BAB I

FREQUENCY DIVIDER

1. Tujuan

Praktikan diharapkan mengerti proses dari frequency divider sebagai pemanfaatan fungsi delay pada suatu system.

1. Dasar Teori

Frequency divider merupakan suatu proses dimana membagi frekuensi yang ada seperti dari 50MHz menjadi 200Hz. Konsep ini sangat diperlukan pada pembuatan system berbasis FPGA karena landasan dasar berupa rangkaian logika. Pada dasarnya konsep ini hampir sama dengan library delay yang digunakan oleh mikrokontroller karena berpedoman pada krystal yang terpasang pada minimum system. Hal ini juga berlaku pada system yang berbasis FPGA yang nantinya dapat digunakan sebagai delay pada pemrograman berbasis FPGA. Apabila diasumsikan bahwa system memakai penggunaan waktu selama 5 ms, berikut contoh perhitungannya :

Didapat nilai yang mewakili 5 ms yaitu 124.999 sehingga nilai tersebut dapat digunakan pada system VHDL. Data tersebut dapat digunakan juga sebagai master dari frekuensi divider yang dipakai untuk beberapa system secara berkelanjutan. Karena pada dasarnya frekuensi divider selalu dipakai dalam suatu system yang memerlukan suatu tindakan terhadap waktu.

1. Bahan yang diperlukan

* Laptop / CPU
* Software Xilinx (min versi 13.2)
* Board FPGA Cmod S6
* Data Ucf FPGA Cmod S6
* Extended Board

1. List Program
2. File VHD

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_unsigned.ALL;

use IEEE.STD\_LOGIC\_arith.ALL;

entity frequency\_divider is

Port ( clk : in STD\_LOGIC;

rst : in STD\_LOGIC;

data\_out : out STD\_LOGIC);

end frequency\_divider;

architecture Behavioral of frequency\_divider is

signal temporal: std\_logic;

signal counter : integer range 0 to 124999 :=0;

begin

clock\_divider: process (rst, clk)

begin

if (rst='1') then

temporal <= '0';

counter <= 0;

elsif rising\_edge(clk) then

if (counter = 124999) then

temporal <= not (temporal);

counter <= 0;

else

counter <= counter + 1;

end if;

end if;

end process;

data\_out <= temporal;

end Behavioral;

BAB II

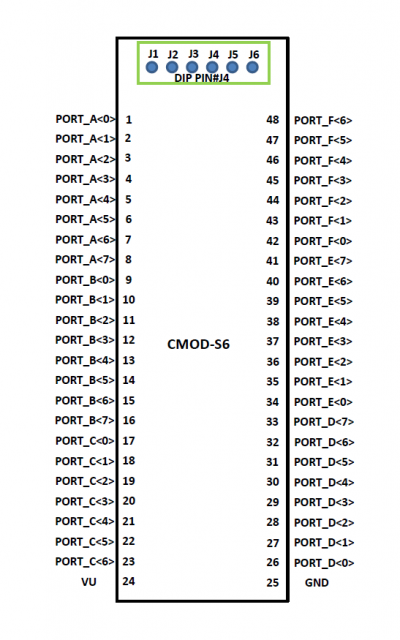
7 SEGMENT

1. Tujuan

Peserta praktikum diharapkan dapat menggunakan I/O sebagai sarana penghubung terhadap penggunaan interface lain seperti rangkaian driver.

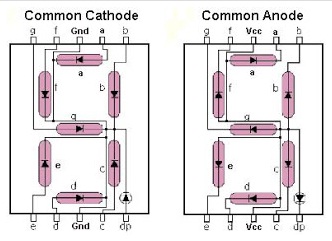
1. Dasar Teori

Penggunaan I/O sangatlah penting dalam menggunakan suatu perangkat keras dari luar seperti IC tambahan, driver motor, dll. Pada FPGA penggunaan I/O secara global sama saja hanya yang membedakan alamat akses I/O karena walau sama-sama jenis FPGA Spartan tetapi jenis Family yang berbeda tetapi secara penggunaan memiliki kesamaan. Berikut I/O pada perangkat FPGA Spartan 6 :



Gambar 2.1 Port I/O Spartan 6

Seven segmen adalah salah satu [perangkat layar](http://id.wikipedia.org/w/index.php?title=Perangkat_layar&action=edit&redlink=1) untuk menampilkan [sistem angka](http://id.wikipedia.org/wiki/Sistem_angka) [desimal](http://id.wikipedia.org/wiki/Desimal) yang merupakan alternatif dari layar *dot-matrix*. Layar tujuh segmen ini seringkali digunakan pada [jam digital](http://id.wikipedia.org/wiki/Jam_digital), meteran elektronik, dan perangkat elektronik lainnya yang menampilkan informasi numerik. Seven segment sendiri memiliki dua jenis yang berbeda yaitu *common* *catoda* dan *common anoda*.



