Mini-Project

COMPUTER ASSIGNMENT

Computer Architecture

PROPOSED TO: Mr. Abbasi

2023

SUBJECT TITLE

Public parking controller

Team Members:

- Navid P.Panahi
- Arsham Masoodi
- Amir Hossein Hanifi

Contents

Over View:

In this project we want to design a controller for a public parking.

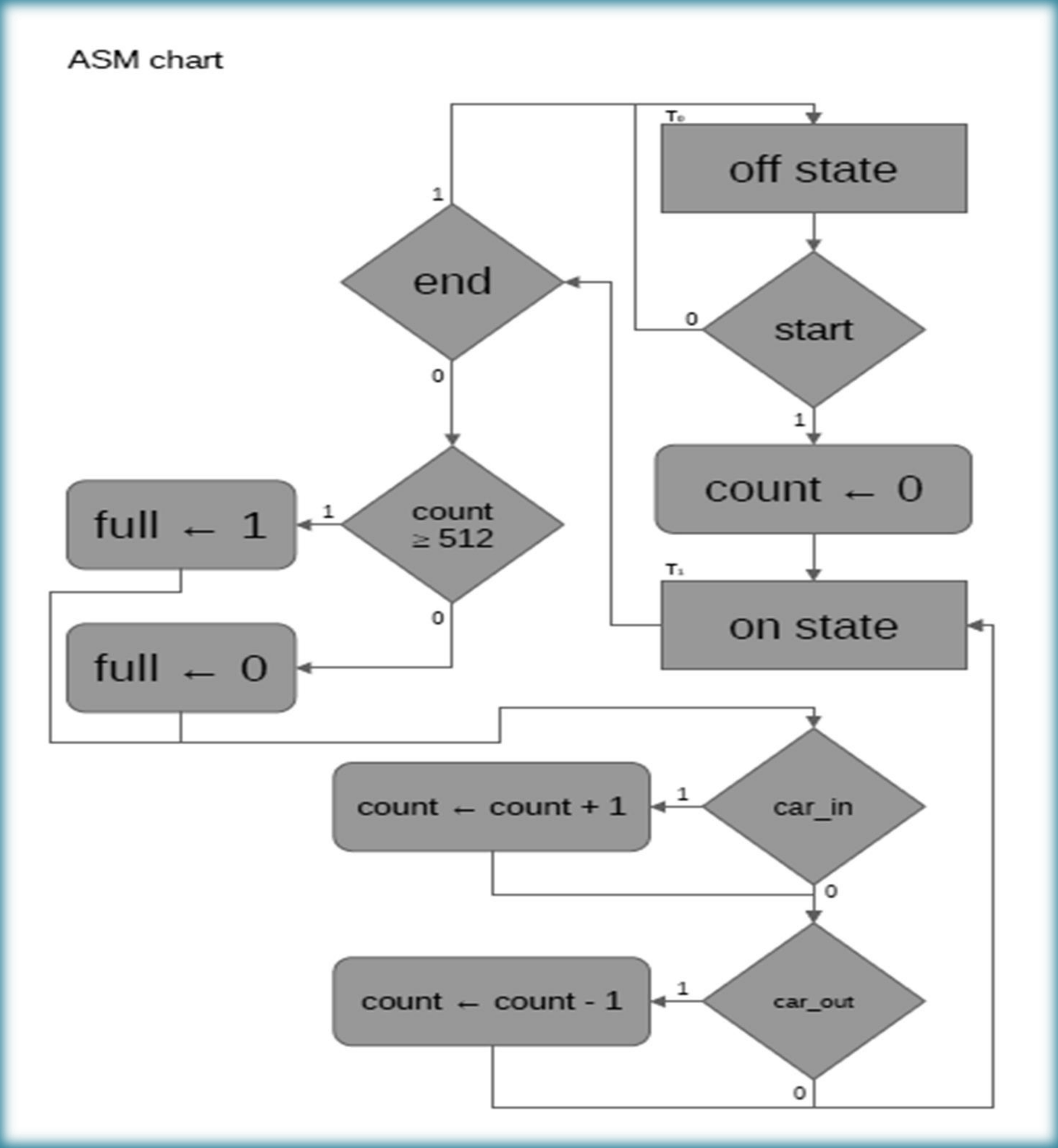
This parking must have at most 512 capacity

a kind of counter to count how many cars are in, this job most done by a sensor on the top of the enter door, that will send a 1 to your board to make them a car is added, also we have the same sensor on the top of the exit door.

