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

<table>
<thead>
<tr>
<th>Time</th>
<th>Pixel Rate</th>
<th>Video Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>1995</td>
<td>20</td>
<td>DVD</td>
</tr>
<tr>
<td>2000</td>
<td>720p</td>
<td>720p</td>
</tr>
<tr>
<td>2005</td>
<td>1080p</td>
<td>1080p</td>
</tr>
<tr>
<td>2013</td>
<td>480</td>
<td>4K</td>
</tr>
</tbody>
</table>
Data Rates – 8 Mpix sensor

<table>
<thead>
<tr>
<th>Hsize</th>
<th>Vsize</th>
<th>Bits</th>
<th>Pic/sec</th>
<th>BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>3840</td>
<td>2160</td>
<td>8</td>
<td>30</td>
<td>1.99 Gbps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>60</td>
<td>3.98 Gbps</td>
</tr>
<tr>
<td>3840</td>
<td>2160</td>
<td>12</td>
<td>30</td>
<td>2.99 Gbps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>60</td>
<td>5.97 Gbps</td>
</tr>
<tr>
<td>3840</td>
<td>2160</td>
<td>16</td>
<td>30</td>
<td>3.98 Gbps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>60</td>
<td>7.96 Gbps</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hsize</th>
<th>Vsize</th>
<th>Bit/pel</th>
<th>Pic/sec</th>
<th>BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>3840</td>
<td>2160</td>
<td>24</td>
<td>30</td>
<td>5.97 Gbps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>60</td>
<td>11.9 Gbps</td>
</tr>
<tr>
<td>3840</td>
<td>2160</td>
<td>32</td>
<td>30</td>
<td>7.96 Gbps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>60</td>
<td>15.9 Gbps</td>
</tr>
<tr>
<td>3840</td>
<td>2160</td>
<td>36</td>
<td>30</td>
<td>8.95 Gbps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>60</td>
<td>17.9 Gbps</td>
</tr>
</tbody>
</table>
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

- **Video In**
  - SDI (HD/SDI)
  - DVI

- **Video Out**
  - DVI
  - HD-SDI/3G-SDI

- **Component Interfaces**
  - PCIe
  - DDR2/3

**Components**
- Processor
- Memory

**Peripheral Interfaces**
- LVDS
- PCIe
- USB 2.0
- SATA
- DVI
- SDI
- HDMI
- Ethernet

**Peripherals**
- USB Ports
- Network
- Display
- Hard Disk Drive
- User Interface
- 2nd Video Source
Select I/O Resources

12bit Serializer

Channel #0
DATA_0_N
DATA_0_P
DATA_2_N
DATA_2_P
D_CLK_0_N
D_CLK_0_P
DATA_4_N
DATA_4_P
DATA_6_N
DATA_6_P

Channel #2
SLVS CLOCK #0
Channel #4
Channel #6
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

HSLVDCl Controlled Impedance Driver with Bidirectional Termination
Image Data Processing

Typical DSP processor – Sequential

Data In

Coefficients

Clock cycles equals number of coeffs

MAC Unit

Data Out

FPGA - Parallel Implementation

Data In

C0

C1

C2

C3

Cn

Data Out

Clock cycles programmable based on data throughput

X

X

X

X

X
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

**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

**Video Processing**
- Video Scaler
- On-Screen-Display
- Image Characterization

**Ethernet** (Video/Control)
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

\[ B_7 = \frac{(B_6 + B_8)}{2} \]
Linear Interpolation

$$R_7 = \frac{(R_2 + R_{12})}{2}$$
Linear Interpolation

\[ G_8 = \frac{(G_3 + G_7 + G_9 + G_{13})}{4} \]
Linear Interpolation

\[ R_8 = \frac{(R_2 + R_4 + R_{12} + R_{14})}{4} \]
Linear Interpolation

\[ B_{12} = \frac{(B_6 + B_8 + B_{16} + B_{18})}{4} \]
Linear Interpolation

\[
G8 = \frac{(G3+G7+G9+G13)}{4}
\]

\[
B7 = \frac{(B6+B8)}{2}
\]

\[
R7 = \frac{(R2+R12)}{2}
\]

\[
R8 = \frac{(R2+R4+R12+R14)}{4}
\]

\[
B12 = \frac{(B6+B8+B16+B18)}{4}
\]
### Color Correction Matrix

- 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

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

**Uncorrected**

**Corrected**
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 $\gamma<1$ is gamma compression, $\gamma>1$ 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

\[
\begin{bmatrix}
Y \\
C_R \\
C_B
\end{bmatrix}
= 
\begin{bmatrix}
c_A & 1-c_A-c_B & c_B \\
c_C(1-c_A) & c_C(c_A+c_B-1) & c_C(-c_B) \\
c_D(-c_A) & c_D(c_A+c_B-1) & c_D(1-c_B)
\end{bmatrix}
\begin{bmatrix}
R \\
G \\
B
\end{bmatrix}
+ 
\begin{bmatrix}
O_Y \\
O_G \\
O_B
\end{bmatrix}
\]
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.

<table>
<thead>
<tr>
<th>Line ( n )</th>
<th>( \bigcirc )</th>
<th>( \bigcirc )</th>
<th>( \bigcirc )</th>
<th>( \bigcirc )</th>
<th>( \bigcirc )</th>
<th>( \bigcirc )</th>
<th>( \bigcirc )</th>
<th>( \bigcirc )</th>
<th>( \bigcirc )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line ( n+1 )</td>
<td>( \bigcirc )</td>
<td>( \bigcirc )</td>
<td>( \bigcirc )</td>
<td>( \bigcirc )</td>
<td>( \bigcirc )</td>
<td>( \bigcirc )</td>
<td>( \bigcirc )</td>
<td>( \bigcirc )</td>
<td>( \bigcirc )</td>
</tr>
<tr>
<td>Line ( n+2 )</td>
<td>( \bigcirc )</td>
<td>( \bigcirc )</td>
<td>( \bigcirc )</td>
<td>( \bigcirc )</td>
<td>( \bigcirc )</td>
<td>( \bigcirc )</td>
<td>( \bigcirc )</td>
<td>( \bigcirc )</td>
<td>( \bigcirc )</td>
</tr>
</tbody>
</table>

444 format | 422 format | 420 format
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 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

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