



**FUNDAMENTAL OF DIGITAL SYSTEM FINAL PROJECT REPORT
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITAS INDONESIA**

BADMINTON UMPIRE HELPER

GROUP BF-2

MUHAMMAD RAIHAN AZHARI	2006468320
MOCHAMMAD SHAFFA P	2006528105
FIKRI AFIF MUSYAFFA	2006536662
ZANA NISWAH AWAHITA	2006577593

Kata Pengantar

Puji syukur kami panjatkan atas kehadiran Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya kami dapat menyelesaikan laporan proyek akhir praktikum Perancangan Sistem Digital ini. Proyek akhir ini mengangkat tema dari permainan badminton dan kami beri judul “Badminton Umpire Helper”.

Laporan ini dibuat dalam rangka pemenuhan tugas proyek akhir praktikum Perancangan Sistem Digital yang tentunya dibantu oleh banyak pihak. Untuk itu, kami juga ingin mengucapkan terima kasih kepada dosen pengampu mata kuliah Perancangan Sistem Digital, Bapak I Gde Dharma Nugraha, S.T., M.T., Ph.D dan Bapak Yan Maraden, S.T., M.T., M.Sc, serta Kak Fulky Hariz Zulkarnaen selaku asisten laboratorium yang telah mendampingi kami dalam menyelesaikan proyek akhir ini.

Terlepas dari semua itu, kami menyadari bahwa laporan ini masih memiliki banyak kekurangan. Untuk itu kami sangat terbuka atas kritik dan saran yang membangun demi kebaikan dari pembaca dalam rangka penyempurnaan kedepannya. Besar harapan kami laporan ini dapat membawa banyak manfaat bagi kami sebagai penulis dan tentunya juga bagi para pembaca.

Depok, Desember 08, 2021

Group PSD 2 BF-2

Daftar Isi

CHAPTER 1	4
PENDAHULUAN	4
1.1 LATAR BELAKANG.....	4
1.2 DESKRIPSI PROYEK.....	4
1.3 TUJUAN	5
1.4 PERAN DAN TANGGUNG JAWAB.....	5
CHAPTER 2	6
IMPLEMENTASI.....	6
2.1 PERALATAN	6
2.2 IMPLEMENTASI.....	6
CHAPTER 3	9
PENGETESAN DAN ANALISIS	9
3.1 PENGETESAN	9
3.2 HASIL	111
3.3 ANALISIS.....	13
CHAPTER 4	15
KESIMPULAN.....	15

CHAPTER 1

PENDAHULUAN

1.1 LATAR BELAKANG

Badminton merupakan salah satu cabang olahraga di Indonesia yang cukup mendapat perhatian dan prioritas dari masyarakat dan pemerintah. Hal ini terbukti dengan banyaknya masyarakat Indonesia yang gemar melakukan olahraga ini. Namun, seperti yang kita ketahui, tidak semua masyarakat Indonesia memahami aturan-aturan badminton yang diterapkan di Indonesia. Hal ini pun berlaku bagi seorang wasit (umpire). Meskipun pertandingan badminton diawasi seorang wasit, jika ia melakukan satu kesalahan saja, maka hasil pertandingan sudah pasti tidak *fair* dan akan membuat kekacauan.

Hal yang menjadi prioritas utama dari seorang wasit ialah konsistensi. Misalnya, konsistensi wasit tersebut dalam menentukan posisi pemain dan pemain mana yang berhak melakukan servis. Agar kondisi yang diterapkan sesuai dengan peraturan yang berlaku, tentunya akan lebih baik jika pertandingan dilengkapi dengan alat yang dapat membantu wasit dalam menyamakan kondisi sesungguhnya dengan kondisi pada diagram atau truth table. Hal ini juga berguna untuk memastikan bahwa wasit tersebut sudah benar dalam menentukan posisi pemain pada kedua team dan pemain mana yang berhak servis.

Selain itu, cara wasit tersebut menghitung poin pertandingan badminton juga sangat krusial dalam pertandingan badminton. Jika ia lengah pada satu poin saja, maka poin-poin selanjutnya akan salah pula. Oleh karena itu, pertandingan tersebut sebaiknya dilengkapi dengan alat yang dapat menyimpan poin secara otomatis. Salah satu alat yang dapat mengimplementasikan hal tersebut ialah seven segment display. Dengan adanya seven segment tersebut, maka poin-poin pertandingan akan secara otomatis tersimpan di seven segment sekaligus menampilkannya secara langsung.

1.2 DESKRIPSI PROYEK

Sistem Badminton Umpire Helper ini terdiri dari 2 fitur, yaitu penghitung poin yang dibuat menggunakan konsep counter dan penentu kondisi servis menggunakan konsep FSM tipe Moore. Fitur penghitung poin ini menggunakan 2 buah seven segment yang fungsinya untuk menampilkan poin yang diperoleh pemain badminton. Sedangkan untuk fitur penentu

servis didasari atas 2 kondisi, yaitu pemain mana yang harus melakukan servis dan di posisi lapangan mana pemain tersebut harus melakukan servis. Output dari rangkaian ini adalah poin dari kedua tim, posisi pemain pada kedua tim sebelum servis dilakukan, dan pemain mana yang berhak melakukan servis.

1.3 TUJUAN

Tujuan dari proyek ini ialah:

1. Memastikan posisi para pemain badminton sudah sesuai aturan
2. Menentukan pemain badminton mana yang akan melakukan servis
3. Menyimpan dan menampilkan poin dalam bentuk seven segment yang terintegrasi
4. Memudahkan wasit pada pertandingan badminton tingkat RT/RW dan sejenisnya (pertandingan semi formal)

1.4 PERAN DAN TANGGUNG JAWAB

Peran dan tanggung jawab yang ada pada kelompok kami adalah sebagai berikut:

Roles	Responsibilities	Person
Membuat Top Level	Menggabungkan component-component yang ada	Zana
Membuat FSM	Menentukan kondisi servis dan posisi melalui FSM tipe Moore Machine	Azhari
Membuat Counter	Menyimpan poin dalam bentuk seven segment 2 bit	Fikri
Membuat Testbench	Memeriksa input/output berdasarkan pertandingan sebenarnya	Shaffa
Membuat Laporan	Membuktikan pertanggung jawaban dari pelaksanaan proyek akhir	Seluruh anggota

Table 1. Roles and Responsibilities

CHAPTER 2

IMPLEMENTASI

2.1 PERALATAN

Alat-alat yang kami gunakan untuk membuat proyek ini diantaranya:

- Notepad++
- ModelSim
- Microsoft Word
- <https://app.diagrams.net/>

2.2 IMPLEMENTASI

Pertama-tama kami merancang Finite State Machine untuk menentukan kondisi siapa yang akan melakukan servis. Kami menggunakan moore machine karena dalam pengaplikasiannya output yang dihasilkan tidak dipengaruhi oleh input, melainkan dipengaruhi oleh state pada saat itu saja. Untuk menguji apakah program ini sudah benar atau belum, kami mengambil contoh pertandingan badminton yang asli. Kemudian kami analisis poin-poin yang diperoleh team X dan team Y, serta siapa yang akan servis pada saat team X mendapatkan poin dan sebaliknya. Team X terdiri dari pemain A dan B serta team Y terdiri dari pemain C dan D. Cara kerja dari finite state machine tersebut sesuai dengan state diagram pada gambar dibawah ini.

