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
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. Luma samples shown here (Chroma processed separately)
5. Convert to frequency components (DCT)
6. Output DCT coefficients
7. Quantize higher frequencies with less bits (weighting)
8. Zero values for frequencies below perception threshold

Compress using zigzag scan and run length encoding for zero values (in blue)
Further compression with Huffman encoding (VLC)
Human eye less sensitive to high frequencies

Spatial Frequency

Sensitivity

Luma samples shown here (Chroma processed separately)
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

GOP (Group Of 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

- 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

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

- 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

\[ S_x = Dx \cos(\theta) + Dy \sin(\theta) \]
\[ S_y = -Dx \sin(\theta) + Dy \cos(\theta) \]
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

00 = BLACK SCREEN
01 = BLUE SCREEN
10 = COLOR BARS
11 = NORMAL VIDEO

Courtesy: Keith Jack, Video Demystified
Spatial Enhancement Illustration

Medical Image Enhancement

Enhanced 1D Signal

Gaussian Enhances Better

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
Video Bypass

ADV7185 NTSC/PAL DECODER

CCIR 601 / 656 4:2:2 format

VIDEO-IN RAM

ADV7194 NTSC/PAL ENCODER

CCIR 601 / 656 4:2:2 format

VIDEO-OUT RAM

MicroBlaze or PicoBlaze

Virtex V4

Y'Cr'Cb' or R'G'B'
4:4:4 progressive

Static Image Previously Loaded From Live Video

FMS3810 Triple 8bit DAC

R'G'B'
4:4:4 progressive

TV PAL/NTSC

PC

4:4:4 progressive
Video Ping-Pong (1)

ADV7185 NTSC/PAL DECODER
- CCIR 601 / 656 4:2:2 format
- Video-in RAM
  - 4:2:2: 444
- Deinterlace
- Crossbar Switch & DMA Control

ADV7194 NTSC/PAL ENCODER
- CCIR 601 / 656 4:2:2 format
- Video-out RAM
  - 4:4:4: 422
- Interlace
- R'G'B' 4:4:4 progressive

FMS3810 Triple 8bit DAC

MicroBlaze or PicoBlaze

Static Image
Previously Loaded From Live Video

Outgoing Video Frame Number N
Outgoing Video Frame Number N+1
Outgoing Video Frame Number N+2
Outgoing Video Frame Number N-1

Y'Cr'Cb' or R'G'B'
4:4:4 progressive

TV PAL/NTSC

PC
Video Ping-Pong (2)

ADV7185 NTSC/PAL DECODER
CCIR 601 / 656 4:2:2 format
Outgoing Video Frame Number N
VIDEO-IN RAM

422:444 Deinterlace
MicroBlaze or PicoBlaze

444:422 Interlace
Y'Cr'Cb' or R'G'B'

Outgoing Video Frame Number N+1

Outgoing Video Frame Number N+2

FMS3810 Triple 8bit DAC
R'G'B'
4:4:4 progressive

Outgoing Video Frame Number N-2

Outgoing Video Frame Number N-3

VIDEO-OUT RAM

ADV7194 NTSC/PAL ENCODER
CCIR 601 / 656 4:2:2 format
Outgoing Video Frame

Static Image Previously Loaded From Live Video

PC

TV PAL/NTSC

MicroBlaze or PicoBlaze

Y'Cr'Cb' or R'G'B'
4:4:4 progressive

Outgoing Video Frame Number N

Outgoing Video Frame Number N-1

Outgoing Video Frame Number N-2

Outgoing Video Frame Number N-3

Static Image Previously Loaded From Live Video

FMS3810 Triple 8bit DAC
R'G'B'
4:4:4 progressive
Video Chromakey

ADV7185
NTSC/PAL DECODER
CCIR 601 / 656 4:2:2 format
Outgoing Video Frame Number N
VIDEO-IN RAM
Outgoing Video Frame Number N-1

