Interfacing to and Processing Data from Image Sensors

March 19th 2013

Jose Roberto Alvarez

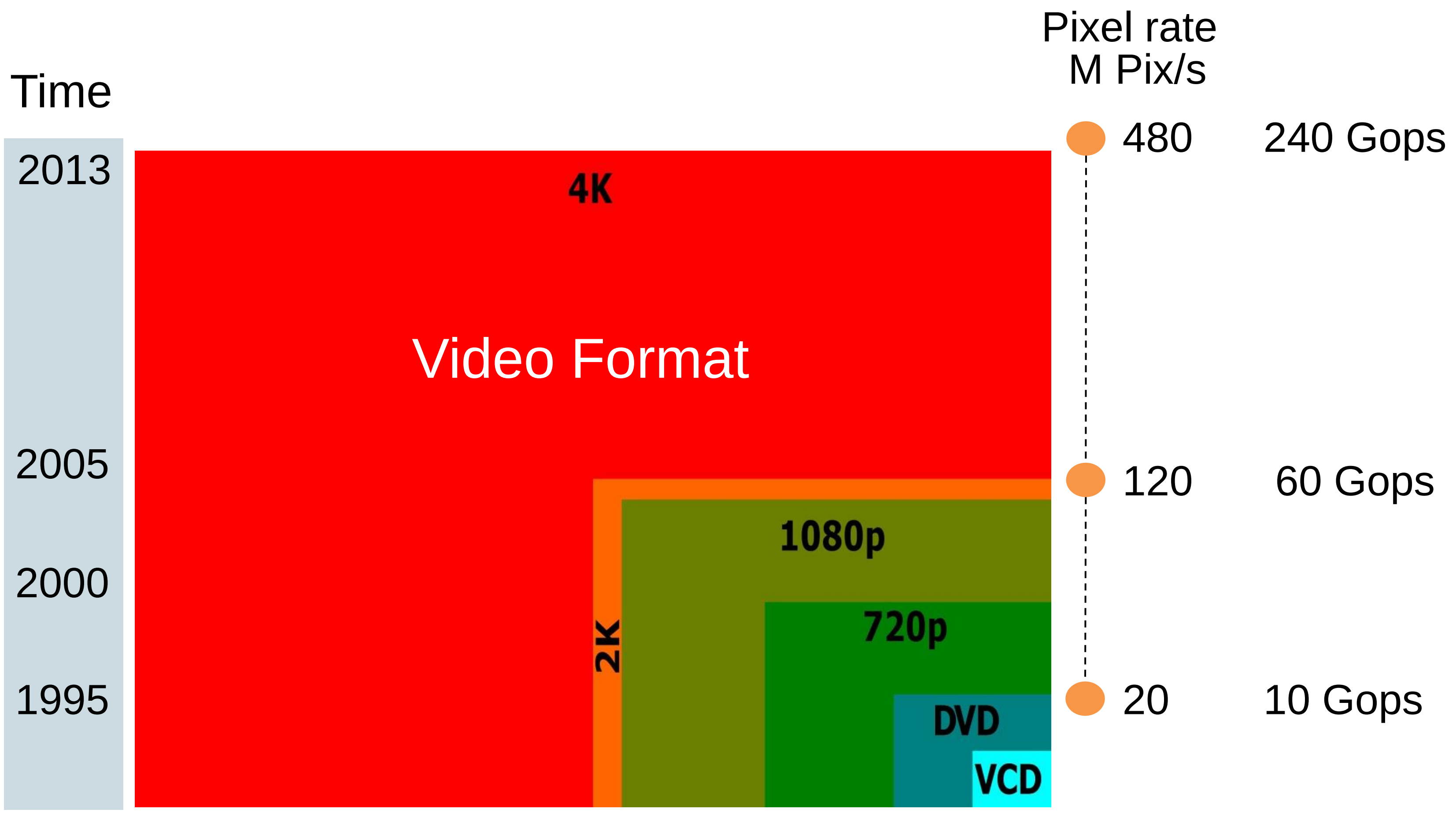
Engineering Director

Xilinx, Inc

Image Sensor considerations

- **Image size**
  - CIF, SD, HD, etc.
- **Frame rates**
  - 30 Fps, 60 Fps, 120 Fps
- **Data width**
  - Bits per word (8,10, 12, ...)
  - Endianess
  - Word Ordering (RBG, BGR, GBR, …)
- **Data Formats**
  - Parallel, serial, rgb, bgr, gbr
- **Color Spaces**
  - RGB, CMY, etc.
- **Synchronization**
  - Blanking, active video, hsync, vsync

Required Pixel Rate Processing vs. Capabilities

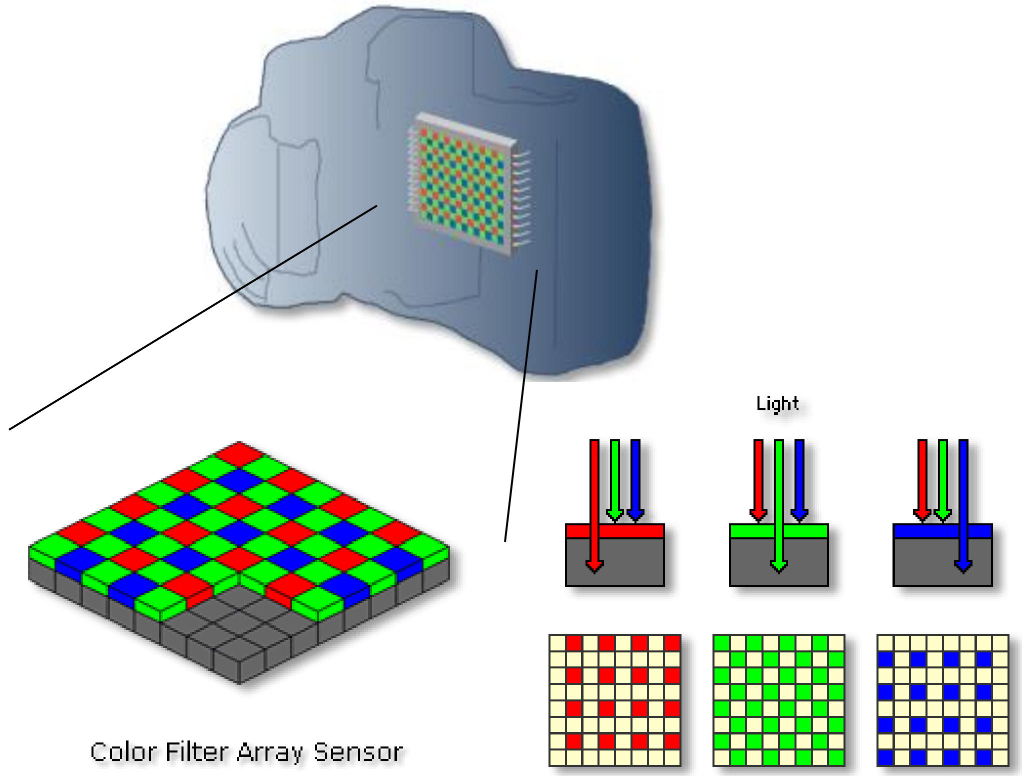


Data Rates – 8 Mpix sensor

Hsize	Vsize	Bits	Pic/sec	BW
3840	2160	8	30	1.99 Gbps
			60	3.98 Gbps
3840	2160	12	30	2.99 Gbps
			60	5.97 Gbps
3840	2160	16	30	3.98 Gbps
			60	7.96 Gbps

