UP DOWN COUNTER VERIFICATION PLAN

MISSDV0524
M.BASAVARAJU

TABLE OF CONTENTS

1.Introduction

Counter is a sequential circuit. A digital circuit which is used for counting pulses is known as a counter. Counter is the widest application of flip-flops. It is a group of flip-flops with a clock signal applied.we have two types of counters those are up counter and down counter.This counter counts from preloaded value to maximum value and down counter counts from maximum value to preload value .

This up-down counter contains four 8-bit registers, namely, preload register, upper limit register, lower limit register and cycle count register.Din is a bi-directional bus for reading and writing data. All these registers are readable with ncs and nrd signal. And this counter starts counting up from the preload value once the start pulse of one clock duration is applied. The cycle count register has the number of cycles this block should start counting. counter starts counting up from preload value and after reaching upper limit, counts down and after reaching lower limit,counts up again and then reaches its preload value.

Application of counters

- Frequency counters
- Digital clock
- Time measurement
- Frequency divider circuits

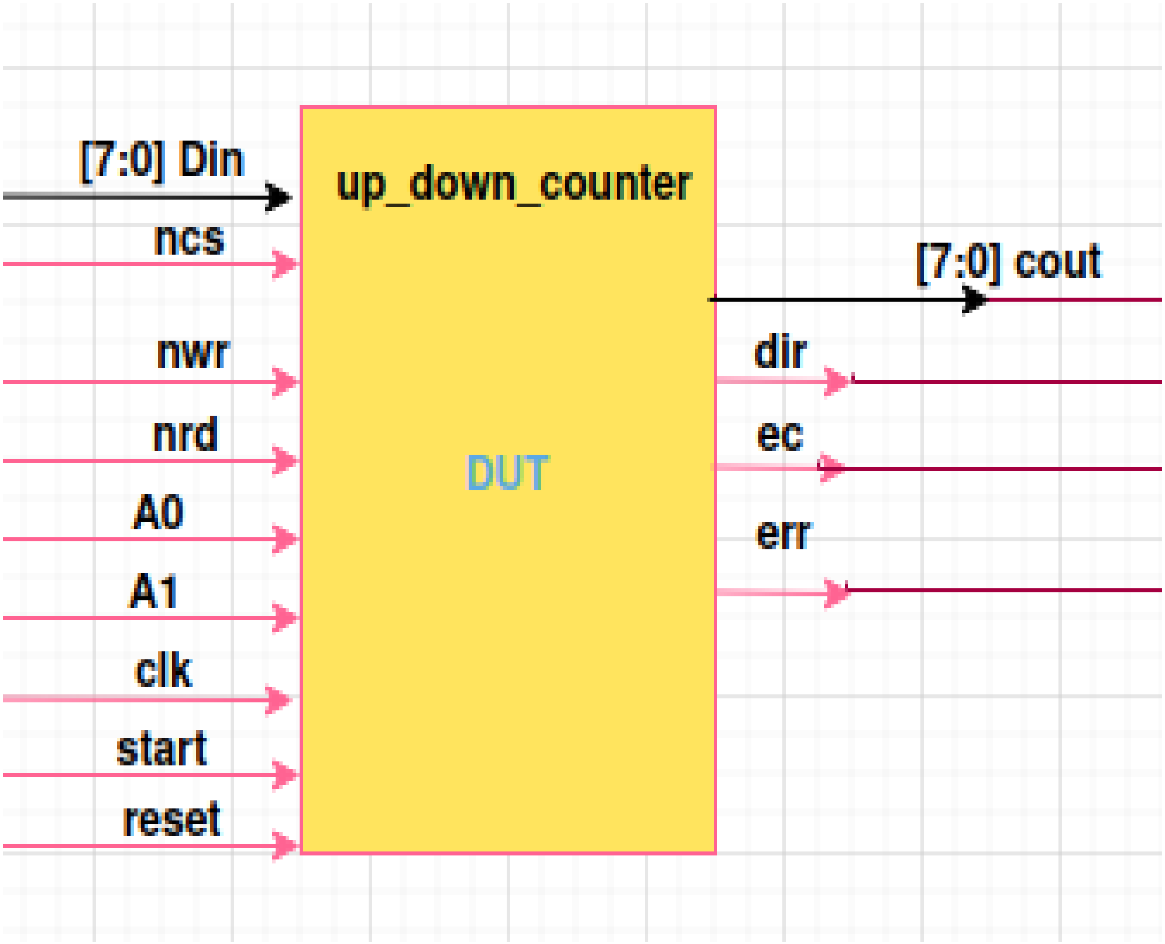
2.Signal Description

S.no	Signal Name	Input/Output/Inout	Width	Description
1	ncs	input	1	active low chip select
2	nrd	input	1	active low read signal
3	nwr	input	1	active low write signal
4	Din	inout	8	data
5	A0	input	1	selection to read or write
6	A1	input	1	selection to read or write
7	start	input	1	start pulse to enable the counting operation
8	reset	input	1	(active low) All 4 registers are reset to 0 except upper limit register, when reset is applied to the block. Upper limit register defaults to 8'hff.
9	clk	input	1	clock signal
10	cout	output	8	cout tells about number of counts
11	err	output	1	error signal tells whether correct data is given or not
12	dir	output	1	dir tells about the whether counter is

				doing up counting or down counting
13	ec	output	1	end cycle indicates the end of one cycle