MicroBlaze or PicoBlaze

if A = green then C=B else C=A

Y'Cr'Cb' or R'G'B'
4:4:4 progressive

ADV7194
NTSC/PAL ENCODER
CCIR 601 / 656 4:2:2 format
Outgoing Video Frame Number N-3
VIDEO-OUT RAM
Outgoing Video Frame Number N-2

FMS3810
Triple 8bit DAC
R'G'B'
4:4:4 progressive

Static Image Previously Loaded From Live Video

Outgoing Video Frame Number N
Outgoing Video Frame Number N-2
Outgoing Video Frame Number N-3

Cross Bar Switch & DMA Control

422:444 Deinterlace
444:422 Interlace
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

<table>
<thead>
<tr>
<th>IP Core or Reference Design</th>
<th>Provider</th>
</tr>
</thead>
<tbody>
<tr>
<td>XAPP219 Transposed Form FIR Filters</td>
<td>Xilinx Inc.</td>
</tr>
<tr>
<td>MAC FIR</td>
<td>Xilinx Inc.</td>
</tr>
<tr>
<td>Serial Distributed Arithmetic FIR Filter</td>
<td>Xilinx Inc.</td>
</tr>
<tr>
<td>Parallel Distributed Arithmetic FIR Filter</td>
<td>Xilinx Inc.</td>
</tr>
<tr>
<td>Distributed Arithmetic FIR Filter</td>
<td>Xilinx Inc.</td>
</tr>
</tbody>
</table>

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

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

Software simulation shown
Noise Removal

- Removal of Gaussian noise

Software simulation shown
Noise Removal

- Salt and Pepper noise removal

Software simulation shown
Noise Reduction

- Noise reduction using temporal filtering across multiple video frames
Temporal Filter

- \( X(k) \) to Frame Buffer
- Frame Buffer to FPGA
- FPGA to Video Buffer
- \( Y(k) \) from Video Buffer

- \( Y(k-1) \) BUF
- \( \sigma^2 \) BUF
- \( \sigma^2_w \) BUF
Image Enhancement Algorithms

• The list is 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

\[(H \times \text{Original}) - (K \times \text{Low_pass_filter}) = \text{Enhanced Image}\]

Enhanced Video Out
Spatial Enhancement

Original, 33%, 66%, 100% Boost
Coherence Domains

Horizontal

Vertical

Spatial

Temporal
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

Flip Flops, SRL16s or Distributed RAM

Horizontal Coherence

Temporal Coherence

Vertical Coherence

True-Dual Port™ Synchronous Block RAM

High-Performance External Memory Interfaces

DDR SDRAM

ZBT® SRAM

QDR SRAM
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

LUT4

INIT=1234

Becomes

SRL16E

D
CE
A3
A2
A1
A0

INIT=1234

A[3:0] 0000 1111

CE
D

Q
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

Even Field (Field Two)

- Line 284
- Line 285
- Line 286
- Line 525

Odd Field (Field One)

- Line 21
- Line 22
- Line 23
- Line 262
- Line 263
NTSC Horizontal Scan Line

- 1 Volt peak to peak
- Horiz. Sync
- Back Porch
- 16 samples
- Digital Line Start
- 122 samples
- 720 samples (0-719)
- 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)
- H, V, and F Transition Here
- Transition Here
- 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)

NTSC line 283, Field 2
NTSC line 21, Field 1
NTSC line 284, Field 2
NTSC line 22, Field 1
NTSC line 285, Field 2
NTSC line 23, Field 1
NTSC line 286, Field 2
NTSC line 24, Field 1

Progressive Line 1
Progressive Line 2
Progressive Line 3
Progressive Line 4
Progressive Line 5
Progressive Line 6
Progressive Line 7

NTSC line 522, Field 2
NTSC line 260, Field 1
NTSC line 523, Field 2
NTSC line 261, Field 1
NTSC line 524, Field 2
NTSC line 262, Field 1
NTSC line 525, Field 2
NTSC line 263, Field 1

Progressive Line 478
Progressive Line 479
Progressive Line 480
Progressive Line 481
Progressive Line 482
Progressive Line 483
Progressive Line 484

