System-on-a-programmable-chip Solution from Altera and Xilinx

Xun Yang
VLSI CAD Lab, Computer Science Department, UCLA
FPGAs with Embedded Microprocessors

- Combination of embedded processors and programmable logic is the next step in the evolution of system integration.

- Advantages:
  - Extreme flexibility of programmable logic and processor
  - Low cost and short time-to-market
  - Hardware/Software partitioning for more efficient implementations
Three families of embedded processor solutions integrated with Altera’s APEX PLD architecture.

- Nios soft-core embedded processor
- ARM-based hard core embedded processor
- MIPS-based hard core embedded processor
The Nios embedded processor is a general purpose RISC CPU implemented as a soft core in Altera APEX 20K devices. The Nios family includes a 32-bit and a 16-bit processor core.

Includes the industry-standard GNUPro software from Cygnus.
<table>
<thead>
<tr>
<th></th>
<th>Nios 32-bit</th>
<th>Nios 16-bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data bus size (bits)</td>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td>ALU width (bits)</td>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td>Internal register width (bits)</td>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td>Address bus size (bits)</td>
<td>33</td>
<td>17</td>
</tr>
<tr>
<td>Instruction size (bits)</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Logic cells (typical)</td>
<td>1,700</td>
<td>1,100</td>
</tr>
<tr>
<td>$f_{\text{MAX}}$ (EP20K200E-1 device)</td>
<td>Up to 50 MHz</td>
<td>Up to 50 MHz</td>
</tr>
</tbody>
</table>
ARM-based embedded processor

- Industry-standard ARM922T 32-bit RISC processor core operating at up to 166 MHz
- Memory management unit (MMU) included for real-time operating system (RTOS) support
- System-on-a-programmable-chip (SOPC) architecture builds upon features of the APEX 20KE family of PLDs, with up to 1,000,000 gates
- Internal single-port SRAM up to 256 Kbytes - Internal dual-port SRAM up to 128 Kbytes
- External SDRAM 133-MHz data rate (PC133) interface up to 512 Mbytes External dual data rate (DDR) 266-MHz data rate (PC266) interface up to 512 Mbytes
- External flash memory up to 32 Mbytes
- PLD configuration/reconfiguration possible via the embedded processor software
- Integrated hardware and software development environment
- C/C++ compiler, source-level debugger, and RTOS support
MIPS-based embedded processor

- Industry-standard MIPS32 4Kc 32-bit RISC processor core operating at up to 166 MHz
- Memory management unit (MMU) included for real-time operating system (RTOS) support
- 32-bit MIPS RISC processor instruction set, user-level compatible with the R3000 and R4000 (32-bit mode)
- Part of the Altera Excalibur embedded processor solutions: system-on-a-programmable-chip (SOPC) architecture builds upon features of the APEX20KE family of PLDs, with up to 1,000,000 gates
- Internal single-port SRAM up to 256 Kbytes
- Internal dual-port SRAM up to 128 Kbytes
- External flash memory up to 32 Mbytes
- PLD configuration/reconfiguration possible via the embedded processor software
- C/C++ compiler, source-level debugger, and RTOS support
Nios Embedded Processor Development Tools

- Quartus-II
- Nios Hardware Development Kit
- Cygnus GNUPro Software Development Kit, Cygwin is required for GNUPro tools to run
- Development Board
Nios Software Development Flow

- Set up communication with the board
  - Select Program>Cygwin>bash
  - Type nios-run –t

- Test the system
  - Compile and download demo
    - nios-build lcd_demo1.c
  - Run the demo
    - nios-run lcd_demo1.srec
Nios Software Development Flow

Define the processor
- Specify CPU data path to be 16 or 32 bit
- If dedicated hardware multipliers are necessary
- How much on-chip ROM or RAM is necessary
- Specify off-chip memory
- Decide the number and type of peripherals
Nios Software Development Flow

- Use Quartus and MegaWizard Plug-In manager to Build the Processor
- Save the Processor Configuration to Flash
- Create the Application Code
- Download the Application Code
- Debug the Code
  - The code can be built with debugging set ON
Nios Software Development Flow

