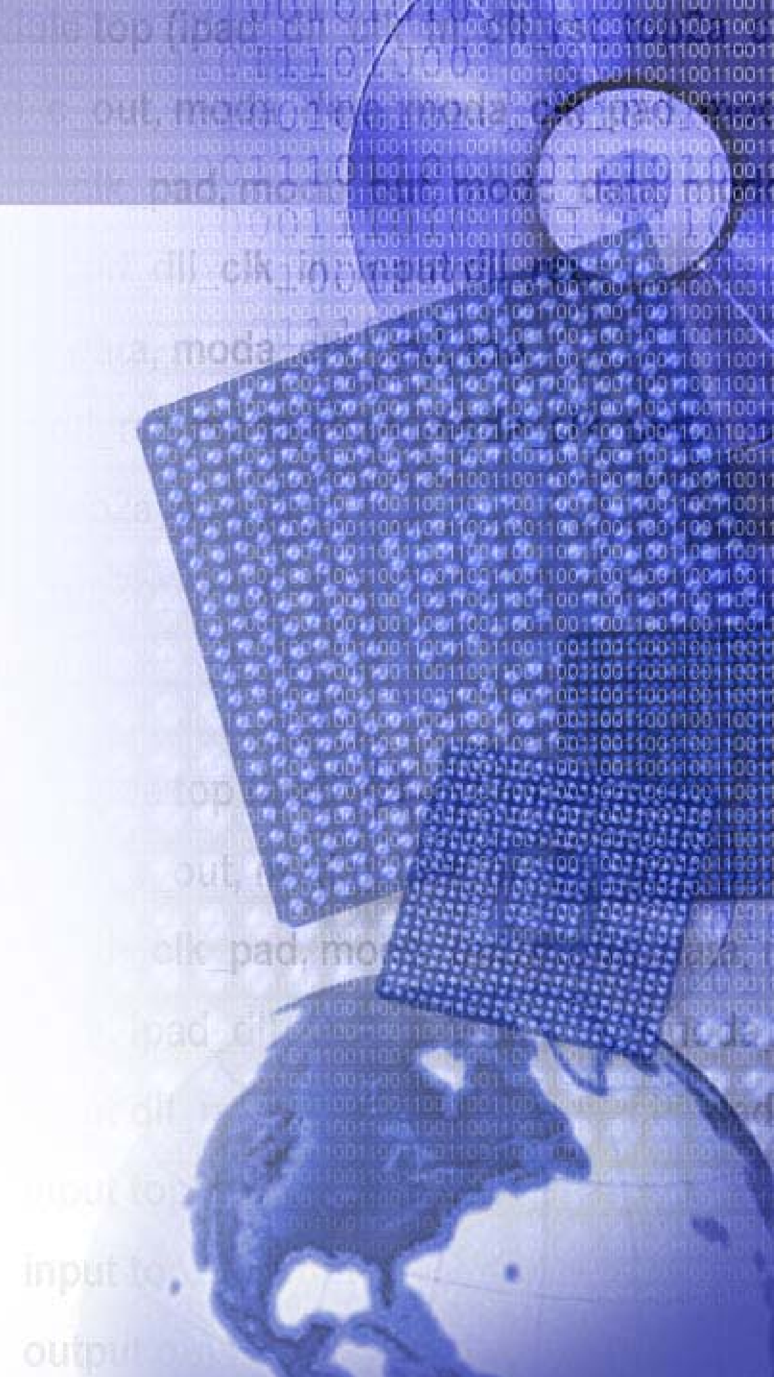
Digital Video & Image Processing

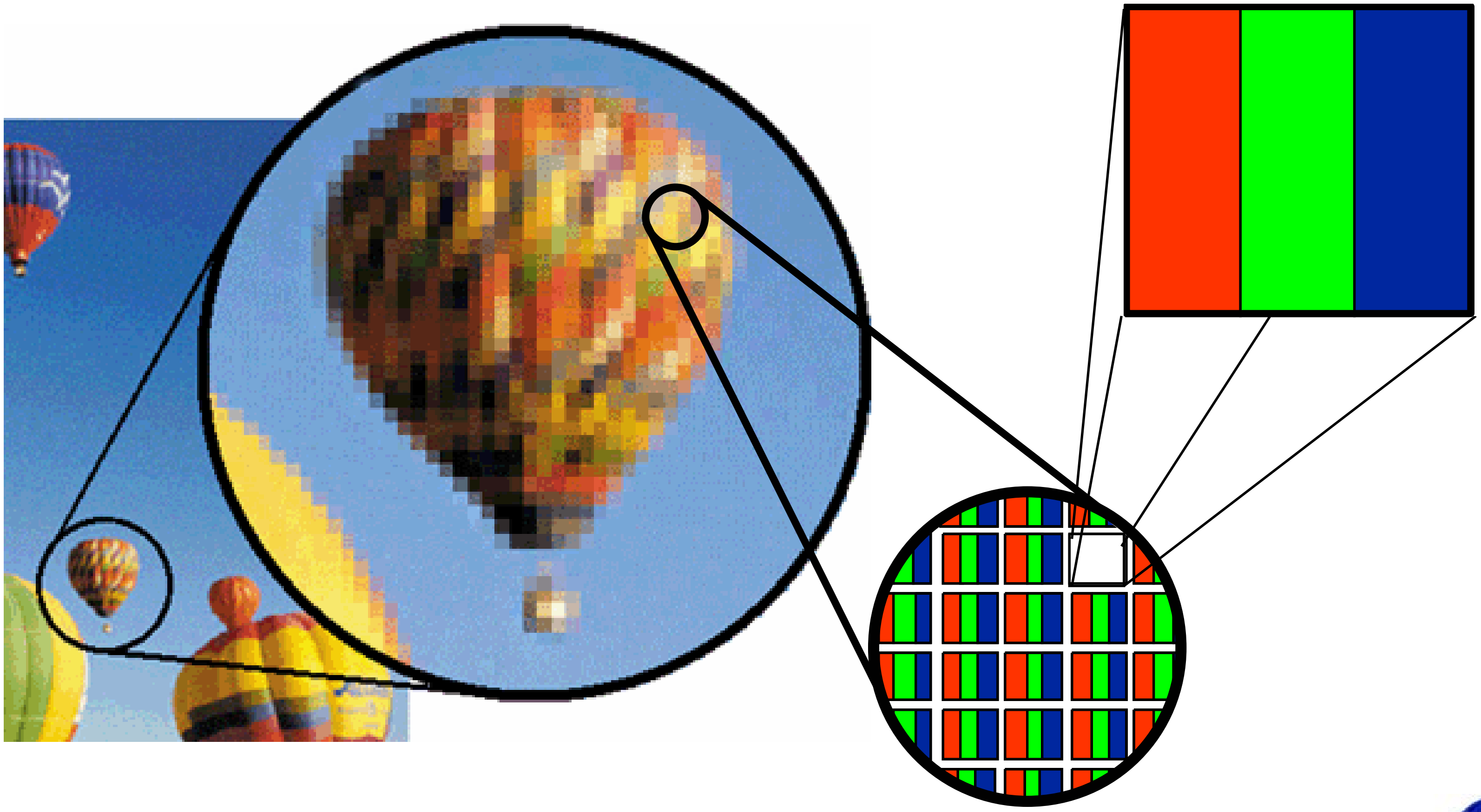
Xilinx Solutions for the Broadcast Chain



What Makes up Digital Images or Digital Video?

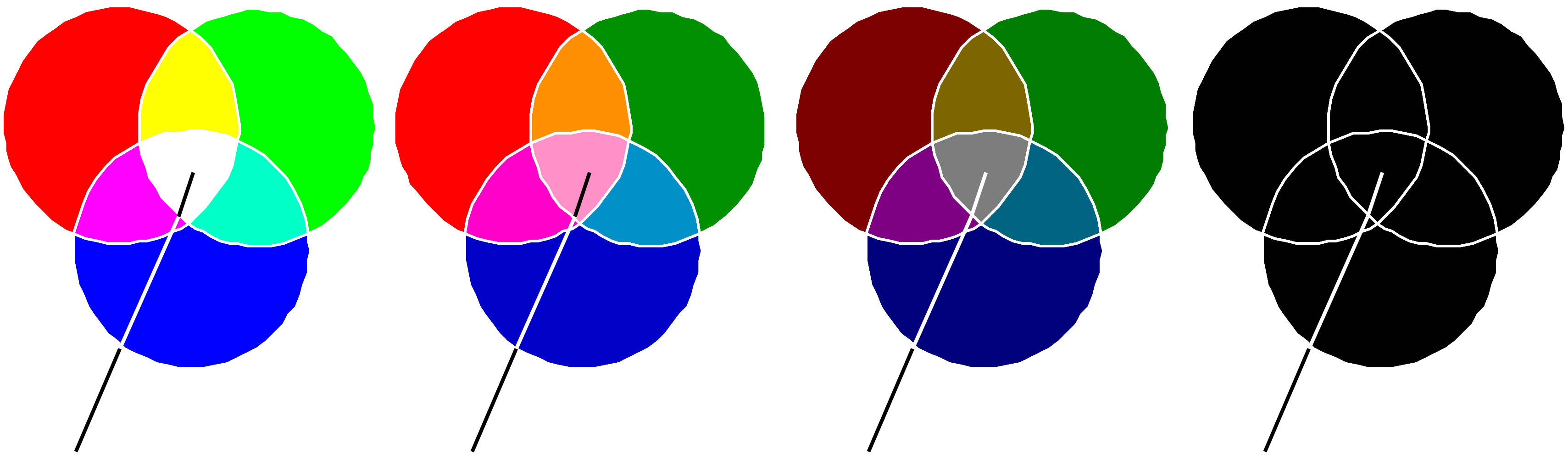
Anatomy of a Pixel

- Each pixel comprised of three color producing sub-pixel elements: Red, Green, and Blue (RGB)



Combining RGB

- Sub-pixel RGB intensity controls overall pixel colour



White =
Max Red
Max Green
Max Blue

Pink =
Max Red
Medium Green
Med./High Blue

Medium Gray =
Medium Red
Medium Green
Medium Blue

Black =
No Red
No Green
No Blue

Bandwidth/Quality Tradeoffs

- Typical high definition (HD) system needs high bandwidth
  - 1920 x 1080 resolution, 24-bit pixels (8-bit Red, Green and Blue values), 30 progressive frames per second

Bandwidth = 1920 x 1080 x 24 x 30 = 1.49Gbps

- Techniques for memory/bandwidth reduction have varying effect on picture quality
- Continuous improvements in compression techniques and filtering to reduce effects on picture quality but allow more data "down the pipe"

Reduce Pixel Levels

- 24-bit image
  - 8 bits each for RGB
  - Over 16 million levels/pixel
- 4-bit image
  - Only 16 levels/pixel

Reduced bandwidth/memory requirements but reduced quality

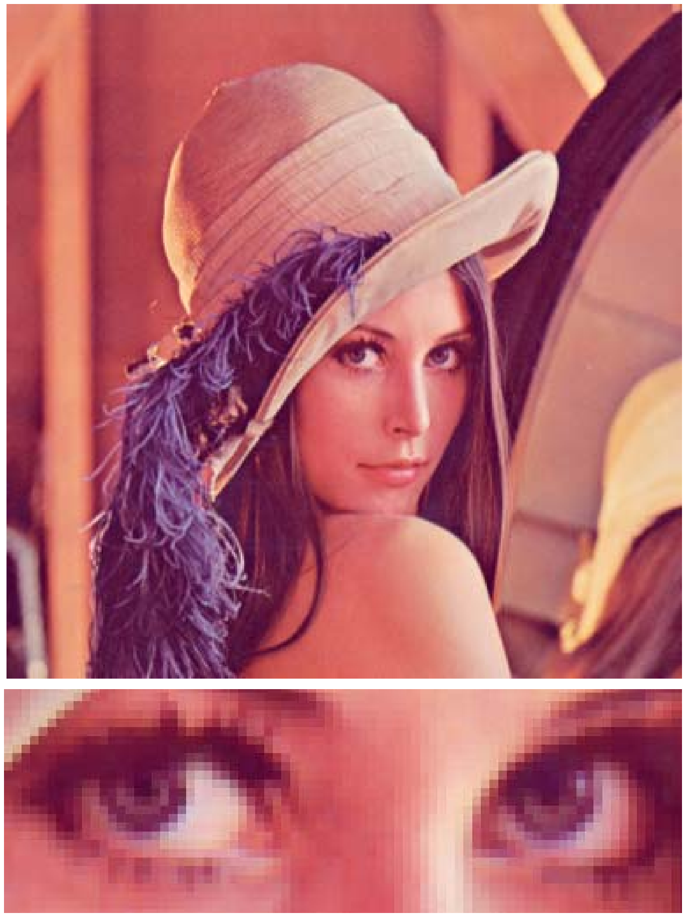
Reduce Spatial Resolution

- Original image
- 1 pixel/unit area

- 64 pixels/unit area

Reduced bandwidth/memory requirements but reduced quality

Compression



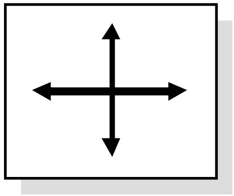
Original uncompressed image

Compressed image with block artifacts

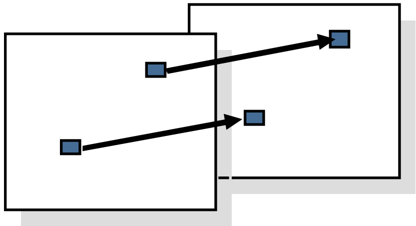
Reduced bandwidth/memory requirements but reduced quality

MPEG Compression

- Spatial Processing
  - Uses DCT within a single picture to enable removal of high frequencies not discernable to human eye



- Temporal Processing
  - Seeking out and removing redundancy between successive images/frames

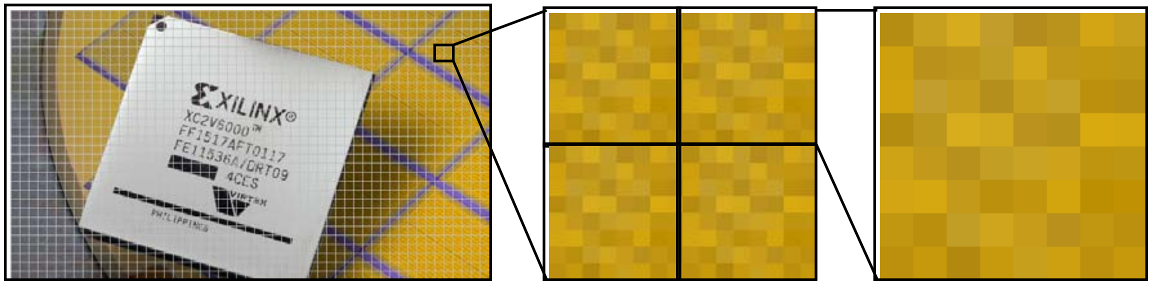


- Variable Length Coding (VLC)
  - Use shortest codes for most common samples
- Run Length Encoding (RLE)
  - Replace long strings of zeros with single command code

Spatial Redundancy

- DCT
  - Returns the discrete cosine transform of ‘video/audio input’
  - Can be referred to as the even part of the Fourier series
  - Converts an image or audio block into its equivalent frequency coefficients
- IDCT
  - Inverse of the DCT function
  - IDCT reconstructs a sequence from its discrete cosine transform (DCT) coefficients

DCT in MPEG Compression



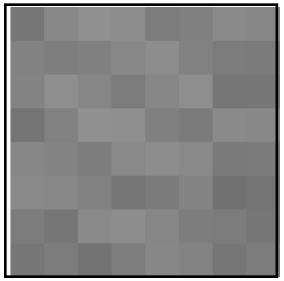
1

Scan picture using 16x16 pixel "macroblock"

2

Scan macroblock in 8x8 blocks

3



Determine luma & chroma pixel values

4

119	137	145	142	125	129	138	136
131	127	129	137	142	130	125	124
132	143	135	125	136	143	119	120
117	130	143	129	124	139	137	138
136	133	126	138	142	138	123	124
138	137	131	119	124	133	115	117
125	119	138	142	135	126	125	119
119	125	116	127	136	132	119	125

Luma samples shown here (Chroma processed separately)

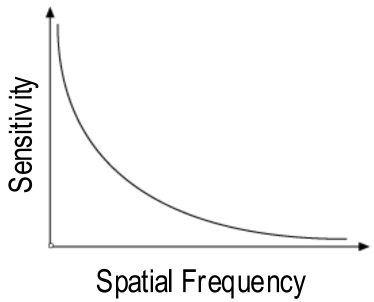
8

Compress using zigzag scan and run length encoding for zero values (in blue)
Further compression with Huffman encoding (VLC)

7

920	15	28	18	14	5	1	0
17	10	9	10	4	1	0	0
10	14	9	5	1	0	0	1
6	5	4	4	2	1	0	0
2	4	7	1	0	0	0	0
2	3	2	1	1	0	0	0
1	3	2	1	0	0	0	0
0	1	0	0	0	0	0	0

Quantize higher frequencies with less bits (weighting)
Zero values for frequencies below perception threshold



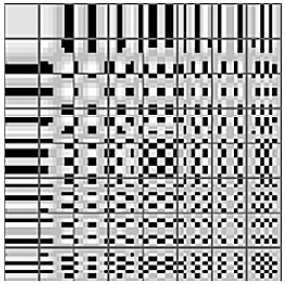
Human eye less sensitive to high frequencies

6

6981	245	500	364	331	139	35	21
205	168	192	278	55	40	12	4
152	288	207	99	26	85	29	10
109	142	166	111	77	66	38	2
55	84	212	56	10	4	11	5
61	136	61	43	47	13	3	2
32	119	61	26	32	1	0	2
24	26	4	31	23	52	49	25

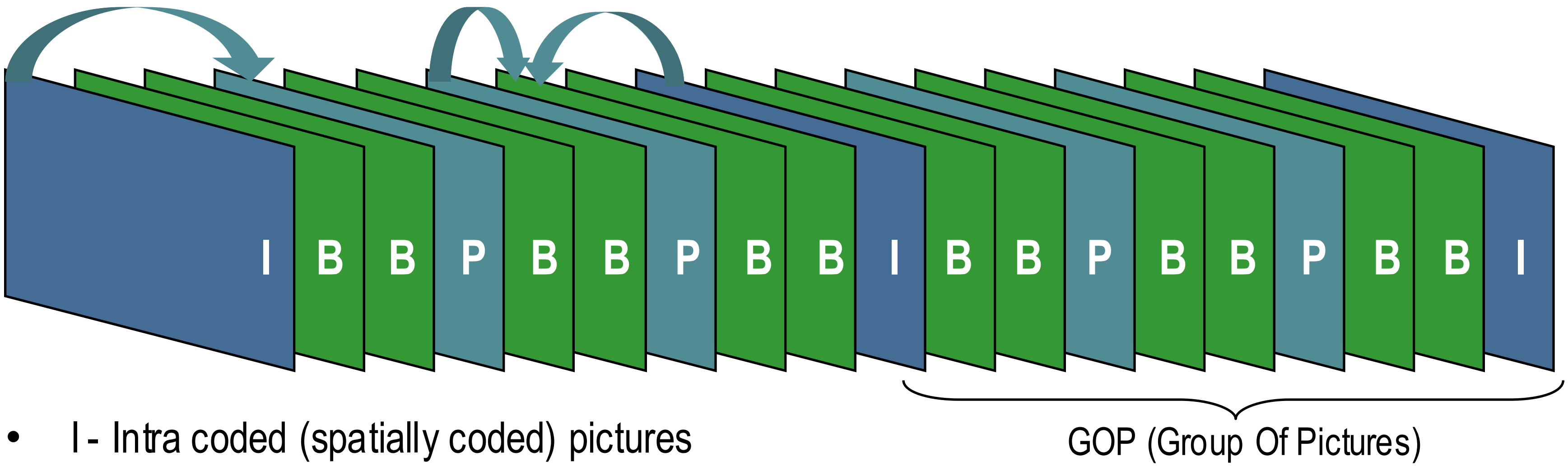
Output DCT coefficients

5



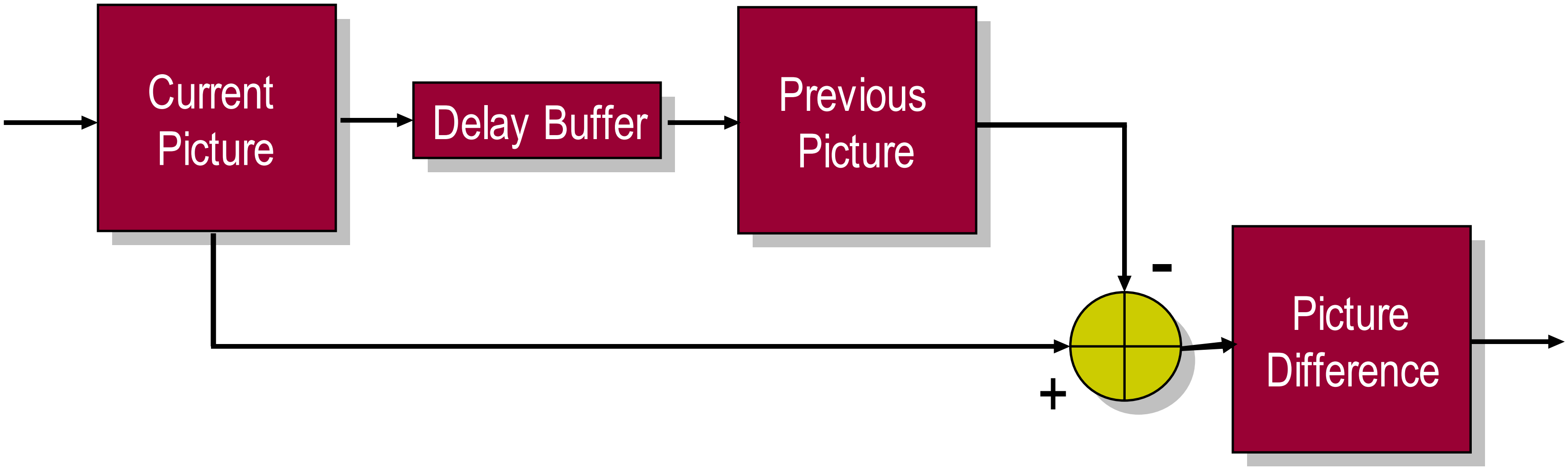
Convert to frequency components (DCT)

Temporal Redundancy



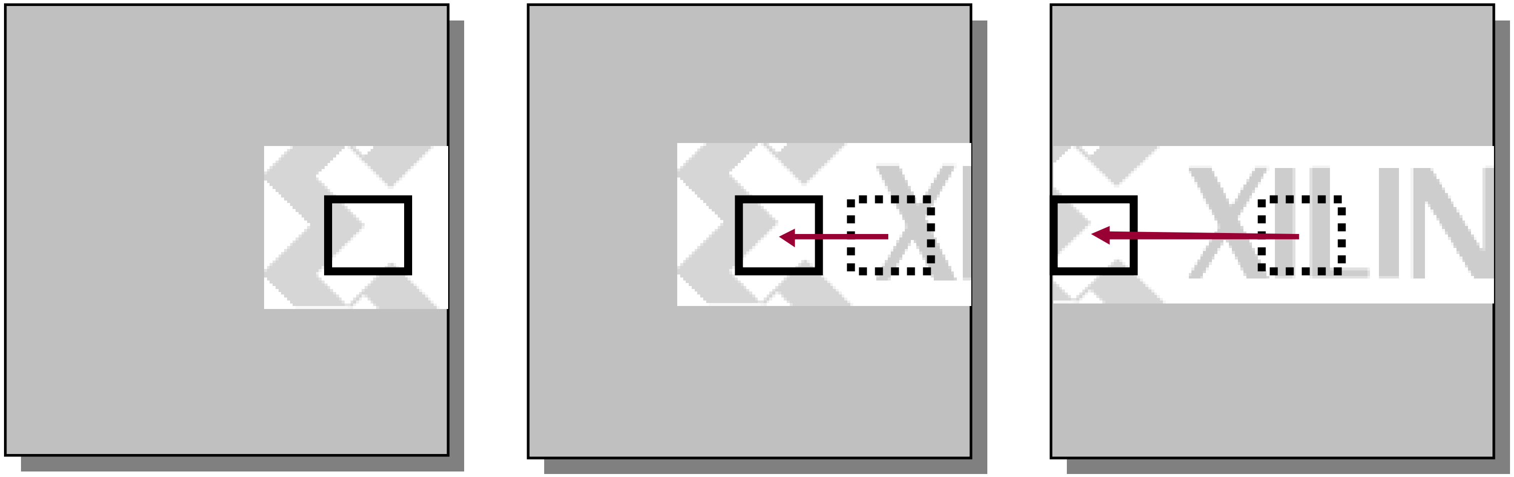
- I - Intra coded (spatially coded) pictures
  - Forms the anchor for a GOP
- P - Forward Predicted pictures
  - Predicted from previous I or P pictures
  - P picture made up of vectors showing where to get pixel data from in previous pictures and/or values that must be added to previous picture to get current pixel value
- B - Bi-directional Predicted pictures
  - Predicted from previous or later I or P pictures (never from other B pictures)
  - Made up of vectors showing where to get pixel data from in previous pictures

Picture Difference



- Difference between successive pictures easy to calculate using subtractor
- Picture difference can also be spatially compressed
  - DCT, VLC, RLE etc. as before

Motion Estimation



Part of image that is common between frames but has moved

Original position of picture segment in previous frame

Mption vector sent with any prediction errors

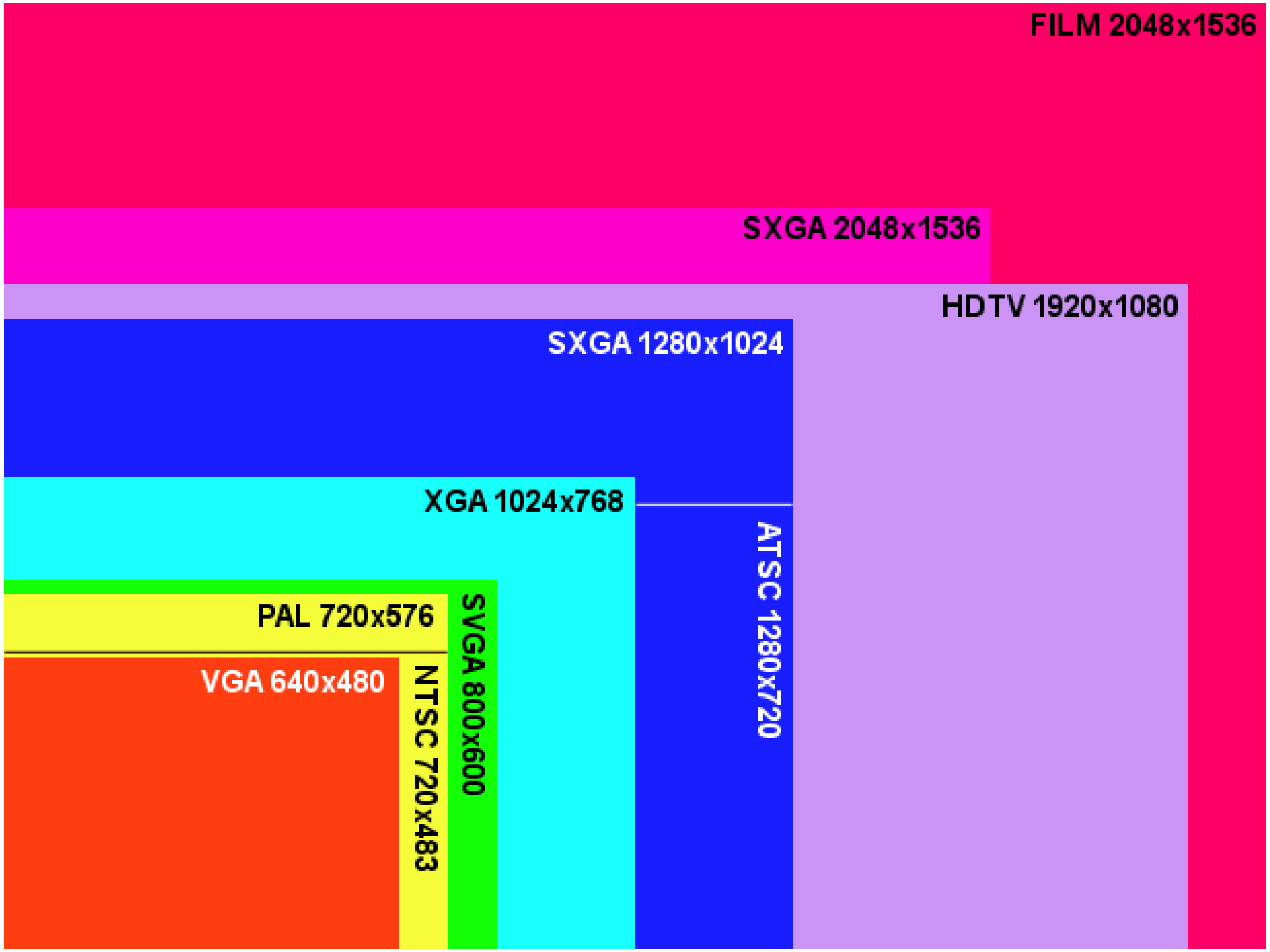
- Estimation predicts next picture by shifting data from previous picture along a calculated motion vector
- In encoder, predicted picture is compared to actual picture and any prediction errors calculated
- Transmitting motion vectors and prediction errors takes much less bandwidth than coding entire picture

Processing Challenges

- Variable resolutions and refresh rates
- Variable scan mode characteristics
- High performance requirements
- Variable file encoding formats
- Variable content security formats

Video System Challenges

A Range of Resolutions



Picture Courtesy: Snell & Wilcox

Video System Challenges
Video Scanning Formats

Definition	Lines/Frame	Pixels/Line	Aspect Ratios	Frame Rates
High (HD)	1080	1920	16:9	23.976p, 24p, 29.97p, 29.97i, 30p, 30i
High (HD)	720	1280	16:9	23.976p, 24p, 29.97p 30p, 59.94p, 60p
Standard (SD)	480	704	4:3, 16:9	23.976p, 24p, 29.97p, 29.97i, 30p, 30i, 59.94p, 60p
Standard (SD)	480	640	16:9	23.976p, 24p, 29.97p, 29.97i, 30p, 30i, 59.94p, 60p

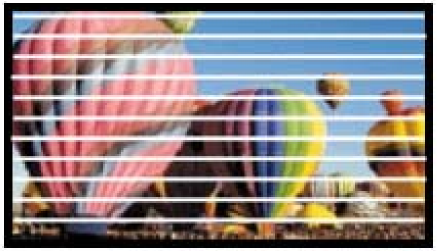
- Table III well known in the broadcast industry
- List of standard formats from ATSC A.53 DTV standard
- 36 different formats available!
- Doesn't take into account line doubling etc.