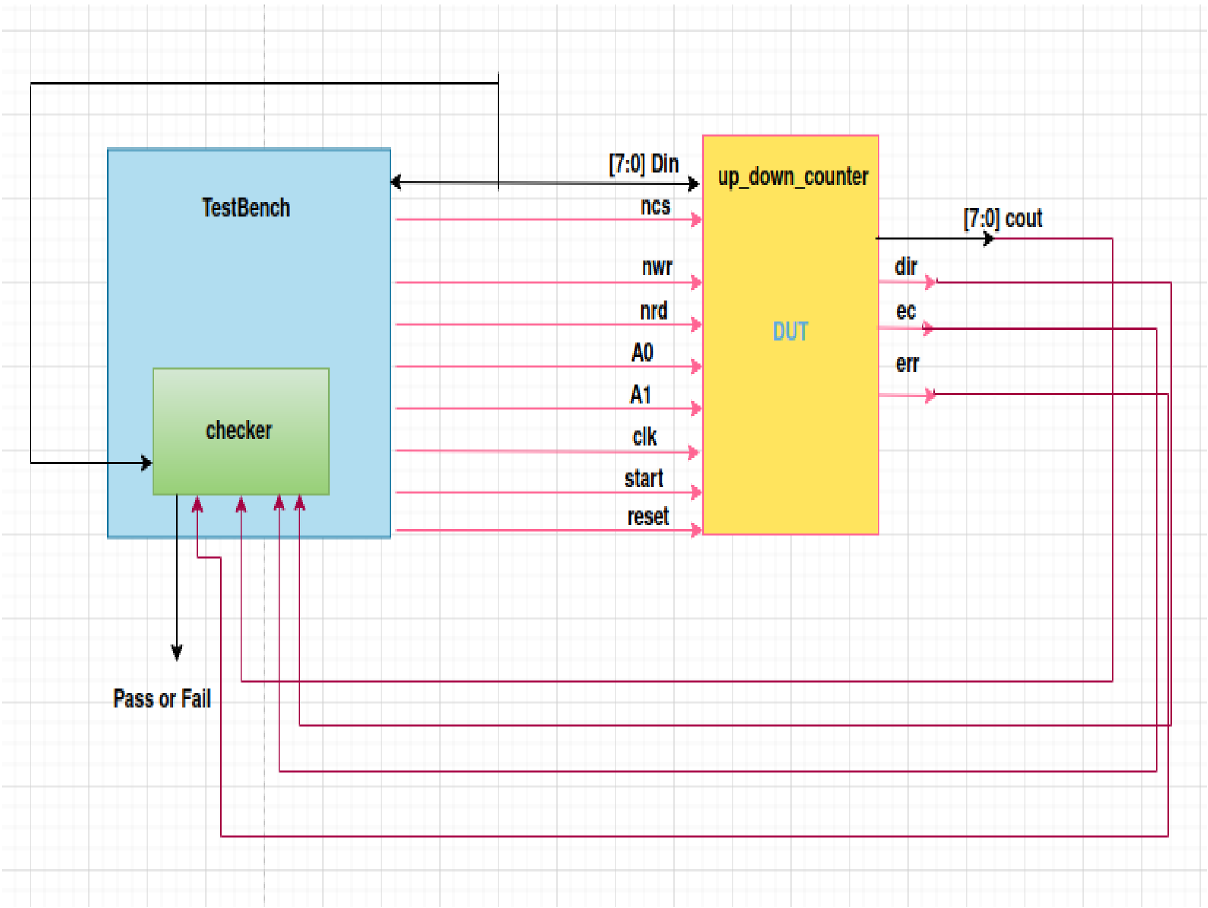
2.1 Table : input & output signals

3 . Design Architecture



3.1 Figure : Design Block diagram

4 . Design Environment

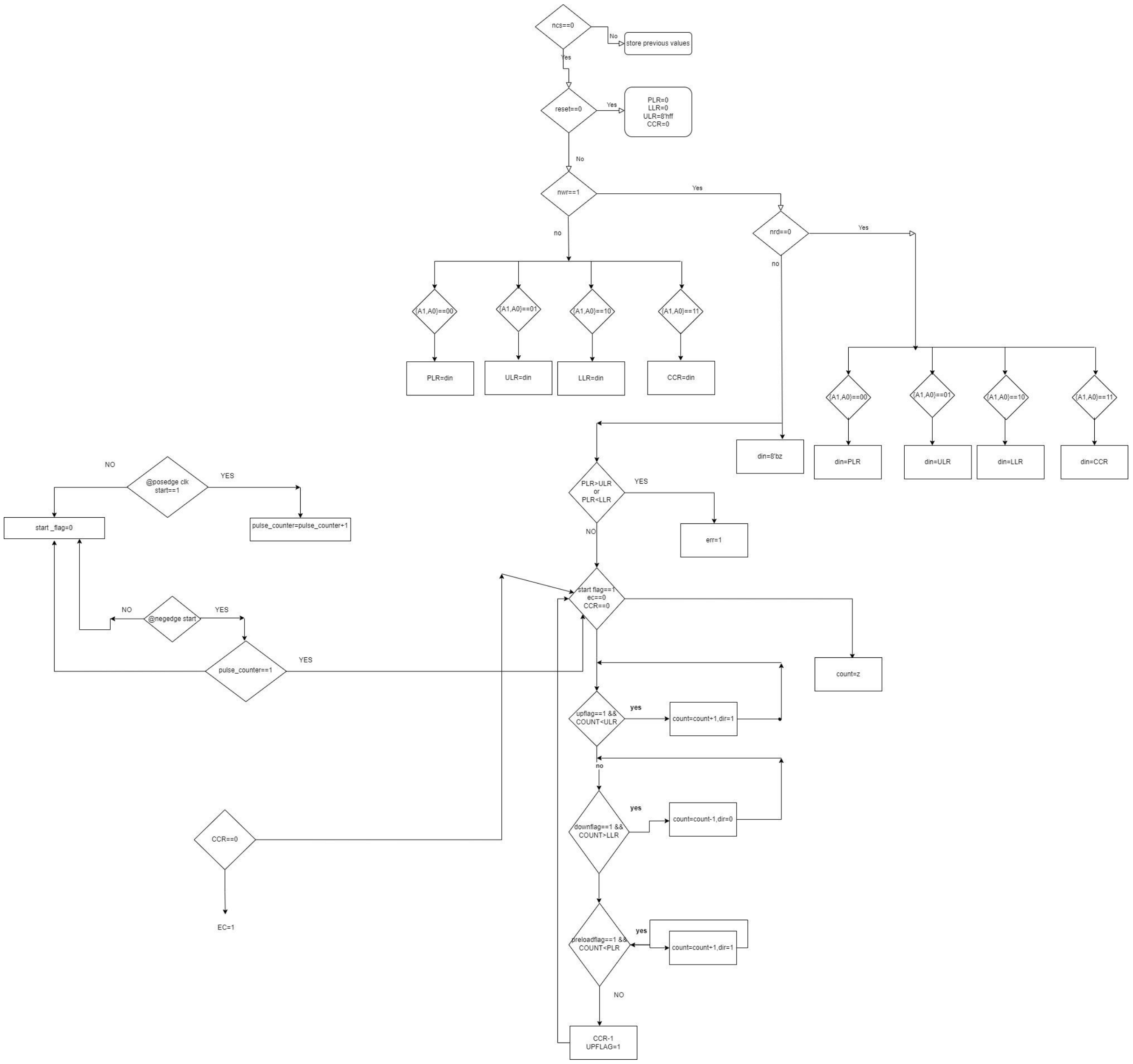


4.1 Figure : Design Environment

Description