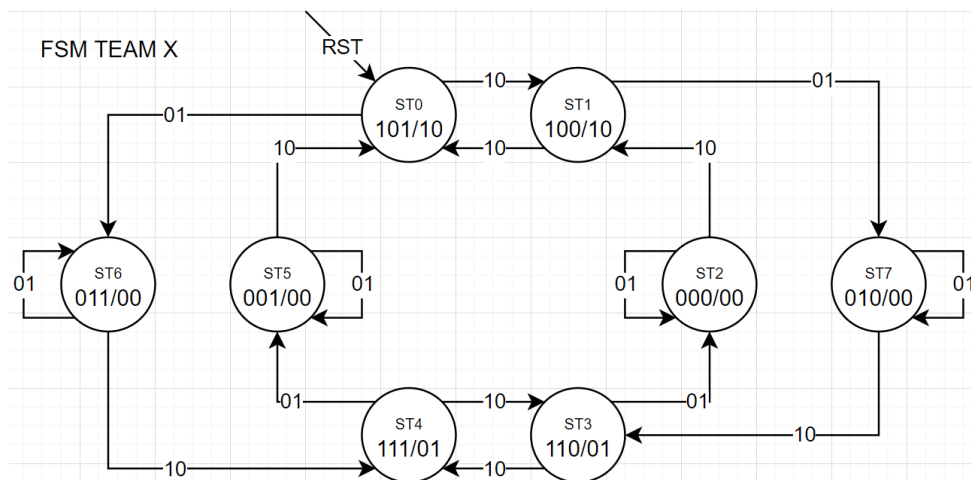


Fig 1. State diagram for Team X

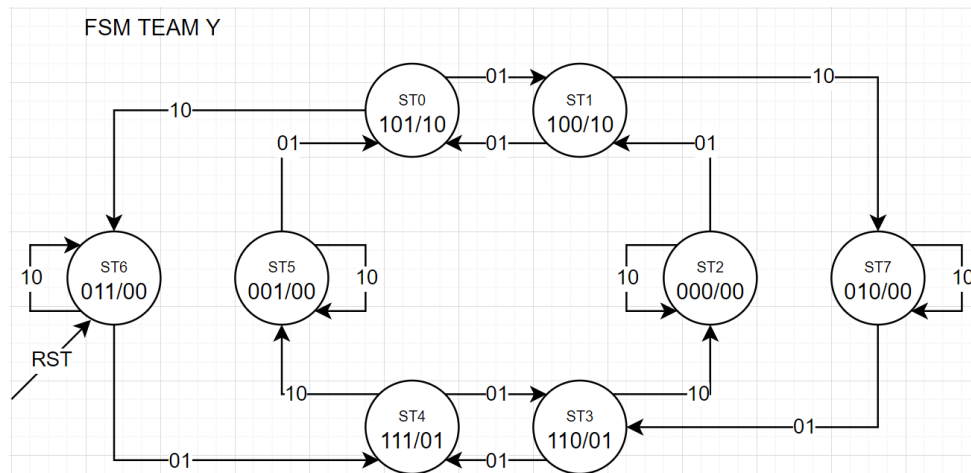


Fig 2. State diagram for Team Y

State diagram tersebut disusun berdasarkan referensi peraturan badminton resmi dari BWF yang sudah tertulis di bagian referensi pada. State diagram tersebut menghasilkan output berupa ServX atau ServY (Pemain mana yang berhak melakukan servis). Kondisi pada finite state diagram bergantung pada 3 keadaan, dan masing-masing keadaan memiliki 2 kemungkinan, sehingga dibutuhkan 8 states dalam finite state diagram ini. BitMSB pada kondisi menunjukkan apakah servis dilakukan oleh team terkait (bit akan bernilai 1) atau dilakukan oleh team lawan (bit akan bernilai 0).

Bit kedua pada kondisi FSM menunjukkan pemain mana yang terakhir melakukan servis. Untuk FSM team X, jika pemain terakhir yang melakukan servis adalah pemain B, maka bit ini akan bernilai 0. Jika pemain terakhir yang melakukan servis adalah pemain A, maka bit ini akan bernilai 1. Sedangkan untuk FSM team Y, jika pemain terakhir yang melakukan servis adalah pemain D, maka bit ini akan bernilai 0. Namun, jika pemain terakhir yang melakukan servis adalah pemain C, maka bit ini akan bernilai 1.

Bit ketiga pada kondisi FSM menunjukkan posisi dari pemain ketika servis dilakukan. Untuk FSM team X, jika pemain A berada di kiri dan pemain B berada di kanan, maka bit akan bernilai 0, dan jika sebaliknya akan bernilai 1. Sedangkan, untuk FSM team Y, jika pemain AC berada di kiri dan pemain D berada di kanan, maka bit akan bernilai 0, dan jika sebaliknya akan bernilai 1.

Sembari melakukan analisis poin dan servis, kami juga bekerja secara paralel untuk hasil output yang diterjemahkan melalui seven segment. Kami merancang seven segment 2 bit

pada setiap team. Sehingga total terdapat 4 seven segment. Setiap poin yang diperoleh akan langsung ditampilkan pada seven segment. Diperlukan modifikasi program VHDL untuk menyinkronkan seven segment 2 bit. Ketika seven segment kanan sudah menunjukkan angka 9, maka seven segment kiri berubah dari angka 0 menjadi angka 1. Pada waktu yang bersamaan, seven segment kanan berubah dari angka 9 menjadi angka 0.

Setelah menyelesaikan state table dan seven segment, kami mulai merancang top level. Menggabungkan seven segment, clocking, dan input dari button. Top level selesai, kemudian kami membuat testbench program secara menyeluruh. Bagian ini merupakan bagian yang membutuhkan waktu cukup banyak dalam penyelesaiannya. Karena kami harus melihat dan memastikan berkali-kali bahwa hasil dari truth table dengan pertandingan sesungguhnya harus sama. Kami mengambil contoh pertandingan sesungguhnya yaitu Ahsan dan Setiawan melawan Gideon dan Sukamuljo, link siaran ulang disematkan pada referensi di bawah.

Dengan diselesaikannya testbench maka program BADMINTON UMPIRE HELPER yang telah kami rancang sudah siap diimplementasikan ke dalam rangkaian sesungguhnya. Namun, karena pada mata kuliah ini kami selaku praktikan hanya sampai membuat simulasinya dan tidak sampai membuat alat sesungguhnya, maka rangkaian simulasi kami telah selesai.

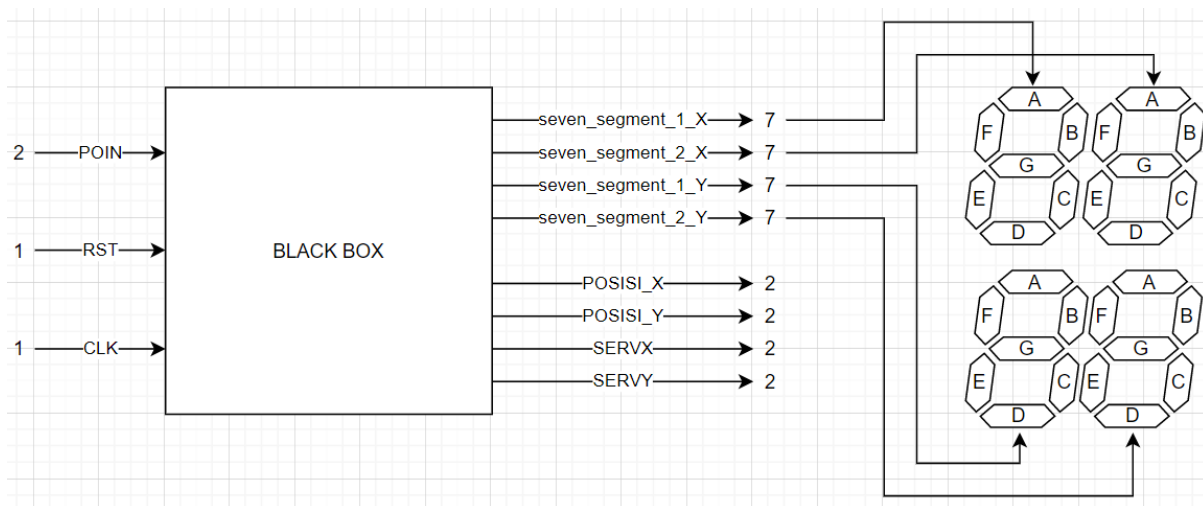


Fig 3. Schematic Black Box

CHAPTER 3

PENGETESAN DAN ANALISIS

3.1 PENGETESAN

Pengujian rangkaian digital badminton umpire helper ini menggunakan testbench berdasarkan pertandingan final BWF world tour pada cabang ganda putra antara Marcus Gideon - Kevin Sanjaya melawan Mohammad Ahsan – Hendra Setiawan. Testbench dilakukan sampai keadaan poin 10 – 11 untuk keunggulan pasangan Mohammad Ahsan dan Hendra Setiawan. Referensi pertandingan berupa video terdapat di bagian referensi dalam laporan ini. Pertandingan tersebut dijadikan parameter untuk truth table yang digunakan untuk pengetesan dibawah ini:

Nomor Test- bench	Poin	Present State				Next State			
		PosisiX	PosisiY	ServX	ServY	PosisiX	PosisiY	ServX	ServY
1.	00	01	01	10	00	01	01	10	00
2.	01	01	01	10	00	01	01	00	01
3.	00	01	01	00	01	01	01	00	01
4.	10	01	01	00	01	01	01	01	00
5.	00	01	01	01	00	01	01	01	00
6.	01	01	01	01	00	01	01	00	10
7.	00	01	01	00	10	01	01	00	10
8.	10	01	01	00	10	01	01	10	00
9.	00	01	01	10	00	01	01	10	00
10.	01	01	01	10	00	01	01	00	01
11.	00	01	01	00	01	01	01	00	01
12.	01	01	01	00	01	01	10	00	01
13.	00	01	10	00	01	01	10	00	01
14.	01	01	10	00	01	01	01	00	01
15.	00	01	01	00	01	01	01	00	01
16.	01	01	01	00	01	01	10	00	01
17.	00	01	10	00	01	01	10	00	01

18.	10	01	10	00	01	01	10	01	00
19.	00	01	10	01	00	01	10	01	00
20.	01	01	10	01	00	01	10	00	10
21.	00	01	10	00	10	01	10	00	10
22.	10	01	10	00	10	01	10	10	00
23.	00	01	10	10	00	01	10	10	00
24.	10	01	10	10	00	10	10	10	00
25.	00	10	10	10	00	10	10	10	00
26.	10	10	10	10	00	01	10	10	00
27.	00	01	10	10	00	01	10	10	00
28.	10	01	10	10	00	10	10	10	00
29.	00	10	10	10	00	10	10	10	00
30.	01	10	10	10	00	10	10	00	01
31.	00	10	10	00	01	10	10	00	01
32.	01	10	10	00	01	10	01	00	01
33.	00	10	01	00	01	10	01	00	01
34.	10	10	01	00	01	10	01	01	00
35.	00	10	01	01	00	10	01	01	00
36.	10	10	01	01	00	01	01	01	00
37.	00	01	01	01	00	01	01	01	00
38.	10	01	01	01	00	10	01	01	00
39.	00	10	01	01	00	10	01	01	00
40.	01	10	01	01	00	10	01	00	10
41.	00	10	01	00	10	10	01	00	10
42.	01	10	01	00	10	10	10	00	10
43.	00	10	10	00	10	10	10	00	10

Untuk role dalam pertandingan ini adalah sebagai berikut:

Team X:

Pemain A : Marcus Gideon

Pemain B : Kevin Sanjaya

Team Y:

Pemain C : Mohammad Ahsan

Pemain D : Hendra Setiawan

Input yang diberikan berupa 2 bit poin dengan deskripsi sebagai berikut:

00 : Kondisi hold (Permainan sedang berlangsung)

10 : Team X (Marcus – Kevin) mendapatkan 1 poin

01 : Team Y (Ahsan – Hendra) mendapatkan 1 poin

Sementara untuk deskripsi output adalah sebagai berikut:

Posisi team X (PosisiX)

01 : Marcus berada di sebelah kanan, Kevin berada di sebelah kiri

10 : Marcus berada di sebelah kiri, Kevin berada di sebelah kanan

Posisi team Y (PosisiY)

01 : Ahsan berada di sebelah kanan, Hendra berada di sebelah kiri

10 : Ahsan berada di sebelah kiri, Hendra berada di sebelah kanan

Servis yang dilakukan team X (ServX)

10 : Marcus melakukan servis

01 : Kevin melakukan servis

00 : Team X tidak berhak melakukan servis

Servis yang dilakukan team Y (ServY)

01 : Ahsan melakukan servis

10 : Hendra melakukan servis

3.2 HASIL

Rangkaian Badminton Umpire Helper yang telah dibuat diujikan menggunakan model sim dengan parameter yang sesuai dengan truth table diatas. Sehingga hasil dari simulasinya adalah sebagai berikut:

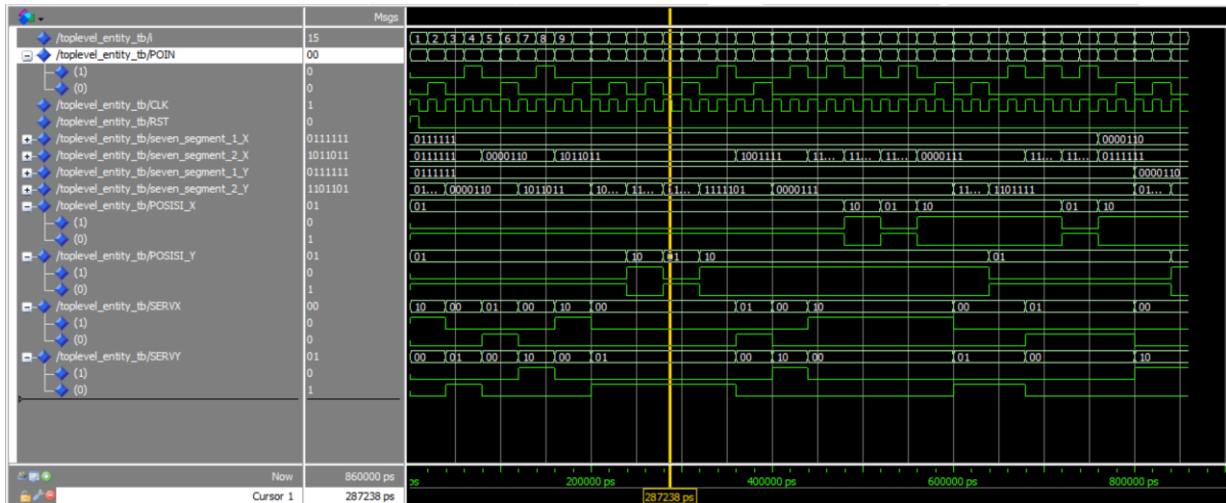


Fig 4. Testing Result

```

VSIM 3> run
VSIM 4> vsim -gui work.toplevel_entity_tb
# End time: 23:02:25 on Dec 07, 2021, Elapsed time: 0:00:40
# Errors: 0, Warnings: 4
# vsim -gui work.toplevel_entity_tb
# Start time: 23:02:25 on Dec 07, 2021
# Loading std.standard
# Loading std.textio(body)
# Loading ieee.std_logic_1164(body)
# Loading ieee.numeric_std(body)
# Loading work.toplevel_entity_tb(bench)
# Loading work.toplevel_entity(main_arch)
# Loading work.servixx(arch_servx)
# Loading work.servisy(arch_servy)
# Loading work.counter_1(arch)
add wave -position insertpoint sim:/toplevel_entity_tb/*
# ** Warning: (vsim-WLF-5000) WLF file currently in use: vsim.wlf
# File in use by: Owner Hostname: DESKTOP-EAJ2VTO ProcessID: 10492
# Attempting to use alternate WLF file "./wlfthil6m4".
# ** Warning: (vsim-WLF-5001) Could not open WLF file: vsim.wlf
# Using alternate file: ./wlfthil6m4
VSIM 6> run -all
VSIM 7>