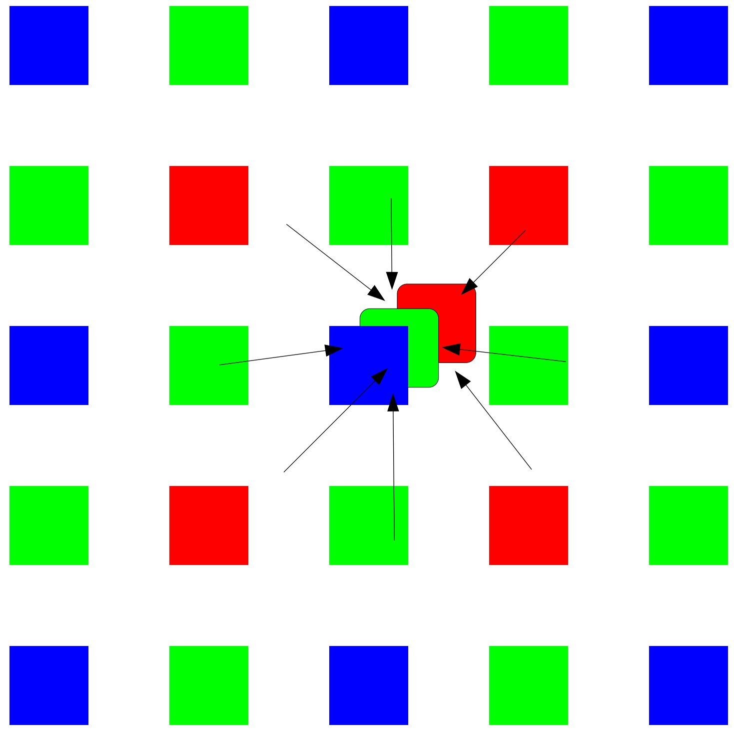
Hsize	Vsize	Bit/pel	Pic/sec	BW
3840	2160	24	30	5.97 Gbps
			60	11.9 Gbps
3840	2160	32	30	7.96 Gbps
			60	15.9 Gbps
3840	2160	36	30	8.95 Gbps
			60	17.9 Gbps

Color Filter Array - Purpose



- Dr. Bryce E. Bayer
- RGB, CMY

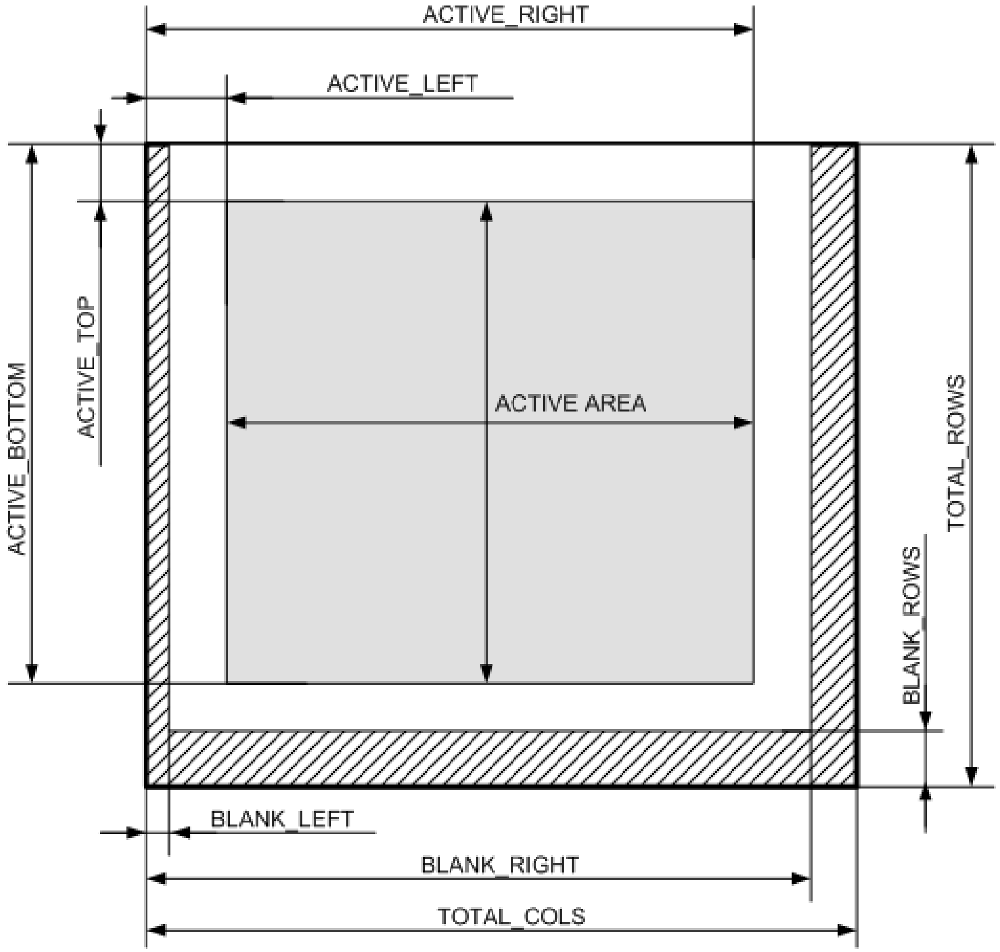
Imaging Sensor Subsampling



Twice as many Green elements

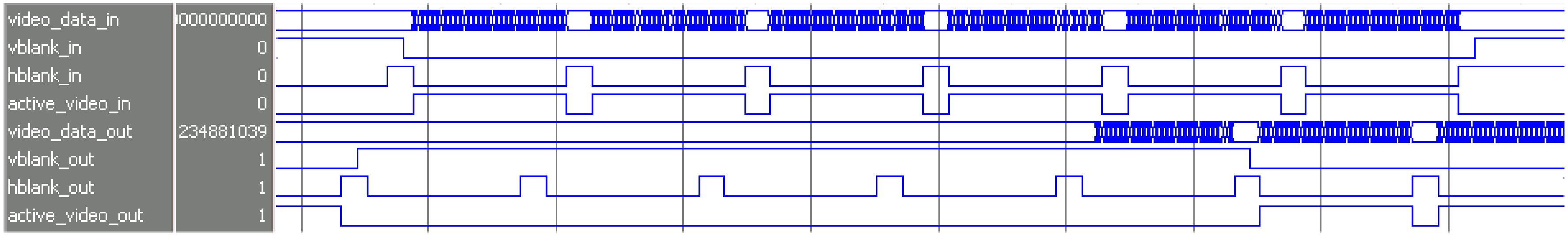
Timing Parameters

- ACTIVE_LEFT
- ACTIVE_RIGHT
- ACTIVE_TOP
- ACTIVE_BOTTOM
- TOTAL_ROWS
- TOTAL_COLS
- BLANK_ROWS
- BLANK_LEFT
- BLANK_RIGHT
- VBLANK_POLARITY
- HBLANK_POLARITY