Video System Challenges

Interlace/Progressive Scan

Interlace

First all odd lines scanned (1/60sec)

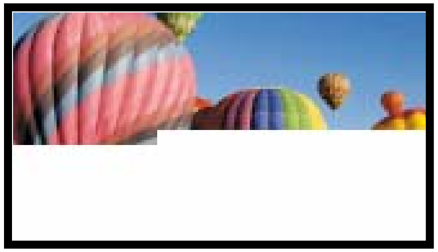


then all even lines (1/60sec)

presenting a full picture (1/30sec)

Progressive

All lines scanned in single pass



presenting a full picture (1/60sec)

Experimenting with Tradeoffs

- It would be nice to have a fully flexible device to use for video processing designs
  - Allows changing of parameters like colour depth, bit accuracy (truncation)
  - Allows exploration of new compression techniques or acceleration of existing algorithms to improve throughput
  - Supports various frame rates and resolutions
  - Implements a wide range of new or existing filters for enhancement or noise reduction

Welcome to Xilinx FPGAs

- FPGAs are a key enabling technology for digital video processing
- Allow experimentation for prototypes leading to differentiation for production
- ***And*** still enable a higher level of system integration with support for:
  - video interfaces, LAN/WAN technologies, other DSP, simple glue, memory control and state machines, backplane protocols…… the list is only limited by the imagination

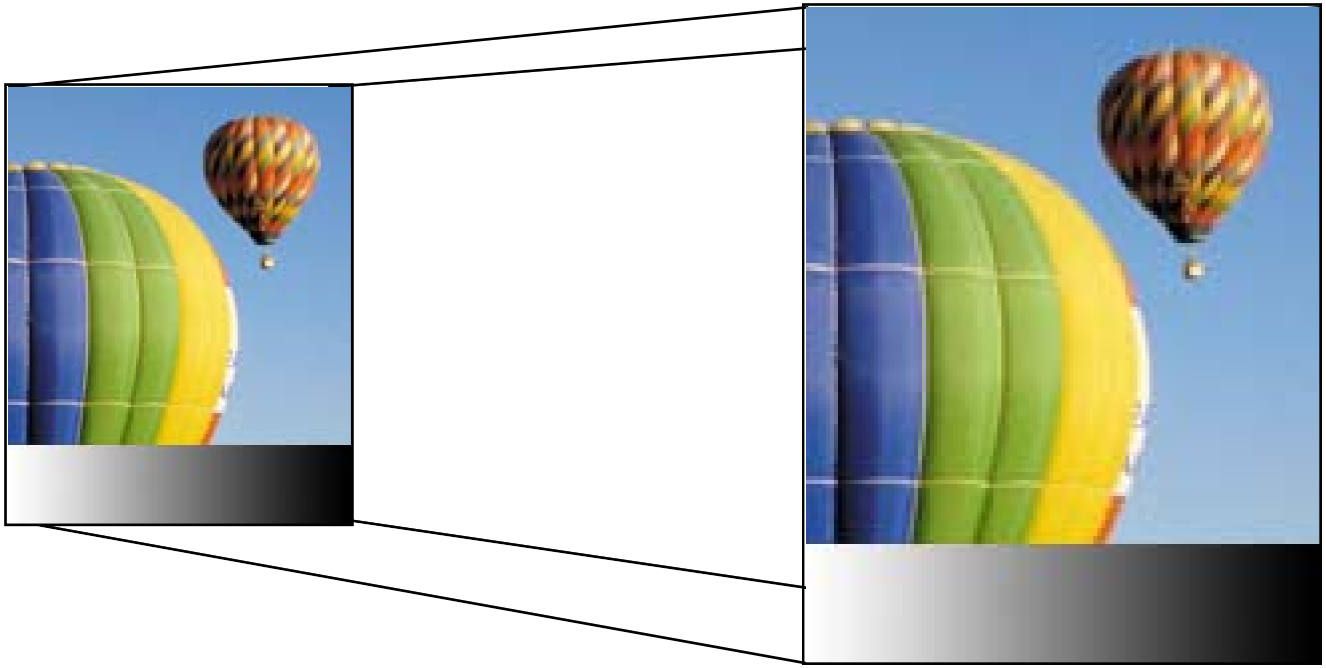
Basic Image and Video Processing

Image Processing Functions

- Pixel processing
  - Scaling
  - Rotation
  - Color/Gamma correction
  - Brightness
  - More colors through dithering
- Frame buffer processing
  - Contrast enhancement
  - Shadow enhancement
  - Sharpness enhancement
  - Chroma key composition
  - Graphic overlay

Basic Image Processing
Scaling

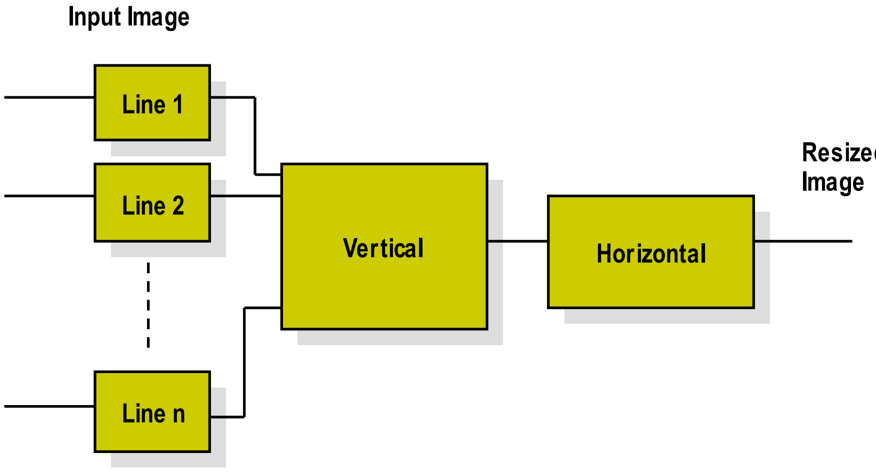
- Fractionally enlarges the incoming data stream as necessary to match the target display resolution
- Pixel processing on-the-fly during image input without frame buffer



Real-Time Image Resizing

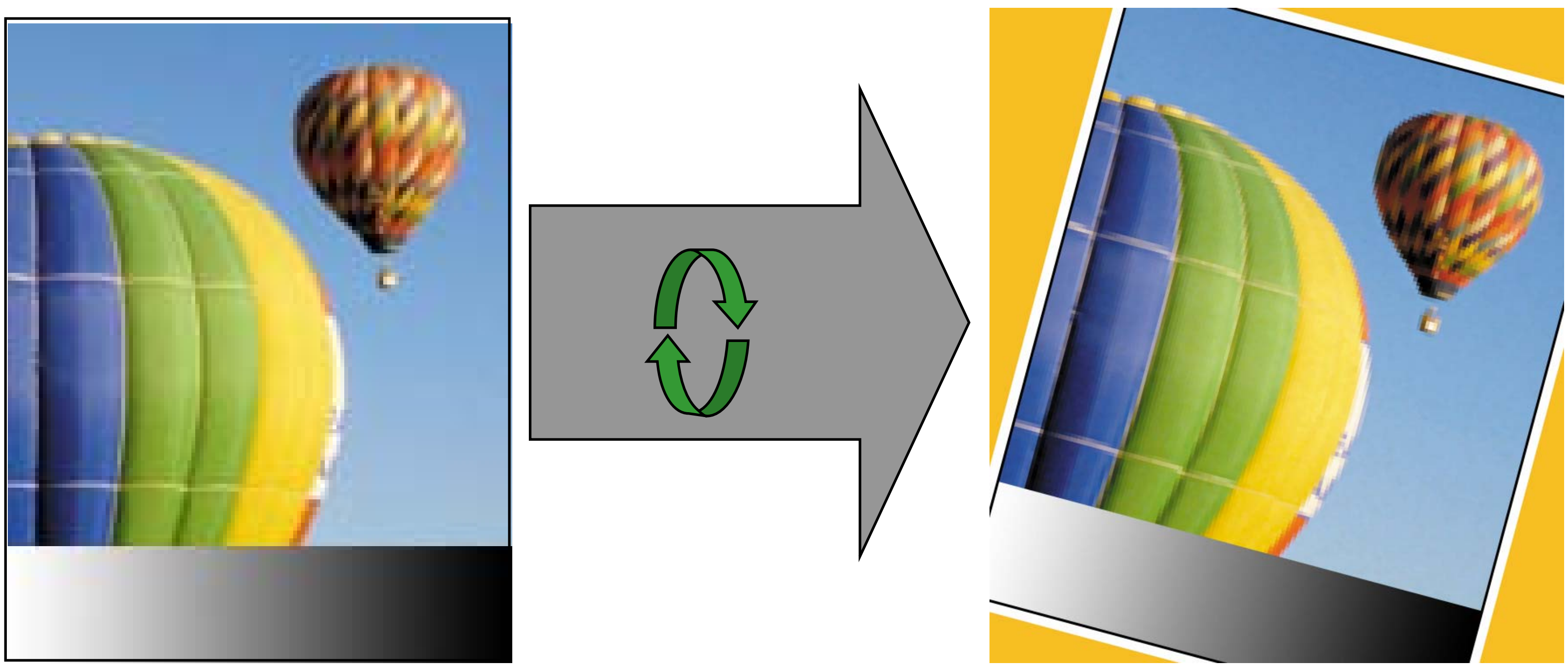
With Low Memory Requirements 2 Dimensional Architecture Upscaling by 2, Downscaling by 4

- Example: 512 x 512 x 8 60 f/s
  - Upscaling by 2, Downscaling by 4
  - 16 pixel resolution
  - 8 Block RAMs for Line Buffers and Coefficient Bank
  - 4 vertical multipliers
  - 4 horizontal multipliers
  - Adder trees
  - Control



Real-Time Image Rotation

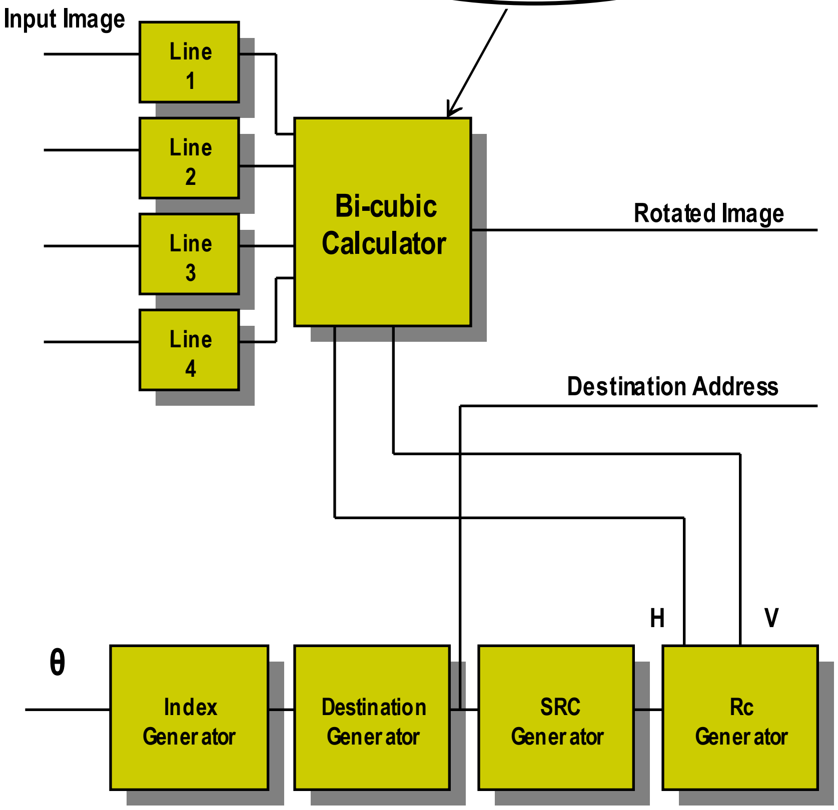
- Non-real time typically implemented using processor and frame store
- Real-time image rotation performed using bi-cubic function in FPGA
  - Pixels remapped to rotational co-ordinates



Real-Time Image Rotation

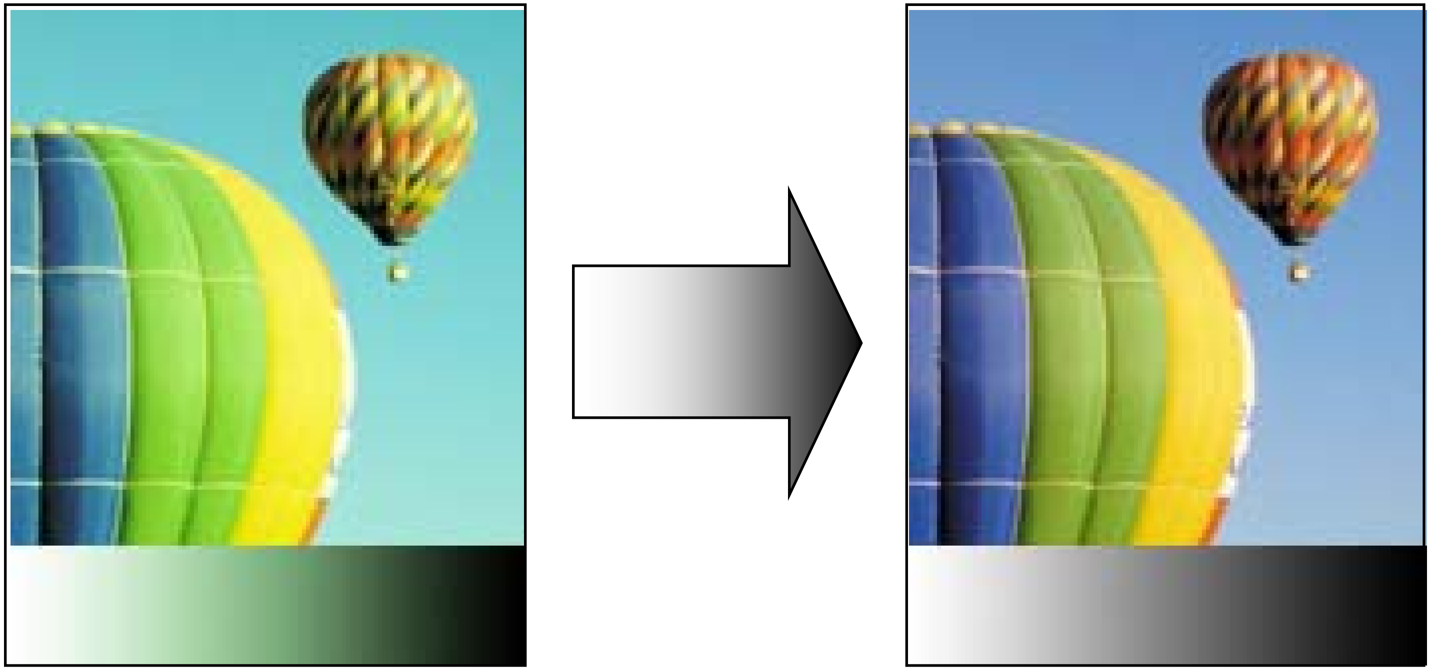
- Example:
  - Medical imaging system
    - 1024 x 1024 x 12 @ 30 f/s
    - 40 MHz Pixel Clock
    - 160 MHz Core Clock
  - Xilinx XC2S300E FPGA
    - 12 Block RAMs for line buffers
    - 2 Block RAMs for RC lookup tables
    - 5 multiplier pixel calculation
    - Sine/Cosine, 2 Block RAMs
    - Dx, Dy calculation
    - Sx, Sy calculation
    - Control

$$S_x = Dx\cos(\theta) + D_y\sin(\theta)$$
$$S_y = -Dx\sin(\theta) + D_y\cos(\theta)$$



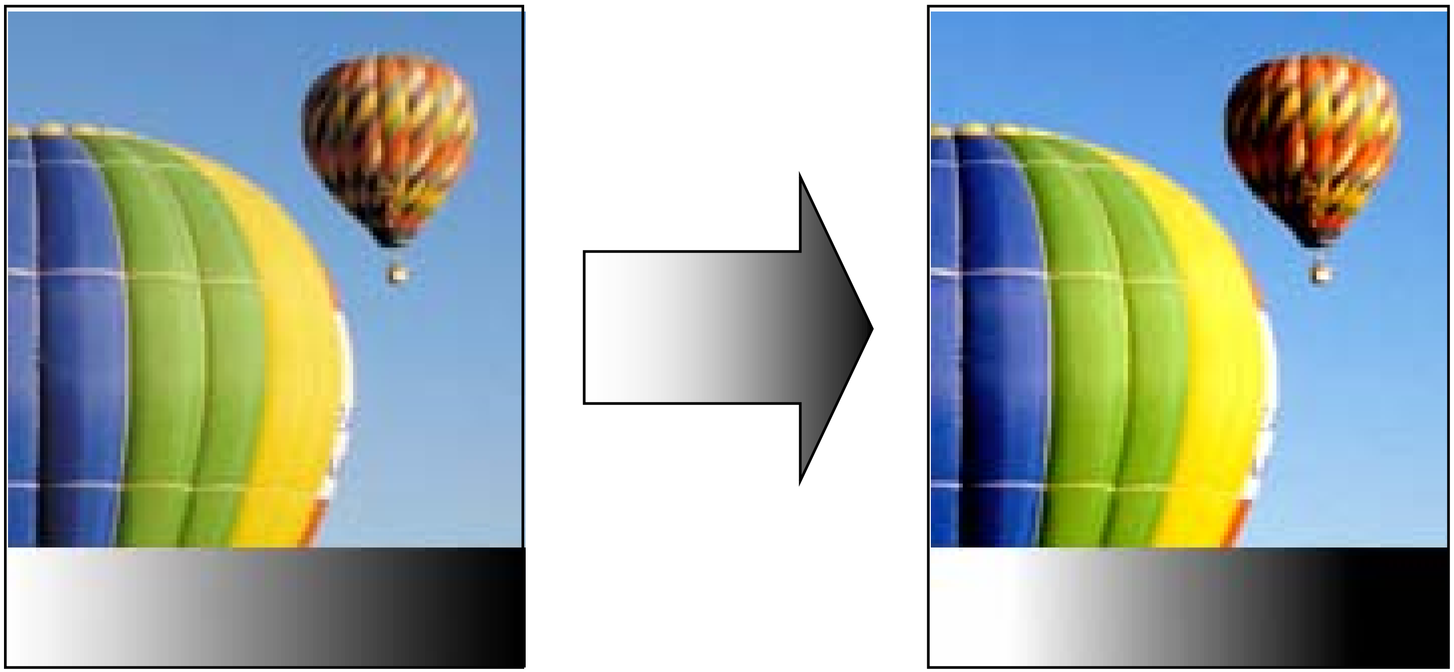
Basic Image Processing
Colour/Gamma Correction

- Adjusts RGB intensities through correction tables
- Required to account for technology specific RGB characteristics (CRT vs. LCD vs. PDP etc.)



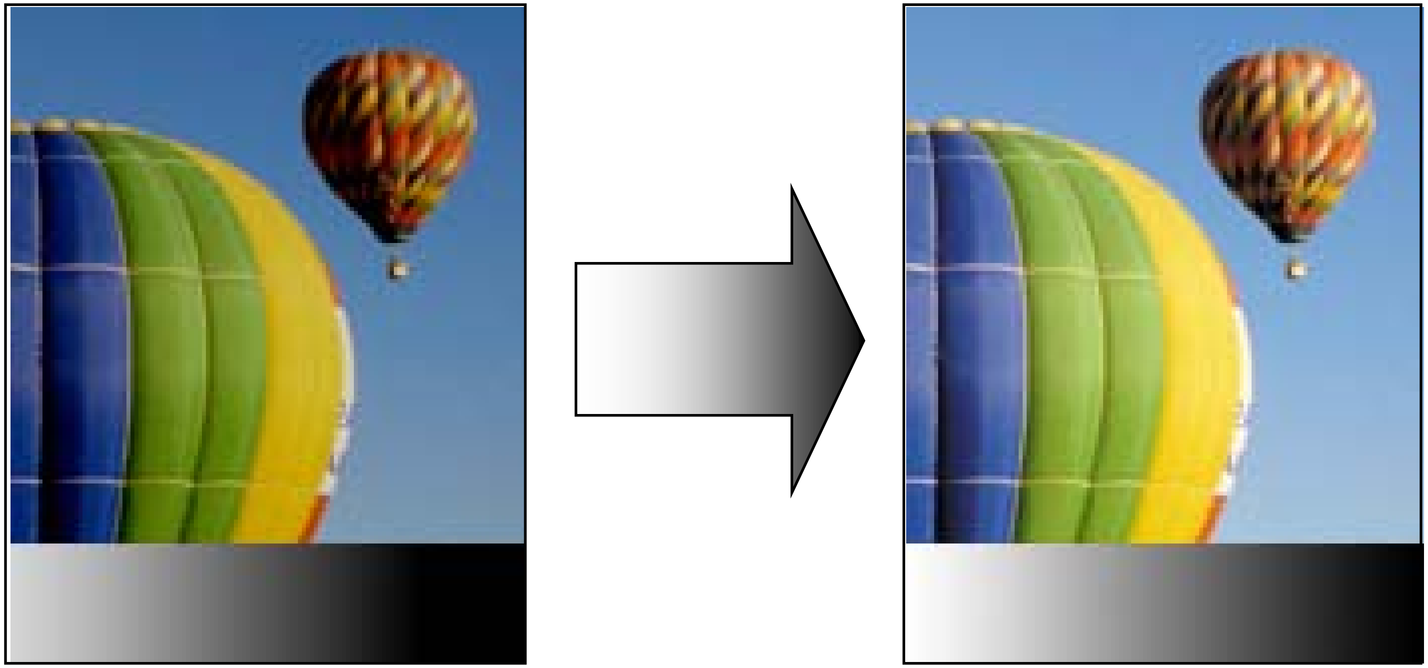
Basic Image Processing
Brightness

- Increases the RGB intensity to the viewer's taste



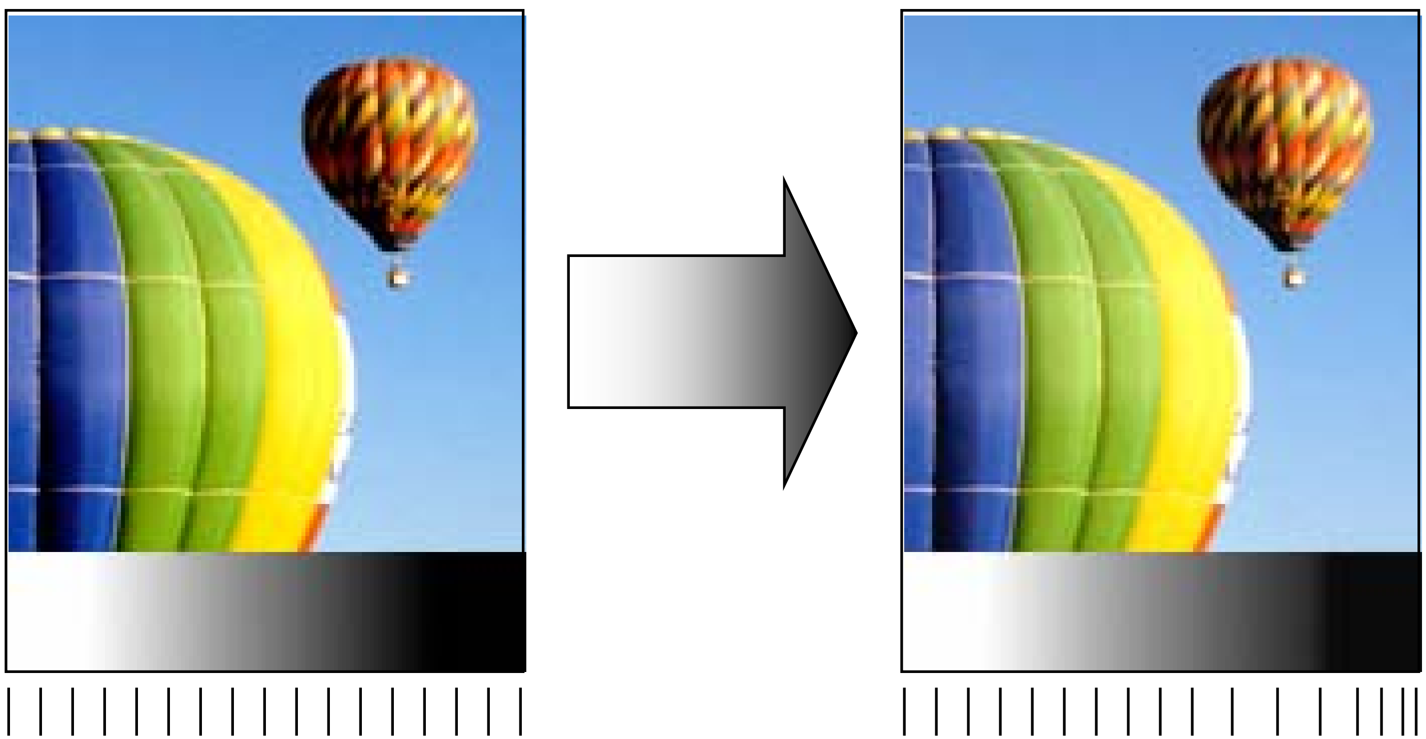
Advanced Image Processing
Contrast Enhancement

- Adjusts RGB intensities to control the degree of difference between light and dark image areas



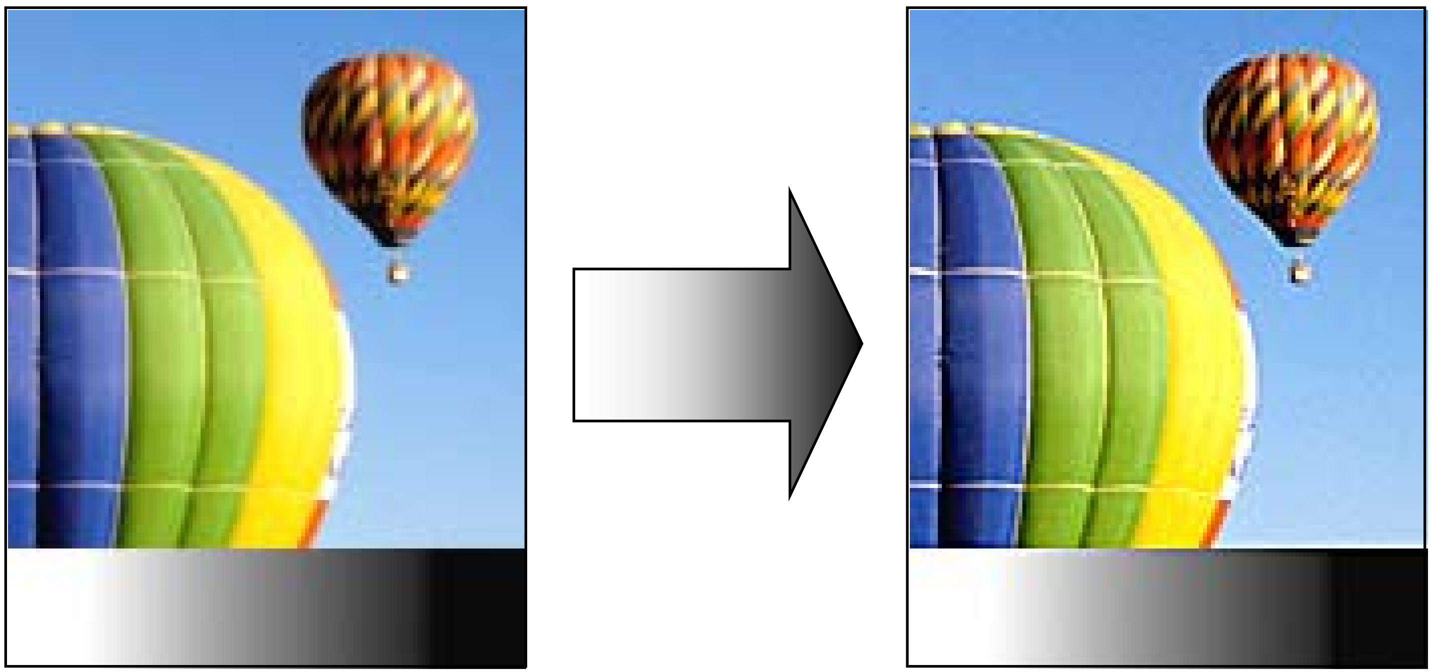
Advanced Image Processing
Shadow Enhancement

- Selectively adjusts RGB intensities in order to lighten dark grayscale regions

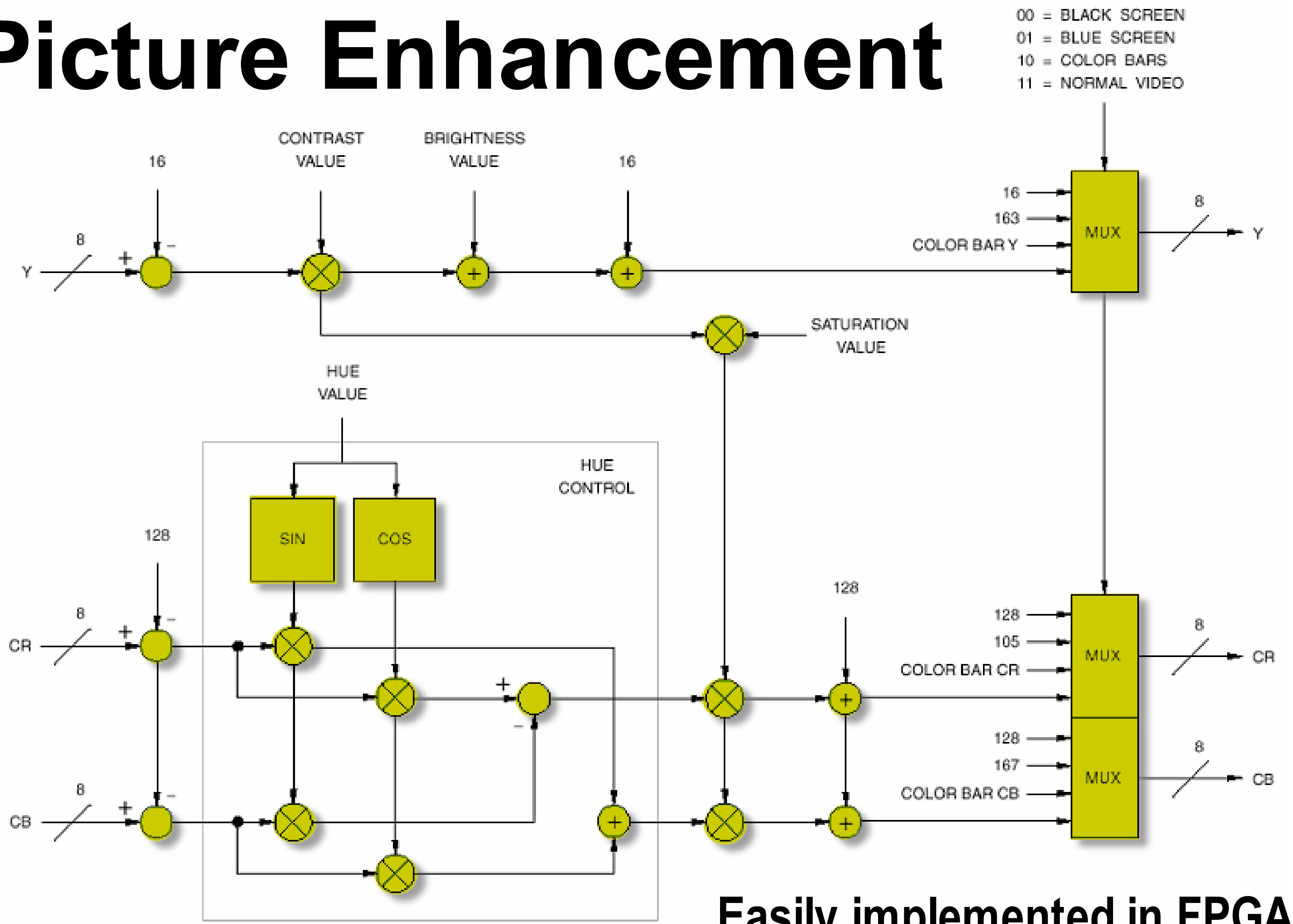


Advanced Image Processing
Sharpness Enhancement

- Adjusts RGB intensities to sharpen the transition between adjacent color regions



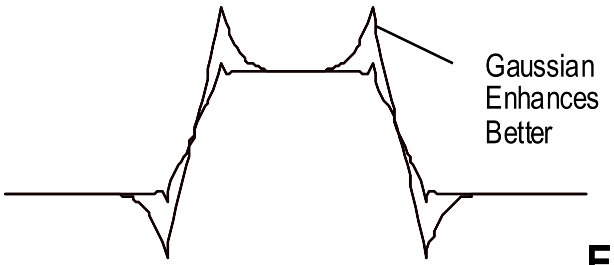
Picture Enhancement



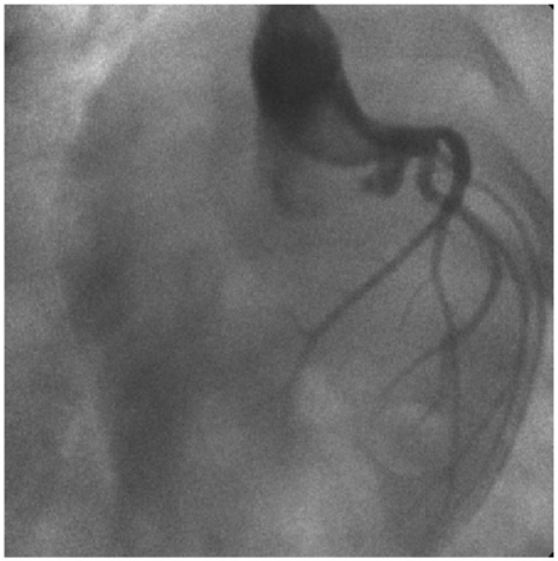
Easily implemented in FPGA

Spatial Enhancement Illustration
Medical Image Enhancement

Enhanced 1D Signal



Input image

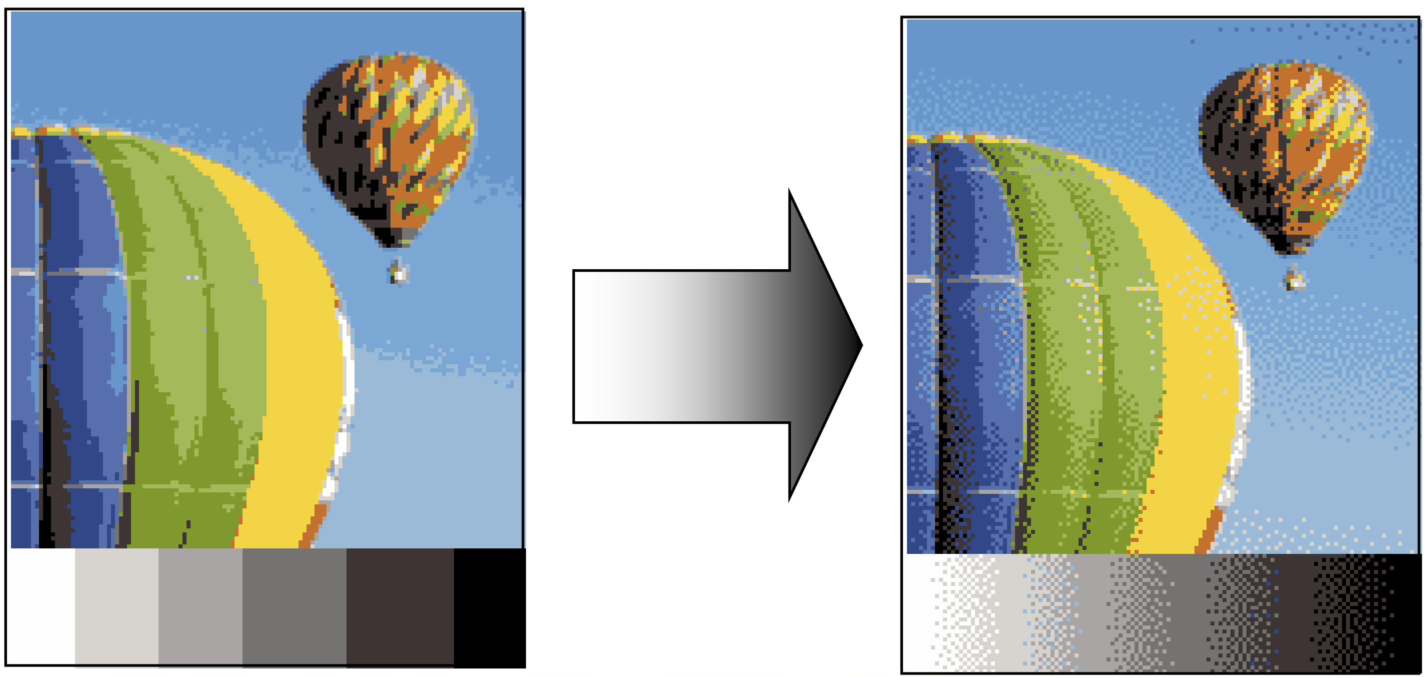


Enhanced image

Basic Image Processing

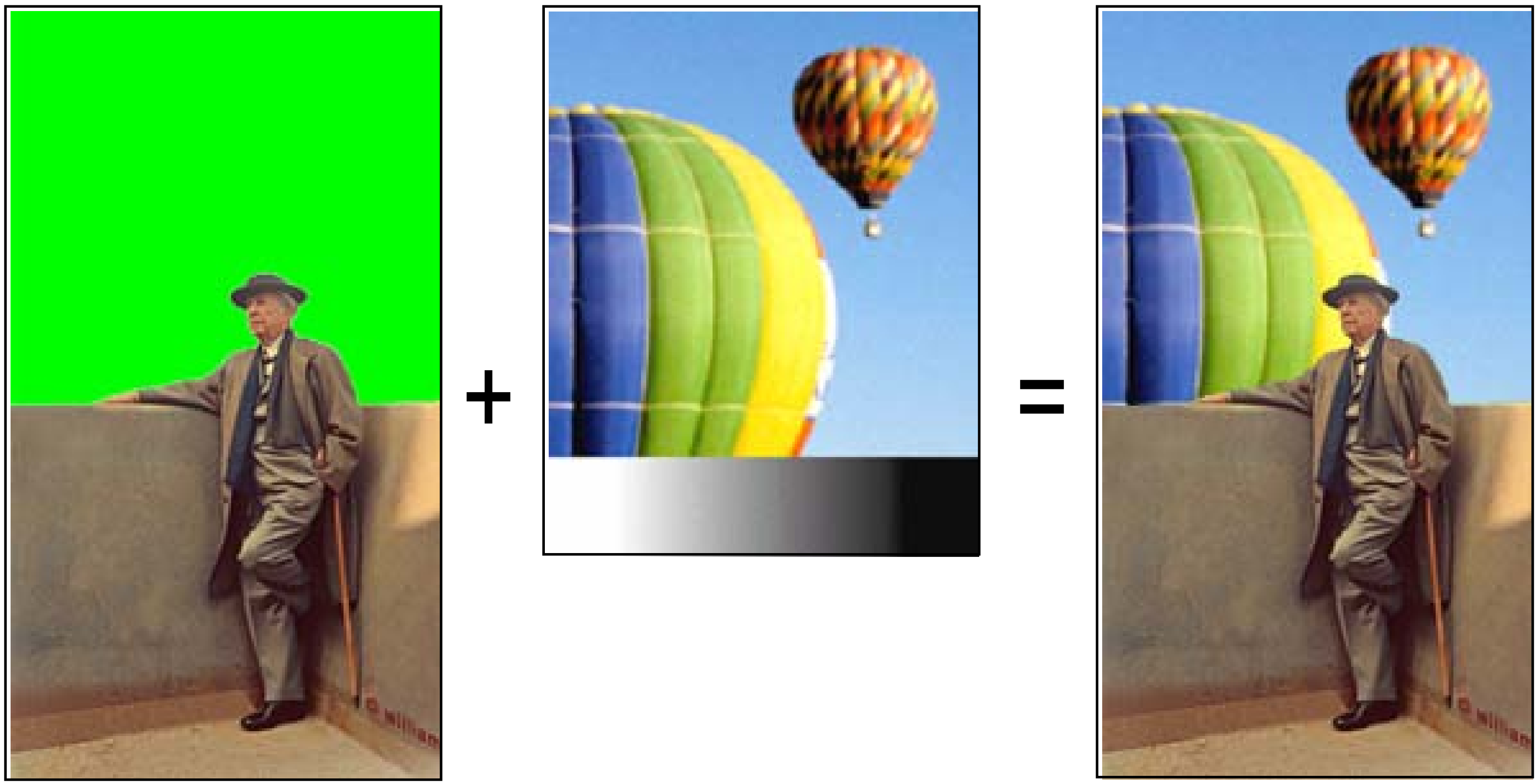
More Colors Through Dithering

- Smoothes out color transitions/banding in bit-depth limited displays
  - Achieve full color with sub 24-bit display technologies like LCD and PDP
- Generates patterns of pixels which the eye blends together into colors the display cannot generate