SMPTE 170M FIELD 1
242 1/2 lines
SMPTE 170M FIELD 2
242 1/2 lines

484 total active lines
485 total active lines
Scan Line Deinterlacing

Using Line Duplication

Pixel adr

Scan Line N-2

pix_wadr  pix_radr
Pixel in  Pixel out
Line Buffer A

Scan Line N

FIFO

wire shift

FIFO

Scan Line N-1

To ZBT

Line Select

XILINX®
Scan Line Deinterlacing

Using Line Interpolation from 2 Lines

\[( \text{Pixel A} + \text{Pixel B} ) / 2 \]
Scan Line Deinterlacing

Using Line Interpolation from 4 Lines

- Scan Line N-4
- Scan Line N-3
- Phantom Line N-2
- Scan Line N-1
- Scan Line N

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!**

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

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

ATSC “Table III”
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. Luma samples shown here (Chroma processed separately)

5. Convert to frequency components (DCT)
6. Output DCT coefficients
7. 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)

Spatial Frequency

Sensitivity

Human eye less sensitive to high frequencies
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

- 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

<table>
<thead>
<tr>
<th>16x16 Macroblock</th>
<th>Luma (Y) Sample Points</th>
<th>Chroma (CrCb) Sample Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>4:2:2</td>
<td><img src="" alt="Diagram" /></td>
<td><img src="" alt="Diagram" /></td>
</tr>
<tr>
<td></td>
<td>Only one horizontal CrCb value used for every Y</td>
<td></td>
</tr>
<tr>
<td>4:2:0</td>
<td><img src="" alt="Diagram" /></td>
<td><img src="" alt="Diagram" /></td>
</tr>
<tr>
<td></td>
<td>Only interpolated CrCb values used</td>
<td></td>
</tr>
</tbody>
</table>

Easy in an FPGA!
Digital Component Video Conversion - 4:4:4 to 4:2:2

• XAPP294
Digital Component Video Conversion - 4:4:4 to 4:2:2

- XAPP294

This is 4:2:2
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

• \( \frac{1}{2} (A + C) \)
Digital Component Video Conversion - 4:2:2 to 4:4:4

- XAPP294

Previous method may need bandwidth limiting, this version is better

- Less multiplies: $\frac{1}{6} (A + D) + \frac{1}{3} (B + C)$
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

“Same” picture but less bandwidth!

Y

Cb

Cr
Colour Space Converter

Y[7:0] → -16 + 1.164 X
Cr[7:0] → -128 + 1.596 X -0.813
Cb[7:0] → -128 * 2.017 X

Limit

R'
G'
B'

Reference DESIGN
Free of Charge - XAPP283

*Using embedded multipliers in XC2V1000
for eight bit video
\[ R' = 1.164(Y' - 16) + 1.596(Cr - 128) \]
\[ G' = 1.164(Y' - 16) + 0.813(Cr - 128) - 0.392(Cb - 128) \]
\[ B' = 1.164(Y' - 16) + 2.017(Cb - 128) \]

For 10 bit video
\[ R' = 1.164(Y' - 64) + 1.596(Cr - 512) \]
\[ G' = 1.164(Y' - 64) + 0.813(Cr - 512) - 0.392(Cb - 512) \]
\[ B' = 1.164(Y' - 64) + 2.017(Cb - 512) \]
Colour Space Converter
Simple Version
(5-8%) Colour Error

for eight bit video
\[ R' = 1.164(Y' - 16) + 1.596(Cr - 128) \]
\[ G' = 1.164(Y' - 16) + 0.813(Cr - 128) - 0.392(Cb - 128) \]
\[ B' = 1.164(Y' - 16) + 2.017(Cb - 128) \]

For 10 bit video
\[ R' = 1.164(Y' - 64) + 1.596(Cr - 512) \]
\[ G' = 1.164(Y' - 64) + 0.813(Cr - 512) - 0.392(Cb - 512) \]
\[ B' = 1.164(Y' - 64) + 2.017(Cb - 512) \]

