DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
UNIVERSITY OF TEHRAN

Fault Tolerant Systems
Fall 2013

Computer Assignment 1

Kamyar Mohajerani

November 11, 2013

1 VERILOG PROGRAMMING LANGUAGE INTERFACE 2.0

A generic VPI (PLI 2.0) code has been developed from scratch. The code includes an `inject_faults` function:

```
void inject_faults(vpiHandle module, int num_faults, int max_depth);
```

- `module` is the handle to the module which is to be examined.
- `num_faults` is the number of random stuck-at faults to be injected into the modules.
- `max_depth` is an optional arguments and is the maximum depth of traversal into sub-modules. When `max_depth = 0` only nets from the top module are considered.

Two PLI task functions have been developed:

`$inject_hlf(<`**module**–name>, <num_faults>)

and

`$inject_hlf_sub(<`**module**–name> ,<num_faults>, <submodule_index>)

One of these two PLI tasks is called depending on whether FAULT_DELAY or MAIN_FAULT are defined. These defines are passed to the simulator as command-line arguments.

```
initial begin
        #`FAULT_DELAY;
`ifdef SUBMODULE_FAULT
    $inject_hlf_sub(SayehTest.U1, `num_faults, `submodule_index);
`elsif MAIN_FAULT
    $inject_hlf(SayehTest.U1, `num_faults);
`endif
```

2 Assembler

An almost complete assembler for the Sayeh processor was developed to translate assembly code into binary machine code. Some features of the assembler include:

- named data section which can be referenced in code
- use of labels for jumps
- literal arithmetics
- simple and extendable Python code

3 Random Faults

4 Random Faults on Each Module

The results can be seen in figure 4.1. Most vulnerable modules are InstructionRegister and WindowPointer.

Figure 3.1: results for 5 random faults in 35 iterations

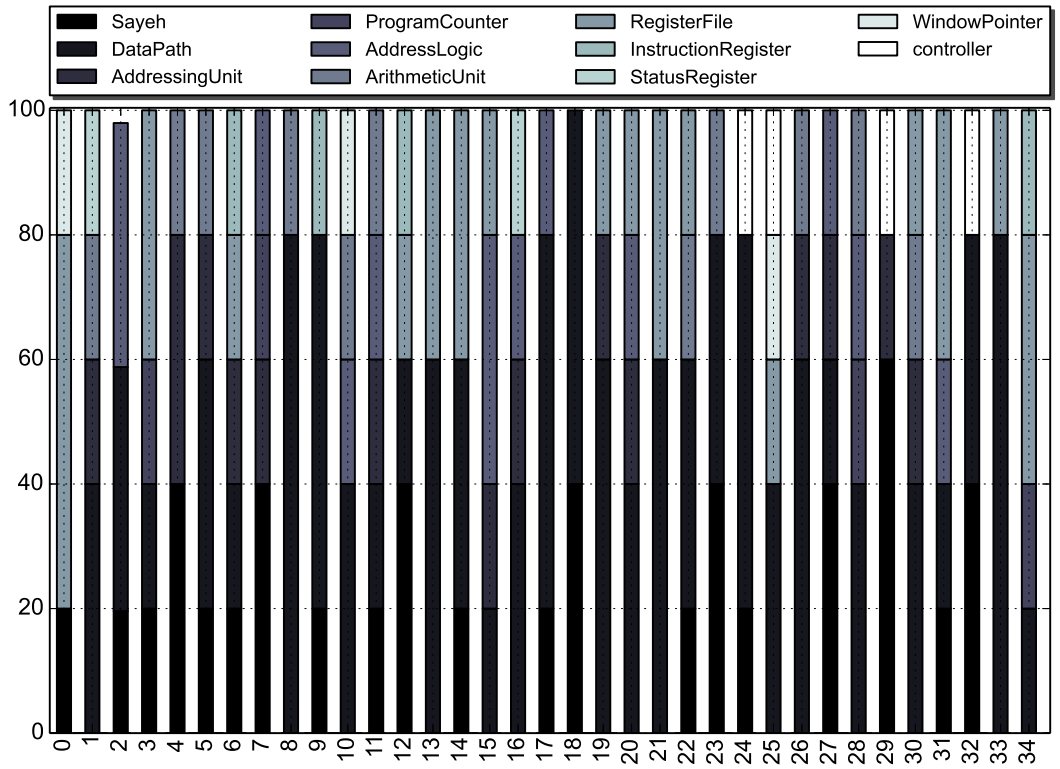

Figure 3.2: results for 2 random faults in 35 iterations

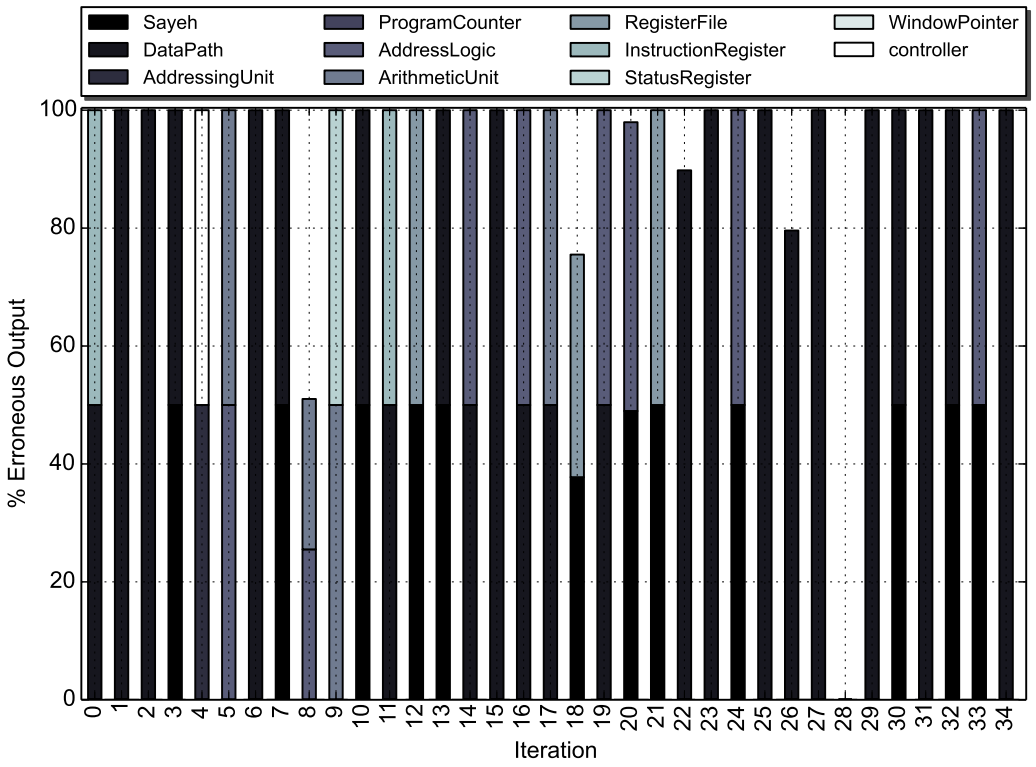

Figure 4.1: results for 1, 2, 3, and 5 random faults in 7 modules for 5 iterations