Perancangan Sistem Digital

Group PA 15

Implementation of DHCP and VLAN

Presented By:

Azka Nabihan Hilmy	2306250541
Dwigina Sitti Zahwa	2306250724
Bonifasius Raditya Pandu	2306242350
Muhammad Riyan Satrio	2306229323

Code Diagram

TB.vhdl

- Modul 4
- Modul 7

Top_Level.vhdl

- Modul 3
- Modul 5
- Modul 7

Simulation

- ModelSIM

FSM_DHCP.vhdl

- Modul 3
- Modul 6
- Modul 8

DHCP_Output.txt

VLAN.vhdl

- Modul 3
- Modul 7

Objectives

- Mengimplementasikan DHCP dan VLAN dalam Desain Digital dengan VHDL Language
- Meningkatkan efisiensi dan efektivitas dalam manajemen jaringan
- DHCP untuk otomatisasi Pengelolaan Konfigurasi dan Manajemen VLAN untuk segmentasi dalm desain digital
- Pemenuhan Praktikum Perancangan Sistem Digital

Perancangan Sistem Digital

Project Description

Mengimplementasikan DHCP untuk **mengalokasikan IP, Subnet mask, dan gateway** secara dinamis dengan segmentasi menggunakan **VLAN ID** dalam desain *digital*. Implementasi ini menggunakan **Top Level** (Top_Level) sebagai **modul utama** untuk meminta dan memberi konfigurasi kepada client dengan menghubungkan **FSM DHCP** (fsm_dhcp) dengan segementasi **VLAN** (vlan). Uji verifikasi DHCP dan VLAN menggunakan **testbench** (tb) dengan 2 testcase, yaitu **5 VLAN dan 10 VLAN** disertai Report dengan **Severity Note** untuk memperlihat informasi konfigurasi

Perancangan Sistem Digital

Finite State Machine ↘

Memberikan **IP Address** kepada *device client* berdasarkan input **network, subnet mask, dan jumlah perangkat**. Server DHCP akan menghitung, menetapkan, dan mencatat **IP Address, Subnet mask, dan gateway** yang telah dialokasikan. Proses FSM dimulai dari **idle, discover, offer, request, acknowledge, dan final** , sampai *device* mendapat IP Address.

• Entity

```
entity DHCP_Server is
    Port ( clk         : in  STD_LOGIC;
           reset       : in  STD_LOGIC;
           network     : in  STD_LOGIC_VECTOR(31 downto 0);
           subnet_mask : in  STD_LOGIC_VECTOR(31 downto 0);
           num_devices : in  integer;
           mulai       : in  STD_LOGIC;
           ip_address  : out STD_LOGIC_VECTOR(31 downto 0);
           gateway     : out STD_LOGIC_VECTOR(31 downto 0);
           subnet_out  : out STD_LOGIC_VECTOR(31 downto 0);
           selesai     : out STD_LOGIC
           );
end DHCP_Server;
```

• Looping Contruct

```
-- Procedure to write to output.txt
procedure write_ip_to_file(
    variable ip_to_write : in std_logic_vector(31 downto 0)
) is
    variable write_buffer : line;
    variable ip_str       : string(1 to 32);  -- Buffer to store IP as string
begin
    -- Convert std_logic_vector to string
    for i in 1 to 32 loop
        if ip_to_write(32 - i) = '1' then
            ip_str(i) := '1';
        else
            ip_str(i) := '0';
        end if;
    end loop;

    -- Write to file
    write(write_buffer, ip_str);
    writeline(output_file, write_buffer);
end procedure;
```

• FSM

```
begin
    case current_state is
        when idle =>
            ip_address <= (others => '0');
            gateway <= (others => '0');
            subnet_out <= (others => '0');
            if mulai = '1' and count < num_devices then
                next_state <= discover;
                count := count + 1;
                selesai <= '0';
            else
                next_state <= idle;
                selesai <= '1';
            end if;
```