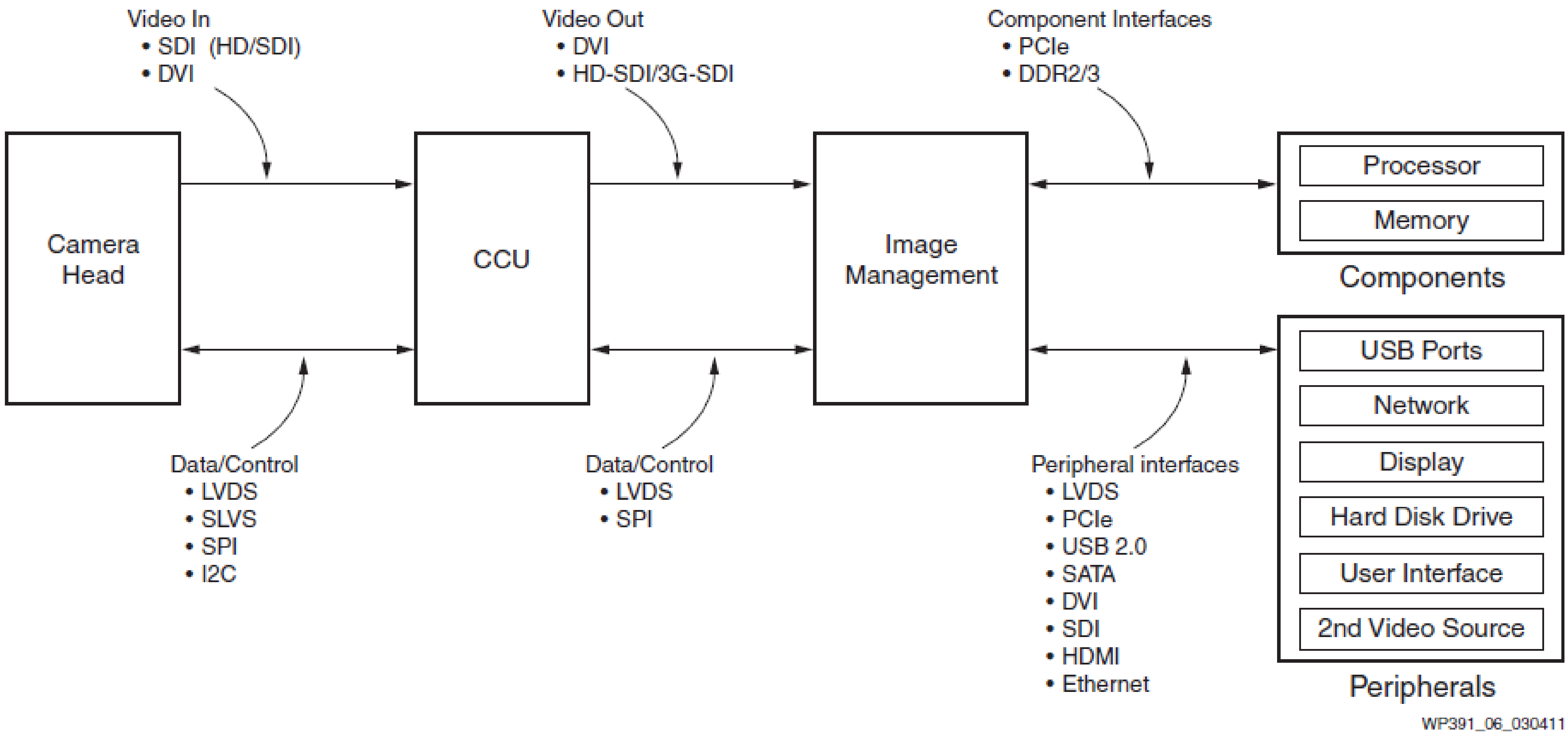


Timing Signal Diagram

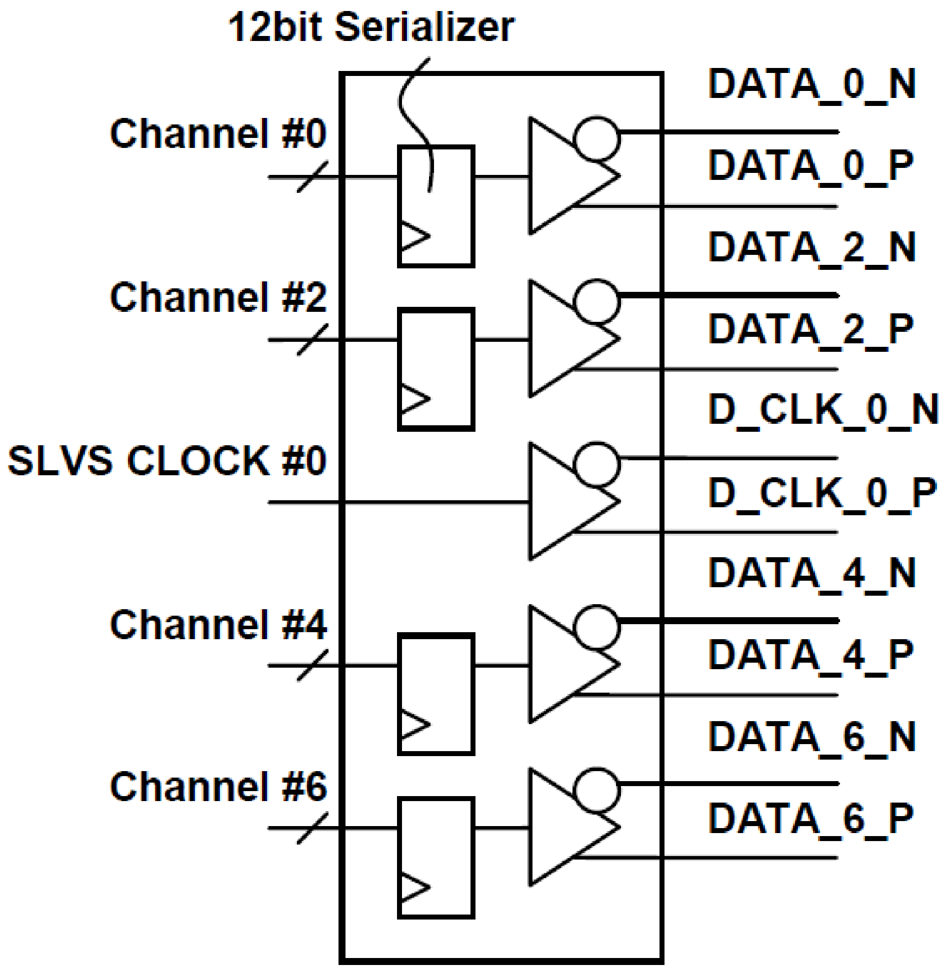
- timing signals: vblank, hblank,active_video



Common Interfaces in Camera Applications



Select I/O Resources



Select I/O Resources

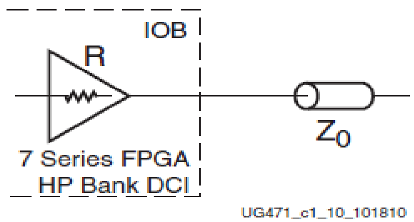
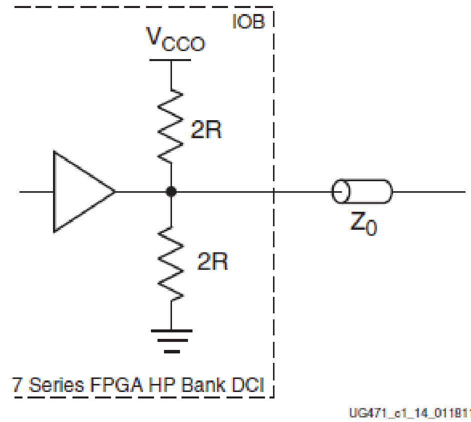
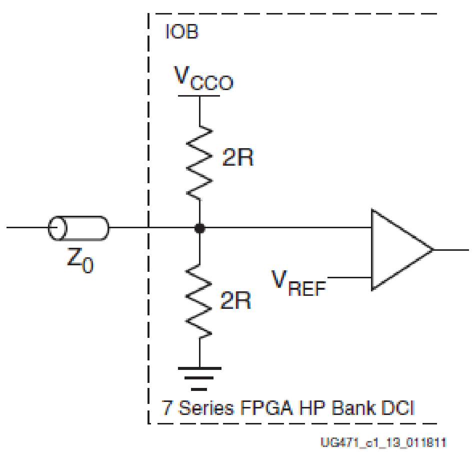


