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

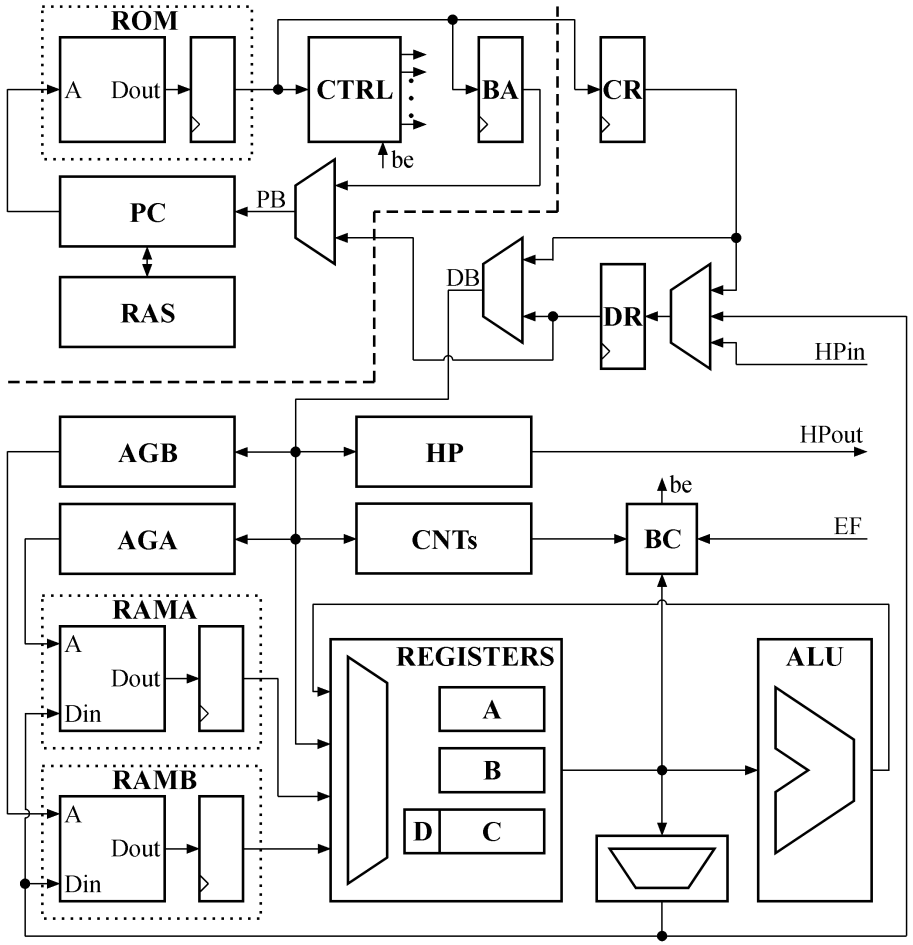
- Requirements and Design Decisions
- Data Path
- Instruction Coding
- Program Flow Control

Requirements and Design Decisions I

- Main requirement
  - Instruction → inner product of two complex vectors
  - Linear and modulo addressing
  - Small and complete instruction set
  - Simplicity of the design

