FPGA Prototyping using Behavioral Synthesis for Improving Video Processing Algorithm and FHD TV SoC Design

Masaru Takahashi
SoC Software Platform Division,
Renesas Electronics Corporation
January 28, 2011
Outline

- Background and Issues
  - Video Processing for FHD TV SoC
  - Estimation System
  - Conventional Design Flow
- Proposal Design Flow
- SystemC Description
- Results of Development
- Conclusion
Background and Issues:
Video Processing for FHD TV SoC

- Picture Quality of SoC is affected by **Video Processing Algorithm**
- **Estimation and Improvement** are necessary before Implementation

**e.g.** I-P Conversion, Noise Reduction
Background and Issues: Estimation System

- Example of Estimation System
- Real Time Estimation by FPGA Prototyping
Background and Issues: Estimation System

- Improved Algorithm is implemented on FPGA

Algorithm

- e.g. I-P Conversion, Noise Reduction

**Diagram**

- Signal Generator
- TV Monitor
- Video Processing
- FPGA
- DRAM
- Evaluation Board
Background and Issues: Conventional Design Flow

Algorithm Improvement

Estimation

RTL Modification/Verification for FPGA

FPGA RTL

Video Processing

FPGA Synthesis

EVA Board

Real Time Estimation

Implementation

RTL Modification/Verification for SoC

SoC RTL

ISSUE: Turn Around Time

e.g.

one month * 5 times
= 5 months
Proposal Design Flow

Algorithm Improvement

SystemC Modification/Verification

Estimation

Behavioral Synthesis

FPGA RTL

Video Processing

Behavioral Synthesis

SoC RTL

FPGA Synthesis

EVA Board

FPGA

Real Time Estimation

Turn Around Time

e.g.

0.5 month * 5 times

= 2.5 months
Proposal Design Flow: Summary

- **Short Turn Around Time** for Estimation
  - SystemC and Behavioral Synthesis
  - Updating FPGA is easy
  - About half as long as without Behavioral Synthesis

- **Easy to Implement on SoC**
  - Same SystemC, different Behavioral Synthesis

Next: How to Describe **SystemC Description**
SystemC Description: What kind of Processing?

- For a pixel of Output Video, **Spatiotemporally neighbor** pixels of Input Video are referred.

![Diagram showing pixel creation and video frames](image-url)
SystemC Description: What kind of Processing?

- Examples: Different Algorithm, Similar Interface

<table>
<thead>
<tr>
<th>I-P Conversion</th>
<th>Noise Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>Noisy frames</td>
</tr>
<tr>
<td>Top field</td>
<td>Creating pixel in missing line</td>
</tr>
<tr>
<td>Bottom field</td>
<td>Creating noise-reduced pixel</td>
</tr>
<tr>
<td>Top field</td>
<td>Noise-Reduced frame</td>
</tr>
<tr>
<td>Output</td>
<td></td>
</tr>
</tbody>
</table>
SystemC Description: Data Input

- Raster scan order

Frame buffers

Line buffers

Shift registers

SystemC

Pixel Creation

Vertical Pixels

Vertical and Horizontal Pixels
SystemC Description: Example

- **Pipeline Synthesis** is available

```c
while(1){
    for( f=0; f<3; f++ ){
        for( y=0; y<H; y++ ){
            for( x=W-1; x>0; x-- ){
                shiftreg[f][y][x] = shiftreg[f][y][x-1];
            }
        }
    }
    shiftreg[f][y][0] = indata[f][y].read();
}
outsig = output_pixel( shiftreg ); //function
outdata.write( outsig );
wait();
```
Results of Development: Example of SoC

- Interlace-Progressive Converter on FHD TV SoC
# Results of Development: Developed IP lists

- Three designs have been developed

<table>
<thead>
<tr>
<th>Function</th>
<th>SystemC Code</th>
<th>Development Period</th>
<th>Gate Count</th>
<th>SoC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interlace-Progressive Converter</td>
<td>13.5 klines</td>
<td>3 Months</td>
<td>6 Months</td>
<td>785 kGates</td>
</tr>
<tr>
<td>Cinema Detection</td>
<td>3.6 klines</td>
<td>3 Months</td>
<td>4 Months</td>
<td>Under development</td>
</tr>
<tr>
<td>MPEG Block Boundary Detection</td>
<td>4.1 klines</td>
<td>3 Months</td>
<td>1 Month</td>
<td>Under development</td>
</tr>
</tbody>
</table>
Results of Development: Development Period

- Detail schedule of MPEG block boundary detection

- **Algorithm Improvement**
  - 6 Months (estimated)
  - 3 Months

- **IP development**
  - 2 Months (estimated) ➔ 1 Month

- **Details**
  - Conventional (estimated) ➔ 1 Month
  - Picture quality estimation ➔ 1 Month
  - First version ➔ Debugging
  - Expanding specification ➔ Debugging
  - Improving accuracy #1 ➔ Improving accuracy #2
  - Verification ➔ RTL coding
  - Debugging ➔ 1 Month
Conclusion

- We provide New Design Flow of Video Processing Algorithm using Behavioral Synthesis.

- Turn Around Time is about half as long as that without Behavioral Synthesis

- Three designs have been developed using New Design Flow