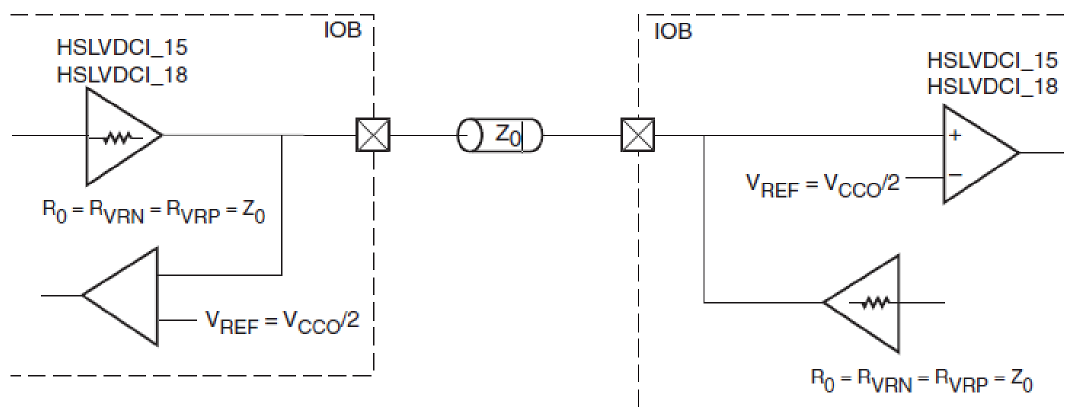
Figure 1-8: **Controlled Impedance Driver**



Driver with Termination to $V_{CCO}/2$ Using DCI Split Termination



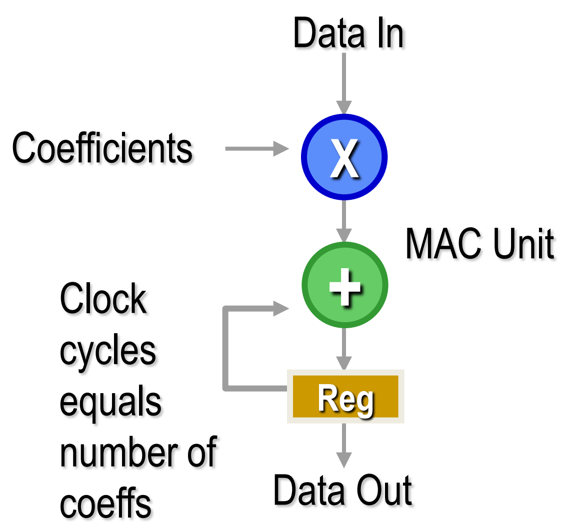
Input Termination to $V_{CCO}/2$ Using Split-Termination DCI



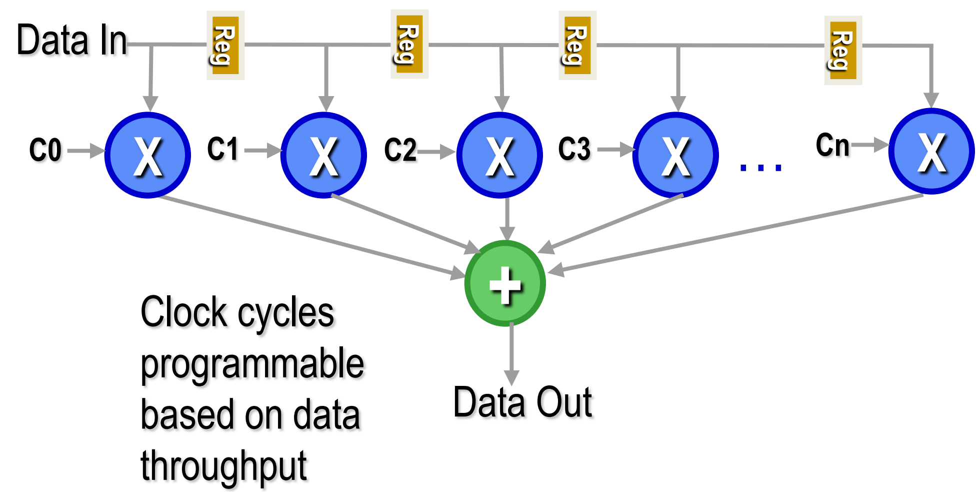
HSLVDCI Controlled Impedance Driver with Bidirectional Termination

Image Data Processing

Typical DSP processor – Sequential



FPGA - Parallel Implementation



Typical Camera Processing Application

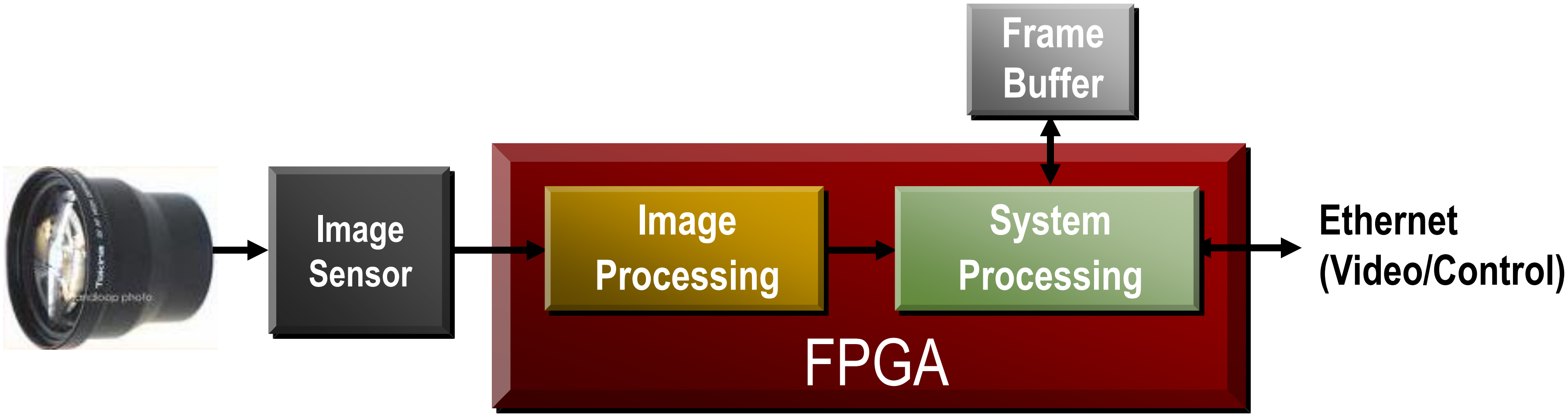


Image Processing

Base System
Defective Pixel Correction
Color Filter Array Demosaic
Color Correction Matrix
Gamma Correction
Color Space Conversion

Extended System
Dark Noise Reduction
Noise Reduction
Chroma Resampler
Picture Enhancement
Statistics/3A example