Gambar 2.2 Seven Segment

1. Bahan yang diperlukan

* Laptop / CPU
* Software Xilinx (min versi 13.2)
* Board FPGA Spartan 3E
* Data Ucf FPGA Spartan 3E (Kit & Papilio one 500k)
* Interface 7 segment & ic driver

1. List Program
2. File VHD

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity seven\_segment is

port (

clk : in std\_logic;

load : in std\_logic;

trans : out std\_logic;

bcd : in std\_logic\_vector(3 downto 0);

seg : out std\_logic\_vector(6 downto 0)

);

end seven\_segment;

architecture Behavioral of seven\_segment is

begin

process (clk,bcd)

begin

if (clk'event and clk='1') then

if (load='1')then

trans <= '1';

case bcd is

when "0000"=> seg <="0111111"; -- '0'

when "0001"=> seg <="0000110"; -- '1'

when "0010"=> seg <="1011011"; -- '2'

when "0011"=> seg <="1001111"; -- '3'

when "0100"=> seg <="1100110"; -- '4'

when "0101"=> seg <="1101101"; -- '5'

when "0110"=> seg <="1111101"; -- '6'

when "0111"=> seg <="0000111"; -- '7'

when "1000"=> seg <="1111111"; -- '8'

when "1001"=> seg <="1101111"; -- '9'

when "1010"=> seg <="1110111"; -- 'A'

when "1011"=> seg <="1111100"; -- 'B'

when "1100"=> seg <="1011000"; -- 'C'

when "1101"=> seg <="1011110"; -- 'D'

when "1110"=> seg <="1111001"; -- 'E'

when others=> seg <="1110001"; -- 'F'

end case;

end if;

end if;

end process;

end Behavioral;

BAB III

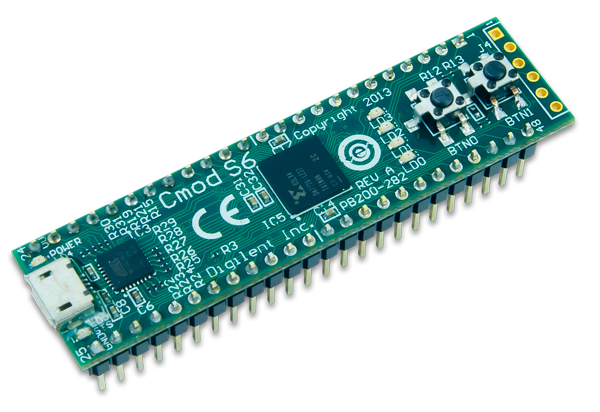
DRIVING 7-SEGMENT

1. Tujuan

Peserta praktikum diharapkan dapat memanfaatkan frekuensi divider dan menggunakan behavioral architecture pada driving menggunakan 7 segment.

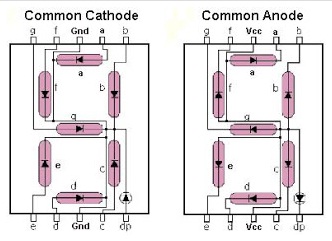
1. Dasar Teori

Penggunaan I/O sangatlah penting dalam menggunakan suatu perangkat keras dari luar seperti IC tambahan, driver motor, dll. Pada FPGA penggunaan I/O secara global sama saja hanya yang membedakan alamat akses I/O karena walau sama-sama jenis FPGA Spartan tetapi jenis Family yang berbeda tetapi secara penggunaan memiliki kesamaan. Berikut I/O pada perangkat FPGA Spartan 6 :



Gambar 3.1 Modul FPGA SPARTAN 6

Seven segmen adalah salah satu [perangkat layar](http://id.wikipedia.org/w/index.php?title=Perangkat_layar&action=edit&redlink=1) untuk menampilkan [sistem angka](http://id.wikipedia.org/wiki/Sistem_angka) [desimal](http://id.wikipedia.org/wiki/Desimal) yang merupakan alternatif dari layar *dot-matrix*. Layar tujuh segmen ini seringkali digunakan pada [jam digital](http://id.wikipedia.org/wiki/Jam_digital), meteran elektronik, dan perangkat elektronik lainnya yang menampilkan informasi numerik. Seven segment sendiri memiliki dua jenis yang berbeda yaitu *common* *catoda* dan *common anoda*.