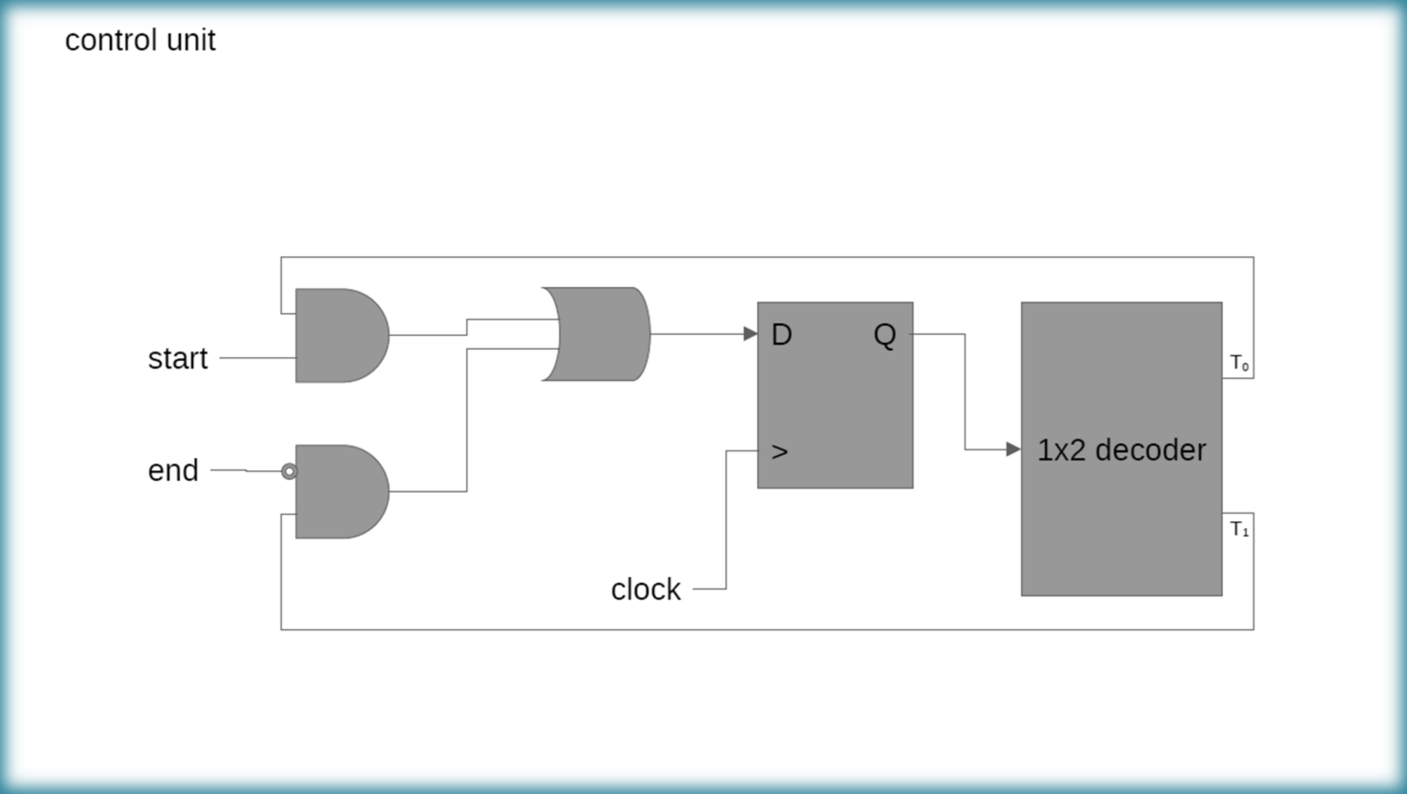
In the design part you have to design a counter which count from down to up and reverse mode.

ASM Chart:

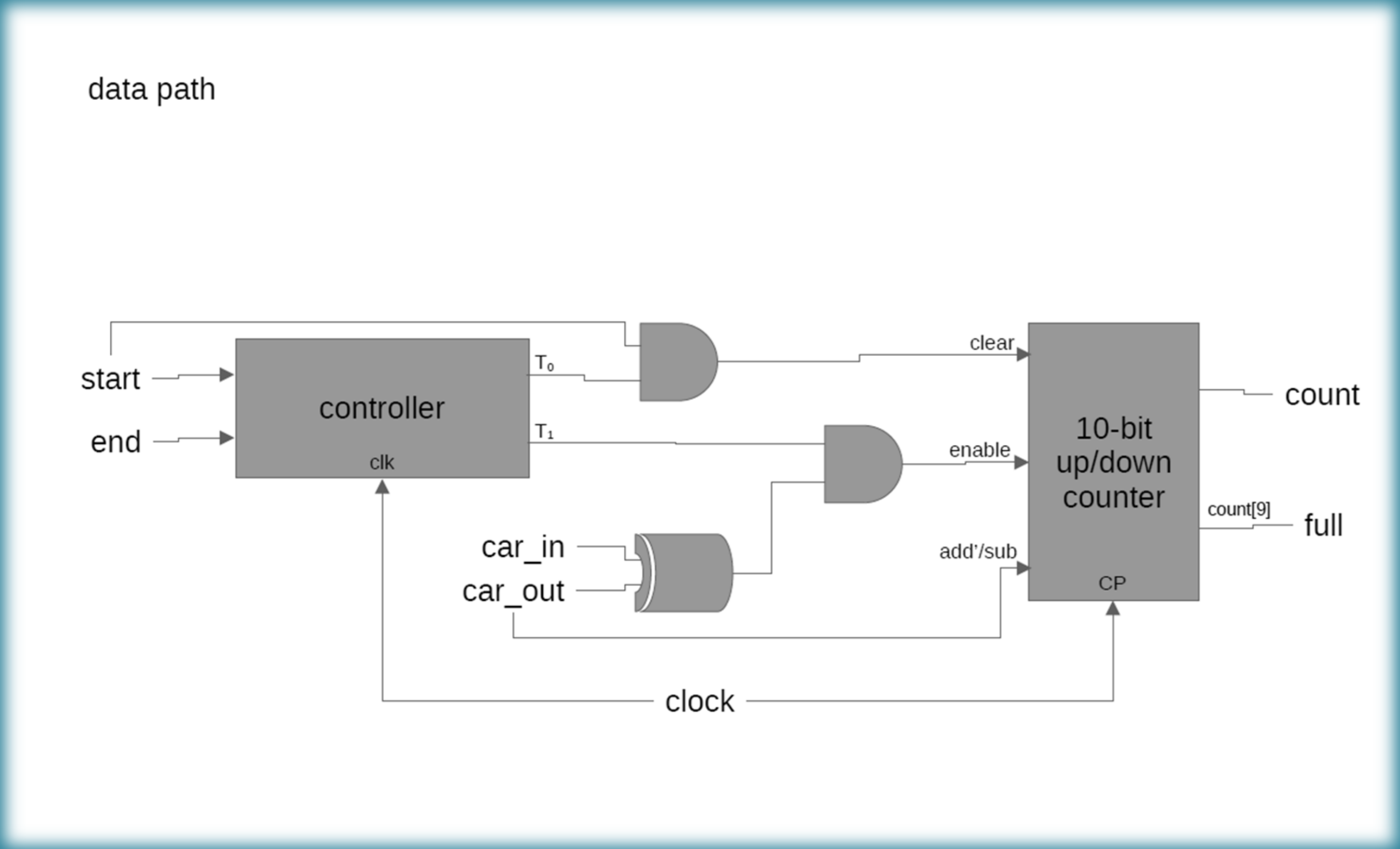
ASM:



Control Unit:



Data_path:


data path
start
end
controller
clk
T0
T1
clear
enable
car_in
car_out
add'/sub
10-bit up/down counter
CP
count
count[9]
full
clock

RTLs:

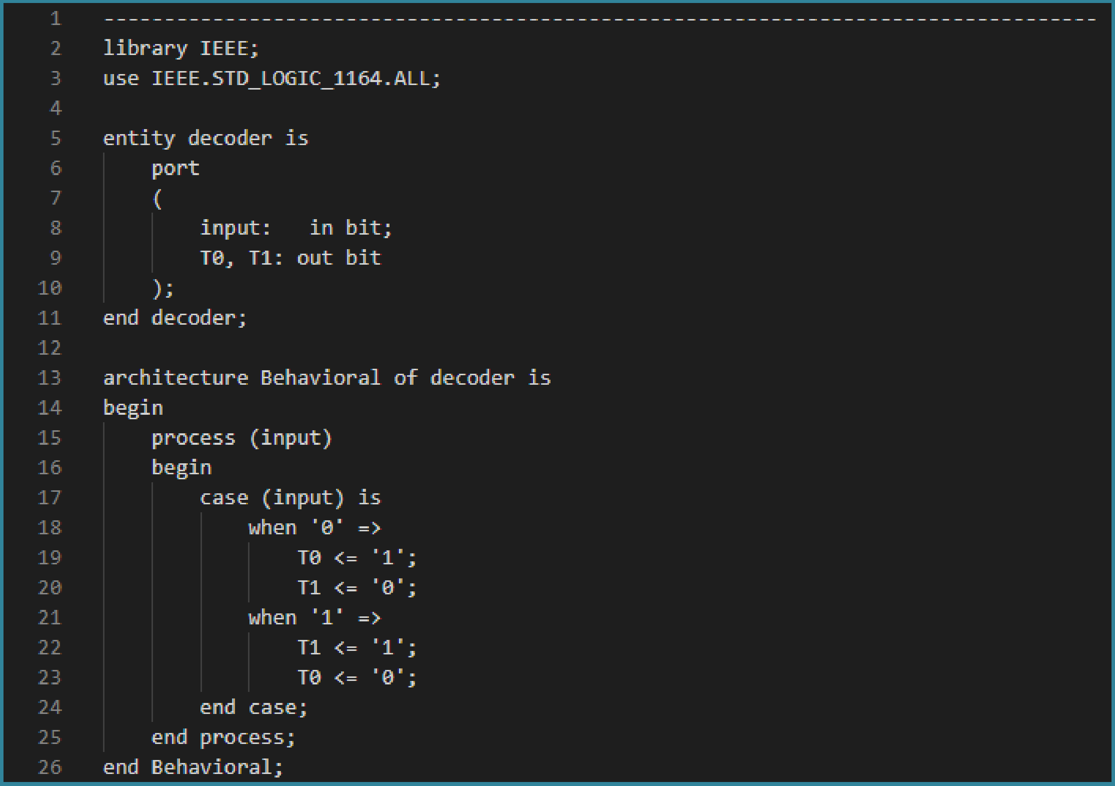
```
conditional RTLs
        start       : count ← 0
        car_in      : count ← count + 1
        car_out     : count ← count - 1
        count >= 512: full ← 1
```

