

# Construcción de soluciones utilizando Vivado y HLS

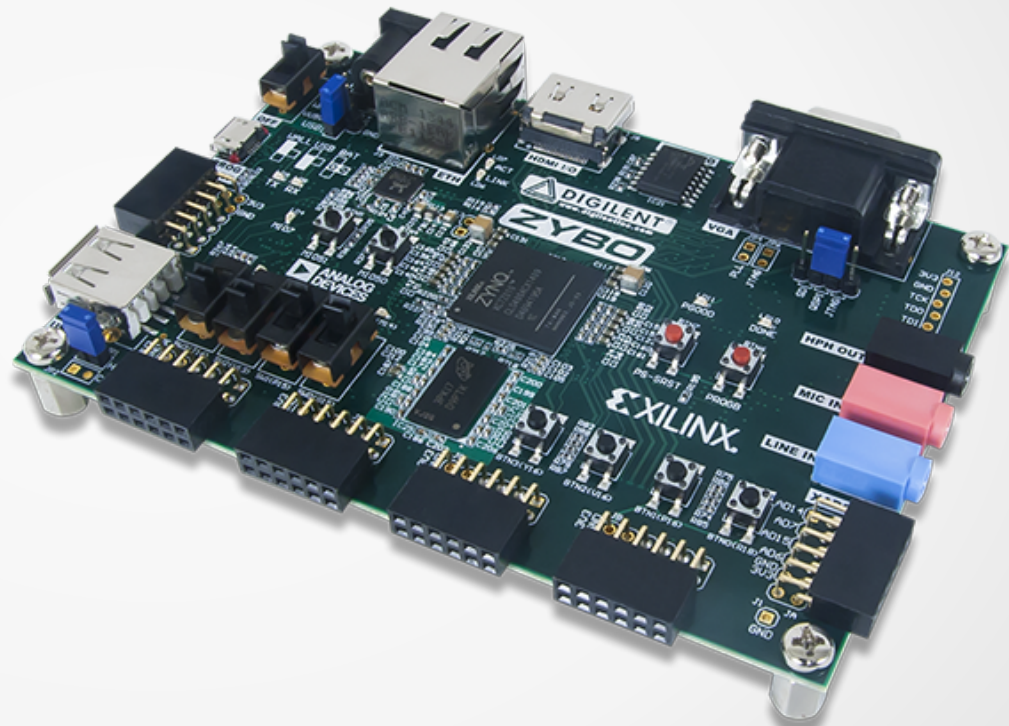


**Patricia Miriam Borensztein**

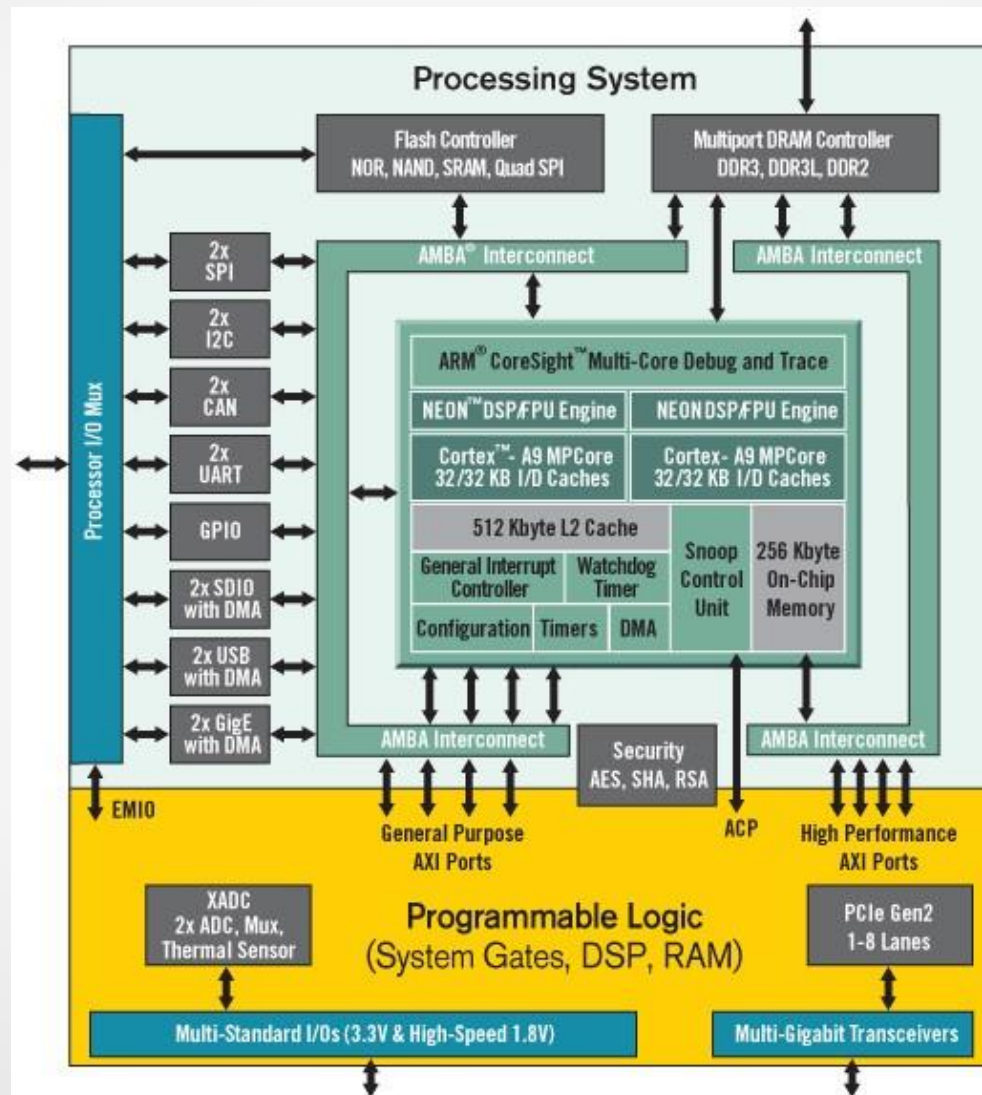
**Miguel Ángel García**

# Zybo

- AP SoC: Zynq 7000
- HDMI
- VGA
- 512MB RAM
- U\$D 189



# AP SoC Zynq 7000



# Suite Vivado

- Vivado IP Integrator: Diseño plataforma hardware
- Vivado SDK: Desarrollo del software
- Vivado HLS: Diseño de IP Cores utilizando lenguajes de ALTO nivel

