Design of a Soft-processor for Educational Purposes

Maciej Heimowski, Piotr Buluk, Aleksander Bruździak, Jakub Komoszewski, Bogdan Pankiewicz
Department of Microelectronic Systems
Gdańsk University of Technology
ul. G. Narutowicza 11/12, 80-233 Gdańsk, Poland

SUMMARY

Understanding microprocessor architecture and low-level programming is fundamental in embedded systems education. These topics are commonly taught using soft-processors implemented in FPGA platforms, which allow students to experiment with processor design and software execution. However, many existing solutions, such as PicoBlaze or Nios processors, are often tied to specific hardware vendors or include relatively complex instruction sets. The article introduces a simple, portable soft-processor designed specifically for educational use. The primary goal is to provide a system that is easy to understand, modify, and implement across different hardware platforms, while still demonstrating key architectural concepts.

The proposed processor follows a RISC-like, load/store architecture with an 8-bit data path. This choice ensures simplicity in hardware implementation while maintaining sufficient flexibility for executing typical programs. The architecture supports both Harvard and von Neumann memory models, depending on how instruction and data memory are arranged. A total of 16 registers are available. Four of them serve dedicated functions. The remaining registers (r4–r15) are general-purpose.

The processor uses a compact instruction set consisting of 16 basic instructions. Each instruction is 16 bits long and divided into four 4-bit fields (nibbles): one for the opcode and three for operands. This structure simplifies decoding and maps efficiently onto FPGA logic resources. The instruction set includes: arithmetic operations (ADD, SUB, ADDC, SUBC, ADDI), logical operations (AND, OR, XOR), shift operations (logical and arithmetic shifts), data movement (load immediate, memory load/store), control flow (conditional and unconditional branches). Despite its simplicity, the instruction set is versatile enough to implement common algorithms by combining operations. Branching is supported in two forms: relative branching (BR) for short jumps, absolute branching (B) for full address space jumps. Branch conditions are determined using the flag register, allowing flexible control flow.

To enable practical use of the processor, a dedicated assembler was developed in Python. The assembler converts assembly source code into a hexadecimal file used for initializing program memory.

The processor was implemented in Verilog using a modular structure, making it easy to analyze and modify individual components. The design is hardware-independent and does not rely on vendor-specific features, ensuring portability across different FPGA platforms.

Extensive testing was conducted: simulation using the Icarus Verilog (iverilog) simulator, automated testbenches for validating ALU operations, execution of example programs such as sorting algorithms and the Collatz sequence. Additionally, the processor was successfully deployed on multiple FPGA development boards, confirming its practical usability.

The proposed processor was compared with the PicoBlaze soft-core processor. The results show that: PicoBlaze uses fewer logic resources and achieves higher clock frequencies, the proposed processor consumes significantly less power, overall performance (instructions per second) is comparable due to differences in execution cycles per instruction. This demonstrates a trade-off between hardware efficiency and power consumption, with the proposed design favoring energy efficiency and simplicity.

To further evaluate the design, an ASIC implementation was developed using a 130 nm technology process. The processor achieved: clock frequency up to 300 MHz, very low power consumption (~1 mW), small core area. These results confirm that the architecture is not only suitable for FPGA-based education but also viable for custom integrated circuit implementations.

The article presents a simple, efficient, and portable soft-processor tailored for educational use. Its minimal instruction set and modular design make it easy to understand and modify, while still supporting meaningful programming tasks. The processor's independence from specific hardware vendors, combined with low power consumption and successful FPGA/ASIC implementations, makes it a valuable platform for teaching computer architecture, digital design, and embedded systems programming.

TABLE I.
FPGA IMPLEMENTATION RESULTS

	Picoblaze	Proposed soft-processor
Max. clock freq.	400 MHz	180 MHz
LUT	102	571
LUTRAM	24	0
FF	82	137
IO	63	94
Power	43 mW	12 mW

TABLE II.
FPGA PLATFORMS TESTED

Development Board	Synthesis & implementation tools
Aldec TySOM-2A	Xilinx Vivado
Digilent Nexys A7-100T	Xilinx Vivado
Sipeed Tang Nano 9K	OSS CAD Suite
Lattice iCE40UP5K-B-EVN	Lattice Radiant