Gambar 2.2 Seven Segment

1. Bahan yang diperlukan

* Laptop / CPU
* Software Xilinx (min versi 13.2)
* Board FPGA Spartan 3E
* Data Ucf FPGA Spartan 3E (Kit & Papilio one 500k)
* Interface 7 segment & ic driver

1. List Program

a). File VHD

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity driv\_seven\_segment is

PORT( clk : IN STD\_LOGIC;

load : IN STD\_LOGIC;

trans : out STD\_LOGIC;

rst : IN STD\_LOGIC;

seg : OUT STD\_LOGIC\_VECTOR(7 DOWNTO 1)

);

end driv\_seven\_segment;

architecture Behavioral of driv\_seven\_segment is

CONSTANT div : INTEGER :=249999;

SIGNAL clk\_div : INTEGER RANGE 0 TO div := 0;

SIGNAL temp : INTEGER RANGE 0 TO 9 := 0;

begin

PROCESS(clk)

BEGIN

IF RISING\_EDGE(clk) THEN

IF rst = '1' THEN

clk\_div <= 0;

temp <= 0;

ELSE

IF load = '1' THEN

trans <= '1';

clk\_div <= clk\_div + 1;

IF clk\_div = div THEN

clk\_div <= 0;

temp <= temp +1;

IF temp = 9 THEN

temp <= 0;

END IF;

END IF;

END IF;

END IF;

END IF;

END PROCESS;

seg <= "0111111" WHEN temp = 0 ELSE

"0000110" WHEN temp = 1 ELSE

"1011011" WHEN temp = 2 ELSE

"1001111" WHEN temp = 3 ELSE

"1100110" WHEN temp = 4 ELSE

"1101101" WHEN temp = 5 ELSE

"1111101" WHEN temp = 6 ELSE

"0000111" WHEN temp = 7 ELSE

"1111111" WHEN temp = 8 ELSE

"1101111";

end Behavioral;

BAB IV

PROM / FLASH

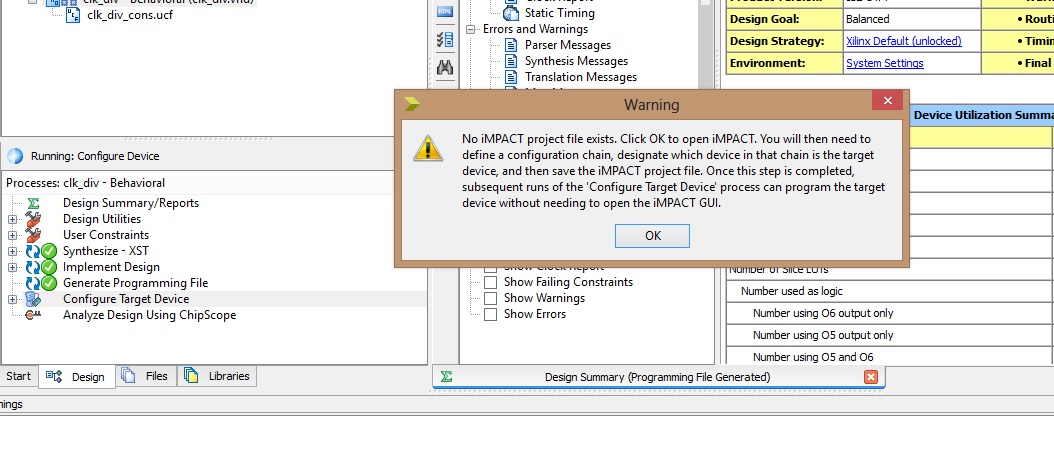
1. Peserta praktikum diharapkan dapat menggunakan PROM/FLASH yang terdapat pada FPGA dan dapat diaplikasikan pada suatu system.
2. DASAR TEORI

PROM/FLASH pada FPGA digunakan perangkat elektronik digital untuk menyimpan data permanen. ROM standard biasa nya diprogram saat di pabrik, sedangkan PROM diprogram setelah terbuat di pabrik.

