Efficient Architecture Verification Framework with FPGA Acceleration for RISC-V Processors

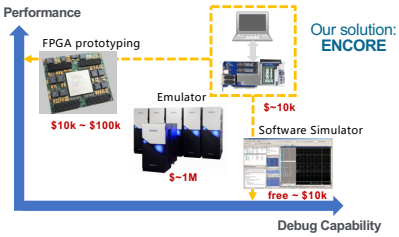
Ziqing Zhang[1,2], Shuoxiang Xu[3*], Yuhan Diao[4*], David Boland[5], Yungang Bao[1,2] and Kan Shi[1,2]

[1] State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences

[2] University of Chinese Academy of Sciences

[3] ShanghaiTech University [4] Imperial College London [5] The University of Sydney

Background



Verification triangle paradox:

- *FPGA Prototyping: high performance and low cost*
- *Emulators: debug capability and high performance*
- *Simulators: low cost and debug capability.*

Figure 1: A comparison among existing verification methods and ENCORE

Chip verification is **important** and **difficult**:

- It takes up to **70%** of the entire chip design cycle
- Even advanced verification methodologies is limited by the verification triangle paradox.

ENCORE Framework

ENCORE is a verification framework:

- Targeting **processor verification**
- With FPGA-accelerated verification + fine-grained debugging capabilities

ENCORE performs automatic checking on a single FPGA SoC:

- Implements RTL design and its SW model on the same FPGA
- Perform checking on the fly, taking hardware snapshots and offloading to software simulators to debug

ENCORE is fast

- … and capable of fine-grained debugging
- … and with low cost

ENCORE Debugging workflow

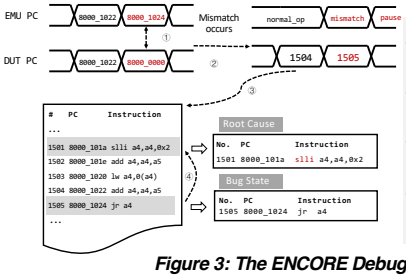


Figure 3: The ENCORE Debugging Flow Example

Once a potential issue is found by self-checking, The following procedure will be taken to reproduce the suspicious context.

- Pauses the entire hardware and the software
- Takes a snapshot of the **entire FPGA fabric**(FFs, RAM, etc.)
- Transfers all the values from the FPGA to a host PC
- Reconstructs the simulation from the paused state in an external simulator such as ModelSim.

The Structure of ENCORE

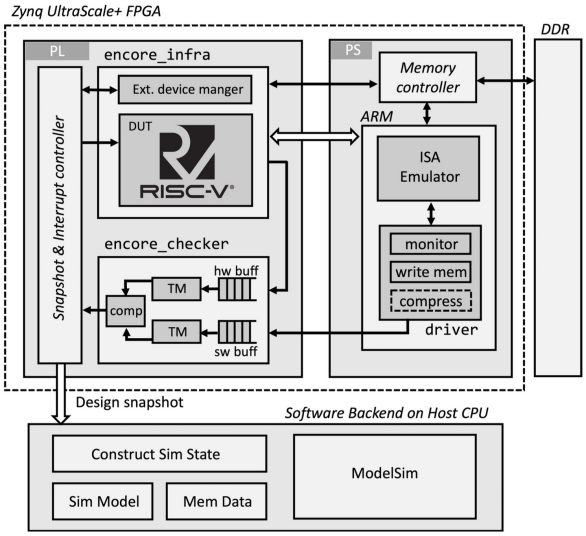


***Figure 2: The ENCORE Framework** where PL and PS represent the programming logic and processing system on the FPGA, respectively.*

The key functions of ENCORE are to perform

- **automatic self-checking** during operation
- **snapshot the design** once a potential issue is identified.

Hardware architecture.

- **encore_infra:** the hardware infrastructure.
- **encore_checker:** the module responsible for test data collection and automatic self-checking.
- **snapshot and interrupt controller:** create snapshots of the current design and initiate interruptions.

Software architecture.

- **the software ISA emulator:** serve as the "golden reference model" to ensure instruction-level accuracy against DUT
- **customized driver:** fetch the key signals state, and write this essential information to the software buffer for comparison

Experiment Setup

Experiment platform:

- FPGA Board: Fidus Sidewinder board
  - With AMD ZYNQ UltraScale+ XCZU19EG FPGA and two 16GB DDR4 memories.
- Host Server: with two AMD Ryzen 5950x 16-core processors

Experiment Configuration:

- System Clock Frequency: 150MHz.
- DUT: NutShell,
  - an open-source 64-bit RISC-V processor core capable of booting Linux.
- ISA emulator: NEMU

Evaluation Result

Performance: compare with SW simulation only

- SW simulation: using Modelsim or Verilator
- **ENCORE is much faster than Verilator: 84x ~ 139x**
- **ENCORE is much faster than Modelsim: 19264x ~ 44187x**

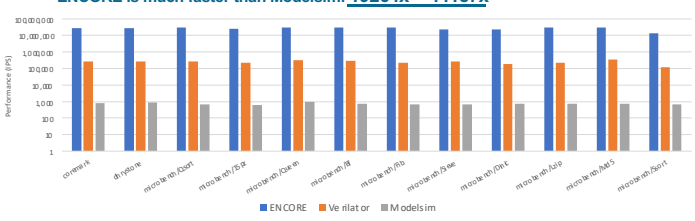


Figure 4: Performance comparison between ENCORE and simulation-only approach.

Resource Usage: compare with conventional debugging approach

- ILA-based: Integrated Logic Analyzer
- **Conventional**: Recompile needed when adding new signals
- **ENCORE**: No recompilation needed, snapshot can view all signals

Two configurations: PC+1 CSR (config1), PC + 32 CSRs (config2)

Table1: Resource Usages of ENCORE and the Example Design

Resource	ENCORE (config1)	ENCORE (config2)	ILA (config1)	ILA (config2)	DUT (Nutshell)
LUTs	563 (0.11%)	1597 (0.31%)	885 (0.17%)	6112 (1.17%)	41589 (7.96%)
Block RAMs	12 (1.22%)	12 (1.22%)	24 (2.44%)	465 (47.26%)	281 (28.6%)
Registers	1429 (0.14%)	1493 (0.14%)	1615 (0.15%)	11194 (1.07%)	57609 (5.51%)