# Assignment 5

## Hardware Fuzzing

Name: Daniel Horan

UIN: 527005307

## Introduction.

In this assignment, the focus is on the development and application of hardware fuzzing techniques to identify vulnerabilities in hardware designs. The main goal is completing and refining an incomplete hardware fuzzer, TheHuzz, specifically designed for the CVA6 processor. The primary objectives include building the missing components of TheHuzz, employing it to test the CVA6 processor, and rigorously analyzing its performance in uncovering potential vulnerabilities. This lab aims to provide a comprehensive understanding of hardware fuzzing as a critical tool in hardware security, highlighting its importance in the early detection and mitigation of security flaws in hardware designs.

## Task F1.

The objective of Task F1 was to develop the 'compute_cov_achieved' function in TheHuzz's feedback engine, which is important for evaluating the fuzzer's effectiveness. Located in fuzz.py, this function calculated the total coverage points achieved by the fuzzer and the corresponding percentage of coverage. It achieved this by iterating through the merged_cov_dict, counting '1's in each binary string to determine coverage points for each type, and then aggregating these counts. The function successfully provided a clear metric of the fuzzer's performance, enhancing its ability to identify vulnerabilities in hardware designs.

## Task  F2.

The objective of Task F2 was to implement the 'get_testcases_to_sim' function in the thehuzz_utils.py module of TheHuzz fuzzer. This function provides a specified number of test cases from the input database to the fuzzer for processing the processor. The implementation involved verifying the availability of the requested number of test cases in the database and then retrieving them accordingly. This function ensured that the fuzzer had an adequate and appropriate set of test cases for effective fuzzing. The task was completed successfully, enabling TheHuzz to systematically access and utilize test cases from the database, thereby streamlining the fuzzing process for processor testing.

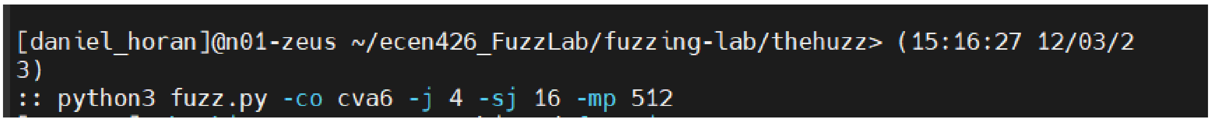
## Task  F3.

The objective Task F3 is focused on implementing the 'calc_no_times_to_mut' function within TheHuzz's fuzz.py file. The primary objective of this function was to evaluate which test cases contributed to increased coverage and determine their suitability for further mutations. The function's logic involved assessing each test case's impact on coverage and, based on this analysis, deciding whether to mutate these test cases and the number of mutations to apply. The implementation process entailed developing algorithms to measure coverage increments and adjust the mutation frequency, ensuring a balanced approach to mutating test cases. This task was completed, enhancing the feedback engine's ability to select and mutate test cases effectively.

## Task  F4.

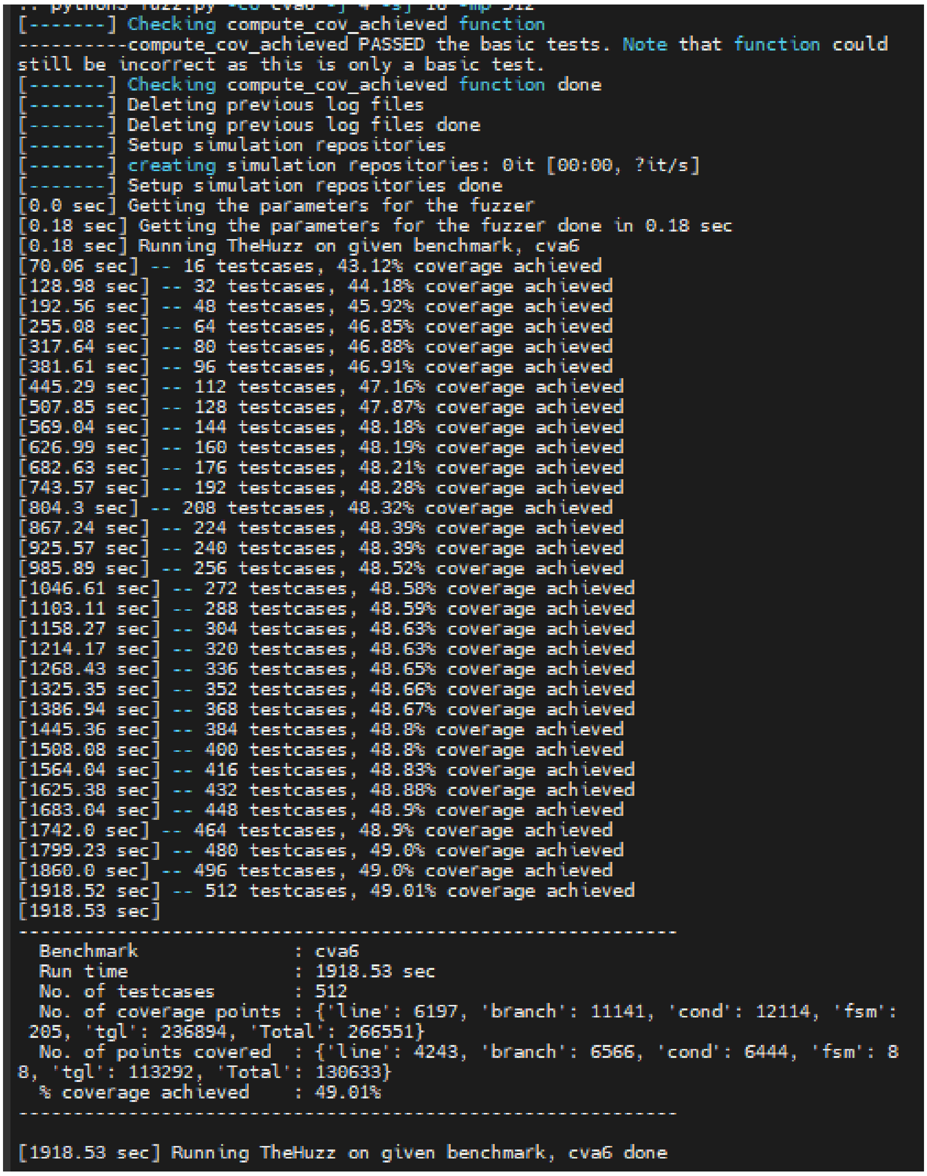
The objective of Task F4 was the development of the 'my_random' function within the prog_mut.py module of TheHuzz. This function's primary purpose was to execute random mutations on specific bits within an instruction. The process entailed identifying predetermined bit positions and subsequently applying stochastic alterations to these bits. The implementation focused on ensuring that these mutations were random yet confined to designated bits. Upon successful completion, this task improved the mutation engine of TheHuzz by incorporating an algorithm for randomized bit mutations.