Video Processing
Video Scaler
On-Screen-Display
Image Characterization

System Processing

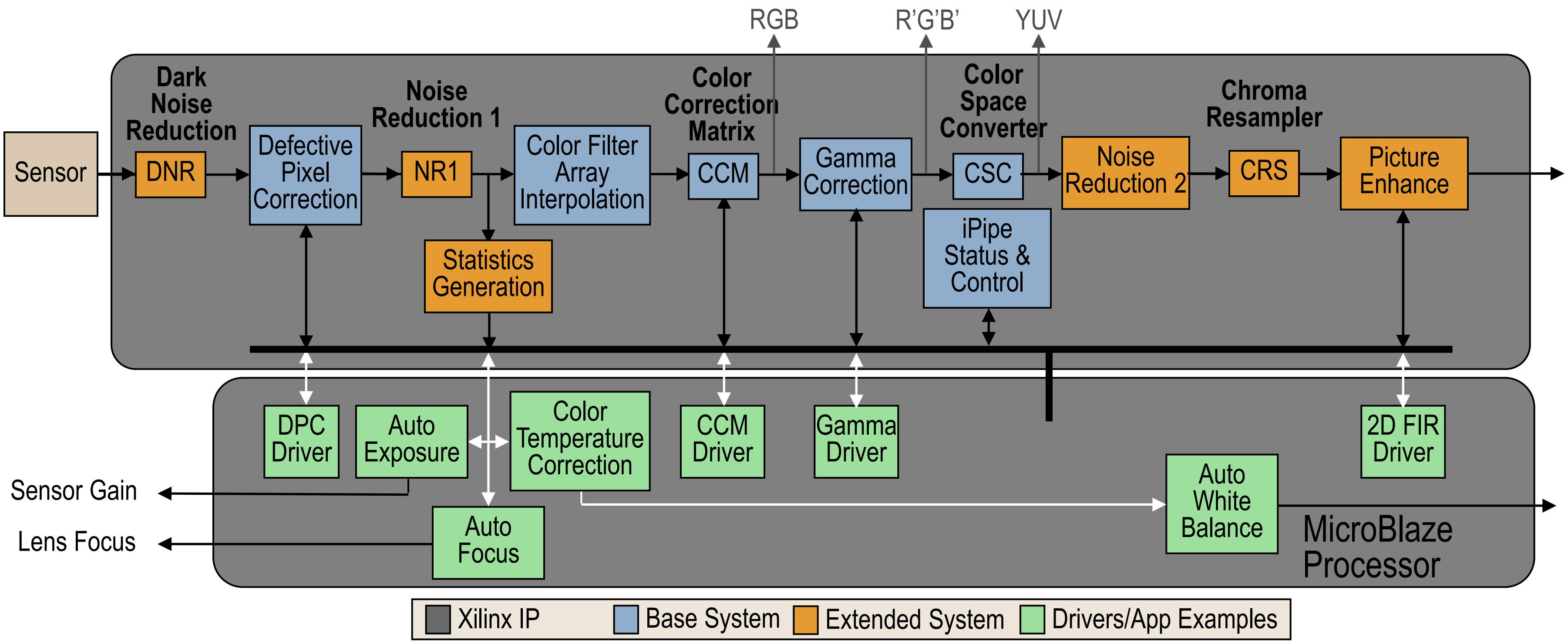
System Processor
MicroBlaze™ 32-bit Soft Processor
Dual Core ARM® Processors
Linux/Software

Interfaces
Tri-mode Ethernet MAC
USB 2.0
UART
GPIO
GigE Vision, Camera Link

Memory
Multi-port Memory Controller
Video Frame Buffer Controller
Hardened Memory Controller

Compression
H.264
MPEG-2
Others

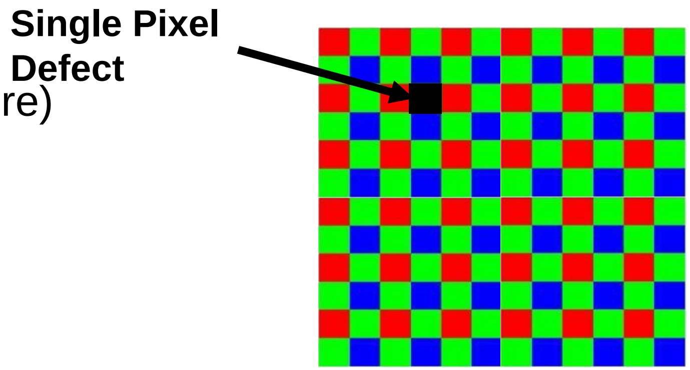
Xilinx Image Processing Pipeline

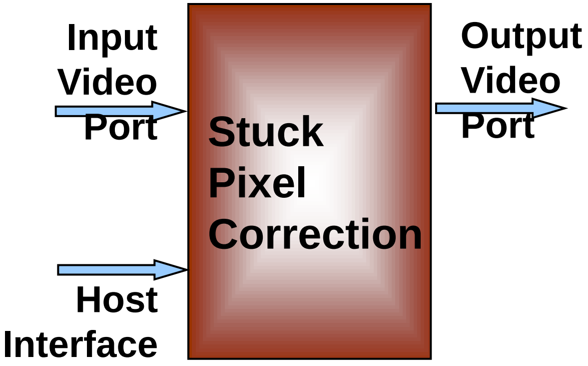


- **Available as individual IP cores**
- **Reference Designs available**

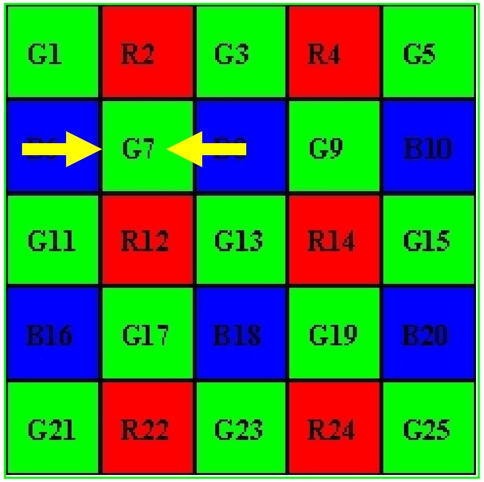
Defective Pixel Correction

- **Corrects defective pixels that are:**
  - Static (always present)
  - Dynamic (function of temperature or exposure) Including:
    - Dead (always low)
    - Hot (always high)
    - Stuck (to a certain value)
- **Features**
  - Dynamic – "On-the-fly" detection and correction
    - Each pixel is compared to the median value of
      - Nearest neighbors
      - Nearest neighbors in the same color plane
    - Programmable thresholds for each comparison
    - Pixels greater than one or more threshold are replaced with median value of nearest neighbors in the same color plane

