FPGA Implementation of Simple Digital Signal Processor

Marko Butorac and Mladen Vucic

University of Zagreb,
Faculty of Electrical Engineering and Computing,
Zagreb, Croatia
Outline

- Introduction

- Processor architecture
  - Requirements and Design Decisions
  - Data Path
  - Instruction Coding
  - Program Flow Control

- Processor Features and Applications

- Conclusion
Introduction I

- Prototyping of digital systems
  - High-level synthesis
  - Moving dedicated hardware to a processor

- Field Programmable Gate Arrays
  - Good prototyping platform
  - Diversity through components
    - memories, multipliers, DSP blocks, configurable logic

- Existing soft core processors
  - Complexity from 8-bit to 32-bit processors
  - Optimized for speed or for minimal resources
  - Various primary usage
    - Encryption, digital signal processing, complex or real arithmetic
Introduction II

- **Objective of this work**
  - design of simple DSP core

- **Key requirements**
  - Main operation → inner product
    - Correlation, convolution, FFT ...
  - Processing → one element per one clock cycle
    - Reading vectors component, MAC operation, incrementing counters, termination condition
  - Simple and small architecture → cascading to create processing chain
Processor Architecture

- Requirements and Design Decisions
- Data Path
- Instruction Coding
- Program Flow Control
Requirements and Design Decisions I

Main requirement

- Instruction $\rightarrow$ inner product of two complex vectors
- Linear and modulo addressing
- Small and complete instruction set
- Simplicity of the design
Requirements and Design Decisions II

- Block diagram

  - Inner product
    - Complex data processing
    - Address generators
    - Counters

  - Data register

  - Host port

  - Program flow control

  - Data & instruction width → 18 bits
    - Xilinx FPGA devices
Data Path I

- Circuit diagram of real parts of complex registers

- 4-1 multiplexer for each register
- ALU → combinatorial function
- Complex register → two 18-bit registers
Data Path II

- Circuit diagram of address generator

- Independent linear and modulo addressing
  - Various addressing with one presetting
Data Path III

- Circuit diagram of loop counter

- Only for loop termination condition

- Described components offer
  - Freedom in forming the instruction set
  - Parallel work
Instruction Coding

- Variable instruction length
  - Constants, branch addresses

- Particular bits → certain part of hardware
  - Decoding is very simple

- Instructions with multiple operations
Program Flow Control I

- “classic” fetch-decode-execute timing \(\rightarrow\) double word load
- pipelined \(\rightarrow\) FILT instruction
- components:
  - ROM
  - Decoder
  - Program counter
  - Return address stack
  - FSM
Program Flow Control II

- Finite state machine for program flow

  - Two possible sequences
    - Non-pipelined instruction
    - Pipelined instruction
Processor Features and Applications I

- **Features**
  - Complex data with 18 bits wide real and imaginary parts
  - Complex arithmetic logic unit
  - Two memories with independent address generators
  - Optimized for inner product
    - $N$ elements vector $\rightarrow N+3$ clock cycles
  - Other instructions execute in 3 clock cycles
    - Simultaneously executes several operations

- **Examples of complex instruction**

  LOADRI CR=#data A=RAMA(AGA_BR) B=RAMB(AGB_BR) Cre=Cim=CR AGA_BR=AGA_BR+1 AGB_BR=AGB_BR+1;

  FLOW BA=label JMP(BA)_IF_LOOP1 CNT1=CNT1-1
Processor Features and Applications II

- Typical applications
  - Filtering
  - Correlation

- Processor cascading
Implementation results

<table>
<thead>
<tr>
<th>Device</th>
<th>XC6SLX150T</th>
<th>XC6VLX240T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed grade</td>
<td>-4</td>
<td>-3</td>
</tr>
<tr>
<td>LUTs</td>
<td>880</td>
<td>887</td>
</tr>
<tr>
<td>Registers</td>
<td>307</td>
<td>307</td>
</tr>
<tr>
<td>Multipliers</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>BRAMs</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>$f_{\text{max}}, \text{MHz}$</td>
<td>93</td>
<td>147</td>
</tr>
</tbody>
</table>

- Reasonable number of cells
- Maximum frequency $\rightarrow$ moderate
  - Platform specific optimizations
Conclusion

- FPGA implementation of digital signal processor
- Optimized for calculation of inner product of complex vectors
- Block RAM is used as program memory
- Dual ported block RAMs are used for data memories
- Simple cascading