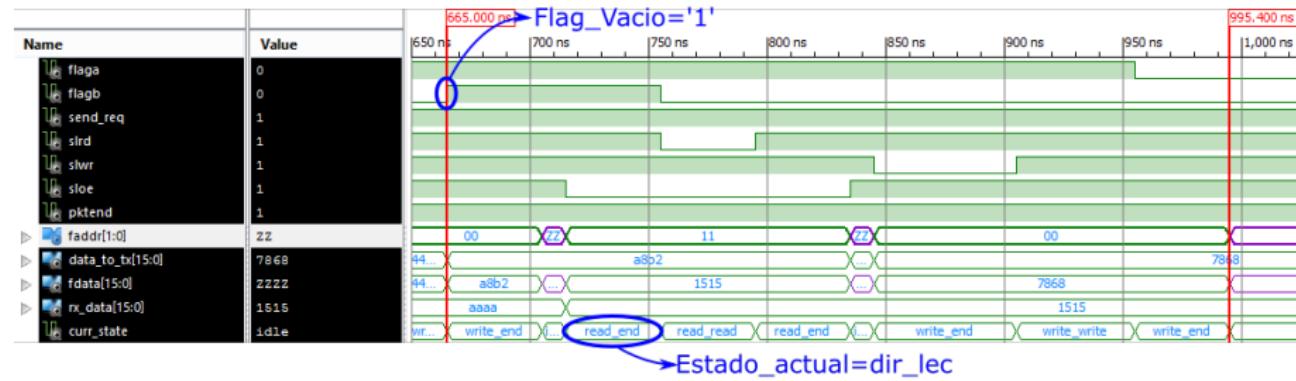
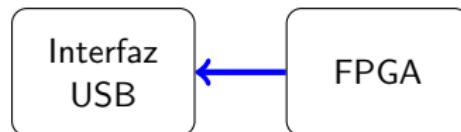
# Verificación Funcional

## Operación de escritura



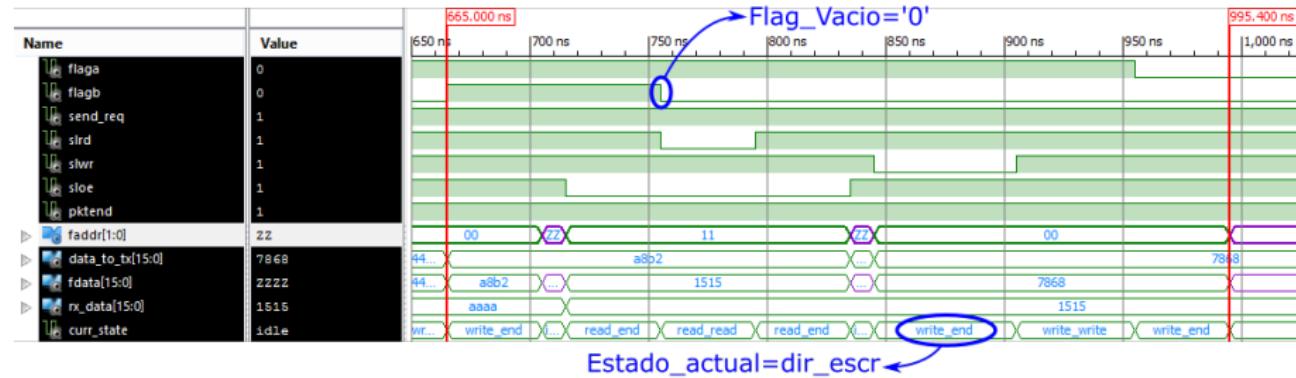
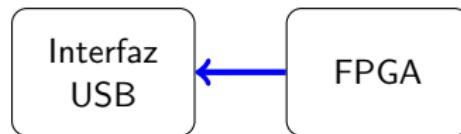
# Verificación Funcional

## Operación de escritura



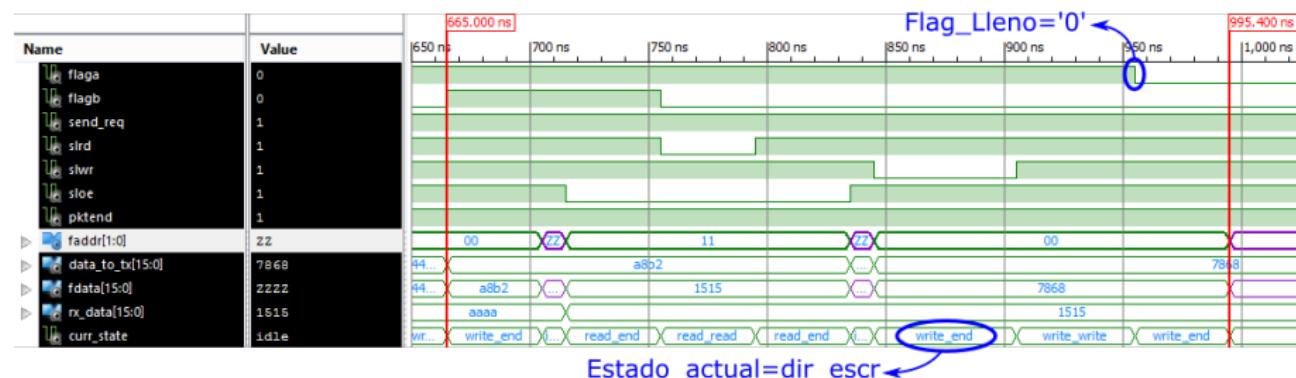
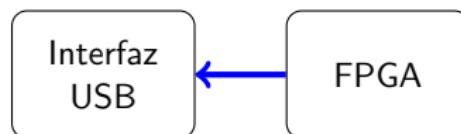
# Verificación Funcional