We will give stimulus from the Testbench to the DUT(design under test) and take outputs of DUT and then give it to the checker. The checker will compare the outputs of DUT with outputs of checker logic so that it will give a result as Pass or Fail .

5.Design flow



URL : click_here_to_view_image

5.1 Flowchart: Design Flow

6.Write Block

Description

This up-down counter contains four 8-bit registers, those are,

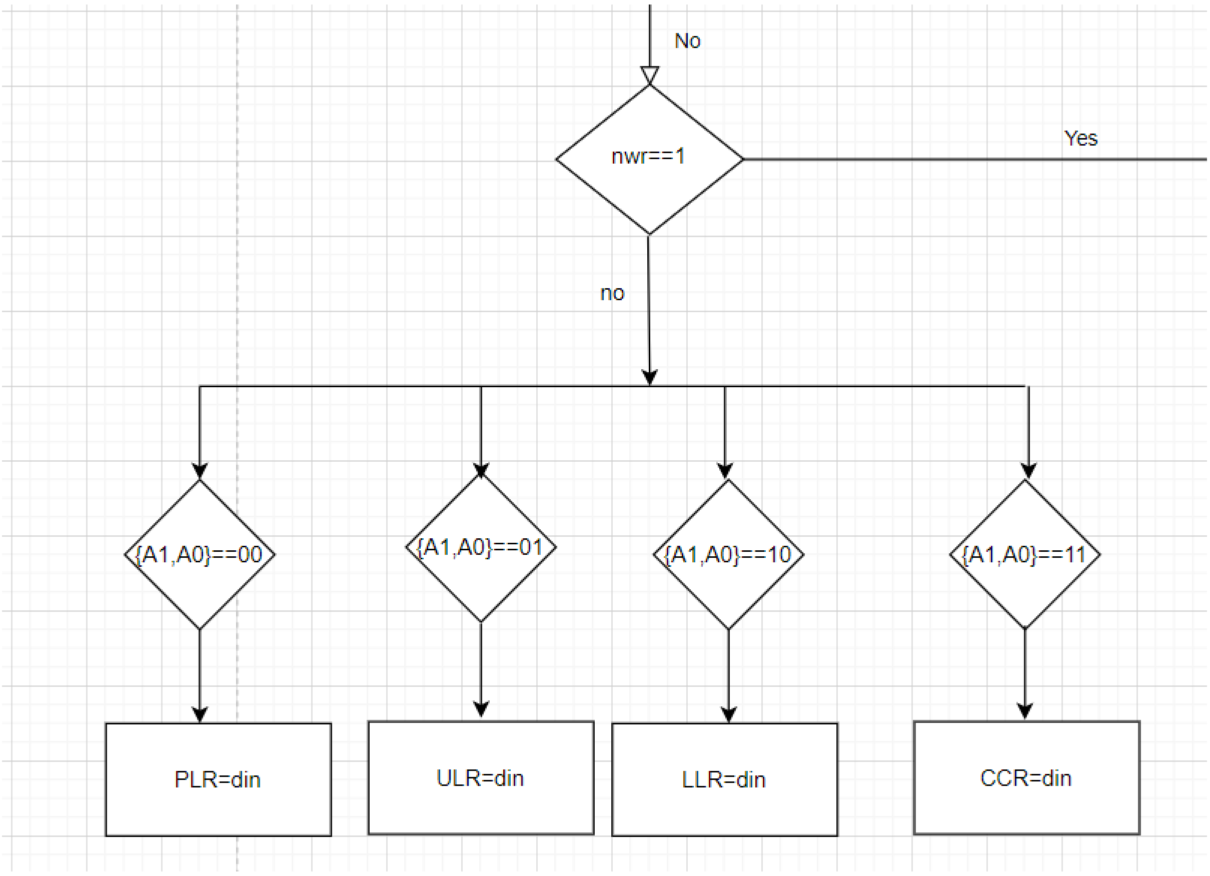
1) Preload register (PLR)
2) Upper limit register (ULR)
3) Lower limit register (LLR)
4) Cycle count register (CCR)

