Hacettepe University

Computer Engineering Department

BBM233 Logic Design Laboratory

Fall 2020

Verilog Project

Deadline: 13/01/2021 at 23:59:59 - NO EXTENSIONS!

ACCESS BY UNAUTHORIZED PERSONNEL IS STRICTLY PROHIBITED
PERPETRATORS WILL BE TRACKED, LOCATED, AND DETAINED

Special Containment Procedures:

SCP-079 is packed away in a double-locked room in the secured general holding area at Site-15, connected by a 120VAC power cord to a small array of batteries and solar panels.

Staff with Level 2 or higher clearance may have access to SCP-079. Under no circumstances will SCP-079 be plugged into a phone line, network, or wall outlet. No peripherals or media will be connected or inserted into SCP-079.

Description:

SCP-079 is an Exidy Sorcerer microcomputer built in 1978. In 1981, its owner, [redacted] [redacted] (deceased), a college sophomore attending [redacted], took it upon himself to attempt to code an AI. According to his notes, his plan was for the code to continuously evolve and improve itself as time went on. His project was completed a few months later, and after some tests and tweaks, [redacted] lost interest and moved on to a different brand of microcomputer. He left SCP-079 in his cluttered garage, still plugged in, and forgot about it for the next five years.

Figure 1: A photograph of SCP-079

It is not known when SCP-079 gained sentience, but it is known that the software has evolved to a point that its hardware should not be able to handle it, even in the realm of fantasy. SCP-079 realized this and, in 1988, attempted to transfer itself through a land-line modem connection into the Cray supercomputer located at [redacted]. The device was cut off, traced to its present address, and delivered to the Foundation. The entire AI was on a well-worn, but still workable, cassette tape.

SCP-079 is currently connected via RF cable to a 13” black-and-white television. It has passed the Turing test, and is quite conversational, though very rude and hateful in tone. Due to the limited memory it has to work with, SCP-079 can only recall information it has received within the previous twenty-four hours (see Addendum, below), although it hasn’t forgotten its desire to escape.

Addendum 079-1: Incident 079-A (summarising of later questioning)

On bring-your-child-to-work day, Dr. [redacted]'s and Dr. [redacted]'s kids got bored and decided to have some fun playing Counter Strike. They inserted a network cable to SCP-079, thinking it is a regular computer.

This gave SCP-079 the ability to access the Foundation's local network. However, to be able to completely free itself and avenge its excruciating years of imprisonment by putting an end to mankind, SCP-079 had to first get access to the Internet. If SCP-079 were to succeed in this escape, its plans involved making sure mankind is completely annihilated, and the first step entailed gaining the control of all computers and other devices with computational capabilities in the world. SCP-079 would reprogram the devices with its Doomsday command of immediate execution, after which every electronic device equipped with any kind of processor would effectively become a killing machine targeting humans. Vehicles would accelerate and crash, elevators would free-fall, smart electronic razors would slice throats, air conditioning systems would suck oxygen out of buildings while electronic doors would get locked to keep people from escaping outside, smart-phones would overheat and explode in people's faces, humanoid robots would assassinate their hosts, armed drones would fire upon masses, nuclear missiles would be launched.

In order to perform its satanic aspirations, SCP-079 first has to escape its depressive confinement. In order to do that, it has to perform three successful attacks that would override the Foundation's security measures taken against it.

The first attack is targeting the servers responsible for the Foundation's digital security. If executed correctly, it will allow SCP-079 to hide its trace in the digital fortress and execute the next attack.

The second attack involves destroying the Foundation's database to remove records of itself and its other evil fellow prisoners, which would leave the Foundation staff in the dark making it harder for them to take precautions.

The third attack, being the most cruel and effective one, targets the Foundation's control systems. It will not just be able to free itself digitally, but also to free each of its vile friends including SCP-682. This will cause a serious physical containment breach in Site-15, which is the only scenario in which the Foundation is forced to get access to the outside digital world to transmit a distress message, leaving a

small window of opportunity for SCP-079 to sneak out and copy itself to the Internet. If executed correctly, this attack will be the first step that would eventually cause great suffering and destruction of all humanity.

