Experiences with Multi-Core SoC Designs with FPGA Prototyping

Software Driven Verification with Fast Functional SoC and System prototypes

White Eagle Systems Technology
William Wu, will@whiteeagle.us
Dr. Jim Tobias, jim@whiteeagle.us

Toshiba America Electronic Components, Inc.
Bob Uvacek, bob.uvacek@taec.toshiba.com
Tool Flow and Partition Strategy - Slice the Bus

- SOC Development Environment
- SoC RTL
- WhiteEagle® RTL-Slicer
- Emulation Ready RTL
- Standard ASIC Design Flow
- Slicer Parameters
- FPGA Synthesis & Place n Route
- Partitioned SoC into FPGA Bit Files

SoC Bus

- CPU1
- CPU2
- USB
- DDR
- LCD
- DMA
- CPU3
- ETH
- PCI
- LL Devices
- LL Devices
- DSP
- FPGA #1
- FPGA #2
- FPGA #3

WhiteEagle® RTL-Slicer

Standard ASIC Design Flow

FPGA Synthesis & Place n Route

Partitioned SoC into FPGA Bit Files
Interconnect Reduction (Shared Bus)

- Shared bus interconnect is usually implemented with multiplexers
- Transform the multiplexers to Tri-State buffers
- Functional Equivalent
Interconnect Signal Reduction (Matrix Bus)

- Transform Matrix Bus to Shared Bus with Time Division Multiplexing
- Speeding up the TDM Modules Obtains Timing Equivalence
Results from an SoC Design

- ARM10, ARM9, DDR, LCD, Ethernet, USB, IDE, ADC, UART, SPI, GPIO, etc
- 4 Million ASIC Gates => 95,000 Xilinx Slices + 170 Xilinx Block Rams
- 3 Virtex-2 part number XC2V8000
- 30 MHz clock speed

- Virtex-II 8000 has space for 1,000,000 ASIC Logic Gates + 1.2 Million Memory Gates
System Setup

Software Development workstation

Emulation including FPGA Board with Analog I/O Daughter Card

Customer Test board System Development Board

PC Workstation

Oscilloscope

MICTOR IC E-box

CABLES

Front Plate
Interface Issues

• Analog IO
  – Separate the analog portion
  – Construct the equivalent behavior with analog components on the expansion card
  – Design wrapper circuit as necessary

• Digital IO
  – DDR
    • Use DCM to generate four phases of the system clock, use the appropriate phase as DQS signal
    • Select SSTL2 IO
  – Strict timing interfaces
    • Divide the design into real world speed and prototyping speed portions
    • Use buffer for communication
    • Design special interface circuit
Software Driven Multi-Core SoC Verification

- Fast Functional Prototypes enable Software Driven Verification
- Software Driven Verification Easily Handles the Huge Task of System Verification
  - Data and Control Bandwidth
  - External Interfaces
  - The Growing Complexity of IP Blocks
- Advantage: Software
  - Software is more flexible and thorough than test vectors
  - Verification software is reused for device bring up and in final product
- FPGA SoC Prototyping Speeds up Software Development