Display & Drive Control using FPGAs

Christoph Glattfelder / Oliver Bründler
Enclustra GmbH
<table>
<thead>
<tr>
<th><strong>Table of Content</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>▪ Company Profile</td>
</tr>
<tr>
<td>▪ Display Controller in a FPGA</td>
</tr>
<tr>
<td>▪ Interfaces</td>
</tr>
<tr>
<td>▪ Controller Architecture</td>
</tr>
<tr>
<td>▪ Conclusion</td>
</tr>
<tr>
<td>▪ Drive Controller in a FPGA</td>
</tr>
<tr>
<td>▪ Velocity Estimation</td>
</tr>
<tr>
<td>▪ Drive Control Loops</td>
</tr>
<tr>
<td>▪ Conclusion</td>
</tr>
<tr>
<td>▪ Questions</td>
</tr>
</tbody>
</table>
Enclustra GmbH – Company Profile

- **Quick Facts**
  - Founded in 2004
  - Located at Technopark Zurich
  - Currently 6 FPGA Engineers
  - Vendor-Independent

- **FPGA Design Center**
  - FPGA-Related Design Services
  - Firmware (VHDL/Verilog)
  - Hardware (incl. analog and digital interfaces)
  - Embedded Software (for FPGA soft processors)

- **FPGA Solution Center**
  - FPGA Modules
    - Mars, Mercury and Saturn
  - IP Cores
    - TFT Display Controller
    - Universal Drive Controller
    - Etc.
Mars MX1/MX2 FPGA Module

- Low-cost, low-power Spartan-6 FPGA
- SO-DIMM form factor
- Single supply voltage
- 16 MB quad SPI Flash
- 256 MB DDR2 SDRAM
- Dual Fast Ethernet (MX1 only)
- Real time clock (MX1 only)
- Gigabit Ethernet (MX2 only)
- Dual multi-gigabit transceivers (MX2 only)
- PCIe endpoint (MX2 only)
Mercury CA1 FPGA Module

- Low-cost Cyclone IV FPGA
- Single supply voltage
- 16 MB SPI Flash
- 256 MB DDR2 SDRAM
- Gigabit Ethernet PHY
- USB 2.0 High-Speed interface
- High-power 8A core power supply
Table of Content

- Company Profile
- Display Controller in a FPGA
  - Interfaces
  - Controller Architecture
  - Conclusion
- Drive Controller in a FPGA
  - Velocity Estimation
  - Drive Control Loops
  - Conclusion
- Questions
Display Basics

<table>
<thead>
<tr>
<th>HSync</th>
<th>DE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HSync</th>
<th>HBP</th>
<th>Pixel Data</th>
<th>HFP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Display Interfaces

Display Basics

[Diagram showing display interface signals such as HSync, VSync, DE, VBP, HBP, Pixel Data, and HFP.]
## Display Interfaces

### Overview

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Interface</th>
<th>Color Depth (per color)</th>
<th>Pixel Clk (MHz)</th>
<th>Data rate (MB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>320x240</td>
<td>Parallel</td>
<td>6</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>640x480</td>
<td>Parallel/LVDS</td>
<td>6/8</td>
<td>20</td>
<td>37</td>
</tr>
<tr>
<td>800x600</td>
<td>Parallel/LVDS</td>
<td>6/8</td>
<td>32</td>
<td>86</td>
</tr>
<tr>
<td>1024x768</td>
<td>LVDS*</td>
<td>6/8</td>
<td>52</td>
<td>142</td>
</tr>
<tr>
<td>1280x1024</td>
<td>LVDS*</td>
<td>6/8</td>
<td>87</td>
<td>236</td>
</tr>
<tr>
<td>1600x1200</td>
<td>LVDS*</td>
<td>8</td>
<td>127</td>
<td>346</td>
</tr>
<tr>
<td>1920x1200</td>
<td>LVDS*</td>
<td>8/10</td>
<td>152</td>
<td>553</td>
</tr>
</tbody>
</table>

* and DVI/HDMI/DisplayPort for external connection
Parallel

- 18/24 bit data, HSync + VSync oder DE, PxlClk
  -> 20..30 signal wires
- Up to 800x600 (33MHz)
**LVDS for Displays**

- Serial high speed interface
- 1 Pixel is transferred over 3..4 LVDS lines (+ clock) with a 7 times higher frequency
- From 640x480 up to 1280x1024 single link (85 MHz PxlClk, 600 Mbps)
- Up to 1920x1200 as dual-link (even/odd pixels)
- higher resolutions as quad-link
### DVI / HDMI

