Solving the Challenges of Increasingly Complex FPGA-Designs Using a High-Level Synthesis Approach with Vivado HLS

Sergei Storojev
Tools and Design Methodology Applications
Xilinx

© Copyright 2015 Xilinx
Agenda

- Vivado HLS - key technology in Xilinx solution
- Vivado HLS Fundamental Concepts
- Case Study: Multichannel FIR Filter Architectural Exploration
- Demo
Vivado HLS - Key Technology in Xilinx Solution
Xilinx Technology Evolution

Programmable Logic Devices
- Enables Programmable Logic

All Programmable Devices
- Enables All Programmable & Smarter Systems
Xilinx Multi-Node Product Portfolio Offering

- **Family**
  - **SPARTAN**
    - 45nm
      - SPARTAN
      - Lowest Cost, Low Power
  - **ARTIX**
    - 28nm
      - ARTIX
      - Lowest Power, Smallest Form Factor, Best-in-Class Performance
  - **ZYNQ**
    - 20nm
      - ZYNQ
      - All Programmable SoC
  - **KINTEX**
    - 28nm
      - KINTEX
      - Best Price/Performance/Watt
  - **VIRTEX**
    - 20nm
      - VIRTEX
      - High Capacity & Bandwidth
  - **KINTEX UltraSCALE**
    - 28nm
      - KINTEX UltraSCALE
      - Optimal Performance/Watt
  - **VIRTEX UltraSCALE**
    - 20nm
      - VIRTEX UltraSCALE
      - Highest Performance & Integration

**Increasing Performance & Integration**
Vivado Design Suite Technology Advantages

- **Integrated Design Environment**
  - IP and System-centric Integration with Fast Verification
  - Fast, Hierarchical and Deterministic Closure Automation with ECO
  - Scalable to 100M Gates

- **Debug and Analysis**

- **Shared Scalable Data Model**

- **Accelerating System Integration**

- **Accelerating Implementation**

- **Vivado™ HLS**

- **UltraFAST™ Design Methodology**

© Copyright 2015 Xilinx
Accelerates Algorithmic C to RTL IP integration

- C, C++ or SystemC
- Vivado™ HLS
- VHDL or Verilog
- System IP Integration

Algorithmic Specification

Micro Architecture Exploration

RTL Implementation

Comprehensive Integration with the Xilinx Design Environment

<table>
<thead>
<tr>
<th>Case Study</th>
<th>Traditional</th>
<th>C-based</th>
<th>Acceleration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design Time, Radar Design</td>
<td>60 days</td>
<td>5 days</td>
<td>12x</td>
</tr>
<tr>
<td>Verification Time, Video</td>
<td>~2 days</td>
<td>10 sec</td>
<td>12,000x</td>
</tr>
</tbody>
</table>

- Rapid RTL architecture exploration via Directives
- Co-optimization with RTL synthesis for optimal QoR
- Generates AXI4-based IP for Vivado IP Integrator

Libraries:
- Arbitrary Precision
- Video, OpenCV
- Math
- Linear algebra
- DSP: FFT and FIR
Design Integration: IP Centric Design Flow

C-based IP Creation

- C/C++, SystemC, OpenCL
- Libraries
  - Arbitrary Precision
  - Video, OpenCV
  - Math
  - Linear algebra
  - DSP: FFT and FIR

Vivado™ HLS

VHDL or Verilog

System Integration

- Vivado IP Integrator
- Vivado RTL
- System Generator for DSP
All Programmable SoCs & Vivado HLS

Accelerates Algorithmic C to Co-Processing Accelerator Integration
Vivado High-Level Synthesis Serves a Wide Range of Applications across Markets

Aerospace and Defense
- Radar, Sonar
- Signals Intelligence

Communications
- LTE MIMO receiver
- Advanced wireless antenna positioning

Industrial, Scientific, Medical
- Ultrasound systems
- Motor controllers

Audio, Video, Broadcast
- 3D cameras
- Video transport

Automotive
- Infotainment
- Driver assistance

Consumer
- 3D television
- eReaders

Test & Measurement
- Communications instruments
- Semiconductor ATE

Computing & Storage
- High performance computing
- Database acceleration
Vivado High-Level Synthesis
Serves a Wide Range of Applications across Markets