VHDL Codes:

ASM Chart: Click Here

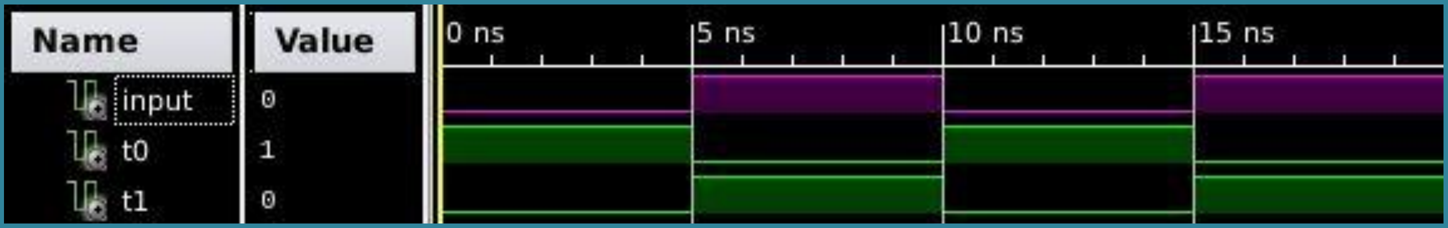
VHDL Code:

Decoder.vhdl: Click Here for test bench

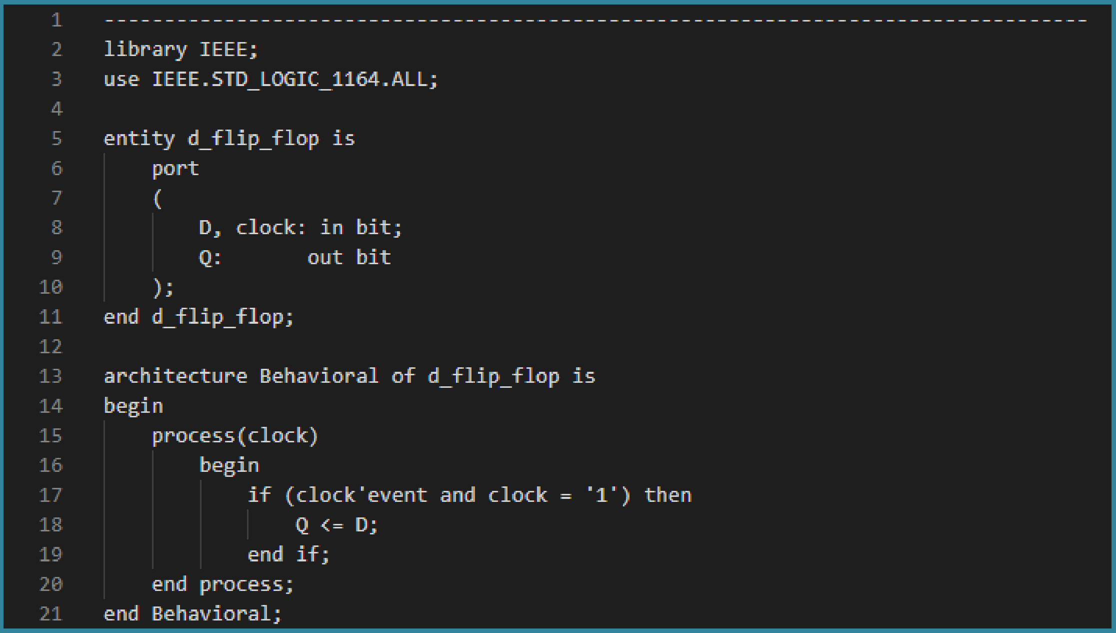


```vhdl
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity decoder is
    port
    (
        input:   in bit;
        T0, T1: out bit
    );
end decoder;

architecture Behavioral of decoder is
begin
    process (input)
    begin
        case (input) is
            when '0' =>
                T0 <= '1';
                T1 <= '0';
            when '1' =>
                T1 <= '1';
                T0 <= '0';
        end case;
    end process;
end Behavioral;
```

WaveLength:

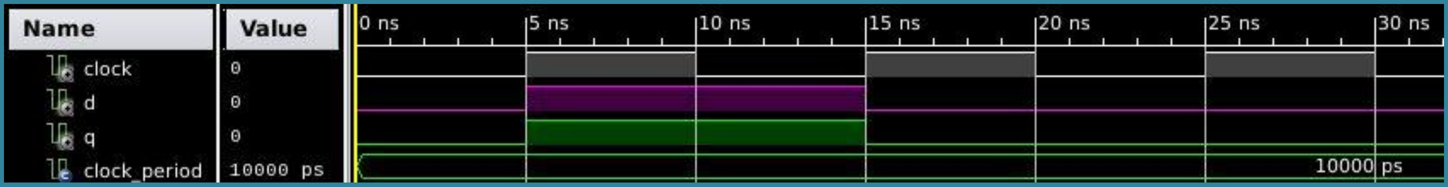


D_Flip Flop.vhdl: [Click Here for test bench]

