**CYBS 3323**

**Test 3**

**Fall 2024**

|  |  |
| --- | --- |
| Name: | Daren Diaz |
| Student ID: | 113487126 |

**Instructions.** You have 210 minutes to complete this exam. You can access any resources of information, including calculators, electronic devices, textbooks, or notes. It would be recommended to solve the problems by yourself. In case of hand-writing, please write your answer clearly. If we cannot read your writing, it may be difficult to be graded.

**Academic Integrity Pledge.** This course operates under the rules of the Office of Academic Integrity at the University of Oklahoma. Your signature endorses the pledge below. After you finish your exam, please sign on the line below:

*I have neither given nor received aid on this examination, nor have I concealed any violations of the Academic Integrity.*

|  |
| --- |
|  |

**1. Logic Gate in the Digital Logic Circuit:**

**Please complete the circuit design, including truth table, and answer the followings:**

**(a). Logic Gates: Which of the following logic gates are called ‘universal gates’?  
(multiple choice)**

1. OR and NOR gates
2. AND and NAND gates
3. OR and AND gates
4. NOR and NAND gates



1. None of above

4 is the correct answer

**(b). Circuit Design: Using the Logic Gates Challenges (<https://www.101computing.net/logic-gates-challenges/>), create a logic circuit**

**With three inputs (A, B, and C), the light bulb should only be on when:**

**Both inputs A and B are off while input C is on  
 All three inputs should be connected to the light bulb**

A computer graphics of arrows and dots

Description automatically generated with medium confidence

**(c). Combinational Logic Circuit: Half Subtractor with Borrow-out**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Circuit | Truth Table | | | |
|  | X | Y | Difference | Borrow |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 |

**(d). Combinational Logic Circuit: 2-to-4 Binary Decoders**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Circuit | Truth Table | | | | | |
|  | A | B | Q0 | Q1 | Q2 | Q3 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |

**(e). What is the purpose of 1.(d)?**

The primary purpose of the **2-to-4 Binary Decoder** logic model is **binary-to-one-hot decoding**, which means converting a 2-bit binary input into four mutually exclusive outputs, where only one output is active at any given time.

(17 Points)

**2. Physically Unclonable Function:**

**Please answer the followings:**

**(a). In the digital circuit design, what is the Physically Unclonable Function (PUF)?**

A **Physically Unclonable Function (PUF)** is a hardware security feature that leverages the inherent physical variations in semiconductor devices to generate unique, unclonable responses. These variations arise due to manufacturing imperfections, which are unpredictable and cannot be reproduced even by the same manufacturing process.

**(b). Please complete the RTL Verilog code to create a module that generates 32-bit PUF ID using a ring oscillator in the FPGA, and explain it.**

A screenshot of a computer

Description automatically generated

In the completed RTL Verilog code, the missing initialization values for the registers (ring\_oscillator, counter, and puf\_id) were added, setting them to 32'b0 during reset to ensure proper startup conditions. The logic for the ring oscillator was implemented by introducing feedback from the XOR of the 31st and 29th bits of the ring\_oscillator register, ensuring randomness influenced by hardware variations. A counter was added to increment on every clock cycle, and the puf\_id register captures the state of the ring oscillator when the counter reaches its maximum value (32'hFFFFFFFF). These additions ensure the module generates a reproducible yet unique 32-bit PUF ID.

**(c). Draw the netlist schematic of the puf\_id module using completed Verilog code from 2.(b)?**

**You can use Xilinx Vivado tools or Web design tools (** [**http://digitaljs.tilk.eu/**](http://digitaljs.tilk.eu/) **)**

A computer screen shot of a computer

Description automatically generated

**(d). Using Rename Obfuscation, please perform the obfuscation of the codes/scripts from 2.(b).**

**A screenshot of a computer

Description automatically generated**

(20 Points)

**3. Cybersecurity Job Interview Questions:**

**Please answer the followings:**

**(a). What is a Fault Injection Attack in Hardware Security?**

A **Fault Injection Attack (FIA)** is a type of hardware attack where the adversary deliberately introduces faults or errors into a system to alter its behavior and extract sensitive information. These faults are typically induced using methods like power glitches, clock tampering, electromagnetic pulses, or laser beams. By observing the system’s incorrect behavior (e.g., skipping encryption steps or returning invalid data), attackers can gain insights into the internal workings of cryptographic algorithms or other secure processes.