- **TMDS (Transition Minimized Differential Signaling)** with 8B/10B encoding
- Data rate 10x higher than pixel clock
- Sync and control signals are transferred during blanking period
- 24 bit color depth
- Up to 165 MHz pixel clock (1.65 Gbps data rate)
- Up to 495 MBytes/s (per Link)
Display Interfaces

DisplayPort

- Similar to PCIe
- 1, 2 or 4 lanes with differential signaling
- 2.7 Gbps or 1.62 Gbps per lane (fixed, independent of the Pixel Clock)
- no dedicated clock channel
- 18..48 bit color depth
- Up to 1080 MBytes/s (netto, 4 Lanes)
- Transceiver required
Display Interfaces

**Low Cost FPGAs (Spartan, Cyclone)**
- Up to 300 Mbps single-ended IO
- Up to 600 Mbps differential IO
- With high speed serial transceiver: up to 3.125 Gbps
- *Alternative for high resolutions*: external serializer or DVI driver

**High End FPGAs (Virtex, Stratix)**
- Up to 800 Mbps single-ended IO
- Up to 1.6 Gbps differential IO, 10:1 serializer
- Up to 11 Gbps with high speed transceiver

<table>
<thead>
<tr>
<th></th>
<th>Parallel (5..30 Mbps)</th>
<th>LVDS (140..600 Mbps)</th>
<th>DVI/HDMI (0.5..1.65 Gbps)</th>
<th>DisplayPort (1.62..2.7 Gbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spartan, Cyclone</td>
<td>IO</td>
<td>Diff IO</td>
<td>Diff IO</td>
<td>-</td>
</tr>
<tr>
<td>Spartan, Cyclone with Transceiver</td>
<td>IO</td>
<td>Diff IO</td>
<td>Transceiver</td>
<td>Transceiver</td>
</tr>
<tr>
<td>Virtex, Stratix</td>
<td>IO</td>
<td>Diff IO</td>
<td>Diff IO / Transceiver</td>
<td>Transceiver</td>
</tr>
</tbody>
</table>
Table of Content

- Company Profile
- **Display Controller in a FPGA**
  - Interfaces
  - **Controller Architecture**
  - Conclusion
- **Drive Controller in a FPGA**
  - Velocity Estimation
  - Drive Control Loops
  - Conclusion
- Questions
Overview

- Source (uC, Camera, GigE, PCIe)
- Memory
- Display Controller
- Interfaces (Touch, Buttons, Rotary-Encoder)
A simple Display Controller

- Register Bank for configuration
- Timing Generator: 2 Counter
- Constant data rate needed: FIFO
- Different Bus and Pixel Clock: Async FIFO
- High Memory Bandwith: Read in bursts
- Banking periods: fill FIFO
2D Acceleration

- DMA Unit
- Draw lines and rectangles
- Clear Video pages
**2D Acceleration**

- DMA Unit
- Draw lines and rectangles
- Clear Video pages
- Copy rectangles (Font)
- Copy with transparency
Overlay and Transparency

- 2 or more Image Units
- Video Overlay
- On Screen Display
Overlay and Transparency

- 2 or more Image Units
- Video Overlay
- On Screen Display
- $\alpha$-Blending
  \[ y = x_1 \cdot \alpha + x_2 \cdot (1 - \alpha) \]
- Color Correction
Table of Content

- Company Profile
- **Display Controller in a FPGA**
  - Interfaces
  - Controller Architecture
  - Conclusion
- **Drive Controller in a FPGA**
  - Velocity Estimation
  - Drive Control Loops
  - Conclusion
- Questions
Conclusion

FPGAs are ideal for flexible or customized display solutions:

- Support for Parallel, LVDS, DVI/HDMI and DisplayPort
- Optional units (2D Accelerator, Overlay, α-Blendig) can be added if needed
- Parallel processing and multiplier are useful for α-Blendig, Color Correction and other signal processing
- Internal Block RAM can be used for pixel FIFO
- DDR2/3 interface for video memory
- System on a programmable chip can be built using softcore processors and peripherals -> long term availability!
Table of Content

- Company Profile
- Display Controller in a FPGA
  - Interfaces
  - Controller Architecture
  - Conclusion
- Drive Controller in a FPGA
  - Velocity Estimation
  - Drive Control Loops
  - Conclusion
- Questions
System Overview

