

IoT Security Project

SEAN CASTER, VINCENZO PISCITELLO, RYAN
HOWERTON, TERRI HEWITT, AND ADAM BARTON

SENIOR DESIGN

FALL 2018

Abstract

Write an abstract here. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

April 2, 2019

Contents

- 1 Introduction** **2**
 - 1.1 Subsection 2
 - 1.1.1 Subsubsection 2
- 2 Appendix** **3**
 - 2.1 Design Synthesis and Analysis 3
 - 2.2 Source Code 3
 - 2.2.1 top_module 3
 - 2.3 Subsection 4

1 Introduction

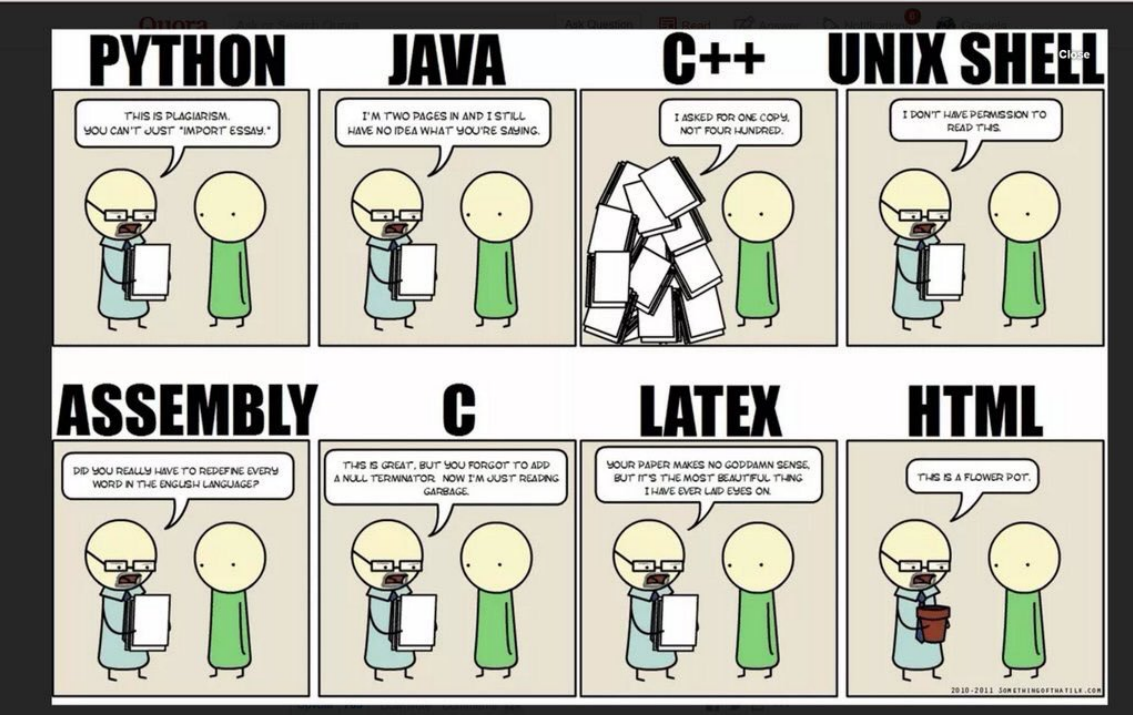


Figure 1: Insert caption here. © Google, 2017.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum [1].

1.1 Subsection

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur [2]. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Example Table:

1	2	3	4	5	6
7	8	9	10		
11	12	13	14		

1.1.1 Subsubsection

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2 Appendix

2.1 Design Synthesis and Analysis

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2.2 Source Code

2.2.1 top_module

```
module top_module(input logic N64_data,
                  input logic remote_data,
                  input logic [7:0] buttons,
                  input logic nes_clk, snes_clk, n64_clk,
                  input logic NES_latch, SNES_latch,
                  input logic nes_reset, snes_reset,
                  input logic remote_reset,
                  input logic reset_n, //reset for clock_counter_1MHz
                  input logic [1:0] controller_select,
                  input logic console_select,
                  output logic nes_output[7:0] //nes controller output data line
                  output logic snes_output[11:0]
);

    logic nes_data_bus;
    logic clk_bus;
    logic clk_4M;
    logic clk_1M;
    logic remote_selection; //12-bit data line from controller MUX to console
    DEMUX
    logic [7:0] N64_state_bus;

    logic [11:0] to_NES_12;
    logic [7:0] to_NES_8;
    logic [11:0] to_SNES;
    logic [11:0] to_active_controller;

    assign to_NES_8 = to_NES_12[7:0];

    //TODO:
    //4 modules

    //Instance of built in module to select 4.29MHz clock
    OSC #( "4.29" ) osc_int (
        .STDBY(1'b0), //Specifies active state
        .OSC(clk_4M), //Output 4.29MHz to clk_4M net
        .SEDSTDBY()); //Leaves SEDSTDBY pin unconnected

    //Instantiation of 1MHz clock divider (4MHz input, 1MHz output)
    clock_counter_1MHz clk_divider(
        .clk_4MHz(clk_4M),
        .reset_n(reset_n),
        .clk_1MHz(clk_1M));

    //Active Controller selector (MUX)
    active_controller(
        .remote_data(to_active_controller), //8-bit raw remote data input
```

```

        .N64_data(N64_state_bus), //8-bit raw N64 data input
        .button_data(buttons), //8-bit raw button input
        .select_bits(controller_select), //2-bit data input
        .selection(remote_selection) //Output 12-bit selection
    );

    //Active Console selector (DEMUX)
    active_console_console_select(
        .controller_selection(remote_selection),
        .select_bit(console_select), //Bit to select console
        .NES_console(to_NES_12), //12-bit data output
        .SNES_console(to_SNES) //12-bit data output
    );

    shift_register_PISO_8bit NES(
        .clk(nes_clk),
        .reset(nes_reset),
        .load(NES_latch),
        .data(to_NES_8),
        .sout(nes_output)
    );

    shift_register_PISO_12bit SNES(
        .clk(snes_clk),
        .reset(snes_reset),
        .load(SNES_latch),
        .data(to_SNES),
        .sout(snes_output)
    );

    //N64 Controller Reader
    n64_button_reader N64_C(
        .clk_4M(clk_4M),
        .clk_1M(clk_1M),
        .n64_data(N64_data),
        .n64_button_state(N64_state_bus),
        .n64_controller_clk(n64_clk)
    );

    //Remote Reader

    remote_translator IR_translator(
        .serial_from_IR(remote_data),
        .OneMHz_clock(clk_1M),
        .reset(remote_reset),
        .to_NES_shift_register(to_active_controller)
    );
endmodule

```

./source/top_module.sv

2.3 Subsection

Source code listing

```

vsim .work.n64_send_cmd
add wave *
force clk_1M 0 0, 1 {1ns} -r 2 ns
force output_enable 1 @ 0
force read_start 0 @ 0
run 100 ns

```

References

- [1] J. Doe and J. Doe, “Title.” <http://www.google.com>, 2013.
- [2] Author, “Title2.” <http://www.yahoo.com>, 2012.