Leros: A Tiny Microcontroller for FPGAs

by Schoeberl Martin, 2011

Beatrice Segalini, Delfina Iriarte

Università degli Studi di Padova

October 23rd, 2020

Table of contents

1 Related Work

- Leros vs. PicoBlaze
- Leros vs. SpartanMC & Nios II

2 Leros Design Decisions

- Optimal Logic Size
- Architecture & Instruction set

3 Leros Implementation

- The Pipeline
- I/O Interface, Extensions, Software Tools

4 Evaluation

5 Conclusion

Introduction

Leros is a *minimalistic* 16-bit processor intended for utility functions in an FPGA based *System on Chip* (SoC) design.

Aims:

- Consume less than 500 logic cells
- Use $1 - 2$ on-chip memories

- Reasonable performance
- High maximum clock frequency

Main aims

Good balance!

Difficulties

Bottlenecks!

Pipelined accumulator architecture

- Directly addressable registers in an *on-chip* memory for local variables
- On-chip: **registers** and **general data**
- **Additional** on-chip memory for the instructions

} Only 2 memory blocks!

Related Work Leros vs. PicoBlaze

PicoBlaze is a tiny microcontroller from which **Leros** takes inspiration. They have several similar features:

Leros

- 16—bit microcontroller
- Low resources
- Fewer restrictions on program and datasize
- More logic cells
- 1 — 2 on-chip memories
- Altera and Xilinx devices
- Truly **open source**

PicoBlaze

- 8—bit microcontroller
- Low resources
- More restrictions on program and datasize
- 96 logic cells (Spartan3)
- 1 on-chip memory
- Xilinx devices only
- Not portable

Related work Leros vs. SpartanMC & Nios II

SpartanMC: small, optimized for FPGAs, 18 bits data-instruction width.

Leros

- Fewer resources
- Simplifies the access to registers
- Avoids unusual clocking

SpartanMC

- 2 phase-shifted clocks
- More complex design
- Splits instruction fetch and write-back

Nios II: 32$-$bit, Altera FPGAs, $\approx$ 700 logic cells.

Leros

- Accumulator-based architecture
- 1 clock cycle per instruction

Nios II

- Sequential implementation
- 6 clock cycles per instruction

Leros Design Decisions Optimal Logic Size

Find optimal relation between number of logic cells and on-chip memory consumption

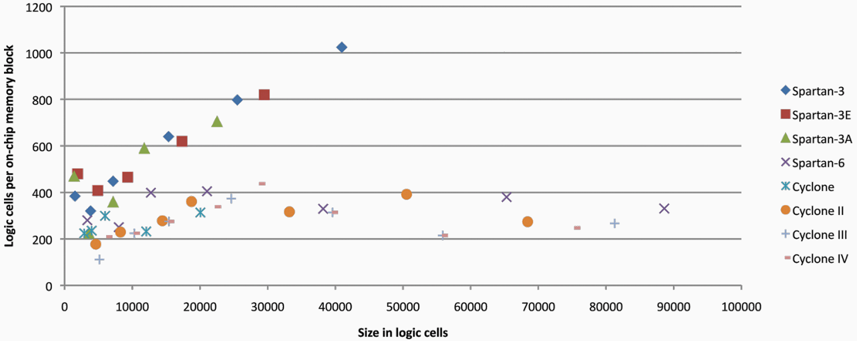


Figure 1 – Number of logic cells per on-chip memory block of low-cost FPGAs from Xilinx and Altera.

Leros design decisions Architecture & Instruction set

Architectural decisions

- The on-chip memories are used only for the instruction memory and the data memory in order to reduce consumption.
- Accumulator design: only a single dedicated register (the accumulator) is connected to the ALU output and provides one input to the ALU.

Instruction Set

- 16 bit data and instructions.
- Accumulator holds one of the operands: called 1 address machines.
- 8 bit immediate values in the instruction.
- I/O ports are accessed with dedicated instructions and an 8 bit address.
- Data memory can be accessed via indirect loads and store.

Leros Implementation

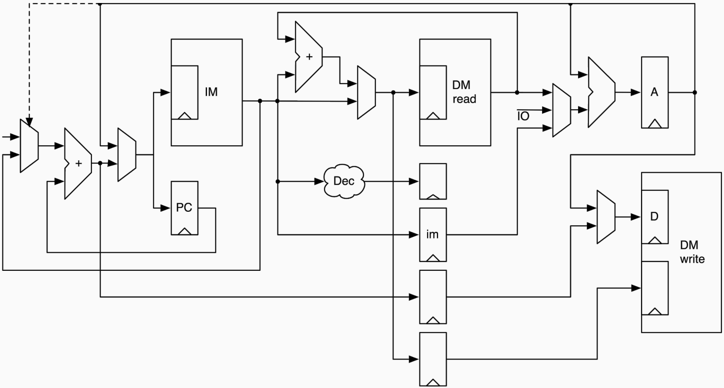
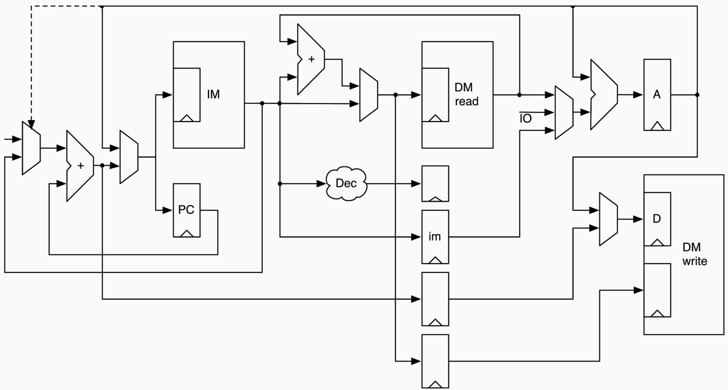


Figure 2 – Pipeline of Leros with the fetch/decode and the read/execute/memory stages.