**(b). What is a Side-Channel Attack in Hardware Security?**

A **Side-Channel Attack (SCA)** exploits unintentional information leakage from a hardware device during its operation. Rather than attacking the algorithm directly, it observes physical properties like power consumption, electromagnetic emissions, timing, or sound produced by the device.

**SCAs are particularly dangerous because they don’t require tampering with the hardware and rely on observing its normal behavior.**

**(c). What is Obfuscation?**

**Obfuscation** in hardware or software security refers to the process of deliberately making the design, source code, or logic of a system harder to understand to prevent reverse engineering or unauthorized use. Obfuscation transforms readable and straightforward structures into more complex forms while maintaining their functionality.

It is commonly used in software protection and hardware design to safeguard intellectual property and prevent tampering.

**(d). What is a Security Misconfiguration?**

A **Security Misconfiguration** occurs when security controls are improperly implemented or configured, leaving systems vulnerable to attacks. Common examples include:

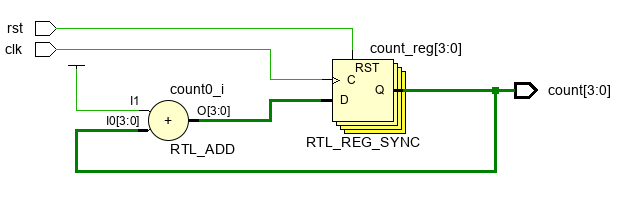
Misconfigurations are among the most common vulnerabilities and can expose systems to unauthorized access, data breaches, or malicious exploitation.

(12 Points)

**4. Chip Architecture:**

**Please answer the followings:**

**(a). Please complete the RTL Verilog code for 4-bit Synchronous Counter as shown in the schematic diagram below.**



`timescale 1ns **/** 1ps

**module** synchronouscounter**(**

**input wire clk,**

**input wire rst,**

**output** **reg** **[**3**:**0**] count**

**);**

**always@(posedge** clk**) begin**

**if(**rst**)**

count **<=** 4'b0000**; // Reset the counter to 0**

**else**

count **<=** count **+** 1**; //Increment the counter by 1**

**end**

**endmodule**

**(b). What is ‘wire’ keyword in Verilog?**

**In Verilog, the wire keyword is used to declare a net. A net is a type of variable that represents physical connections between hardware components (like wires in real circuits). Nets carry signals continuously and are typically driven by outputs of gates, modules, or continuous assignments.**

**(c). Refer the Verilog source code below, please draw the netlist schematic of the Parity Generator.**

`timescale 1ns **/** 1ps

**module** parityGenerator**(**

**output** **[**4**:**0**]** DOUT**,**

**output** parity**,**

**input** **[**3**:**0**]** DIN

**);**

**assign** parity **=** DIN**[**0**]** **^** DIN**[**1**]** **^** DIN**[**2**]** **^** DIN**[**3**];**

**assign** DOUT **=** **{** DIN**,** parity **};**

**endmodule**

You can use Xilinx Vivado tools or Web design tools ( <http://digitaljs.tilk.eu/> ) for schematic diagram.

