SCIENCE ▪ PASSION ▪ TECHNOLOGY

Institute of Technical Informatics ■ Inffeldgasse 16/I, 8010 Graz, Austria ■ *office@iti.tugraz.at*

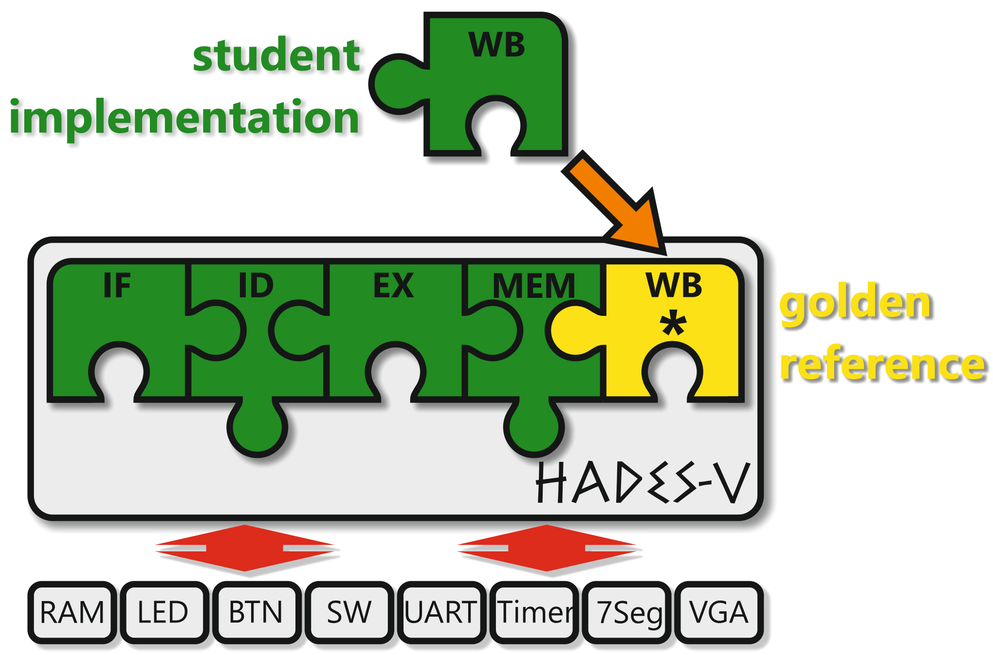
HADES-V

Learning by Puzzling: A Modular Approach to RISC-V Processor Design Education

Tobias Scheipel, David Beikircher, Florian Riedl

Institute of Technical Informatics, Graz University of Technology
tobias.scheipel@tugraz.at

Ever thought about developing a processor from scratch and bringing it to life on an FPGA? With HADES-V, you'll delve into hardware design and create your own pipelined 32-bit RISC-V processor, mastering efficient computing principles and practical FPGA implementation.



SV fetch_stage.sv

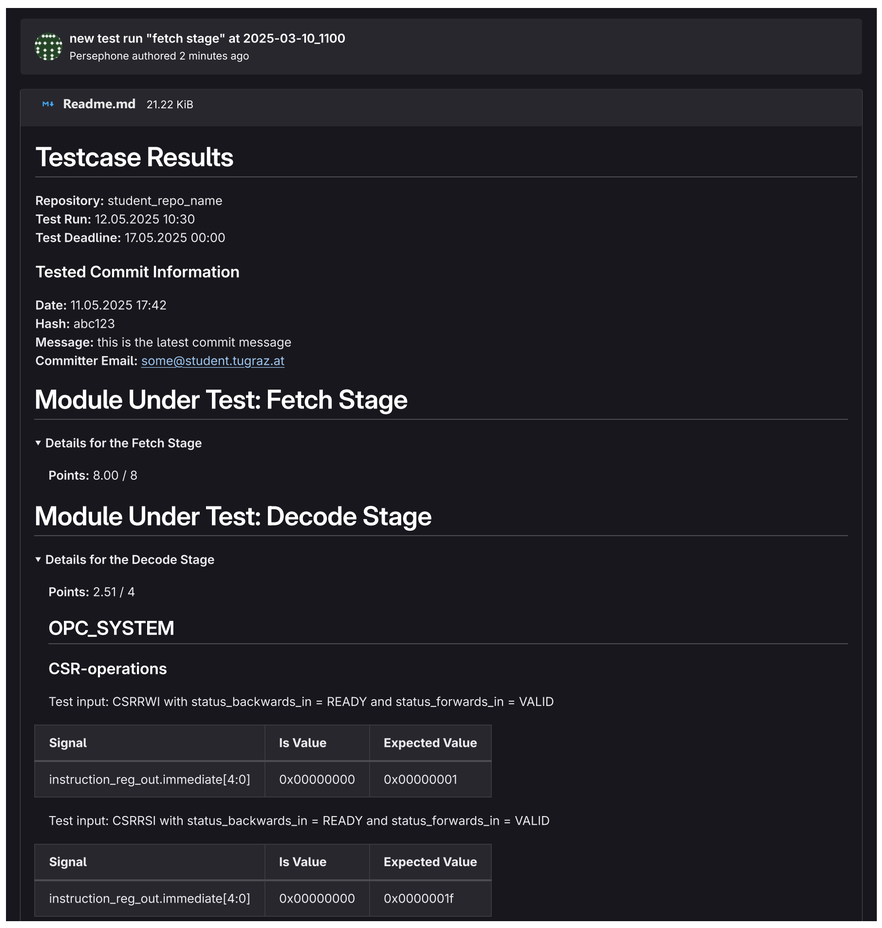
```sv
module fetch_stage (
  input logic clk,
  input logic rst,

  // Memory interface
  wishbone_interface.master wb,

  //  Output data
  output logic [31:0] instruction_reg_out,
  output logic [31:0] program_counter_reg_out,

  // Pipeline control
  output pipeline_status::forwards_t  status_forwards_out,
  input  pipeline_status::backwards_t status_backwards_in,
  input  logic [31:0] jump_address_backwards_in
);

  // TODO: Delete the following line and implement this module.
  ref_fetch_stage golden(.*);

endmodule
```