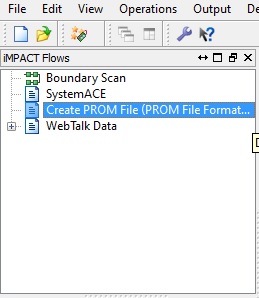
1. BAHAN YANG DIPERLUKAN

* Laptop / CPU
* Software Xilinx (min versi 13.2)
* Board FPGA Spartan 6
* Data Ucf FPGA Spartan 6
* Interface motor stepper dan driver uln2003

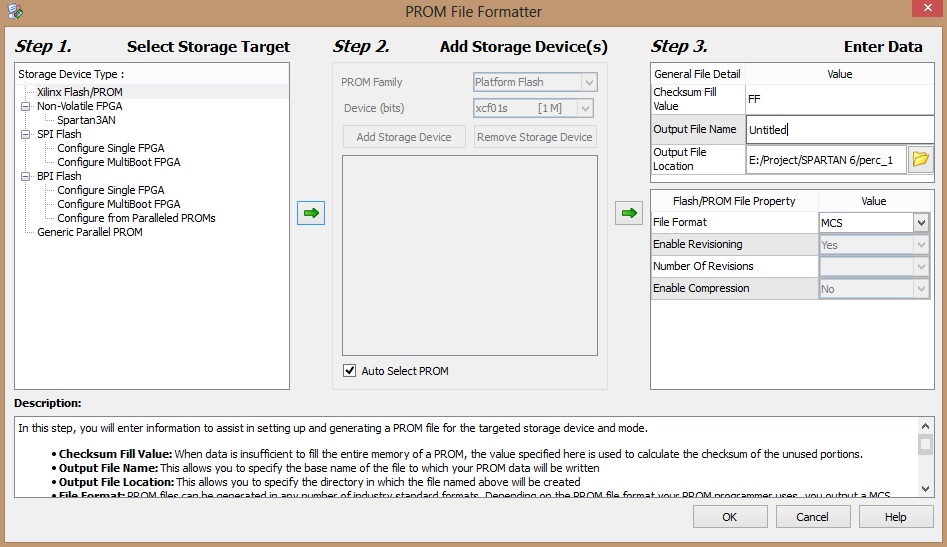
1. LANGKAH
2. Compile semua program lalu masuk ke *Configure Target Device* lalu klik OK



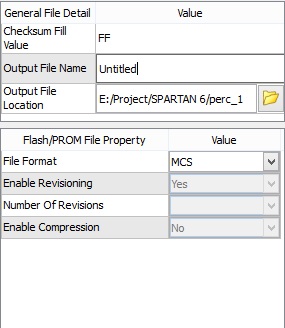
1. Maka akan muncul tampilan iMPACT lalu pilih *CREATE PROM FILE*



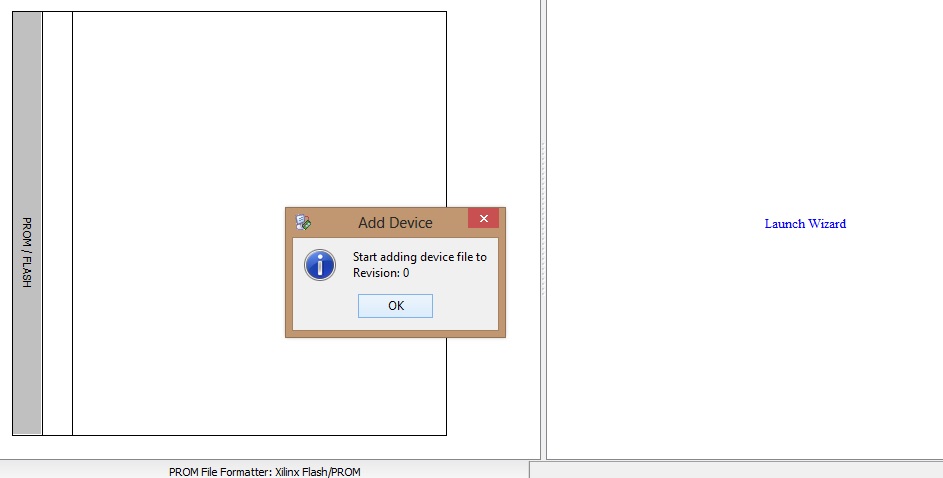
1. Lalu akan muncul kotak dialog seperti dibawah ini, pilih *Auti Select PROM*



