ECE 464

Project 3 - Sequential Circuit Study Report

Radoslaw Jelonkiewicz **Krishna Vivek Reddy** **Sai Venkata Dhruva Mandala**

December 3, 2019

1.Introduction

Theory: Sequential Logic Circuit is type of circuit, that depends not only on present input like **Combinational Logic** , but also depends on the past ones. The main difference between **Combinational Logic** , and **Sequential Logic** is, SL has state (memory).

a) Goal of the investigation:

For this experiment we will be analyzing the behaviour of certain test vectors for sequential circuit **s27.bench**.

2. Experiment Procedure

For this experiment you will need:

1. Small **s27.bench** file
2. Full Fault list generator which will generate .txt file in order to be able to refer which fault to use.
3. The simulator code **p3sim.py** with extra code to include fault list generation.

a)Generating the fault list:

In order to know which fault we want to use, we first have to obtain the entire list of faults to choose from. This can be done by using full fault list generation code.

1. The number of faults for **s27.bench** is equal to **76**.

b)Test Vectors:

For this experiment we need to input our values manually. Here we have taken a sample of 4 random test vectors with 4 faults to observe the behaviour. The test vectors are:

1. **3**
2. **1**2
3. **1**5
4. **6**

After that we can proceed to the experiment itself. After inputting data into our simulator. The short description of the features of the simulator below:

1. Prompt the user to input a sequential bench file (default: circ.bench)
2. prompt the user to input a TV t in integer (default: t=0) Note: you should support negative integers with t=-1 for a TV with all 1's, t=- 2 for a TV of 1111...110, etc
3. Prompt the user to input cycle number n (default: n=5)
    - Output circuit simulation result (all FF's content, values of the Primary Output) for a good circuit with the same TV t applying on the Primary Input, for n cycles.
4. Prompt the user to type in a single fault f (default: the first input line of the bench circuit stuck-at- 0)

- Output fault simulation result (all FF's content, values of the Primary Output) for a bad circuit under fault f, with the same TV t applying on the Primary Input, for n cycles.
- Summarize whether f is detected or not, in which cycle.