```

Fig 5. Run Result

Pada gambar 3 tidak terdapat report yang menunjukkan bahwa output rangkaian sudah sesuai dengan testbench dan truth table. Untuk memastikan kebenaran dari truth table terhadap referensi pertandingan, diambil contoh satu kasus testbench untuk dicocokkan dengan pertandingan aslinya. Testbench yang diambil adalah testbench ke-15 dengan kondisi sebagai berikut:

	Mags
/toplevel_entity_tb/i	15
/toplevel_entity_tb/POIN	00
/toplevel_entity_tb/CLK	1
/toplevel_entity_tb/RST	0
/toplevel_entity_tb/seven_segment_1_X	0111111
/toplevel_entity_tb/seven_segment_2_X	1011011
/toplevel_entity_tb/seven_segment_1_Y	0111111
/toplevel_entity_tb/seven_segment_2_Y	1101101
/toplevel_entity_tb/POSISI_X	01
/toplevel_entity_tb/POSISI_Y	01
/toplevel_entity_tb/SERVX	00
/toplevel_entity_tb/SERVY	01

Fig 6. Output testbech ke-15

Kemudian untuk kondisi pada pertandingan sebenarnya adalah sebagai berikut:

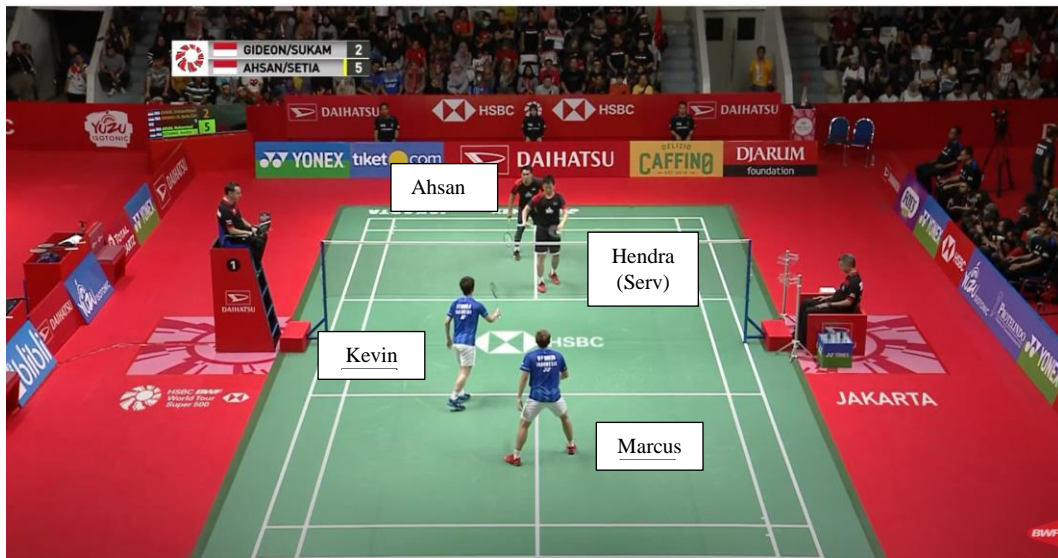


Fig 7. Pertandingan Final BWF World Tour

Pada team X, Marcus berada di kanan Kevin (posisi $X = 01$) dan team X tidak berhak untuk melakukan servis ($ServX = 00$). Untuk team Y, Ahsan berada di sebelah kanan Hendra (Sisi kanan dan kiri relatif sesuai dengan hadapan pemain ke net) (posisi $y = 01$) dan Hendra melakukan servis ($ServY = 01$).

Menurut output dari simulasi menggunakan model sim, pada testbench terkait poin dari team X (`seven_segment_2_X`) adalah 2 dan poin dari team Y (`seven_segment_2_Y`) adalah 5. Dimana pada signal vector `seven_segment_2_X` dan `seven_segment_2_Y`, nilai g merupakan MSB dan nilai a merupakan LSB. Output ini sesuai dengan pertandingan pada gambar 5, dimana poin untuk team X adalah 2 dan poin untuk team Y adalah 5.

3.3 ANALISIS

Berdasarkan simulasi yang dilakukan, keadaan dan output akan berubah apabila poin bernilai 01 atau 10 serta clock pulse naik (rising edge). Namun, perubahan pada output tersebut membutuhkan delay 1 clock pulse.

Pengujian dilakukan dengan input poin sesuai dengan pertandingan referensi, dan diselingi dengan angka 00 pada setiap perubahan poin sebagai hold state. Hold state ini diperlukan untuk dapat melihat perubahan (next state) dari suatu input point yang diberikan. Selain itu juga, pada saat sedang berlangsungnya permainan di kondisi sebenarnya (permainan dalam kondisi hidup) input poin sebelumnya akan di hold hingga perubahan poin selanjutnya.

Untuk output pada seven segment decoder, MSB merupakan bit pada pin g dan LSB merupakan bit pada pin a. Seven segment decoder menggunakan prinsip counter up 2 bit. Pada testbench jika seven segment decoder telah bernilai 9, maka seven segment decoder tersebut akan kembali ke 0 sedangkan seven segment decoder lainnya akan melakukan *increment*.

Pada bagian pengecekan servis, hasil simulasi tersebut sesuai dengan cara kerja finite state diagram pada gambar dan keterangan outputnya sudah dijelaskan pada bagian “Pengetesan” dalam laporan ini. Mengacu kepada peraturan servis permainan ganda oleh BWF, nilai ServX akan berubah jika team terkait kehilangan poin atau team terkait memenangkan poin dimana servis sebelumnya dilakukan oleh team lawan. Hal ini juga berlaku untuk nilai ServY. Untuk sinyal PosisiX, nilai akan berubah jika team terkait memenangkan poin dimana yang servis dilakukan oleh pemain yang berasal dari team terkait. Hal ini juga berlaku untuk nilai dari sinyal PosisiY.

CHAPTER 4

KESIMPULAN

Berdasarkan proyek akhir yang kami buat ini, kami dapat menyimpulkan bahwa:

- Fitur yang dimiliki Badminton Umpire Helper ini adalah penghitung poin dan penentu kondisi servis.
- Penghitung poin merupakan implementasi dari konsep counter, dan penentu kondisi servis merupakan implementasi dari konsep FSM tipe moore.
- Finite State Machine (FSM) yang digunakan adalah dalam bentuk model moore.
- Rangkaian ini memiliki 3 keluaran, yaitu poin dari kedua tim, posisi pemain pada kedua tim sebelum servis dilakukan, dan pemain mana yang berhak melakukan servis.
- Dibutuhkan seven segment display sebanyak 2 buah yang menampilkan poin perolehan untuk masing-masing tim, sehingga totalnya dibutuhkan 4 buah.
- Penggunaan hold state membantu memperjelas perubahan next state dari suatu input.