## Operación de escritura



# Verificación Funcional

## Operación de escritura

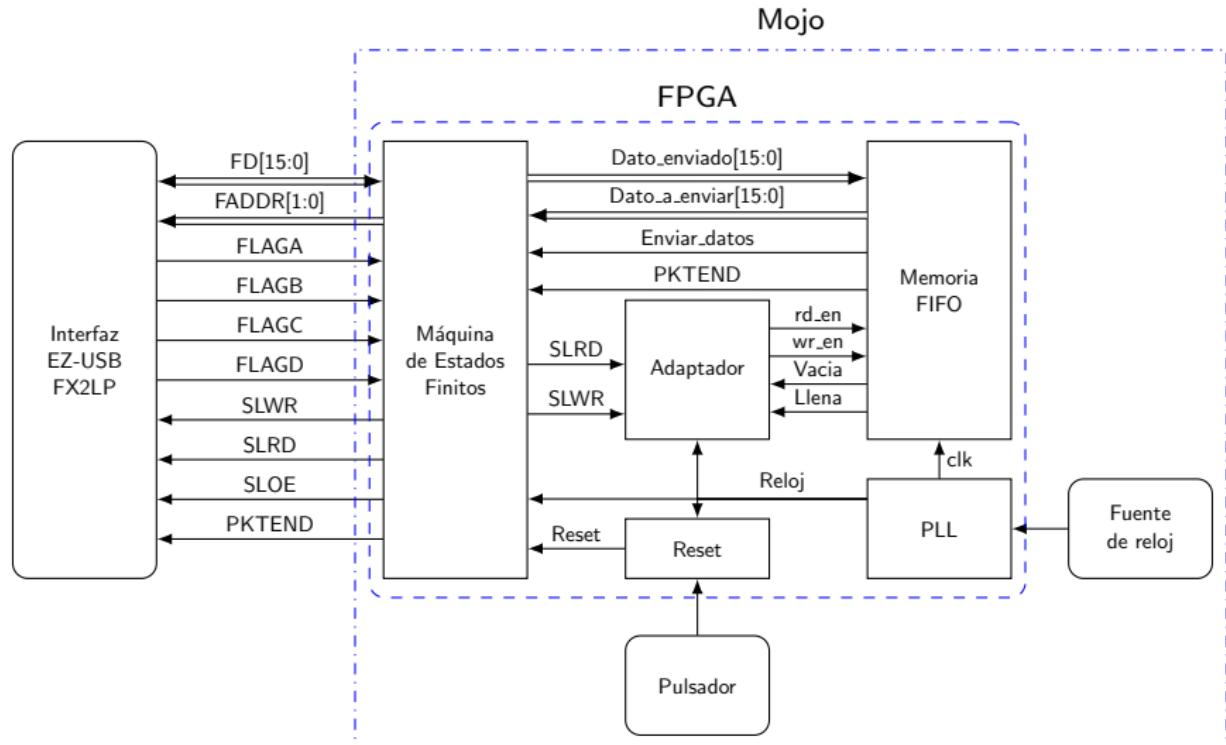


# Agenda

## 3 Evaluación y validación

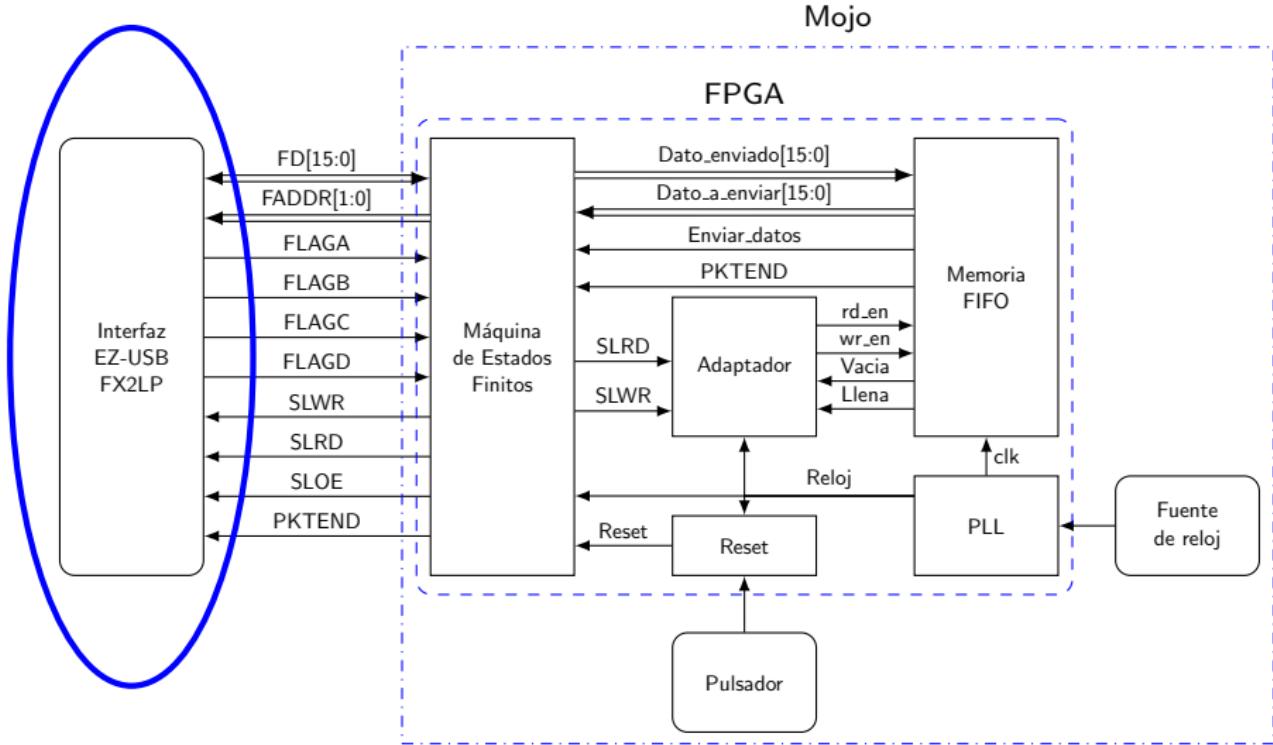
# Sistema de pruebas

## Arquitectura



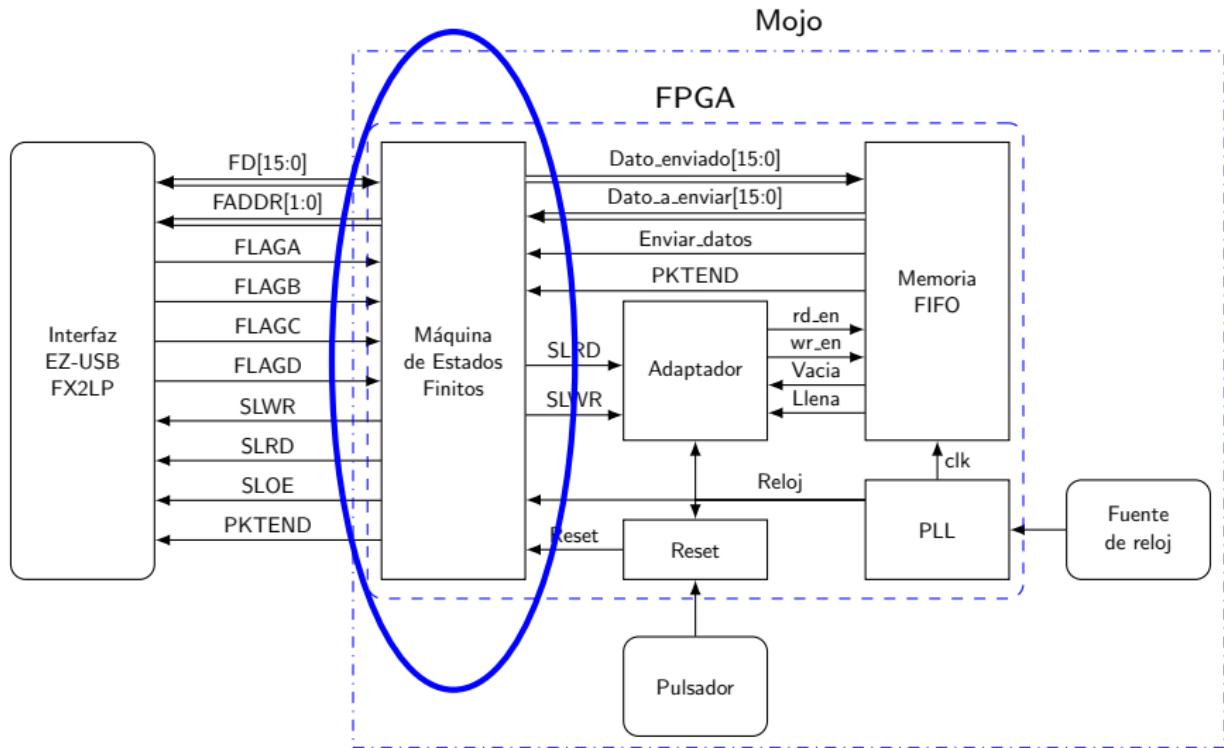
# Sistema de pruebas

## Arquitectura



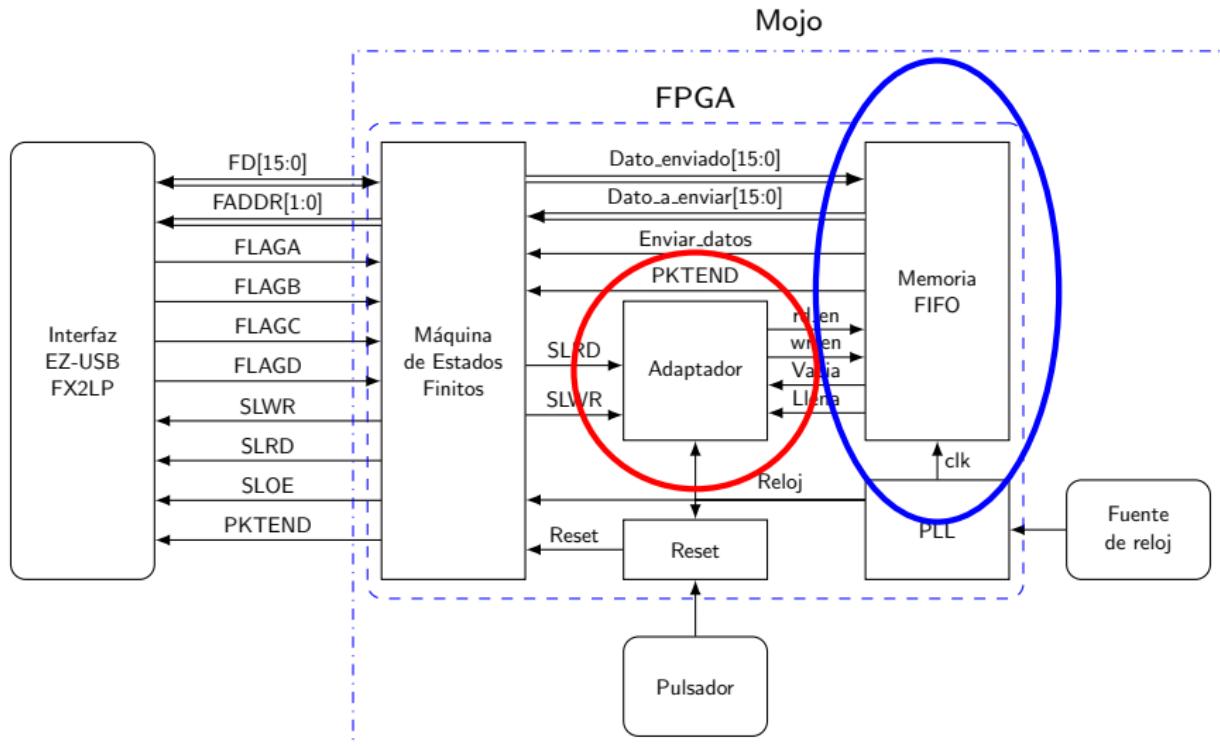
# Sistema de pruebas

## Arquitectura



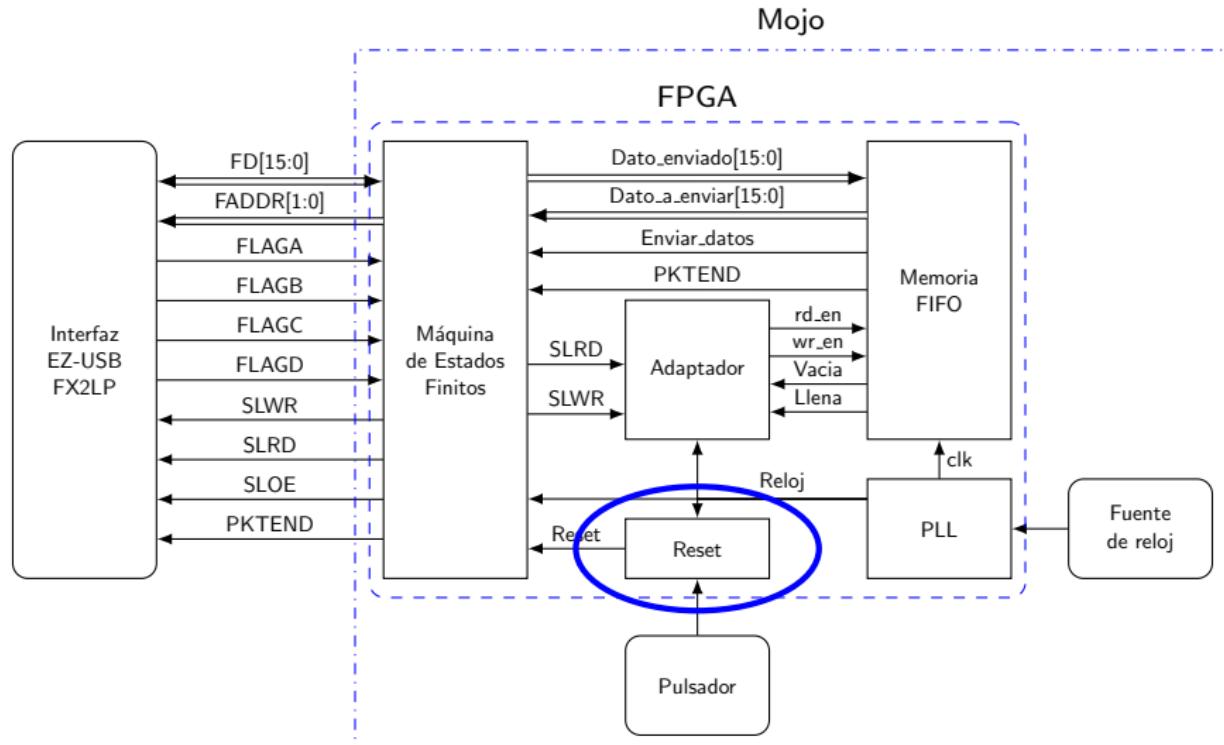
# Sistema de pruebas

## Arquitectura



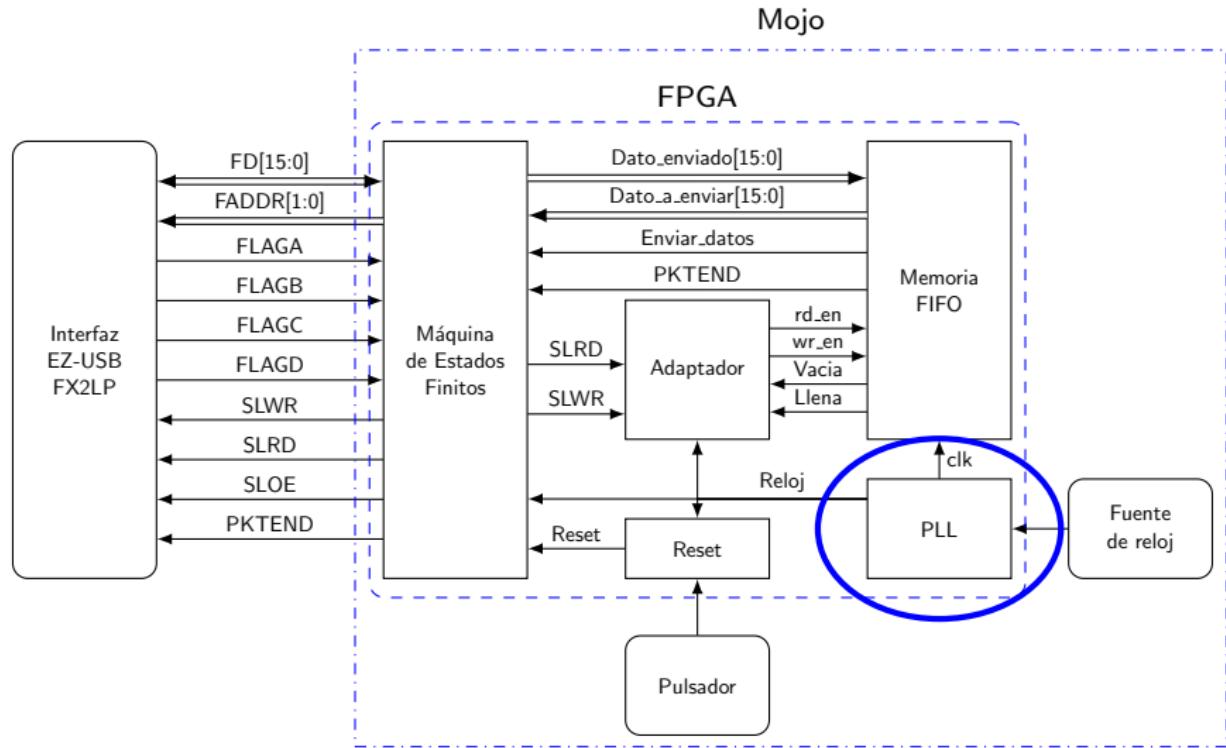