Requirements and Design Decisions II

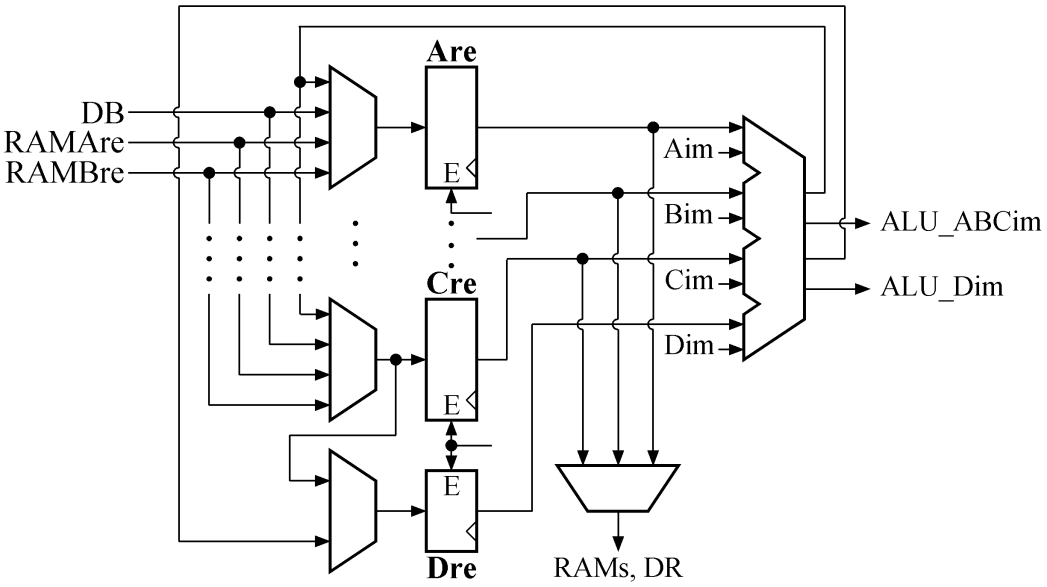
- Block diagram



- 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

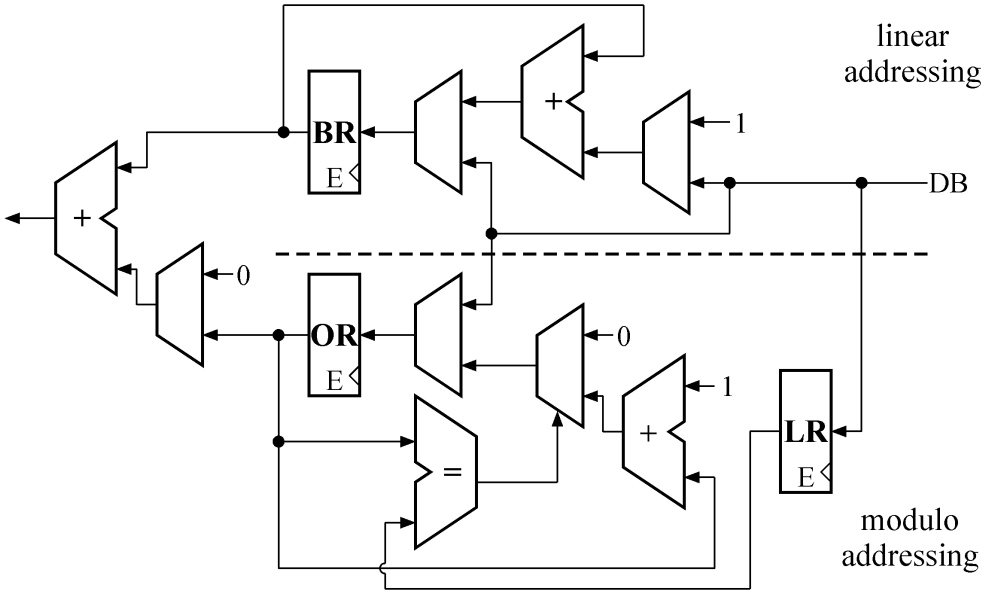
- Circuit diagram of real parts of complex registers



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

Data Path II

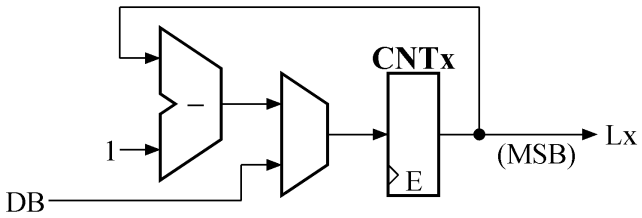
- Circuit diagram of address generator



  - Independent linear and modulo addressing
    - Various addressing with one presetting

