# Mixed-Signal-SoC-Design-Marathon

# Design of Multipurpose Counter

A Multipurpose counter is designed using verilog code in esim.

## Contents

\* [Abstract]

\* [Introduction]

\* [Tools used]

\* [Reference circuit]

\* [Verilog code]

\* [Model created]

\* [Waveforms]

\* [Author]

\* [Ackwedgement]

\* [Reference]

## Abstract

This report presents a 4 four-bit multipurpose counter design written in Verilog(digital) .

## Introduction

The 4-bit counter starts incrementing from 4'b0000 to 4'h1111 and come back to 4'b0000. It will keep counting as long as it is provided with a running clock, and reset is held high.

The rollover happens when the most significant bit of the final addition gets discarded. When the counter is at a maximum value of 4'h1111 and gets one more count request, the counter tries to reach 5'b10000, but since it can support only 4-bits, the MSB will be discarded, resulting in 0.

## Tools Used

eSim

It is an Open Source EDA developed by FOSSEE, IIT Bombay. It is used for electronic circuit simulation. It is made by the combination of two software namely NgSpice and KiCAD. For more details refer:

https://esim.fossee.in/home

NgSpice

It is an Open Source Software for Spice Simulations. For more details refer:

http://ngspice.sourceforge.net/docs.html

Makerchip

It is an Online Web Browser IDE for Verilog/System-verilog/TL-Verilog Simulation. For more details refer:

https://www.makerchip.com/

Verilator

It is a tool which converts Verilog code to C++ objects. For more details refer:

https://www.veripool.org/verilator/

## Reference Circuit Details

As we can see in this multi-purpose counter, apart from counting straight from 4’h0000, we can decide the starting number using load input. In case our load pin is high, it is taking the input from data\_in pin and incrementing the value thereafter. And on the output side we have end\_cnt pin which gets high once counter has completed its maximum counting.

<img width="595" alt="counter\_pic" src="https://user-images.githubusercontent.com/100422485/194709672-6123fe17-00e4-4592-ac8f-5389f4bd6f8b.png">

## Verilog Code

\TLV\_version 1d: tl-x.org

\SV

/\* verilator lint\_off UNUSED\*/ /\* verilator lint\_off DECLFILENAME\*/ /\* verilator lint\_off BLKSEQ\*/ /\* verilator lint\_off WIDTH\*/ /\* verilator lint\_off SELRANGE\*/ /\* verilator lint\_off PINCONNECTEMPTY\*/ /\* verilator lint\_off DEFPARAM\*/ /\* verilator lint\_off IMPLICIT\*/ /\* verilator lint\_off COMBDLY\*/ /\* verilator lint\_off SYNCASYNCNET\*/ /\* verilator lint\_off UNOPTFLAT \*/ /\* verilator lint\_off UNSIGNED\*/ /\* verilator lint\_off CASEINCOMPLETE\*/ /\* verilator lint\_off UNDRIVEN\*/ /\* verilator lint\_off VARHIDDEN\*/ /\* verilator lint\_off CASEX\*/ /\* verilator lint\_off CASEOVERLAP\*/ /\* verilator lint\_off PINMISSING\*/ /\* verilator lint\_off BLKANDNBLK\*/ /\* verilator lint\_off MULTIDRIVEN\*/ /\* verilator lint\_off WIDTHCONCAT\*/ /\* verilator lint\_off ASSIGNDLY\*/ /\* verilator lint\_off MODDUP\*/ /\* verilator lint\_off STMTDLY\*/ /\* verilator lint\_off LITENDIAN\*/ /\* verilator lint\_off INITIALDLY\*/

//Your Verilog/System Verilog Code Starts Here-

module anandita\_counter(input [3:0]data\_in,input load,clk,rst, en\_cnt,clear, output reg [3:0] data\_out, output reg end\_cnt);

parameter [3:0] base=4'b1111;

always @(posedge clk or posedge rst)

begin

if(rst)

data\_out <= 0;

else if(!en\_cnt)

data\_out <= data\_out;

else if(load)

data\_out <= data\_in;

else if(clear)

data\_out<=0;

else

data\_out <=data\_out+1;

if(data\_out == base)

end\_cnt <= 1;

else

end\_cnt<= 0;

end

endmodule

//Top Module Code Starts here:

module top(input logic clk, input logic reset, input logic [31:0] cyc\_cnt, output logic passed, output logic failed);

logic [3:0] data\_in;//input

logic load;//input

logic rst;//input

logic en\_cnt;//input

logic clear;//input

logic [3:0] data\_out;//output

logic [3:0] end\_cnt;//output

//The $random() can be replaced if user wants to assign values

assign data\_in =0101;

assign load = 0;

assign rst =0;

assign en\_cnt =1;

assign clear = 0;

anandita\_counter anandita\_counter(.data\_in(data\_in), .load(load), .clk(clk), .rst(rst), .en\_cnt(en\_cnt), .clear(clear), .data\_out(data\_out), .end\_cnt(end\_cnt));

endmodule

## Model created

<img width="960" alt="model\_created" src="https://user-images.githubusercontent.com/100422485/194707709-b2b05207-255d-4404-baa0-184bc75a247f.PNG">

<img width="959" alt="model\_symbol" src="https://user-images.githubusercontent.com/100422485/194707828-1eea7618-7e6b-40cf-b896-40457e62e15d.PNG">

## Waveforms

<img width="764" alt="counter\_new\_waveform" src="https://user-images.githubusercontent.com/100422485/194707624-0f9d93ac-6c70-4d70-b28e-94063785f58d.PNG">

## Author

```bash

Anandita, National Institute of Technology, Patna

```

## Acknowledgements

- Kunal Ghosh, Co-founder, VSD Corp. Pvt. Ltd. - kunalpghosh@gmail.com

-FOSSEE, IIT Bombay

-Steve Hoover (Founder, Redwood EDA)

-Sumanto Kar (eSim Team, FOSSEE, IIT Bombay)

## References

-[1] NPTEL Lecture on Digital Design by Professor Srinivasan.