- Transition to Auto-Booting Code
  - Code in on-chip memory, change on-chip RAM to on-chip ROM and rebuild the design using code to initialize ROM
  - Code in off-chip memory, remove the monitor entirely, change the reset address to point to the program in flash memory. Add a routine that copies the executable code from flash to memory at start time.
Xilinx’s Solution

- Virtex-II family Platform FPGAs
- MicroBlaze soft processor core
- AllianceCORE third-party IP provider
- Other processor cores tested in Xilinx devices and open source processor cores: 2900, 6502, 8051, Java, MIPS-like, PIC, SAPRC, Proprietary.
Platform FPGAs

- Offer a single platform that can address nearly any application include processing, DSP, connectivity.
- Embedded PowerPC processors, CoreConnect Buses, MicroBlaze soft processors and related peripherals and interfaces extend FPGA applications into embedded systems
Platform FPGAs

![Diagram showing FPGAs and associated components with labels such as MicroBlaze, UART, PPC 405, PowerPC, and CoreConnect Technology.]
Virtex-II Family

- Virtex-II devices with densities from 40,000 up to 10 million gates, enhanced system memory, flexible system interfaces with signal integrity, complex system clock management, and on-board EMI management, with embedded multipliers and improved arithmetic functions, flexible IP-immersion fabric.
## Virtex-II Family

<table>
<thead>
<tr>
<th>Virtex-II Part Number</th>
<th>XC2V40</th>
<th>XC2V80</th>
<th>XC2V250</th>
<th>XC2V500</th>
<th>XC2V1000</th>
<th>XC2V1500</th>
<th>XC2V2000</th>
<th>XC2V3000</th>
<th>XC2V4000</th>
<th>XC2V6000</th>
<th>XC2V8000</th>
<th>XC2V10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic Cells</td>
<td>57</td>
<td>115</td>
<td>345</td>
<td>691</td>
<td>115</td>
<td>172</td>
<td>241</td>
<td>322</td>
<td>518</td>
<td>760</td>
<td>104</td>
<td>138240</td>
</tr>
<tr>
<td>BRAM(Kb)</td>
<td>72</td>
<td>144</td>
<td>432</td>
<td>576</td>
<td>720</td>
<td>864</td>
<td>100</td>
<td>172</td>
<td>216</td>
<td>259</td>
<td>302</td>
<td>3456</td>
</tr>
<tr>
<td>Multipliers</td>
<td>4</td>
<td>8</td>
<td>24</td>
<td>32</td>
<td>40</td>
<td>48</td>
<td>68</td>
<td>96</td>
<td>120</td>
<td>144</td>
<td>168</td>
<td>192</td>
</tr>
<tr>
<td>DCM Units</td>
<td>4</td>
<td>4</td>
<td>8</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>
Virtex-II Family

<table>
<thead>
<tr>
<th>function</th>
<th>Industry’s fastest DSP core</th>
<th>Xilinx Virtex-E-08</th>
<th>Xilinx Virtex-II</th>
</tr>
</thead>
<tbody>
<tr>
<td>MACs per second</td>
<td>8.8 billion</td>
<td>128 billion</td>
<td>600 billion</td>
</tr>
<tr>
<td>(multiply and accumulate)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIR filter</td>
<td>17 MSPS @1.1GHz</td>
<td>160 MSPS @160MHz</td>
<td>180 MSPS @180MHz</td>
</tr>
<tr>
<td>FFT</td>
<td>7.7 μs @800 MHz</td>
<td>41 μs @100 MHz</td>
<td>&lt;1 μs @140 MHz</td>
</tr>
</tbody>
</table>
Conventional Programmable DSP

- Fixed inflexible structure
  - Typically 1-4 MACs
  - Fixed data width
- Serial processing limits data throughput
  - Time-shared MACs
  - High clock frequency creates difficult system challenge
Virtex-II Family

- Flexible structure
  - Distributed DSP resources (registers, memory, multiplier, LUT)

- Parallel processing maximizes data throughput
  - Support any level parallelism
  - Optimal performance/cost trade-off