Perancangan Sistem Digital

FSM_DHCP.VHDL

- **Entity**

```
entity DHCP_Server is
    Port ( clk          : in  STD_LOGIC;
           reset        : in  STD_LOGIC;
           network      : in  STD_LOGIC_VECTOR(31 downto 0);
           subnet_mask  : in  STD_LOGIC_VECTOR(31 downto 0);
           num_devices  : in  integer;
           mulai        : in  STD_LOGIC;
           ip_address   : out STD_LOGIC_VECTOR(31 downto 0);
           gateway      : out STD_LOGIC_VECTOR(31 downto 0);
           subnet_out   : out STD_LOGIC_VECTOR(31 downto 0);
           selesai      : out STD_LOGIC
           );
end DHCP_Server;
```

→ INPUT

→ OUTPUT

- **Looping Construct**

```
-- Procedure to write to output.txt
procedure write_ip_to_file(
    variable ip_to_write : in std_logic_vector(31 downto 0)
) is
    variable write_buffer : line;
    variable ip_str       : string(1 to 32);  -- Buffer to store IP as stri
begin
    -- Convert std_logic_vector to string
    for i in 1 to 32 loop
        if ip_to_write(32 - i) = '1' then
            ip_str(i) := '1';
        else
            ip_str(i) := '0';
        end if;
    end loop;

    -- Write to file
    write(write_buffer, ip_str);
    writeline(output_file, write_buffer);
end procedure;
```

→ Convert std_logic_vector - String

→ Menuliskan IP Address pada textio

- **FSM**

```
begin
    case current_state is
        when idle =>
            ip_address <= (others => '0');
            gateway <= (others => '0');
            subnet_out <= (others => '0');
            if mulai = '1' and count < num_devices then
                next_state <= discover;
                count := count + 1;
                selesai <= '0';
            else
                next_state <= idle;
                selesai <= '1';
            end if;

        when discover =>
            -- Set gateway and subnet mask
            set_gateway := std_logic_vector(unsigned(subnet_mask(7 downto 0)) + 1);
            gateway <= network(31 downto 8) & set_gateway;
            if count = 1 then last_ip <= network(31 downto 8) & set_gateway;
            end if;

            next_state <= offer;

        when offer =>
            -- Calculate next IP
            set_ip := std_logic_vector(unsigned(last_ip(7 downto 0)) + 1);
            allocated_ip <= network(31 downto 8) & set_ip;

            subnet_out <= subnet_mask;
            next_state <= request;

        when request =>
            ip_address <= allocated_ip;
            last_ip <= allocated_ip;
            next_state <= ack;

        when ack =>
            next_state <= finish;

        when finish =>
            -- Write the allocated IP to the output file
            temp_ip := allocated_ip;  -- Use temporary variable
            write_ip_to_file(temp_ip);  -- Call procedure to write to file
            next_state <= idle;

    end case;
```

→ Tahapan IDLE untuk memulai Proses DHCP

→ Discover menetapkan Subnet mask dan gateway

→ Offer menghitung IP Address yang akan dialokasikan

→ Client menyetuji konfigurasi yang diberi

→ Menuliskan output dan FSM selesai

Perancangan Sistem Digital

VLAN ↘

Mengalokasi dan Memberikan VLAN ID dalam rentang tertentu berdasarkan *input* total VLAN(maks 10) dan secara bertahap menghitung/ menetapkan VLAN ID ke array internal dan *output* VLAN. VLAN berjalan dalam tahapan inisialisasi, alokasi VLAN, dan selesai (done) sebagai tanda berakhir.

- **Entity**

```
entity VLAN_Allocation is
    Port ( clk           : in  STD_LOGIC;
           reset         : in  STD_LOGIC;
           total_vlans   : in  integer range 1 to 10;  -- In
           vlan          : out STD_LOGIC_VECTOR(9 downto 0);
           done          : out std_logic
           );                 -- Process completion flag
end VLAN_Allocation;
```

- **Allocation Behavioral**

```
architecture Behavioral of VLAN_Allocation is

    -- Signal declarations
    type integer_array is array (0 to 9) of integer;  -- Define the integer arr
    signal vlan_id_array_sig : integer_array := (others => 0);  -- Signal to st
    signal vlan_counter      : integer := 0;
    signal state             : integer := 0;
    signal all_vlans         : std_logic_vector(9 downto 0) := (others => '0');
```