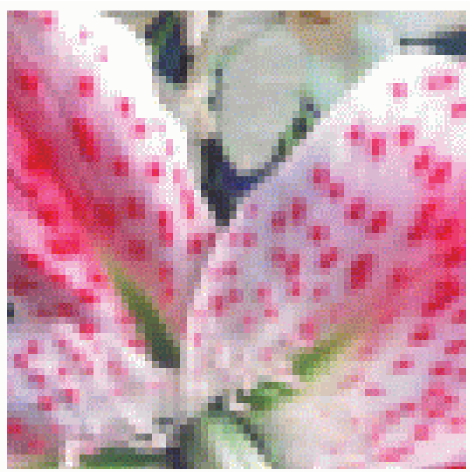




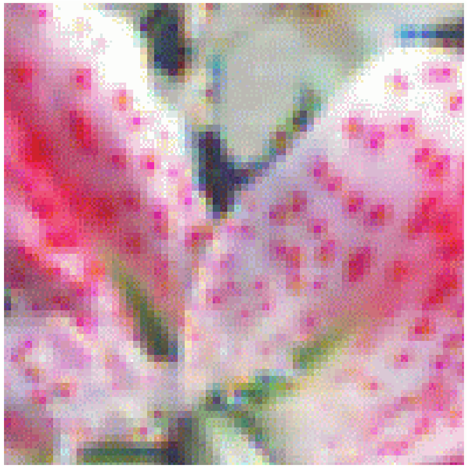
Linear Interpolation



$$B7 = (B6+B8) / 2$$

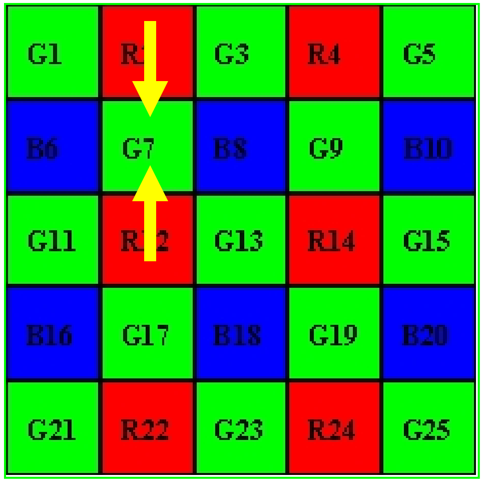


Ideal

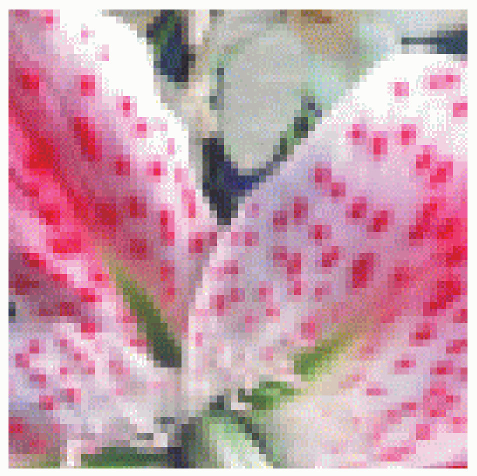


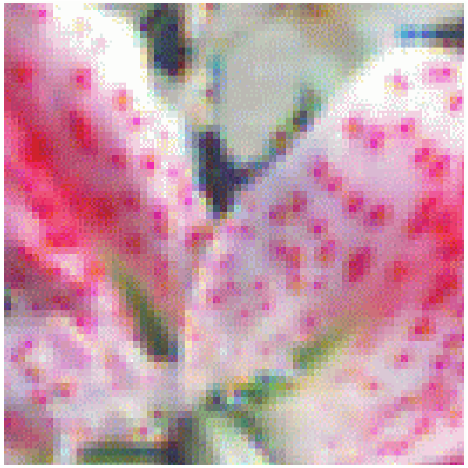
Interpolated

Linear Interpolation

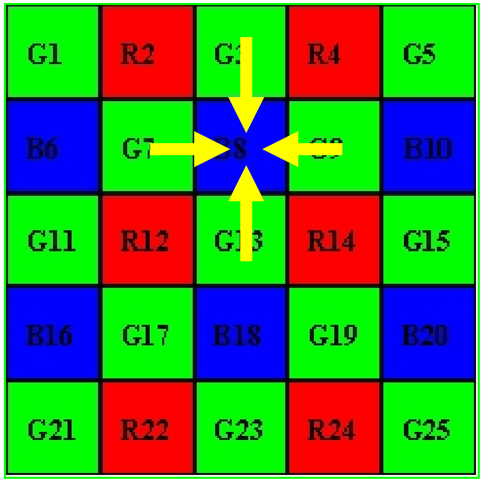


$$R7 = (R2+R12) / 2$$

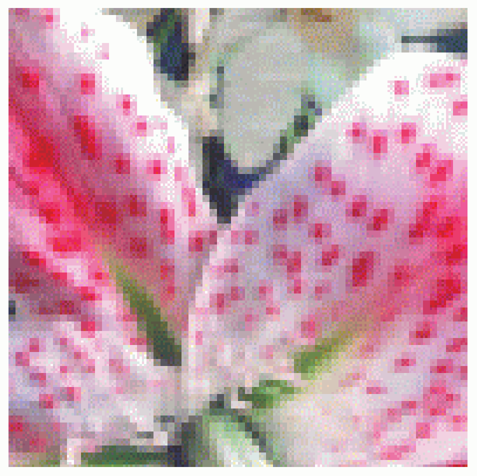

Ideal

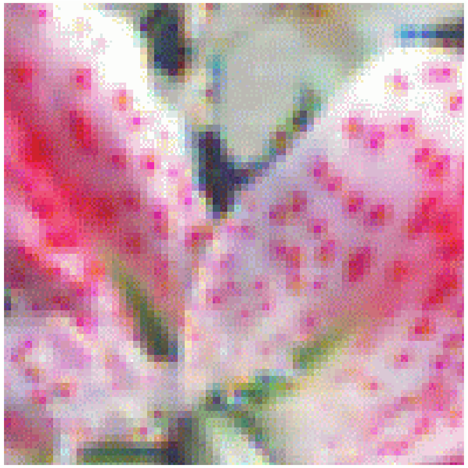

Interpolated

Linear Interpolation

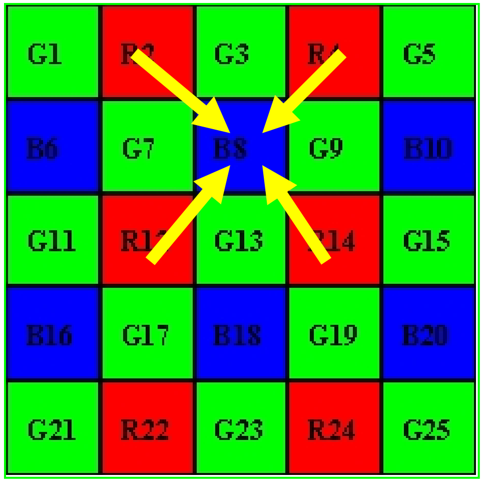


$$G8 = (G3+G7+G9+G13) / 4$$

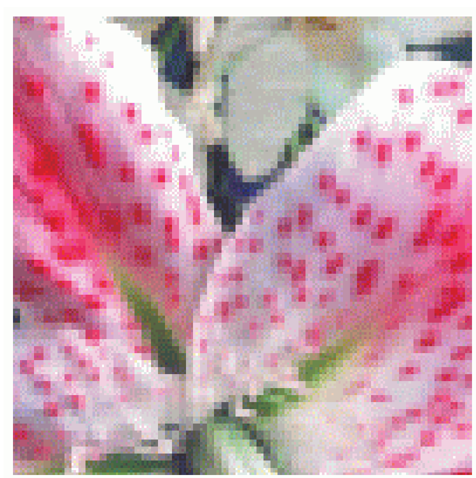

Ideal


Interpolated

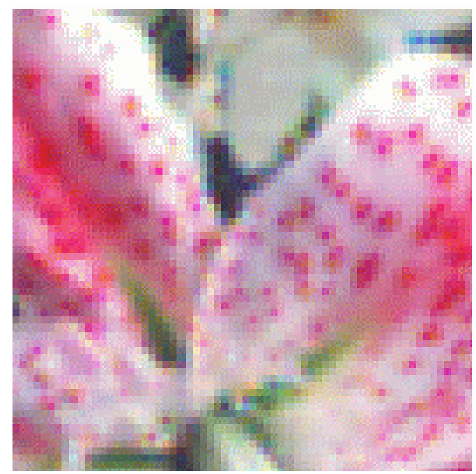
Linear Interpolation



R8 = (R2+R4+R12+R14) /4

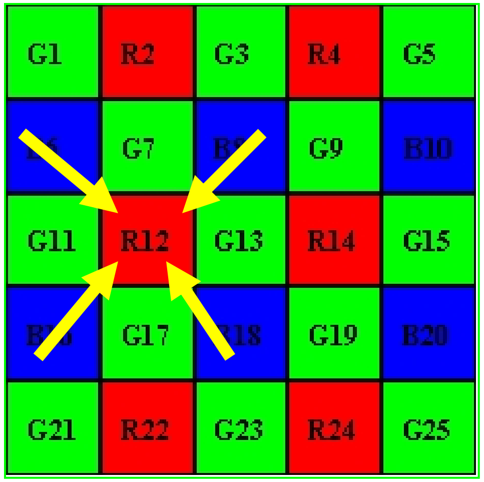


Ideal

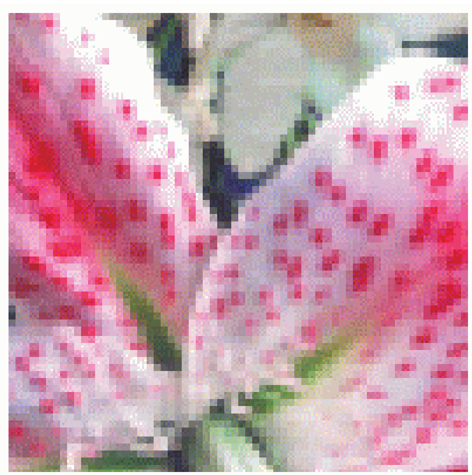


Interpolated

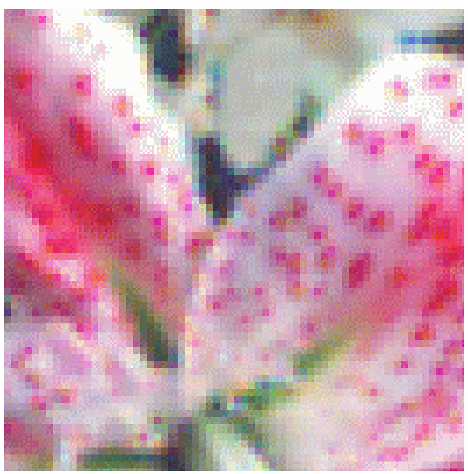
Linear Interpolation



B12 = (B6+B8+B16+B18) / 4

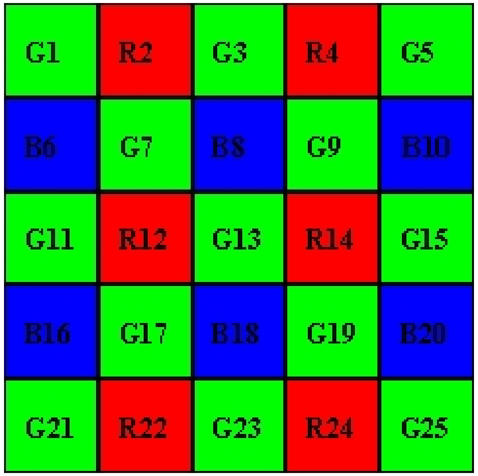


Ideal



Interpolated

Linear Interpolation



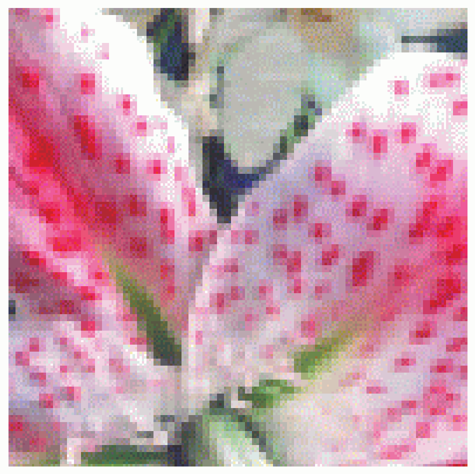
G8 = (G3+G7+G9+G13) / 4

B7 = (B6+B8) / 2

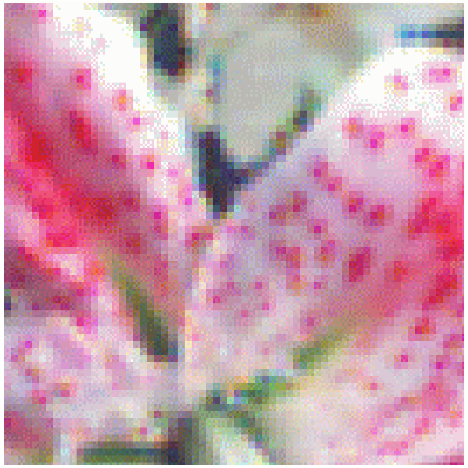
R7 = (R2+R12) / 2

R8 = (R2+R4+R12+R14) /4

