TRAFFICT LIGHT SIMULATION USING VHDL

Kelompok 7

Dosen: Indra Hartanto Tambunan, Ph.D

Anggota Kelompok :

Agung Pane (14S21026)

Natanael Situmorang (14S21033)

Elia Pasaribu (14S21005)

Institut Teknologi Del

I. PENDAHULUAN

Lalu lintas saat ini menjadi bagian penting dari manajemen perkotaan yang efektif dan aman. Sangat penting untuk mengembangkan dan menguji sistem kendali lampu lalu lintas. Tujuan proyek ini adalah untuk menggunakan bahasa VHDL (Very High-Speed Integrated Circuit Hardware Description Language) untuk merancang dan menguji sistem berbasis sirkuit terprogram untuk berbagai aplikasi, termasuk kendali lalu lintas. Dengan menggunakan VHDL, simulasi ini akan memungkinkan kita untuk memahami dan mengoptimalkan kinerja sistem kendali lampu lalu lintas dalam berbagai kondisi lalu lintas. Proyek ini ajuga memberi kita kesempatan untuk mempelajari dinamika lalu lintas dan cara mengatasi masalah yang mungkin muncul.

II. LANDASAN TEORETIS

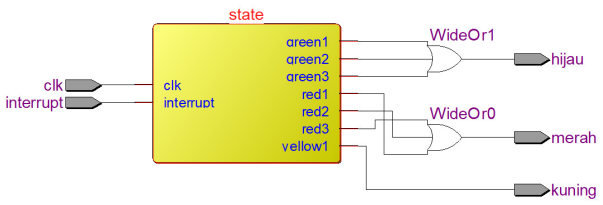
A. Finite State Machine (FSM)

mesin finite-state (FSM), adalah model komputasi matematis. FSM adalah mesin abstrak yang dapat berada tepat di salah satu dari banyak keadaan finit. FSM dapat berubah sesuai dengan banyak masukan; perubahan dari satu negara ke negara lain disebut transisi. FSM didasarkan pada daftar masukan transisi, keadaan awal, dan statusnya. Mesin finite-state (deterministik) dan non-finite-state adalah dua kategori mesin. Mesin keadaan-terbatas deterministik dan non-deterministik dapat dibangun dengan cara yang sama.

Perilaku FSM dapat dilihat di banyak perangkat yang ada di masyarakat modern, yang melakukan tindakan yang telah ditetapkan sebelumnya tergantung pada urutan peristiwa yang disajikan. Contoh sederhananya adalah mesin penjual otomatis, yang mengeluarkan produk ketika kombinasi koin yang tepat disimpan; lift, yang urutan pemberhentiannya ditentukan oleh lantai yang diminta pengendara; lampu lalu lintas, yang mengubah urutan saat mobil menunggu; dan kunci kombinasi, yang memerlukan masukan urutan angka dalam urutan yang benar.

III. HASIL DAN ANALISIS

A. Architecture



Dalam gambar ini, blok logika utama adalah state. Blok ini bertanggung jawab untuk menentukan keadaan lampu lalu lintas. Keadaan lampu lalu lintas dapat berupa merah, kuning, atau hijau. Blok state menerima input dari tiga blok, yaitu:

- WideOr1, yang menerima input dari lampu hijau untuk setiap arah lalu lintas.
- WideOr0, yang menerima input dari lampu merah untuk setiap arah lalu lintas.
- interrupt, yang menerima input dari interupsi.

Output dari blok state adalah sinyal yang menentukan keadaan lampu lalu lintas. Sinyal ini kemudian diteruskan ke blok-blok lampu lalu lintas, yaitu:

- green1, green2, dan green3, yang mengendalikan lampu hijau untuk setiap arah lalu lintas.
- red1, red2, dan red3, yang mengendalikan lampu merah untuk setiap arah lalu lintas.
- yellow1, yang mengendalikan lampu kuning untuk arah lalu lintas pertama.