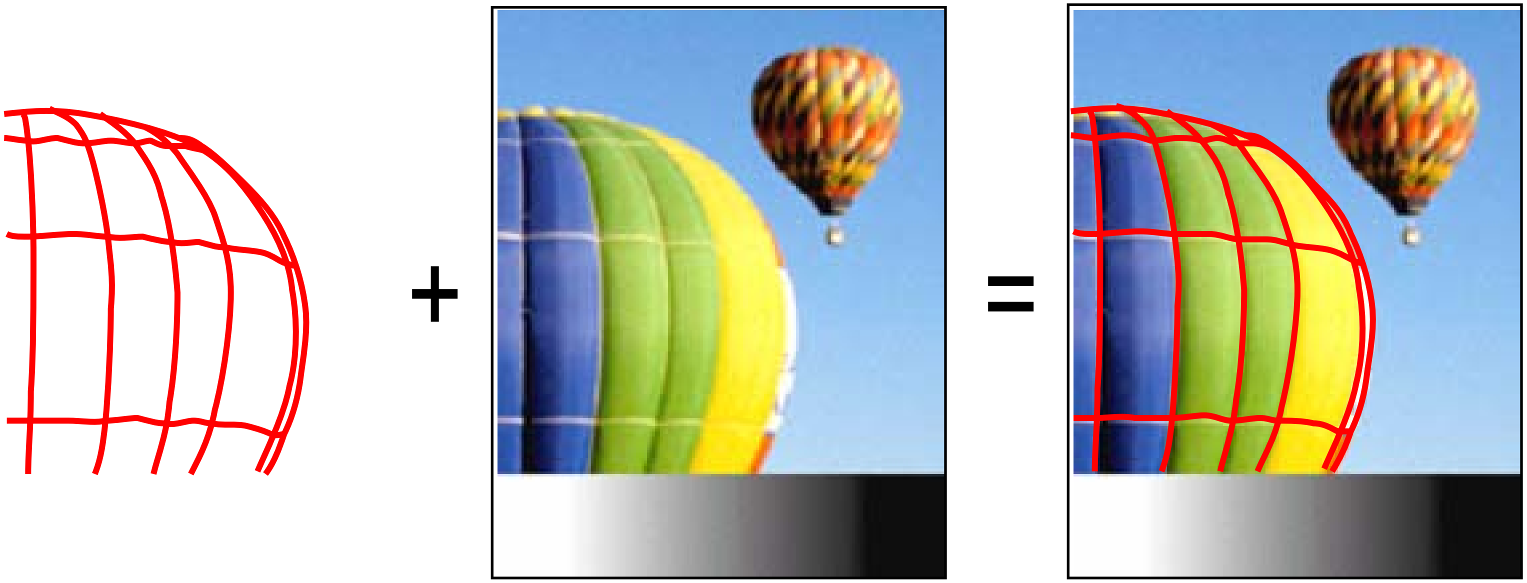
Advanced Image Processing
Chroma Keyed Compositing

- Composites 2 images together, replacing a specific RGB value in one image (chroma) with the data pixel from the other



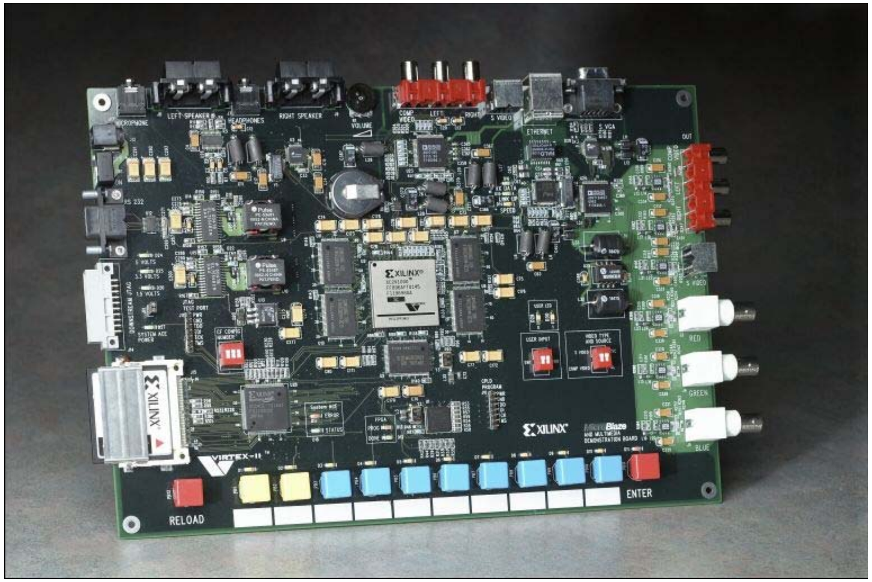
Advanced Image Processing

Graphic Overlay



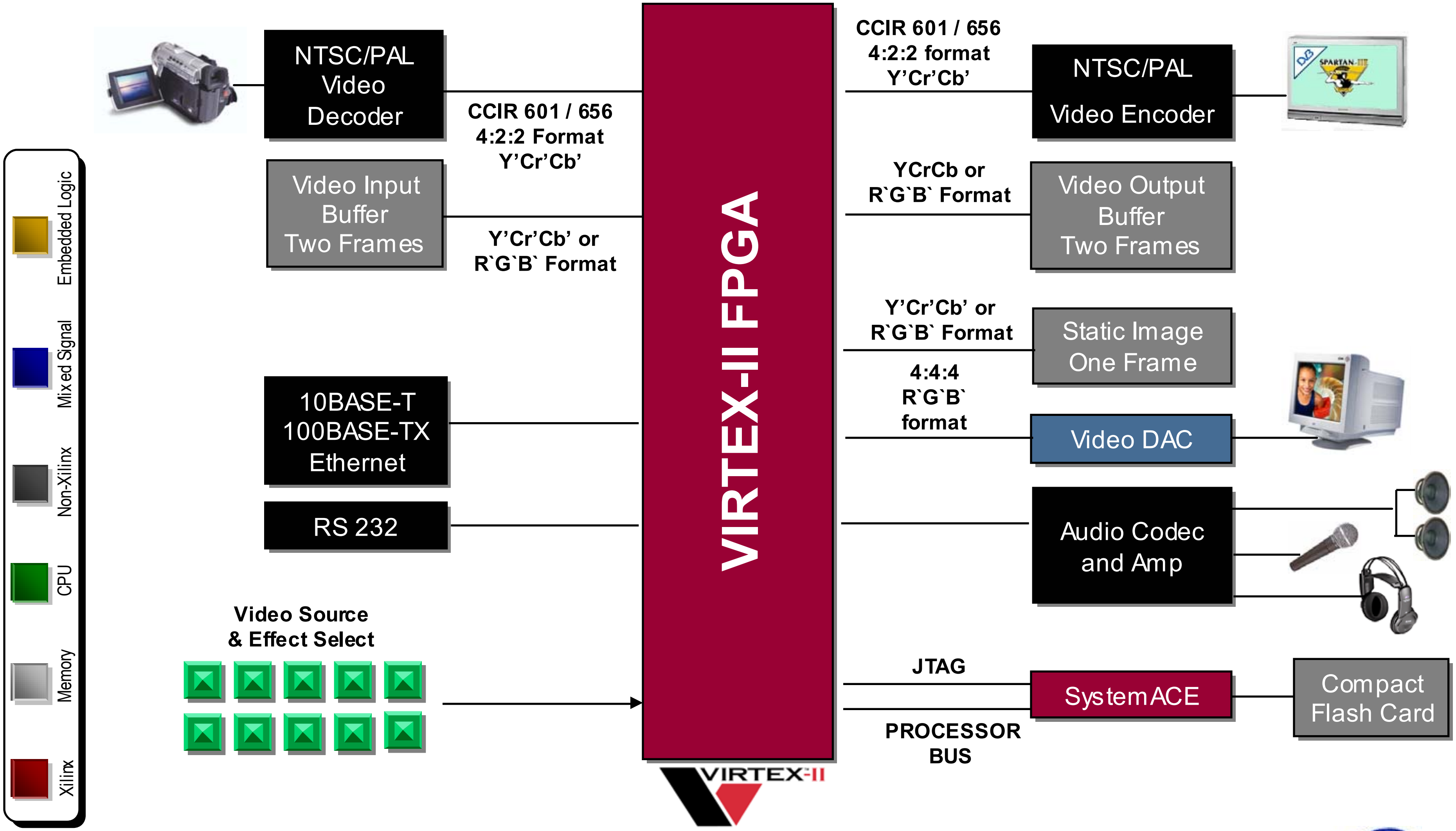
Mixer Functions in Virtex-II Pro FPGA

- Fade
- Mix
- Wipe
- Chromakey
- Logo Insertion

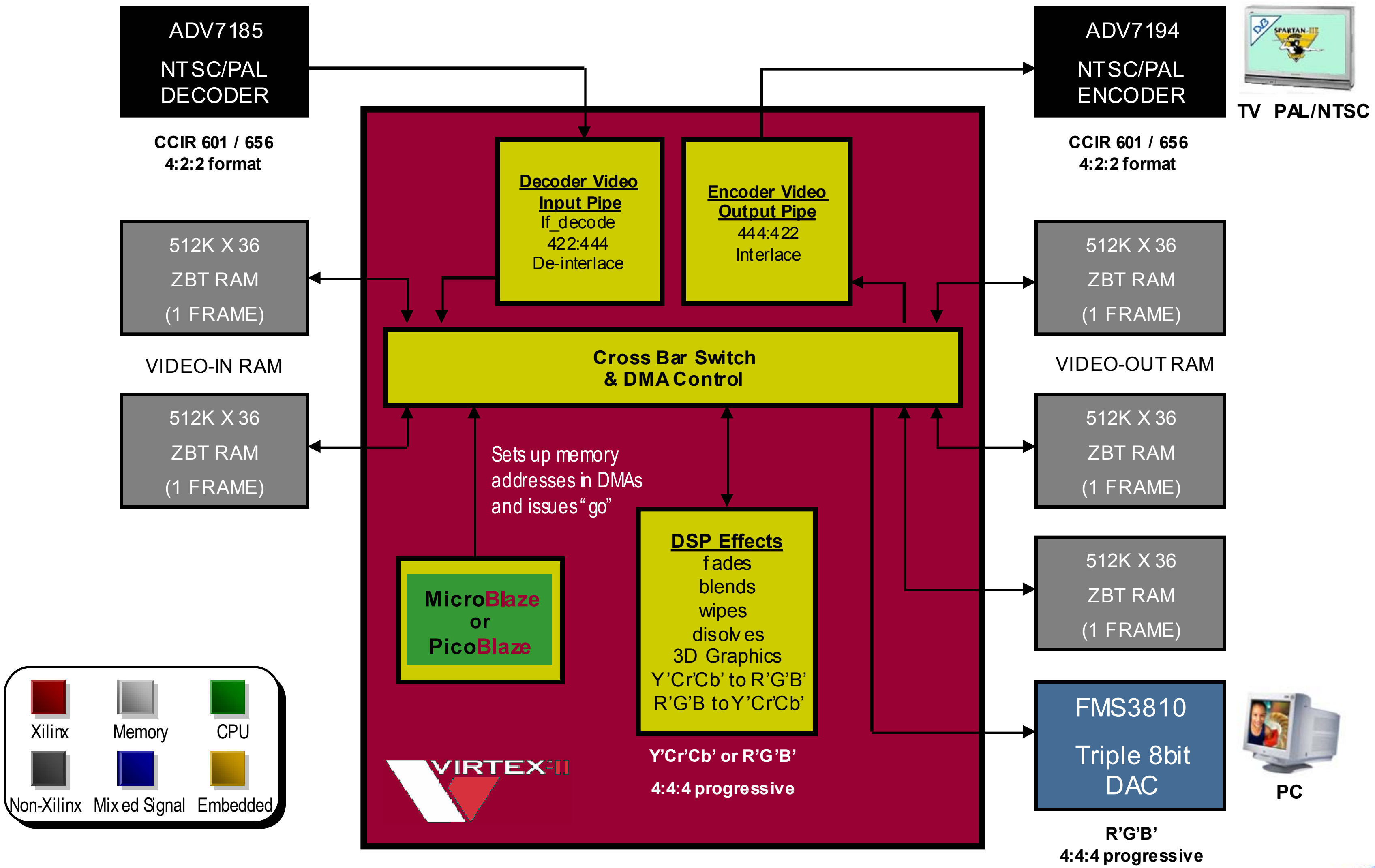


- MicroBlaze & Multimedia Demonstration Board demonstrates many mixer capabilities in FPGA

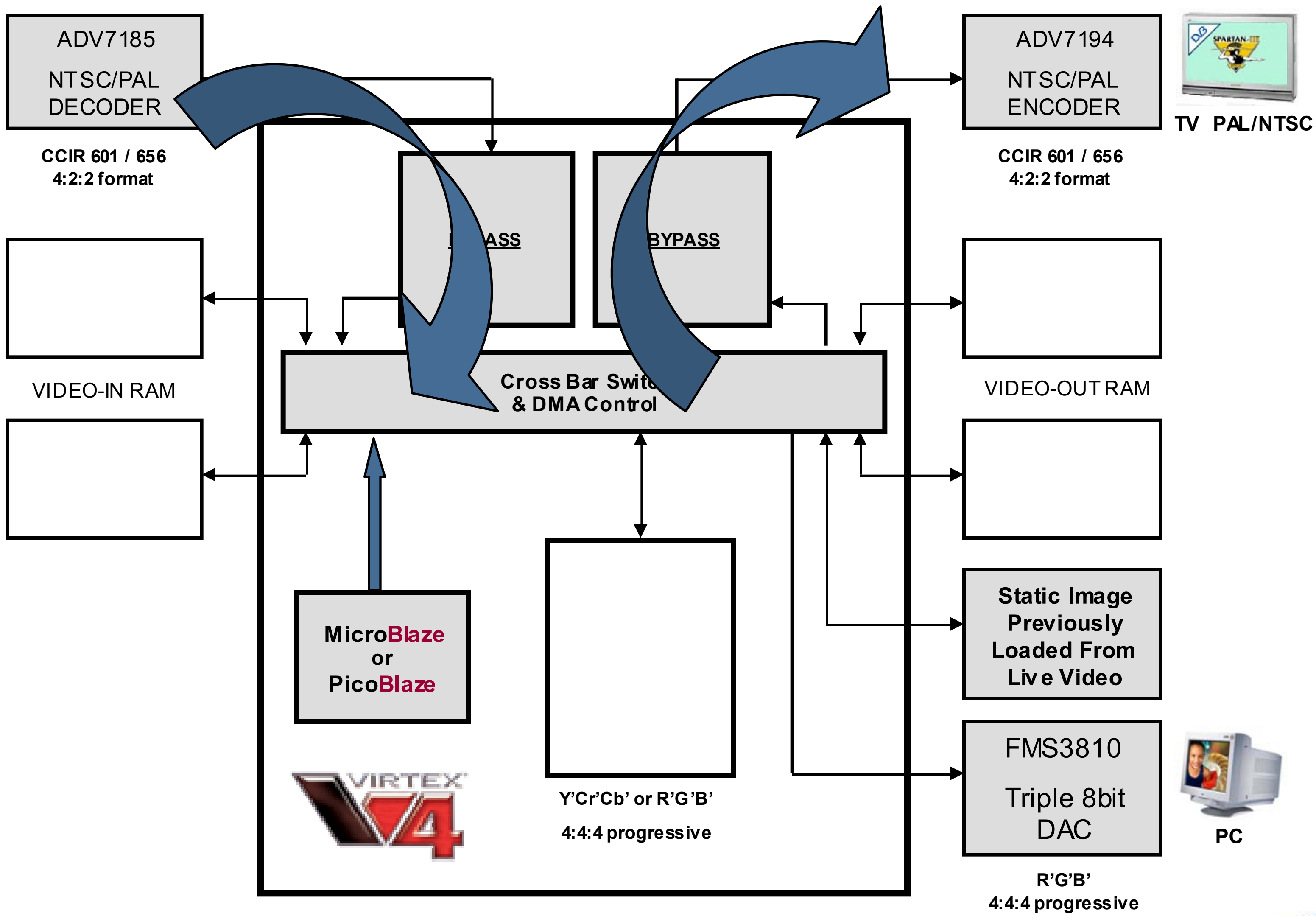
Multimedia Demo Board



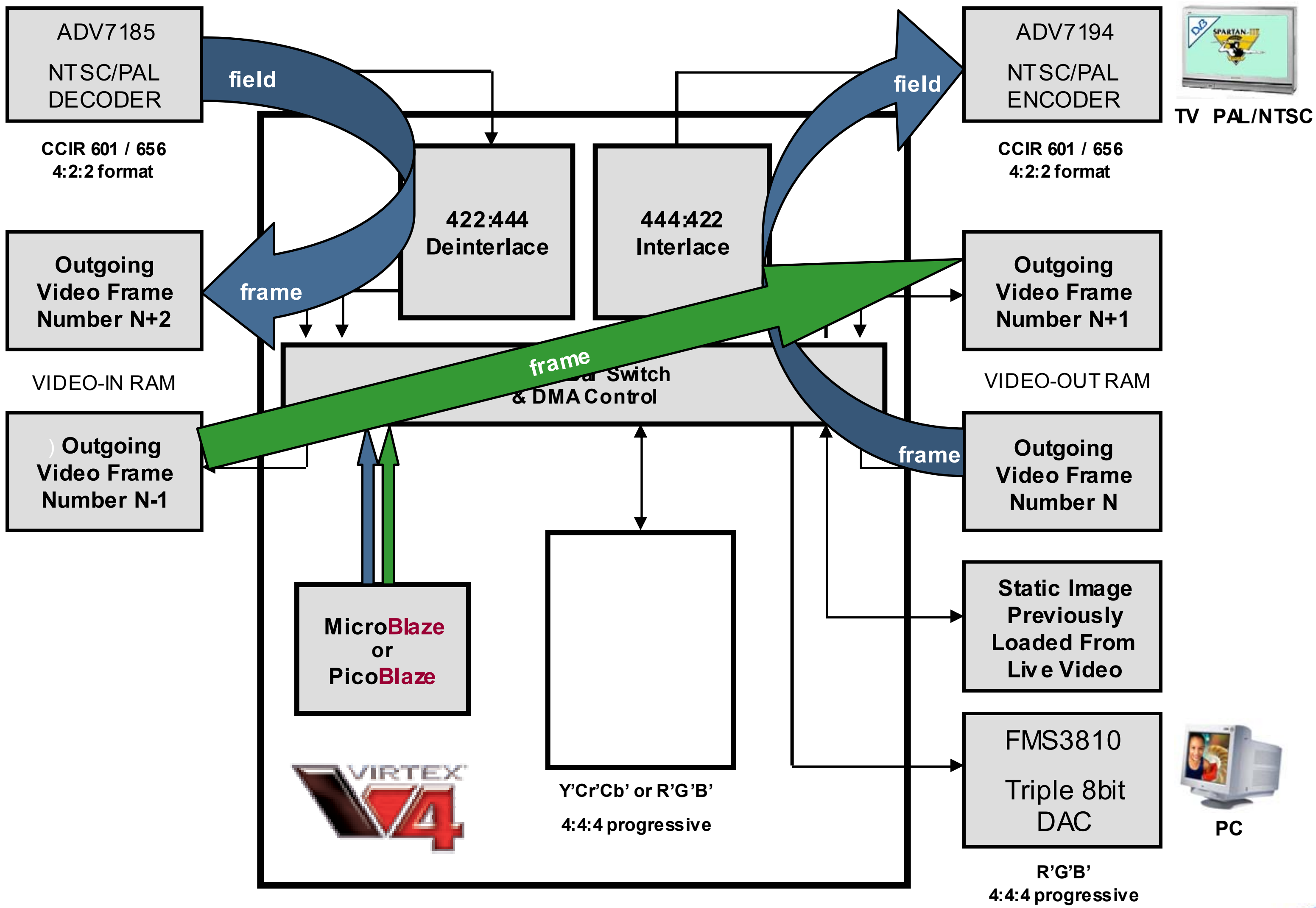
Multimedia Platform



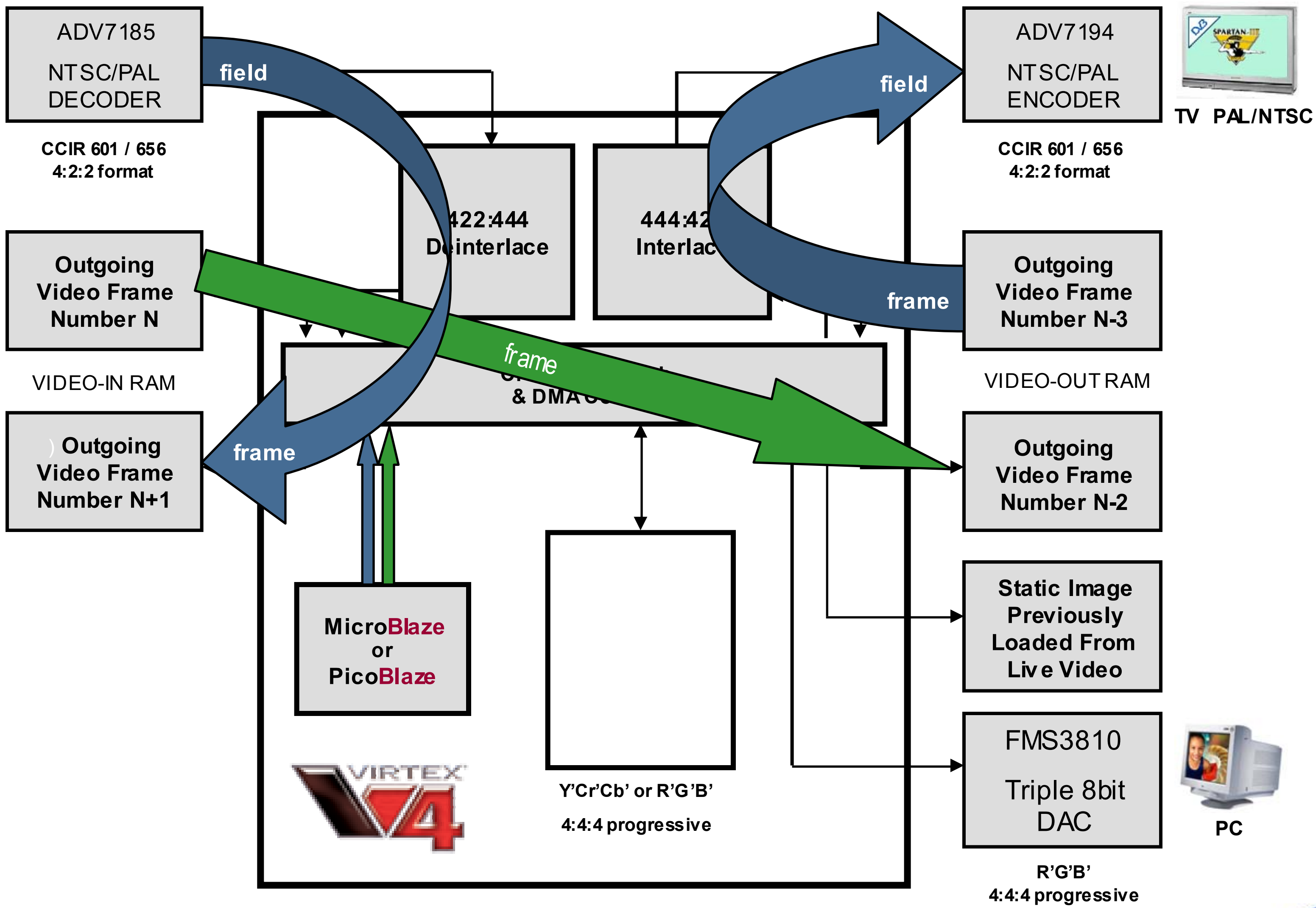
Video Bypass



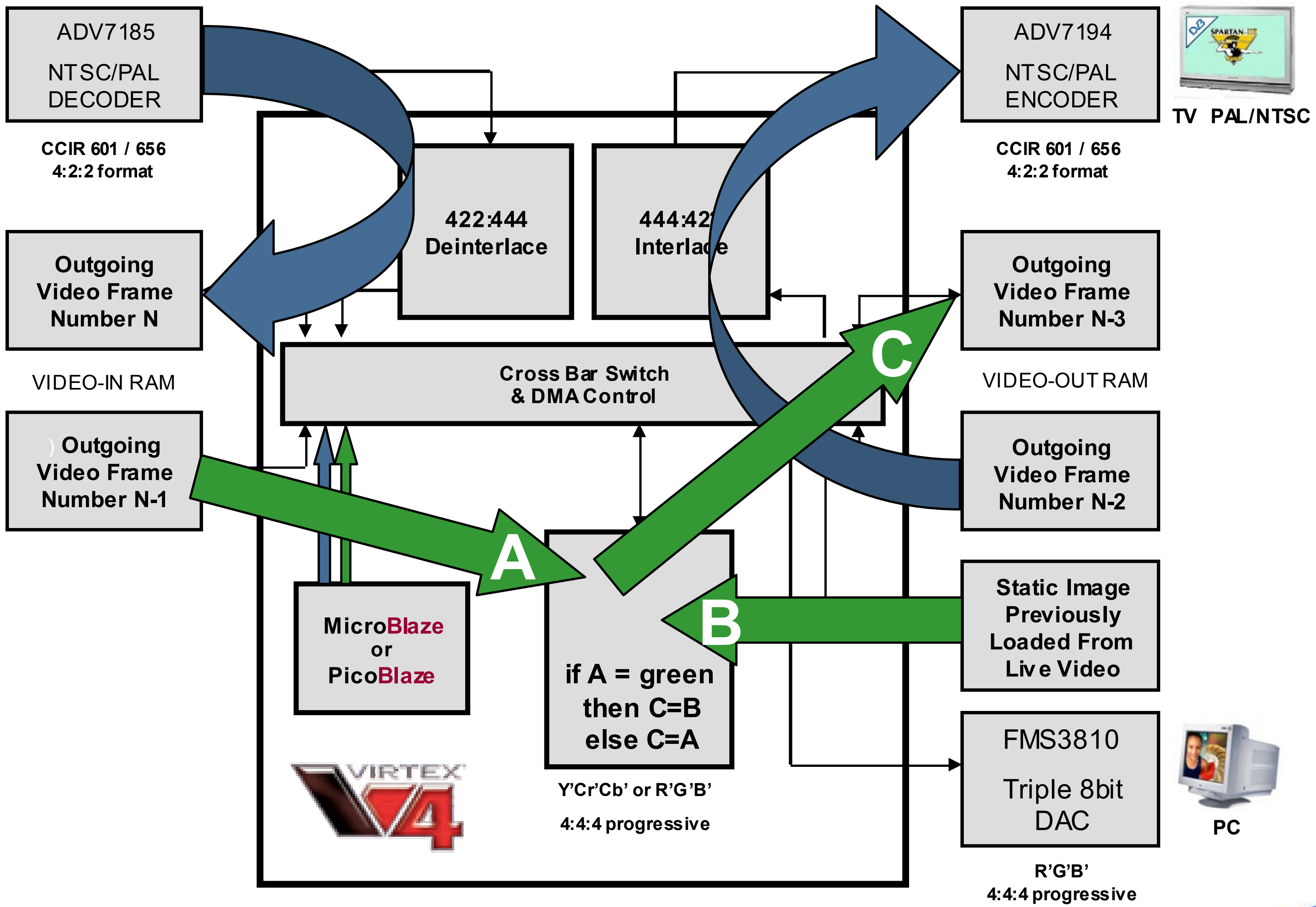
Video Ping-Pong (1)



Video Ping-Pong (2)



Video Chromakey



Noise Reduction & Filtering

FIR Filters for Xilinx FPGAS

- Most image filtering can be done based on two-dimensional FIR filters
  - Programmability allows experimentation with different coefficients, filter windows etc to get the best picture

IP Core or Reference Design		Provider	
XAPP219 Transposed Form FIR Filters	Reference DESIGN	Xilinx Inc.	XILINX
MAC FIR	LogiCORE	Xilinx Inc.	XILINX
Serial Distributed Arithmetic FIR Filter	LogiCORE	Xilinx Inc.	XILINX
Parallel Distributed Arithmetic FIR Filter	Alliance CORE	Xilinx Inc.	XILINX
Distributed Arithmetic FIR Filter	Alliance CORE	Xilinx Inc.	XILINX