Less Resource Needed
Can’t experiment like this with other devices!
2-D DCT Operation

1-D DCT on Rows

1-D DCT on Columns

Application Note and Reference Design Available
2-D IDCT Operation

1-D IDCT on Rows

1-D IDCT on Columns

Application Note and Reference Design Available
Partial MPEG H/W Acceleration

32-bit Embedded CPU

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

Processor Interface
- Motion Estimation
- Motion Compensation
- DCT / IDCT
- Other Custom Logic

Memory Controller

Dedicated H/W acceleration blocks

SRAM
SDRAM
Customized MPEG Implementation

Applications
- Digital TV
- Plasma displays
- LCD displays
- Set-top boxes

Multiple-processor instantiations

Resolution, frame rate, profile, level & QoR dependent

32-bit Soft-CPU up to 100 D-MIPS

Motion Estimation
Motion Compensation
DCT / IDCT
Other Peripherals
Memory Controller

2-processor instantiations

Dedicated H/W acceleration blocks

Applications
- Digital TV
- Plasma displays
- LCD displays
- Set-top boxes

32-bit Soft-CPU up to 100 D-MIPS

Processor Interface

SRAM
SDRAM
High Performance MPEG Applications

Applications
- Studio applications
- Digital cinema

Up to 4
- PowerPC and Multiple MicroBlaze instantiations

Resolution, frame rate, profile, level & QoR dependent

32-bit Hard CPU
420 D-MIPS

32-bit Soft-CPU
up to 100 D-MIPS

Motion Estimation
Motion Compensation
DCT / IDCT
Other Peripherals
Memory Controller

Processor Interface

SRAM
SDRAM

Pipelined and Parallel Dedicated H/W acceleration blocks
Xilinx Solutions
Performance Limitation of Conventional DSP

- 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

Example
256 Tap FIR Filter = 256 multiply and accumulate (MAC) operations per data sample
Math-intensive algorithms dominate the processing capacity.
Xtreme Processing™

C++ Code Stack

Control Tasks

FIR Filter

Control Tasks

FIR Filter

PowerPC Processor

OCM RAM

FIR Engine (fabric/multipliers)

PowerPC with Application-Specific Hardware Acceleration

XTREME Processing™

The Virtex-4 Advantage

Traditional FIR Filter

FIR Filter

Processing time
Area vs. Performance

Processor requires a >10 GHz Clock for equivalent performance

DSP Processors: Sequential Processing

FPGA: Parallel Processing

200 MHz Clock

Area (Free Transistors)

Time

5 nsec

500 nsec
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

- Analog Video
- RGB Video
- A/D Converter
- Optional Digital Decode
- FPGA Display System Utility
- System Control
- Image Processing
- Buffers / Memories
- I/O Controllers
- User Designed I/O

- SDRAM
- SRAM
- FLASH
- Hard Disk
- Sub-System I/O
- Sub-System Controllers
- Clock Mgmt

- 1394
- USB 2.0
- Ethernet
- TMDS
- LVDS
- PCI
- Etc.
# Xilinx Video IP and Cores