VLAN.VHDL

- **Entity**

```
entity VLAN_Allocation is
    Port ( clk           : in  STD_LOGIC;
           reset         : in  STD_LOGIC;
           total_vlans   : in  integer range 1 to 10;  -- In
           vlan          : out STD_LOGIC_VECTOR(9 downto 0);
           done          : out std_logic
           );              -- Process completion flag
end VLAN_Allocation;
```

→ Memasukkan banyaknya VLAN (maks 10)

→ Menampilkan VLAN yang digunakan dan inisialisasi selesai (done)

- **Architecture**

```
architecture Behavioral of VLAN_Allocation is

    -- Signal declarations
    type integer_array is array (0 to 9) of integer;  -- Define the integer array type for internal signals
    signal vlan_id_array_sig : integer_array := (others => 0);  -- Signal to store VLAN IDs
    signal vlan_counter      : integer := 0;
    signal state             : integer := 0;
    signal all_vlans         : std_logic_vector(9 downto 0) := (others => '0');
```

→ Mendeklarasika arsitektur

→ Mendeklarasikan signal yang digunakan untuk keperluan program

Perancangan Sistem Digital

VLAN.VHDL

```
    -- Process to handle VLAN allocation
    process(clk, reset)
    begin
        if reset = '1' then
            state <= 0;
            vlan_counter <= 0;
            vlan_id_array_sig <= (others => 0);
            all_vlans <= (others => '0');
        elsif rising_edge(clk) then
            case state is
                when 0 =>  -- Initial state, start the process
                    if total_vlans > 0 then
                        done <= '0';
                        all_vlans <= (others => '0');
                        state <= 1;  -- Proceed to VLAN allocation
                    end if;

                when 1 =>  -- VLAN allocation
                    for i in 0 to 9 loop
                        -- Calculate VLAN ID (multiples of 10 starting from 10)
                        if i < total_vlans then
                            vlan_id_array_sig(i) <= (i + 1) * 10;
                            -- Update all_vlans for output
                            all_vlans(i) <= '1';  -- Example of setting an active VLAN
                        end if;
                    end loop;
                    -- Update output signal and move to the next state
                    vlan <= all_vlans;

                    vlan_counter <= vlan_counter + 1;

                    if vlan_counter = 3 then
                        state <= 2;
                    end if;

                when 2 =>  -- Indicate completion
                    done <= '1';
                    vlan_counter <= 0;
                    state <= 0;  -- Go back to initial state

                when others =>  -- Default case
                    state <= 0;

            end case;
        end if;
    end process;
end Behavioral;
```

→ Kondisi untuk Mereset

→ Inisialisasi dan memulai proses (FMS) untuk Alokasi VLAN

→ Alokasi VLAN dan menghitung VLAN ID (kelipatan 10)

→ Proses selesai dan kembali ke initial

→ Saat case dalam default

Perancangan Sistem Digital

TOP LEVEL

Modul utama yang menghubungkan FSM (fsm_dhcp) pada DHCP Server dengan VLAN Manager (vlan) untuk mengalokasikan IP Address secara otomatis ke setiap VLAN aktif. Client akan meng-*input* banyaknya VLAN (maks 10), mengaktifkan VLAN, selain itu akan menampilkan array IP Address yang dialokasikan, status VLAN, dan proses selesai (done)

- **Entity**

```
entity Top_Level is
    Port (
        clk           : in  STD_LOGIC;
        reset         : in  STD_LOGIC;
        enable_dhcp   : in  STD_LOGIC;
        total_vlans   : in  integer range 1 to 10;  -- Inp
        ip_addresses  : out ip_array;  -- Output IP addres
        status        : out STD_LOGIC_VECTOR(9 downto 0);
        selesai       : out STD_LOGIC_VECTOR(9 downto 0)
    );
end Top_Level;
```

- **Structural Top Level**

```
architecture Structural of Top_Level is
    -- Components
    component DHCP_Server is
    Port ( clk          : in  STD_LOGIC;
           reset        : in  STD_LOGIC;
           network      : in  STD_LOGIC_VECTOR(31 downto 0);
           subnet_mask  : in  STD_LOGIC_VECTOR(31 downto 0);
           num_devices  : in  integer;
           mulai        : in  STD_LOGIC;
           ip_address   : out STD_LOGIC_VECTOR(31 downto 0);
           gateway      : out STD_LOGIC_VECTOR(31 downto 0);
           subnet_out   : out STD_LOGIC_VECTOR(31 downto 0);
           selesai      : out STD_LOGIC
           );
    end component;
```