- two-staged pipeline
- Program Counter (PC)
- Data Memory (DM)
- Instruction Memory (IM)
- Accumulator register (A)

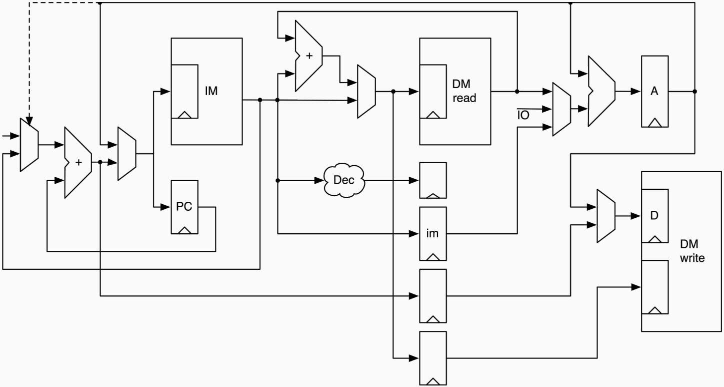
Leros Implementation The Pipeline



2 pipeline stages:

- fetch-decode instruction from IM;
- reads from DM and execute.

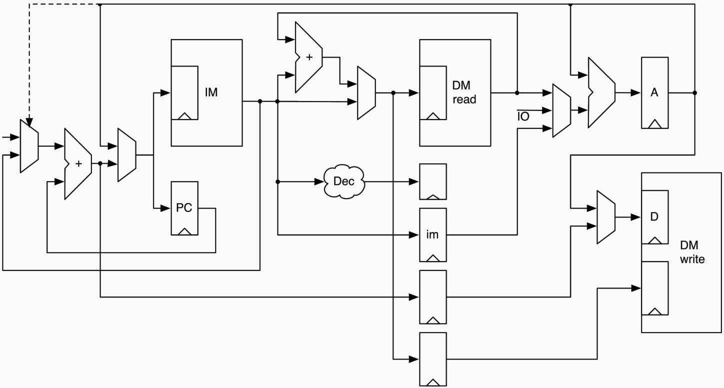
Leros Implementation The Pipeline



Addresses:

- IM-PC → PC+1, instruction+constant, A.
- DM read-write → constant (on-chip registers), DM+offset (loads-stores).
- DM write data → A (store instructions), PC (save PC in register).

Leros Implementation The Pipeline

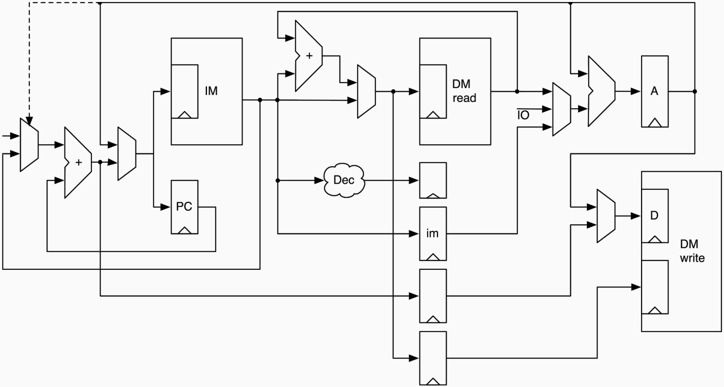


Memory load and store: requires based register and offset. 2 instructions:

1. register address → DM
2. DM → DM+offset

Result is then loaded in A.

Leros Implementation The Pipeline



On chip memories: independent read-write ports → concurrent wire, from same address and same cycle?

1. read new value
2. read old value ⟹ depend on the FPGA family
3. undefined

Leros Implementation I/O Interface, Extensions, Software Tools

I/O Interface

- 8−bit I/O address
- read-write signal
- 16−bit input-output data signals

$\Longrightarrow$ simplicity

Extensions

- *deadline* instruction (= programmable timer)
- *memory controller* and *arbiter* attached to I/O interface, for dynamical change of content of the instruction and data memories
- additional ports

Software Tools

- Java compiler: *Muvium*

Evaluation

FPGA	Logic (LC)	Memory (blocks)	Fmax (MHz)	BRAM Fmax (MHz)
Cyclon	195	1	132	256
Cyclon II	190	1	146	235
Cyclon III	188	1	150	315
Cyclon IV	189	1	160	315
Spartan-3E	188	1	129	297
Spartan-6	112	1	182	320

Table 1 – Synthesis results of Leros for different FPGAs

Processor	Logic (LC)	Memory (blocks)	Fmax (MHz)
Leros	188	1	115
PicoBlaze	177	1	117
SpartanMC	1271	3	50

Table 2 – Comparison of Leros with PicoBlaze and SpartanMC on a Spartan XC3S500E-4

Conclusion

The Leros design is:

- open source and available from `https://github.com/schoeberl/leros`;
- versatile and portable VHDL code;
- optimised to balance the resource consumption between logic cells and on-chip memory;
- ideal for utility functions (e.g. intelligent I/O interfaces);
- small sized: many-core architectures in medium sized FPGAs.

References I

[1] M. Schoeberl.
Leros: A tiny microcontroller for fpgas.
In *2011 21st International Conference on Field Programmable Logic and Applications*, pages 10–14, 2011.