<table>
<thead>
<tr>
<th>Video and Image Processing IP</th>
<th>Vendor</th>
<th>Sign Once</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-D Discrete Cosine Transform</td>
<td>Xilinx</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>2-D DCT/IDCT Forward and Inverse Discrete Cosine Transform</td>
<td>Xilinx</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>YUV2RGB Color Space Converter</td>
<td>Xilinx</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>RGB2YCrCb Color Space Converter Core</td>
<td>Xilinx</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>RGB2YUYV Color Space Converter Core</td>
<td>Xilinx</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>YCrCb2RGB Color Space Converter</td>
<td>Xilinx</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>logiCVC - Compact Video Controller</td>
<td>Xylon</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Parameterized Symmetrical 2D FIR</td>
<td>Xilinx</td>
<td>N/A</td>
<td>PreLinx</td>
</tr>
<tr>
<td>Parameterized Line Buffer</td>
<td>Xilinx</td>
<td>N/A</td>
<td>PreLinx</td>
</tr>
<tr>
<td>Video De-Interlace</td>
<td>Xilinx</td>
<td>N/A</td>
<td>PreLinx</td>
</tr>
<tr>
<td>Raster To Block / Block To Raster</td>
<td>Xilinx</td>
<td>N/A</td>
<td>PreLinx</td>
</tr>
<tr>
<td>2-D Min/Max/Median Filters For MatLAB</td>
<td>Xilinx</td>
<td>N/A</td>
<td>PreLinx</td>
</tr>
<tr>
<td>2-D Discrete Wavelet Transform</td>
<td>Xilinx</td>
<td>N/A</td>
<td>PreLinx</td>
</tr>
<tr>
<td>2-D Discrete Wavelet Transform (3 Comps)</td>
<td>Xilinx</td>
<td>N/A</td>
<td>PreLinx</td>
</tr>
<tr>
<td>2D FIR</td>
<td>Xilinx</td>
<td>N/A</td>
<td>PreLinx</td>
</tr>
<tr>
<td>Serial Distributed Arithmetic FIR</td>
<td>Xilinx</td>
<td>N/A</td>
<td>LogiCore</td>
</tr>
<tr>
<td>Parallel Distributed Arithmetic FIR</td>
<td>Xilinx</td>
<td>N/A</td>
<td>LogiCore</td>
</tr>
<tr>
<td>Distributed Arithmetic FIR</td>
<td>Xilinx</td>
<td>N/A</td>
<td>LogiCore</td>
</tr>
</tbody>
</table>

Up to date info at www.xilinx.com/ipcenter
# Video AllianceCOREs

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

[www.xilinx.com/ipcenter](www.xilinx.com/ipcenter)
<table>
<thead>
<tr>
<th>Application Note</th>
<th>Title</th>
<th>Version</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>XAPP248</td>
<td>&quot;Digital Video Test Pattern Generators&quot;</td>
<td>v1.0</td>
<td>01/02</td>
</tr>
<tr>
<td>XAPP288</td>
<td>&quot;Serial Digital Interface (SDI) Video Decoder&quot;</td>
<td>v1.0</td>
<td>10/01</td>
</tr>
<tr>
<td>XAPP298</td>
<td>&quot;Serial Digital Interface (SDI) Video Encoder&quot;</td>
<td>v1.0</td>
<td>10/01</td>
</tr>
<tr>
<td>XAPP172</td>
<td>&quot;The Design of a Video Capture Board Using the Spartan Series&quot;</td>
<td>v1.0</td>
<td>03/99</td>
</tr>
<tr>
<td>XAPP208</td>
<td>&quot;IDCT Implementation in Virtex Devices for MPEG Applications&quot;</td>
<td>v1.1</td>
<td>12/99</td>
</tr>
<tr>
<td>XAPP241</td>
<td>&quot;Virtex-EM FIR Filter for Video Applications&quot;</td>
<td>v1.1</td>
<td>10/00</td>
</tr>
<tr>
<td>XAPP284</td>
<td>&quot;3 x 3 Matrix Multiplier for 3D Graphics and Video&quot;</td>
<td>v1.1</td>
<td>10/01</td>
</tr>
<tr>
<td>XAPP283</td>
<td>&quot;Color Space Conversion&quot;</td>
<td>v1.0</td>
<td>07/01</td>
</tr>
<tr>
<td>XAPP219</td>
<td>&quot;Transposed Form FIR Filters&quot;</td>
<td>v1.2</td>
<td>10/01</td>
</tr>
<tr>
<td>XAPP270</td>
<td>&quot;High-Speed DES and Triple DES Encryptor/Decryptor&quot;</td>
<td>v1.0</td>
<td>08/01</td>
</tr>
</tbody>
</table>

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

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

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

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

Flexibility of DSP Processors

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
Questions?
espsteam@xilinx.com