Key Features & Approach

- **Jigsaw Puzzle Methodology:**
  - Modular Pipeline Stages (IF, ID, EX, MEM, WB) built one at a time.
  - Golden References (precompiled Verilator [1] libraries) for validation.
- **Hands-On Tools:**
  - SystemVerilog for hardware description.
  - Verilator [1] & GTKWave [4] for simulation.
  - AMD Vivado [2] for synthesis on Basys3 [3] board.
- **Open Educational Resource (OER):**
  - **Instruction Guide** [5] and open-source **Code Template** [6].
  - Material licensed under CC BY 4.0 and MIT licenses.
    → Encourages global collaboration.

Testing & Assessment

- GIT-based test system PERSEPHONE.
- Automated grading and detailed student performance insights.
- Immediate feedback via pass/fail test outputs.
  → debugging, iterative refinement, and solution ownership.



new test run "fetch stage" at 2025-03-10_1100
Persephone authored 2 minutes ago

Readme.md 21.22 KiB

Testcase Results

Repository: student_repo_name
Test Run: 12.05.2025 10:30
Test Deadline: 17.05.2025 00:00

Tested Commit Information

Date: 11.05.2025 17:42
Hash: abc123
Message: this is the latest commit message
Committer Email: some@student.tugraz.at

Module Under Test: Fetch Stage

▾ Details for the Fetch Stage

Points: 8.00 / 8

Module Under Test: Decode Stage

▾ Details for the Decode Stage

Points: 2.51 / 4

OPC_SYSTEM

CSR-operations

Test input: CSRRWI with status_backwards_in = READY and status_forwards_in = VALID

Signal	Is Value	Expected Value
instruction_reg_out.immediate[4:0]	0x00000000	0x00000001

Test input: CSRRSI with status_backwards_in = READY and status_forwards_in = VALID

Signal	Is Value	Expected Value
instruction_reg_out.immediate[4:0]	0x00000000	0x0000001f

Didactical Structure

- **Incremental Modules:** Students tackle increasing complexity without being overwhelmed.
- **Collaborative Learning:** Peer reviews, group discussions, online chat room, and project showcases.
- **Real-World Implementation:** Final designs synthesized on an FPGA, subsequently reinforcing HW/SW co-design.
- **Creativity Incentive:** The HADES-V Award for standout implementations and custom extensions.

github.com/tscheipel/HaDes-V

Bibliography

[1] Wilson Snyder. Verilator. https://verilator.org/.
[2] Advanced Micro Devices, Inc. Vivado Design Suite. https://www.xilinx.com/support/download.html.
[3] Basys 3 Reference Manual. Digilent, Inc. https://reference.digilentinc.com/reference/programmable-logic/basys-3/reference-manual.
[4] Tony Bybell. GTKWave. https://gtkwave.sourceforge.net/.
[5] Tobias Scheipel, David Beikircher, Florian Riedl. Microcontroller Design, Lab: HaDes-V Instruction Guide. https://doi.org/10.3217/nytm4-grv34.
[6] Tobias Scheipel, David Beikircher, Florian Riedl. HaDes-V Template Repository. https://github.com/tscheipel/HaDes-V.

scheipel.com/oer