# Sistema de pruebas

## Arquitectura



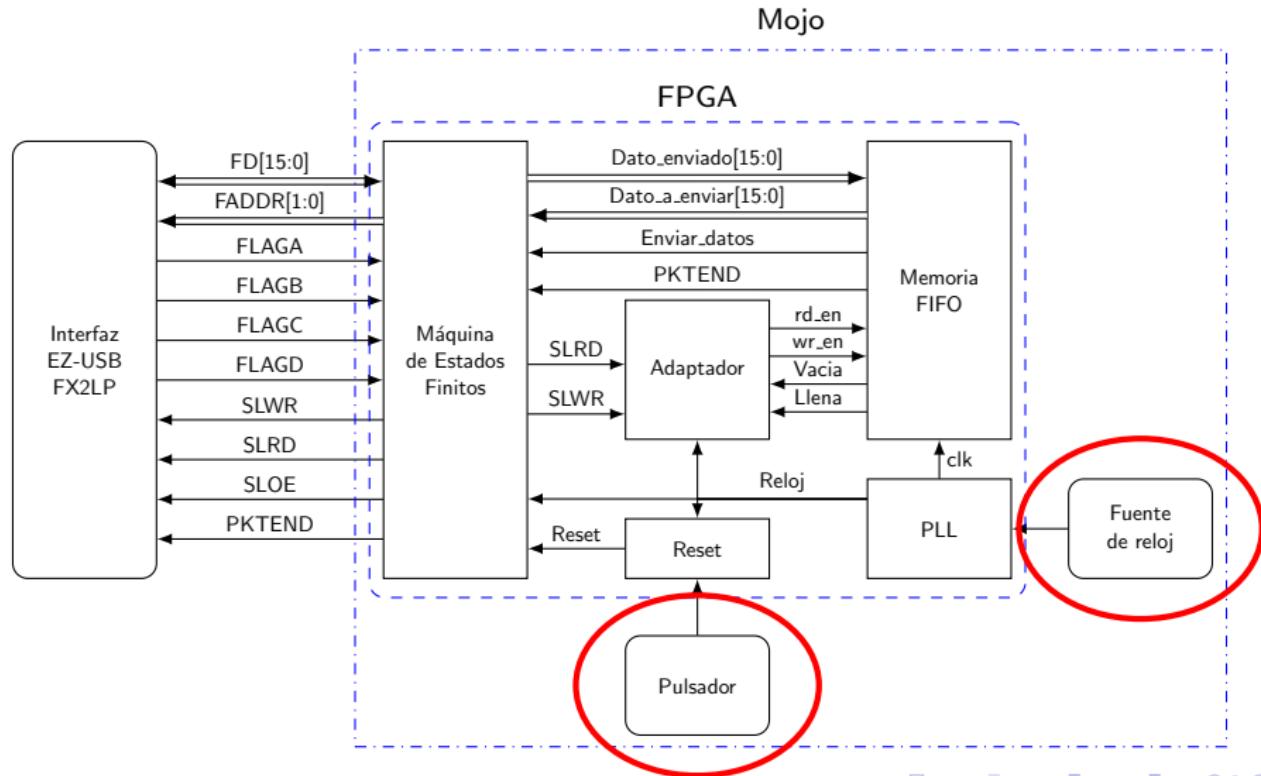
# Sistema de pruebas

## Arquitectura



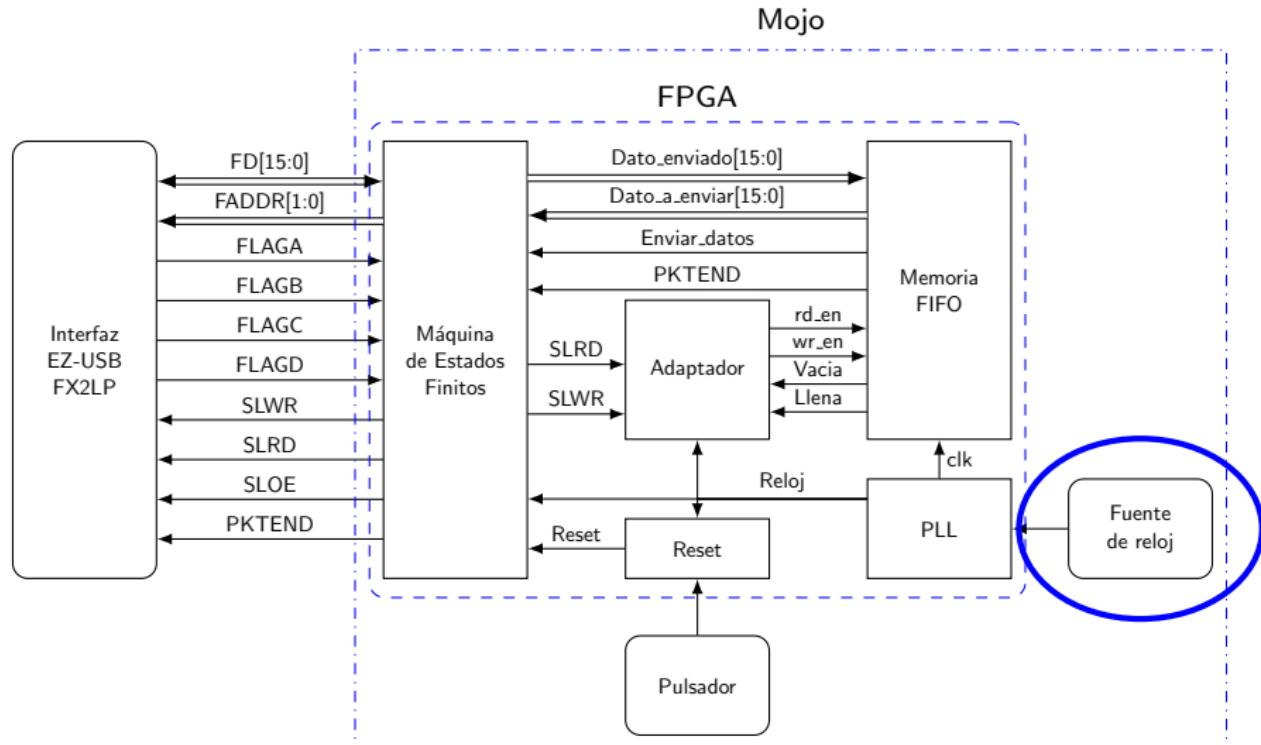
# Sistema de pruebas

## Arquitectura



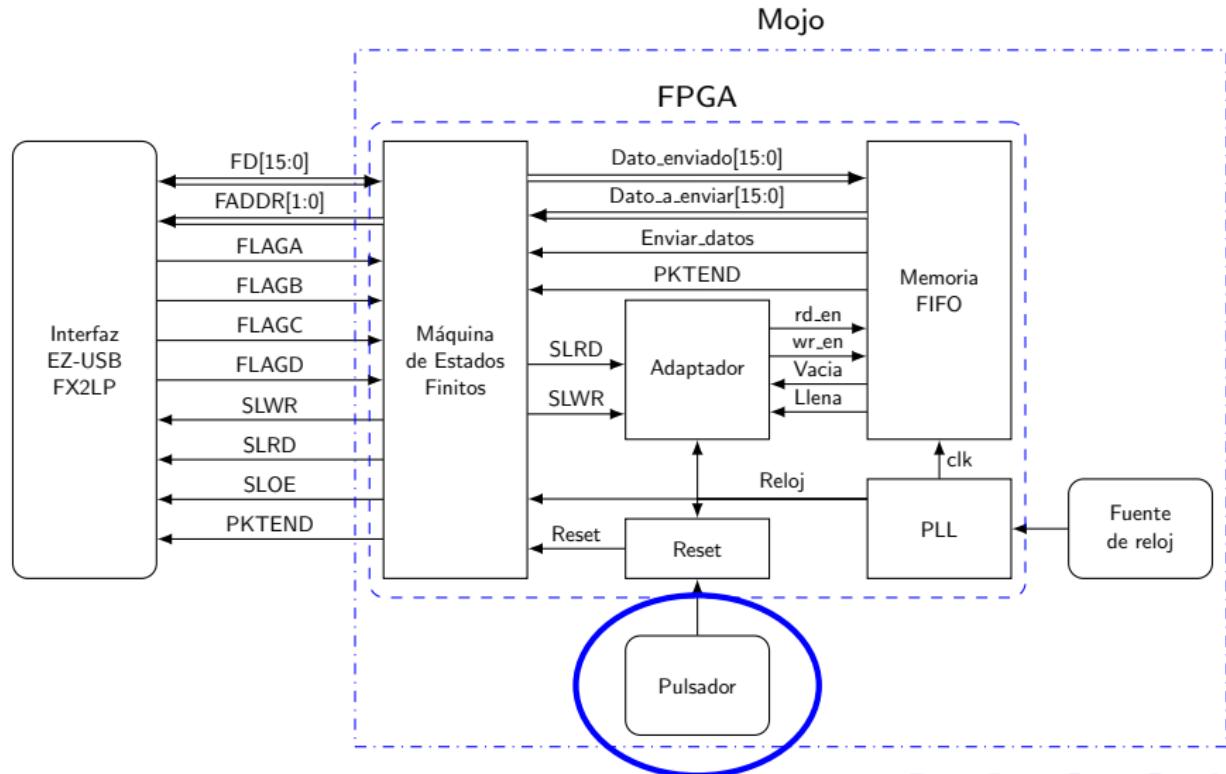
# Sistema de pruebas

## Arquitectura



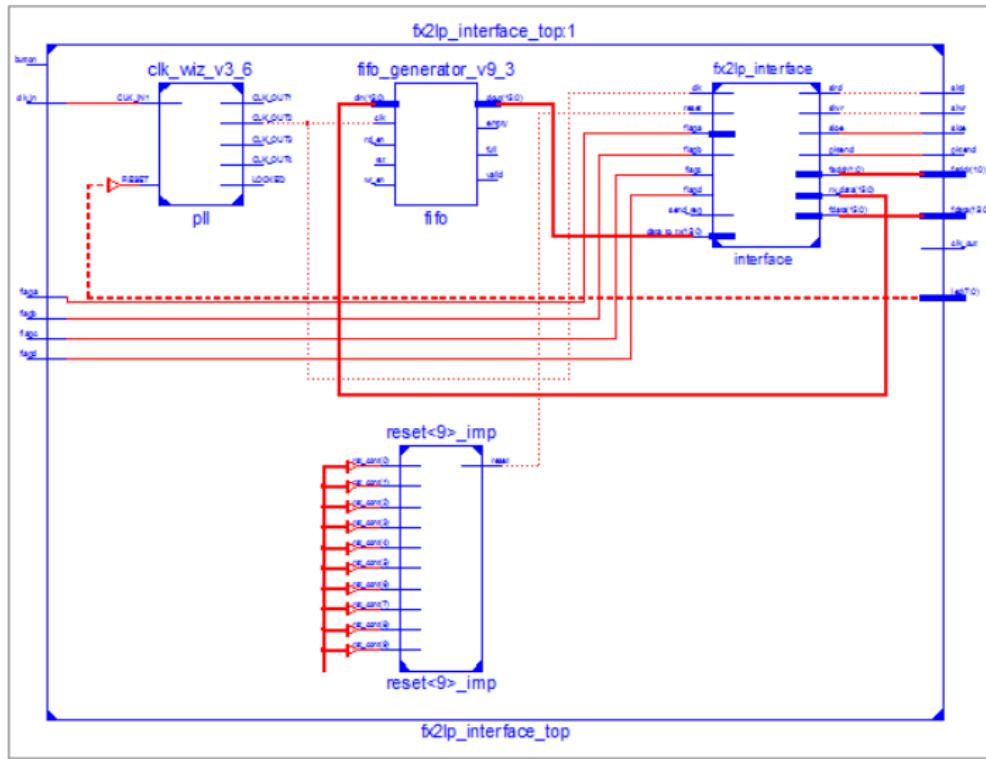
# Sistema de pruebas

## Arquitectura



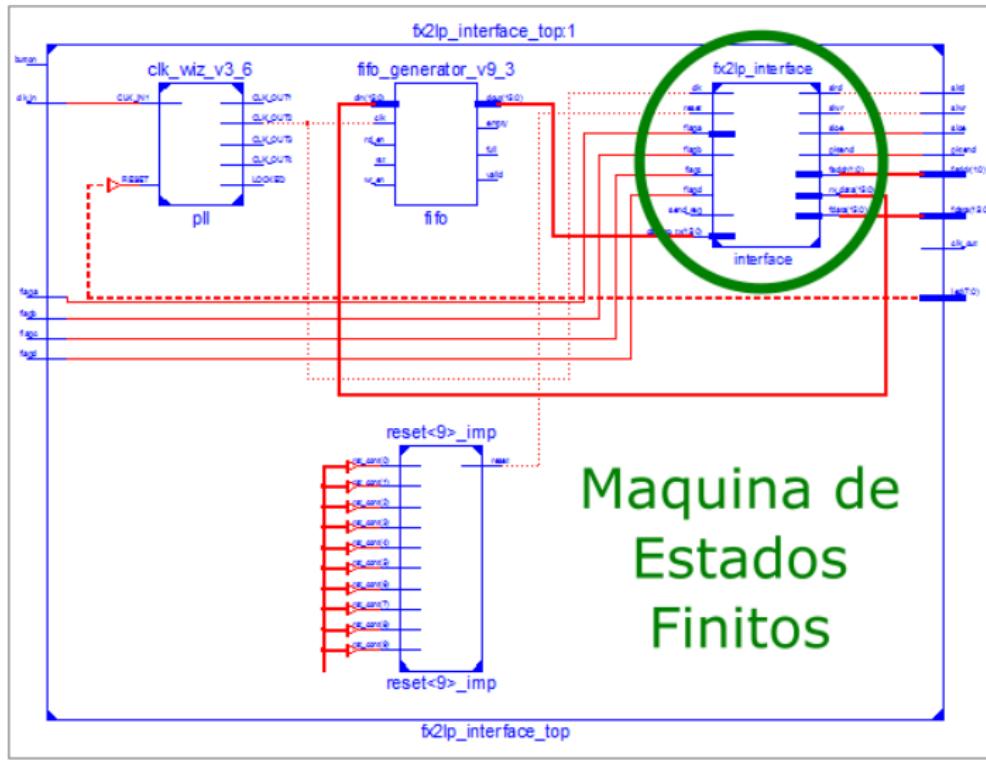
# Sistema de Pruebas

Implementación en VHDL



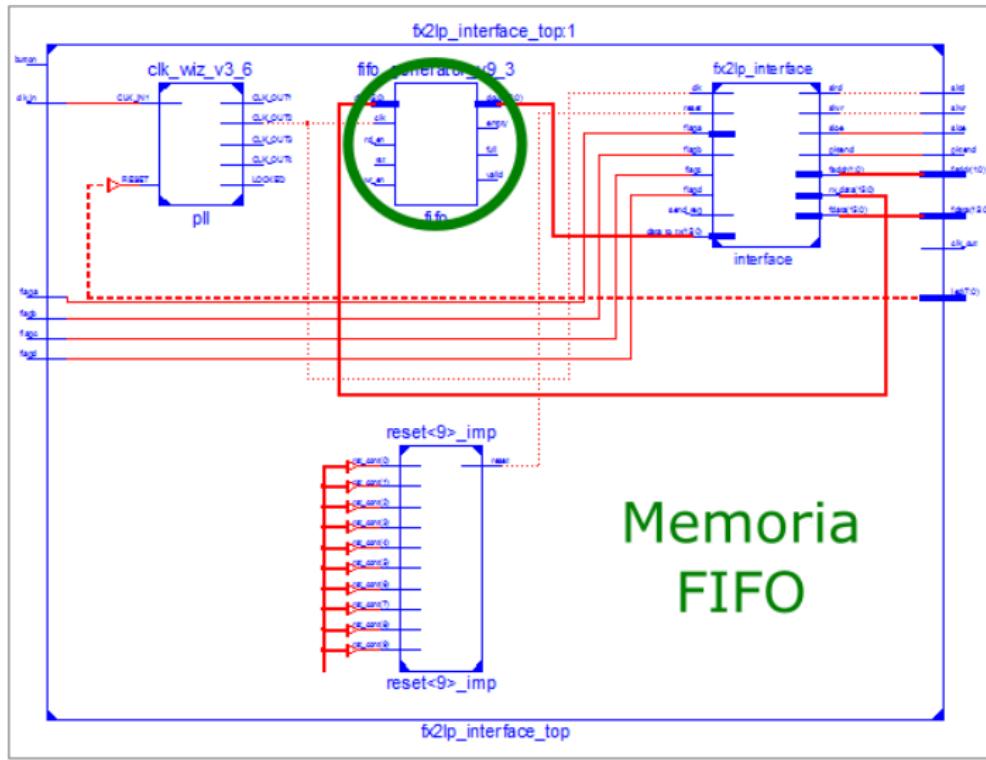