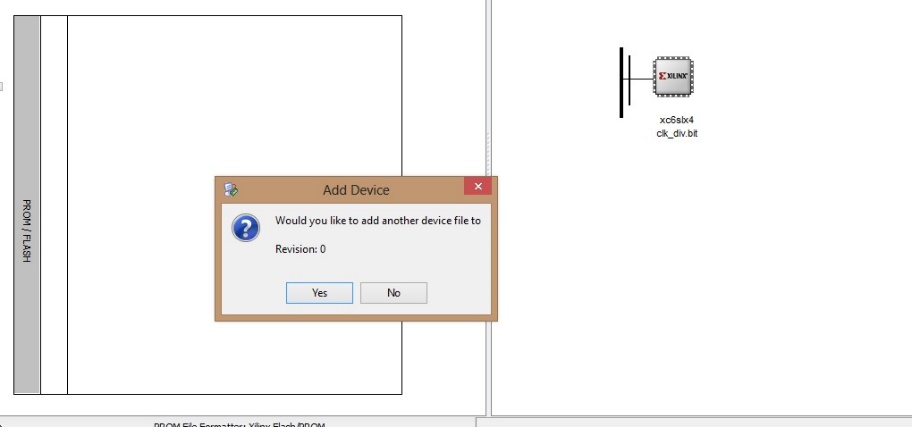
1. Isi nama file pada *output file name* dan pilih untuk tempat menyimpina file hasil compile PROM



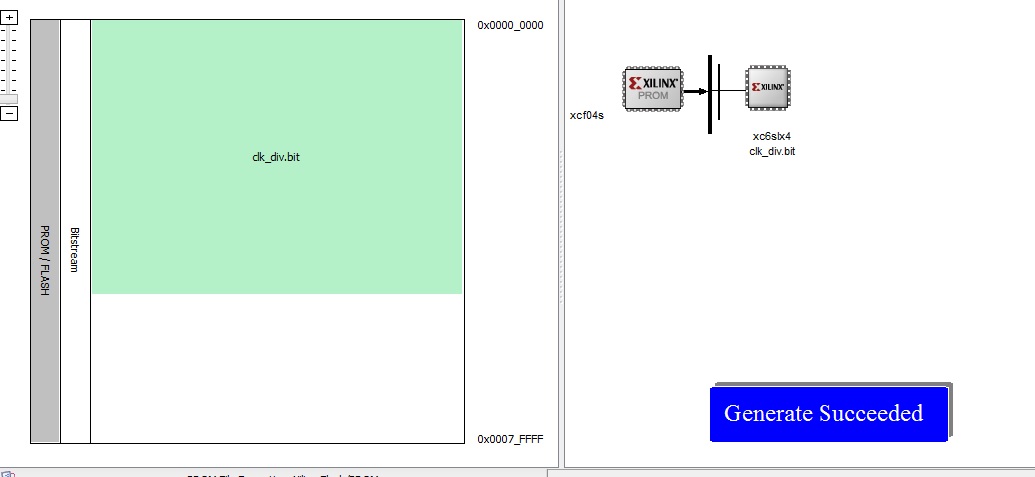
1. Lalu pilih OK



1. Lalu *Generate File* dan pilih .bit yang akan dimasukkan ke dalam PROM lalu OK/OPEN, jika muncul tampilan seperti dibawah ini maka pilih NO.



1. Jika selesai maka akan muncul tampilan seperti dibawah ini



1. Lakukan proses pemrogran FPGA melalui iMPACT seperti bab praktikum sebelumnnya dengan memasukkan file ***.bit*** dan menambahkan file hasil compile prom dengan ekstensi .mcs dan pilih S25FL128S lalu OK dan pilih PROGRAM untuk memprogram FPGA. Jika PROM terisi maka akan muncul tulisan FLASH.