Data Path III

- Circuit diagram of loop counter



  - Only for loop termination condition

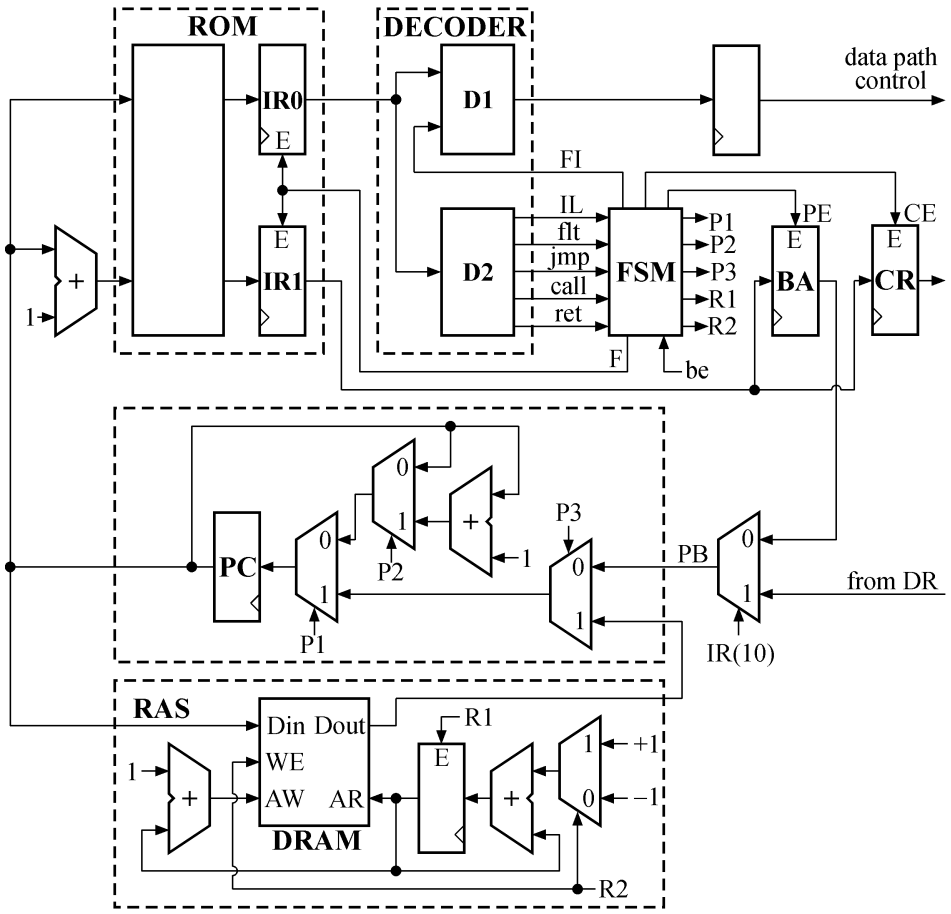
- Described components offer
  - Freedom in forming the instruction set
  - Parallel work

Instruction Coding

Bit	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOP	*IL*	0	0	0	0	0	0	DB	*AGA_BR*		*AGB_BR*		*AGA_OR*		*AGB_OR*		*AGA_LR*	*AGB_LR*
LOAD	*IL*	0	I	R	EA	EB	EC	DB	*AGA_BR*		*AGB_BR*		A_IN		B_IN		C_IN	
STORE	*IL*	1	1	0	RAMA	RAMB	X	DB	*AGA_BR*		*AGB_BR*		X				ABC_OUT	
MOV	*IL*	1	0	0	X	X	X	DB	CNT_OP		REG_SEL				DR_IN		ABC_OUT	
ALU	*IL*	1	1	1	EA	EB	EC	ALU_OPER					A_IN		B_IN		C_IN	
FLOW	*IL*	1	0	1	J	C	R	PB	*CNT_OP*		*CNT_SEL*		O_FLG_MSK		CNT_FLG_MSK			
FILT	0	0	0	0	1	1	1	X	(AGA_BR=00)		(AGB_BR=00)		(A_IN=01)		(B_IN=00)		(C_IN=11)	