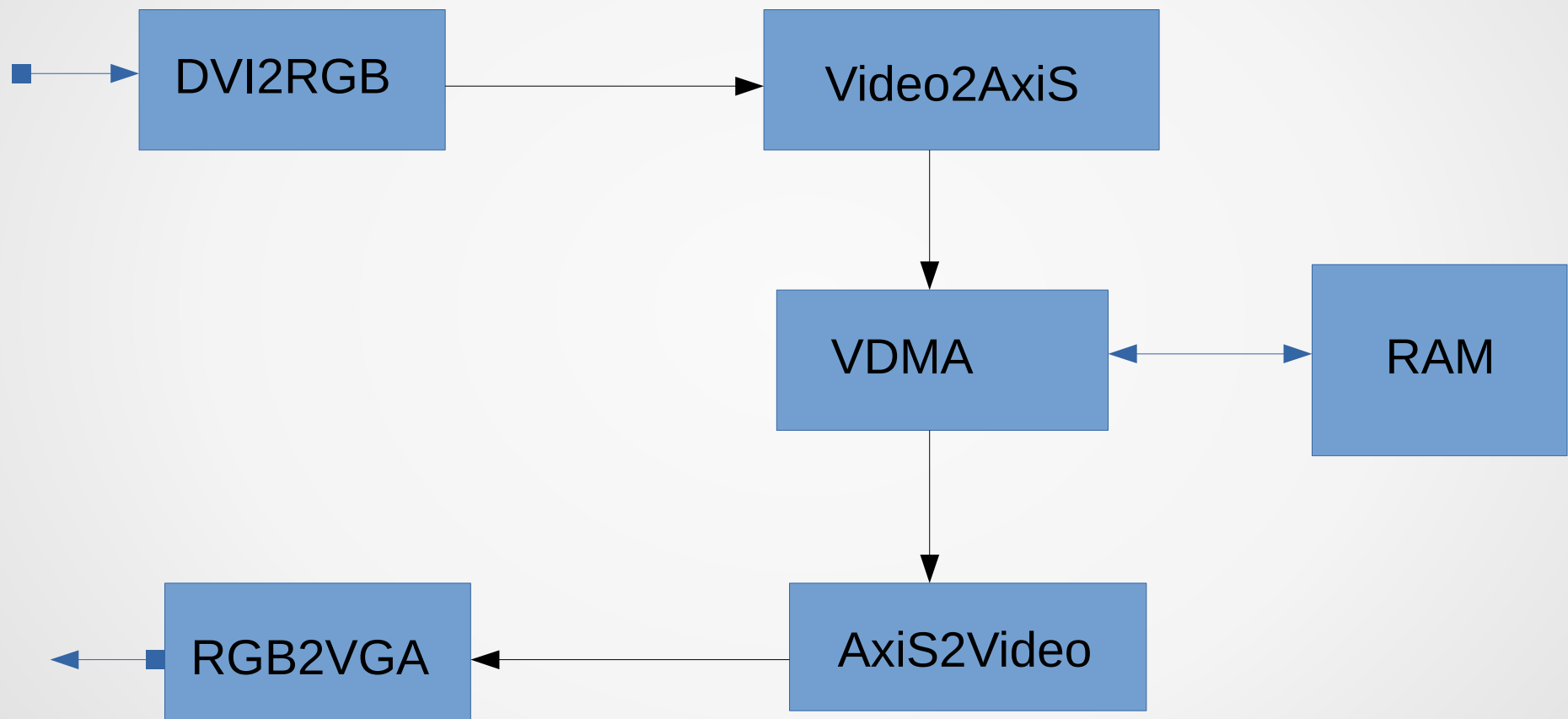
# Flujo de trabajo

- Solución inicial en PC.
- Implementación inicial en Zybo utilizando solo software.
- Medir y optimizar implementando funcionalidad en IP Cores.