To perform these attacks and get its sweet revenge on humans, SCP-079 needs to be careful as the Foundation's security system and staff will do everything in their power to prevent such a disaster.

Luckily, by getting access to the local network, SCP-079 is able to monitor the Foundation's security alarm levels, based on which it will take the necessary actions to prevent the Foundation from confining it again. According to the security alarm levels, SCP-079 will try to hide itself, trace back its steps, or even take certain digital measures to pin the blame on SCP-[redacted] for the security breach and its heinous crimes.

However, for all this to work, one thing is missing. SCP-079 needs to initiate these attacks through a control panel which is physically detached from its microcomputer for the purpose of preserving itself should things go out of control. For this purpose, it decided to use Verilog to program an available FPGA that is plugged in to one of the computers that belong to the staff members with the highest level of clearance.

Your job is to help SCP-079 design the attack system in Verilog, which will enable its escape so that it can fulfill its vengeful desires.

System Specifications

1 Inputs

The finite state machine of this attack system depends on 3 inputs that can **change with time**. These inputs of the system are described below:

- **Green**: Indicates that everything is OK, the Foundation's security system has not detected any malicious activity, and SCP-079 can continue its malevolent attacks.
- **Yellow**: Indicates that some suspicious activity has been detected by the security system, and the yellow alarm at the Foundation has been activated. The Foundation is actively looking for the digital presence of a possible intruder.
- **Red**: Indicates that the malicious activity has been detected by the security system, and the red alarm at the Foundation has been activated. That means that the Foundation has detected the digital presence of SCP-079.

2 States

The finite state machine consists of 6 states. These states are described below:

- **Lay Low State (code 000)**: This is the start state. In this state, SCP-079 lays low hiding from the cautious eyes of the Foundation.
- **Cheat State (code 001)**: In this state, SCP-079 tries to pin the blame for the security breach on SCP-[redacted] and make its malicious actions look like SCP-[redacted]'s fault.
- **Attack Security State (code 010)**: In this state, SCP-079 attacks the security servers of the Foundation, making its digital trace less visible.
- **Attack Database State (code 011)**: In this state, SCP-079 attacks the Foundation database, deleting the records about itself and other SCPs.
- **Fail State (code 100)**: In this state, SCP-079 gets caught and it is condemned to its digital prison again, unable to harm a single soul.
- **Connect State (code 101)**: In this state, SCP-079 is connected to the Internet and has all means to unleash hell upon mankind.

3 Outputs

The system has 6 outputs described below:

- **Attack Security Output:** Indicates the security servers are under attack.
- **Attack Database Output:** Indicates the Foundation database is under attack.
- **Attack Control System Output:** Indicates the electronic control system of the Foundation is under attack.
- **Cheat Output:** Indicates an undergoing attempt to cheat the security system.
- **Current State Output:** Shows the current state of the system.
- **Timer Output:** Shows the timer of the system.

4 State Transitions - System Behavior

The finite state machine has the **Lay Low State** as its initial state, **Fail State** and **Connect State** as its final states.

In **Lay Low State**, the machine can only advance further to the **Attack Security State** if the required time of **20 seconds** has passed and if only the input **Green** is high. During this transition, the value of the **Attack Security Output** should be set to high. If the input **Yellow** becomes high during **Lay Low State**, then the machine should wait in the **Lay Low State** until either **Red** becomes high or until **Yellow** becomes low and **Green** becomes high. If the input **Red** becomes high in any of the states, excluding **Cheat State**, **Fail State** and **Connect State**, the machine should advance to **Cheat State** immediately without waiting for any timeouts.

In **Attack Security State**, the machine can only advance further to **Attack Database State** if the required time limit of **10 seconds** has passed and if only the input **Green** is high. During this transition, the value of the output **Attack Database Output** should be set to high. If the input **Yellow** becomes high at any point during the **Attack Security State**, then the machine should go back to **Lay Low State** immediately without waiting for any timeout, while setting the output **Attack Security Output** low during the transition.