Aerospace and Defense
- Radar, Sonar
- Signals Intelligence

Communications
- LTE MIMO receiver
- Advanced wireless antenna

Industries
- Ultra-HD
- Medical

Audio, Video, Broadcast
- 3D cameras
- Video transport

Consumer
- 3D television
- eReaders

Aerospace and Defense
- Radar, Sonar
- Signals Intelligence

Industrial, Scientific, Medical
- Ultrasound systems
- Motor controllers

Automotive
- Infotainment
- Driver assistance

Test & Measurement
- Communications instruments
- Semiconductor ATE

Computing & Storage
- High performance computing
- Database acceleration
"The combination of Vivado IPI and HLS has been invaluable to our development. The combination of these abstractions allowed us to develop our algorithms in C++ and rapidly integrate the resulting IP, saving greater than 15X in development costs versus an RTL approach."

~Ties Bos, director of Software and FPGA at Gainspeed, Inc.
All Programmable Abstractions
Programming FPGA/SoC at Software Abstraction Level

SDx™
SDAccel™, SDNet™, MathWorks Zynq Design C models / TLM

Vivado Design Suite
HW/SW Interface (drivers…)
Platform & IP Integration
IP Integr.  System Generator  HLS
State of the Art Implementation
Co-optimized with Architecture

Libraries
IP HLS IP RTL IP

System Engineer
SW Engineer
Platform Integrator
System Engineer
HW Designer
DSP SW Engineer
System Engineer

Faster time to differentiation & revenue
Vivado HLS

C based code

Abstract, no clocks!

Vivado™ HLS

Specific, timed

IP

Accelerates Algorithmic C to RTL Creation
Vivado HLS – Synthesis

- Directives / Pragmas
- Constraints
- Libraries
  - Arbitrary Precision
  - Video
  - Math
  - Linear algebra
  - IP: FFT and FIR

C, C++, SystemC
OpenCL C

Vivado™ HLS

FSM
Datapath

MUL
ADD
DIV
FPU

Connectable IP / Verified RTL

Coding style impacts hardware realization

Abstract, untimed

Target optimized, timed, Connectivity ready

Accelerates Algorithmic C to RTL Creation
Loops Optimization: Latency & Throughput

Examples

void foo_top (...) {
...
add: for (i=0;i<=3;i++)
{
    b = a[i] + b;
...
}

Unrolling

void foo_top (...) {
...
add: for (i=1;i<=2;i++)
{
    op_READ;
    op_COMPUTE;
    op_WRITE;
}
...

Pipelining

w/o pipelining
loop latency = 6
throughput = 3

pipelining
loop latency = 4
throughput = 1
Arrays – Partitioning and Reshaping

- Arrays are the fundamental construct to describe memories...
  - Array accesses can often be performance bottlenecks

- **Partitioning** splits an array into independent arrays
  - Arrays can be partitioned on any of their dimensions

Example: ARRAY_PARTITION set to “cyclic”, factor of 2

- **Reshaping** combines array elements into wider containers
  - Different arrays into a single physical memory
  - New RTL memories are automatically generated without changes to C code
Vivado HLS – Interfaces

Accelerates Algorithmic C to RTL Creation

- Directives / Pragmas
- AXI
- FIFO Interface
- BRAM Interface
- Default HLS protocols
- Handshake Interface

C based code

Connectable IP

Abstract, untimed

Ready to use IP block

© Copyright 2015 Xilinx
Vivado HLS – Verification

C, C++ Testbench and C, C++, SystemC

Abstract, untimed testbench

Single testbench for C-Sim and RTL-Sim

Vivado™ HLS

C simulation
- GCC
- G++
- SystemC

C / RTL Co-simulation
- Xsim
- ISim
- Questa SIM
- VCS
- NCSim
- Riviera
- OSCI

Connectable IP / Verified RTL

Creating a self-checking test bench is highly recommended!