Perancangan Sistem Digital

TOP_LEVEL.VHD

- **Entity**

```
entity Top_Level is
    Port (
        clk           : in  STD_LOGIC;
        reset         : in  STD_LOGIC;
        enable_dhcp   : in  STD_LOGIC;
        total_vlans   : in  integer range 1 to 10;  -- Inp
        ip_addresses  : out ip_array;  -- Output IP addres
        status        : out STD_LOGIC_VECTOR(9 downto 0);
        selesai       : out STD_LOGIC_VECTOR(9 downto 0)
    );
end Top_Level;
```

Mengaktifkan dan memasukan total VLAN

Menampilkan IP, status, dan indikasi selesai

- **Top Level Structural**

```
architecture Structural of Top_Level is
    -- Components
    component DHCP_Server is
    Port ( clk          : in  STD_LOGIC;
           reset        : in  STD_LOGIC;
           network      : in  STD_LOGIC_VECTOR(31 downto 0);
           subnet_mask  : in  STD_LOGIC_VECTOR(31 downto 0);
           num_devices  : in  integer;
           mulai        : in  STD_LOGIC;
           ip_address   : out STD_LOGIC_VECTOR(31 downto 0);
           gateway      : out STD_LOGIC_VECTOR(31 downto 0);
           subnet_out   : out STD_LOGIC_VECTOR(31 downto 0);
           selesai      : out STD_LOGIC
           );
    end component;
```

Menggunakan DHCP Server untuk alokasi ke setiap VLAN yang aktif

- **Component VLAN Allocation**

```
component VLAN_Allocation is
Port ( clk           : in  STD_LOGIC;
       reset         : in  STD_LOGIC;
       total_vlans   : in  integer range 1 to 10;  -- Inp
       vlan          : out STD_LOGIC_VECTOR(9 downto 0);
       done          : out std_logic
       );                  -- Process completion flag
end component;
```

Menghubungkan VLAN Manager

```
-- Internal signals
signal valid_vlan    : std_logic;
signal vlan_id_internal : std_logic_vector(9 downto 0);
signal ip_allocated  : ip_array;
signal ip : std_logic_vector(31 downto 0);
signal selesai_signal : std_logic_vector(9 downto 0);
signal dhcp_enable   : std_logic_vector(9 downto 0);

constant network_values : ip_array := (
    "11000000101010000000000100000000",  -- 192.168.1.0
    "11000000101010000000001000000000",  -- 192.168.2.0
    "11000000101010000000001100000000",  -- 192.168.3.0
    "11000000101010000000010000000000",  -- 192.168.4.0
    "11000000101010000000010100000000",  -- 192.168.5.0
    "11000000101010000000011000000000",  -- 192.168.6.0
    "11000000101010000000011100000000",  -- 192.168.7.0
    "11000000101010000000100000000000",  -- 192.168.8.0
    "11000000101010000000100100000000",  -- 192.168.9.0
    "11000000101010000000101000000000"   -- 192.168.10.0
);
```

Nilai constant network untuk setiap VLAN ID

TOP_LEVEL.VHD

- **DHCP Server**

Meng-generate DHCP Server berdasrkan VLAN (maks 10)

```
-- Generate DHCP Servers based on VLANs
gen_dhcp: for i in 0 to 9 generate
    dhcp_inst: if i < 10 generate
        dhcp_inst: DHCP_Server
            port map (
                clk           => clk,
                reset         => reset,
                network       => network_values(i),  -- 192.168.i.0
                subnet_mask   => "11111111111111111111111100000000",
                num_devices   => 10,
                mulai         => std_logic(dhcp_enable(i)),
                ip_address    => ip_allocated(i),
                gateway       => open,
                subnet_out    => open,
                selesai       => selesai_signal(i)
            );
    end generate;
end generate;
```