# Sistema de Pruebas

Implementación en VHDL



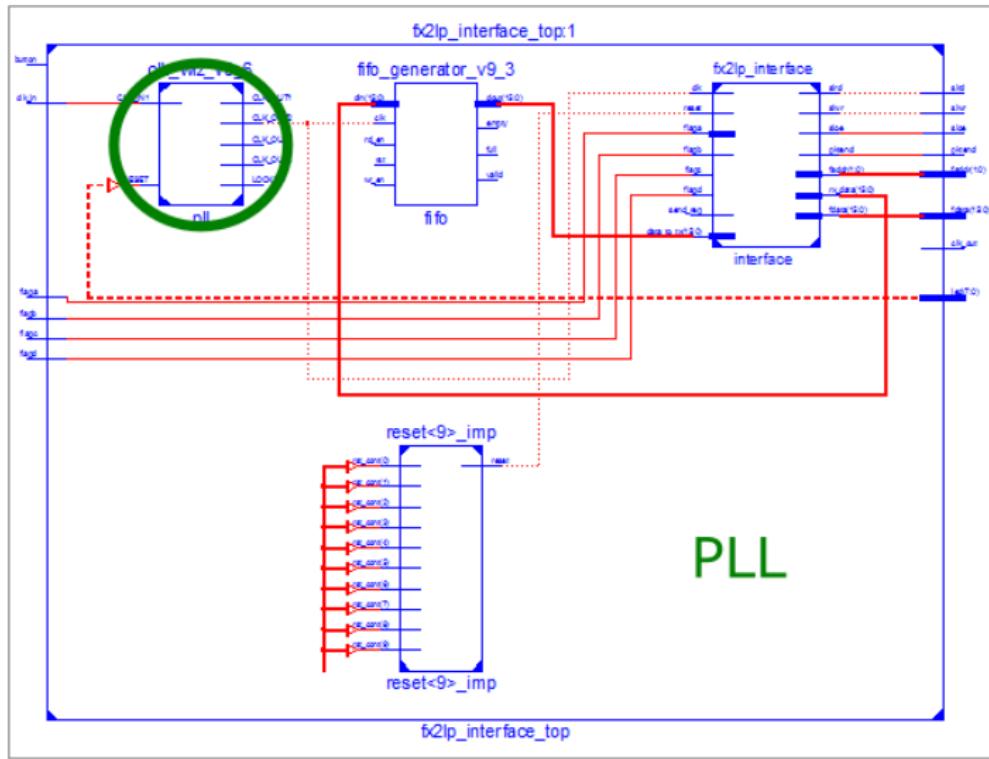
# Sistema de Pruebas

## Implementación en VHDL



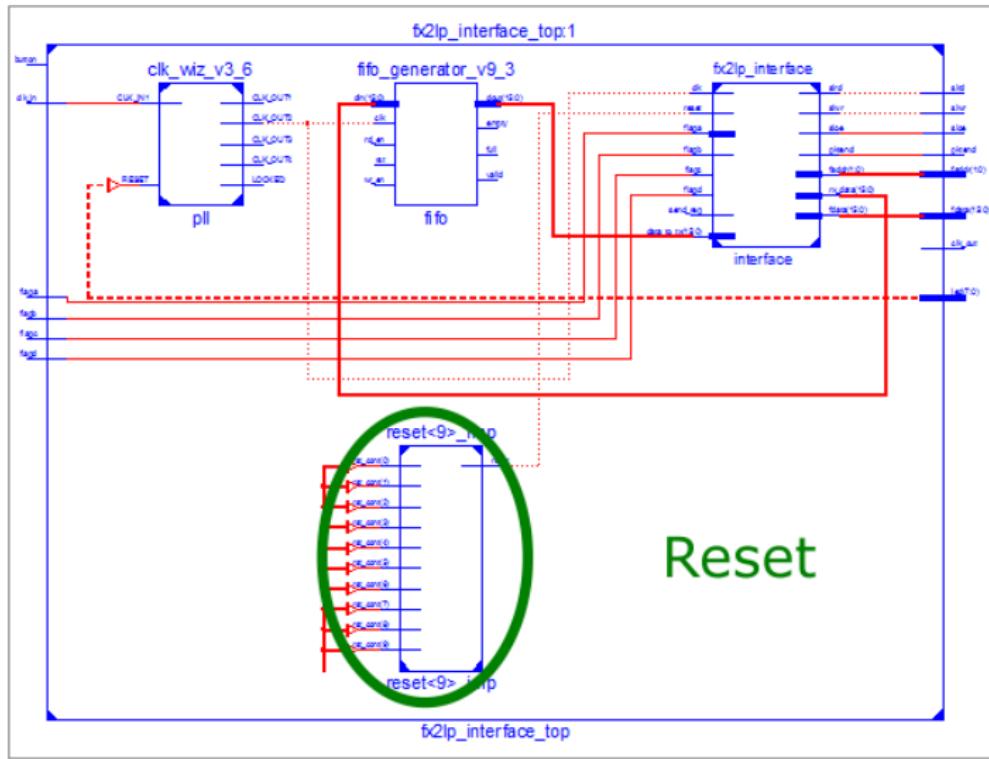
# Sistema de Pruebas

## Implementación en VHDL



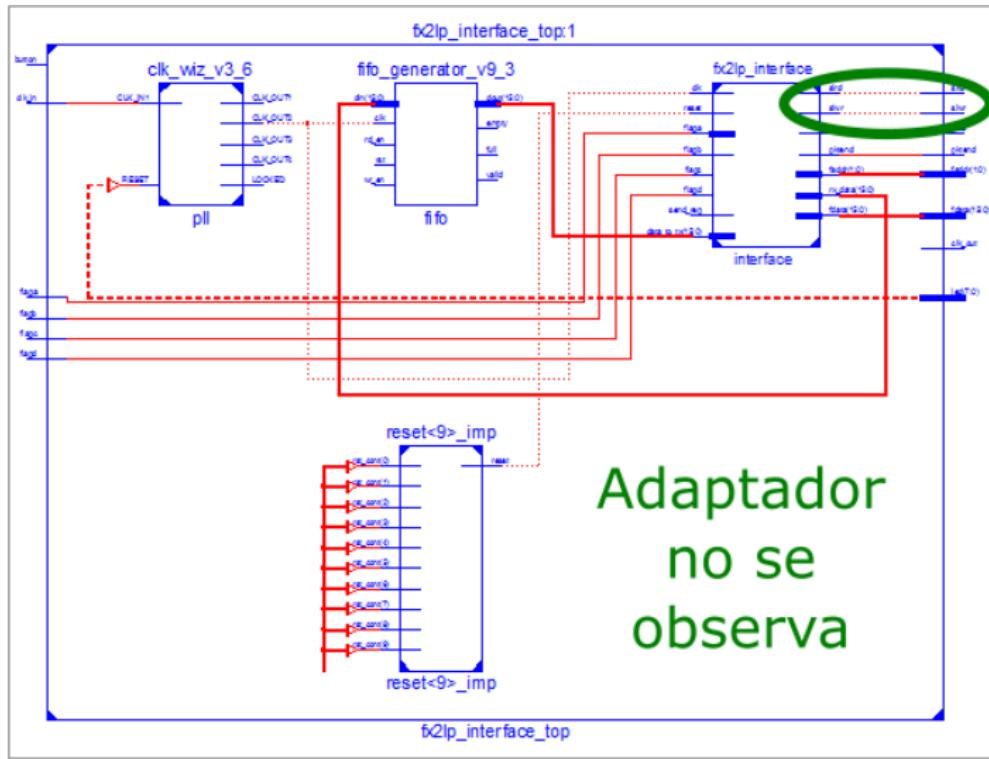
# Sistema de Pruebas

## Implementación en VHDL



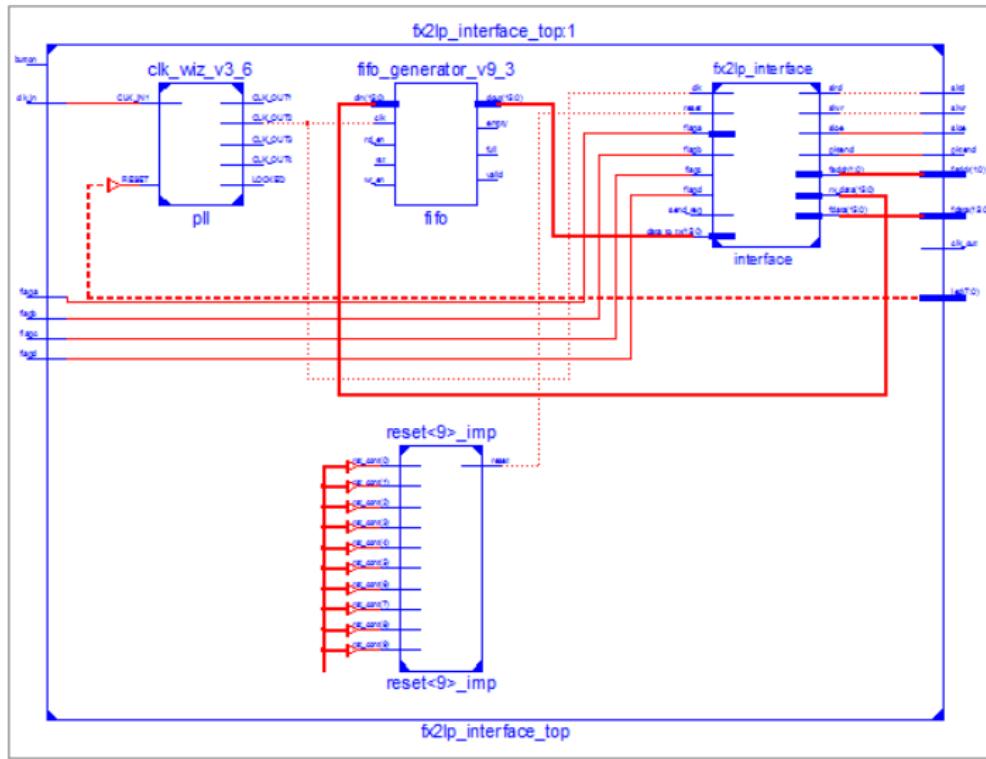
# Sistema de Pruebas

Implementación en VHDL



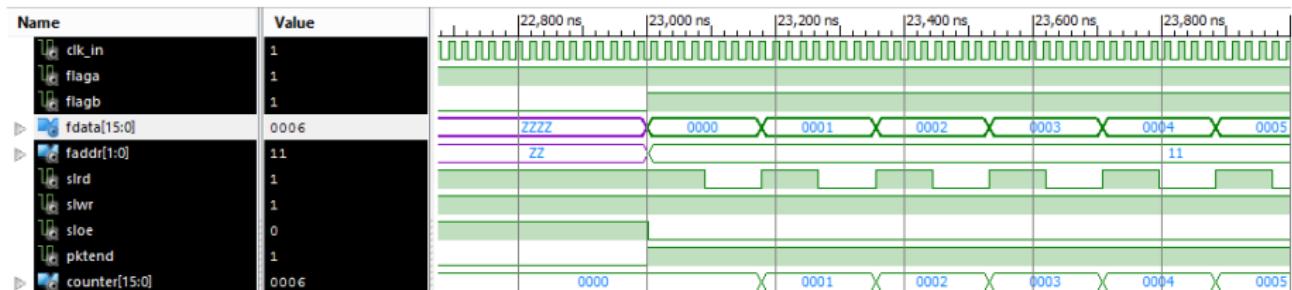
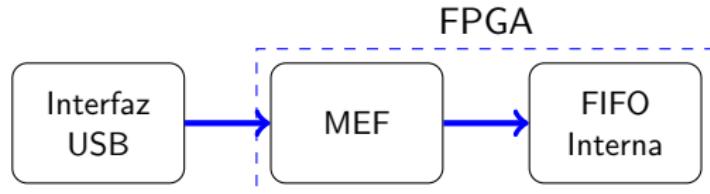
# Sistema de Pruebas