See http://www.xilinx.com/ipcenter for more details

Noise Removal

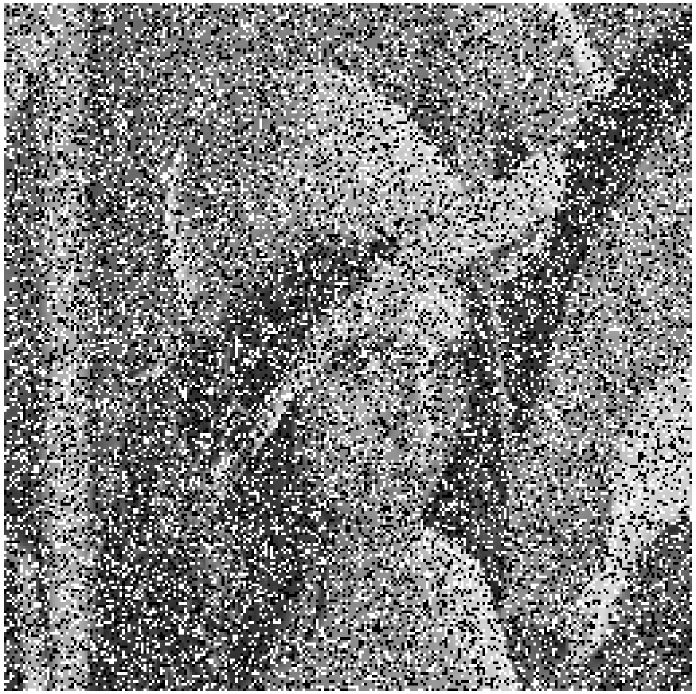
 

- Removal of impulsive noise
  - e.g. Scratches on film

Noise Removal

- Removal of Gaussian noise

Noise Removal

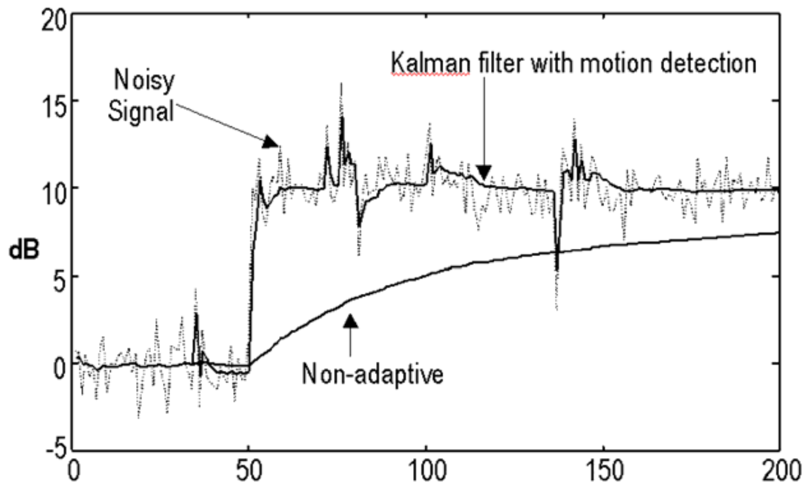


- Salt and Pepper noise removal

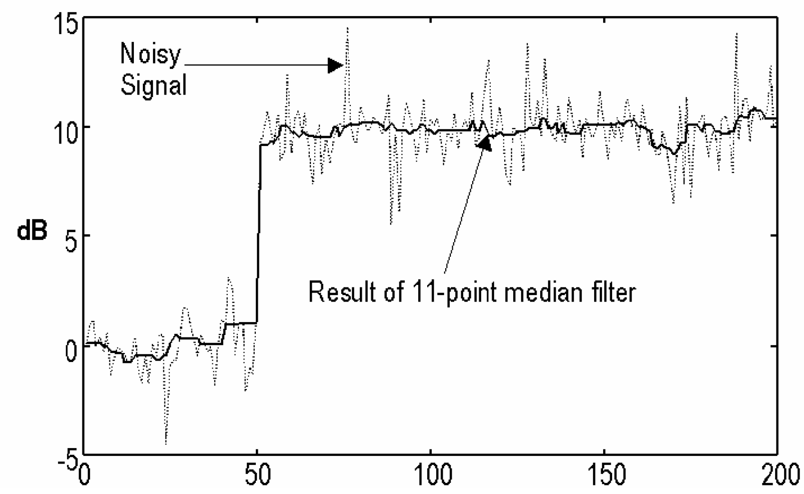
Noise Reduction

- Noise reduction using temporal filtering across multiple video frames

Kalman Filter



Median Filter



Temporal Filter

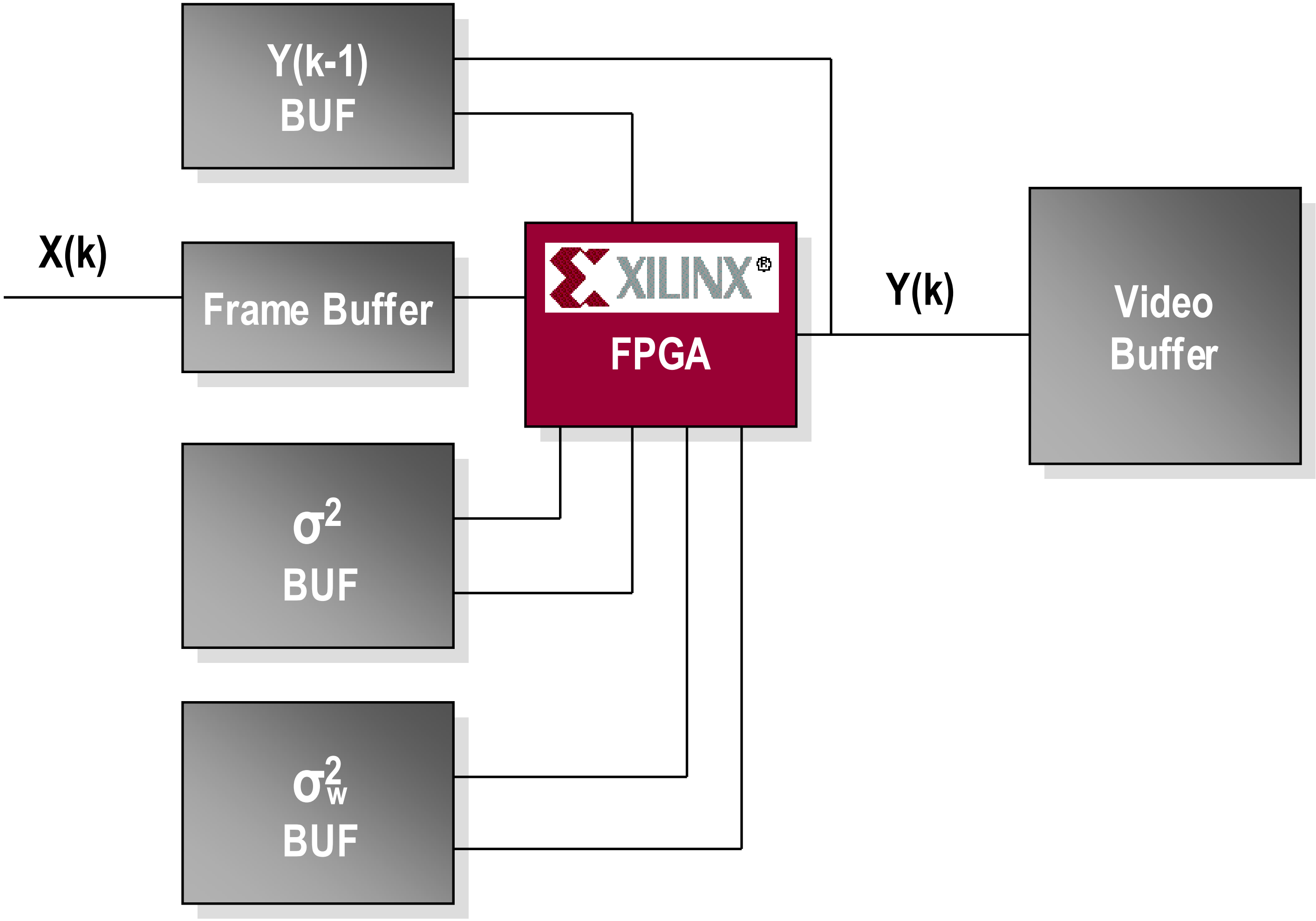
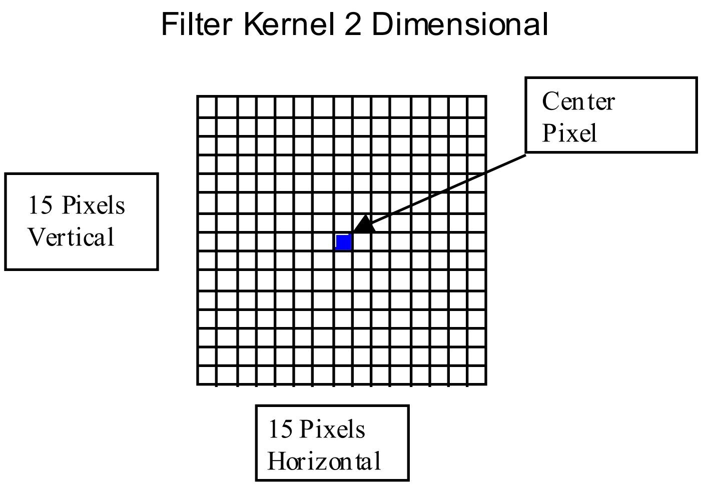
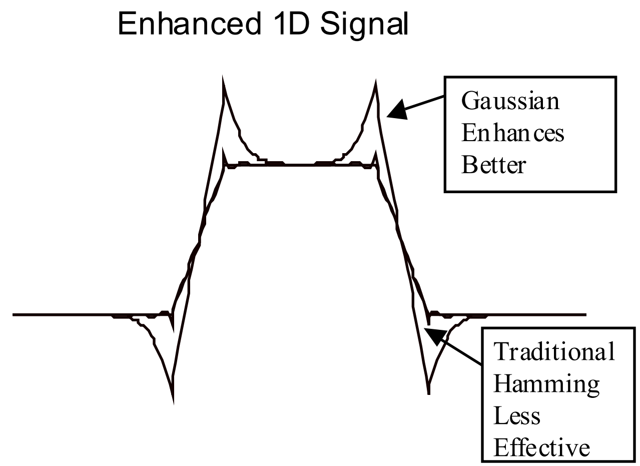


Image Enhancement Algorithms

- The list in endless, but these are a few examples of image enhancement algorithms
  - Spatial Filter Unsharp Masking
  - Digital Max-Detail
  - Laplacian of Gaussian Filter
  - Adaptive Histogram Equalization
  - Adaptive Kalman Temporal Filter
  - Non-Linear Median Filter
  - Non-Linear Fuzzy Filter
  - Wavelet Decomposition
- Or you may have a better version of your own
  - Is there an ASSP to support your idea?

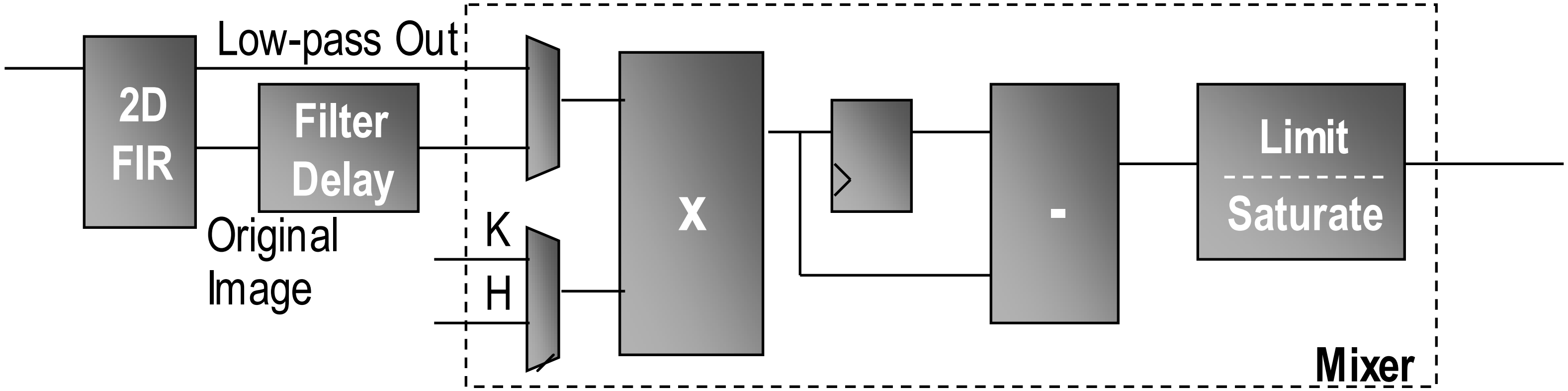
FPGAs for Spatial Enhancement



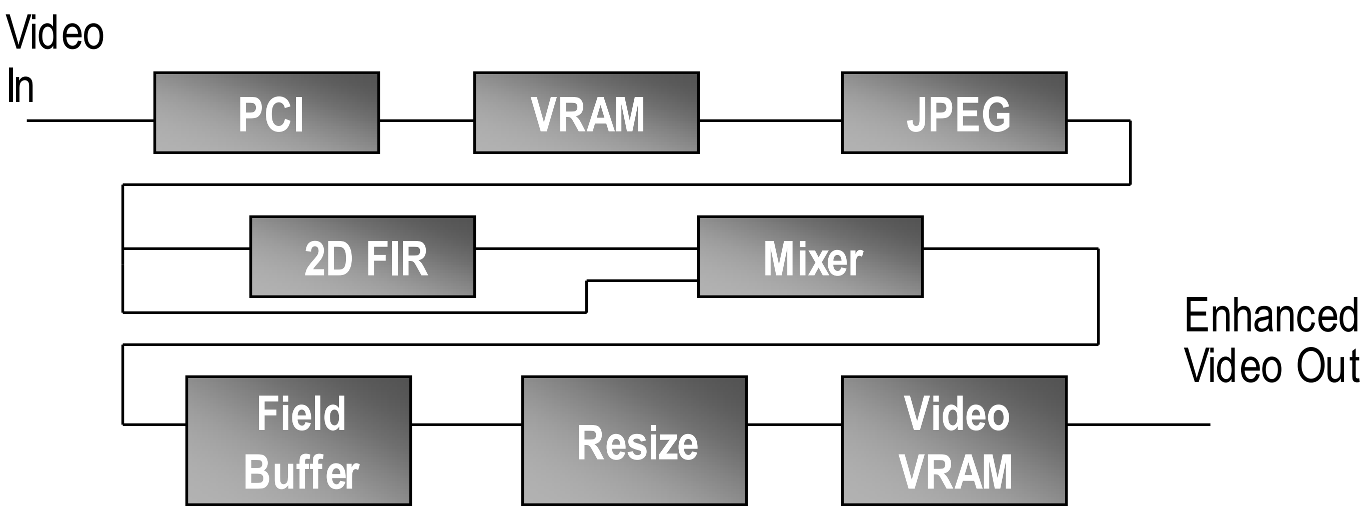


- User selectable kernels
- Experiment to find the filter that gives the best results (on-the-fly)
- Parameterisable filter coefficients and windows
- No sacrifice in performance with real-time calculations possible

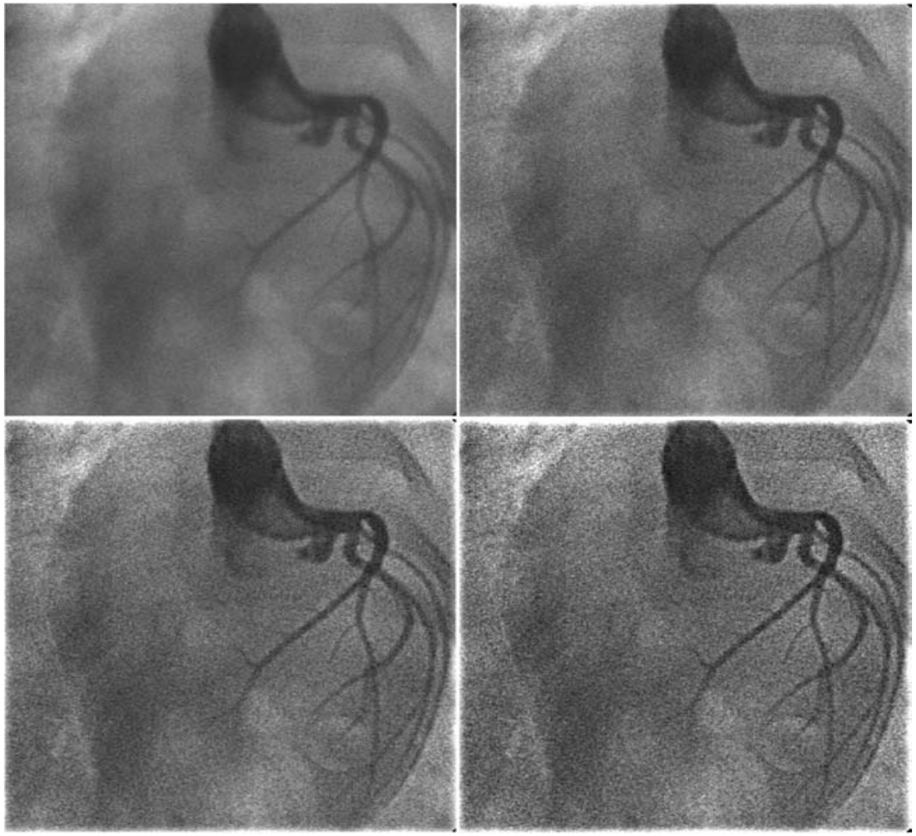
Spatial Enhancement



(H * Original) - (K * Low_pass_filter) = Enhanced Image

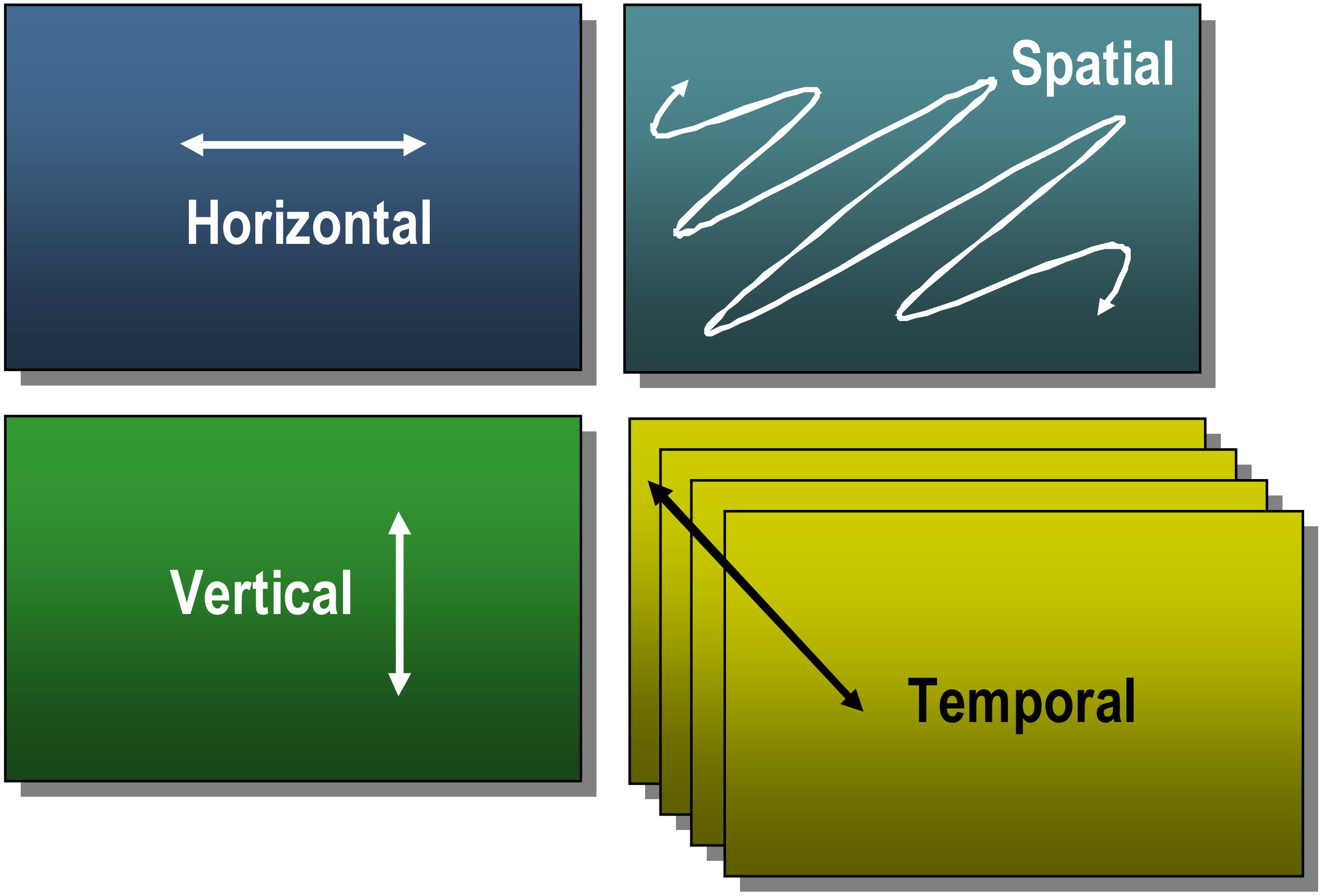


Spatial Enhancement



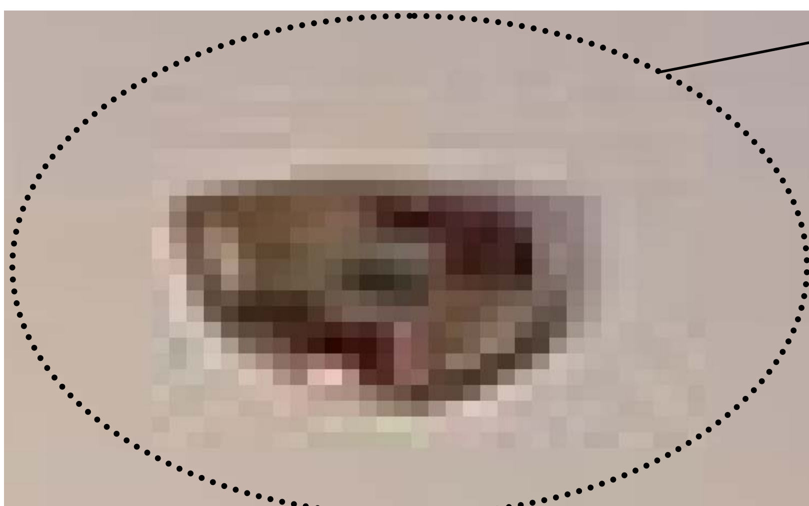
Original, 33%, 66%, 100% Boost

Coherence Domains



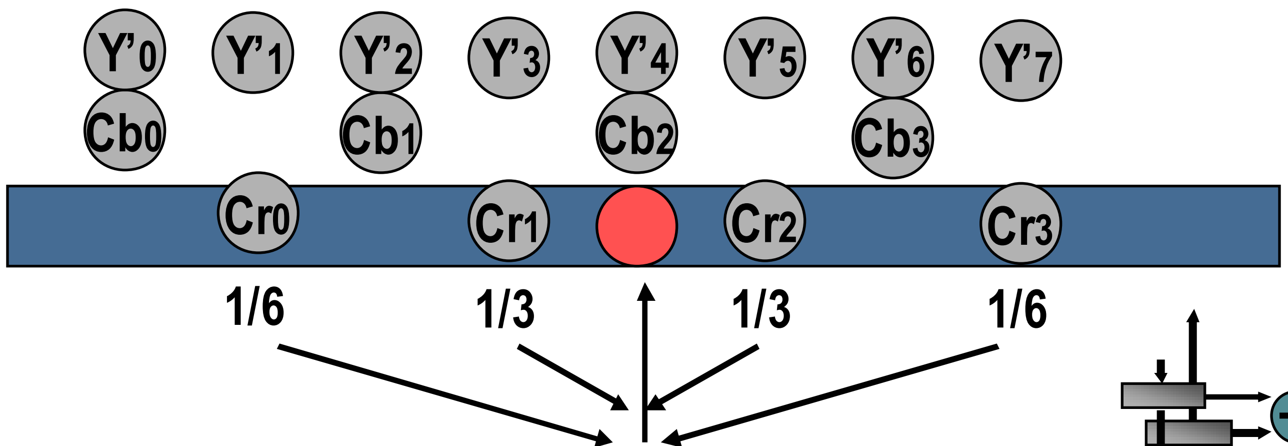
Scene Coherence

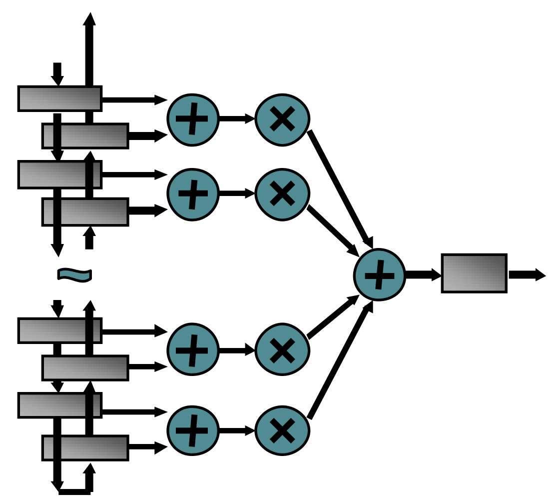
- Notice high frequencies have gone



Coherence Exploited

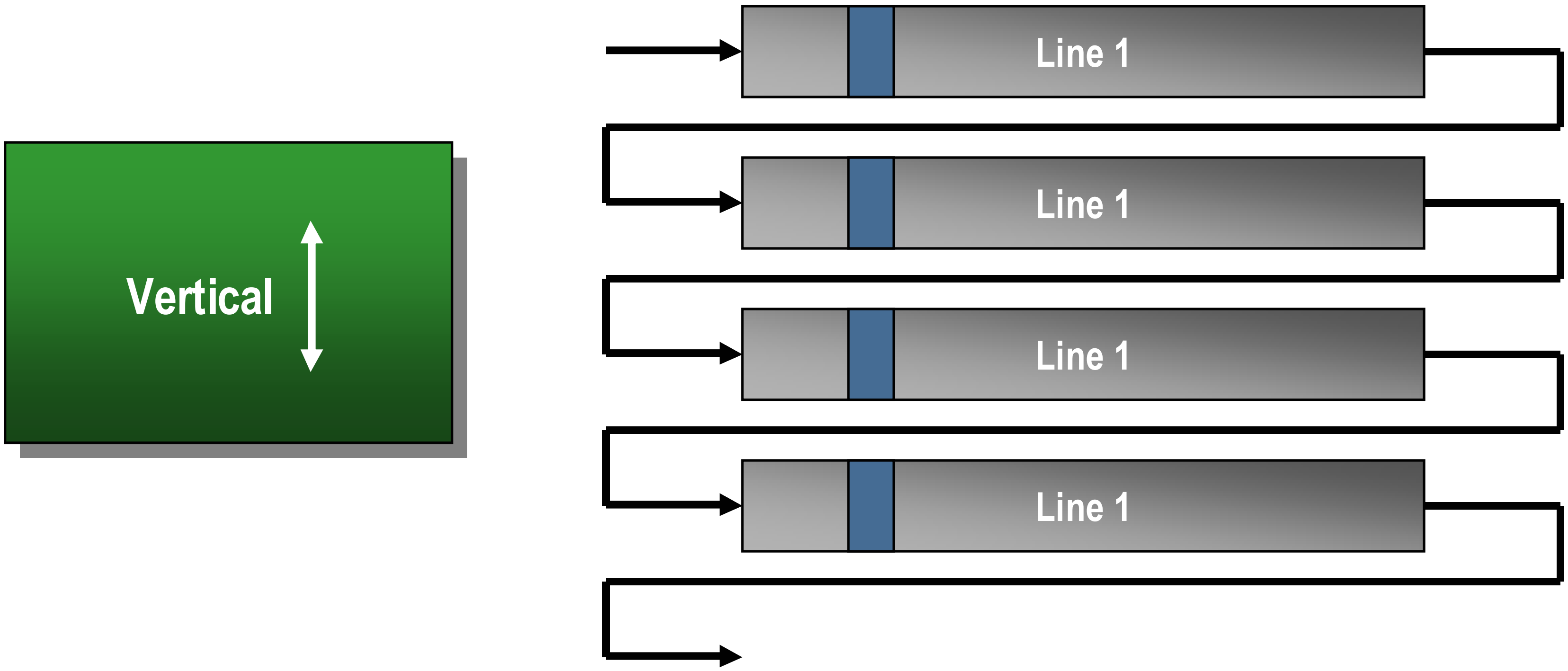
- Horizontal coherence to generate missing data
- FFs & SRL16s look at data in a horizontal stripe





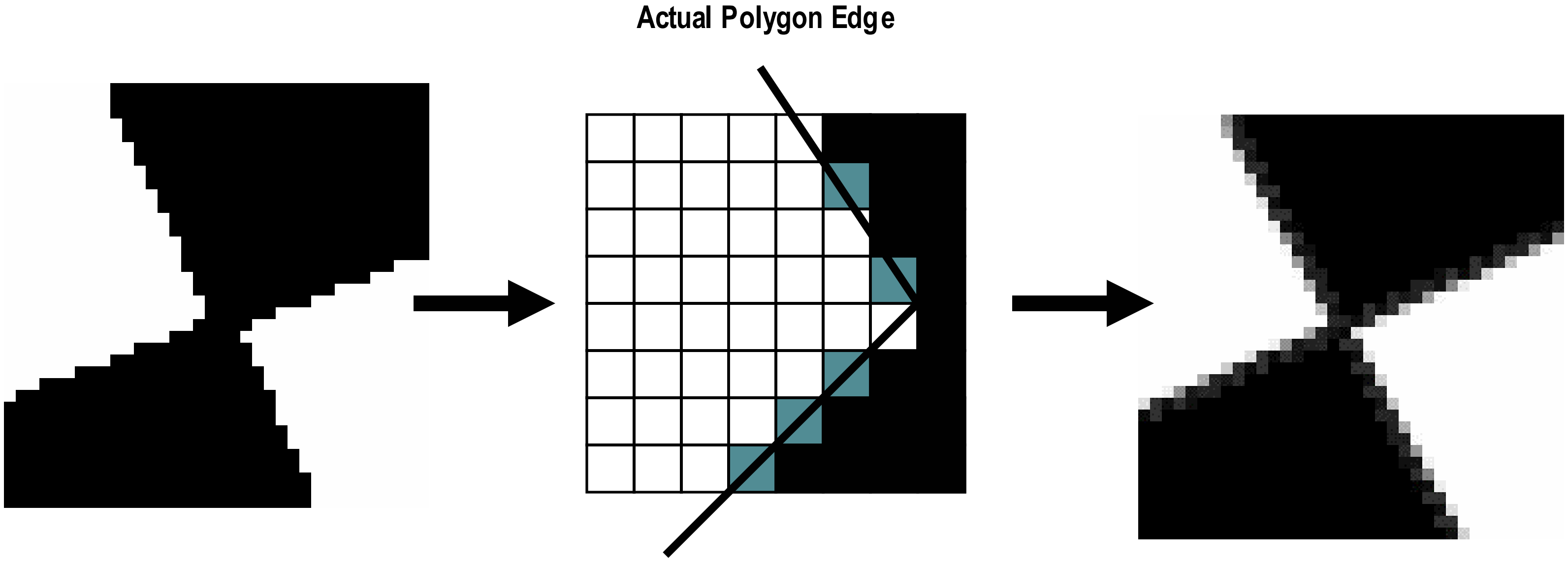
Coherence Exploited

- Vertical coherence to generate missing data
- Line buffers (Block RAM) looks at data in a vertical stripe



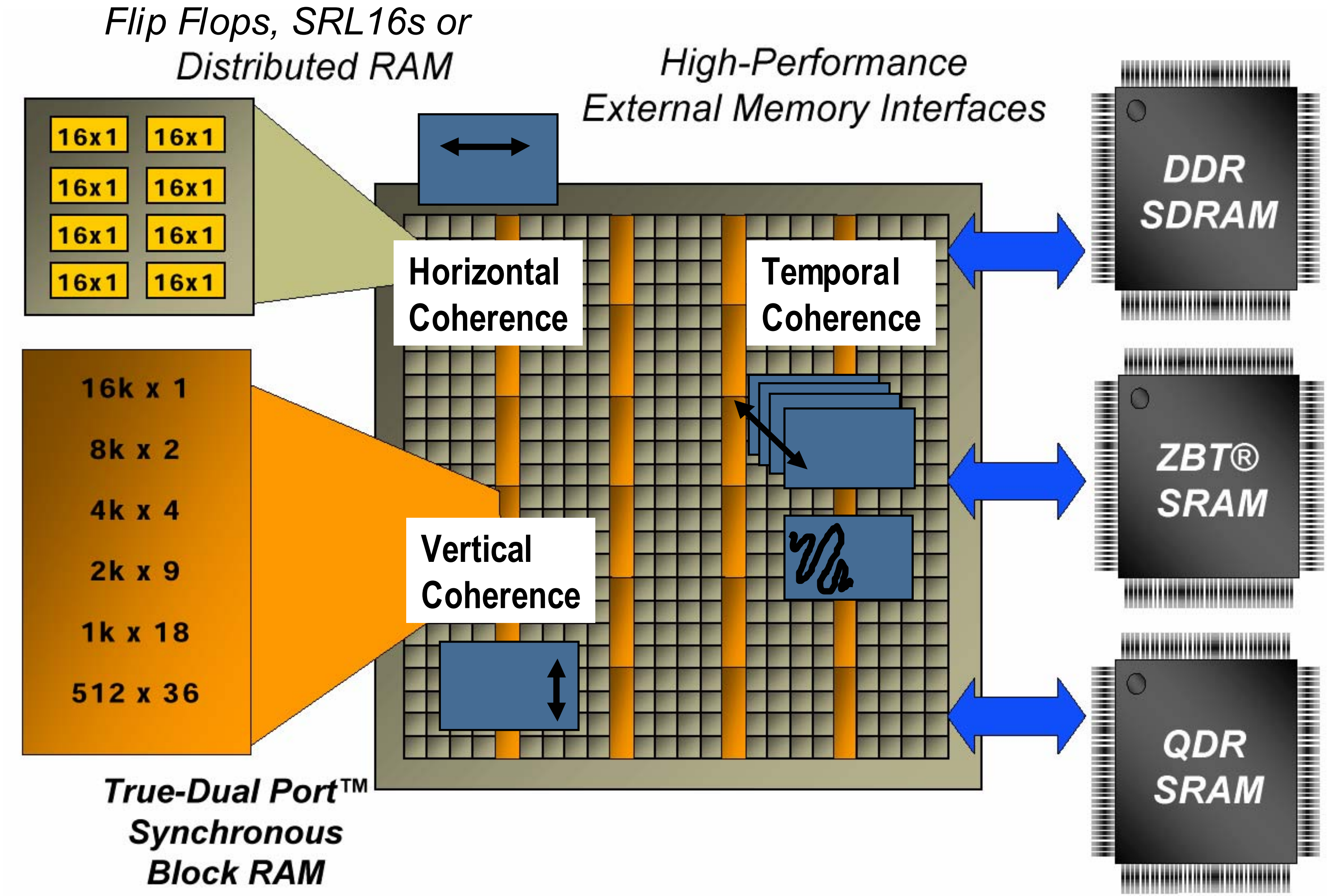
Coherence Exploited

- Spatial coherence to generate grey edge data
- Requires external frame buffers to look at data in a spatial area

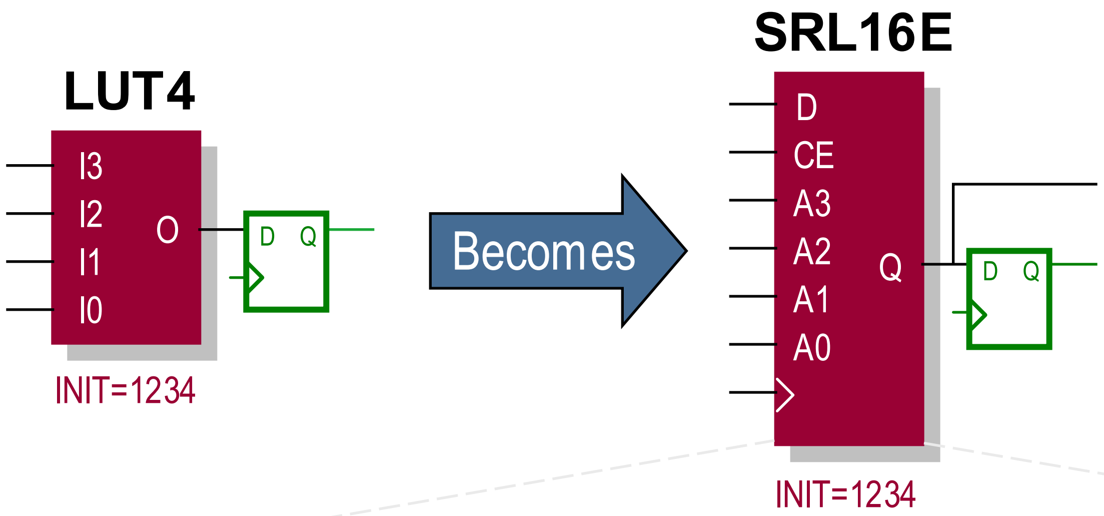


Edge Anti-Aliasing Example

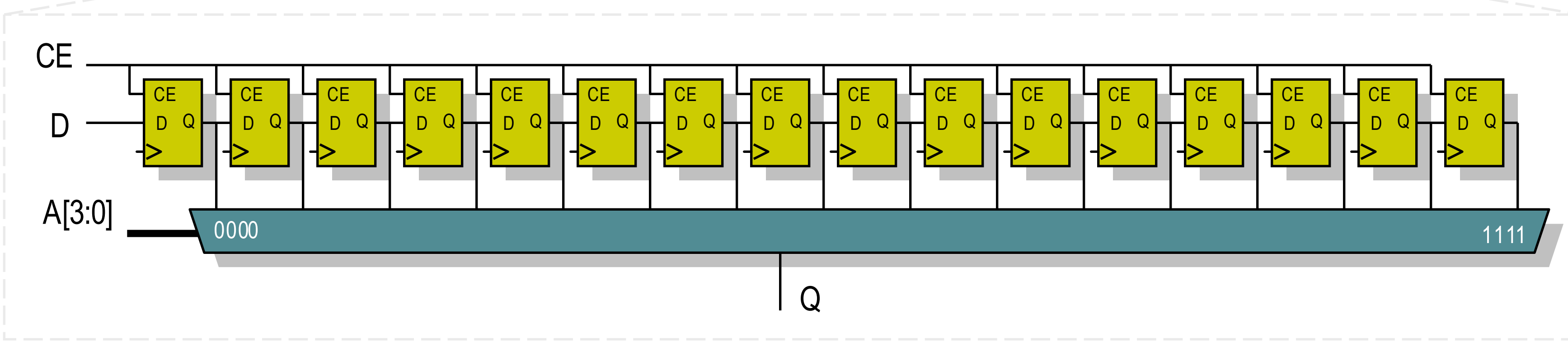
Coherence & Memory Options



Shift Register LUT - SRL16E



- Reading of flip-flop contents is completely independent
  - Address selects which flip-flop is read
- Read process is asynchronous, but dedicated flip-flop is available for synchronization



