Testing and Fault Tolerance
(01RKZOQ / 01RKZOV / 01RKZQW)

Assignments

2023/2024

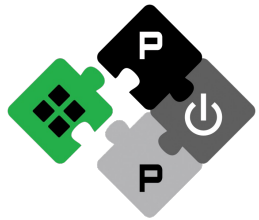
Two options (choose one)

1) Testing assignment

2) Fault tolerance assignment

DUT: cv32e40p (for both)

- Originally known as the PULP RI5CY core, the CORE-V CV32E40P is a 32bit, 4-stage RISC-V core that implements, RV32IMFC[Xpulp], has an optional 32-bit FPU supporting the F and Zfinx extensions and custom instruction set extensions for DSP operations.
- We have forked the original repo and customized the simulation environment: https://github.com/cad-polito-it/cv32e40p_tftlab
- The synthesis files are **not** included in the repo.



Assignment 1: Testing

- **Goal**: to evaluate the effectiveness of SBST in testing delay faults
- You are asked to:
  - Analyze a delay fault list (with timing information)
  - Select meaningful faults to inject in the fault simulations
  - Build a SBST program able to cover as much as possible the faults of interest (you can start from available SBST programs developed for transition faults)

- Refer to the following page for more details: https://github.com/cad-polito-it/cv32e40p_tftlab/blob/tftlab/Testing_General_Assignment_2023.md

Assignment 1 toolkit

- PrimeTime (PT) flow to generate timing information (slack) for each fault site
- QuestaSim logic simulation flow (with or without timing information)
- Z01X fault simulation flow for stuck-at, transition faults, and small delay defects
- A set of SBST programs to starts with
- Python script to create a Z01X fault list from PT report, using some parameters:
  - K: multiply each PT-computed slack by a factor K
  - M: filter out fault sites with a slack > M
  - S: ignore the PT-computed slack and use a delay S

Assignment 2: Fault tolerance

- **Goal**: to harden the core and evaluate the effectiveness of the proposed solution in terms of fault coverage
- You are asked to:
  - Analyze the RTL and find possible points (e.g., registers) to harden (e.g., via duplication, triplication, checkers on specific point of the circuits, parity, etc.)
  - Synthesize the circuit and find the new “test points” in the gate-level netlist to use for the fault simulations
  - Run some SBST programs to show the fault coverage gain using your solution.

- Refer to the following page for more details: https://github.com/cad-polito-it/cv32e40p_tftlab/blob/tftlab/Fault_Tolerance_General_Assignment_2023.md

Assignment 2 toolkit

- Design Compiler synthesis flow
- QuestaSim logic simulation flow
- Z01X fault simulation flow
- A set of SBST programs (a few adjustments may be needed).

Procedural issues

- You are requested to
  - Work alone or in groups (max 3 students per group)
  - Create or join a group using the section “Groups for assignments” of Exercise (by Dec. 17)
  - Complete their work by Jan. 28, 2024 (included), submitting a single zip/rar file for each group containing the following data
    - The project folder (with all scripts needed to re-run the flow). You can keep the fault reports and do “make clean” to save space.
    - A short report
- A meeting will be organized with each group (after Jan. 28) to check the results.

The report

- It must include a description of
  - The analysis you did on the fault list, the timing on the various modules you have analyzed, how have how applied the timing in the fault simulation…
  - Which results you achieved in terms of fault coverage and test application time.
- Reasonable size: 5 pages (do not make it too long)

Limitations

- Each group can launch only one synthesis, logic simulation, or fault simulation experiment at a time
- Be careful: endless simulation (e.g., due to exceptions or endless loops) may saturate the available disk space. Please check!

Hint: use Git

1. Fork the “tftlab” branch of the repo in your GitHub account (one fork per group is enough)
2. Create one branch per person (your own “master” branch)
3. Use git merge on the tftlab branch once you have something to share with the others in the group

Do not upload on GitHub (use .gitignore):

- The technology library files (techlib folder)
- The preliminary SBST programs received in the toolkit.