Accelerates Algorithmic C to RTL Creation

Automatic generation of RTL testbench
Design Methodology (UG902)

5 steps process to improve your design

<table>
<thead>
<tr>
<th>Table 1-7: Optimization Strategy Step 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Directives and Configurations</strong></td>
</tr>
<tr>
<td>INTERFACE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 1-8: Optimization Strategy Step 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Directives and Configurations</strong></td>
</tr>
<tr>
<td>DATAFLOW LOOP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 1-9: Optimization Strategy Step 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Directives and Configurations</strong></td>
</tr>
<tr>
<td>ARRAY_PARTITION</td>
</tr>
<tr>
<td>DEPENDENCE</td>
</tr>
<tr>
<td>INLINE</td>
</tr>
<tr>
<td>UNROLL</td>
</tr>
<tr>
<td>Config Array Partition</td>
</tr>
</tbody>
</table>
Case Study:
Multichannel FIR Filter Architectural Exploration
Goal: explore different architecture solutions

- Targeting ZYNQ 7045
- At 50 MSPS fixed data rate, 81 coefficient filter
- 32-bit integer or 32-bit Floating-Point (FP) I/O samples and coefficients

13 different architectures considered

- Fixed the 50 MSPS data rate, 3 different clock frequencies:
  - 50, 150, 300 MHz with respectively II=1, II=3, II=6

### Systolic MAC FIR Filter

<table>
<thead>
<tr>
<th>Asymmetric</th>
<th>Odd-symmetric</th>
</tr>
</thead>
<tbody>
<tr>
<td>25x18 bits integer math accuracy (II=1, II=3, II=6)</td>
<td>25x18 bits integer math accuracy (II=3)</td>
</tr>
<tr>
<td>32x32 bits integer math accuracy (II=1, II=3, II=6)</td>
<td>32x32 bits integer math accuracy (II=3, II=6)</td>
</tr>
<tr>
<td>32-bit floating point accuracy (II=1, II=3, II=6)</td>
<td>32-bit floating point accuracy (II=6)</td>
</tr>
</tbody>
</table>
### Overall Performance Summary

<table>
<thead>
<tr>
<th>Architecture</th>
<th>expected</th>
<th>Syntesis Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>II</td>
<td>MHz</td>
</tr>
<tr>
<td>25x18 asym MAC FIR</td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>25x18 asym MAC FIR</td>
<td>3</td>
<td>150</td>
</tr>
<tr>
<td>25x18 asym MAC FIR</td>
<td>6</td>
<td>300</td>
</tr>
<tr>
<td>25x18 odd-sym MAC FIR</td>
<td>3</td>
<td>150</td>
</tr>
<tr>
<td>32 float asym MAC FIR</td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>32 float asym MAC FIR</td>
<td>3</td>
<td>150</td>
</tr>
<tr>
<td>32 float asym MAC FIR</td>
<td>6</td>
<td>300</td>
</tr>
<tr>
<td>32 float odd-sym MAC FIR</td>
<td>6</td>
<td>300</td>
</tr>
<tr>
<td>32x32 asym MAC FIR</td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>32x32 asym MAC FIR</td>
<td>3</td>
<td>150</td>
</tr>
<tr>
<td>32x32 asym MAC FIR</td>
<td>6</td>
<td>300</td>
</tr>
<tr>
<td>32x32 odd-sym MAC FIR</td>
<td>3</td>
<td>150</td>
</tr>
<tr>
<td>32x32 odd-sym MAC FIR</td>
<td>6</td>
<td>300</td>
</tr>
<tr>
<td>ZYNQ 7045 max resources</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To accomplish the work
- RTL Approach will take days
- Vivado HLS: 6 hours only
  - 4 hours: C coding
  - 2 hours: Tcl scripting, HLS Execution
Demo
Conclusion

Vivado HLS

- Key technology in Xilinx solution

- Reduces >10x design time and >100x verification time

- Enables rapid RTL architecture exploration for optimal QoR