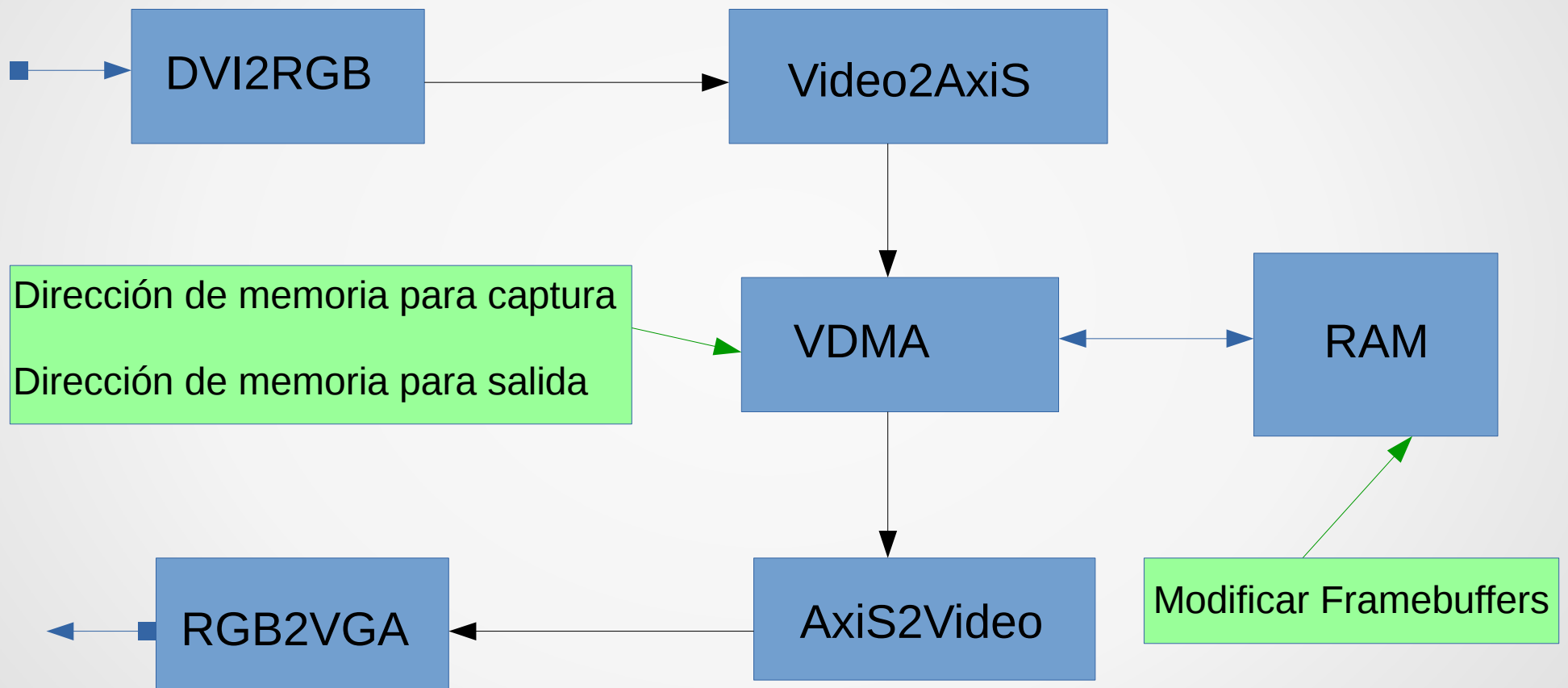
# Demo: procesamiento de video

- Capturar video → aplicarle un filtro → mostrar resultado
- Input: HDMI
- Output: VGA
- Basado en proyecto de Digilent:  
<http://github.com/Digilent/ZYBO>