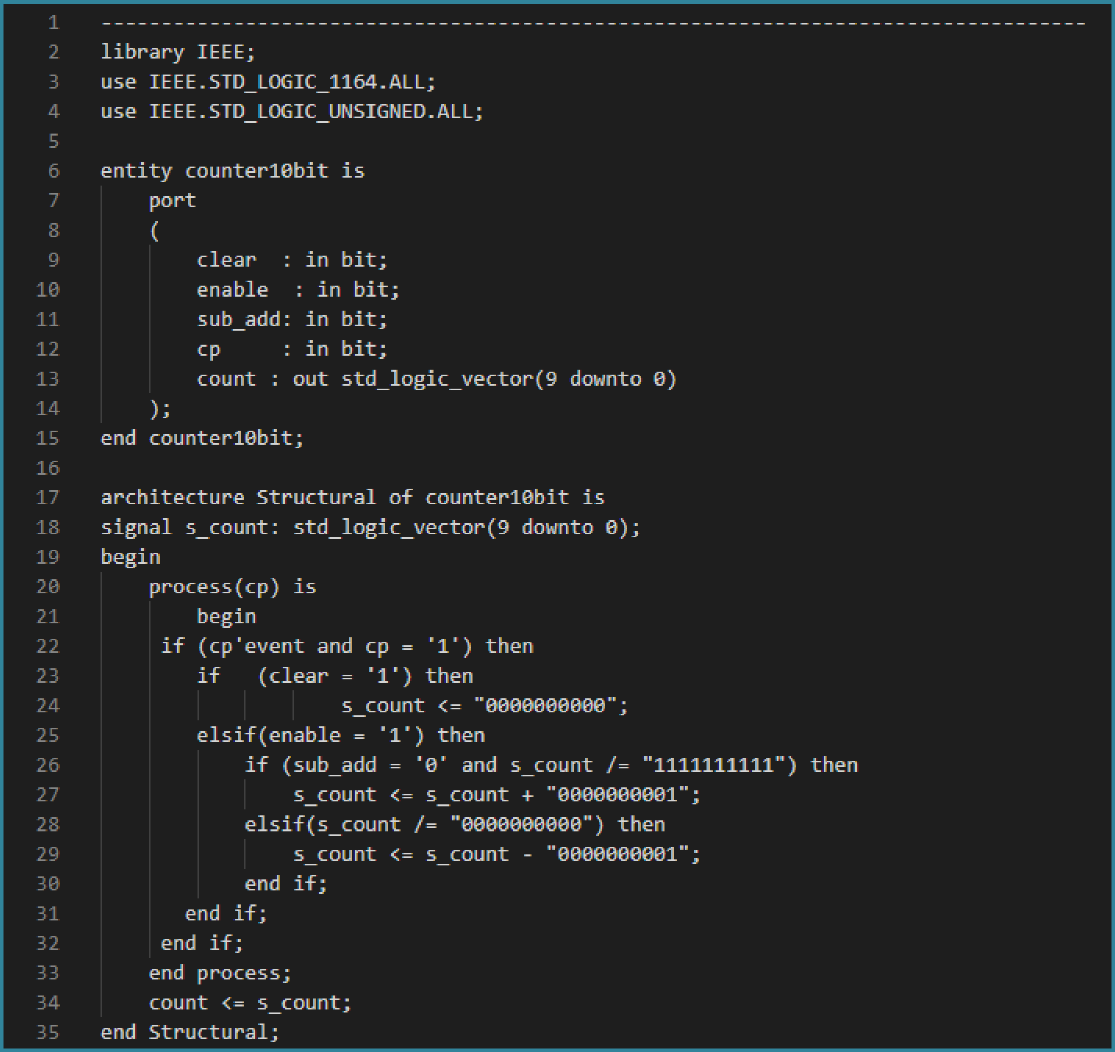


```
--------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity d_flip_flop is
    port
    (
        D, clock: in bit;
        Q:        out bit
    );
end d_flip_flop;

architecture Behavioral of d_flip_flop is
begin
    process(clock)
        begin
            if (clock'event and clock = '1') then
                Q <= D;
            end if;
    end process;
end Behavioral;
```

WaveLength:

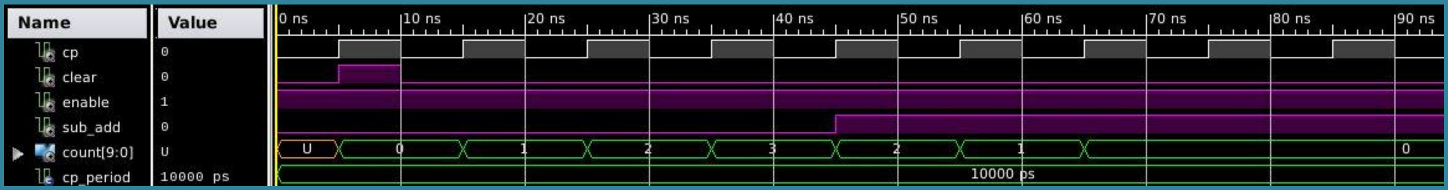


Counter10bit.vhdl: [Click Here for test bench]