Komponen DHCP untuk network 192.168.X.0/24

Proses untuk menangani rst dan clk

```
-- Process to handle reset and clock logic
process(clk, reset)
begin
    if rising_edge(clk) then
        for i in 0 to 9 loop
            if i < total_vlans then
                -- Enable DHCP for active VLANs
                dhcp_enable(i) <= enable_dhcp;
            else
                -- Disable DHCP for inactive VLANs
                dhcp_enable(i) <= '0';
            end if;
        end loop;
    end if;
end process;
```

- **Output**

```
    -- Connect outputs
    ip_addresses <= ip_allocated;
    status <= vlan_id_internal;
    selesai <= selesai_signal;

end Structural;
```

Output untuk Top Level

Perancangan Sistem Digital

TESTBENCH

Modul yang digunakan untuk menguji modul Top_Level. Testbench menggunakan stimulus untuk mensimulasikan permintaan IP dan pemrosesan VLAN dengan memastikan FSM telah berjalan sesuai tahapan DHCP. Stimulus yang diberikan berupa jumlah VLAN dan DHCP Enable, dimana terdapat test case 3, 5, 7, dan 10 VLAN

- **Entity**

```
entity Top_Level_tb is
-- Testbench has no ports
end Top_Level_tb;
```

- **Testbench Behavioral**

```
architecture Behavioral of Top_Level_tb is=
    -- Component declaration for the Unit Under Test (UUT)
    component Top_Level
        Port (
            clk           : in  STD_LOGIC;
            reset         : in  STD_LOGIC;
            enable_dhcp   : in  STD_LOGIC;
            total_vlans   : in  integer range 1 to 10;
            ip_addresses  : out ip_array;
            status        : out STD_LOGIC_VECTOR(9 downto 0);
            selesai       : out STD_LOGIC_VECTOR(9 downto 0)
        );
    end component;

    -- Signals for driving inputs and observing outputs
    signal clk           : std_logic := '0';
    signal reset         : std_logic := '0';
    signal enable_dhcp   : std_logic := '0';
    signal total_vlans   : integer := 1;
    signal ip_addresses  : ip_array;
    signal status        : std_logic_vector(9 downto 0);
    signal selesai       : std_logic_vector(9 downto 0);

    -- Clock period definition
    constant clk_period : time := 1 ns;
```

Perancangan Sistem Digital

TB.VHDL

- **Function to_binary_string**

```vhdl
-- Function to convert std_logic_vector to string format
function to_binary_string(input : std_logic_vector(9 downto 0)) return string is
    variable result : string(1 to 10);
begin
    for i in 0 to 9 loop
        if input(i) = '1' then
            result(10-i) := '1';
        else
            result(10-i) := '0';
        end if;
    end loop;
    return result;
end function;
```

Mengubah 10 bit std_logic_vector ke dalam bentuk string. Digunakan dalam report statement

- **Clock Process**

```vhdl
-- Clock generation process
clk_process : process
begin
    while true loop
        clk <= '0';
        wait for clk_period / 2;
        clk <= '1';
        wait for clk_period / 2;
    end loop;
    wait;
end process;
```

Menghasilkan input clock untuk stimulus UUT

- **Stimulus Process**

```vhdl
-- Stimulus process
stim_proc: process
begin
    -- Reset the system
    reset <= '1';
    wait for 2 ns;
    reset <= '0';

    -- Test with 5 VLANs
    total_vlans <= 5;
    enable_dhcp <= '1';
    wait until selesai = "1111111111";
    report "Test with 5 VLANs completed. Current status: " & to_binary_string(status) &
        ", Completion: " & to_binary_string(selesai);
    for i in 0 to 4 loop  -- Adjust range to match 'total_vlans'
        report "VLAN ID " & integer'image(i) & ": IP allocated: " &
            integer'image(to_integer(unsigned(ip_addresses(i)(31 downto 24)))) & "." &
            integer'image(to_integer(unsigned(ip_addresses(i)(23 downto 16)))) & "." &
            integer'image(to_integer(unsigned(ip_addresses(i)(15 downto 8)))) & "." &
            integer'image(to_integer(unsigned(ip_addresses(i)(7 downto 0))));
    end loop;
```