## Terminal.



```
[daniel_horan]@n01-zeus ~/ecen426_FuzzLab/fuzzing-lab/thehuzz> (15:16:27 12/03/2
3)
:: python3 fuzz.py -co cva6 -j 4 -sj 16 -mp 512
```

Screenshot 1: *Command used*



```
:: python3 fuzz.py -co cva6 -j 4 -sj 16 -mp 512
[-------] Checking compute_cov_achieved function
----------compute_cov_achieved PASSED the basic tests. Note that function could
still be incorrect as this is only a basic test.
[-------] Checking compute_cov_achieved function done
[-------] Deleting previous log files
[-------] Deleting previous log files done
[-------] Setup simulation repositories
[-------] creating simulation repositories: 0it [00:00, ?it/s]
[-------] Setup simulation repositories done
[0.0 sec] Getting the parameters for the fuzzer
[0.18 sec] Getting the parameters for the fuzzer done in 0.18 sec
[0.18 sec] Running TheHuzz on given benchmark, cva6
[70.06 sec] -- 16 testcases, 43.12% coverage achieved
[128.98 sec] -- 32 testcases, 44.18% coverage achieved
[192.56 sec] -- 48 testcases, 45.92% coverage achieved
[255.08 sec] -- 64 testcases, 46.85% coverage achieved
[317.64 sec] -- 80 testcases, 46.88% coverage achieved
[381.61 sec] -- 96 testcases, 46.91% coverage achieved
[445.29 sec] -- 112 testcases, 47.16% coverage achieved
[507.85 sec] -- 128 testcases, 47.87% coverage achieved
[569.04 sec] -- 144 testcases, 48.18% coverage achieved
[626.99 sec] -- 160 testcases, 48.19% coverage achieved
[682.63 sec] -- 176 testcases, 48.21% coverage achieved
[743.57 sec] -- 192 testcases, 48.28% coverage achieved
[804.3 sec] -- 208 testcases, 48.32% coverage achieved
[867.24 sec] -- 224 testcases, 48.39% coverage achieved
[925.57 sec] -- 240 testcases, 48.39% coverage achieved
[985.89 sec] -- 256 testcases, 48.52% coverage achieved
[1046.61 sec] -- 272 testcases, 48.58% coverage achieved
[1103.11 sec] -- 288 testcases, 48.59% coverage achieved
[1158.27 sec] -- 304 testcases, 48.63% coverage achieved
[1214.17 sec] -- 320 testcases, 48.63% coverage achieved
[1268.43 sec] -- 336 testcases, 48.65% coverage achieved
[1325.35 sec] -- 352 testcases, 48.66% coverage achieved
[1386.94 sec] -- 368 testcases, 48.67% coverage achieved
[1445.36 sec] -- 384 testcases, 48.8% coverage achieved
[1508.08 sec] -- 400 testcases, 48.8% coverage achieved
[1564.04 sec] -- 416 testcases, 48.83% coverage achieved
[1625.38 sec] -- 432 testcases, 48.88% coverage achieved
[1683.04 sec] -- 448 testcases, 48.9% coverage achieved
[1742.0 sec] -- 464 testcases, 48.9% coverage achieved
[1799.23 sec] -- 480 testcases, 49.0% coverage achieved
[1860.0 sec] -- 496 testcases, 49.0% coverage achieved
[1918.52 sec] -- 512 testcases, 49.01% coverage achieved
[1918.53 sec]
------------------------------------------------------------
  Benchmark              : cva6
  Run time               : 1918.53 sec
  No. of testcases       : 512
  No. of coverage points : {'line': 6197, 'branch': 11141, 'cond': 12114, 'fsm':
 205, 'tgl': 236894, 'Total': 266551}
  No. of points covered  : {'line': 4243, 'branch': 6566, 'cond': 6444, 'fsm': 8
8, 'tgl': 113292, 'Total': 130633}
  % coverage achieved    : 49.01%
------------------------------------------------------------

[1918.53 sec] Running TheHuzz on given benchmark, cva6 done
```

Screenshot 2: *The final output of the script*

## Feedback:

The assignment focused on developing and analyzing TheHuzz's components, mirroring real-world hardware security tasks. Though time-consuming, the process was relevant, providing practical insights into the field's challenges. The task emphasized the need for detailed and accurate work in hardware security, showcasing the critical aspects of hardware fuzzing.