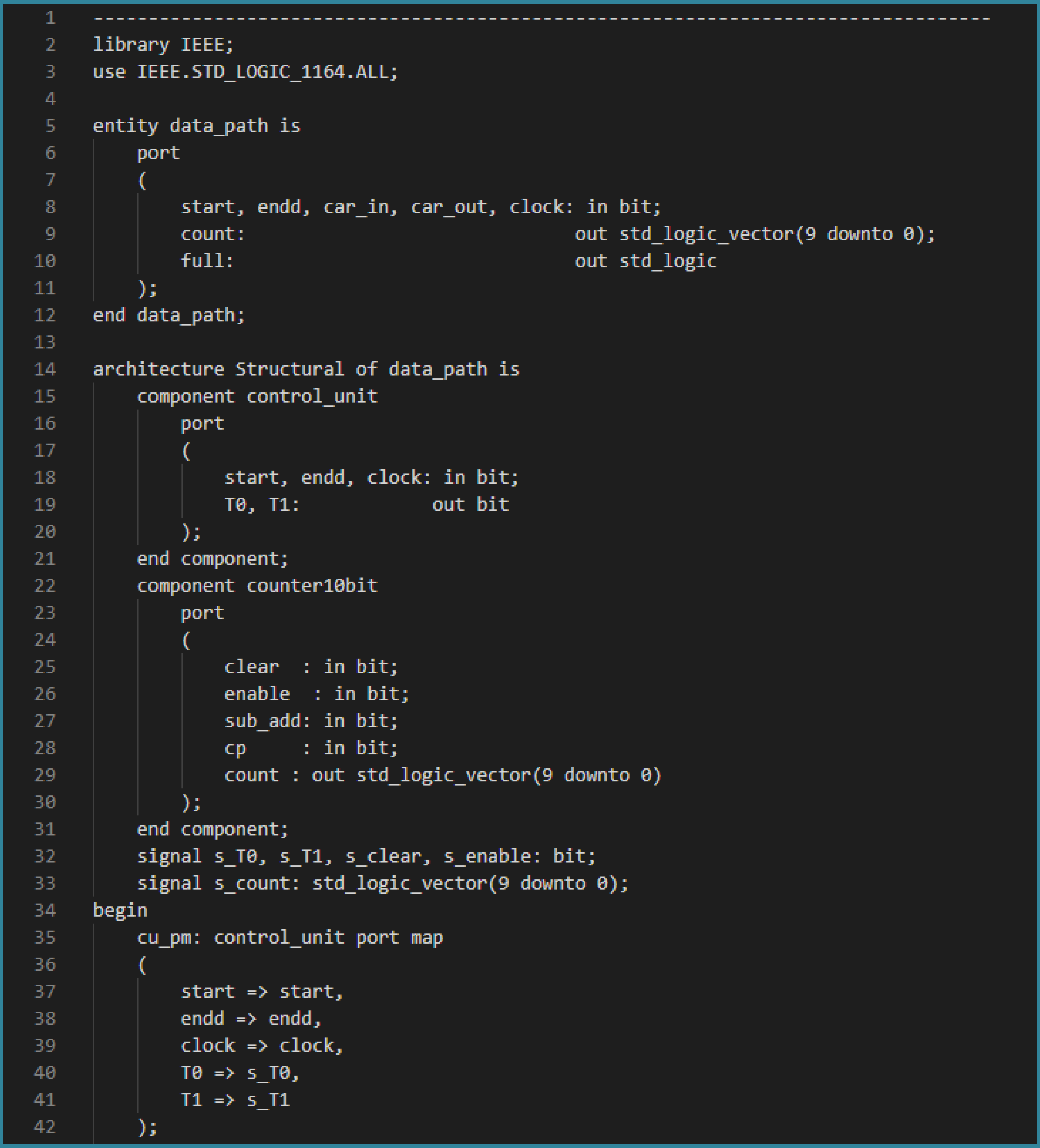


```vhdl
---------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity counter10bit is
    port
    (
        clear   : in bit;
        enable  : in bit;
        sub_add: in bit;
        cp      : in bit;
        count : out std_logic_vector(9 downto 0)
    );
end counter10bit;

architecture Structural of counter10bit is
signal s_count: std_logic_vector(9 downto 0);
begin
    process(cp) is
        begin
     if (cp'event and cp = '1') then
        if   (clear = '1') then
                    s_count <= "0000000000";
        elsif(enable = '1') then
            if (sub_add = '0' and s_count /= "1111111111") then
                s_count <= s_count + "0000000001";
            elsif(s_count /= "0000000000") then
                s_count <= s_count - "0000000001";
            end if;
       end if;
     end if;
    end process;
    count <= s_count;
end Structural;
```