- FPGAs also supports serial processing
Virtex-II Family DSP Development Environment

- System level Design & Simulation
  - The MathWorks Partnership
  - Simulink & Matlab
  - System Generator for Simulink

- FPGA Design & Simulation

- FPGA Implementation
The Xilinx SYSTEM Generator for The MathWorks Simulink

System Design Domain:
- Algorithm design
- System verification (floating point)
- Conversion to fixed point
- Optimization and re-verification

MATLAB and Simulink

Hardware Design Domain:
- Automatic generation of HDL
- Hardware simulation
- Synthesis, place and route
- Timing verification

The Xilinx SYSTEM Generator

Foundation/Alliance
MicroBlaze Soft Core Structure

[Diagram of MicroBlaze Soft Core Structure]

- On-Chip Instruction Memory 0-256KB
- Program Counter
- Control Unit
- Instruction Buffer
- Machine Status Register
- Register File 32 x 32 bit
- Shift/Logical, Add/Subtract, Multiply
- CoreConnect OPB/IF
- Off-Chip Memory 0-4GB
- Watchdog Timer
- General Purpose I/O
- Timer/Counters
- On-Chip Data Memory 0-256KB
- Data Bus Controller
- UART
- CoreConnect OPB/IF
- Off-Chip Memory 0-4GB
MicroBlaze’s Performance Advantage

- 32-Bit Harvard ISA Processor: 50 D-MIPS, 800 Luts
- MicroBlaze
  - 50 D-MIPS Each!
- Nearest Competitor: Half the Size
  - of the nearest competitor
  - (Two processors in the same silicon space)

Graph showing Dhrystone MIPS Performance and Clock Rate (MHz).
MicroBlaze Software Flow

- Allow user to select processor parameters and peripheral configurations and generate netlist files as well as the required header and driver files
Xilinx + ARC Cores

- Xilinx has teamed up with ARC Cores to offer the ARC soft 32-bit configurable processor solution for use in Xilinx's Virtex and Spartan series FPGAs.
- The ARC is a user customizable 32-bit soft-core microprocessor architecture suitable for embedded processor based applications.
ARC Structure
## ARC features

<table>
<thead>
<tr>
<th>Example Implementation</th>
<th>Basecase ARC</th>
<th>Basecase ARC</th>
<th>ARC with DSP extensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Tested</td>
<td>2S150-6</td>
<td>V400E-8</td>
<td>V400E-8</td>
</tr>
<tr>
<td>CLB Slices</td>
<td>1538</td>
<td>1517</td>
<td>4800</td>
</tr>
<tr>
<td>Performance (MHz)</td>
<td>37</td>
<td>41</td>
<td>23</td>
</tr>
<tr>
<td>Xilinx Tools</td>
<td>M2.1i SP6</td>
<td>M2.1i SP6</td>
<td>M2.1i SP6</td>
</tr>
<tr>
<td>Special Features</td>
<td>9 Block RAMs</td>
<td>9 Block RAMs</td>
<td>29 Block RAMs</td>
</tr>
</tbody>
</table>
ARC Development Environment

- C/C++ compiler, assembler/linker, SeeCode source-level debugger
- Supports a large variety of commercially distributed RTOS
- ARChitect GUI can be used to safely create custom configured processor
- Adding a few appropriate instructions specifically needed for the application
Equally configurable software tools and debugging tools, compiler must be aware of the instruction set and take advantage of features such as multipliers or barrel shifters
LavaCORE Structure

- Lava CORE Processor
  - ALU
  - Register File
  - Local Memory
  - Interrupt Controller
  - Programmable Timers
  - Encryption Unit

- Floating Point Unit

- External Memory

- Garbage Collector

Optional Modules
LavaCORE Configuration Environment

Figure 2: LavaCORE Configuration Environment
LavaCORE Configuration Environment

- Analyze an application to determine which Java bytecode instructions can be omitted
- FormalSYNTHESIS technology is used to synthesize an implementation leading to application specific optimal solutions
- A gate level implementation is generated together with an HDL testbench and LavaOS
- Hardware interface provides an implementation debugging bridge for the synthesized hardware