B12 = (B6+B8+B16+B18) / 4



Ideal



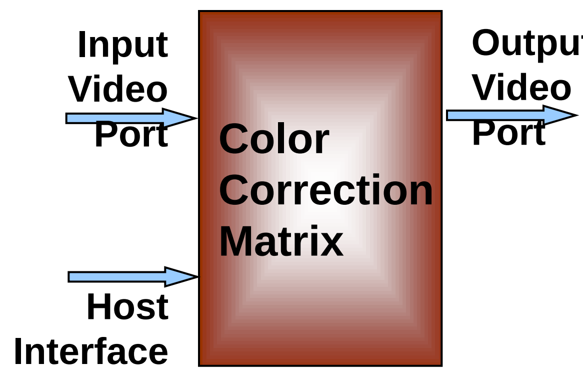
Interpolated

Color Correction Matrix

$$\begin{bmatrix} R_c \\ G_c \\ B_c \end{bmatrix} = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} ROffset \\ BOffset \\ GOffset \end{bmatrix}$$

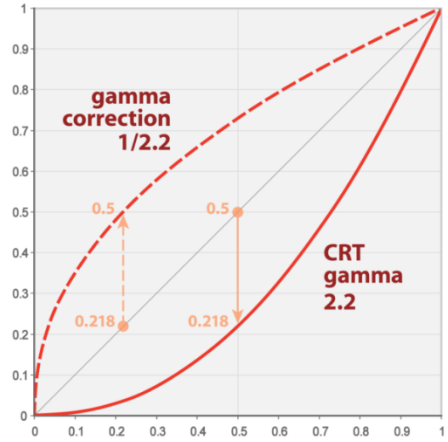
Uncorrected **Corrected**

- **Enables white balance, color cast, brightness and contrast corrections for RGB images**
  - 3x3 programmable coefficient matrix multiplier with offset compensation
- **Features:**
  - Optimal resource usage and high performance
  - Optional CMY input to RGB output color conversion
  - Independent clipping and clamping control



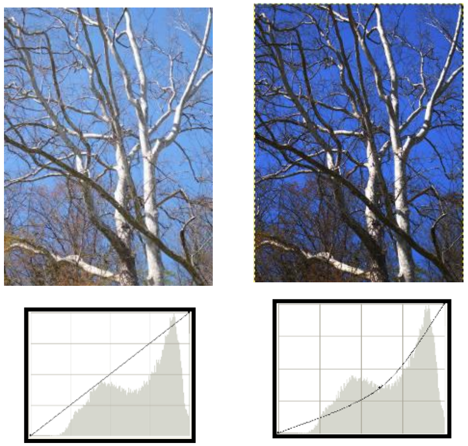
Gamma Correction

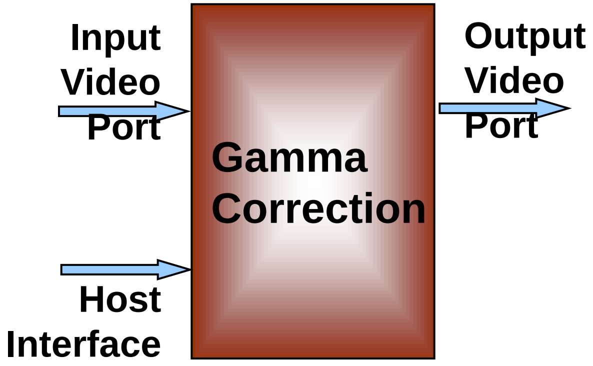
- Gamma correction (gamma compression, gamma encoding) encodes linear luminance or RGB values into video signals.
- Gamma expansion is the inverse, occurs in CRT monitors due the nonlinearity of the electron-gun current–voltage curve.
- Gamma correction is defined by $V_{out} = V_{in}^{\gamma}$, where the input and output values are between 0 and 1.
- The case γ<1 is gamma compression, γ>1 is gamma expansion.