These registers are programmable through the proper selection of A0,A1 , ncs, nwr and Din[7:0]signals. Din is a bi-directional bus.

writing into these registers is synchronous to the clock and nwr and A1,A0 are valid for one clock.

6.1 Table : write block

input Data	Selection lines{A1,A0}	Registers
Din=10	{A1,A0}= 2’b00	data loaded to PLR PLR=10
Din=15	{A1,A0}= 2’b01	data loaded to ULR ULR=15
Din=5	{A1,A0}= 2’b10	data loaded to LLR LLR=5
Din=2	{A1,A0}= 2’b11	data loaded to CCR CCR=2



6.1 Flowchart: Write block

7.Read Block

Description

Data can be read through **Din** if and only if data is being already written.We have taken a temporary register(**reg_out**) since **Din** is an inout port which is a wire data type. Since we are working synchronously with the clock we will use always blocks so in always blocks the left hand side assignment must be register data type. we will take a temporary register, so that we can load the data into that temporary register which is useful in read operation .

eg: for (ncs=0,nrd=0,nwr!=0)

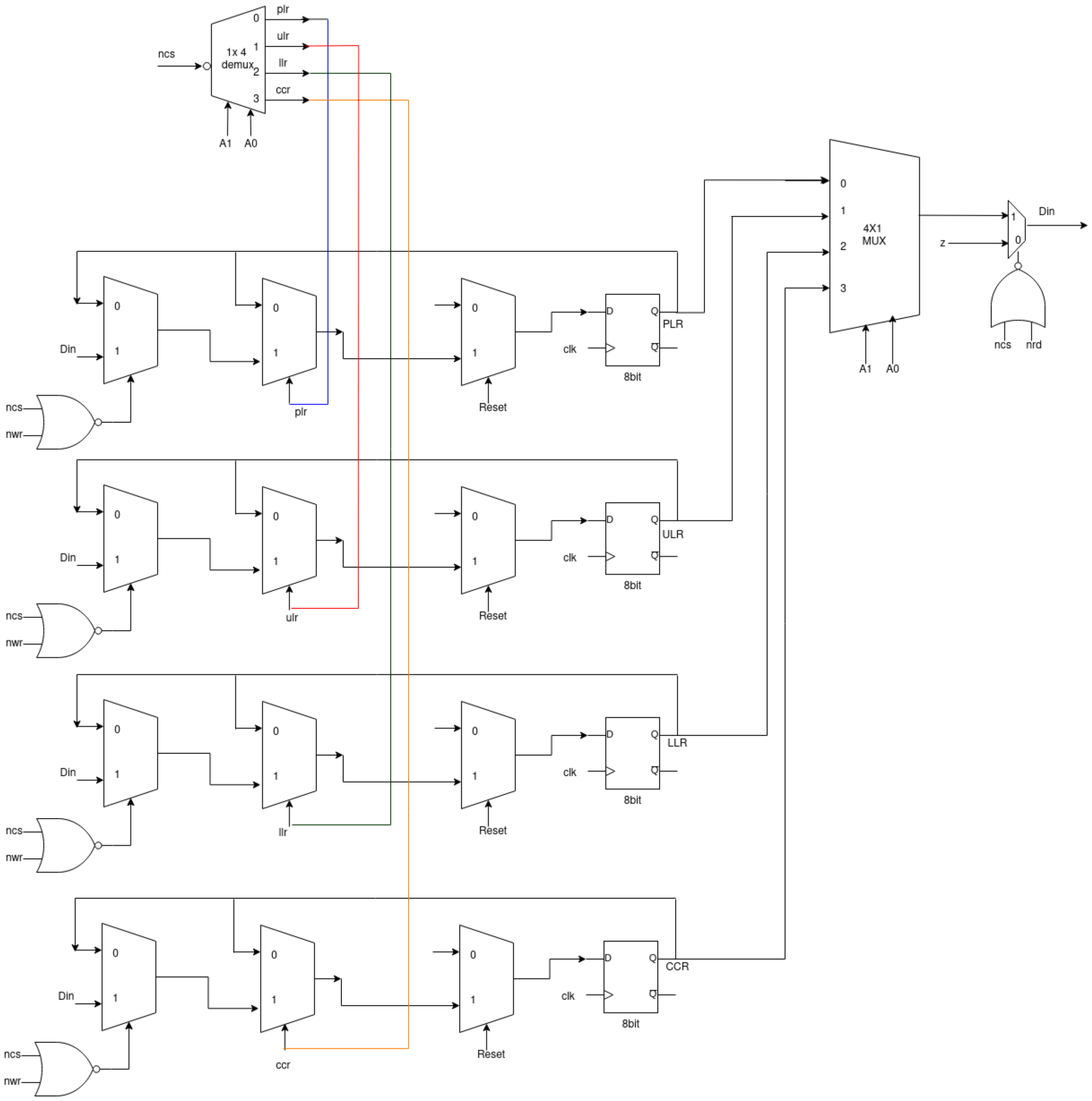
Selection lines{A1,A0}	Registers	DATA read
{A1,A0}= 2’b00	data loaded from PLR	reg_out=10
{A1,A0}= 2’b01	data loaded from ULR	reg_out=15
{A1,A0}= 2’b10	data loaded from LLR	reg_out=5
{A1,A0}= 2’b11	data loaded from CCR	reg_out=2