A diagram of a party

Description automatically generated

**(d). Refer the Verilog source code from 4.(c), please simulate the tristate buffer module with the bench test module as follow.**

**module** tb\_parityGenerator**;**

**wire** **[**4**:**0**]** DOUT**;**

**wire** parity**;**

**reg** **[**3**:**0**]** DIN**;**

parityGenerator pgrtr**(**DOUT**,** parity**,** DIN**);**

**initial**

**begin**

$display**(**"RSLT\tD is parity with DOUT"**);**

DIN **=** 4'b0011**;** **#**10**;**

**if** **(** **(**parity **==** 0**)** **&&** **(**DOUT **===** **{** DIN**,** 1'b0 **})** **)**

$display**(**"PASS\t%p is %p with %p"**,** DIN**,** parity**,** DOUT**);**

**else**

$display**(**"FAIL\t%p is %p with %p"**,** DIN**,** parity**,** DOUT**);**

DIN **=** 4'b1011**;** **#**10**;**

**if** **(** **(**parity **==** 1**)** **&&** **(**DOUT **===** **{** DIN**,** 1'b1 **})** **)**

$display**(**"PASS\t%p is %p with %p"**,** DIN**,** parity**,** DOUT**);**

**else**

$display**(**"FAIL\t%p is %p with %p"**,** DIN**,** parity**,** DOUT**);**

DIN **=** 4'b1111**;** **#**10**;**

**if** **(** **(**parity **==** 0**)** **&&** **(**DOUT **===** **{** DIN**,** 1'b0 **})** **)**

$display**(**"PASS\t%p is %p with %p"**,** DIN**,** parity**,** DOUT**);**

**else**

$display**(**"FAIL\t%p is %p with %p"**,** DIN**,** parity**,** DOUT**);**

**end**

**endmodule**

<Verilog source code for a test bench of the parityGenerator module>

You can use Xilinx Vivado tools or write an expected result by hands

(20 Points)

5. Simple Codes/Scripts:

Please answer the followings:

(a). Recall your Homework #7, select six random numbers with a seed number of 8086 between 10 and 60

A screenshot of a computer program

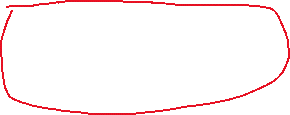
Description automatically generated



(b). Recall your Project #2, create a Lottery Simulator (a.k.a. PowerBall) using Python. Select five numbers between 1 and 69 for the white balls, then select one number between 1 and 26 for the red Powerball. Play one game for your luck.

A computer screen shot of a program

Description automatically generated



**(c). What is a Class B private IP address range in IPv4, including the subnet mask?**

Class B private IP addresses in IPv4 are reserved for use within private networks and range from **172.16.0.0 to 172.31.255.255**, as defined by RFC 1918. These addresses are not routable on the public internet and are typically used in internal networks for medium to large organizations. The default subnet mask for a Class B network is **255.255.0.0** (or /16 in CIDR notation), allowing up to **65,534 usable IP addresses per network**. Devices within this range communicate internally, and access to the internet requires **Network Address Translation (NAT)** to map private addresses to public ones. This ensures secure and efficient use of IP addresses in private environments.

**(d). Recall your homework #6, using a Python client script, connect to the TCP/IP server 216.249.138.192 with a port number 2024, show the message you received.**

Screens screenshot of a computer screen

Description automatically generated

**(e). Using a PyArmor, generate the obfuscated code of the Python client script in P5.(d).**

Example obfuscated code for client script (After running PyArmor, the obfuscated file will look something like this):

**from pytransform import pyarmor\_runtime**

**pyarmor\_runtime()**

**def obfuscated\_function():**

**# This is an obfuscated version of your function**

**pass**

(33 Points)

**6. Cryptography:**

**Please answer the followings:**

**(a). The message that you received from 5.(d), decrypt the message with Cisco type 7 password cracker. ( https://www.ifm.net.nz/cookbooks/passwordcracker.html )**

**(b) Recall the Project #0: download the T3p6b\_encrypted\_music\_mp3.enc file and decrypt it with a scaled-down DES algorithm using key (0x586) to listen. What is the name of the song and musician?**

**(c). From Problem 6.(b), show the MD5 checksum of each file (encrypted file/decrypted file).**

(18 Points)