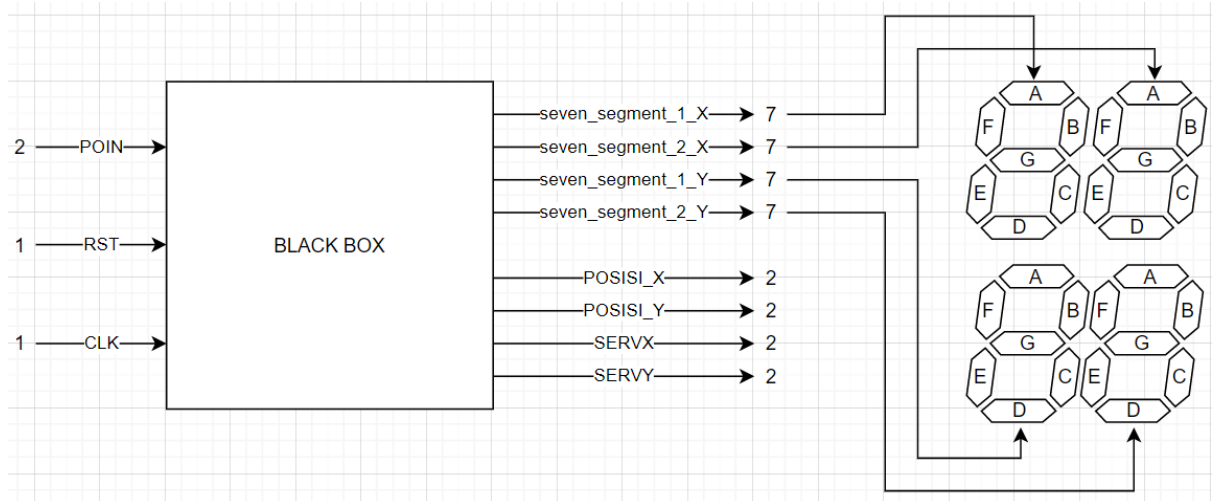
REFERENSI

- [1] TV, BWF. “F | MD | GIDEON/SUKAMULJO (INA) [1] vs. AHSAN/SETIAWAN (INA) [2] | BWF 2020” Youtube, diunggah oleh BWF TV, 19 Januari 2020, <https://www.youtube.com/watch?v=tikBq2hkdko>
- [2] Jamil, N. (2016). *Laws -Badminton Part II -Section 1A Laws of Badminton*. https://system.bwfbadminton.com/documents/folder_1_81/Regulations/Laws/Part%20II%20Section%201A%20-%20Laws%20of%20Badminton%20-%20June%202016%20Revised%202.pdf
- [3] Jonas Julian Jensen. (January, 2020). “DUAL 7-SEGMENT DISPLAY FPGA CONTROLLER” <https://vhdlwhiz.com/dual-7-segment-display/>
- [4] Pamungkas, Maulana Aditya. 2019. “PENGEMBANGAN APLIKASI PERWASITAN BULUTANGKIS”. Skripsi. Tidak diterbitkan. Fakultas Ilmu Keolahragaan. Universitas Negeri Yogyakarta: Yogyakarta. http://eprints.uny.ac.id/66820/1/SKRIPSI_MaulanaAdityaPamungkas_15602241050.pdf

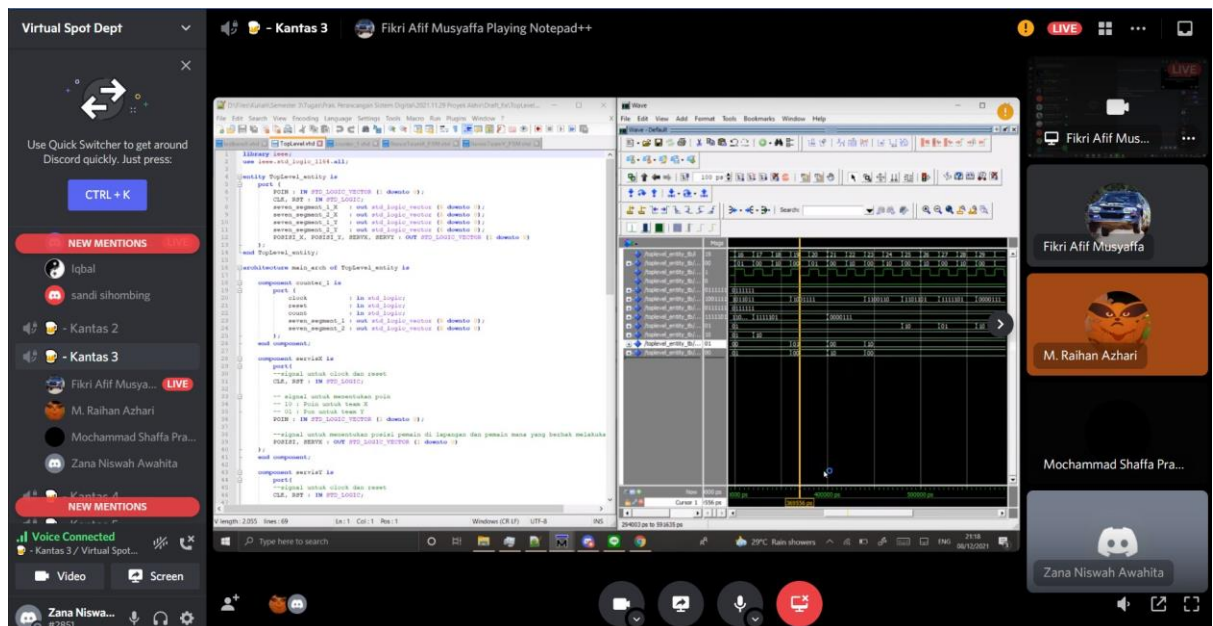
LAMPIRAN

Lampiran A: Skematik Proyek

Final Schematic



Lampiran B: Dokumentasi



ServisTeamX_FSM.vhd

```
library ieee;
use ieee.std_logic_1164.all ;

entity servisX is
```

```

port(
    --signal untuk clock dan reset
    CLK, RST : IN STD_LOGIC;

    -- signal untuk menentukan poin
    -- 10 : Poin untuk team X
    -- 01 : Poin untuk team Y
    POIN : IN STD_LOGIC_VECTOR (1 downto 0);

    --signal untuk menentukan posisi pemain di lapangan dan pemain mana
    yang berhak melakukan servisX
    POSISI, SERVX : OUT STD_LOGIC_VECTOR (1 downto 0)
);

end servisX;

architecture arch_servx of servisX is
    --state dari FSM
    type state_types is (ST0, ST1, ST2, ST3, ST4, ST5, ST6, ST7);
    signal present_state, next_state : state_types;
    --kondisi dari FSM dalam biner
    signal kondisi : std_logic_vector (2 downto 0);
begin

    sync_proc : process(CLK, next_state, RST)
    begin
        --jika reset bernilai 1 maka akan dikembalikan ke ST0 sebagai kondisi
        default memulai match
        if(RST = '1') then
            present_state <= ST0;
        elsif(rising_edge(CLK)) then
            present_state <= next_state;
        end if;
    end process sync_proc;

    comb_proc : process(present_state, POIN)
    begin
        --Output di set ke angka 00 untuk mencegah bug
        SERVX <= "00";
        POSISI <= "00";

        case present_state is
            --Assign kondisi sesuai dengan FSM
            --Keterangan SERVX
            --10 = Pemain A berhak melakukan servis
            --01 = Pemain B berhak melakukan servis
            --00 = Pemain team X tidak berhak melakukan servis
            --Keterangan Posisi
            -- 01 = Pemain A dikanan, Pemain B dikiri
            -- 10 = Pemain A dikiri, Pemain B dikanan
            when ST0 =>
                kondisi <= "101";
                SERVX <= "10";
                POSISI <= "01";
                --Jika team X mendapat poin
                if(POIN = "10") then
                    next_state <= ST1;
                --jika team Y mendapat point
                elsif(POIN = "01") then