Xilinx FIR Filter Solution

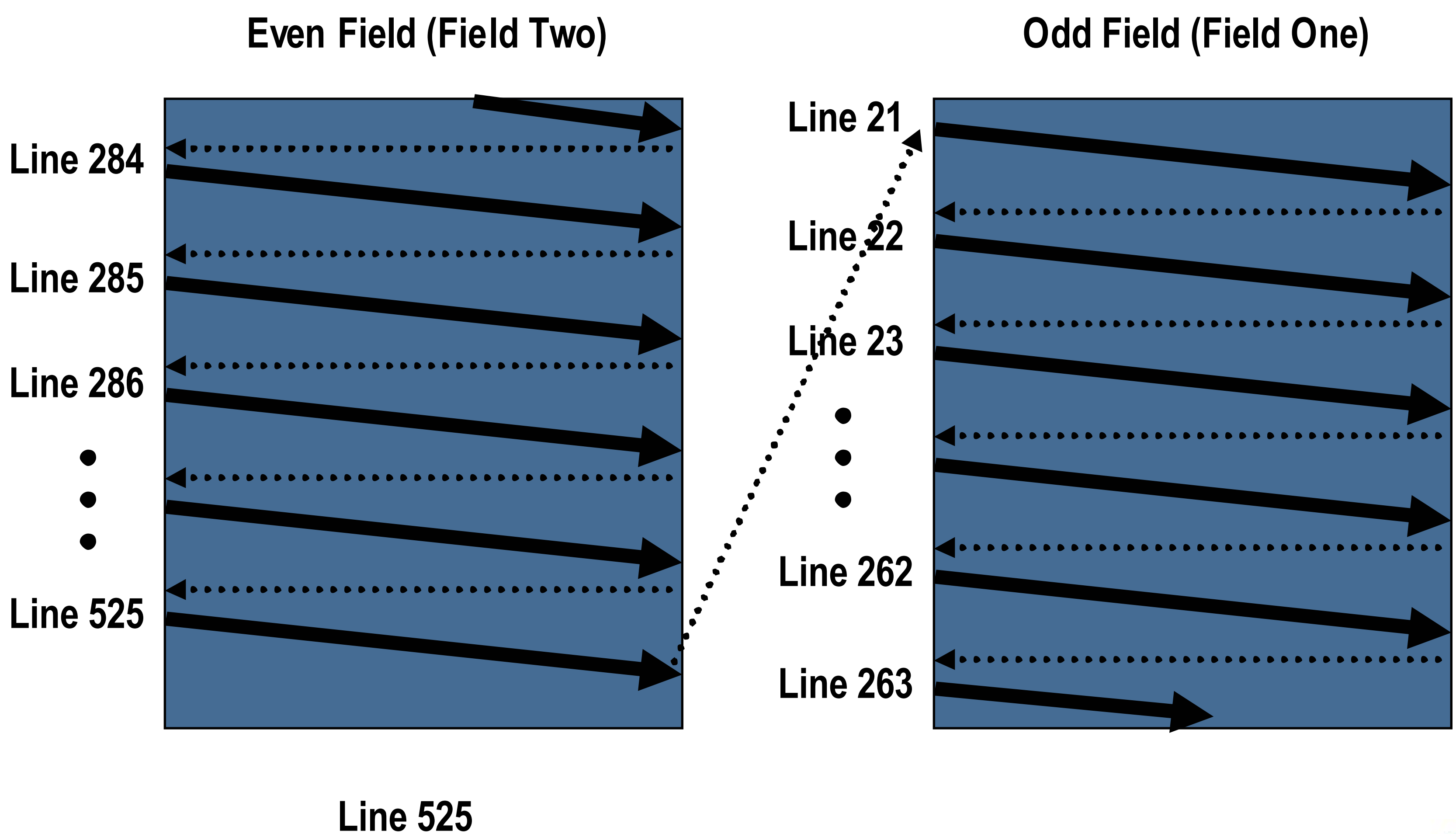
- Virtex logic slice utilization for
  - FIR filter configurations
  - Half-band filter configurations
  - Hilbert transformer configurations
  - Interpolated filter FIR filter configurations
  - Partial to full parallel implementation
- 2D FIR example
  - 1Kx1K image size
  - 8-12 bit pixel data
  - 64x64 kernel size
  - 128 multipliers
  - 64 Line buffers (1024 in length)
  - Summation tree in distributed logic
  - Single-chip solution

Benefits using Xilinx FPGAs

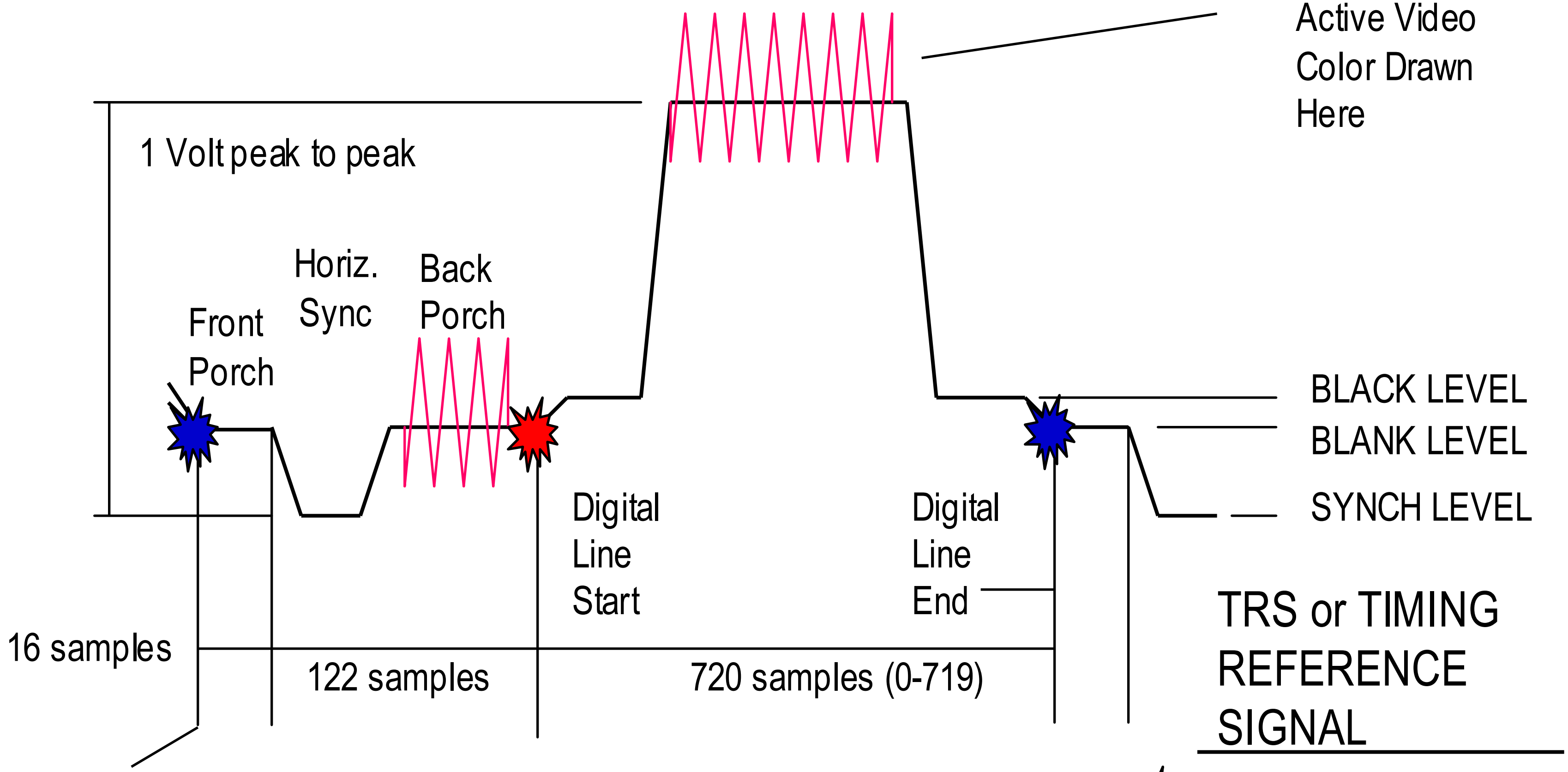
- High Performance
  - Exploit parallelism and can reach sample rate from 1 Mega Sample Per Second (MSPS) up to over 180 MSPS
- Flexibility
  - Highly parameterizable, area efficient high-performance FIR Filter
- Highly Optimized
  - Optimized filters for single rate, half-band, Hilbert transform and interpolated FIR Filters
  - Also takes advantage of symmetry

Lines & Fields

NTSC Interlaced Scan



NTSC Horizontal Scan Line



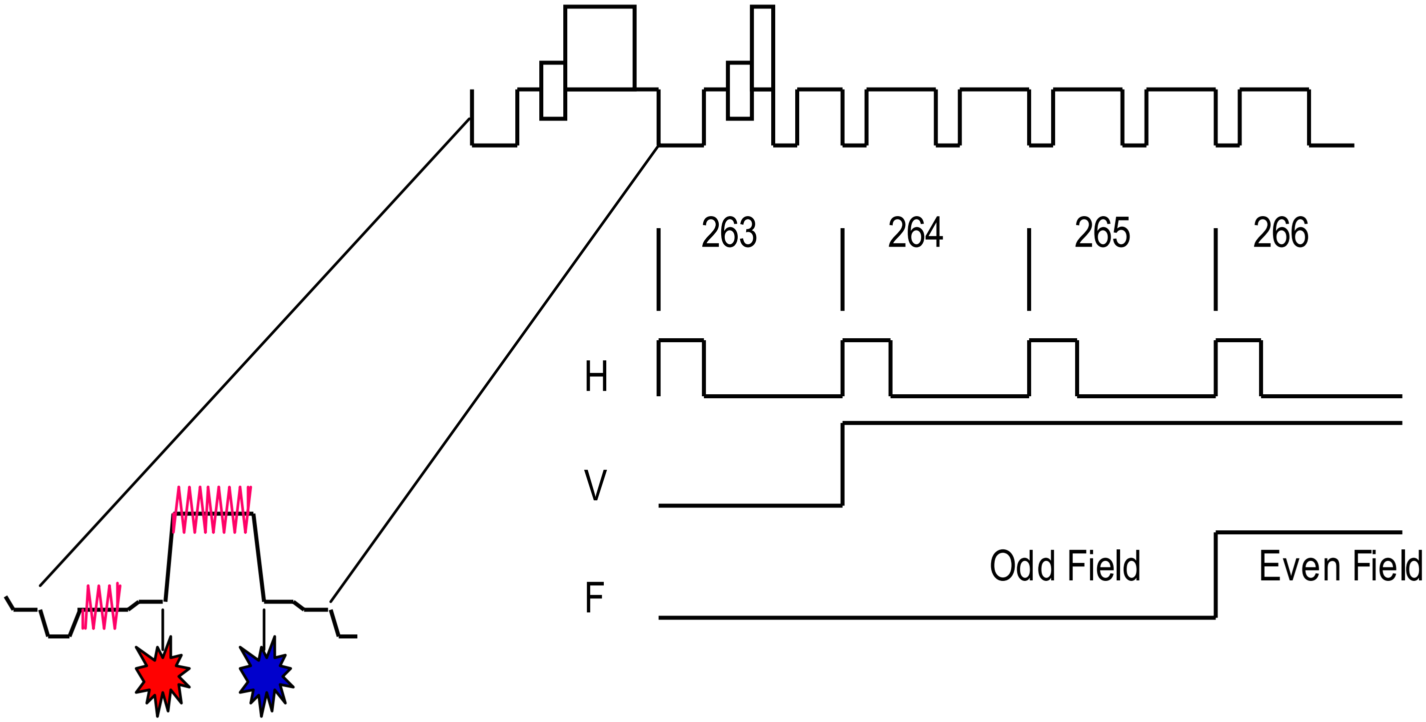
Sample Rate
Y'Cr or Y'Cb every 13.5 MHz (SDTV, NTSC or PAL)
Y'Cr or Y'Cb every 74.25 MHz (HDTV)

TRS or TIMING REFERENCE SIGNAL

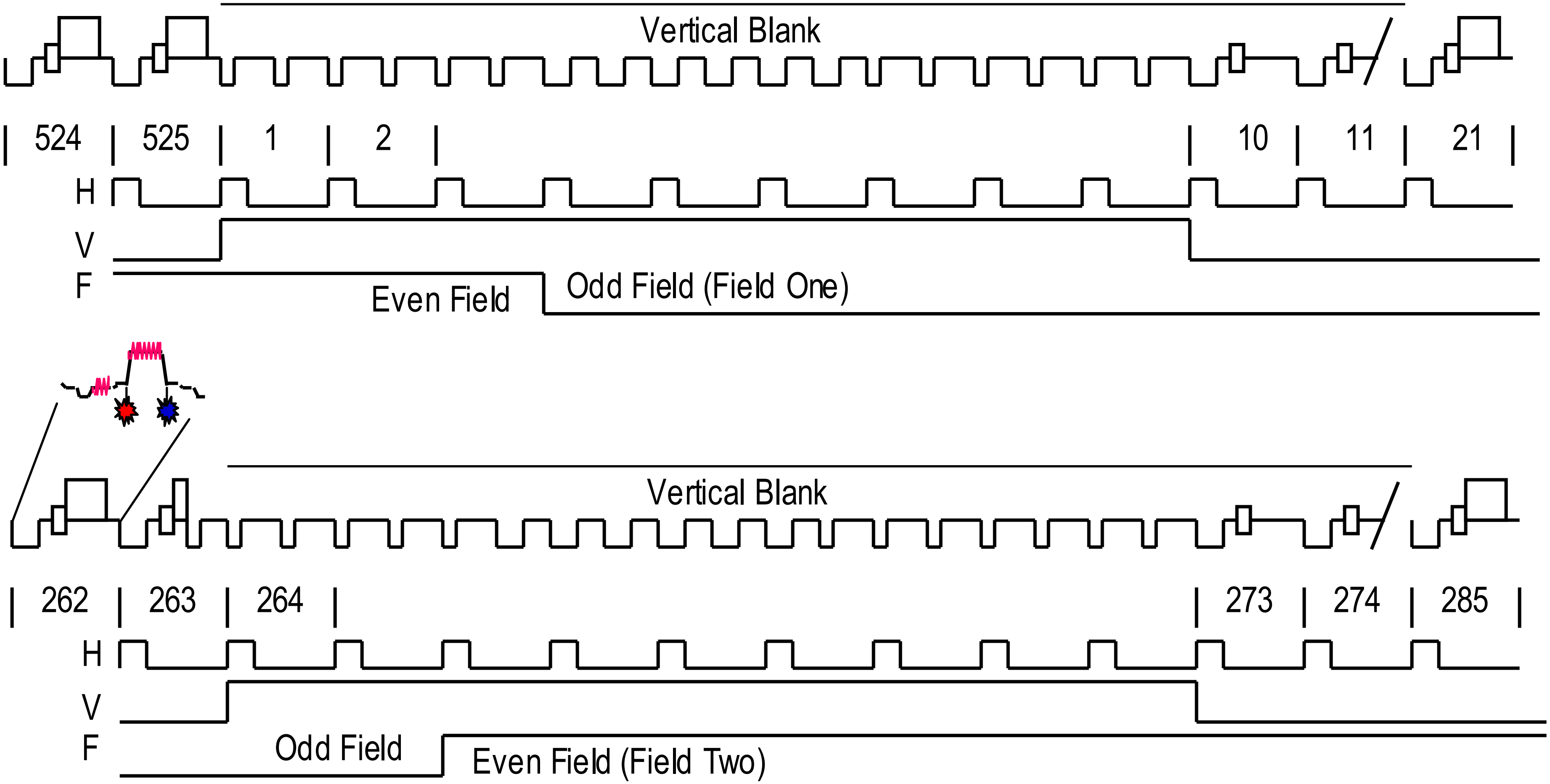
EAV - FF 00 00 XY
SAV - FF 00 00 XY

PAL Scan Line is Similar

Composite NTSC 525 Vertical Timing Detail



Composite NTSC 525 Vertical Timing Detail

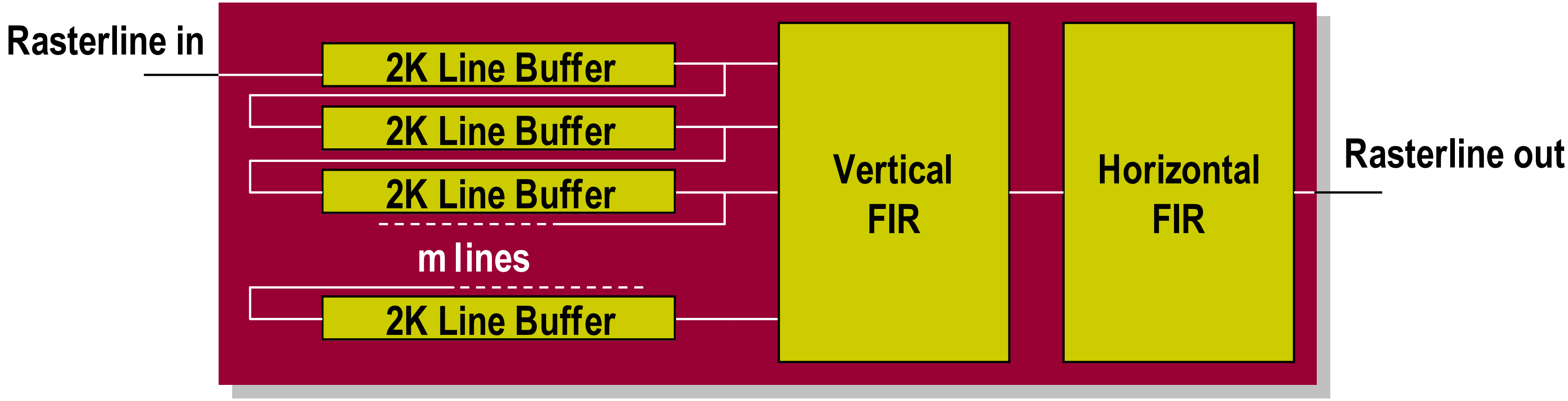


Line/Field Decoding

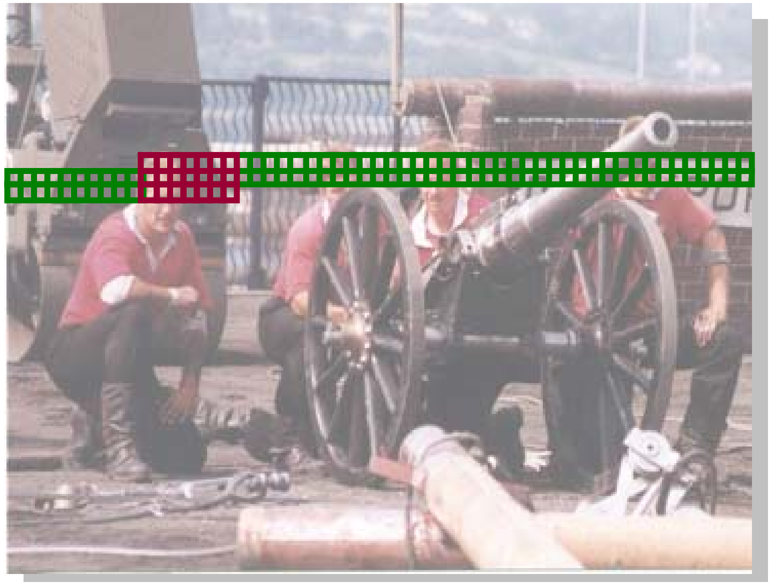
- Simple in a FPGA!
- Find the TRS
  - i.e. The pattern FF 00 00 XY
- Decode XY
  - Gives you H and F
- Format detection is done by counting SAVs during active video

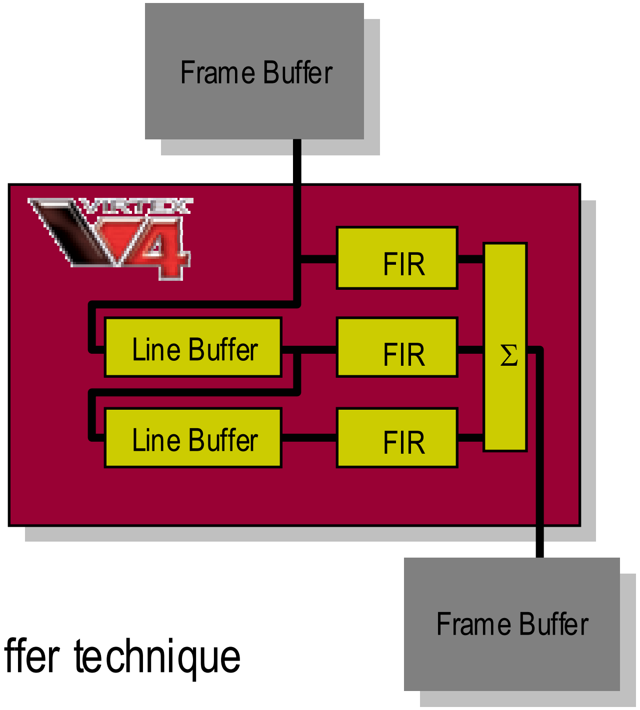
Line Buffering

- Line buffers feed horizontal and vertical FIR filters to do real time image processing without frames store



2D Image Processing Using BlockRAM Line Buffers

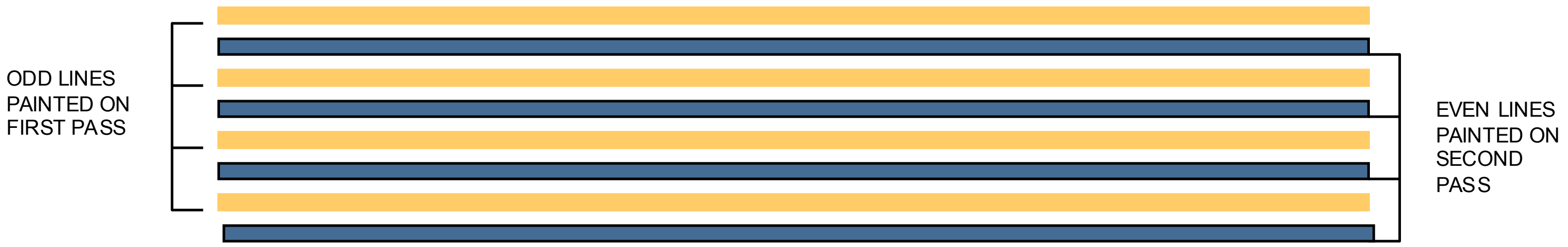




- Line buffers provided by Block RAM using cyclic buffer technique
- 768 Pixel Line Buffers (8-bit)
  - 576 per Device
- 1920 Pixel Line Buffers (36-bit = 12-bit RED + 12-bit GREEN + 12-bit BLUE)
  - 51 per Device

Scan Line De-interlacing

- INTERLACE VIDEO (broadcast video)
  - Half the lines of a frame in a single pass (242 lines @ 30 Hz)



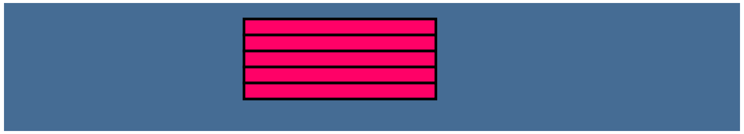
- PROGRESSIVE SCAN VIDEO (computer monitors)
  - All lines of a frame in a single pass (484 lines @ 60Hz)
  - MPEG works on progressive scanned images

Scan Line De-interlacing

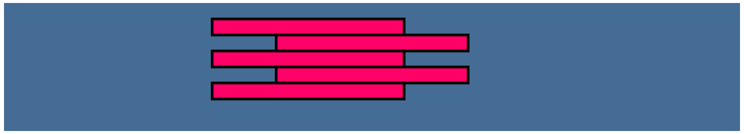
- De-interlacing (line doubling) is process of converting interlaced video into progressive scan video
- Various techniques
  - Scan line duplication from a single field
  - Field merge
    - 2X resolution but motion problems
  - Scan line interpolation from a single field
    - Only 1X resolution
  - Combination approach, field merge (non-moving), interpolate moving objects but difficult motion detection problem
  - Scan Line Interpolation from a single frame

Scan Line De-interlacing

- Field Merge Problems
- Object in motion will have “double image”



Object with no motion

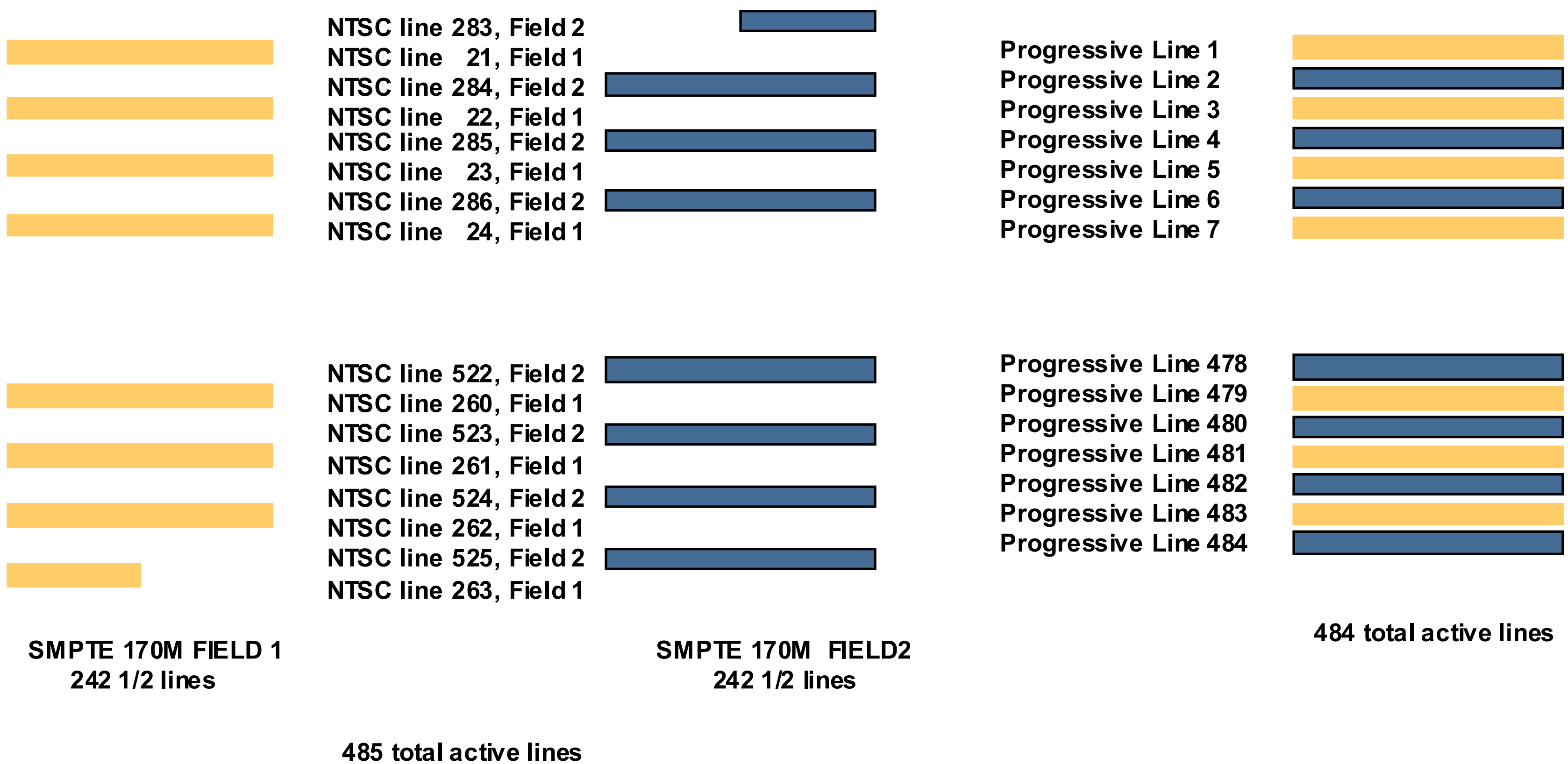


Object in motion

Alternate lines are displaced by horizontal motion

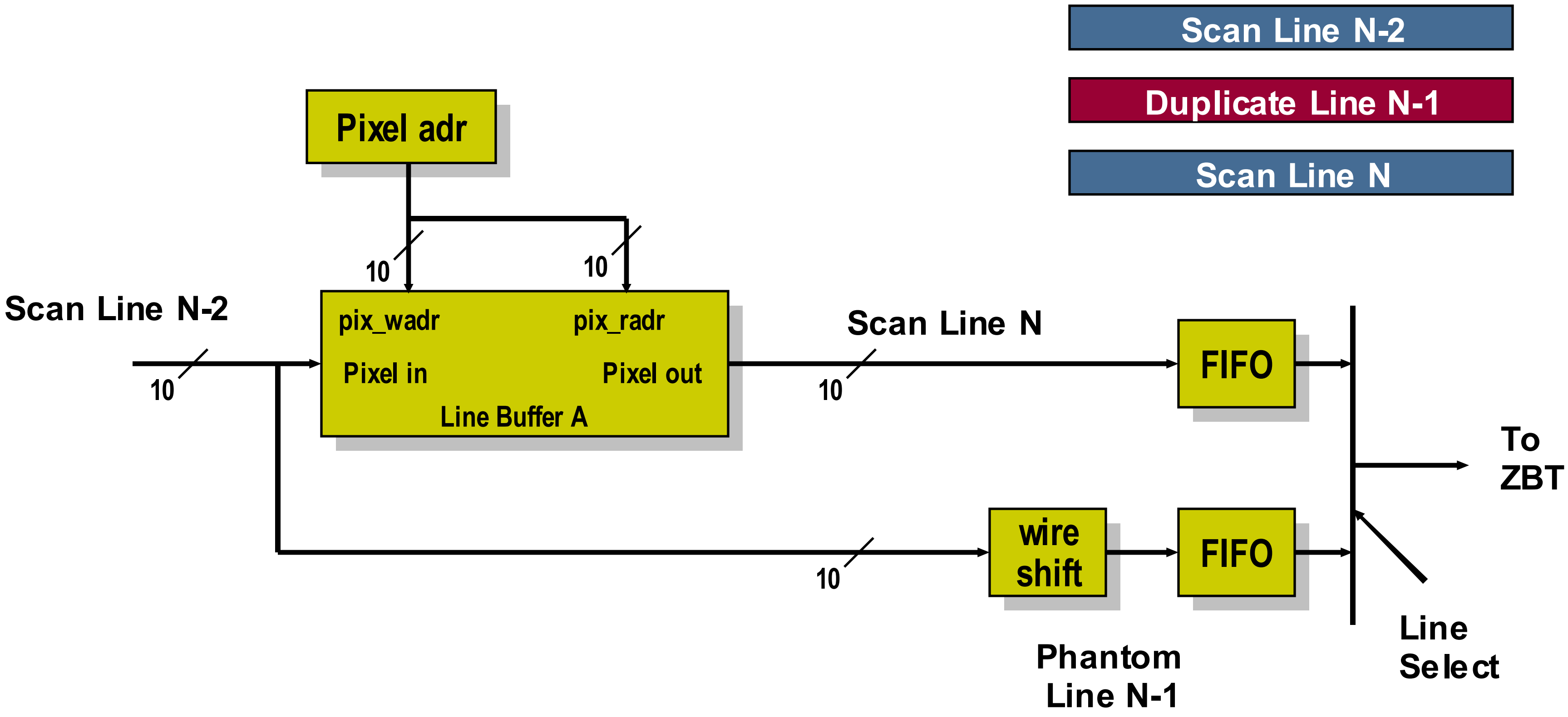
Scan Line De-interlacing

(Green = Field 2, Yellow = Field 1)