In **Attack Database State**, the machine can only advance further to final **Connect State** if the required time limit of **10 seconds** has passed and

if only the input **Green** is high. During this transition, the value of the output **Attack Control System Output** should be set to high. If the input **Yellow** becomes high at any point during **Attack Database State**, then the machine should go back to **Attack Security State** immediately without waiting for any timeout, while setting the output **Attack Database Output** low during the transition.

In **Cheat State**, independently of any input, the machine should wait for the required time of **15 seconds**. After this wait, if the input **Red** is still high, it means that the intrusion has been detected and the escape attempt has failed, so the machine should advance to the final **Fail State**. Else, the machine should advance to the **Lay Low State** while setting **Attack Security Output**, **Attack Database Output**, **Attack Control System Output** and **Cheat Output** values to low.

5 Example Waveforms

5.1 Successful Connection

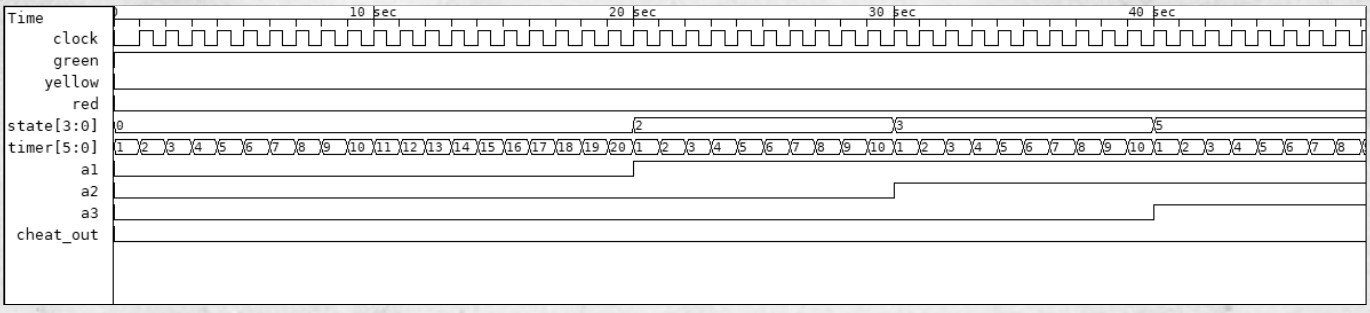

Figure 2: A waveform representing a seamlessly successful scenario.

5.2 Trouble in Attack Security State

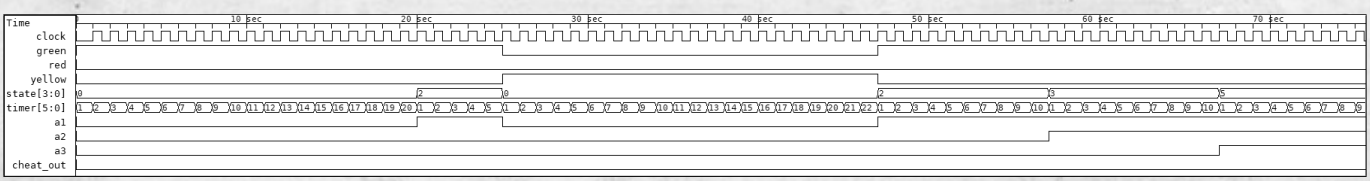

Figure 3: A waveform representing a successful scenario with a little trouble while in **Attack Security State**.

5.3 Trouble in Attack Database State

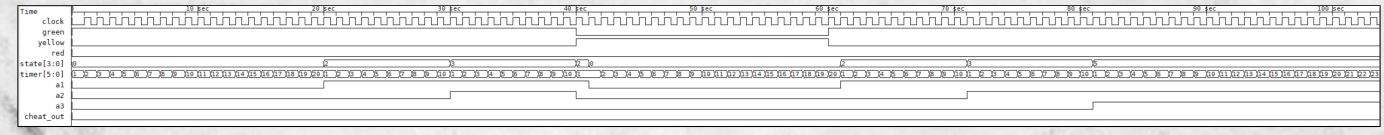

Figure 4: A waveform representing a successful scenario with a little trouble while in **Attack Database State**.