```

```

        next_state <= ST6;
    --kondisi hold
else
    next_state <= ST0;
end if;
when ST1 =>
    kondisi <= "100";
    SERVX <= "10";
    POSISI <= "10";
    --Jika team X mendapat poin
    if(POIN = "10") then
        next_state <= ST0;
    --jika team Y mendapat point
    elsif(POIN = "01") then
        next_state <= ST7;
    --kondisi hold
    else
        next_state <= ST1;
    end if;
when ST2 =>
    kondisi <= "000";
    SERVX <= "00";
    POSISI <= "10";
    --Jika team X mendapat poin
    if(POIN = "10") then
        next_state <= ST1;
    --jika team Y mendapat point
    elsif(POIN = "01") then
        next_state <= ST2;
    --kondisi hold
    else
        next_state <= ST2;
    end if;
when ST3 =>
    kondisi <= "110";
    SERVX <= "01";
    POSISI <= "10";
    --Jika team X mendapat poin
    if(POIN = "10") then
        next_state <= ST4;
    --jika team Y mendapat point
    elsif(POIN = "01") then
        next_state <= ST2;
    --kondisi hold
    else
        next_state <= ST3;
    end if;
when ST4 =>
    kondisi <= "111";
    SERVX <= "01";
    POSISI <= "01";
    --Jika team X mendapat poin
    if(POIN = "10") then
        next_state <= ST3;
    --jika team Y mendapat point
    elsif(POIN = "01") then
        next_state <= ST5;
    --kondisi hold
    else
        next_state <= ST4;
    end if;

```

```

when ST5 =>
    kondisi <= "001";
    SERVX <= "00";
    POSISI <= "01";
    --Jika team X mendapat poin
    if(POIN = "10") then
        next_state <= ST0;
    --jika team Y mendapat point
    elsif(POIN = "01") then
        next_state <= ST5;
    --kondisi hold
    else
        next_state <= ST5;
    end if;
when ST6 =>
    kondisi <= "011";
    SERVX <= "00";
    POSISI <= "01";
    --Jika team X mendapat poin
    if(POIN = "10") then
        next_state <= ST4;
    --jika team Y mendapat point
    elsif(POIN = "01") then
        next_state <= ST6;
    --kondisi hold
    else
        next_state <= ST6;
    end if;
when ST7 =>
    kondisi <= "010";
    SERVX <= "00";
    POSISI <= "10";
    --Jika team X mendapat poin
    if(POIN = "10") then
        next_state <= ST3;
    --jika team Y mendapat point
    elsif(POIN = "01") then
        next_state <= ST7;
    --kondisi hold
    else
        next_state <= ST7;
    end if;
end case;
end process comb_proc;
end arch_servx;

```

ServisTeamY_FSM

```

library ieee;
use ieee.std_logic_1164.all ;

entity servisY is
    port(
        --signal untuk clock dan reset

```

```

    CLK, RST : IN STD_LOGIC;

    -- signal untuk menentukan poin
    -- 10 : Poin untuk team X
    -- 01 : Poin untuk team Y
    POIN : IN STD_LOGIC_VECTOR (1 downto 0);

    --signal untuk menentukan posisi pemain di lapangan dan pemain mana
    yang berhak melakukan servis
    POSISI, SERVY : OUT STD_LOGIC_VECTOR (1 downto 0)
);

end servisY;

architecture arch_servY of servisY is
    --state dari FSM
    type state_types is (ST6, ST0, ST1, ST2, ST3, ST4, ST5,ST7);
    signal present_state, next_state : state_types;
    --kondisi dari FSM dalam biner
    signal kondisi : std_logic_vector (2 downto 0);
begin

    sync_proc : process(CLK, next_state, RST)
    begin
        --jika reset bernilai 1 maka akan dikembalikan ke ST0 sebagai kondisi
        default memulai match
        if(RST = '1') then
            present_state <= ST6;
        elsif(rising_edge(CLK)) then
            present_state <= next_state;
        end if;
    end process sync_proc;

    comb_proc : process(present_state, POIN)
    begin
        --Output di set ke angka 00 untuk mencegah bug
        SERVY <= "00";
        POSISI <= "00";

        case present_state is
            --Assign kondisi sesuai dengan FSM
            --Keterangan SERVX
            --10 = Pemain C berhak melakukan servis
            --01 = Pemain D berhak melakukan servis
            --00 = Pemain team Y tidak berhak melakukan servis
            --Keterangan Posisi
            -- 01 = Pemain C dikanan, Pemain D dikiri
            -- 10 = Pemain C dikiri, Pemain D dikanan
            when ST6 =>
                kondisi <= "011";
                SERVY <= "00";
                POSISI <= "01";
                --Jika team Y mendapat poin
                if(POIN = "01") then
                    next_state <= ST4;
                --jika team X mendapat point
                elsif(POIN = "10") then
                    next_state <= ST6;
                --kondisi hold
                else

```

```

        next_state <= ST6;
    end if;
when ST0 =>
    kondisi <= "101";
    SERVY <= "10";
    POSISI <= "01";
    --Jika team Y mendapat poin
    if(POIN = "01") then
        next_state <= ST1;
    --jika team X mendapat point
    elsif(POIN = "10") then
        next_state <= ST6;
    --kondisi hold
    else
        next_state <= ST0;
    end if;
when ST1 =>
    kondisi <= "100";
    SERVY <= "10";
    POSISI <= "10";
    --Jika team Y mendapat poin
    if(POIN = "01") then
        next_state <= ST0;
    --jika team X mendapat point
    elsif(POIN = "10") then
        next_state <= ST7;
    --kondisi hold
    else
        next_state <= ST1;
    end if;
when ST2 =>
    kondisi <= "000";
    SERVY <= "00";
    POSISI <= "10";
    --Jika team Y mendapat poin
    if(POIN = "01") then
        next_state <= ST1;
    --jika team X mendapat point
    elsif(POIN = "10") then
        next_state <= ST2;
    --kondisi hold
    else
        next_state <= ST2;
    end if;
when ST3 =>
    kondisi <= "110";
    SERVY <= "01";
    POSISI <= "10";
    --Jika team Y mendapat poin
    if(POIN = "01") then
        next_state <= ST4;
    --jika team X mendapat point
    elsif(POIN = "10") then
        next_state <= ST2;
    --kondisi hold
    else
        next_state <= ST3;
    end if;
when ST4 =>
    kondisi <= "111";
    SERVY <= "01";

```

```

        POSISI <= "01";
        --Jika team Y mendapat poin
        if(POIN = "01") then
            next_state <= ST3;
        --jika team X mendapat point
        elsif(POIN = "10") then
            next_state <= ST5;
        --kondisi hold
        else
            next_state <= ST4;
        end if;
    when ST5 =>
        kondisi <= "001";
        SERVY <= "00";
        POSISI <= "01";
        --Jika team Y mendapat poin
        if(POIN = "01") then
            next_state <= ST0;
        --jika team X mendapat point
        elsif(POIN = "10") then
            next_state <= ST5;
        --kondisi hold
        else
            next_state <= ST5;
        end if;

    when ST7 =>
        kondisi <= "010";
        SERVY <= "00";
        POSISI <= "10";
        --Jika team Y mendapat poin
        if(POIN = "01") then
            next_state <= ST3;
        --jika team X mendapat point
        elsif(POIN = "10") then
            next_state <= ST7;
        --kondisi hold
        else
            next_state <= ST7;
        end if;
    end case;
end process comb_proc;
end arch_servY;

```

Counter_1

```

library ieee;
use ieee.std_logic_1164.all;

entity counter_1 is
    port (
        clock          : in std_logic;
        reset           : in std_logic;
        count           : in std_logic;