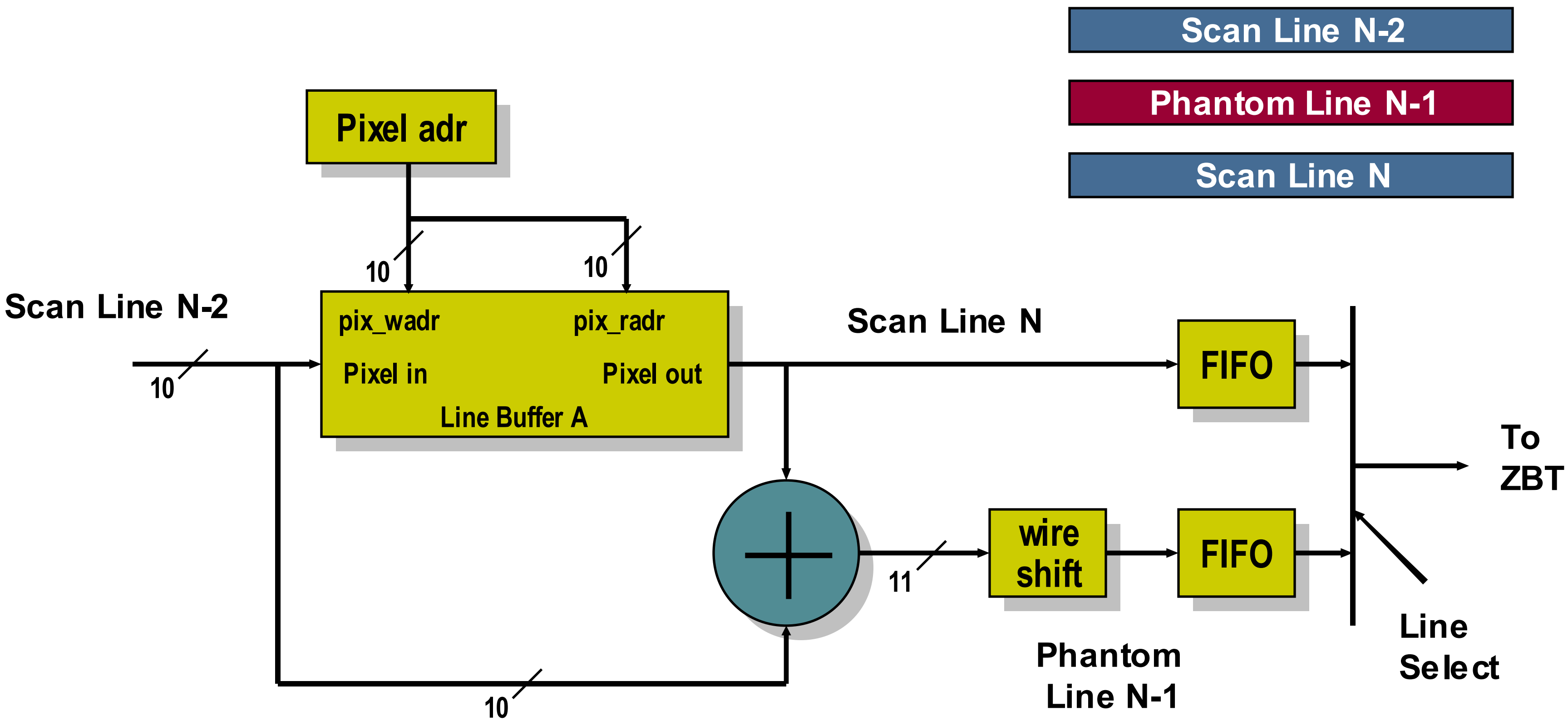
Scan Line Deinterlacing

Using Line Duplication



Scan Line Deinterlacing

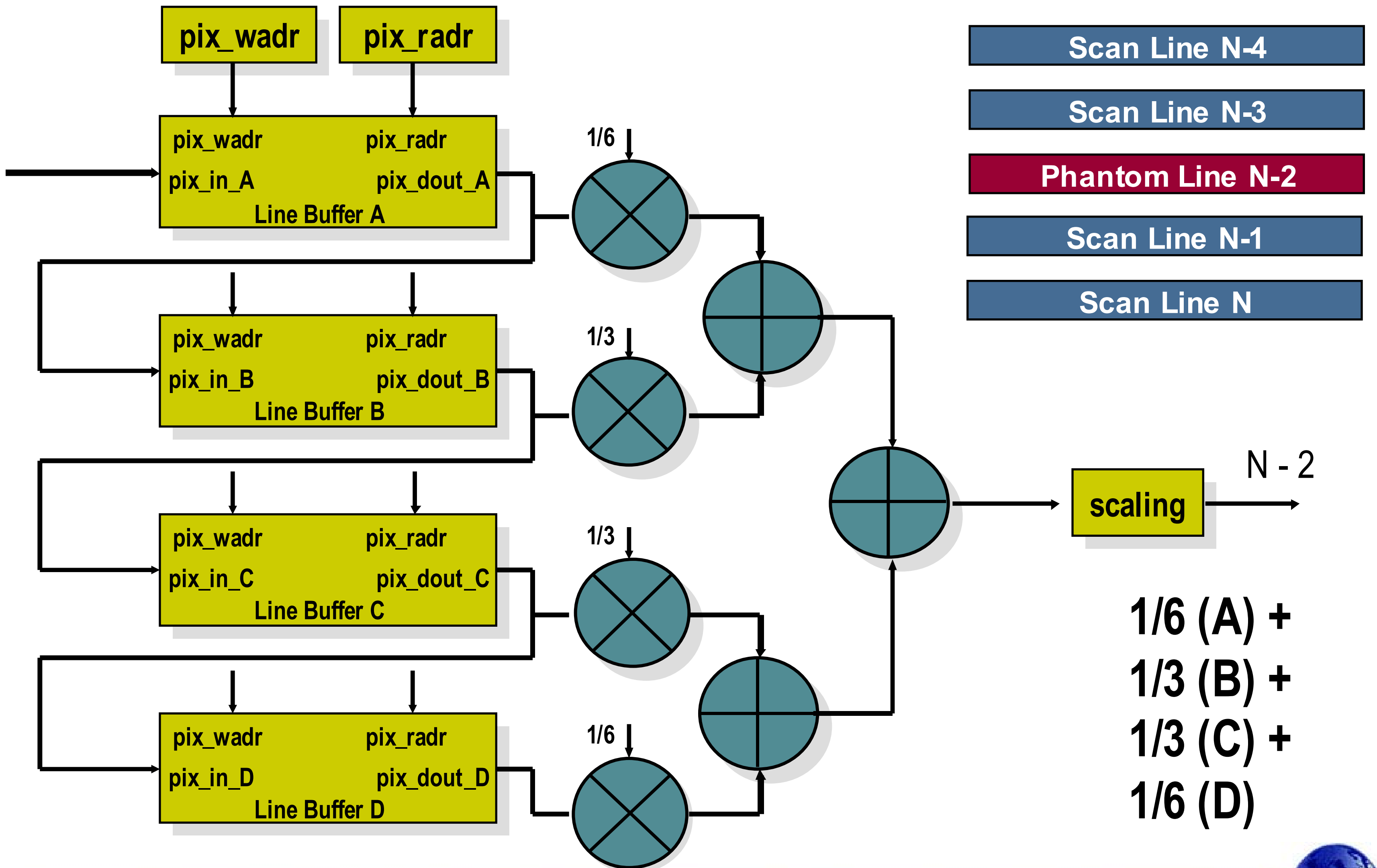
Using Line Interpolation from 2 Lines



(Pixel A + Pixel B) / 2

Scan Line Deinterlacing

Using Line Interpolation from 4 Lines



1/6 (A) +
1/3 (B) +
1/3 (C) +
1/6 (D)

Compression

Video Compression

- Bandwidth is precious!
- MPEG compression helps get the most out of available bandwidth
  - A trade off between amount of data to be sent and acceptable picture quality
- Uncompressed high-definition pictures take too much bandwidth to send down a 6MHz or 8MHz cable channel (up to 40Mbps)
- **1920 x 1080 24-bit pixels @ 30 frames per second = 1.49Gbps!**

Definition	Lines/Frame	Pixels/Line	Aspect Ratios	Frame Rates
High (HD)	1080	1920	16:9	23.976p, 24p, 29.97p, 29.97i, 30p, 30i
High (HD)	720	1280	16:9	23.976p, 24p, 29.97p 30p, 59.94p, 60p
Standard (SD)	480	704	4:3, 16:9	23.976p, 24p, 29.97p, 29.97i, 30p, 30i, 59.94p, 60p
Standard (SD)	480	640	16:9	23.976p, 24p, 29.97p, 29.97i, 30p, 30i, 59.94p, 60p

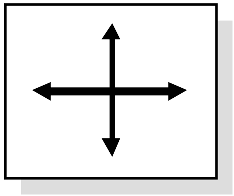
ATSC “Table III”

- Storage of content is also much more efficient with video compression

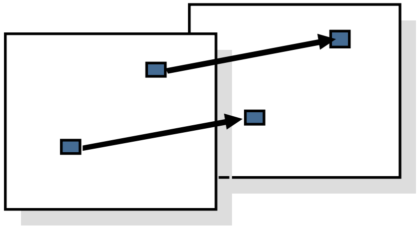
XILINX

MPEG Compression

- Spatial Processing
  - Uses DCT within a single picture to enable removal of high frequencies not discernable to human eye



- Temporal Processing
  - Seeking out and removing redundancy between successive images/frames

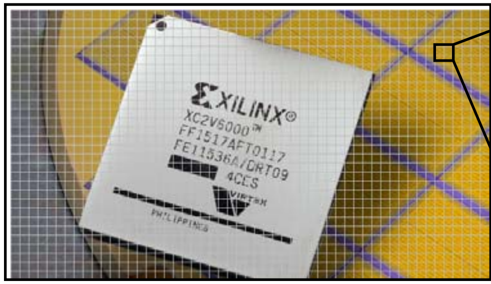


- Variable Length Coding (VLC)
  - Use shortest codes for most common samples
- Run Length Encoding (RLE)
  - Replace long strings of zeros with single command code

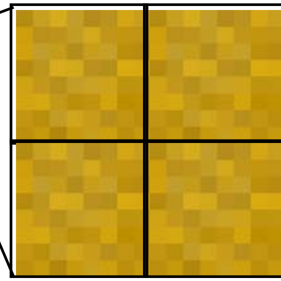
Spatial Redundancy

- DCT
  - Returns the discrete cosine transform of 'video/audio input'
  - Can be referred to as the even part of the Fourier series
  - Converts an image or audio block into its equivalent frequency coefficients
- IDCT
  - Inverse of the DCT function
  - IDCT reconstructs a sequence from its discrete cosine transform (DCT) coefficients

DCT in MPEG Compression

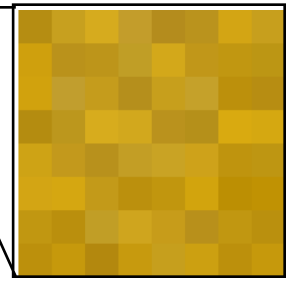


1



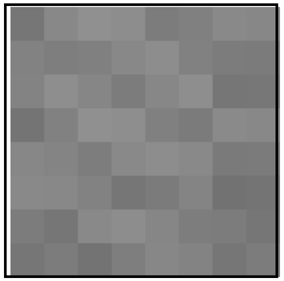
Scan picture using 16x16 pixel "macroblock"

2



Scan macroblock in 8x8 blocks

3



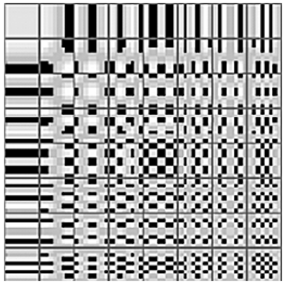
Determine luma & chroma pixel values

4

119	137	145	142	125	129	138	136
131	127	129	137	142	130	125	124
132	143	135	125	136	143	119	120
117	130	143	129	124	139	137	138
136	133	126	138	142	138	123	124
138	137	131	119	124	133	115	117
125	119	138	142	135	126	125	119
119	125	116	127	136	132	119	125

Luma samples shown here (Chroma processed seperately)

5

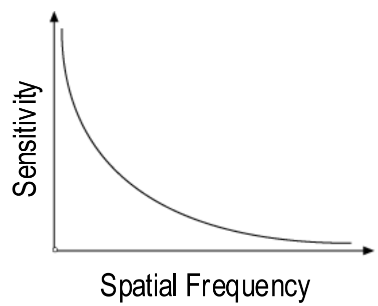


Convert to frequency components (DCT)

6

6981	245	500	364	331	139	35	21
205	168	192	278	55	40	12	4
152	288	207	99	26	85	29	10
109	142	166	111	77	66	38	2
55	84	212	56	10	4	11	5
61	136	61	43	47	13	3	2
32	119	61	26	32	1	0	2
24	26	4	31	23	52	49	25

Output DCT coefficients



Human eye less sensitive to high frequencies

7

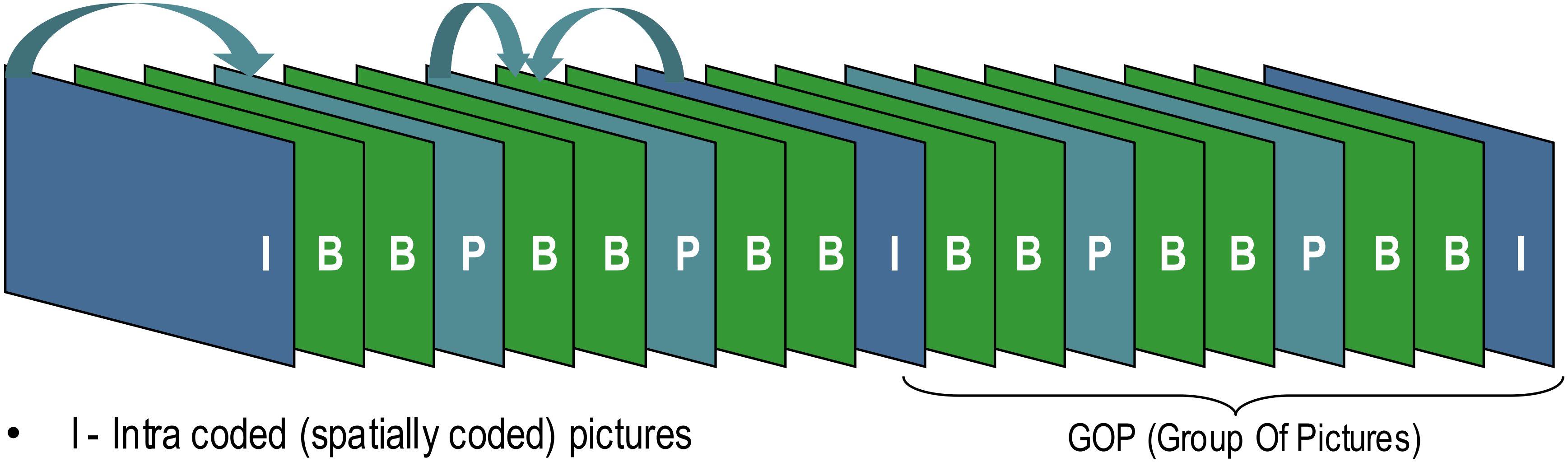
920	15	28	18	14	5	1	0
17	10	9	10	4	1	0	0
10	14	9	5	1	0	0	1
6	5	4	4	2	1	0	0
2	4	7	1	0	0	0	0
2	3	2	1	1	0	0	0
1	3	2	1	0	0	0	0
0	1	0	0	0	0	0	0

Quantize higher frequencies with less bits (weighting)
Zero values for frequencies below perception threshold

8

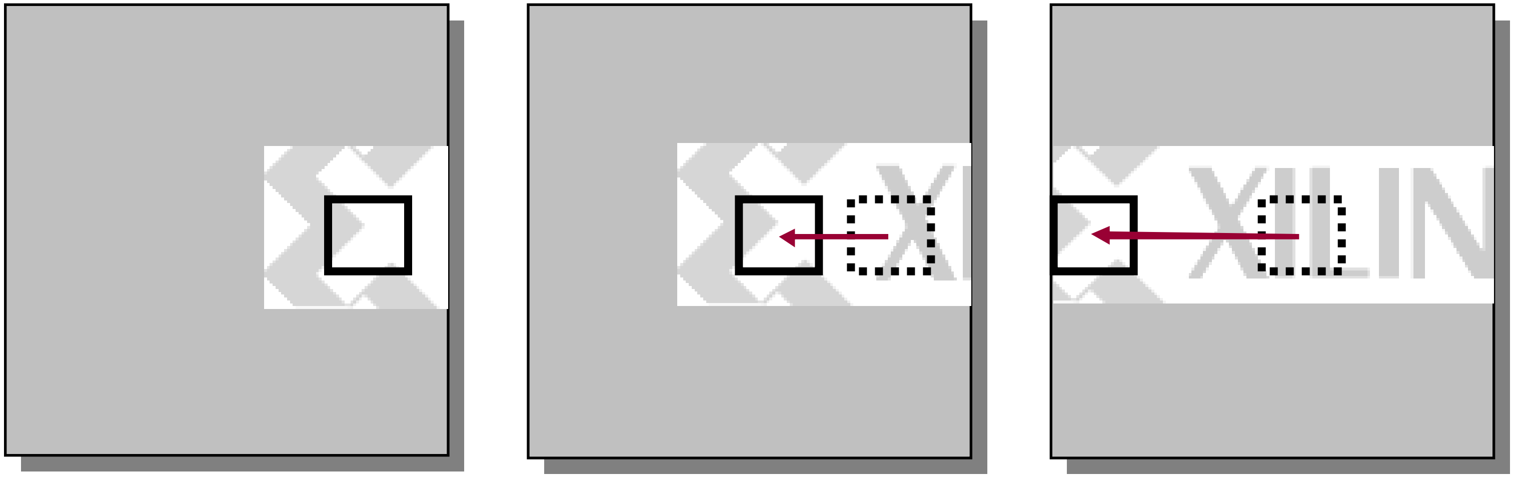
Compress using zigzag scan and run length encoding for zero values (in blue)
Further compression with Huffman encoding (VLC)

Temporal Redundancy



- I - Intra coded (spatially coded) pictures
  - Forms the anchor for a GOP
- P - Forward Predicted pictures
  - Predicted from previous I or P pictures
  - P picture made up of vectors showing where to get pixel data from in previous pictures and/or values that must be added to previous picture to get current pixel value
- B - Bidirectionally Predicted pictures
  - Predicted from previous or later I or P pictures (never from other B pictures)
  - Made up of vectors showing where to get pixel data from in previous pictures

Motion Estimation



Part of image that is common between frames but has moved

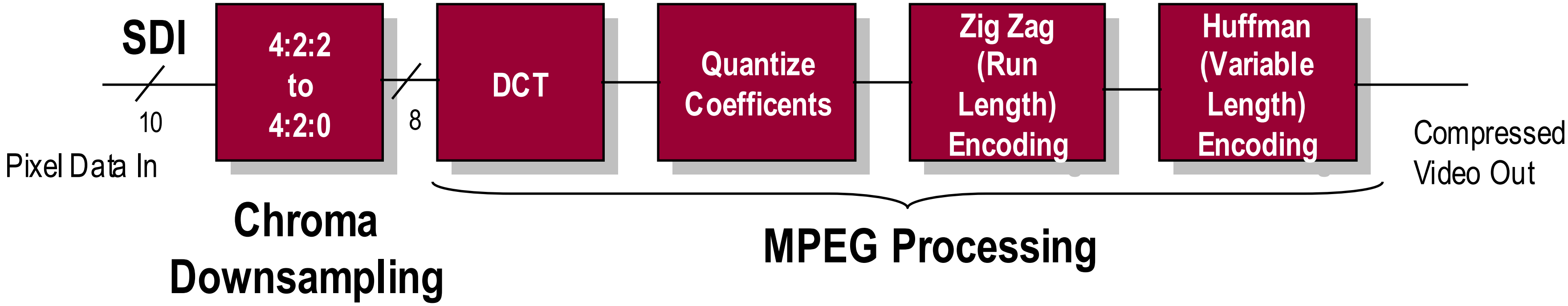
Original position of picture segment in previous frame

Motion vector sent with any prediction errors

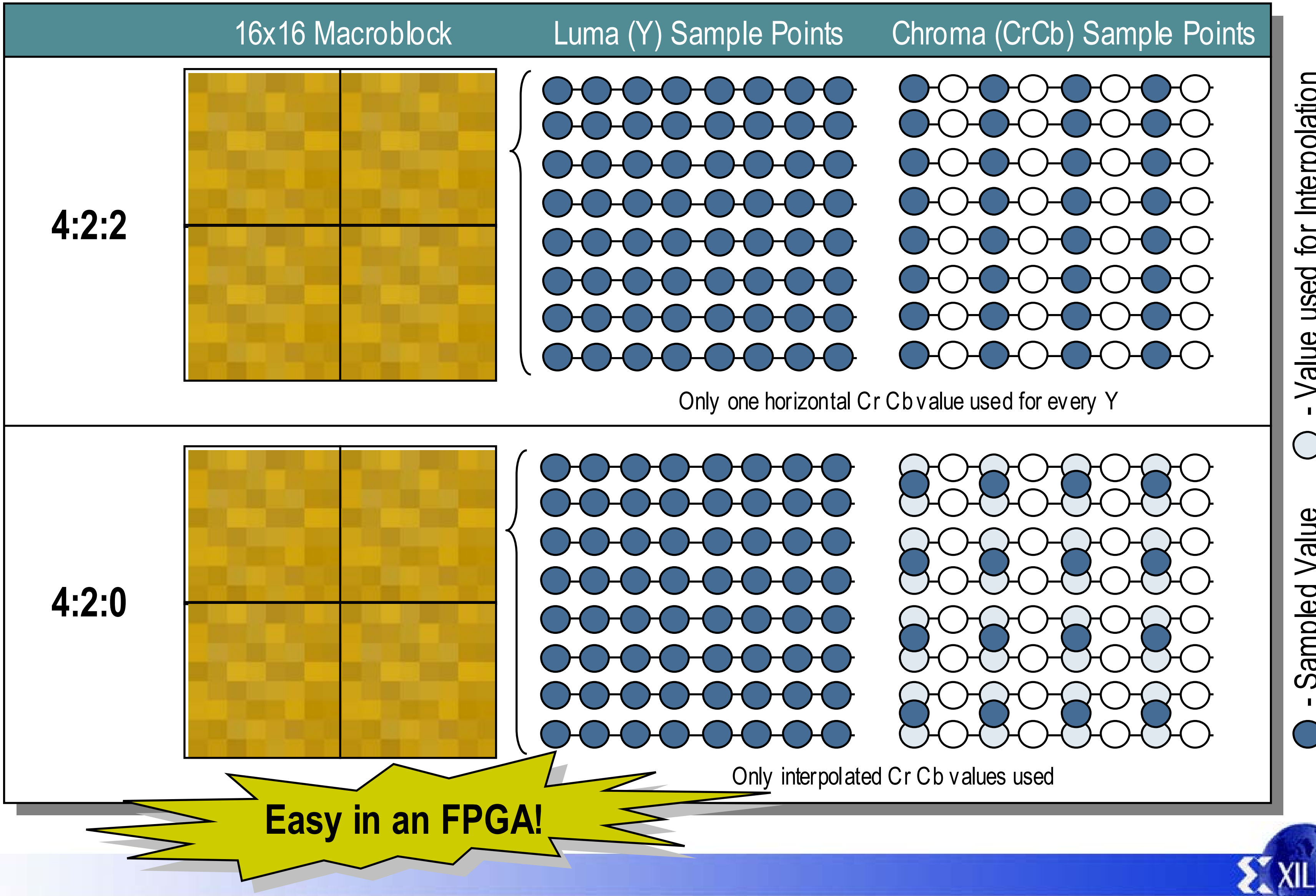
- Estimation predicts next picture by shifting data from previous picture along a calculated motion vector
- In encoder, predicted picture is compared to actual picture and any prediction errors calculated
- Transmitting motion vectors and prediction errors takes much less bandwidth than coding entire picture

Chroma Downsampling

- Most MPEG-2 applications use 8-bit 4:2:0 sampling
- But incoming data usually 10-bit 4:2:2 video
  - Maybe via SDI (Serial Digital Interface) for example
- Conversion therefore needed before MPEG processing
  - This chroma “downsampling” is lossy form of data compression

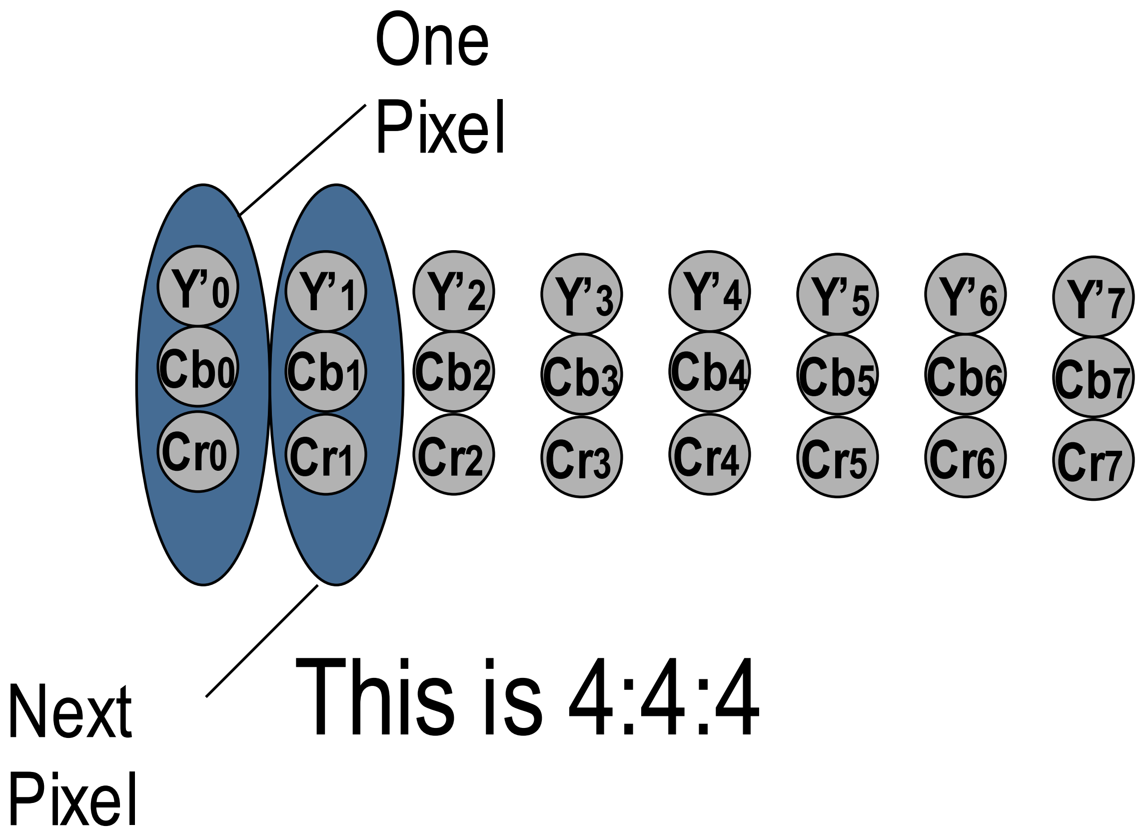


4:2:2 and 4:2:0 Sampling



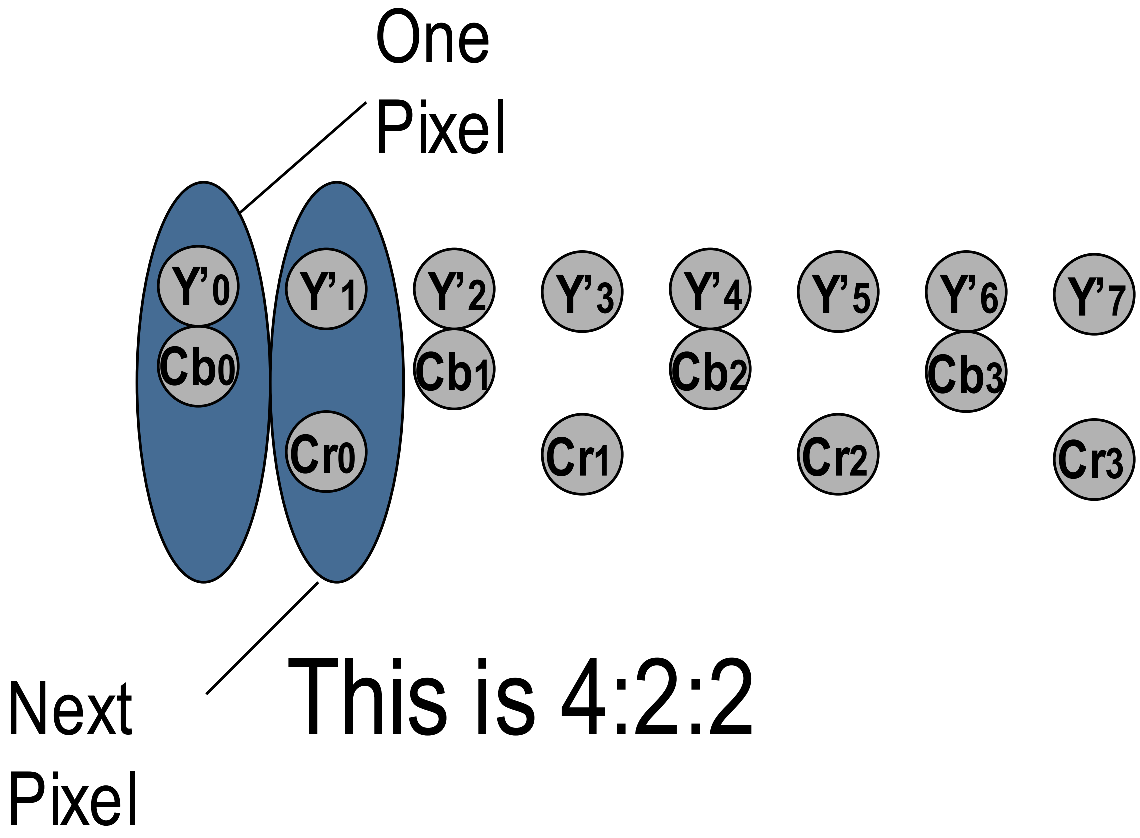
Digital Component Video Conversion - 4:4:4 to 4:2:2

- XAPP294



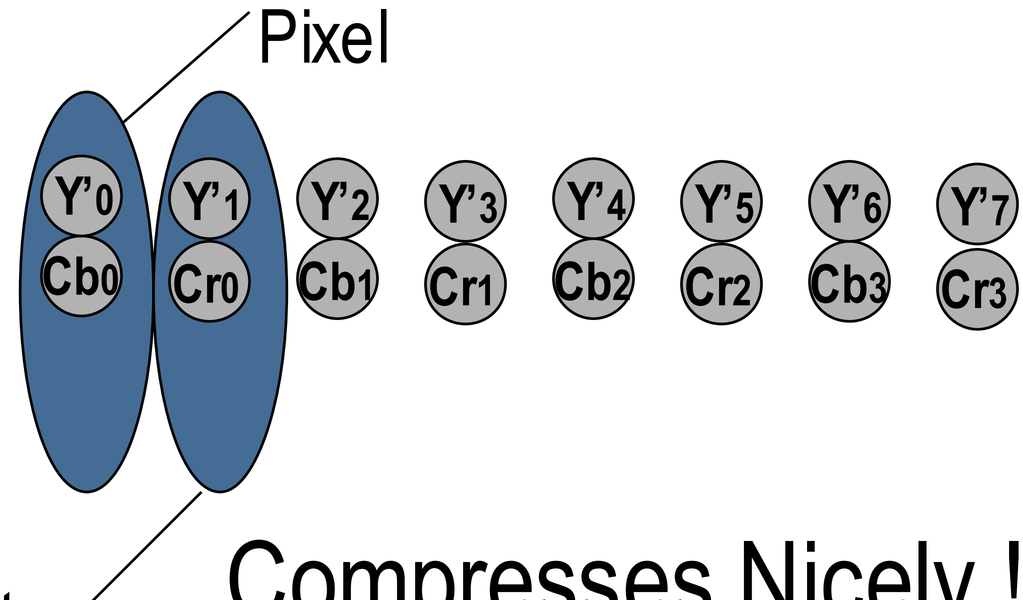
Digital Component Video Conversion - 4:4:4 to 4:2:2

- XAPP294



Digital Component Video Conversion - 4:4:4 to 4:2:2

- XAPP294

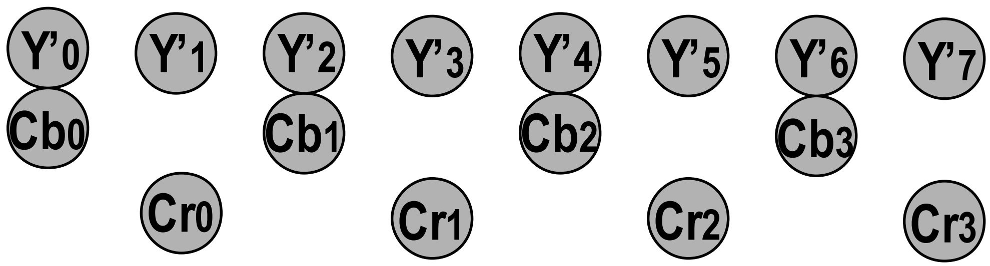


Compresses Nicely !

(and is imperceptible to human eye)

Digital Component Video Conversion - 4:2:2 to 4:4:4

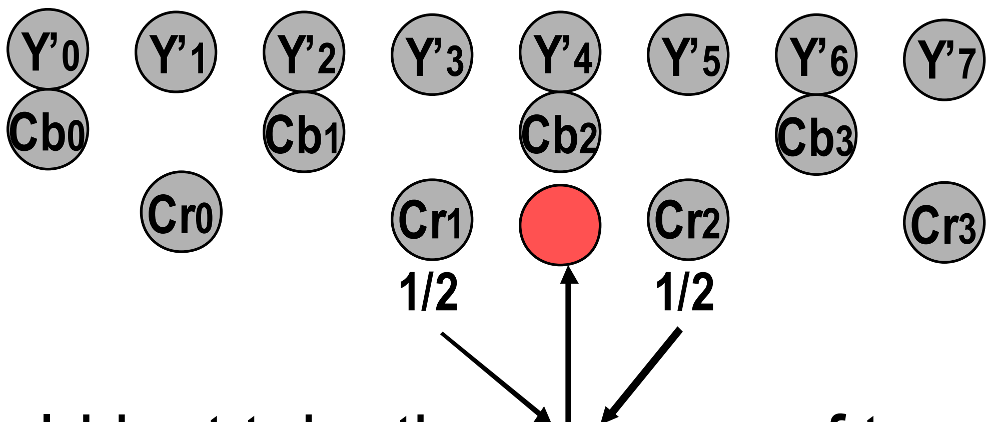
- XAPP294



- How do we get back the missing samples?