Reset sistem di awal

Menunggu seluruh DHCP server selesai dijalankan

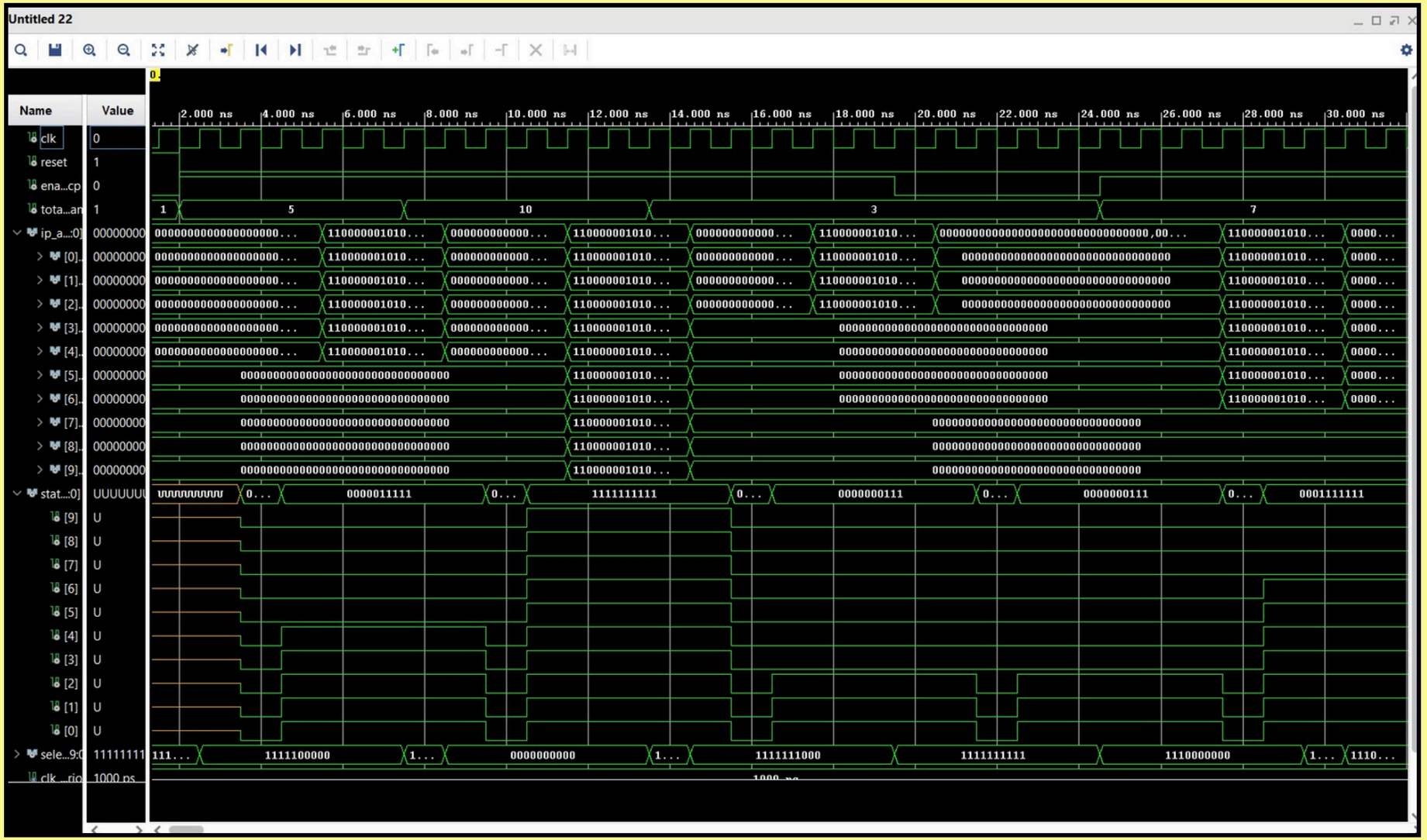
Report IP address yang dialokasikan dalam format IP address (x.x.x.x)

```vhdl
    enable_dhcp <= '0';
    wait for 5ns;
    report "DHCP disabled. Current status: " & to_binary_string(status) &
        ", Completion: " & to_binary_string(selesai);
```

Test case ketika enable_dhcp diset ‘0’ (DHCP dinonaktifkan)

