Problems Included

Required tools - vivado, symbiyosys, verilator, tl_verilog compiler
Category 0: Sanity Check
Category 1: RTL Design
Category 2: Verification
Category 3: Debugging
Category 4: Communication

Sanity Check

- ☐ Warm-Up :
  - ☐ Provided: Bit file provided
  - ☐ Task: setup Vivado lab edition & program the board
- ☐ Switch it on :
  - ☐ Provided: Bit file & XDC file
  - ☐ Task: identify pins involved & brute force to get the flag
- ☐ U Ain't Reading That :
  - ☐ Provided: Bit file & XDC file
  - ☐ Task: identify UART connection, launch putty, brute force baud rate
- ☐ Flag City :
  - ☐ Provided: Bit file & XDC & 2 RTLs
  - ☐ Task: Identify two different inputs, flag & validate the flag

RTL Design

- ☐ Master the I2C :
  - ☐ Verilator : Provided test bench and skeleton file for I2C
  - ☐ Task : Complete the I2C FSM implementation in Verilog code to get the flag from C executable

Debugging

- ☐ The Third Eye :
  - ☐ Provided: Bit file
  - ☐ Task: Put all switches in valid position, set push button as trigger and read the flag
- ☐ ILA (Controlling the input to the board):
  - ☐ Provided: Bit file and the ltx file
  - ☐ Task: write tcl script to get the flag
- ☐ VCD file :
  - ☐ Provided: VCD file with interaction and data exchange
  - ☐ Task: load it in waveform viewer and read the transactions, write a python script to get the values

- [ ] Z3 :
    - [ ] Provided: netlist in JSON format
    - [ ] Task: To find the key which unlocks each bit at a time and then get ASCII corresponding to it

Communication

- [ ] Piano Beats :
    - [ ] Provided: Bit file generating random pattern
    - [ ] Task: The same character required to be passed a large number of times to get the flag
- [ ] is it Random :
    - [ ] Provided: Bit file generating random numbers
    - [ ] Task: Identify that its LFSR, decide the size of it, and then the polynomial

Task	Category	Points
Sanity Check	Warm-Up	10
Switch it on	Warm-Up	20
U Ain't Reading That	Warm-Up	60
Flag City	Warm-Up	50
Piano Beats	Communication	100
Is it Random	Communication	140
The Third Eye	Debugging	80
Eavesdropping is bad	Debugging	180
Hack the PCB	Debugging	160
Temple Run	Debugging	200
Master the I2C	Design	200

Other Ideas and References

FPGA - Vivado

1. Design
    a. FSM - based on a given set of asserts. Required output generated when simulated with a given tb file.
    b. Verilator - emulator
2. Testing
    a. Writing testbench for TL-Verilog module
3. STA
    a. STA problem modeled as a game - basically formulation required
4. Communication
    a. Serial communication - brute force or something more specific
    b. Identifying LFSR polynomial
5. Debugging
    a. ILA - probing and debug
    b. Waveform file given, identify the flag

TL Verilog
Symbiyosys - FIFO Verification, Wishbone counter Vivado ILA
Vivado UART
Verilator
Wireshark
Cocotb
OISC: Subleq
Sorting or any other operation followed by flag
USB brute force
Delay file
USB ILA
FIFO/Wishbone counter for verification
Some other SAT problem
TL Verilog writing testbench - or debugging
Verilator
ZipCPU
UART Simulator
Reversing from Synthesized Netlist
TCL Scripting
https://github.com/ZipCPU/wbuart32/blob/master/bench/cpp/uartsim.cpp
(http://zipcpu.com/zipcpu/2017/11/07/wb-formal.html)
https://www.xilinx.com/support/documentation/sw_manuals/xilinx2019_1/ug835-vivado-tcl-commands.pdf