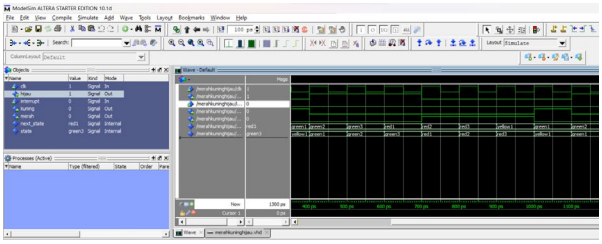
B. Code VHDL

Berikut kami lampirkan kode program yang kami gunakan untuk project kami:

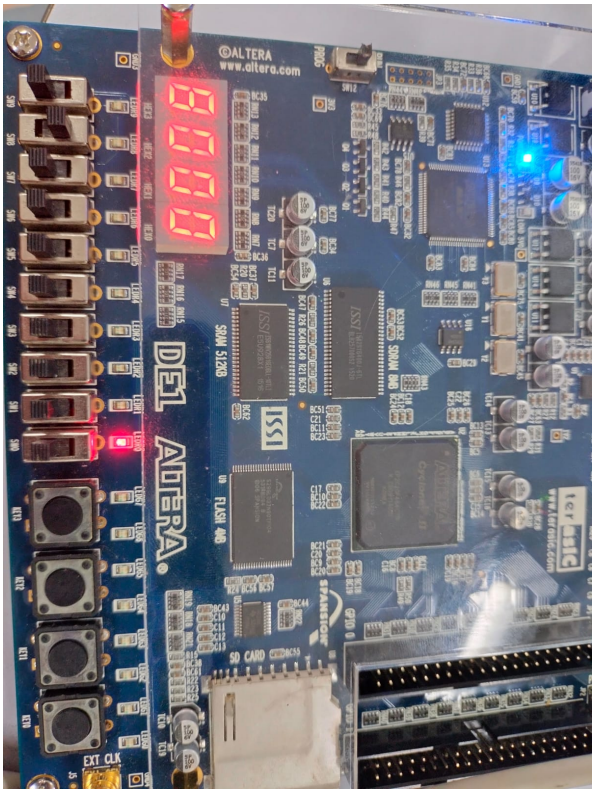
```
ENTITY merahkuninghijau IS
  PORT (
    clk, interrupt : IN  BIT;
    merah, kuning, hijau : OUT BIT
  );
END merahkuninghijau;

ARCHITECTURE behavior OF
merahkuninghijau IS
  TYPE state_type IS (red1, red2, red3,
yellow1, green1, green2, green3);
  SIGNAL state, next_state : state_type;

BEGIN
PROCESS (clk)
BEGIN
    IF clk'event AND clk = '1' THEN
        state <= next_state;
    END IF;
END PROCESS;

  PROCESS (state, interrupt)
  BEGIN
    CASE state IS
      WHEN red1 =>
        IF interrupt = '1' THEN
          next_state <= red3;
        ELSE
          next_state <= red2;
        END IF;

      WHEN red2 =>
        IF interrupt = '1' THEN
          next_state <= red3;
        ELSE
          next_state <= red3;
        END IF;

      WHEN red3 =>
        IF interrupt = '1' THEN
          next_state <= red3;
        ELSE
          next_state <= yellow1;
        END IF;

      WHEN yellow1 =>
        IF interrupt = '1' THEN
          next_state <= red3;
        ELSE
          next_state <= green1;
        END IF;

      WHEN green1 =>
        IF interrupt = '1' THEN
          next_state <= red3;
        ELSE
          next_state <= green2;
        END IF;

      WHEN green2 =>
        IF interrupt = '1' THEN
          next_state <= red3;
        ELSE
          next_state <= green3;
        END IF;

      WHEN green3 =>
        IF interrupt = '1' THEN
          next_state <= red3;
        ELSE
          next_state <= red1;
        END IF;
    END CASE;
  END PROCESS;

  PROCESS (state)
  BEGIN
    CASE state IS
      WHEN red1 | red2 | red3 =>
        merah <= '1';
        kuning <= '0';
        hijau <= '0';

      WHEN yellow1 =>
        merah <= '0';
        kuning <= '1';
        hijau <= '0';

      WHEN green1 | green2 | green3 =>
        merah <= '0';
        kuning <= '0';
        hijau <= '1';

      WHEN OTHERS =>
        merah <= '0';
        kuning <= '0';
        hijau <= '0';
    END CASE;
  END PROCESS;
END behavior;
```

Dan hasil yang diberikan seperti berikut:

- Modelsim



- Altera



Analisis:

kode tersebut merupakan suatu FSM yang menjalankan pengontrol lampu lalu lintas. FSM ini memiliki tujuh status: red1, red2, red3, yellow1, green1, green2, dan green3. lampu ini akan saling beralih antara keadaan berdasarkan nilai clk dan input interupsi. Input clk adalah sinyal clock yang menggerakkan transisi status FSM. Input interupsi adalah input asinkron yang dapat menyebabkan FSM bertransisi ke status red3 kapan saja.