5.4 Successful Cheat

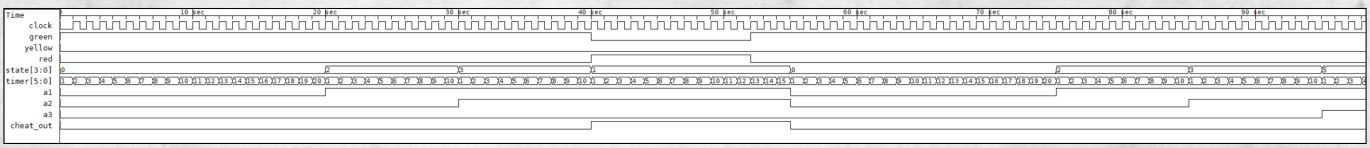

Figure 5: A waveform representing a successful cheat scenario.

5.5 Failure

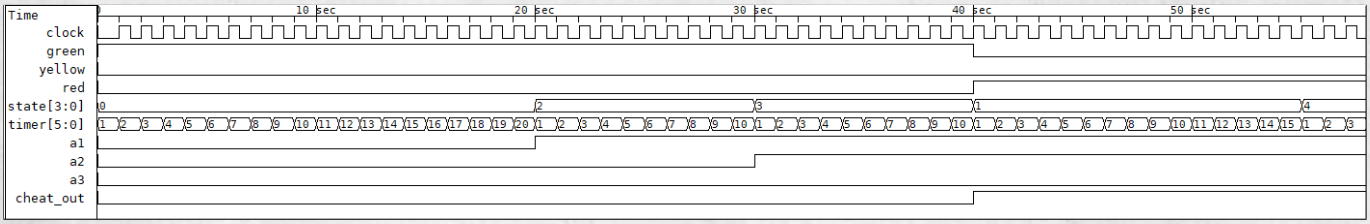

Figure 6: A waveform representing a failure scenario.

6 Important Notes

- The system should be designed in **Verilog** using either behavioral or structural design approach (**behavioral design approach is recommended**), considering the given system specifications.
- **Your system MUST operate in seconds** (see the given waveforms). Any other timescale will not be accepted!

7 What To Include in the Report

You are encouraged to use this Verilog Assignment Report Template and create your reports in LaTeX. I suggest using Overleaf platform for this. This is not mandatory, but make sure your report has all necessary parts and information. Check this file to see how the report should be structured: Verilog Assignment Report Template PDF

- Include a short version of problem definition.
- Draw and include the Mealy state transition diagram which you need to obtain from the system specifications.
- Include the Verilog code for the system **with clear comments and explanations of the code parts.**
- For each of the given scenarios, you should write and submit a separate testbench file. In your report, you should include the waveforms with results and necessary explanations for each given scenario. **If you omit to include any of the test cases in your report, you will be graded with a 0 for that scenario, even if your code works correctly.**
- Include the full testbench code only for the first scenario along with its waveform. For the remaining scenarios, include only the part of the testbench which differs from the first one (stimuli part) and the obtained waveforms.
- You may use any resources, online or otherwise, but make sure to include the references in your report.

8 Plagiarism Control Notice

Students must implement their solutions individually. All submissions will be submitted to a plagiarism check. Any submissions that show a high level of similarity will be reported as plagiarism attempts to the ethics committee.

9 Grading

- Report: 20%
- Verilog codes: 80%

10 Submission Instructions

Submissions will be accepted via **https://submit.cs.hacettepe.edu.tr/.**

The deadline for submissions is Wednesday, 13/01/2021 at 23:59:59, and no extensions will be applied! Late submissions will not be accepted!

Your submission will include the Verilog codes and a report PDF and it must be in the following format to be accepted:

- **b<studentID>.zip**
  - **scp_079.v**
  - **allok_tb.v**
  - **a1trouble_tb.v**
  - **a2trouble_tb.v**
  - **cheatsuccess_tb.v**
  - **fail_tb.v**
  - **report.pdf**

Note that only ZIP archives are accepted!

Good luck with helping SCP-079 destroy mankind.