Perancangan Sistem Digital

Testing



Result

- 3 VLAN

```
Note: Test with 3 VLANs completed. Current status: 0000000111, Completion: 1111111111
Time: 19500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc  File: C:/RIyan/Tugas/Perancangan
Note: VLAN ID 0: IP allocated: 192.168.1.4
Time: 19500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc  File: C:/RIyan/Tugas/Perancangan
Note: VLAN ID 1: IP allocated: 192.168.2.4
Time: 19500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc  File: C:/RIyan/Tugas/Perancangan
Note: VLAN ID 2: IP allocated: 192.168.3.4
Time: 19500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc  File: C:/RIyan/Tugas/Perancangan
Note: DHCP disabled. Current status: 0000000111, Completion: 1111111111
Time: 24500 ps  Iteration: 0  Process: /Top_Level_tb/stim_proc  File: C:/RIyan/Tugas/Perancangan
```

- 5 VLAN

```
Note: Test with 5 VLANs completed. Current status: 0000011111, Completion: 1111111111
Time: 7500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc  File: C:/RIyan/Tugas/Perancangan
Note: VLAN ID 0: IP allocated: 192.168.1.2
Time: 7500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc  File: C:/RIyan/Tugas/Perancangan
Note: VLAN ID 1: IP allocated: 192.168.2.2
Time: 7500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc  File: C:/RIyan/Tugas/Perancangan
Note: VLAN ID 2: IP allocated: 192.168.3.2
Time: 7500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc  File: C:/RIyan/Tugas/Perancangan
Note: VLAN ID 3: IP allocated: 192.168.4.2
Time: 7500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc  File: C:/RIyan/Tugas/Perancangan
Note: VLAN ID 4: IP allocated: 192.168.5.2
Time: 7500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc  File: C:/RIyan/Tugas/Perancangan
```

- 7 VLAN

```
Note: Test with 7 VLANs completed. Current status: 0001111111, Completion: 1111111111
Time: 29500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc  File: C:/RIyan/Tugas/Perancangan
Note: VLAN ID 0: IP allocated: 192.168.1.5
Time: 29500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc  File: C:/RIyan/Tugas/Perancangan
Note: VLAN ID 1: IP allocated: 192.168.2.5
Time: 29500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc  File: C:/RIyan/Tugas/Perancangan
Note: VLAN ID 2: IP allocated: 192.168.3.5
Time: 29500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc  File: C:/RIyan/Tugas/Perancangan
Note: VLAN ID 3: IP allocated: 192.168.4.4
Time: 29500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc  File: C:/RIyan/Tugas/Perancangan
Note: VLAN ID 4: IP allocated: 192.168.5.4
Time: 29500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc  File: C:/RIyan/Tugas/Perancangan
Note: VLAN ID 5: IP allocated: 192.168.6.3
Time: 29500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc  File: C:/RIyan/Tugas/Perancangan
Note: VLAN ID 6: IP allocated: 192.168.7.3
Time: 29500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc  File: C:/RIyan/Tugas/Perancangan
```

- 10 VLAN

```
Note: Test with 10 VLANs completed. Current status: 1111111111,
Time: 13500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc
Note: VLAN ID 0: IP allocated: 192.168.1.3
Time: 13500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc
Note: VLAN ID 1: IP allocated: 192.168.2.3
Time: 13500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc
Note: VLAN ID 2: IP allocated: 192.168.3.3
Time: 13500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc
Note: VLAN ID 3: IP allocated: 192.168.4.3
Time: 13500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc
Note: VLAN ID 4: IP allocated: 192.168.5.3
Time: 13500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc
Note: VLAN ID 5: IP allocated: 192.168.6.2
Time: 13500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc
Note: VLAN ID 6: IP allocated: 192.168.7.2
Time: 13500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc
Note: VLAN ID 7: IP allocated: 192.168.8.2
Time: 13500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc
Note: VLAN ID 8: IP allocated: 192.168.9.2
Time: 13500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc
Note: VLAN ID 9: IP allocated: 192.168.10.2
Time: 13500 ps  Iteration: 4  Process: /Top_Level_tb/stim_proc
```

Perancangan Sistem Digital

Thank You!

Group PA15