Implementación en VHDL



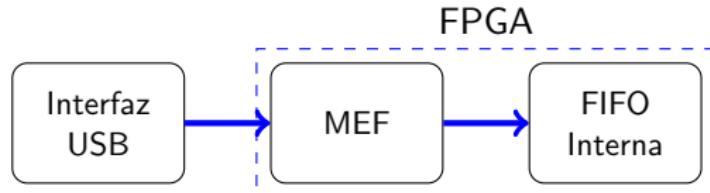
# Verificación Funcional

## Recepción de datos

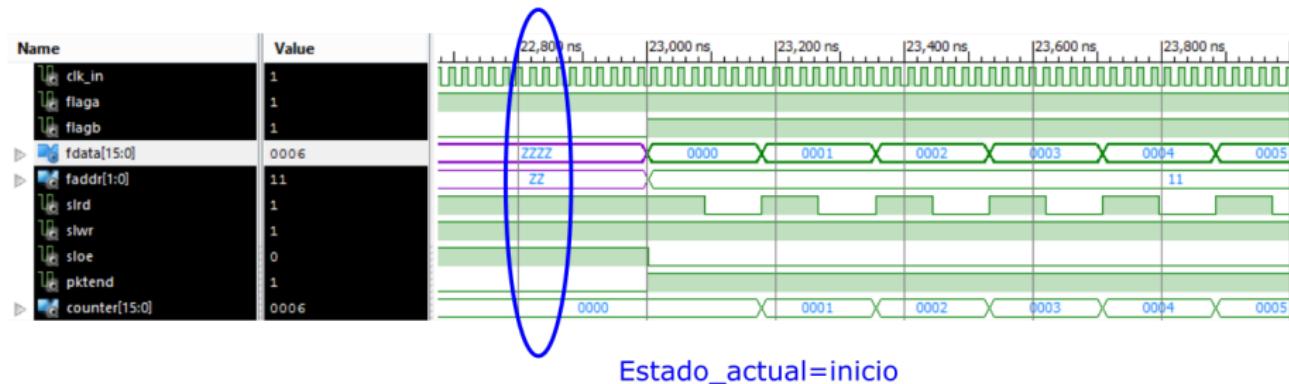


# Verificación Funcional

## Recepción de datos

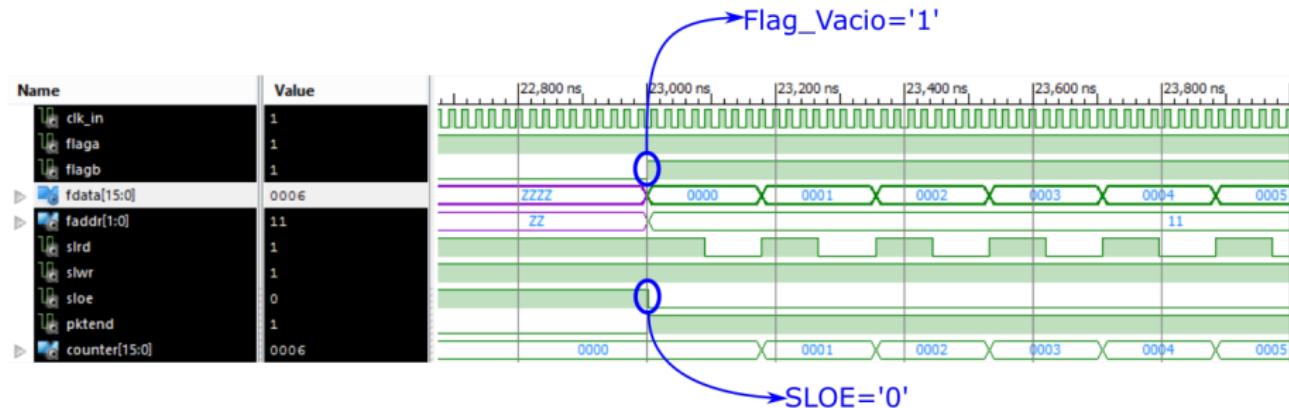
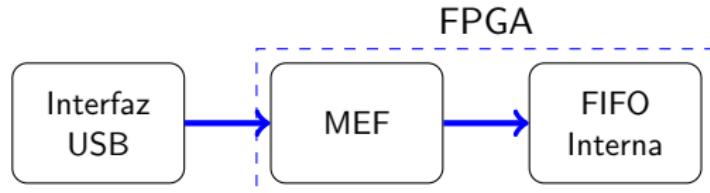


Sistema sin estímulos



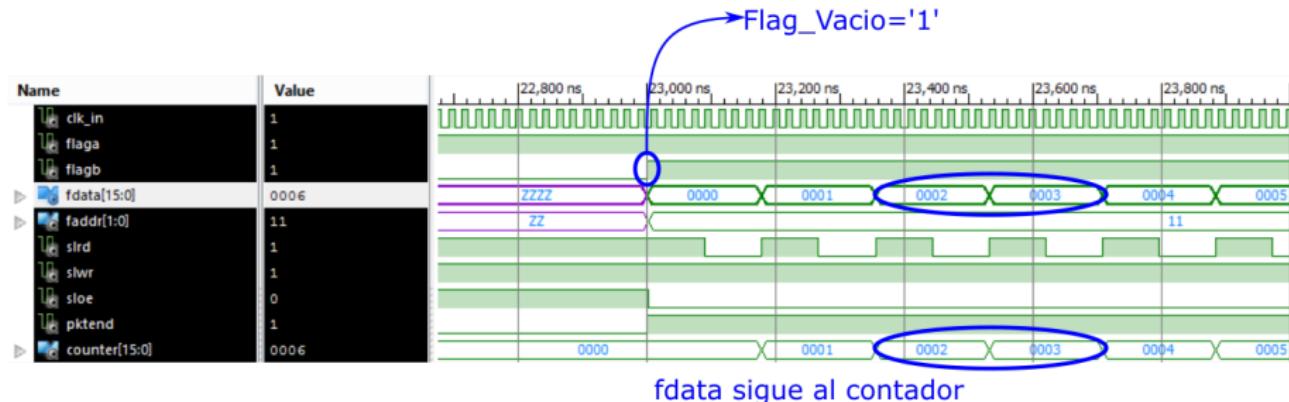
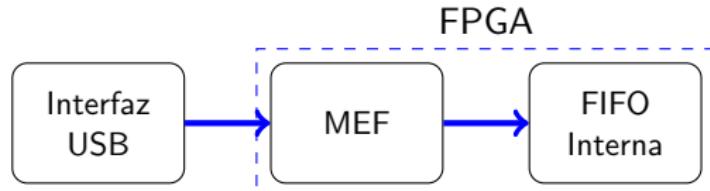
# Verificación Funcional

## Recepción de datos



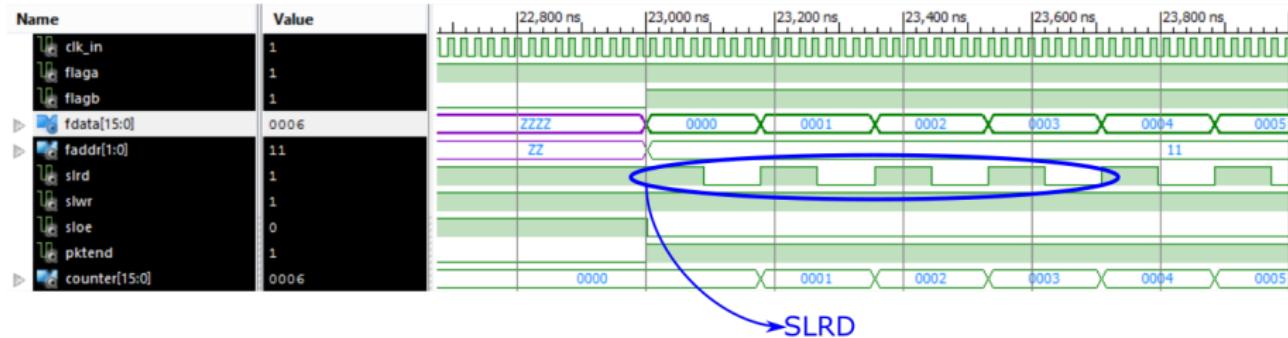
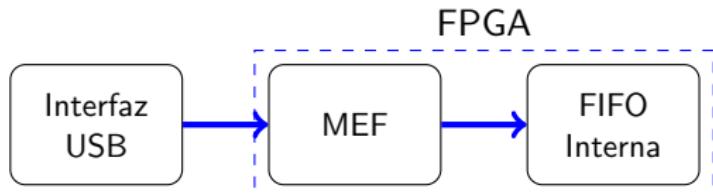
# Verificación Funcional

## Recepción de datos



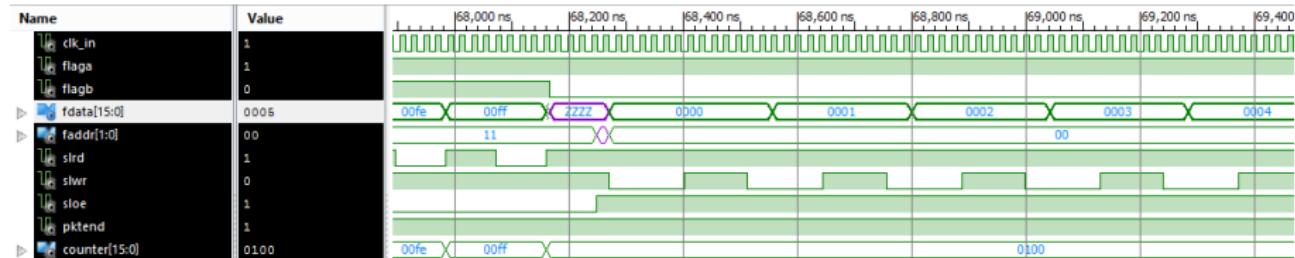
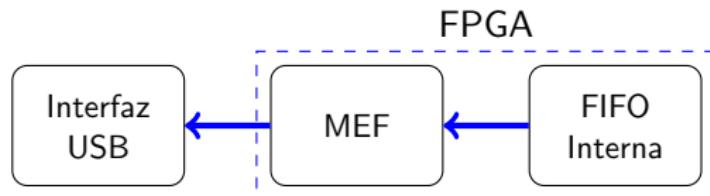
# Verificación Funcional