```

```

        seven_segment_1 : out std_logic_vector (6 downto 0) := "0111111";
        seven_segment_2 : out std_logic_vector (6 downto 0) := "0111111"
    );
end counter_1;

architecture arch of counter_1 is
    signal digit : integer range 0 to 30;
begin
    process (clock, reset, count, digit)
    begin
        if rising_edge(clock) then
            if reset = '1' then
                seven_segment_1 <= "0111111";
                seven_segment_2 <= "0111111";
                digit <= 0;
            else
                if count = '1' then
                    digit <= digit + 1;
                    case digit is
                        when 0 => seven_segment_2 <= "0000110";
                        when 1 => seven_segment_2 <= "1011011";
                        when 2 => seven_segment_2 <= "1001111";
                        when 3 => seven_segment_2 <= "1100110";
                        when 4 => seven_segment_2 <= "1101101";
                        when 5 => seven_segment_2 <= "1111101";
                        when 6 => seven_segment_2 <= "0000111";
                        when 7 => seven_segment_2 <= "1111111";
                        when 8 => seven_segment_2 <= "1101111";
                        when 9 =>
                            seven_segment_1 <= "0000110";
                            seven_segment_2 <= "0111111";
                        when 10 => seven_segment_2 <= "0000110";
                        when 11 => seven_segment_2 <= "1011011";
                        when 12 => seven_segment_2 <= "1001111";
                        when 13 => seven_segment_2 <= "1100110";
                        when 14 => seven_segment_2 <= "1101101";
                        when 15 => seven_segment_2 <= "1111101";
                        when 16 => seven_segment_2 <= "0000111";
                        when 17 => seven_segment_2 <= "1111111";
                        when 18 => seven_segment_2 <= "1101111";
                        when 19 =>
                            seven_segment_1 <= "1011011";
                            seven_segment_2 <= "0111111";
                        when 20 => seven_segment_2 <= "0000110";
                        when others =>
                            seven_segment_1 <= "0111111";
                            seven_segment_2 <= "0111111";
                    end case;
                end if;
            end if;
        end if;
    end process;
end arch;

```

TopLevel


```

library ieee;
use ieee.std_logic_1164.all;

entity TopLevel_entity is
  port (
    POIN : IN STD_LOGIC_VECTOR (1 downto 0);
    CLK, RST : IN STD_LOGIC;
    seven_segment_1_X : out std_logic_vector (6 downto 0);
    seven_segment_2_X : out std_logic_vector (6 downto 0);
    seven_segment_1_Y : out std_logic_vector (6 downto 0);
    seven_segment_2_Y : out std_logic_vector (6 downto 0);
    POSISI_X, POSISI_Y, SERVX, SERVY : OUT STD_LOGIC_VECTOR (1 downto
0)
  );
end TopLevel_entity;

architecture main_arch of TopLevel_entity is

  component counter_1 is
    port (
      clock      : in std_logic;
      reset      : in std_logic;
      count      : in std_logic;
      seven_segment_1 : out std_logic_vector (6 downto 0);
      seven_segment_2 : out std_logic_vector (6 downto 0)
    );
  end component;

  component servisX is
    port(
      --signal untuk clock dan reset
      CLK, RST : IN STD_LOGIC;

      -- signal untuk menentukan poin
      -- 10 : Poin untuk team X
      -- 01 : Pon untuk team Y
      POIN : IN STD_LOGIC_VECTOR (1 downto 0);

      --signal untuk menentukan posisi pemain di lapangan dan pemain mana
yang berhak melakukan servisX
      POSISI, SERVX : OUT STD_LOGIC_VECTOR (1 downto 0)
    );
  end component;

  component servisY is
    port(
      --signal untuk clock dan reset
      CLK, RST : IN STD_LOGIC;

      -- signal untuk menentukan poin
      -- 10 : Poin untuk team X
      -- 01 : Pon untuk team Y
      POIN : IN STD_LOGIC_VECTOR (1 downto 0);

      --signal untuk menentukan posisi pemain di lapangan dan pemain mana
yang berhak melakukan servis
      POSISI, SERVY : OUT STD_LOGIC_VECTOR (1 downto 0)
    );
  end component;

  signal poin_x : std_logic;

```

```

    signal poin_y    : std_logic;

begin
    poin_x <= poin(1);
    poin_y <= poin(0);
    FSM_X   : servisX port map (CLK, RST, POIN, POSISI_X, SERVX);
    FSM_Y   : servisY port map (CLK, RST, POIN, POSISI_Y, SERVY);
    score_X : counter_1 port map (CLK, RST, poin_x, seven_segment_1_X,
seven_segment_2_X);
    score_Y : counter_1 port map (CLK, RST, poin_y, seven_segment_1_Y,
seven_segment_2_Y);

end main_arch;

```

Testbench

```

library IEEE;
use IEEE.Std_logic_1164.all;
use IEEE.Numeric_Std.all;

entity TopLevel_entity_tb is
end TopLevel_entity_tb;

architecture bench of TopLevel_entity_tb is

    component TopLevel_entity
    port (
        POIN : IN STD_LOGIC_vector (1 downto 0);
        CLK, RST : IN STD_LOGIC;
        seven_segment_1_X : out std_logic_vector (6 downto 0);
        seven_segment_2_X : out std_logic_vector (6 downto 0);
        seven_segment_1_Y : out std_logic_vector (6 downto 0);
        seven_segment_2_Y : out std_logic_vector (6 downto 0);
        POSISI_X, POSISI_Y, SERVX, SERVY : OUT STD_LOGIC_vector (1
downto 0)
    );
    end component;

    constant T           : time      := 20 ns;
    constant max_clk     : integer   := 43;
    constant count       : integer   := 0;
    signal i             : integer   := 0;

    signal POIN: STD_LOGIC_vector (1 downto 0);
    signal CLK, RST: STD_LOGIC;
    signal seven_segment_1_X: std_logic_vector (6 downto 0);
    signal seven_segment_2_X: std_logic_vector (6 downto 0);
    signal seven_segment_1_Y: std_logic_vector (6 downto 0);
    signal seven_segment_2_Y: std_logic_vector (6 downto 0);
    signal POSISI_X, POSISI_Y, SERVX, SERVY: STD_LOGIC_vector (1 downto 0)
;

begin

```

```

    uut: TopLevel_entity port map (POIN, CLK, RST, seven_segment_1_X,
seven_segment_2_X, seven_segment_1_Y, seven_segment_2_Y, POSISI_X,
POSISI_Y, SERVX, SERVY);

    clock_process: process
    begin
        if (i < max_clk) then i <= i + 1;
        CLK <= '1';
        wait for T/2;
        CLK <= '0';
        wait for T/2;
        i <= i + 1;
        else wait;
        end if;
    end process;

    RST <= '1', '0' after T/2;

stimulus: process

    begin
        POIN <= "00";
        wait for T;
        assert(POSISI_X = "01" and POSISI_Y = "01" and SERVX = "10" and
SERVY = "00")
        report "tes gagal pada testbench " & integer'image(1) severity
error;
        POIN <= "01";
        wait for T;
        assert(POSISI_X = "01" and POSISI_Y = "01" and SERVX = "10" and
SERVY = "00")
        report "tes gagal pada testbench " & integer'image(2) severity
error;
        POIN <= "00";
        wait for T;
        assert(POSISI_X = "01" and POSISI_Y = "01" and SERVX = "00" and
SERVY = "01")
        report "tes gagal pada testbench " & integer'image(3) severity
error;
        POIN <= "10";
        wait for T;
        assert(POSISI_X = "01" and POSISI_Y = "01" and SERVX = "00" and
SERVY = "01")
        report "tes gagal pada testbench " & integer'image(4) severity
error;
        POIN <= "00";
        wait for T;
        assert(POSISI_X = "01" and POSISI_Y = "01" and SERVX = "01" and
SERVY = "00")
        report "tes gagal pada testbench " & integer'image(5) severity
error;
        POIN <= "01";
        wait for T;
        assert(POSISI_X = "01" and POSISI_Y = "01" and SERVX = "01" and
SERVY = "00")
        report "tes gagal pada testbench " & integer'image(6) severity
error;
        POIN <= "00";
        wait for T;
        assert(POSISI_X = "01" and POSISI_Y = "01" and SERVX = "00" and
SERVY = "10")