# Demo: hardware de input / output



# Demo: Control desde el software





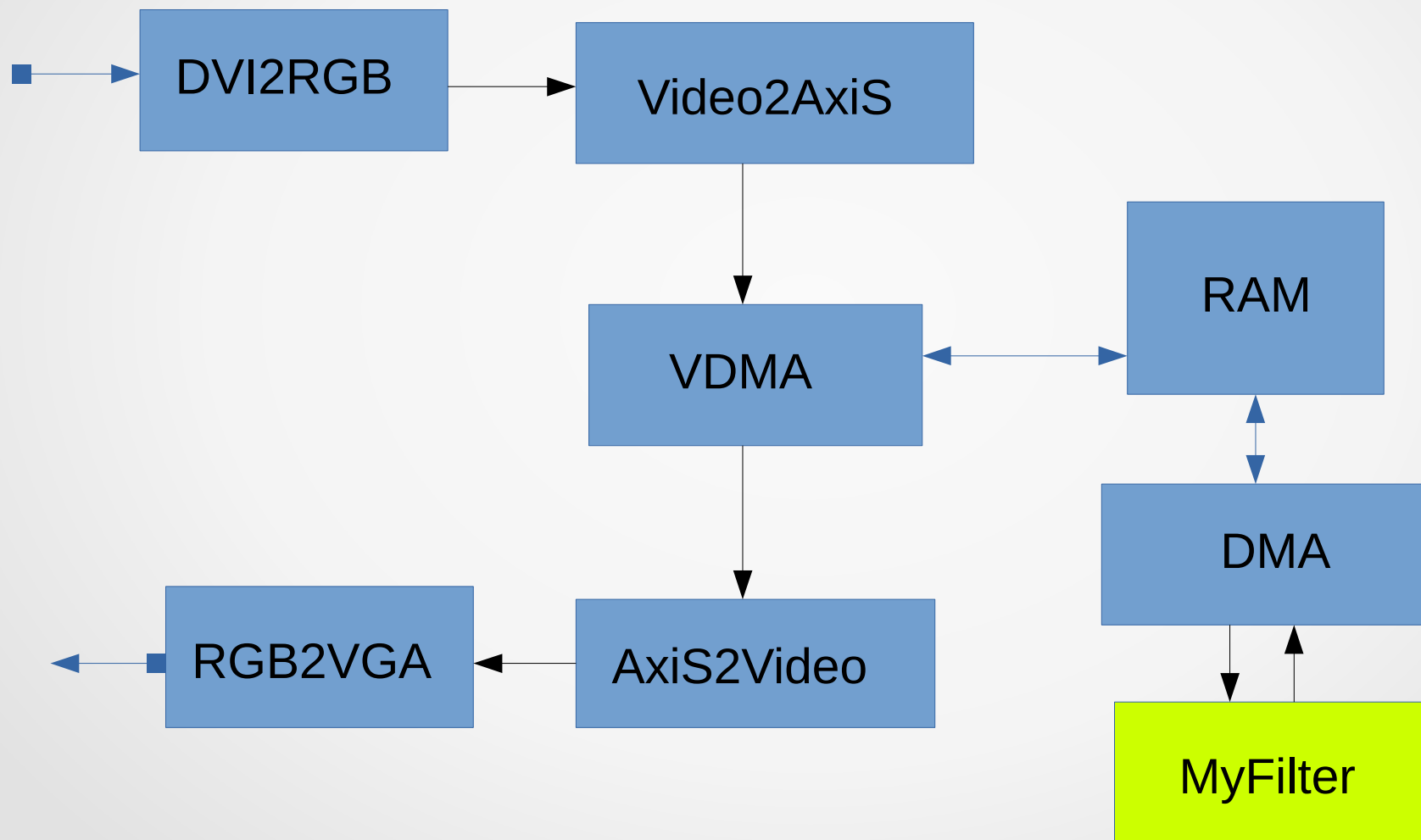
# Demo: prueba inicial

- Antes de optimizar chequear que todo funcione !

# Demo: Optimización

- Delegar el filtro de video al hardware
  - Diseñar IP Core
  - Integrarlo
  - Controlarlo desde el software

# Demo: hardware de input / output



# Demo: MyFilter en HLS

- Vivado High Level Synthesis (HLS) → permite especificar hardware en C o C++.
- Traducción “casi” automática a HDL, requiere:
  - Definir interfaces
  - Decidir sobre optimizaciones
  - Decidir los tipos de memorias a utilizar

Para esto se utilizan “directivas”

- Ojo: el resultado se ve impactado por la forma en que se codifica el algoritmo!

# Patrón de diseño IP Cores para imágenes

- Basado en documento técnico de **Xilinx** (Vallina).