Gamma Correction

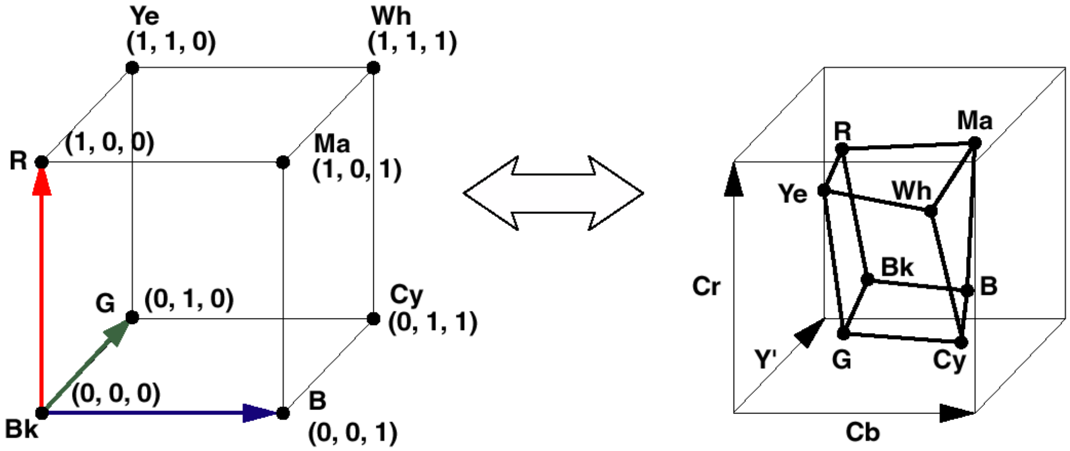
- **Manipulates image data to match the response of display devices**
  - Programmable Look-Up Table (LUT) Structure
- **Features**
  - Three implementation options:
    - Single LUT applied to all color channels
    - Independent LUT's for each of three colors
  - Multiple Block RAM usage optimization options
  - Multiple processor interface options





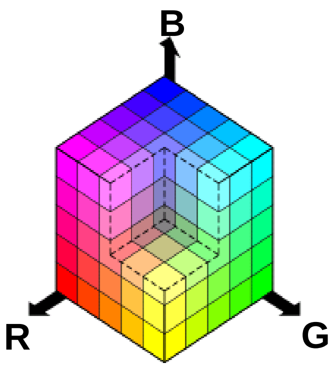
Color-Space Conversion

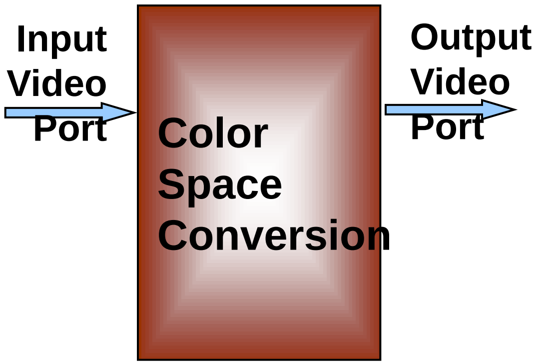
$$\begin{bmatrix} Y \\ C_R \\ C_B \end{bmatrix} = \begin{bmatrix} c_A & 1-c_A-c_B & c_B \\ c_C(1-c_A) & c_C(c_A+c_B-1) & c_C(-c_B) \\ c_D(-c_A) & c_D(c_A+c_B-1) & c_D(1-c_B) \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} O_Y \\ O_C \\ O_C \end{bmatrix}$$



Color Space Converter

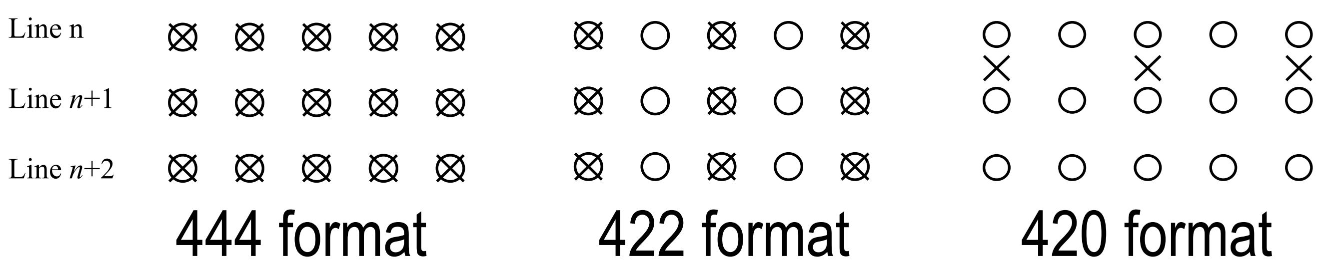
- **RGB to YCrCb Color Space Converter**
- **YCrCb to RGB Color Space Converter**
  - Simplified 3x3 matrix multiplier
  - Converts three input color samples to three output samples in a single CLK cycle
- **Features**
  - Built-in support for
    - SD (ITU 601)
    - HD (ITU 709) PAL
    - HD (ITU 709) NTSC
    - YUV
  - User defined conversion matrices supported
  - Optimal resource usage and high performance





Chroma Sub-Sampling

- Chrominance information frequently needs to be sub-sampled in order to reduce processing, storage and transmission overhead.
- Commonly used sub-sampled chroma formats: 422 and 420.
- Conversion to 422 format requires only horizontal FIR filtering.
- Conversion to 420 format requires vertical interpolation between chrominance components as well



Summary

- The essence of embedded vision is applying sophisticated algorithms to image data
- Interfacing to image sensors is complex
- Implementing sophisticated algorithms on real-time video data is challenging
- FPGAs and FPGA-processor combinations address the key challenges for these applications
  - Flexible / High Speed Interfaces
    - Physical connection
    - Data formats
    - Serial, Parallel, Standard (LVDS, HDMI, MIPI, etc.)
  - Multiple Data Formats
  - Logic resources for flexible parallel processing
  - Design resources to support embedded vision applications
  - RTL, DSP, Embedded, High Level Synthesis flows
  - Library of intellectual property cores (IP)

The Embedded Vision Alliance

Free Resources on Embedded Computer Vision

The Embedded Vision Alliance web site, at www.Embedded-Vision.com, covers embedded vision applications and technology, including interviews and demonstrations

Register on the Alliance web site for free access to:

- The Embedded Vision Academy—free in-depth tutorial articles, video "chalk talks," code examples, and discussion forums.
- Embedded Vision Insights—bimonthly newsletter with industry news and updates on new resources available on the Alliance website.

Embedded vision technology and services companies interested in becoming sponsoring members of the Alliance may contact info@Embedded-Vision.com

Embedded Vision Insights
The Latest Developments on Designing Machines that See

Follow us

Embedded Vision Summit

A Free Educational Event for Engineers—San Jose, April 25th

Learn how to use the hottest new technology in the industry to create "machines that see"

- Technical presentations on sensors, processors, tools, and design techniques
- Keynote by Prof. Pieter Abbeel, UC Berkeley, a leader in developing machine intelligence
- Cool demonstrations and opportunities to meet with leading vision technology suppliers

embedded VISION SUMMIT
APRIL 25 2013 • SAN JOSE

Co-located with UBM Electronics' DESIGN West

- DESIGN West also includes the Embedded Systems Conference, Black Hat Summit, and exhibits

The Summit is free, but space is limited. To register to attend, go to www.embedded-vision.com/embedded-vision-summit

Interfacing to and Processing Data from Image Sensors

March 19th 2013

Jose Roberto Alvarez

Engineering Director

Xilinx, Inc