Digital Component Video Conversion - 4:2:2 to 4:4:4

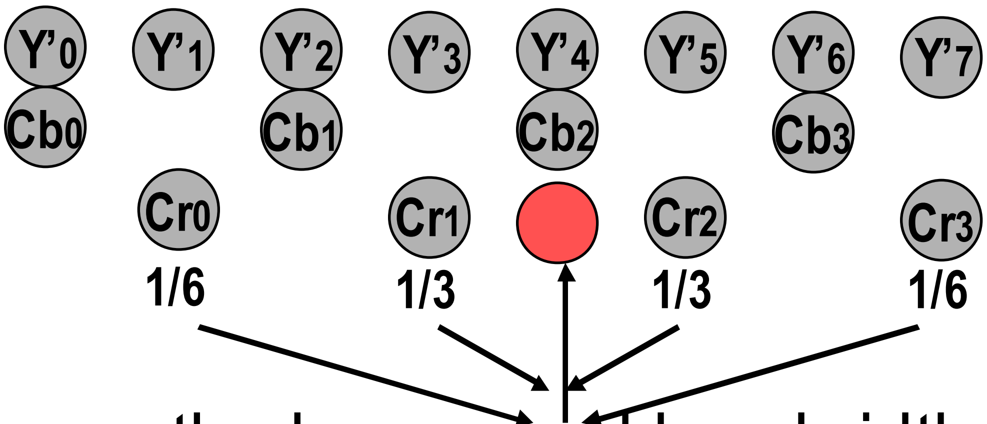
- XAPP294



- You could just take the average of two values to get the missing one between
- 1/2 (A + C)

Digital Component Video Conversion - 4:2:2 to 4:4:4

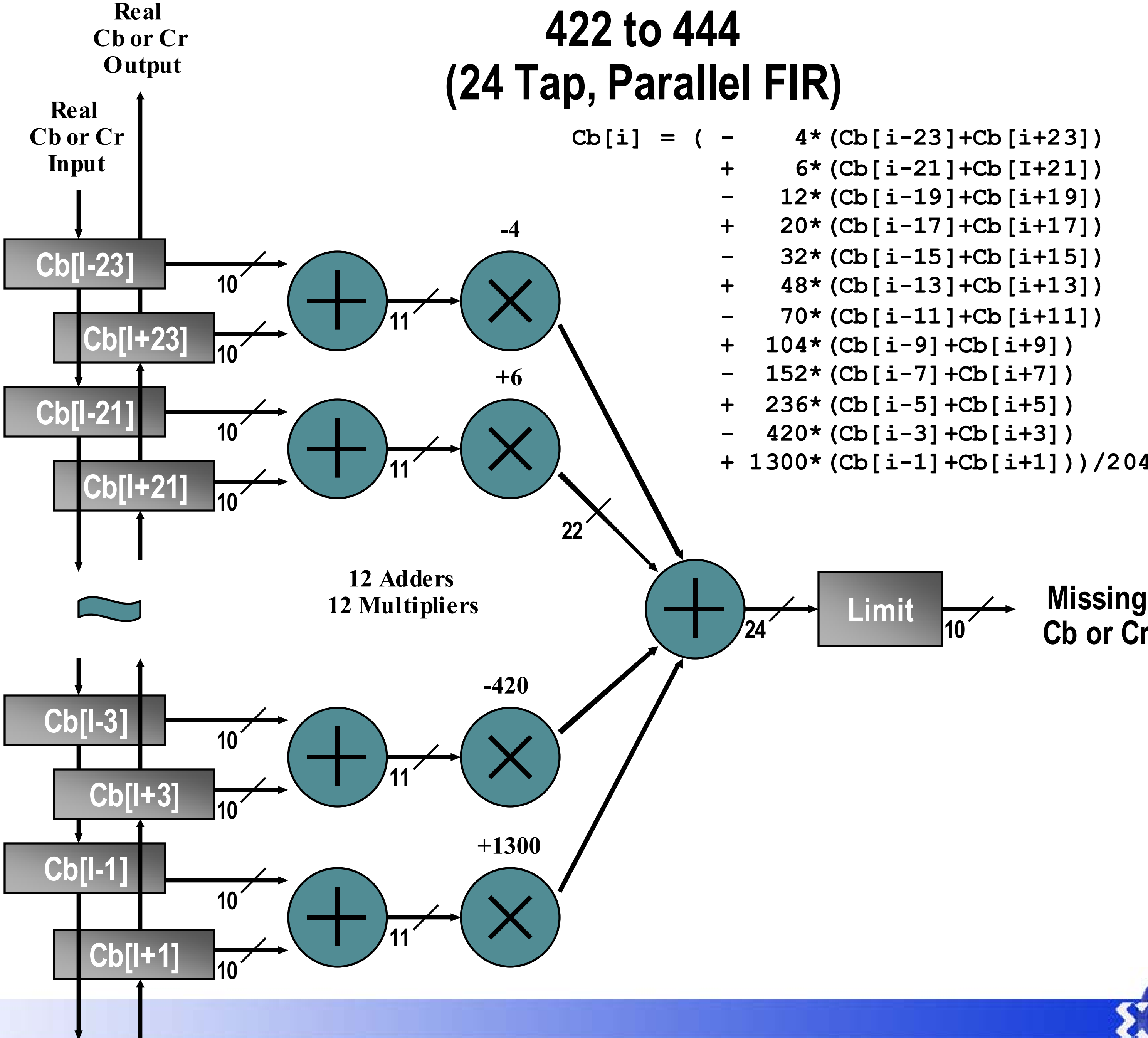
- XAPP294



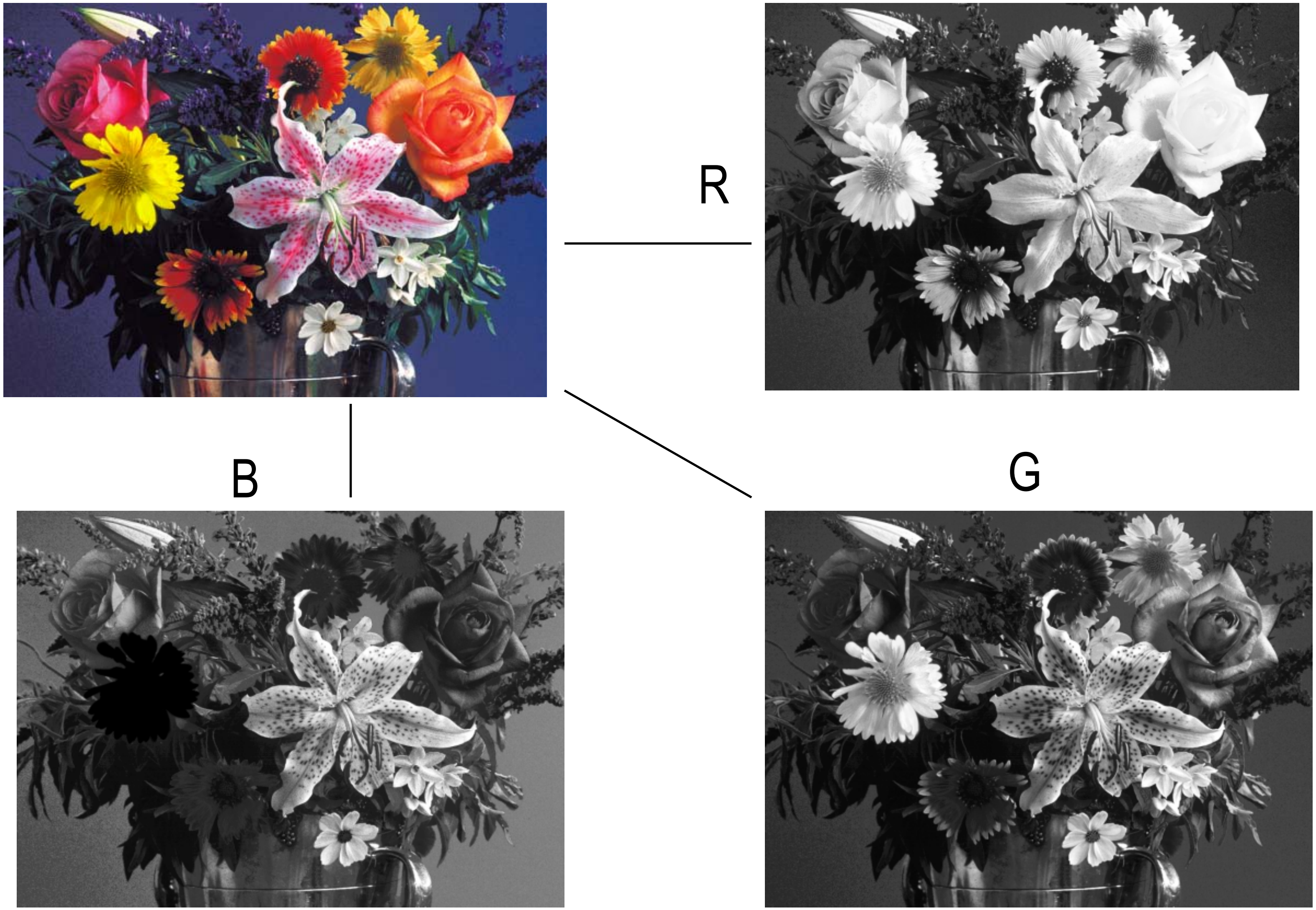
- Previous method may need bandwidth limiting, this version is better
- Less multiplies → 1/6 (A + D) + 1/3 (B + C)

422 to 444
(24 Tap, Parallel FIR)

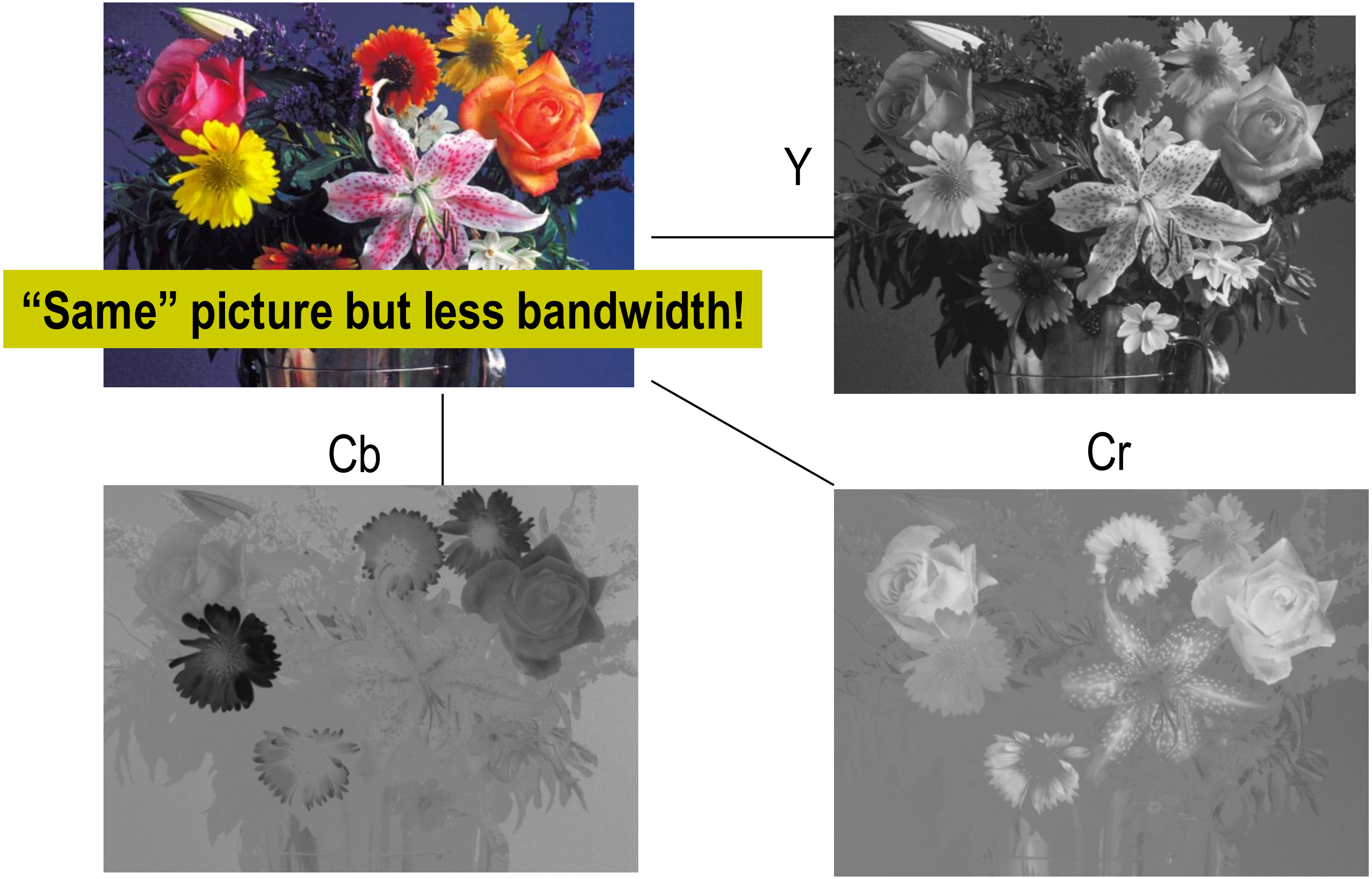
```
Cb[i] = ( -    4*(Cb[i-23]+Cb[i+23])
          +    6*(Cb[i-21]+Cb[I+21])
          -   12*(Cb[i-19]+Cb[i+19])
          +   20*(Cb[i-17]+Cb[i+17])
          -   32*(Cb[i-15]+Cb[i+15])
          +   48*(Cb[i-13]+Cb[i+13])
          -   70*(Cb[i-11]+Cb[i+11])
          +  104*(Cb[i-9]+Cb[i+9])
          -  152*(Cb[i-7]+Cb[i+7])
          +  236*(Cb[i-5]+Cb[i+5])
          -  420*(Cb[i-3]+Cb[i+3])
          + 1300*(Cb[i-1]+Cb[i+1]))/2048;
```



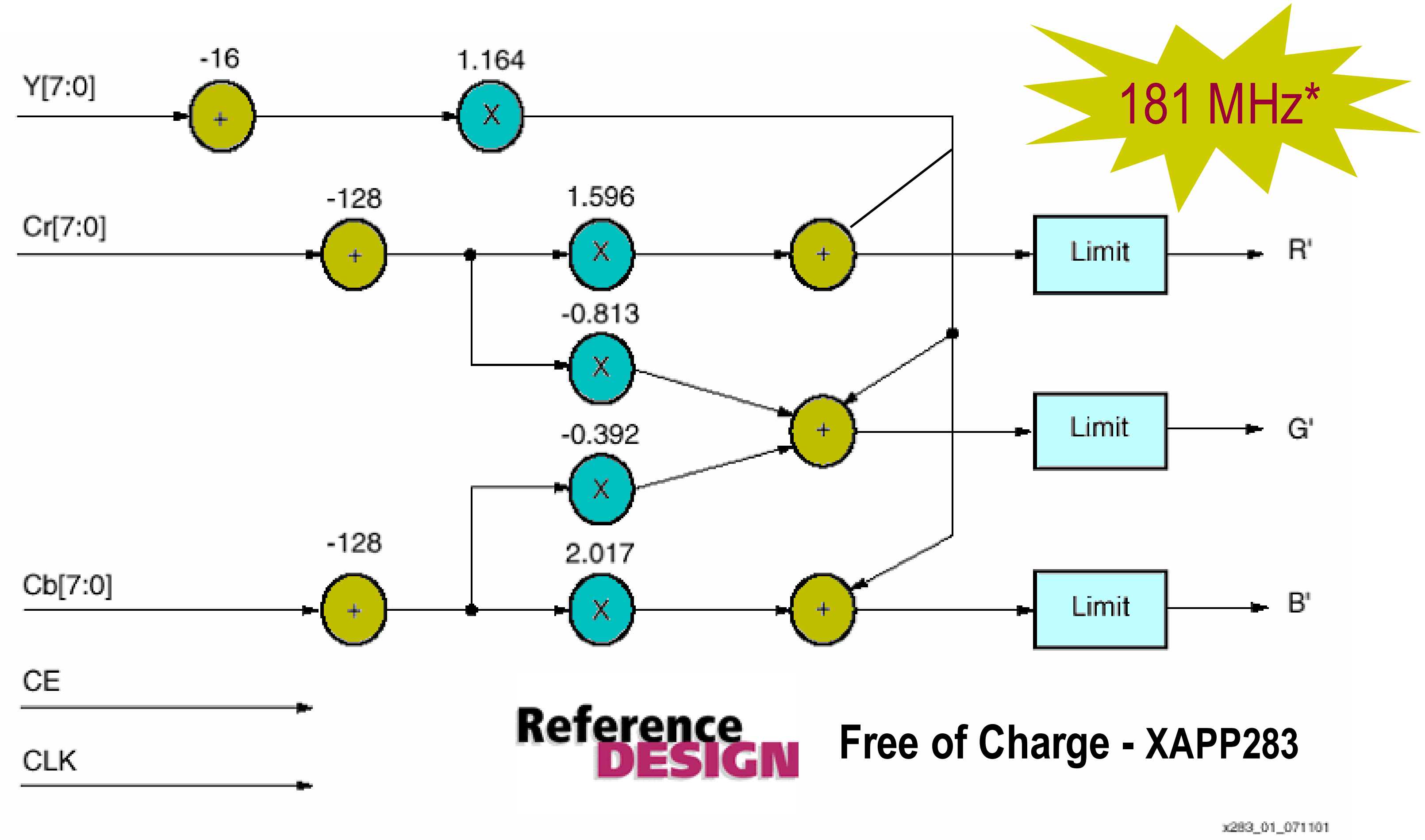
Digital Colour Images



Digital Color Images



Colour Space Converter



Reference DESIGN

Free of Charge - XAPP283

*Using embedded multipliers in XC2V1000

Colour Space Converter

for eight bit video

R' = 1.164(Y'-16) + 1.596(Cr-128)

G' = 1.164(Y'-16) + .813(Cr-128) - .392(Cb-128)

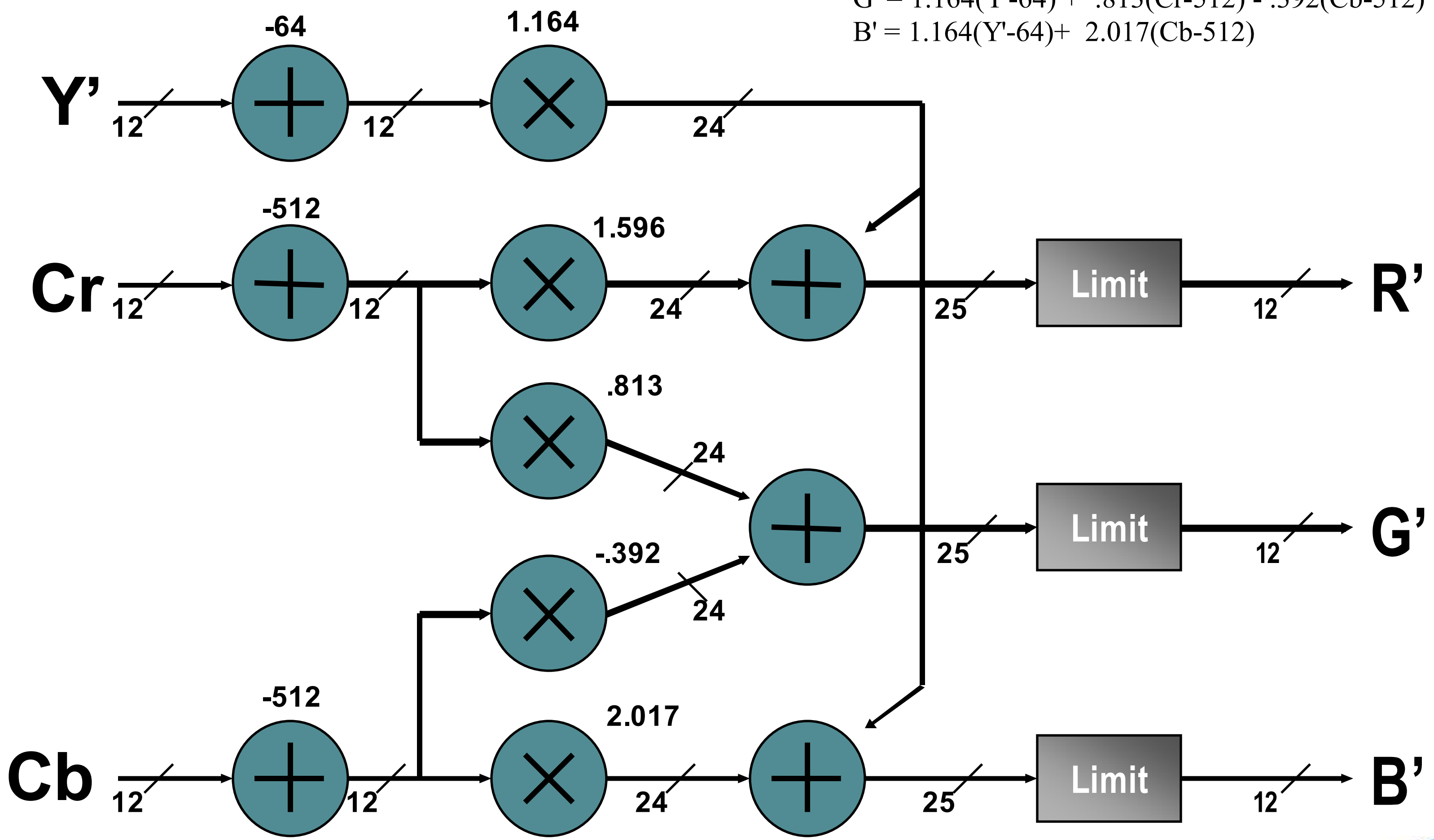
B' = 1.164(Y'-16)+ 2.017(Cb-128)

For 10 bit video

R' = 1.164(Y'-64) + 1.596(Cr-512)

G' = 1.164(Y'-64) + .813(Cr-512) - .392(Cb-512)

B' = 1.164(Y'-64)+ 2.017(Cb-512)



Colour Space Converter Simple Version (5-8%)Colour Error

for eight bit video

R' = 1.164(Y'-16) + 1.596(Cr-128)

G' = 1.164(Y'-16) + .813(Cr-128) - .392(Cb-128)

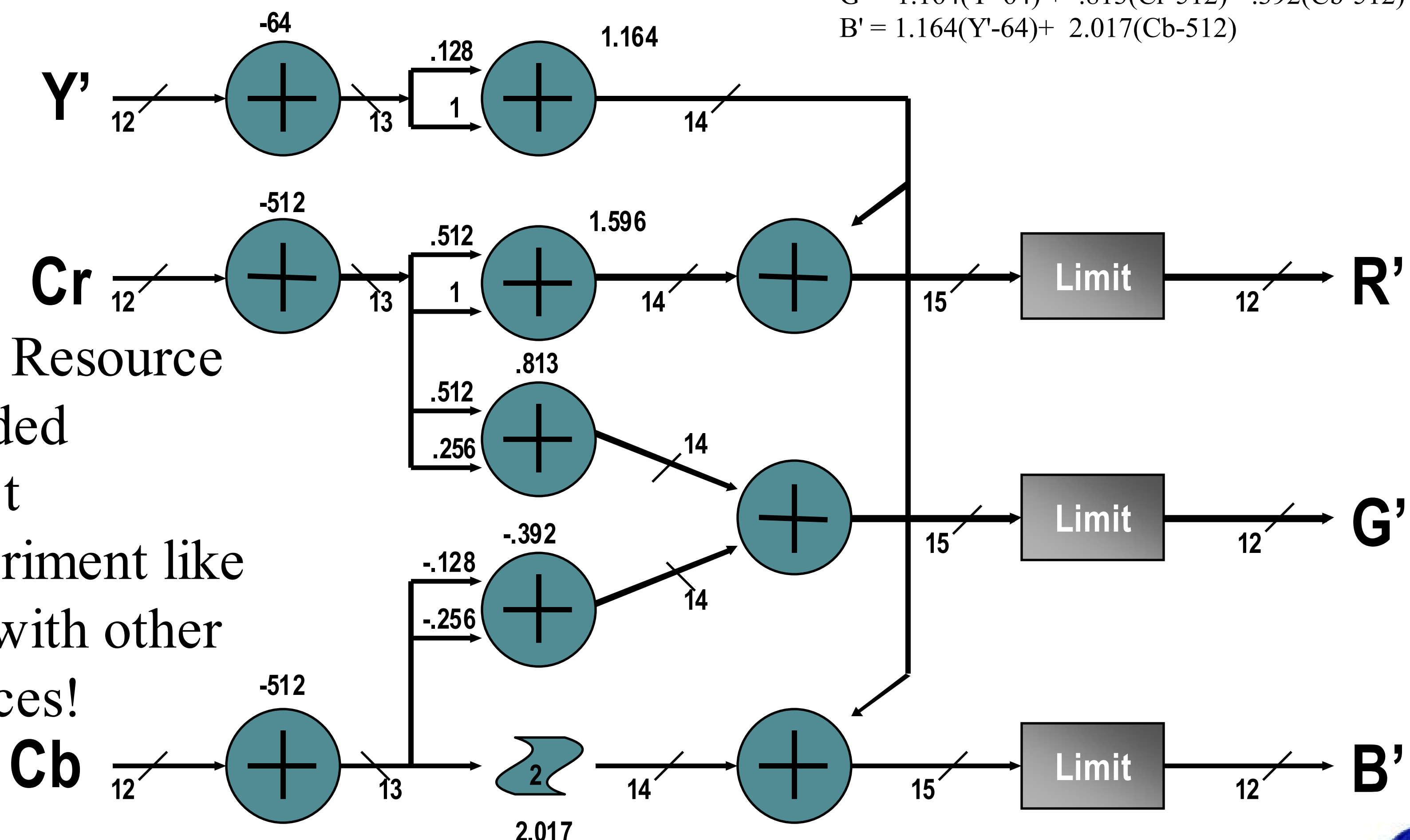
B' = 1.164(Y'-16)+ 2.017(Cb-128)

For 10 bit video

R' = 1.164(Y'-64) + 1.596(Cr-512)

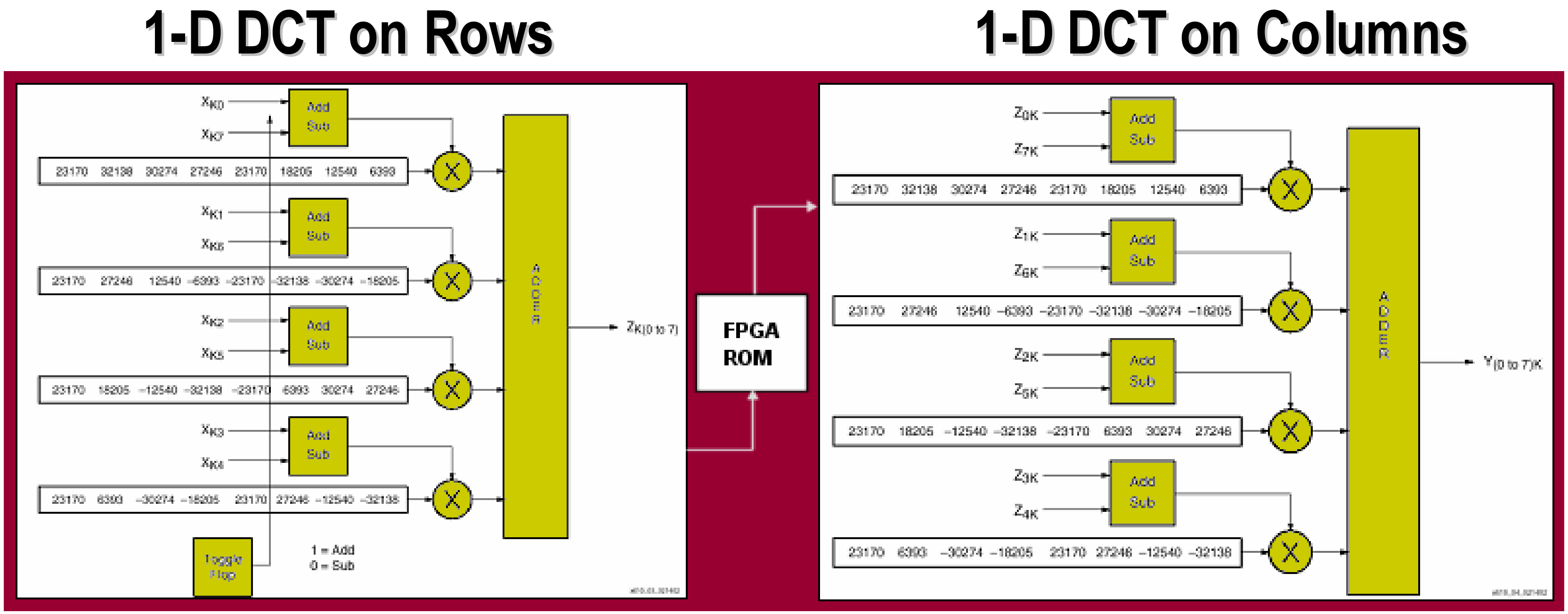
G' = 1.164(Y'-64) + .813(Cr-512) - .392(Cb-512)

B' = 1.164(Y'-64)+ 2.017(Cb-512)



Less Resource Needed Can't experiment like this with other devices!

2-D DCT Operation



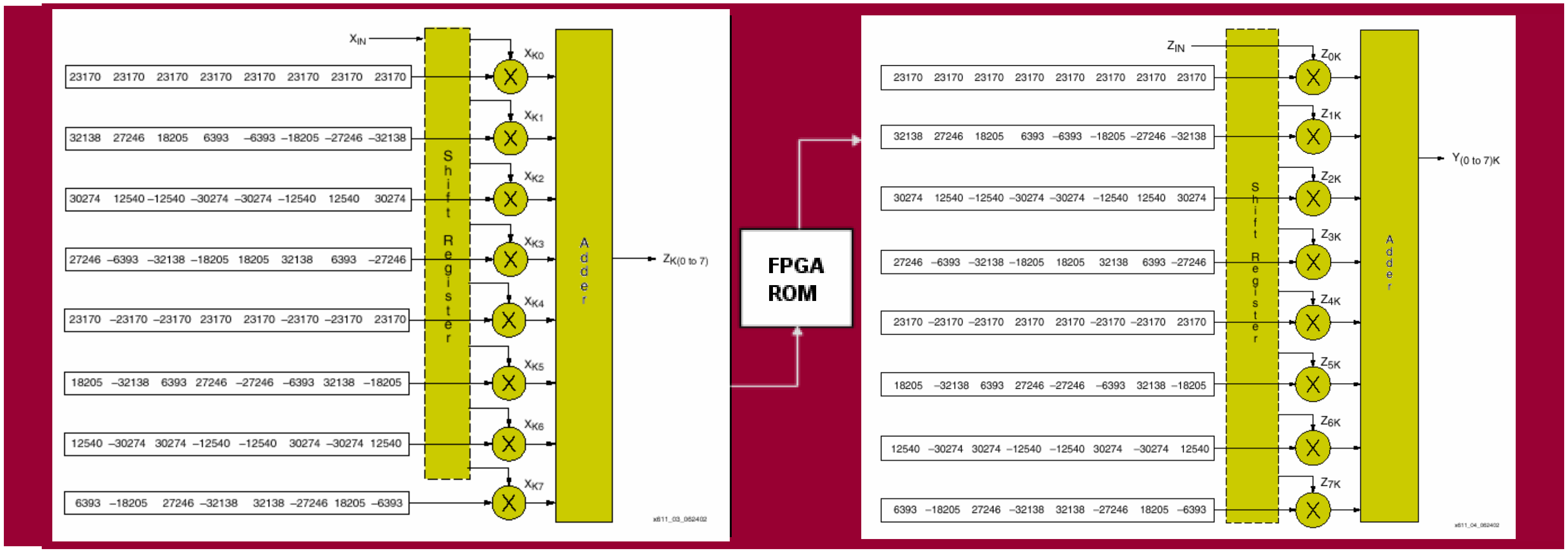
Reference DESIGN

Application Note and Reference Design Available
http://www.xilinx.com/xapp/xapp610.pdf