- Aplicable a IP Cores que procesan **secuencialmente** los píxeles de una imagen:
  - Caracteriza entradas y salidas como flujos unidireccionales (**streams**).



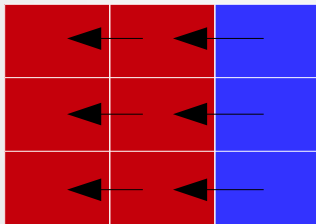
# Patrón: Jerarquía de memoria

- Para procesar cada píxel suele ser necesario un contexto de NxM alrededor del mismo.
  - Guardamos una porción de la imagen en dos tipos de **buffers**:
    - **Line Buffers (LB)**: acceso a un elemento para escritura y otro para lectura por ciclo: Block RAM
    - **Window**: acceso simultáneo a todos sus elementos: LUTs RAM

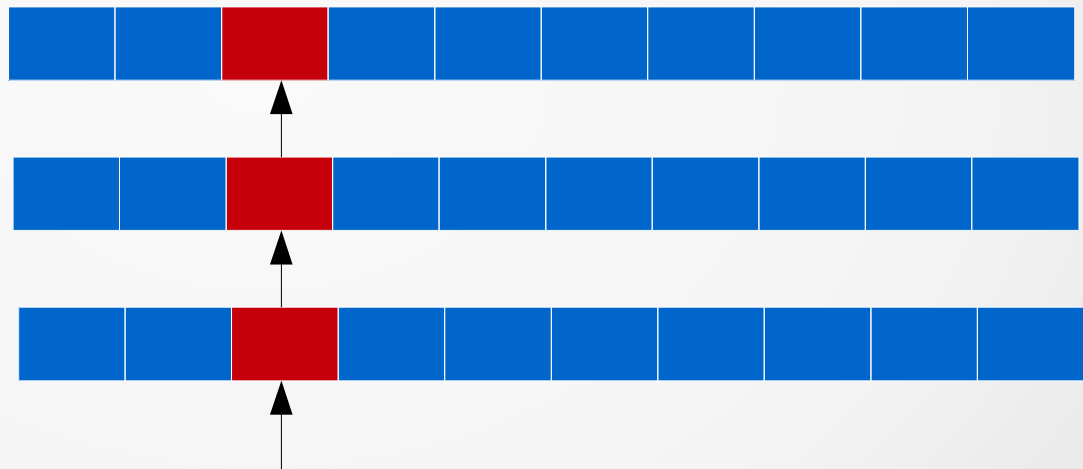
# Patrón: algoritmo general, paso 1

**Leer** un píxel del stream + **desplazar** Window y Line Buffers (ej 3<sup>er</sup> píxel de una línea):

Window:



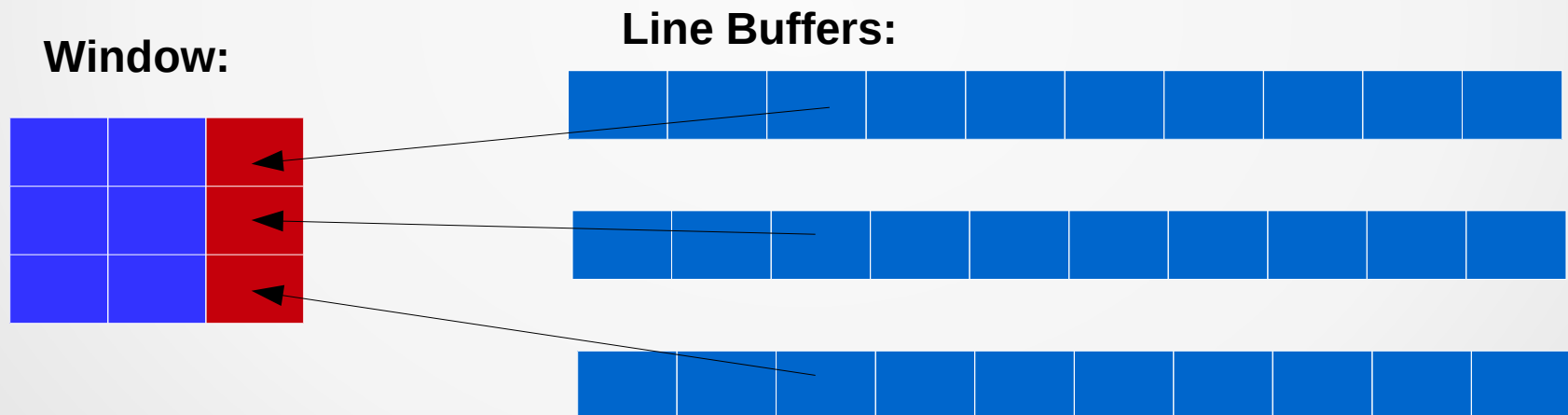
Line Buffers:



Stream de entrada

# Patrón: algoritmo general, paso 2

**Alimentar Window desde Line Buffers:**

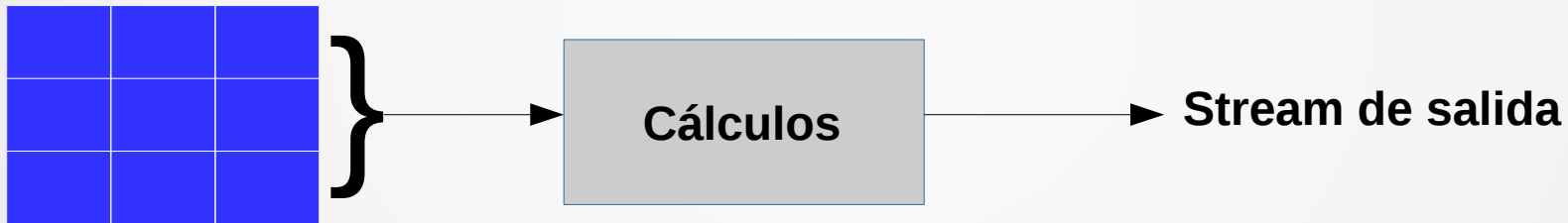




# Patrón: algoritmo general, paso 3

## Procesar Window:

Window:



# Demo: Integración

- Vivado IP Integrator:
  - Agregar controlador DMA
  - Agregar nuestro nuevo IP

# Demo: Control del IP

- Controlar el IP desde el software utilizando el driver generado por Vivado HLS.

# Muchas Gracias

<http://github.com/miguelgarcia/zybo-sase-2016>