## Recepción de datos



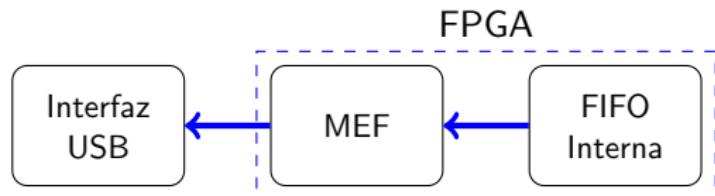
# Verificación Funcional

## Transmisión de datos

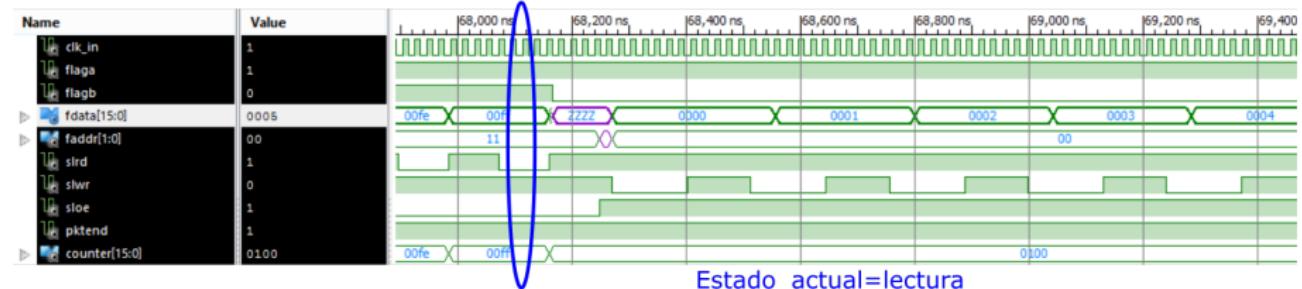


# Verificación Funcional

## Transmisión de datos

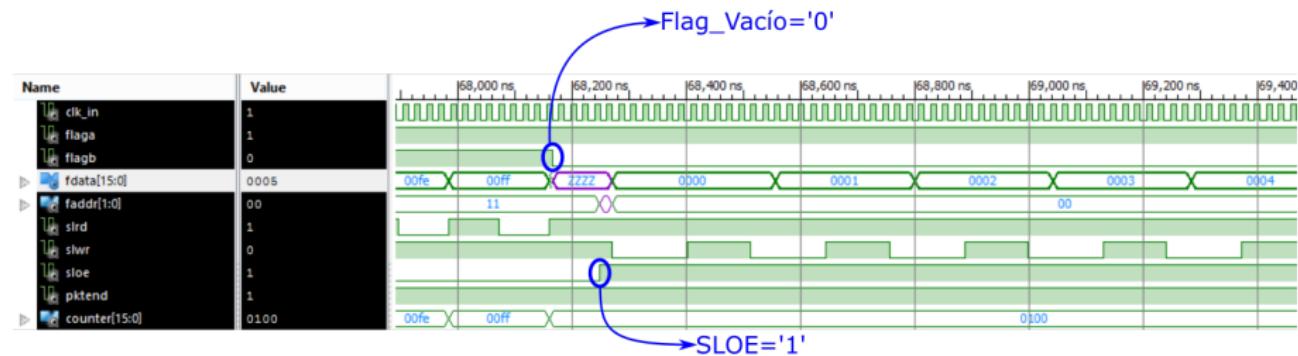
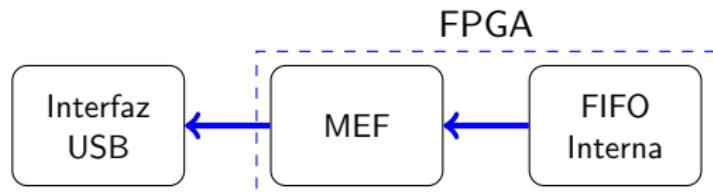


Sistema leyendo la Interfaz USB



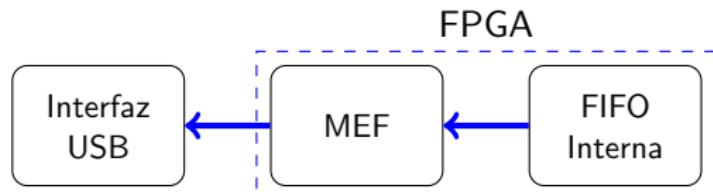
# Verificación Funcional

## Transmisión de datos

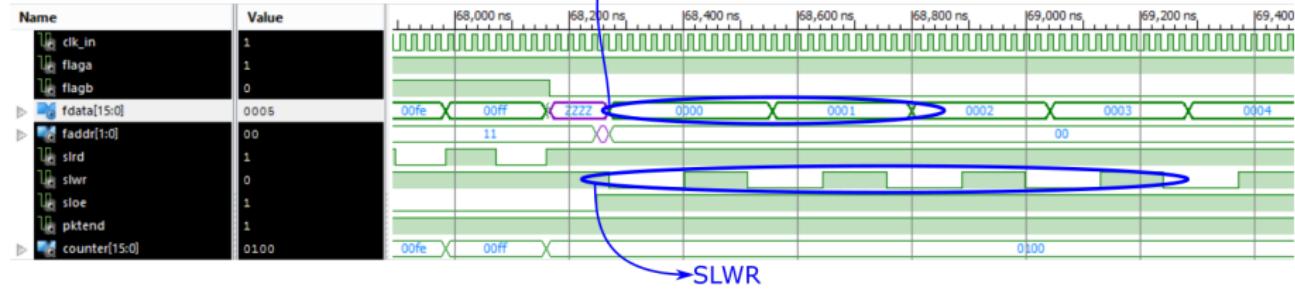


# Verificación Funcional

## Transmisión de datos

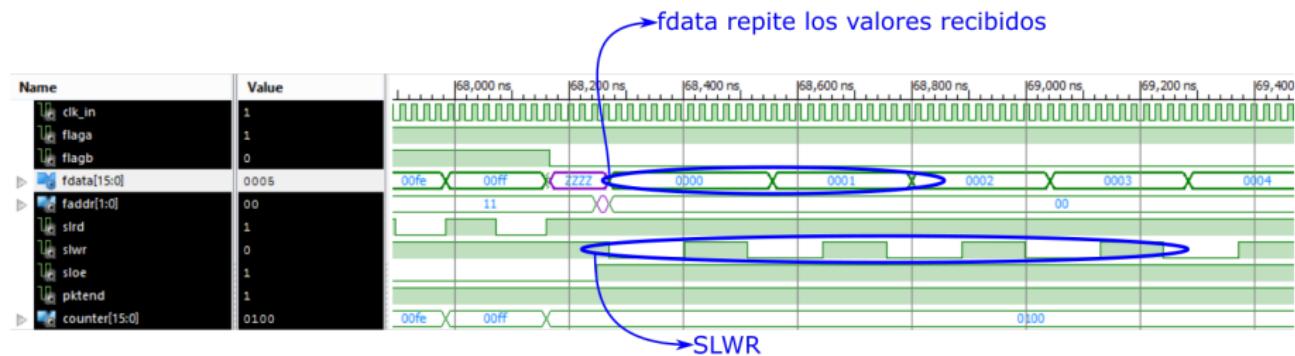
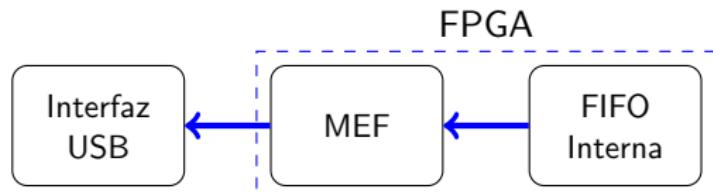


la memoria FIFO del FPGA  
tiene acceso al bus



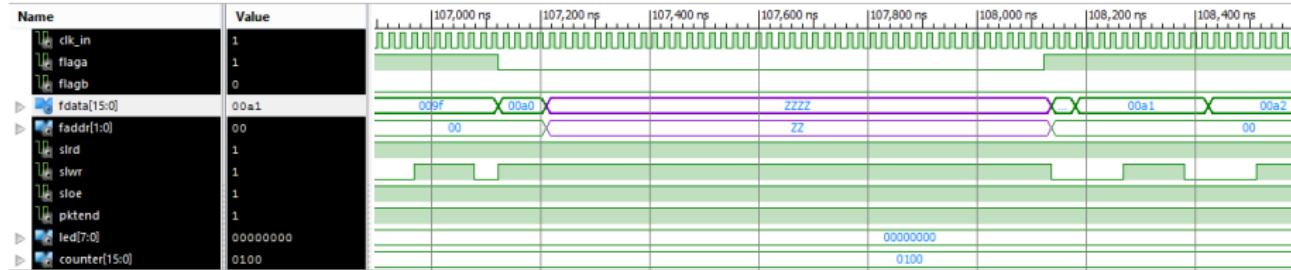
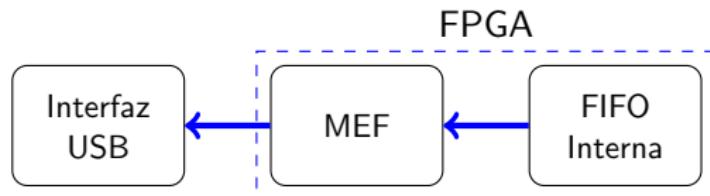
# Verificación Funcional

## Transmisión de datos



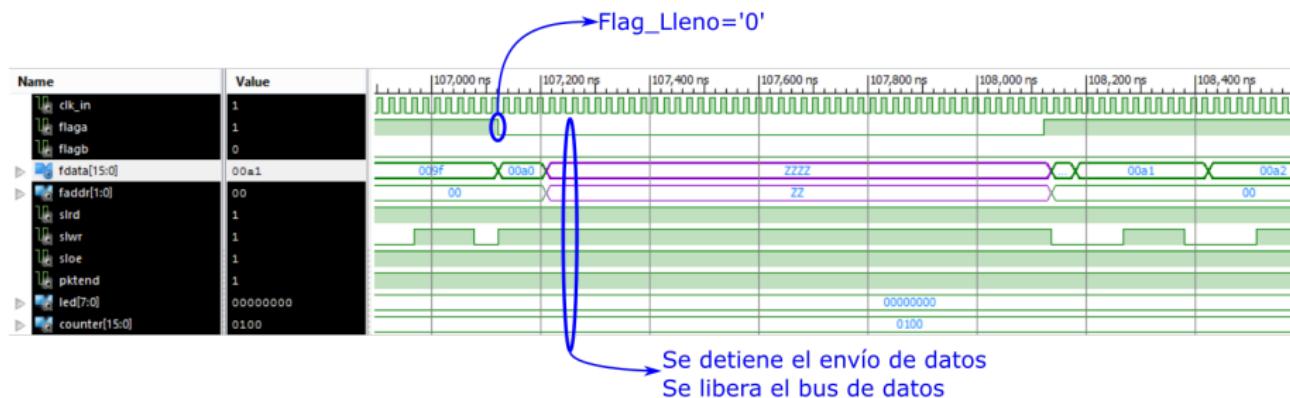
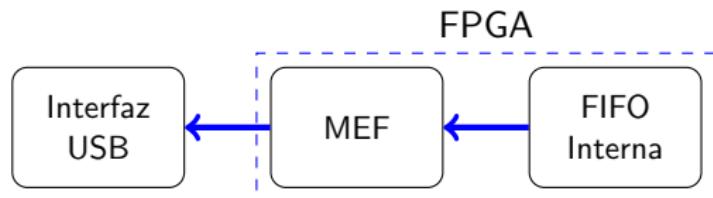
# Verificación Funcional

## Transmisión de datos



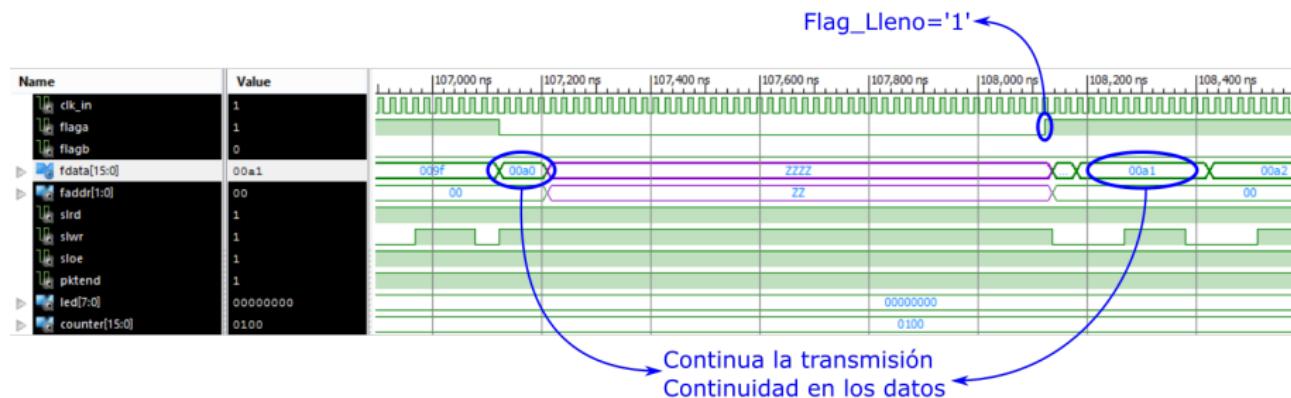
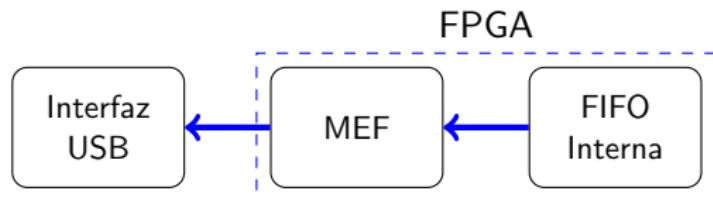
# Verificación Funcional