2-D IDCT Operation

1-D IDCT on Rows

1-D IDCT on Columns

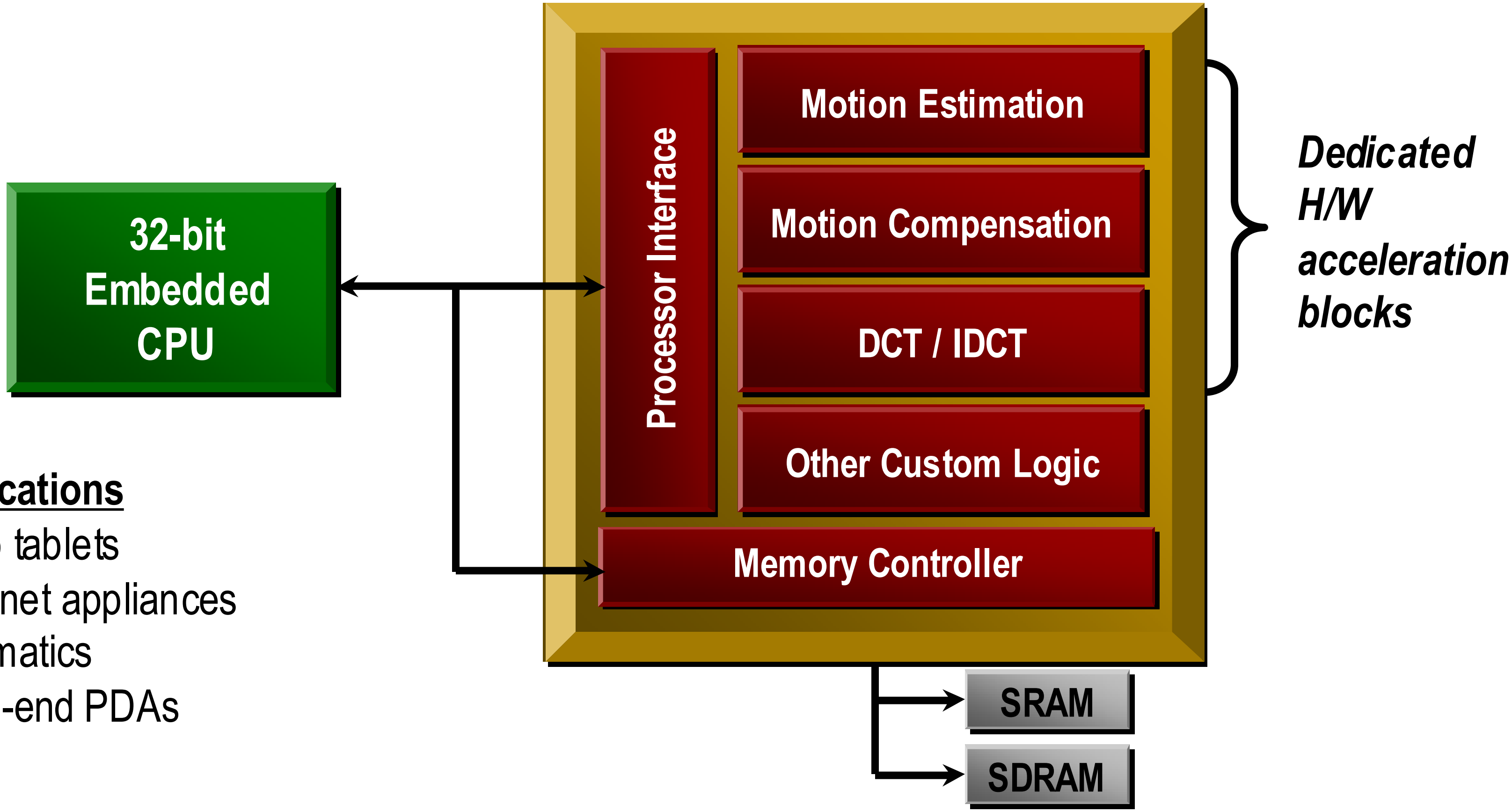


Reference DESIGN

Application Note and Reference Design Available

http://www.xilinx.com/xapp/xapp611.pdf

Partial MPEG H/W Acceleration



Applications
- Web tablets
- Internet appliances
- Telematics
- High-end PDAs

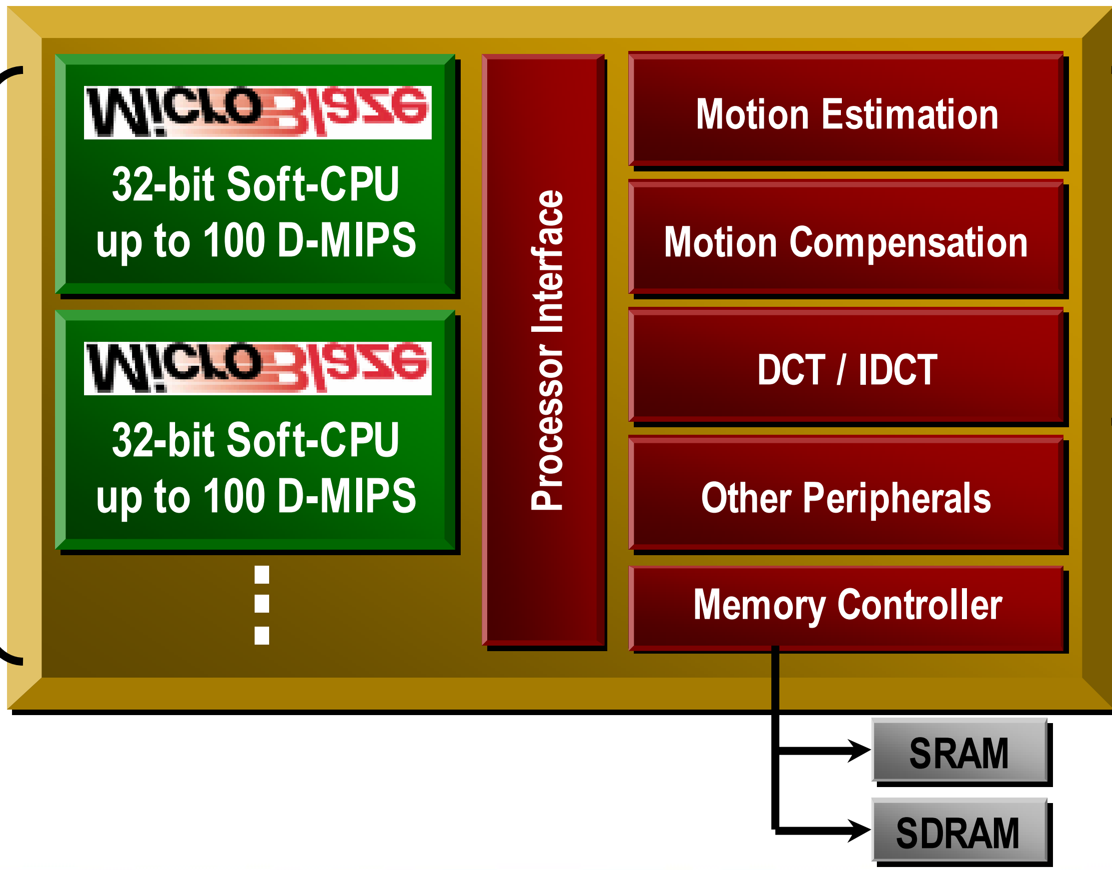
Customized MPEG Implementation

Applications

Digital TV
Plasma displays
LCD displays
Set-top boxes

Multiple-processor instantiations

Resolution, frame rate, profile, level & QoR dependent



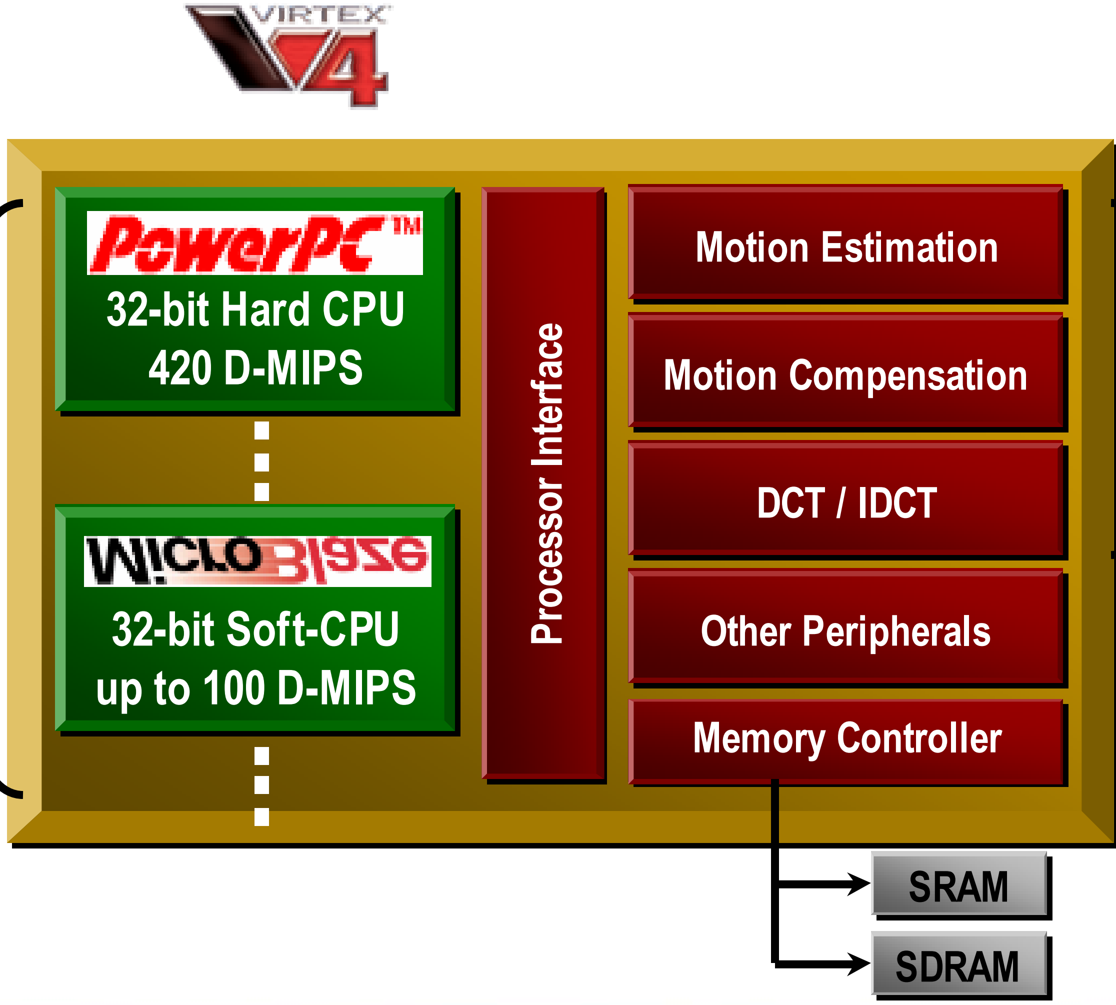
Dedicated H/W acceleration blocks

High Performance MPEG Applications

Applications
Studio applications
Digital cinema

Up to 4 PowerPC and Multiple MicroBlaze instantiations

Resolution, frame rate, profile, level & QoR dependent



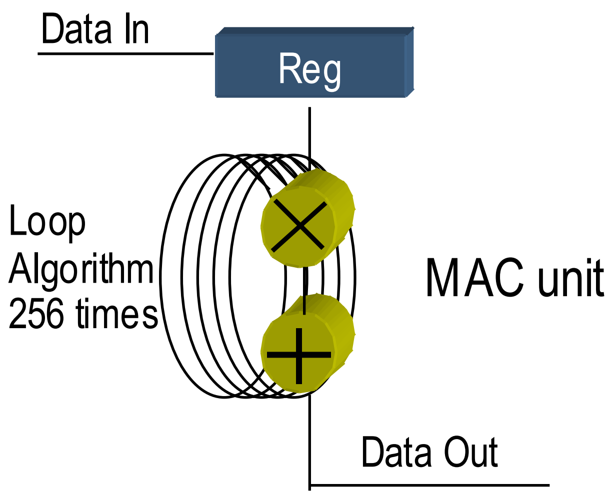
Pipelined and Parallel Dedicated H/W acceleration blocks

Xilinx Solutions

Performance Limitation of Conventional DSP

Conventional DSP Device

(Von Neumann architecture, or extensions thereof)

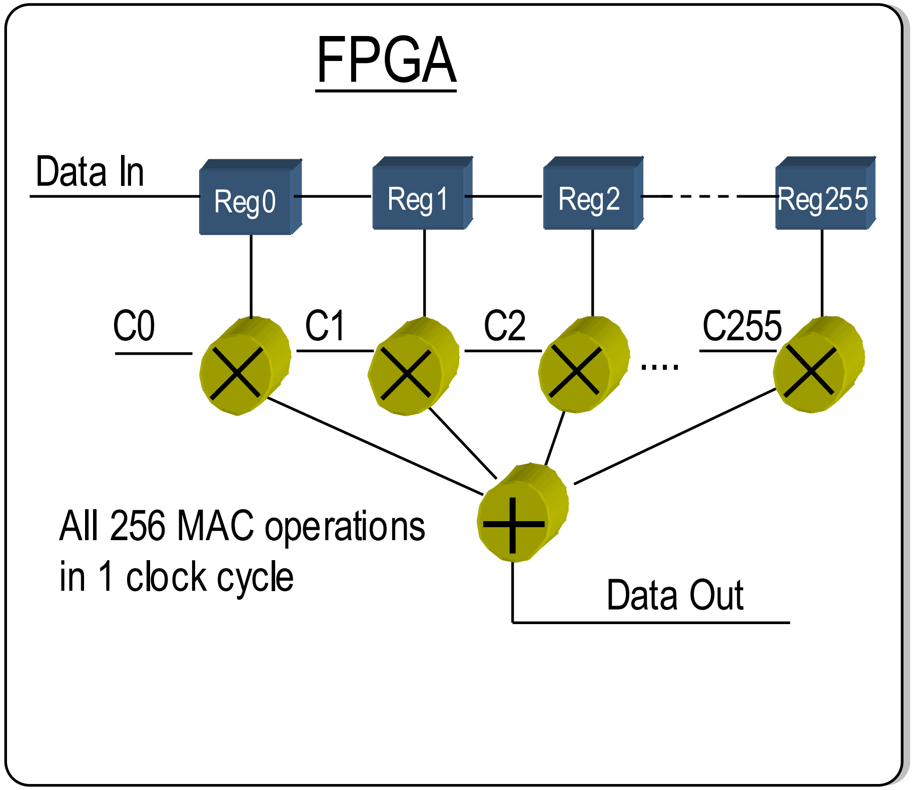


- Fixed inflexible architecture
  - Typically 1-4 MAC units
  - Fixed data width
- Serial processing limits data throughput
  - Time-shared MAC unit
  - High clock frequency creates difficult system-challenge

Example

256 Tap FIR Filter = 256 multiply and accumulate (MAC) operations per data sample

FPGA Performance Advantage

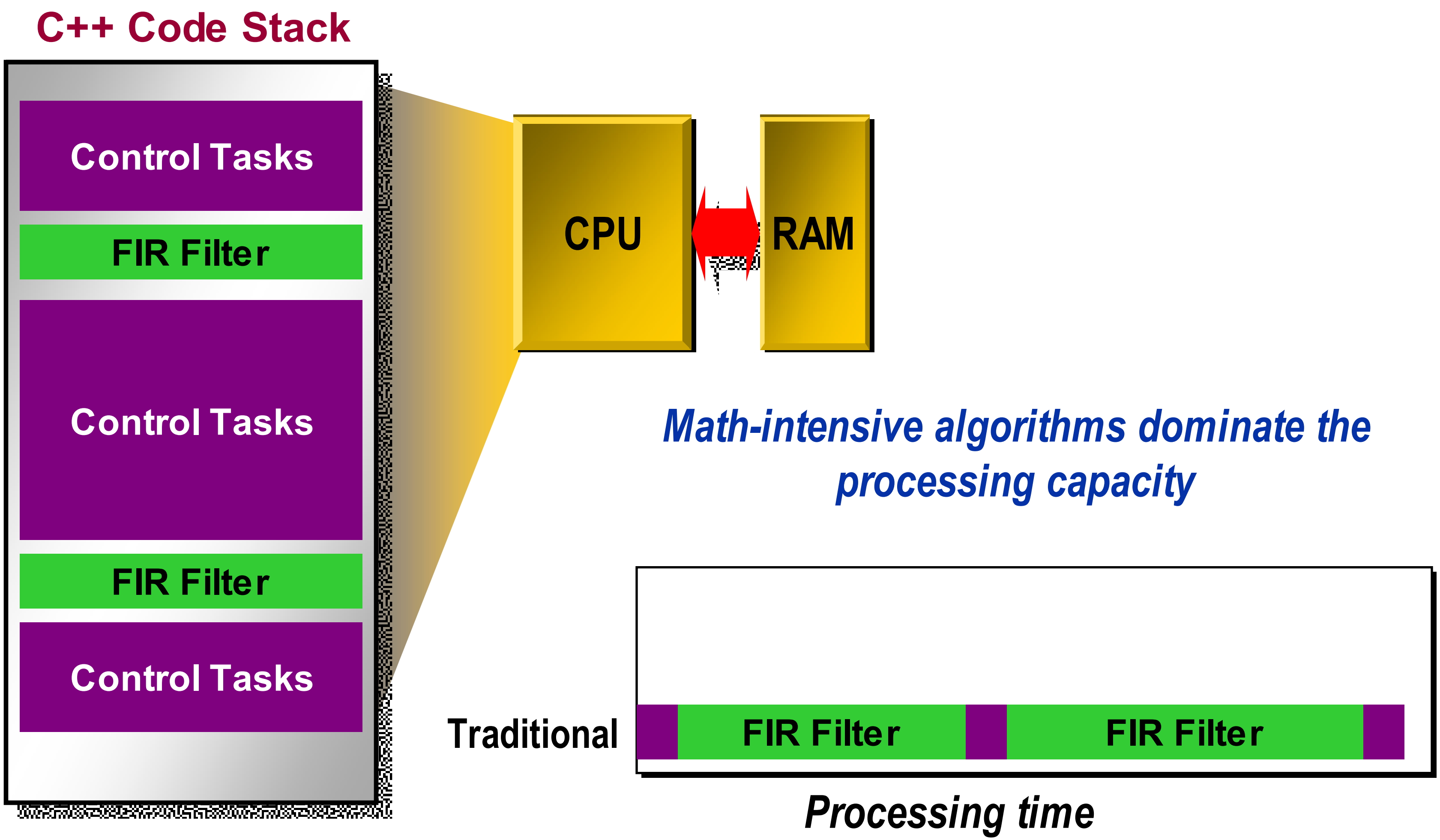


- Flexible architecture
  - Distributed DSP resources (LUT, registers, multipliers, & memory)
- Parallel processing maximizes data throughput
  - Support any level of parallelism
  - Optimal performance/cost tradeoff

XtremeDSP

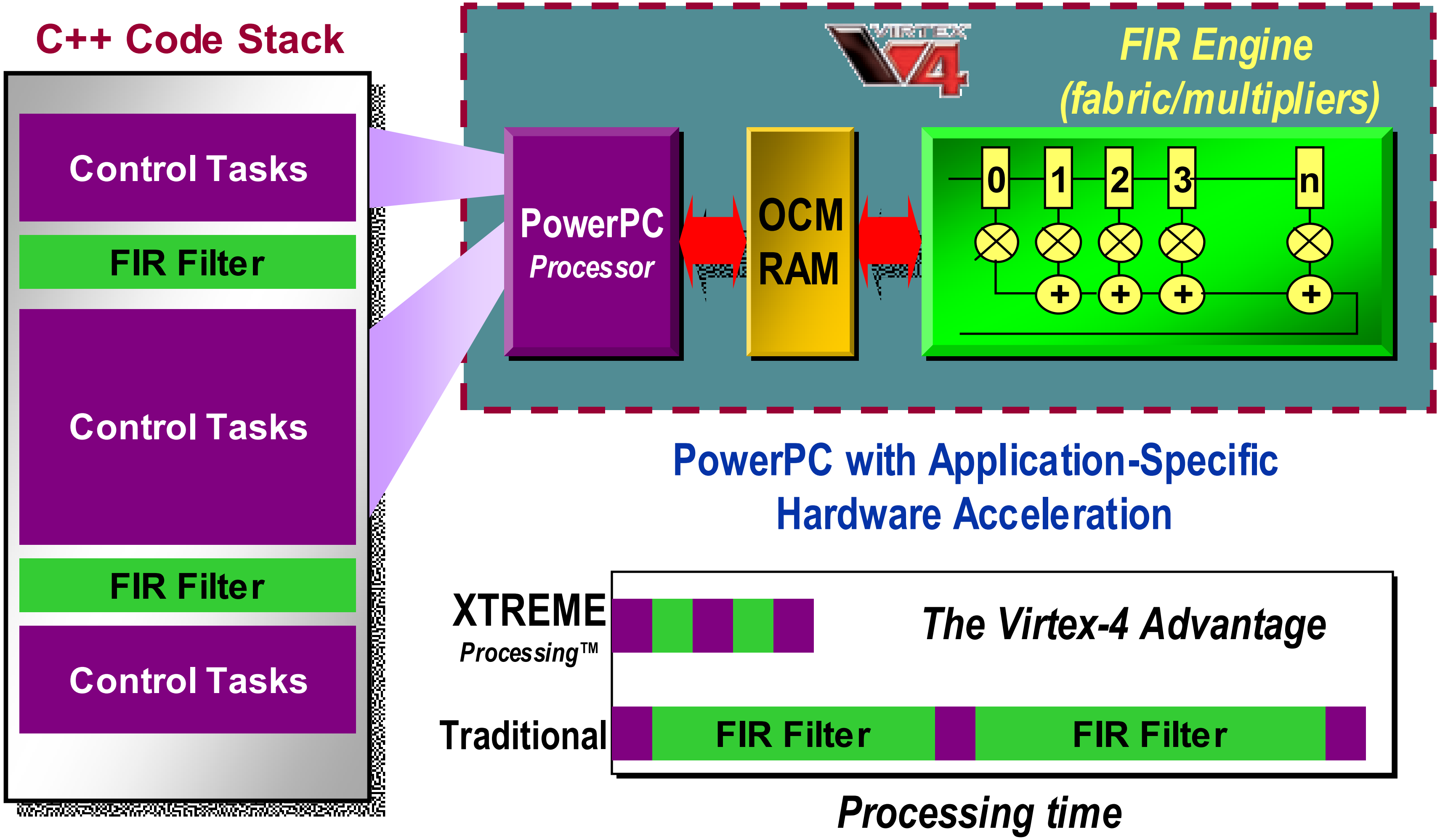
Example
256 Tap FIR Filter = 256 multiply and accumulate (MAC) operations per data sample

Traditional Processing

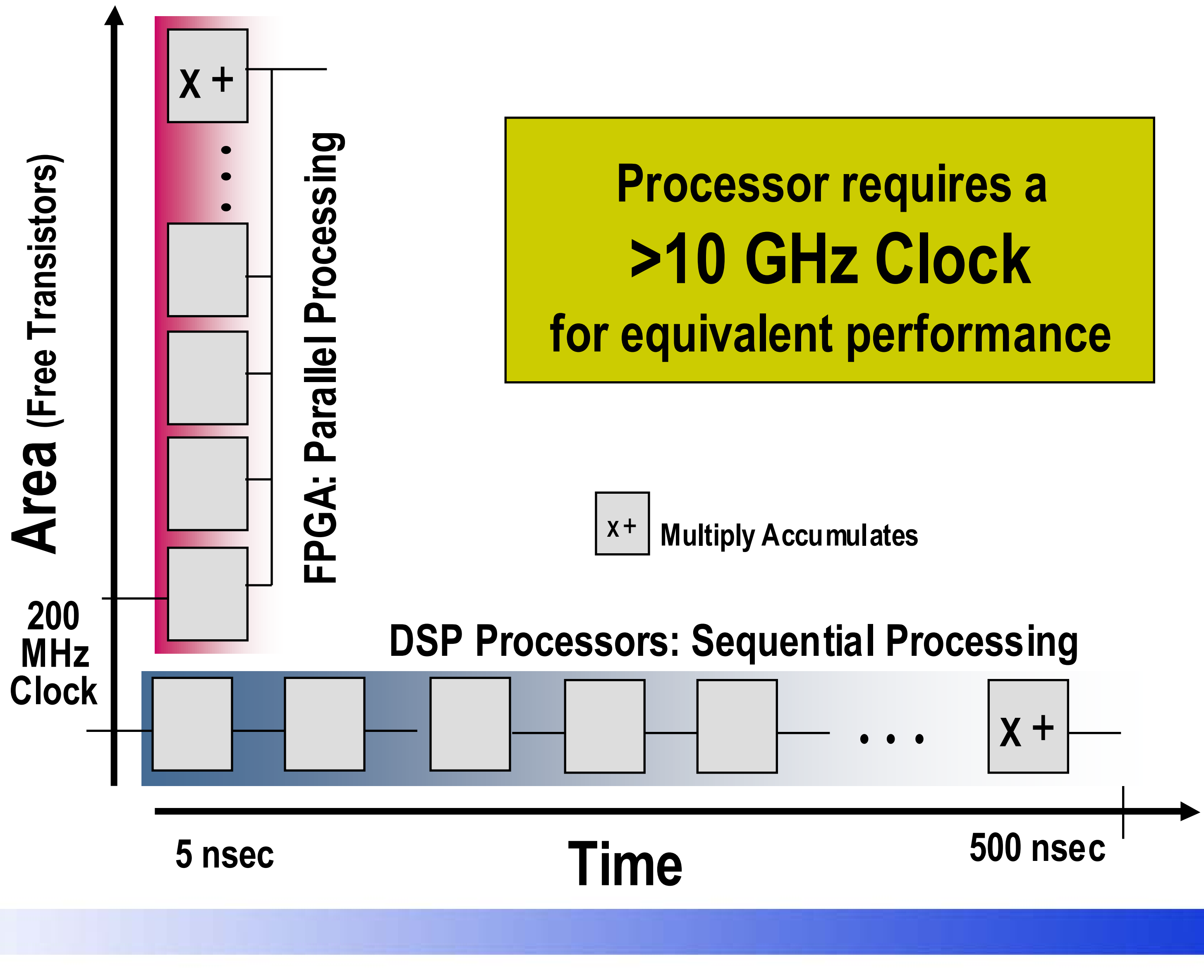


Math-intensive algorithms dominate the processing capacity

Xtreme Processing™



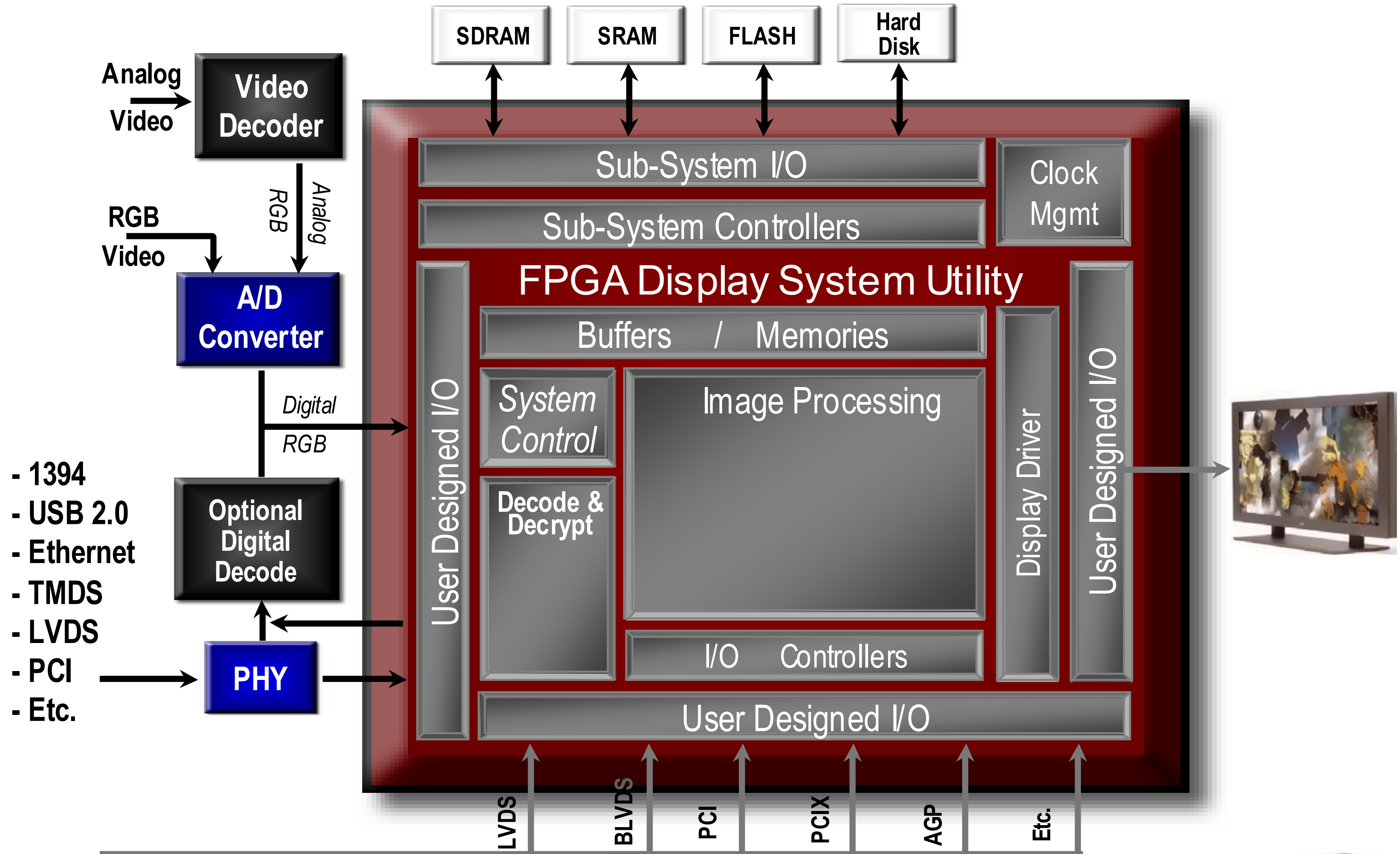
Area vs. Performance



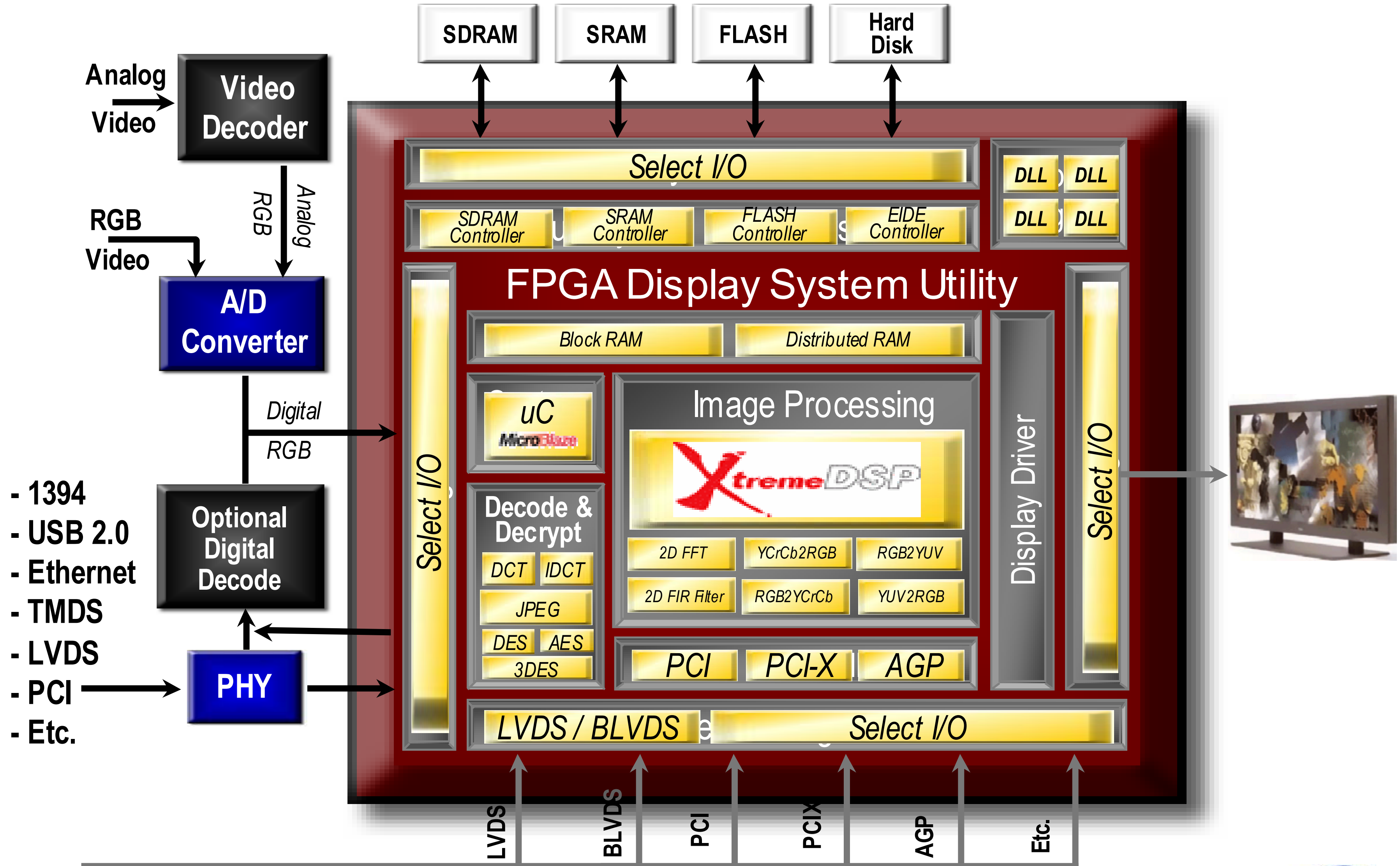
System Generator for Simulink

- Bridges gap between FPGA and DSP design flows
  - Used with Simulink®/MATLAB® from The MathWorks
- Automatically generates HDL/optimized algorithms
  - Shortens learning curve
  - HW redesign eliminated
  - Optimal implementation

Video System Design



FPGA Standard Features & IP



XILINX

Xilinx Video IP and Cores

Video and Image Processing IP	Vendor	Sign Once	Comment
1-D Discrete Cosine Transform	Xilinx	✓	
2-D DCT/IDCT Forward and Inverse Discrete Cosine Transform	Xilinx	✓	
YUV2RGB Color Space Converter	Xilinx	✓	
RGB2YCrCb Color Space Converter Core	Xilinx	✓	
RGB2YUV Color Space Converter Core	Xilinx	✓	
YCrCb2RGB Color Space Converter	Xilinx	✓	
logiCVC - Compact Video Controller	Xylon	✓	
Parameterized Symmetrical 2D FIR	Xilinx	N/A	PreLinx
Parameterized Line Buffer	Xilinx	N/A	PreLinx
Video De-Interlace	Xilinx	N/A	PreLinx
Raster To Block / Block To Raster	Xilinx	N/A	PreLinx
2-D Min/Max/Median Filters For MatLAB	Xilinx	N/A	PreLinx
2-D Discrete Wavelet Transform	Xilinx	N/A	PreLinx
2-D Discrete Wavelet Transform (3 Comps)	Xilinx	N/A	PreLinx
2D FIR	Xilinx	N/A	PreLinx
Serial Distributed Arithmetic FIR	Xilinx	N/A	LogiCore
Parallel Distributed Arithmetic FIR	Xilinx	N/A	LogiCore
Distributed Arithmetic FIR	Xilinx	N/A	LogiCore

LogiCORE

Up to date info at www.xilinx.com/ipcenter

XILINX

Video AllianceCOREs

Video and Image Processing IP	Vendor	Sign Once
Motion JPEG Decoder Core V1.0	Amphion	✓
Motion JPEG Codec Core V1.0	Amphion	✓
Motion JPEG Encoder Core V2.0	Amphion	✓
FASTJPEG_C DECODER	Barco Silex	✓
FASTJPEG_BW DECODER	Barco Silex	✓
DCT/IDCT 2D	Barco Silex	✓
HUFFD Huffman Decoder Core	CAST	✓
BB_2DDWT-Block-Based 2D Discrete Wavelet Transform	CAST	✓
LB_2DFDWT - Line-Based Programmable Forward DWT	CAST	✓
IDCT: 2D Inverse Discrete Cosine Transform	CAST	✓
RC_2DDWT: Combine 2D Forward/Inverse Discrete Wavelet Tran	CAST	✓
DCT_FI: Combined 2D Forward / Inverse Discrete Cosine Transfo	CAST	✓
DCT: 2D Forward Discrete Cosine Transform	CAST	✓
Longitudinal Time Code Generator	Deltatec	✓
JPEG CODEC	InSilicon	
YCrCb2RGB Color Space Converter	Perigree	✓
RGB2YCrCb Color Space Converter	Perigree	✓
FIDCT Forward/Inverse Discrete Cosine Transform	TILAB	

www.xilinx.com/ipcenter

Available Application Notes

XAPP248 "Digital Video Test Pattern Generators" v1.0 (01/02)
XAPP288 "Serial Digital Interface (SDI) Video Decoder" v1.0 (10/01)
XAPP298 "Serial Digital Interface (SDI) Video Encoder" v1.0 (10/01)
XAPP172 "The Design of a Video Capture Board Using the Spartan Series" v1.0 (03/99)
XAPP208 "IDCT Implementation in Virtex Devices for MPEG Applications" v 1.1 (12/99)
XAPP241 "Virtex-EM FIR Filter for Video Applications" v1.1 (10/00)
XAPP284 "3 x 3 Matrix Multiplier for 3D Graphics and Video" v1.1 (10/01)
XAPP283 "Color Space Conversion" v1.0 (07/01)
XAPP219 "Transposed Form FIR Filters" v1.2 10/01
XAPP270 "High-Speed DES and Triple DES Encryptor/Decryptor" v1.0 (8/01)

support.xilinx.com/apps/appsweb.htm

Key FPGA Features for Video

- For SDTV (13.5 MHz)
  - LUTs, SRL16s and FFs for Distributed Math, pipelines
- For HDTV (74.25 MHz)
  - Inferred MULT_AND 8x8 or 10x10 Multipliers are Efficient.
- Virtex-II embedded multiplier can be used to save other logic or in compression where multiplies need 24 bit accuracy
- Block RAM useful for Line Buffers and Line Fifos
  - SDTV - 858 x 24 bits, 858 x 30
  - HDTV - 1920 x 24 or 1920 x 30
- High Speed Serial IO (LVDS, LVPECL_EXT, or Rocket IO) for high speed video transmission
- DCM allows easy generation of a bit rate clock for distributed arithmetic
- MicroBlaze and PicoBlaze for protocol layers in compression and other video algorithms

FPGAs for HD Digital Video

Spartan-IIE Silicon Features	Value for Digital Video Applications
FPGA Fabric and Routing, Up to 300,000 System Gates	Performance in excess of 30 billion MACs/second
Delay Locked Loops (DLLs)	Clock multiplication and division, clock mirror, Improve I/O Perf.
SelectIO - HSTL-I, -III, -IV	High-speed SRAM interface
SelectIO - SSTL3-I, -II; SSTL2-I, -II	High-speed DRAM interface
SelectIO - GTL, PCI, AGP	Chip-to-Backplane, Chip-to-Chip interfaces
Differential Signaling - LVDS, Bus LVDS, LVPECL	Bandwidth management (saving the number of pins), reduced power consumption, reduced EMI, high noise immunity
SRL-16	16-bit Shift Register ideal for capturing high-speed or burst-mode data and to store data in DSP applications
Distributed RAM	DSP Coefficients, Small FIFOs
Block RAM	Video Line Buffers, Cache Tag Memory, Scratch-pad Memory, Packet Buffers, Large FIFOs

The Best of Both Worlds

- Off-the-shelf devices
- Faster time-to-market
- Rapid adoption of standards
- Real time prototyping

Flexibility of DSP Processors

- Parallel processing
- Support high data rates
- Optimal bit widths
- No real-time software coding

Performance of Custom ICs

Xilinx DSP Solutions Offer the Best of Both Worlds With Low Cost!

XtremeDSP for Video

- Unrivalled DSP Performance
  - TeraMAC/s via FPGA and Embedded Multiplier fabric for:
    - Multimedia compression - MPEG2, MPEG4, H.26L, MJPEG, JPEG2000
    - Video processing - Integrated line buffers, enhancement, pattern recognition, noise reduction, resizing, rotation, scalability
    - Convergence of emerging technologies in multimedia over IP & wireless
- For Standard Definition Pixel Rates (13.5 MHz pixels)
    - SDTV test equipment, broadcast test equipment, studio effects equipment, scan rate converters, frame rate converters, MPEG-2 codecs
- For High Definition Pixel Rates or Multiple Channels of Standard Definition (74.25 MHz pixels)
    - HDTV test equipment, broadcast test equipment, home theater projection devices, advanced studio effects, conversions from SDTV, MPEG-2 4:2:2 profile codecs

Xilinx Video Solutions

- Enables the designer to add real value to his system
  - Allows for experimentation in development that leads to differentiation in production
  - Chipsets that support your *exact* requirements are never available!!
- Supports high definition real-time processing
  - Allows for hardware acceleration of key algorithms
  - More information down the pipe
  - Less memory requirements for off-line processing
- System on a chip integration
  - More channels on less chips
  - Saves valuable board space and can reduce overall BOM cost

XILINX®

Questions?

espteam@xilinx.com