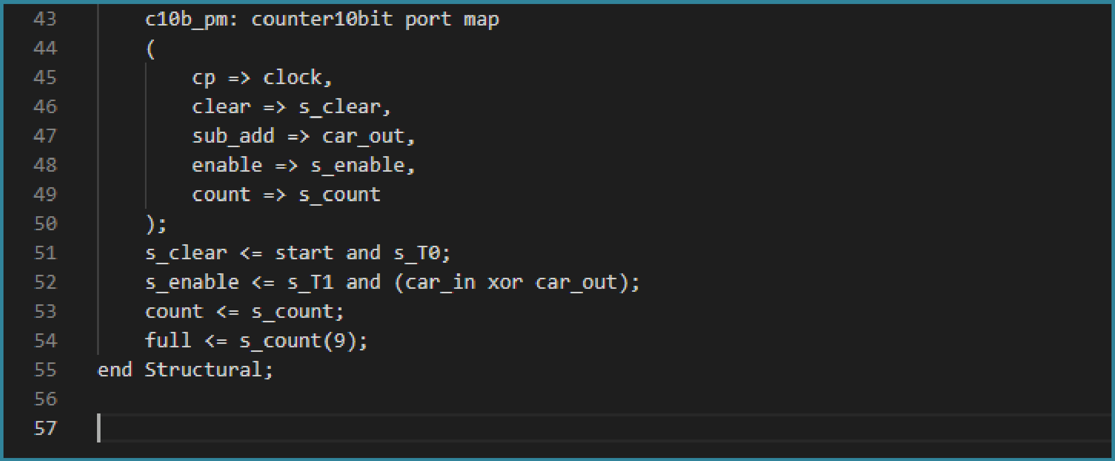
WaveLength:



DataPath.vhdl: Click Here for test bench

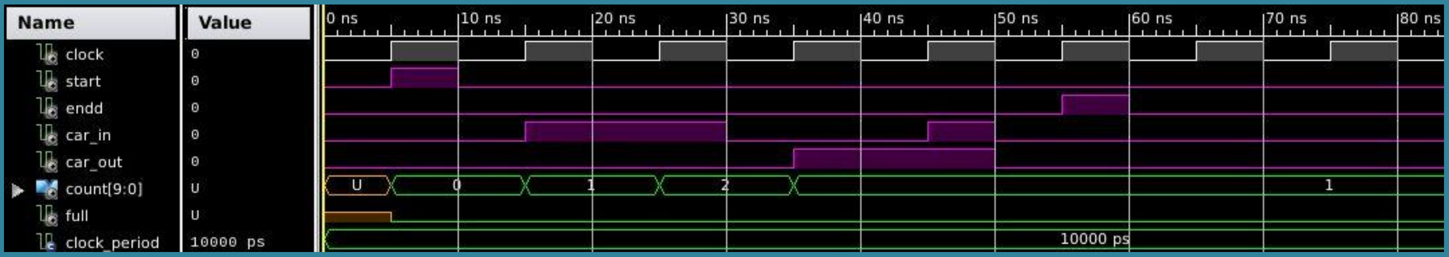


```vhdl
--------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity data_path is
    port
    (
        start, endd, car_in, car_out, clock: in bit;
        count:                              out std_logic_vector(9 downto 0);
        full:                               out std_logic
    );
end data_path;

architecture Structural of data_path is
    component control_unit
        port
        (
            start, endd, clock: in bit;
            T0, T1:            out bit
        );
    end component;
    component counter10bit
        port
        (
            clear  : in bit;
            enable  : in bit;
            sub_add: in bit;
            cp     : in bit;
            count : out std_logic_vector(9 downto 0)
        );
    end component;
    signal s_T0, s_T1, s_clear, s_enable: bit;
    signal s_count: std_logic_vector(9 downto 0);
begin
    cu_pm: control_unit port map
    (
        start => start,
        endd => endd,
        clock => clock,
        T0 => s_T0,
        T1 => s_T1
    );
```