```

```

    report "tes gagal pada testbench " & integer'image(7) severity
error;
    POIN <= "10";
    wait for T;
    assert(POSISI_X = "01" and POSISI_Y = "01" and SERVX = "00" and
SERVY = "10")
    report "tes gagal pada testbench " & integer'image(8) severity
error;
    POIN <= "00";
    wait for T;
    assert(POSISI_X = "01" and POSISI_Y = "01" and SERVX = "10" and
SERVY = "00")
    report "tes gagal pada testbench " & integer'image(9) severity
error;
    POIN <= "01";
    wait for T;
    assert(POSISI_X = "01" and POSISI_Y = "01" and SERVX = "10" and
SERVY = "00")
    report "tes gagal pada testbench " & integer'image(10) severity
error;
    POIN <= "00";
    wait for T;
    assert(POSISI_X = "01" and POSISI_Y = "01" and SERVX = "00" and
SERVY = "01")
    report "tes gagal pada testbench " & integer'image(11) severity
error;
    POIN <= "01";
    wait for T;
    assert(POSISI_X = "01" and POSISI_Y = "01" and SERVX = "00" and
SERVY = "01")
    report "tes gagal pada testbench " & integer'image(12) severity
error;
    POIN <= "00";
    wait for T;
    assert(POSISI_X = "01" and POSISI_Y = "10" and SERVX = "00" and
SERVY = "01")
    report "tes gagal pada testbench " & integer'image(13) severity
error;
    POIN <= "01";
    wait for T;
    assert(POSISI_X = "01" and POSISI_Y = "10" and SERVX = "00" and
SERVY = "01")
    report "tes gagal pada testbench " & integer'image(14) severity
error;
    POIN <= "00";
    wait for T;
    assert(POSISI_X = "01" and POSISI_Y = "01" and SERVX = "00" and
SERVY = "01")
    report "tes gagal pada testbench " & integer'image(15) severity
error;
    POIN <= "01";
    wait for T;
    assert(POSISI_X = "01" and POSISI_Y = "01" and SERVX = "00" and
SERVY = "01")
    report "tes gagal pada testbench " & integer'image(16) severity
error;
    POIN <= "00";
    wait for T;
    assert(POSISI_X = "01" and POSISI_Y = "10" and SERVX = "00" and
SERVY = "01")

```

```

    report "tes gagal pada testbench " & integer'image(17) severity
error;
    POIN <= "10";
    wait for T;
    assert(POSISI_X = "01" and POSISI_Y = "10" and SERVX = "00" and
SERVY = "01")
    report "tes gagal pada testbench " & integer'image(18) severity
error;
    POIN <= "00";
    wait for T;
    assert(POSISI_X = "01" and POSISI_Y = "10" and SERVX = "01" and
SERVY = "00")
    report "tes gagal pada testbench " & integer'image(19) severity
error;
    POIN <= "01";
    wait for T;
    assert(POSISI_X = "01" and POSISI_Y = "10" and SERVX = "01" and
SERVY = "00")
    report "tes gagal pada testbench " & integer'image(20) severity
error;
    POIN <= "00";
    wait for T;
    assert(POSISI_X = "01" and POSISI_Y = "10" and SERVX = "00" and
SERVY = "10")
    report "tes gagal pada testbench " & integer'image(21) severity
error;
    POIN <= "10";
    wait for T;
    assert(POSISI_X = "01" and POSISI_Y = "10" and SERVX = "00" and
SERVY = "10")
    report "tes gagal pada testbench " & integer'image(22) severity
error;
    POIN <= "00";
    wait for T;
    assert(POSISI_X = "01" and POSISI_Y = "10" and SERVX = "10" and
SERVY = "00")
    report "tes gagal pada testbench " & integer'image(23) severity
error;
    POIN <= "10";
    wait for T;
    assert(POSISI_X = "01" and POSISI_Y = "10" and SERVX = "10" and
SERVY = "00")
    report "tes gagal pada testbench " & integer'image(24) severity
error;
    POIN <= "00";
    wait for T;
    assert(POSISI_X = "10" and POSISI_Y = "10" and SERVX = "10" and
SERVY = "00")
    report "tes gagal pada testbench " & integer'image(25) severity
error;
    POIN <= "10";
    wait for T;
    assert(POSISI_X = "10" and POSISI_Y = "10" and SERVX = "10" and
SERVY = "00")
    report "tes gagal pada testbench " & integer'image(26) severity
error;
    POIN <= "00";
    wait for T;
    assert(POSISI_X = "01" and POSISI_Y = "10" and SERVX = "10" and
SERVY = "00")

```

```

    report "tes gagal pada testbench " & integer'image(27) severity
error;
    POIN <= "10";
    wait for T;
    assert(POSISI_X = "01" and POSISI_Y = "10" and SERVX = "10" and
SERVY = "00")
    report "tes gagal pada testbench " & integer'image(28) severity
error;
    POIN <= "00";
    wait for T;
    assert(POSISI_X = "10" and POSISI_Y = "10" and SERVX = "10" and
SERVY = "00")
    report "tes gagal pada testbench " & integer'image(29) severity
error;
    POIN <= "01";
    wait for T;
    assert(POSISI_X = "10" and POSISI_Y = "10" and SERVX = "10" and
SERVY = "00")
    report "tes gagal pada testbench " & integer'image(30) severity
error;
    POIN <= "00";
    wait for T;
    assert(POSISI_X = "10" and POSISI_Y = "10" and SERVX = "00" and
SERVY = "01")
    report "tes gagal pada testbench " & integer'image(31) severity
error;
    POIN <= "01";
    wait for T;
    assert(POSISI_X = "10" and POSISI_Y = "10" and SERVX = "00" and
SERVY = "01")
    report "tes gagal pada testbench " & integer'image(32) severity
error;
    POIN <= "00";
    wait for T;
    assert(POSISI_X = "10" and POSISI_Y = "01" and SERVX = "00" and
SERVY = "01")
    report "tes gagal pada testbench " & integer'image(33) severity
error;
    POIN <= "10";
    wait for T;
    assert(POSISI_X = "10" and POSISI_Y = "01" and SERVX = "00" and
SERVY = "01")
    report "tes gagal pada testbench " & integer'image(34) severity
error;
    POIN <= "00";
    wait for T;
    assert(POSISI_X = "10" and POSISI_Y = "01" and SERVX = "01" and
SERVY = "00")
    report "tes gagal pada testbench " & integer'image(35) severity
error;
    POIN <= "10";
    wait for T;
    assert(POSISI_X = "10" and POSISI_Y = "01" and SERVX = "01" and
SERVY = "00")
    report "tes gagal pada testbench " & integer'image(36) severity
error;
    POIN <= "00";
    wait for T;
    assert(POSISI_X = "01" and POSISI_Y = "01" and SERVX = "01" and
SERVY = "00")

```

```

    report "tes gagal pada testbench " & integer'image(37) severity
error;
    POIN <= "10";
    wait for T;
    assert(POSISI_X = "01" and POSISI_Y = "01" and SERVX = "01" and
SERVY = "00")
    report "tes gagal pada testbench " & integer'image(38) severity
error;
    POIN <= "00";
    wait for T;
    assert(POSISI_X = "10" and POSISI_Y = "01" and SERVX = "01" and
SERVY = "00")
    report "tes gagal pada testbench " & integer'image(39) severity
error;
    POIN <= "01";
    wait for T;
    assert(POSISI_X = "10" and POSISI_Y = "01" and SERVX = "01" and
SERVY = "00")
    report "tes gagal pada testbench " & integer'image(40) severity
error;
    POIN <= "00";
    wait for T;
    assert(POSISI_X = "10" and POSISI_Y = "01" and SERVX = "00" and
SERVY = "10")
    report "tes gagal pada testbench " & integer'image(41) severity
error;
    POIN <= "01";
    wait for T;
    assert(POSISI_X = "10" and POSISI_Y = "01" and SERVX = "00" and
SERVY = "10")
    report "tes gagal pada testbench " & integer'image(42) severity
error;
    POIN <= "00";
    wait for T;
    assert(POSISI_X = "10" and POSISI_Y = "10" and SERVX = "00" and
SERVY = "10")
    report "tes gagal pada testbench " & integer'image(43) severity
error;

    wait;
end process;

end bench;

```