



Nama Edgrant Henderson Suryajaya

NPM 2206025016

Kode Asisten

Jenis Tugas TP

A. Teori

1. Finite state mesin (FSM) adalah sebuah sistem abstrak yang responsnya tergantung pada state saat ini dan masukan dari luar. FSM dapat digunakan untuk merepresentasikan perilaku sistem sekuensial yang berubah-ubah sesuai dengan kondisi tertentu. Contoh FSM dalam kehidupan sehari-hari adalah mesin penjual otomatis, lampu lalu lintas, alarm jam, dll.

Tipe state machine ada 2. Pertama adalah Moore State Machine yang outputnya hanya ditentukan oleh keadaan sekarang, tidak dipengaruhi oleh input. Kedua adalah Mealy State Machine yang keadaan outputnya ditentukan oleh keadaan sekarang dan input.

State diagram adalah representasi dari rangkaian sekuensial secara diagram. Terdapat current state, next state, input dan output dalam sebuah state diagram. Perbedaan state diagram untuk state machine mealy dan moore adalah pada mealy, outputnya pada tanda panah sebelah input, sedangkan pada state machine moore, output pada statenya sendiri.

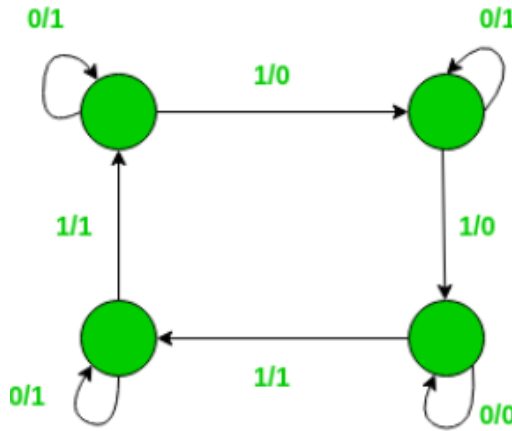


Figure - Mealy machine

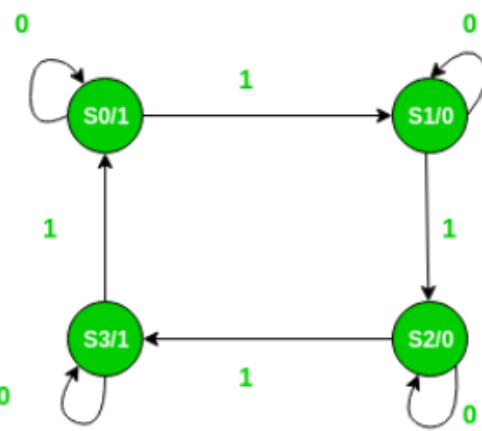


Figure - Moore machine

Referensi:

FTUI, Digital Laboratory “Dasar Teori Modul 7: FINITE STATE MACHINE”, 2023.

[online]. [Accessed: 12-NOV-2023].

GeeksforGeeks, “Difference between Mealy machine and Moore machine,”

GeeksforGeeks, May 09, 2023. Available:

<https://www.geeksforgeeks.org/difference-between-mealy-machine-and-moore-machine/>

Chang and Zhu David, 2.3 Finite State Machine (FSM) Concept and Implementation.

Stanford University, 2015. Available:

https://web.stanford.edu/class/cs123/lectures/CS123_lec07_Finite_State_Machine.pdf

2. Terdapat jenis data komposit yang umum digunakan dalam VHDL, yaitu array dan record. Array adalah jenis data yang terdiri dari kumpulan elemen dengan jenis yang sama. Ukuran dari dapat ditentukan atau tidak ditentukan saat mendeklarasikan tipe array. Sintaks sebagai berikut

```
-- Mendeklarasikan tipe array dengan ukuran yang ditentukan
type <type_name> is array (<range>) of <type>;[Size1][Size2]

-- Mendeklarasikan tipe array dengan ukuran yang tidak ditentukan
type <type_name> is array (natural <range>) of <type>;
type <type_name> is array (positive <range>) of <type>;
```

Record adalah jenis data yang terdiri dari kumpulan elemen dengan jenis yang berbeda-beda. Record dapat digunakan untuk mengelompokkan sinyal yang berkaitan atau menyederhanakan daftar port dalam. Sintaks sebagai berikut

```
-- Mendeklarasikan tipe record
type <record_name> is record
    -- Mendeklarasikan elemen-elemen yang membentuk record
    <element_name> : <type>;
    <element_name> : <type>;
end record <record_name>;
```

Enum adalah sebuah tipe data yang menyimpan list value. Enum bisa digunakan untuk melabel beberapa value agar mempermudah membaca kode. Syntaxnya sebagai berikut.

```
type T_STATE is
    (INIT, RED, REDYELLOW, GREEN, YELLOW);
```



Referensi:

John, "VHDL Record, Array and Custom Types - FPGA tutorial," FPGA Tutorial, Nov. 16, 2021. Available: <https://fpgatutorial.com/vhdl-records-arrays-and-custom-types/>

"courses:system_design:vhdl_language_and_syntax:extended_data_types:enumeration_types [VHDL-Online]." Available: https://www.vhdl-online.de/courses/system_design/vhdl_language_and_syntax/extended_data_types/enumeration_types



B. Latihan

a. Ubahlah state diagram berikut menjadi kode VHDL

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity tp8 is
    port (
        clk, inn : in std_logic;
        outt : out std_logic
    );
end entity tp8;

architecture rtl of tp8 is
    type state is (s0, s1, s2);
    signal currentState, nextState : state := s0;
begin
    clock_upadte: process(clk)
    begin
        if rising_edge(clk) then
            currentState <= nextState;
        end if;
    end process clock_upadte;

    state_update: process(inn)
    begin
        case currentState is
            when s0 =>
                if inn = '1' then
                    nextState <= s1;
                    outt <= '0';
                end if;

            when s1 =>
                if inn = '0' then
                    nextState <= s0;
                    outt <= '1';
                else
                    nextState <= s2;
                end if;
            when s2 =>
                -- This block is not visible in the provided image
            end case;
        end process state_update;
    end architecture rtl;
```



```
        outt <= '0';  
    end if;  
  
    when s2 =>  
        if inn = '0' then  
            nextState <= s0;  
            outt <= '1';  
        else  
            nextState <= s2;  
            outt <= '0';  
        end if;  
    end case;  
end process state_update;  
  
end architecture rtl;
```

b. Ubahlah kode berikut menjadi state diagram