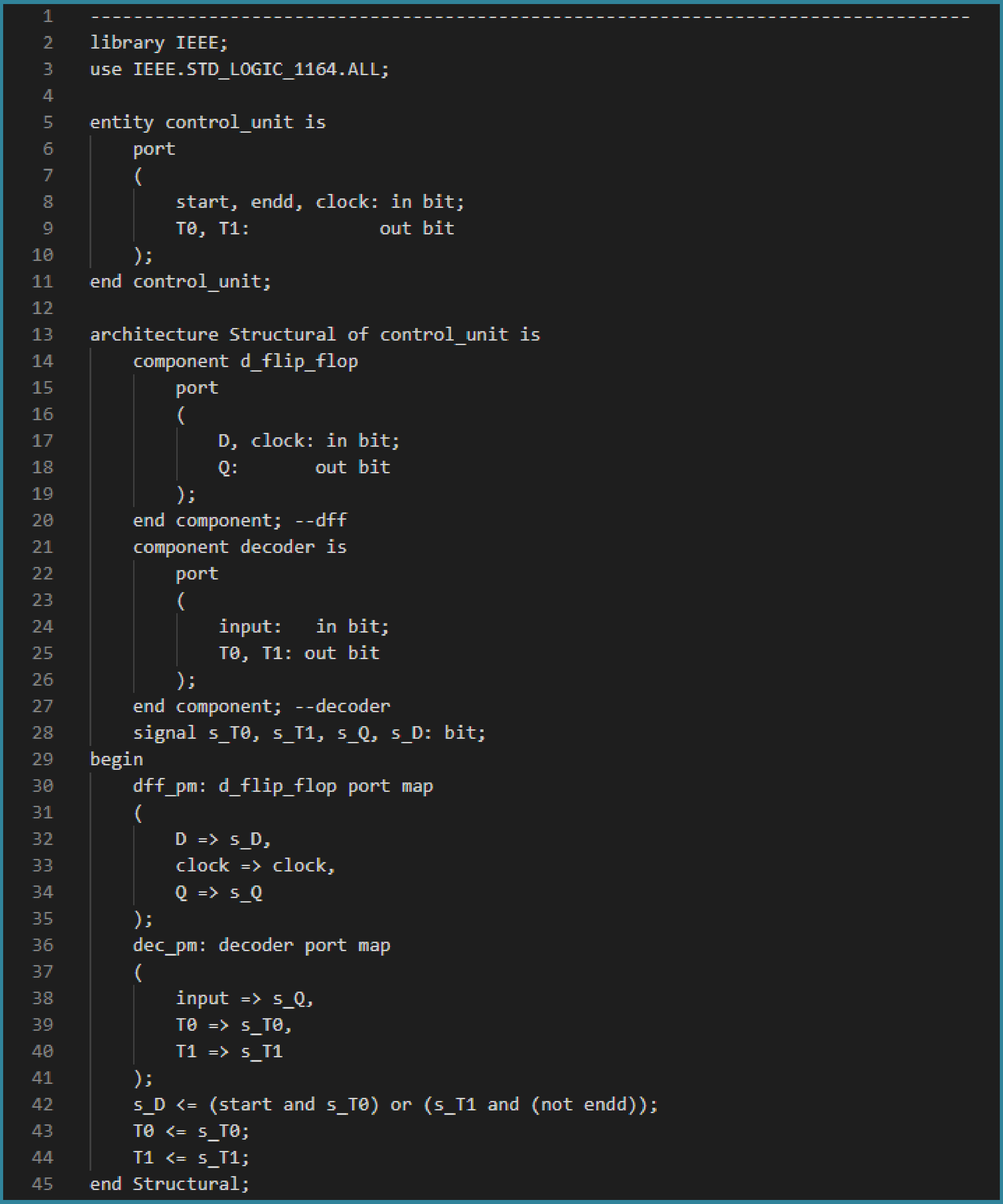


```vhdl
    c10b_pm: counter10bit port map
    (
        cp => clock,
        clear => s_clear,
        sub_add => car_out,
        enable => s_enable,
        count => s_count
    );
    s_clear <= start and s_T0;
    s_enable <= s_T1 and (car_in xor car_out);
    count <= s_count;
    full <= s_count(9);
end Structural;

```

WaveLength:

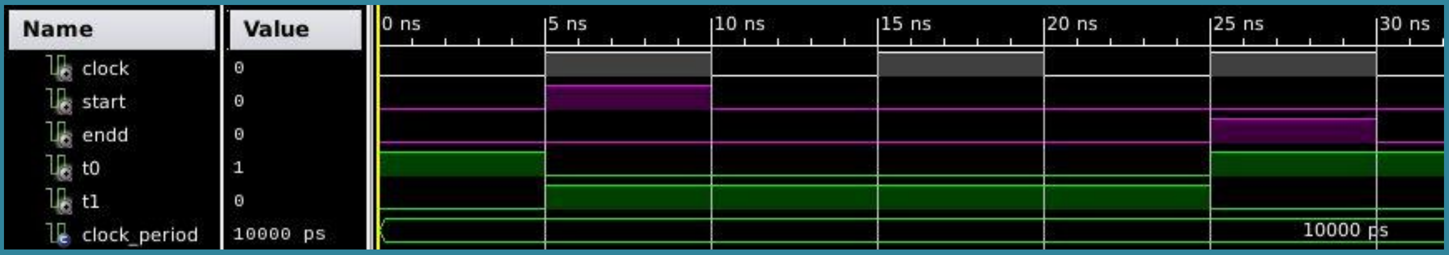

Name
Value
clock 0
start 0
endd 0
car_in 0
car_out 0
count[9:0] U
full U
clock_period 10000 ps
0 ns
10 ns
20 ns
30 ns
40 ns
50 ns
60 ns
70 ns
80 ns
U
0
1
2
1
10000 ps

Control_Unit.vhdl: Click Here for test bench



```
-----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity control_unit is
    port
    (
        start, endd, clock: in bit;
        T0, T1:            out bit
    );
end control_unit;

architecture Structural of control_unit is
    component d_flip_flop
        port
        (
            D, clock: in bit;
            Q:        out bit
        );
    end component; --dff
    component decoder is
        port
        (
            input:   in bit;
            T0, T1: out bit
        );
    end component; --decoder
    signal s_T0, s_T1, s_Q, s_D: bit;
begin
    dff_pm: d_flip_flop port map
    (
        D => s_D,
        clock => clock,
        Q => s_Q
    );
    dec_pm: decoder port map
    (
        input => s_Q,
        T0 => s_T0,
        T1 => s_T1
    );
    s_D <= (start and s_T0) or (s_T1 and (not endd));
    T0 <= s_T0;
    T1 <= s_T1;
end Structural;
```

WaveLength:



```
clock          0
start          0
endd           0
t0             1
t1             0
clock_period   10000 ps
```

Thank you all for your attention 😊

BasuArch