- Position can be measured
- Velocity has to be estimated
Conventional velocity estimation
- Measure distance over a fixed time
- Resolution +/- 1 increment
- Resolution is proportional to speed
- Scaling can not improve results at low speeds (excessive measurement time)
Velocity estimation using FPGAs

- Measure time over a known distance
- Resolution +/- 1 clock cycle
- Resolution is reciprocally proportional to speed
- Scaling at high speeds possible
  - Vary measurement distance

\[ v = \frac{s}{t} \]
Comparison

- **1 ms measurement time**
  - 50MHz clock
- **Low speed (1k Inc/s encoder)**
  - Conventional: 1 bit
  - Proposed: 16 bit
- **High speed (5M Inc/s encoder)**
  - Conventional: 12-13 bit
    - Independent values every 1ms
  - Proposed: 12-13 bit
    - Scaling 512x
    - Independet values every ~100us
Velocity estimation using FPGAs

Implementation

- Perfectly suited for FPGAs
  - Ringbuffer in block-RAMs
  - Too performance hungry for processors
    - Binary divider
    - Every increment has to be stored (up to 5M Inc/s)
Conclusion

- Improved speed resolution
  - Especially for low speed
- Improved time resolution
  - New value every increment
- Scalability
  - Constant resolution over a wide speed range
- Need high processing power
  - Reasonable realization only on FPGAs (and ASICS)
Table of Content

- Company Profile
- Display Controller in a FPGA
  - Interfaces
  - Controller Architecture
  - Conclusion
- Drive Controller in a FPGA
  - Velocity Estimation
  - Drive Control Loops
  - Conclusion
- Questions
Introduction

- High performance drive controls use 2 to 3 control loops
- Example controller periods
  - Current control 100kHz
  - Velocity control 25kHz
  - Position control 5kHz
Drawbacks using CPUs

- Extremely high realtime requirements
- Only little processing power left for other, more typical CPU tasks
  - Trajectory planning
  - Communication, often over a fieldbus
  - High level control
- Poor scalability to additional axes
Fast drive control loops on FPGAs

Concept using FPGAs

- All control loops are calculated in HW
- Control Task in soft or external CPU
  - Interface optimized for low realtime requirements
  - Only responsible for typical CPU tasks
Example architectures

- Still much processing power needed in control CPU
- But much lower realtime requirements
- Very little processing power needed in control CPU
- No realtime requirements after the move is calculated
Fast drive control loops on FPGAs

Concept of TDM

- FPGAs are fast... Often faster than required
- Use available processing power to minimize resource usage

![Diagram of TDM concept with resources and time (t) axes, and function F(x)]
Fast drive control loops on FPGAs

Efficient implementation

- Requirements
  - 3 controllers per axis
  - 3/4 multiplications per controller
  - Realworld PIDs need also limiting, barrel shifter etc.
  - Price (chip size)

- Implementation
  - TDM on three levels
  - Only one multiplier and one adder
  - Control logic is reused on different levels
  - Well scalable to the number of axis
  - Example: 8 axes, 200kHz current control, 80MHz clock
    - 8 axis x 200kSps x 3 PIDs = 4.8M PIDs/s
    - 80MHz / 4.8M PIDs = 16 cycles per PID
Table of Content

- Company Profile
- Display Controller in a FPGA
  - Interfaces
  - Controller Architecture
  - Conclusion
- Drive Controller in a FPGA
  - Velocity Estimation
  - Drive Control Loops
  - Conclusion
- Questions
Conclusion

- FPGAs are used in many drive control systems for fieldbus access
- Long term availability is important for industrial applications
- FPGAs are available on the newest process nodes
  - Prices are falling
  - Devices are growing
  - New features as hard CPUs further reduce the BOM
Table of Content

- Company Profile
- Display Controller in a FPGA
  - Interfaces
  - Controller Architecture
  - Conclusion
- Drive Controller in a FPGA
  - Velocity Estimation
  - Drive Control Loops
  - Conclusion
- Questions
Christoph Glattfelder  
Enclustra GmbH  
glattfelder@enclustra.com  
Fon +41 43 343 39 49

Oliver Bründler  
Enclustra GmbH  
bruendler@enclustra.com  
Fon +41 43 343 39 48

Slides in PDF format:  
http://www.enclustra.com/de/company/publications/