IV. SIMPULAN

Proyek simulasi lampu lalu lintas menggunakan VHDL adalah upaya penting dalam merancang dan mengimplementasikan sistem kontrol lampu lalu lintas secara digital. Proyek ini bertujuan untuk mensimulasikan operasi lampu lalu lintas dengan memodelkan berbagai kondisi lalu lintas dan mengoptimalkan algoritma kontrol. Ini dilakukan dengan menggunakan kekuatan bahasa deskripsi perangkat keras VHDL. Metode ini membantu proyek ini memahami proses perancangan perangkat keras dan memungkinkan pengujian dan validasi yang efektif sebelum diimplementasikan dalam lingkungan nyata. Proyek ini berhasil menunjukkan berbagai keadaan lampu lalu lintas dan bagaimana mereka bertindak terhadap gangguan (gangguan).

REFERENSI

[1.] *Modul 5 Praktikum Arsitektur Sistem Komputer*. Teknik Elektro IT Del.2019

[2.] Hanindhito, Bagus. *Modul Praktikum EL3111, Arsitektur Sistem Komputer*, Sekolah Teknik Elektro dan Informatika, Bandung, 2014.

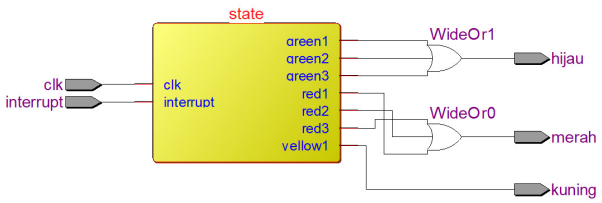
[3.] Bryant, Randal E. dan David R. O'Hallaron. *Computer System: A Programmer's Perspective.Prentice Hall*. USA.2011

[4.] Bryant, Randal, dan David O'Hallaron. *Computer Systems: A Programmer's Perspective 2nd Edition*. 2011. Massachusetts: Pearson Education Inc.

[5.] Patterson, David, dan John Hennessy. *Computer Organization and Design: The Hardware/Software Interface*. 2012. Waltham: Elsevier Inc.

LAMPIRAN

Architecture



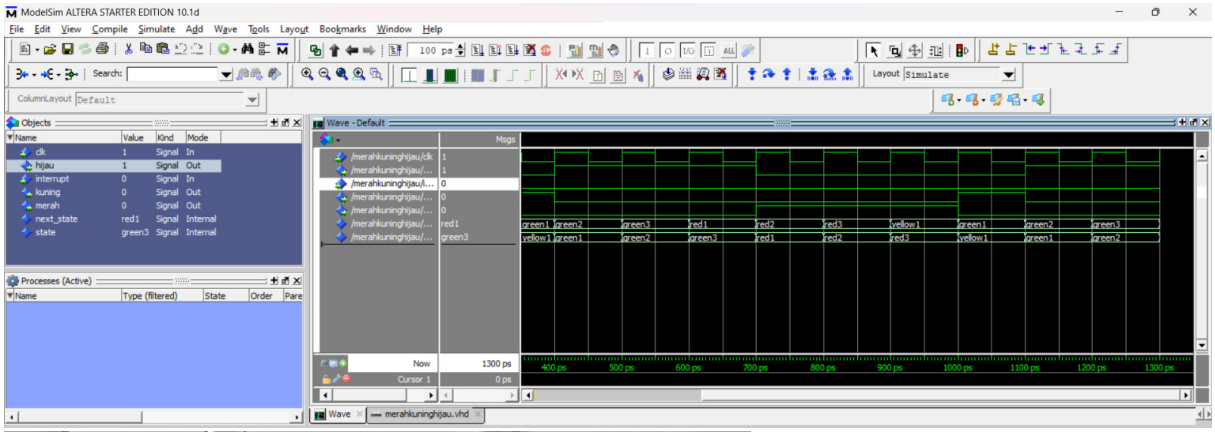
Code:

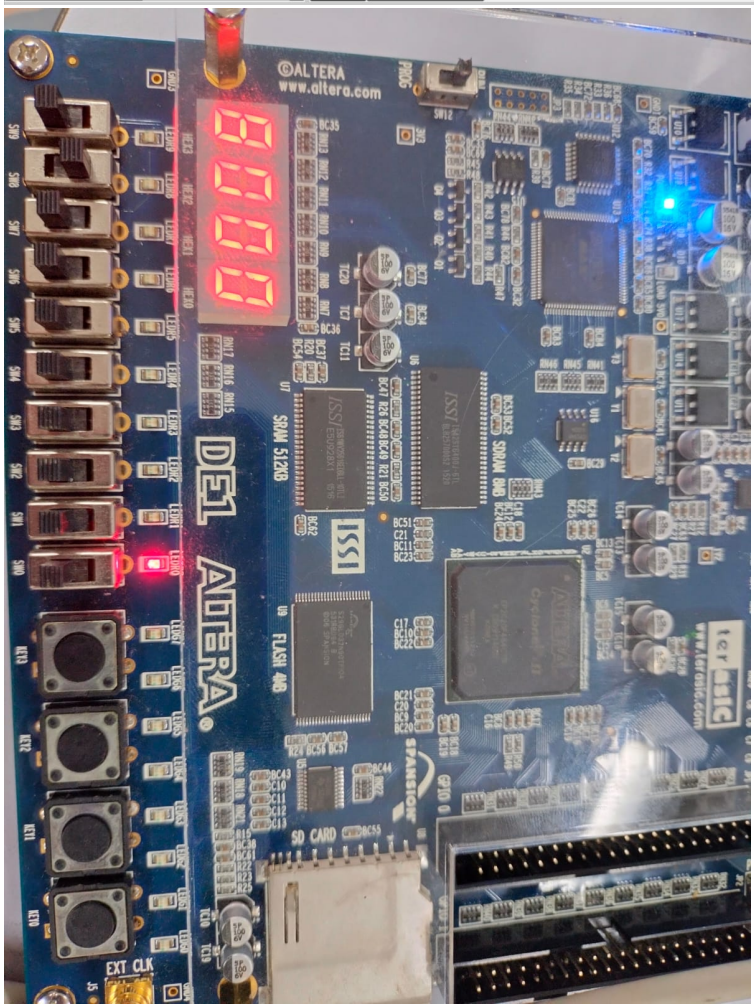
```
ENTITY merahkuninghijau IS
  PORT (
    clk, interrupt : IN  BIT;
    merah, kuning, hijau : OUT BIT
  );
END merahkuninghijau;

ARCHITECTURE behavior OF merahkuninghijau IS
  TYPE state_type IS (red1, red2, red3, yellow1, green1, green2, green3);
  SIGNAL state, next_state : state_type;

BEGIN
PROCESS (clk)
BEGIN
    IF clk'event AND clk = '1' THEN
        state <= next_state;
    END IF;
END PROCESS;

  PROCESS (state, interrupt)
  BEGIN
    CASE state IS
      WHEN red1 =>
        IF interrupt = '1' THEN
          next_state <= red3;
        ELSE
          next_state <= red2;
        END IF;

      WHEN red2 =>
        IF interrupt = '1' THEN
          next_state <= red3;
        ELSE
          next_state <= red3;
        END IF;

      WHEN red3 =>
```

```
      IF interrupt = '1' THEN
        next_state <= red3;
      ELSE
        next_state <= yellow1;
      END IF;

    WHEN yellow1 =>
      IF interrupt = '1' THEN
        next_state <= red3;
      ELSE
        next_state <= green1;
      END IF;

    WHEN green1 =>
      IF interrupt = '1' THEN
        next_state <= red3;
      ELSE
        next_state <= green2;
      END IF;

    WHEN green2 =>
      IF interrupt = '1' THEN
        next_state <= red3;
      ELSE
        next_state <= green3;
      END IF;

    WHEN green3 =>
      IF interrupt = '1' THEN
        next_state <= red3;
      ELSE
        next_state <= red1;
      END IF;
  END CASE;
END PROCESS;

PROCESS (state)
BEGIN
  CASE state IS
    WHEN red1 | red2 | red3 =>
      merah <= '1';
      kuning <= '0';
      hijau <= '0';

    WHEN yellow1 =>
      merah <= '0';
      kuning <= '1';
      hijau <= '0';

    WHEN green1 | green2 | green3 =>
      merah <= '0';
      kuning <= '0';
      hijau <= '1';
```

```
        WHEN OTHERS =>
          merah <= '0';
          kuning <= '0';
          hijau <= '0';
      END CASE;
    END PROCESS;
END behavior;
```

Output:





Link Video Presentasi:
https://youtu.be/KnGg-tSHJY8