## Transmisión de datos



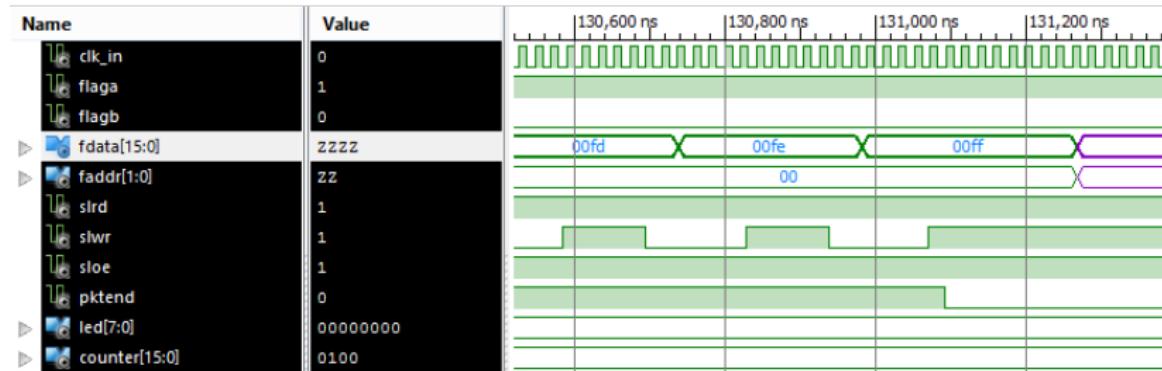
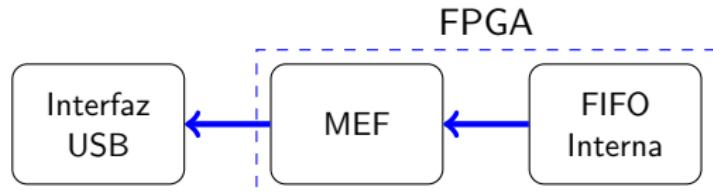
# Verificación Funcional

## Transmisión de datos



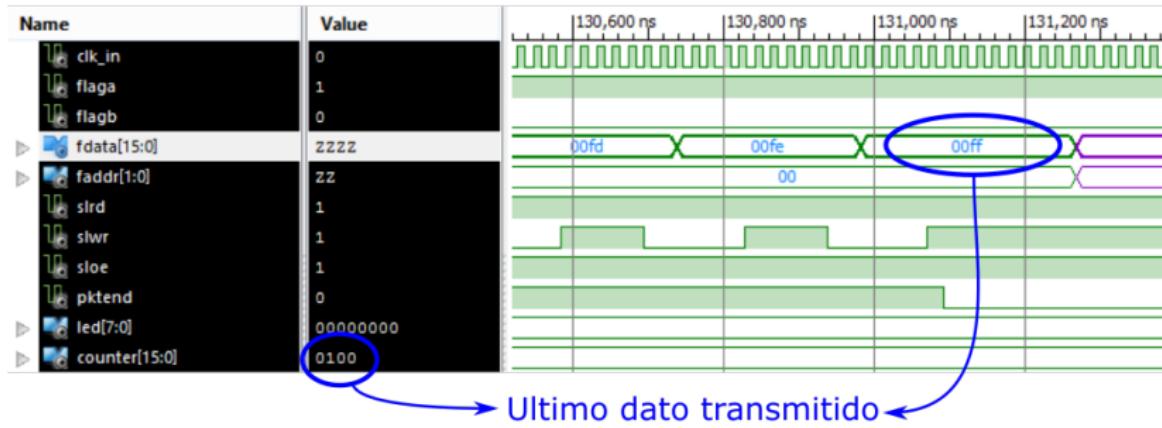
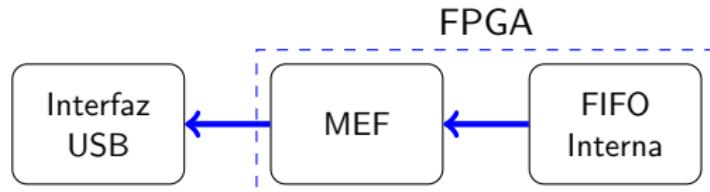
# Verificación Funcional

## Transmisión de datos



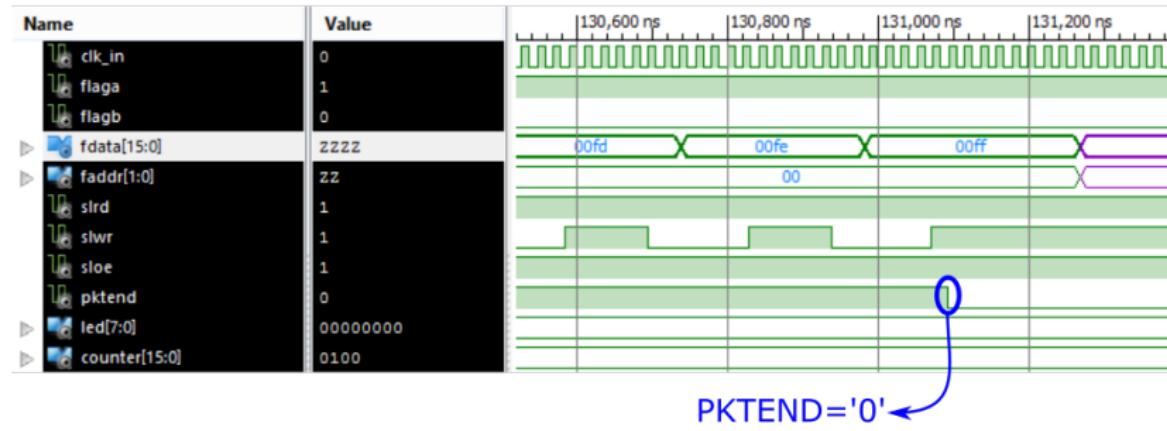
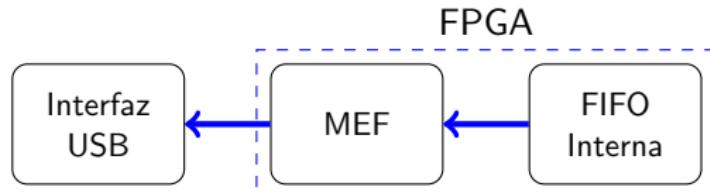
# Verificación Funcional

## Transmisión de datos



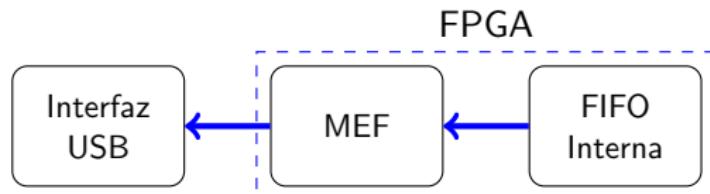
# Verificación Funcional

## Transmisión de datos

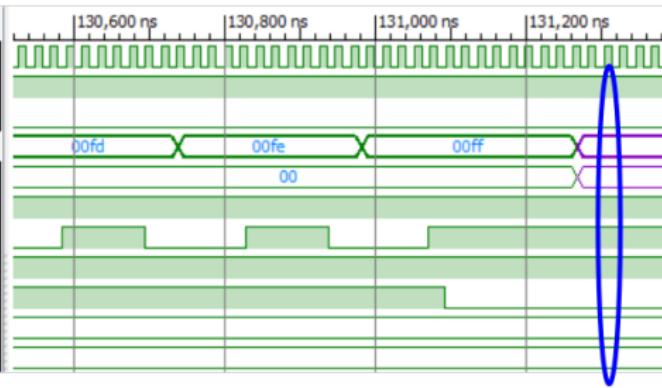


## Verificación Funcional

## Transmisión de datos

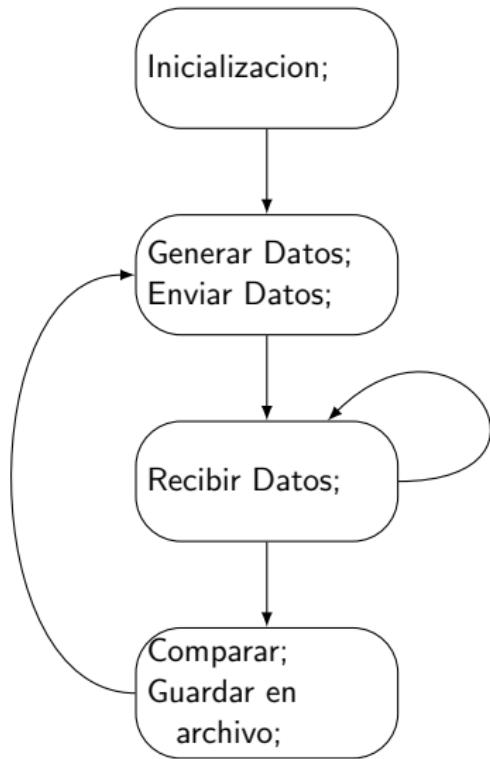


Name	Value
dk_in	0
flaga	1
flagb	0
fdata[15:0]	zzzz
faddr[1:0]	zz
slrd	1
slwr	1
sloe	1
pktend	0
led[7:0]	00000000
counter[15:0]	0100



Estado\_actual=inicio

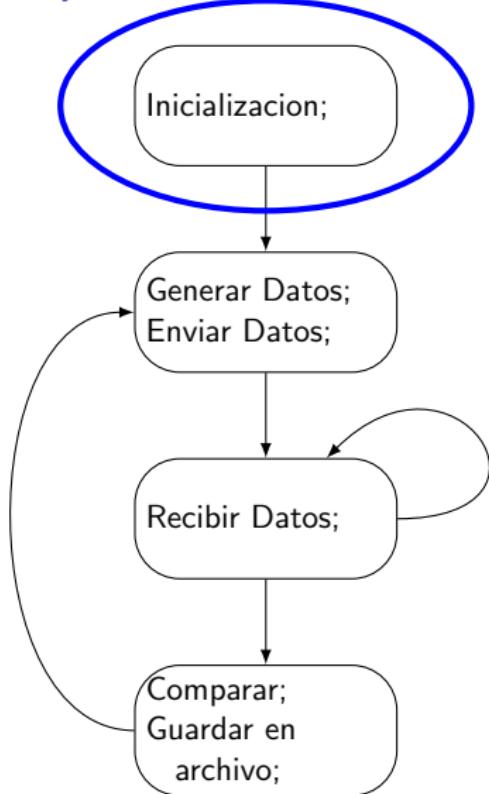
# Programa de PC para la transmisión de datos hacia y desde el FPGA



Se realizó un programa de computadoras, escrito en lenguaje C, para el cual se utilizó la biblioteca libusb-1.0, que permite obtener acceso al puerto USB como así también transmitir y recibir datos a través de él.