Below: Running the simulator

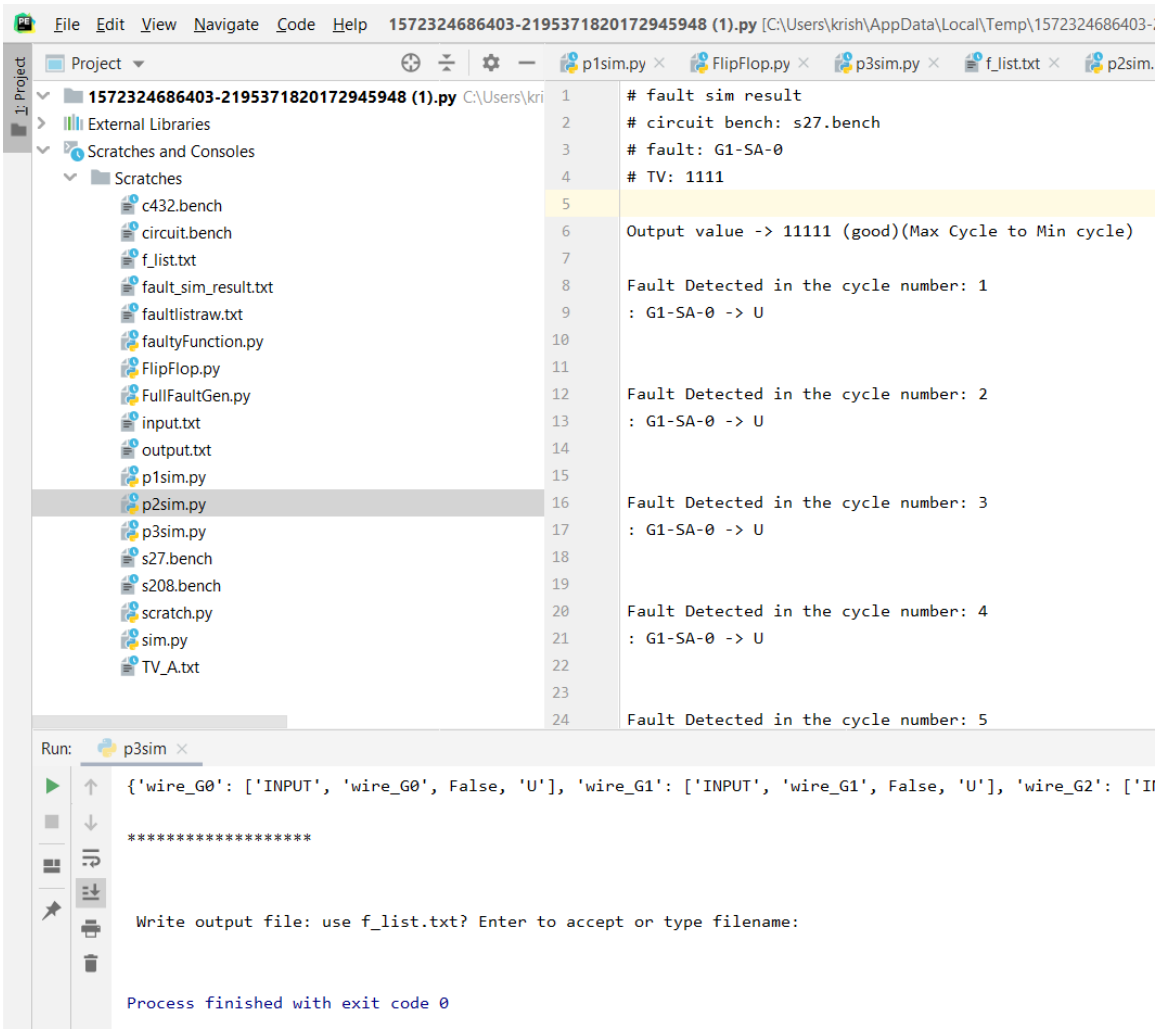

Figure 1: TV 15

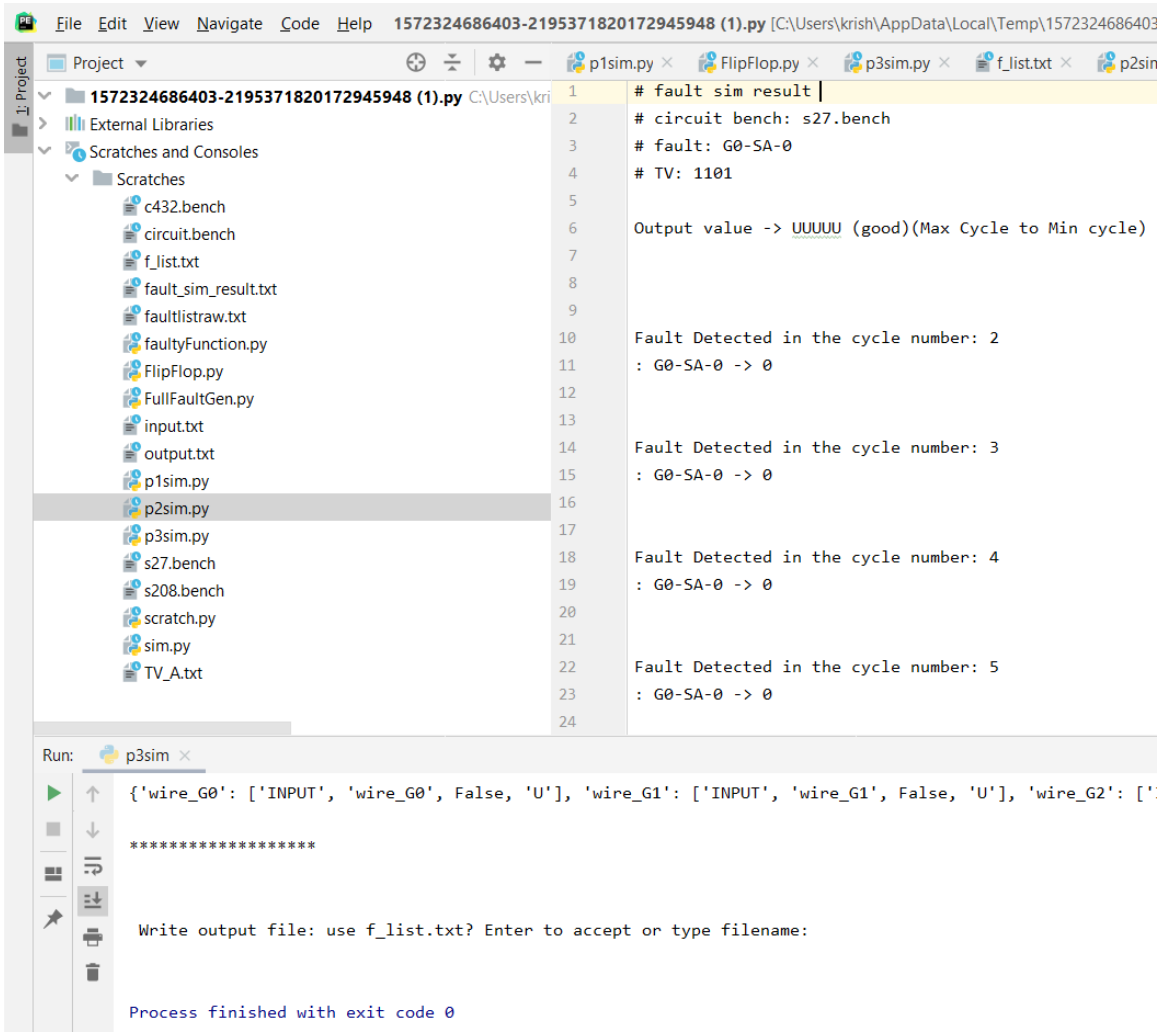

Figure 2: TV 13

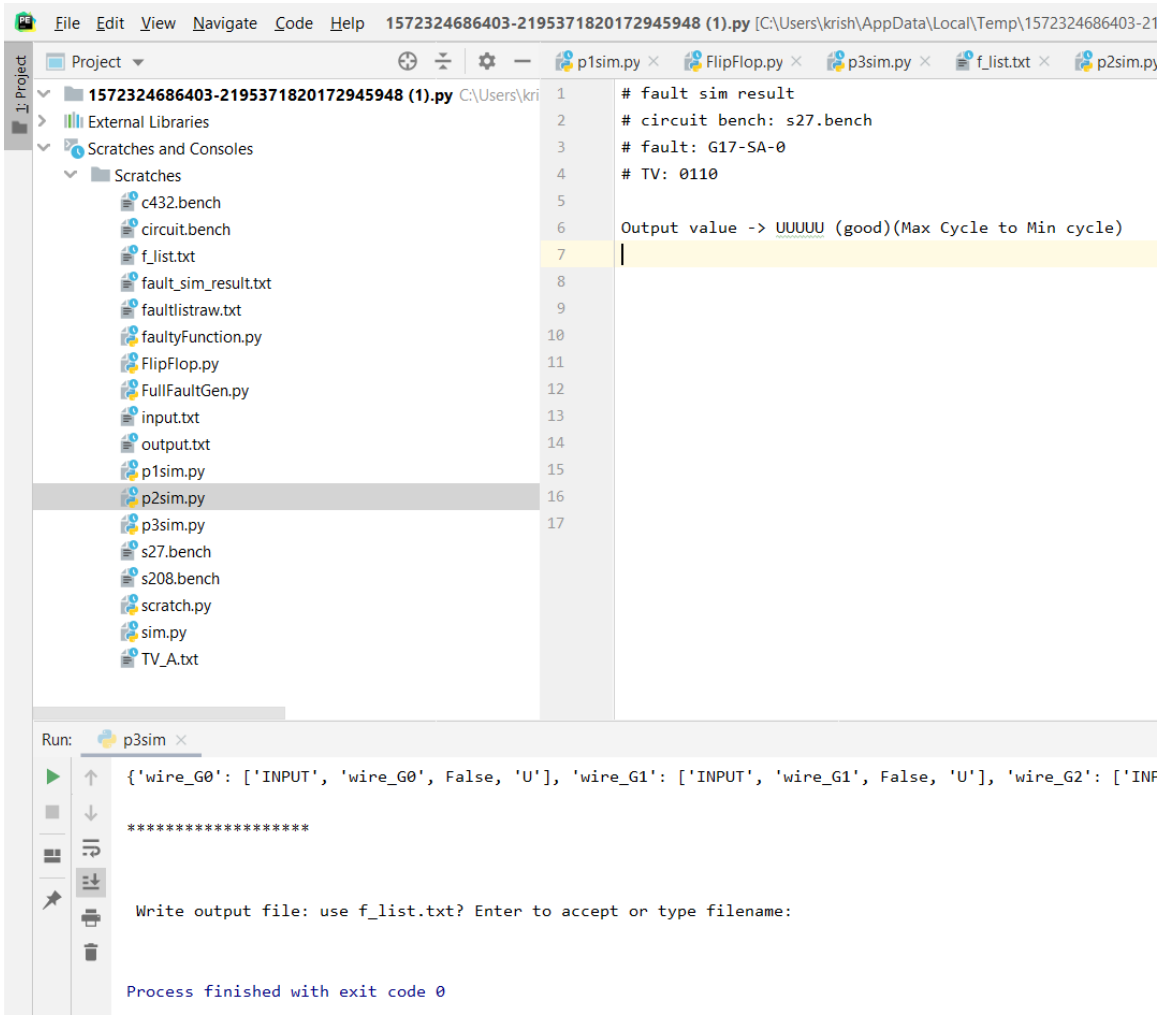
File Edit View Navigate Code Help 1572324686403-2195371820172945948 (1).py [C:\Users\krish\AppData\Local\Temp\1572324686403-21
Project
1572324686403-2195371820172945948 (1).py C:\Users\kri
External Libraries
Scratches and Consoles
Scratches
c432.bench
circuit.bench
f_list.txt
fault_sim_result.txt
faultlistraw.txt
faultyFunction.py
FlipFlop.py
FullFaultGen.py
input.txt
output.txt
p1sim.py
p2sim.py
p3sim.py
s27.bench
s208.bench
scratch.py
sim.py
TV_A.txt
p1sim.py
FlipFlop.py
p3sim.py
f_list.txt
p2sim.py
fault sim result
circuit bench: s27.bench
fault: G17-SA-0
TV: 0110
Output value -> UUUUU (good)(Max Cycle to Min cycle)
Run:
p3sim
{'wire_G0': ['INPUT', 'wire_G0', False, 'U'], 'wire_G1': ['INPUT', 'wire_G1', False, 'U'], 'wire_G2': ['INP
****
Write output file: use f_list.txt? Enter to accept or type filename:
Process finished with exit code 0