- Variable instruction length
  - Constants, branch addresses
- Particular bits → certain part of hardware
  - Decoding is very simple
- Instructions with multiple operations

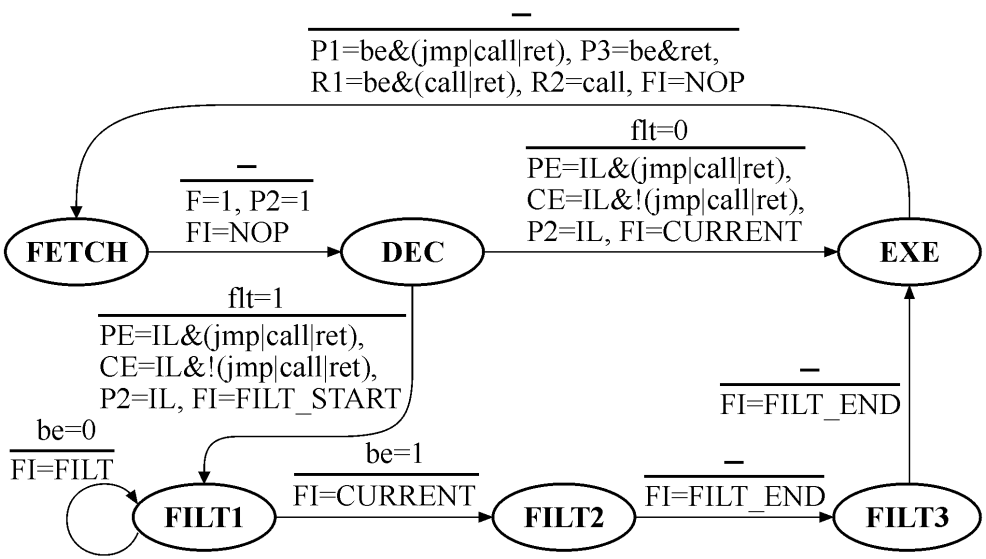
Program Flow Control I



- "classic" fetch-decode-execute timing → double word load
- pipelined → 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 → *N*+3 clock cycles
  - Other instructions execute in 3 clock cycles
    - Simultaneously executes several operations

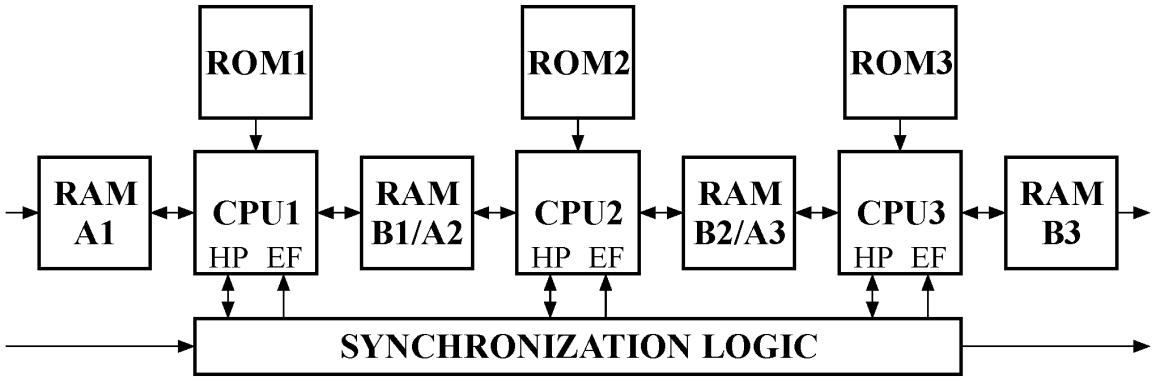
- Examples of complex instruction

```
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

Device	XC6SLX150T	XC6VLX240T
Speed grade	-4	-3
LUTs	880	887
Registers	307	307
Multipliers	4	4
BRAMs	3	3
f_{max}, MHz	93	147

- Reasonable number of cells
- Maximum frequency → 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