# Programa de PC para la transmisión de datos

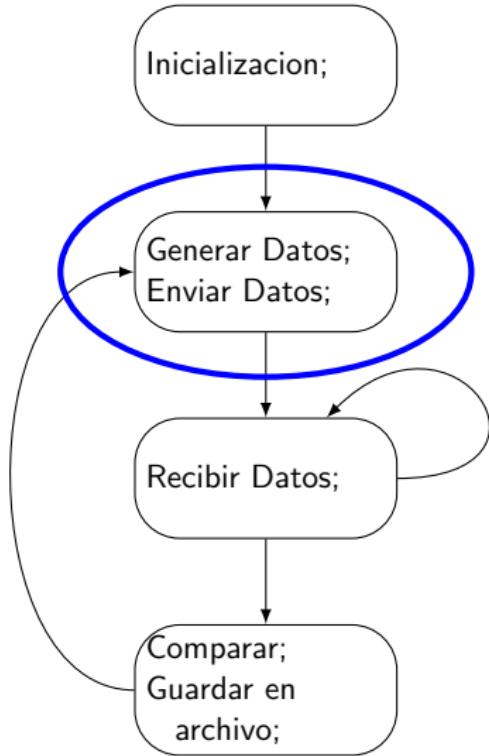
hacia y desde el FPGA



## Inicialización:

- Se listan los dispositivos
- Se busca y selecciona el adecuado
- Se configura la interfaz

# Programa de PC para la transmisión de datos hacia y desde el FPGA

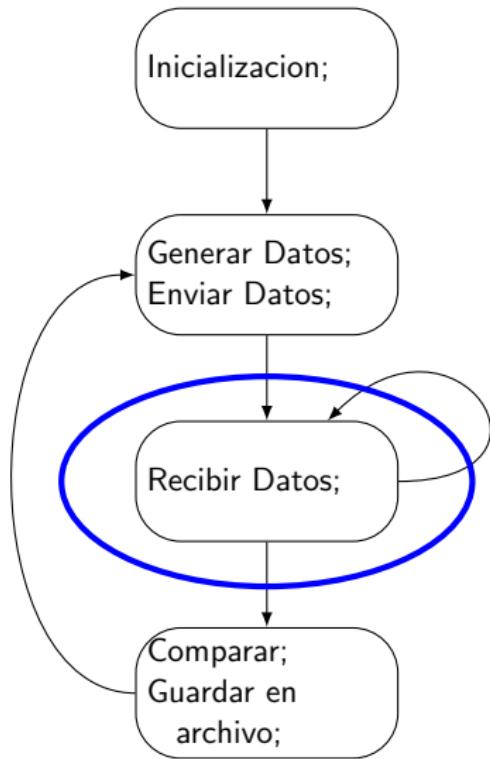


## Generar Datos:

	Paridad columna		
Paridad fila	1	0	1
Fila 1	0	0	1
Fila 2	1	1	0
Fila 3	0	1	0
Paridad fila	1	0	1

- Se generan datos aleatorios de 7 bits
- Se agrega un bit de paridad.
- Cada 7 bytes generados, se agrega 1 Byte con las paridades verticales.
- Se generan 16 tramas de 8 bytes, para alcanzar un paquete de 128 Bytes de datos.

# Programa de PC para la transmisión de datos hacia y desde el FPGA

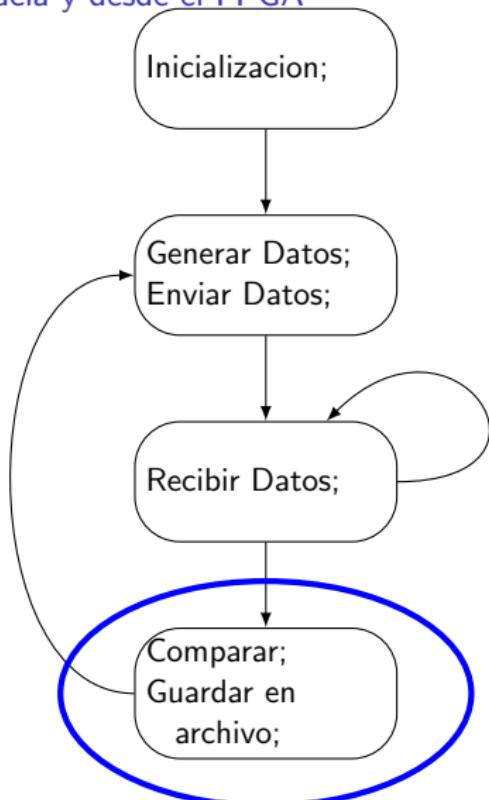


Recibir datos:

Se solicita la recepción de datos y se espera  
hasta que estos lleguen

# Programa de PC para la transmisión de datos

hacia y desde el FPGA

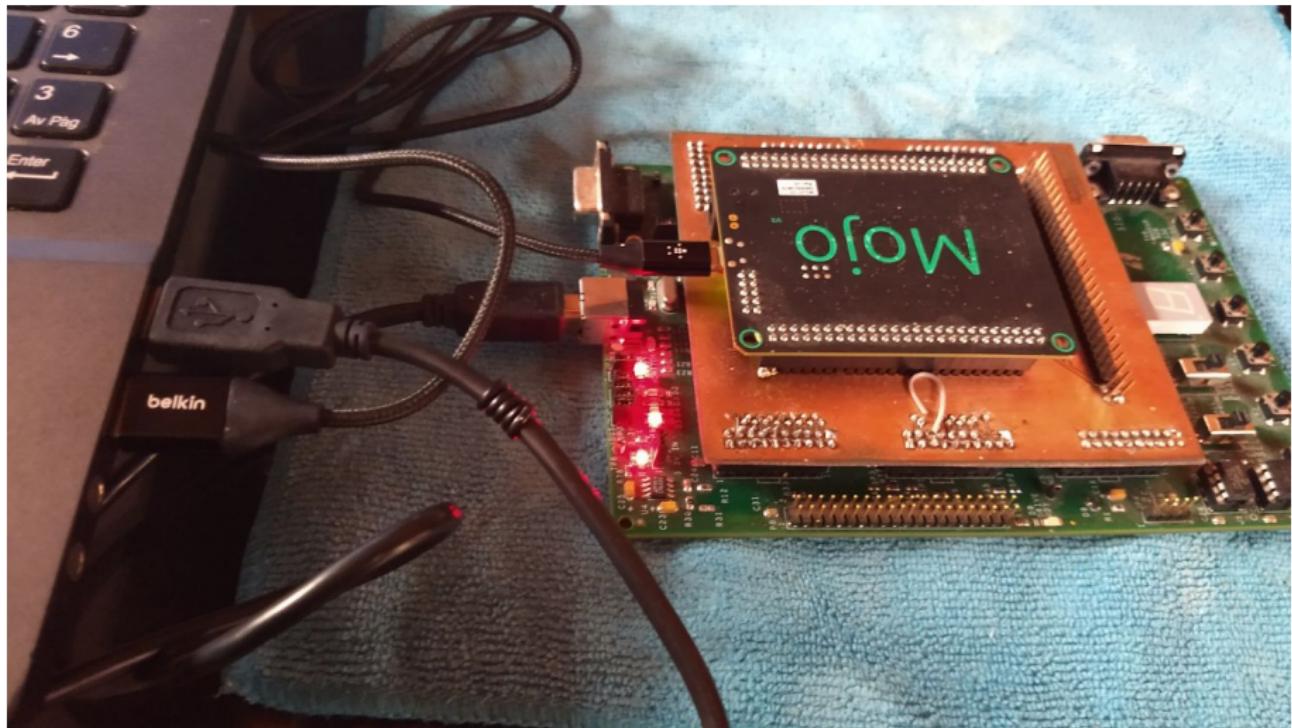


Comparar:

- Se chequea que los datos recibidos no tuvieron errores.
- Se comparan los datos recibidos con los enviados.

Guardar en archivo;

# Sistema en funcionamiento



# Resultados de la prueba de comunicación

- El sistema envió y recibió paquetes durante 24 horas, logrando la correcta transmisión y recepción de 388.191.289 paquetes de 128 bytes cada uno.
- El sistema desarrollado estableció una comunicación USB de alta velocidad.
- La tasa de transferencia de datos es de 12,4 Mbps.
- No hubo perdida ni errores en los datos transmitidos.
- La tasa efectiva de transmisión de datos útiles fue de 9,12 Mbps.

# Agenda

## 4 Resultados y Trabajos Futuros

# Conclusiones

- Se desarrollo un sistema de comunicación USB 2.0 que permite conectar una FPGA con una PC. El enlace pudo ser operado a 480 Mbps.
- El sistema elaborado permite leer y escribir datos en forma robusta, es decir, sin perder datos.
- Se adquirieron y entendieron conceptos importantes de la norma USB 2.0.
- Se seleccionó y se configuró el controlador FX2LP como interfaz USB.
- Se seleccionó el FPGA Spartan 6 de Xilinx Inc, incorporado en la placa de desarrollo Mojo v3.

# Conclusiones

- Se desarrolló una máquina de estados finitos implementada con el FPGA que comanda la comunicación entre este dispositivo y la interfaz USB.
- El desarrollo de la máquina de estados ocupa el 1% de las Slices del FPGA, dejando mucho recurso para cualquier otro sistema que se implemente.
- Se desarrolló un programa de computadoras que permite enviar y recibir datos a través del puerto USB.
- Se logró transmitir información desde y hacia la PC a una tasa efectiva de 9,12 Mbps.

# Trabajos futuros

- Realizar una prueba de máxima transferencia de datos a través del envío ininterrumpido de datos desde la FPGA hacia la PC.
- Implementar una comunicación síncrona entre ambos sistemas.
- Estudiar la migración del sistema desarrollado hacia USB 3.1

Muchas gracias.

# Consultas

Consultas.

# Material Adicional

## fx2lp\_interface\_top Project Status (04/23/2020 - 15:52:42)

<b>Project File:</b>	Spartan6.xise	<b>Parser Errors:</b>	No Errors
<b>Module Name:</b>	fx2lp_interface.top	<b>Implementation State:</b>	Placed and Routed
<b>Target Device:</b>	xc6slx9-2tqg144	<b>Errors:</b>	
<b>Product Version:</b>	ISE 14.7	<b>Warnings:</b>	
<b>Design Goal:</b>	Balanced	<b>Routing Results:</b>	All Signals Completely Routed
<b>Design Strategy:</b>	Xilinx Default (unlocked)	<b>Timing Constraints:</b>	All Constraints Met
<b>Environment:</b>	System Settings	<b>Final Timing Score:</b>	0 (Timing Report)

### Device Utilization Summary

Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	83	11,440	1 %	
Number used as Flip Flops	79			
Number used as Latches	4			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	80	5,720	1 %	

Continúa en la siguiente página

## Device Utilization Summary (cont.)

Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number used as logic	79	5,720	1 %	
Number using O6 output only	35			
Number using O5 output only	1			
Number using O5 and O6	43			
Number used as ROM	0			
Number used as Memory	0	1,440	0 %	
Number used exclusively as route-thrus	1			
Number with same-slice register load	1			
Number with same-slice carry load	0			
Number with other load	0			
Number of occupied Slices	39	1,430	2 %	
Number of MUXCYs used	44	2,860	1 %	
Number of LUT Flip Flop pairs used	106			
Number with an unused Flip Flop	35	106	33 %	
Number with an unused LUT	26	106	24 %	
Number of fully used LUT-FF pairs	45	106	42 %	
Continúa en la siguiente página				

## Device Utilization Summary (cont.)

Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of unique control sets	14			
Number of slice register sites lost to control set restrictions	77	11,440	1 %	
Number of bonded IOBs	37	102	36 %	
Number of LOCed IOBs	36	37	97 %	
IOB Flip Flops	1			
IOB Latches	16			
Number of RAMB16BWERS	1	32	3 %	
Number of RAMB8BWERS	0	64	0 %	
Number of BUFI02/BUFI02_2CLKs	1	32	3 %	
Number used as BUFI02s	1			
Number used as BUFI02_2CLKs	0			
Number of BUFI02FB/BUFI02FB_2CLKs	1	32	3 %	
Number used as BUFI02FBs	1			
Number used as BUFI02FB_2CLKs	0			
Number of BUFG/BUFGMUXs	3	16	18 %	
Number used as BUFGs	3			
Continúa en la siguiente página				

## Device Utilization Summary (cont.)

Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number used as BUFGMUX	0			
Number of DCM/DCM_CLKGENs	0	4	0 %	
Number of ILOGIC2/ISERDES2s	16	200	8 %	
Number used as ILOGIC2s	16			
Number used as ISERDES2s	0			
Number of IODELAY2/IODRP2/IODRP2_MCBs	0	200	0 %	
Number of OLOGIC2/OSERDES2s	1	200	1 %	
Number used as OLOGIC2s	1			
Number used as OSERDES2s	0			
Number of BSCANs	0	4	0 %	
Number of BUFHs	0	128	0 %	
Number of BUFPLLs	0	8	0 %	
Number of BUFPLL_MCBs	0	4	0 %	
Number of DSP48A1s	0	16	0 %	
Number of ICAPs	0	1	0 %	
Number of MCBs	0	2	0 %	
Continúa en la siguiente página				

## Device Utilization Summary (cont.)

Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of PCILOGICSEs	0	2	0 %	
Number of PLL_ADVs	1	2	50 %	
Number of PMVs	0	1	0 %	
Number of STARTUPs	0	1	0 %	
Number of SUSPEND_SYNCs	0	1	0 %	
Average Fanout of Non-Clock Nets	2.91			

## Detailed Reports

Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	jue 23. abr 15:51:44 2020	0	26 Warnings (0 new)	14 Infos (10 new)
Translation Report	Current	jue 23. abr 15:51:53 2020	0	0	2 Infos (0 new)
Map Report	Current	jue 23. abr 15:52:24 2020			
Place and Route Report	Current	jue 23. abr 15:52:34 2020	0	6 Warnings (0 new)	0
Power Report					
Post-PAR Static Timing Report	Current	jue 23. abr 15:52:40 2020	0	0	3 Infos (0 new)
Bitgen Report	Out of Date	mié 1. abr 14:38:01 2020	0	4 Warnings (0 new)	1 Info (0 new)

Date Generated: 05/31/2020 - 15:05:40