Figure 3: TV 6

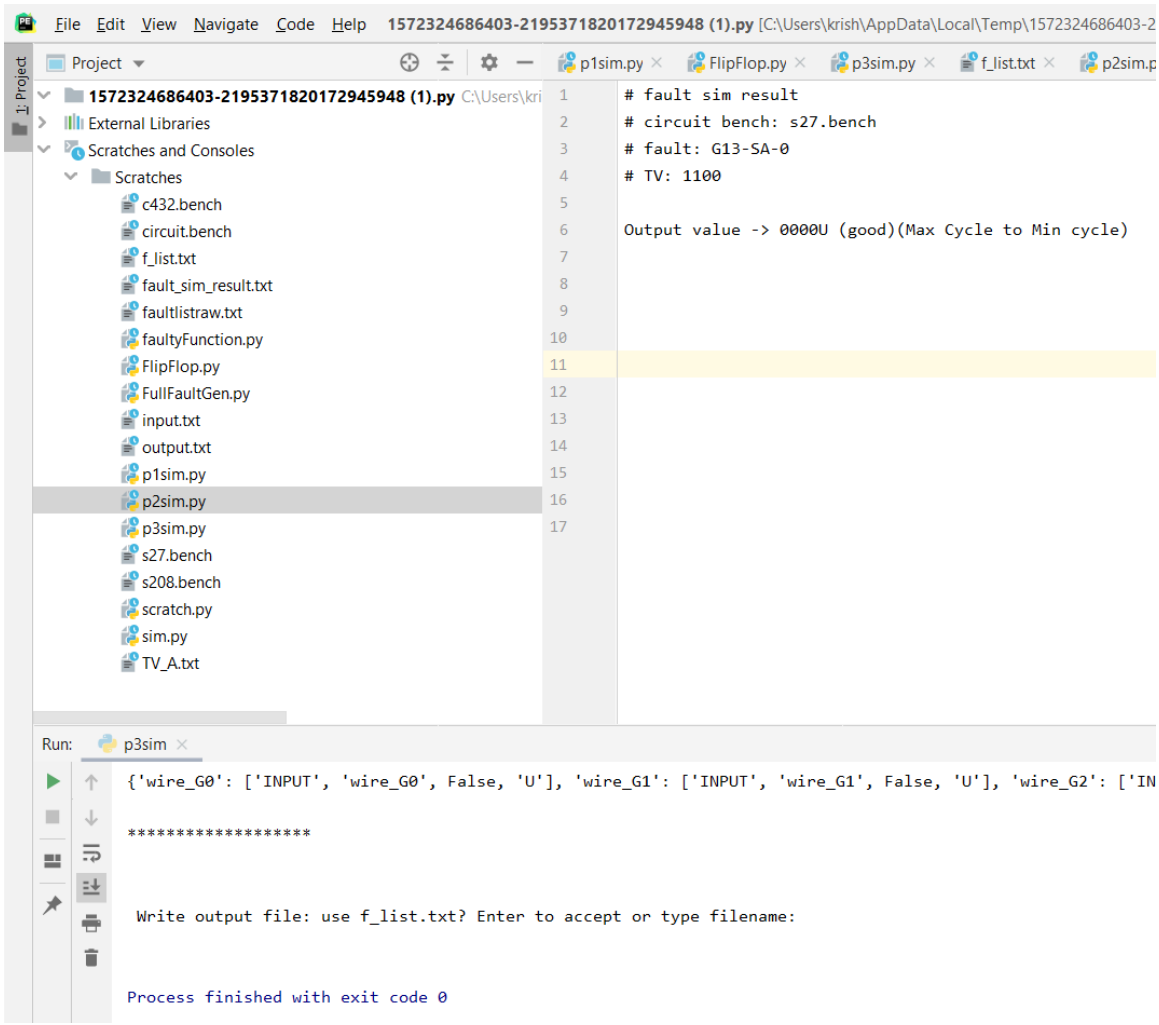

Figure 4: TV 12

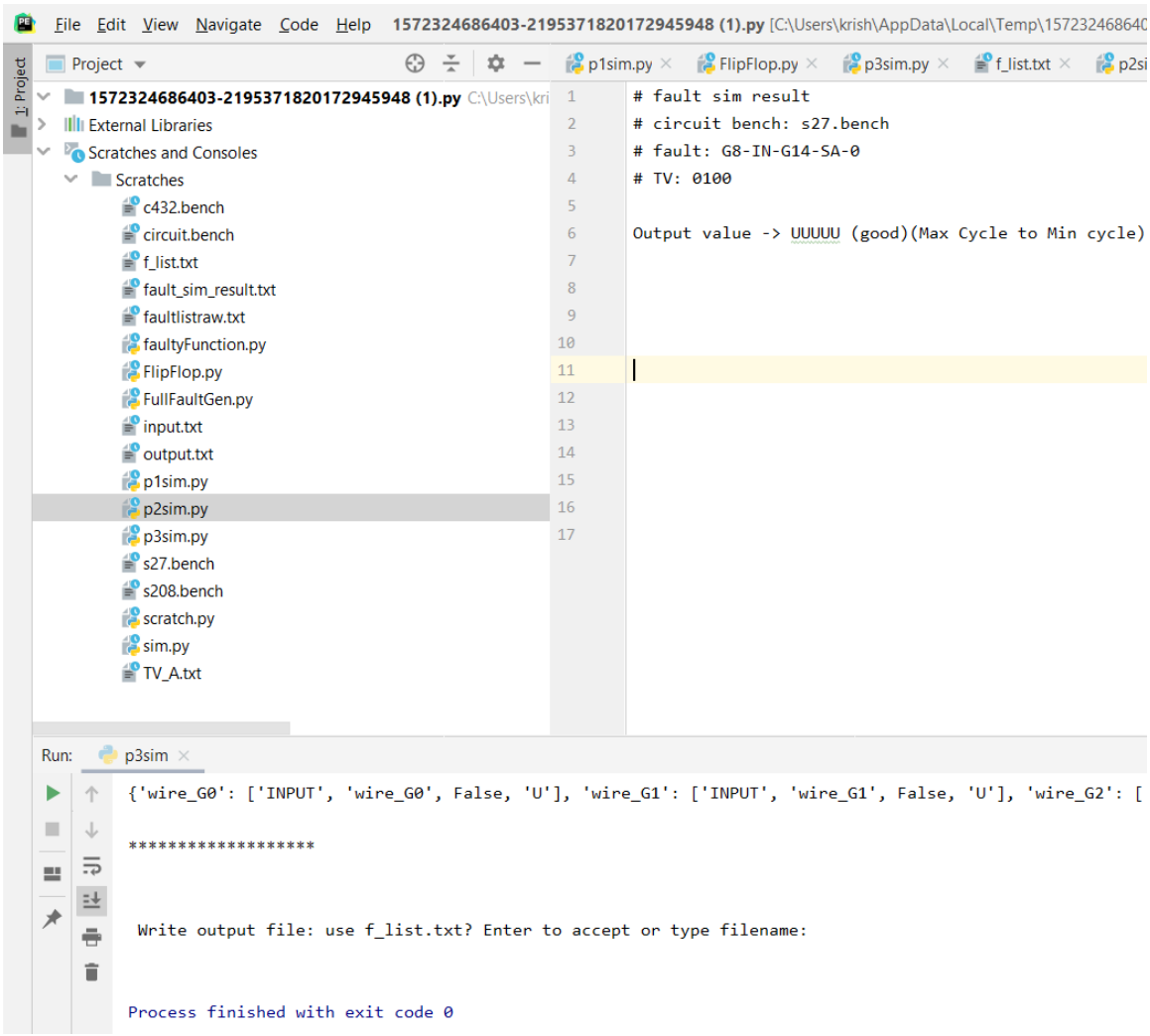

Figure 5: TV -12

3. Experiment Results

These are the following results of the experiment:

Test Vector	Fault	Good Output	Fault Output	Detected
15	G1-SA-0	11111	UUUUU	Yes
13	G0-SA-0	UUUUU	00000	Yes
6	G17-SA-0	UUUUU	UUUUU	No
12	G13-SA-0	0000U	0000U	No
-12	G8-IN-G14-SA-0	UUUUU	UUUUU	No

Table 1: Test Vectors

4. Explanation of the result

As we can see from **Table 1** Sequential Logic circuit has an average probability to find fault at randomly picked Test Vector. In experiment out of 5 integers we were able to find fault only at **two** of them.

5. Was this expected to happen?

The answer is **yes** - Finding faults like that in something as complex as Sequential Logic will take time to detect faults and it will be time consuming. At this experiment itself we tested 5 different Test Vectors, from which only 2 were able to detect the fault. At expanded time it will take more and more time to detect all the faults. In actual application we wouldn't want to do it manually as we are doing in **Project 3**, in reality we would want to use proper algorithms/methods to test it, such as **path-sensitizing method**.

6. How it applies to 464 material?

During our semester and past 3 projects we were mostly concentrated around **Combinational Logic Design** only. However we never did anything related to **Sequential Logic Design**. This project/experiment wraps this topic, by making an actual application of the material. It was important to cover this material, as most of the modern digital devices consist of both **Sequential** and **Combinational** Logic. Nearly every device nowadays is strictly dependent on the memory and how clock works, that why this experiment is conclusion to how it operates, and how we apply theory learned during the semester into real life application.