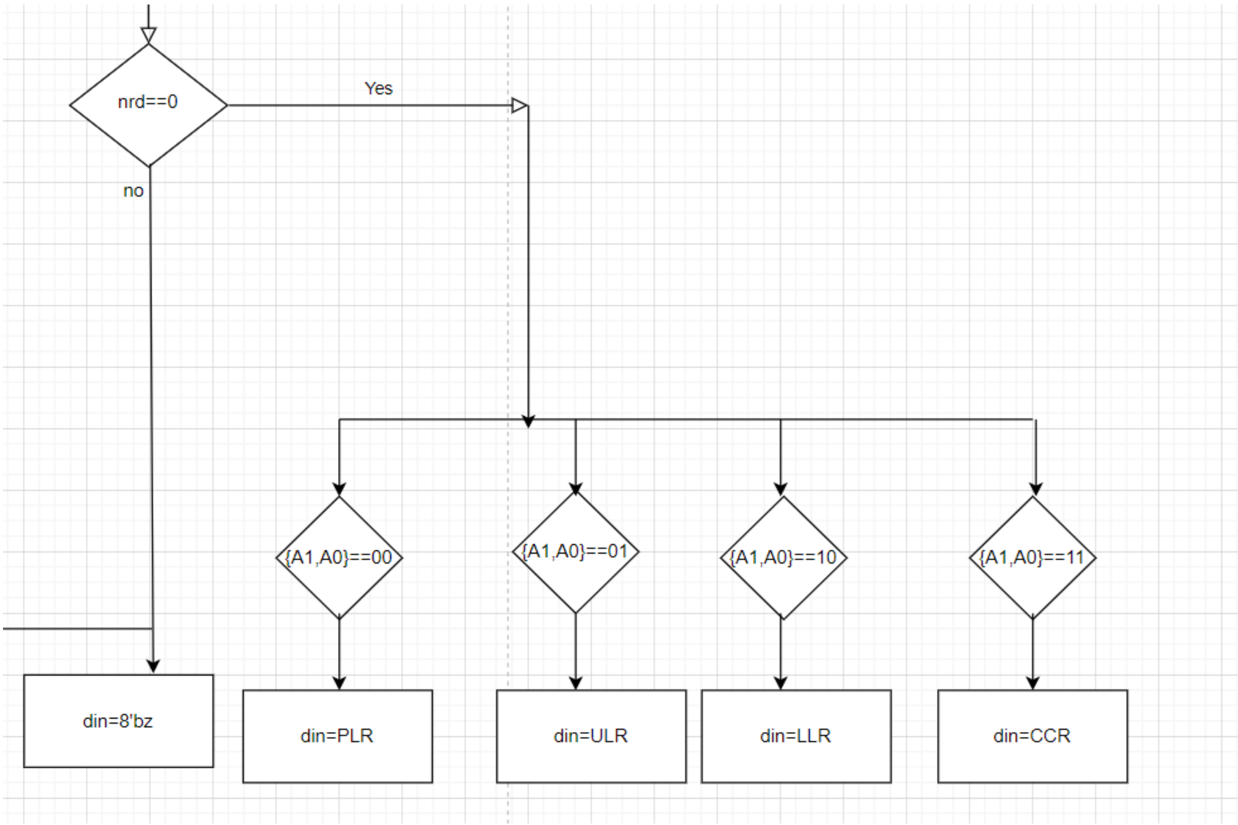
7.1 Table : Read and block

ncs	reset	nwr	nrd	description
1	x	x	x	stores previous values
0	0	x	x	reset default values stored into registers
0	1	0	x	write operation based on {A1,A0}
0	1	1	0	read
0	1	1	1	no read no write

\

7.1 Diagram :read and write block





7.1 Flowchart : read & write

8.Count Block

Description

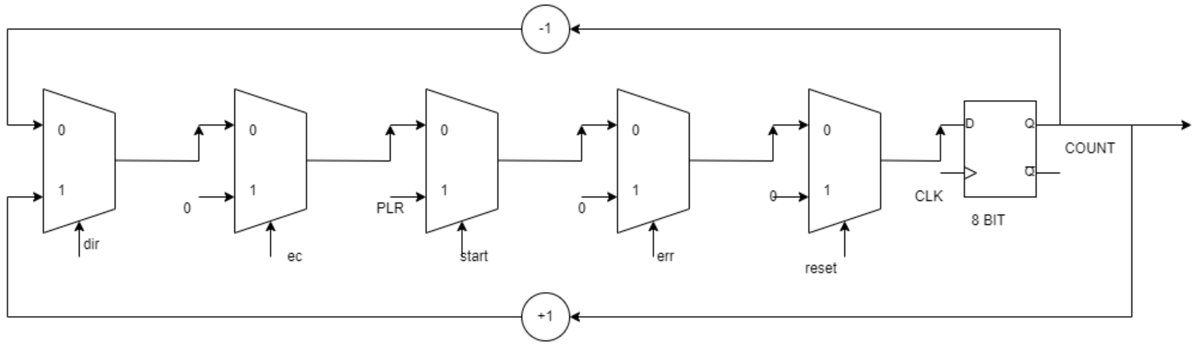
Counter block is used for up counting and down counting based on register values. counting starts only when the start pulse with one duration is applied and ncs is low and error signal is zero and number of cycles!=0

One cycle means ⇒ the time taken for the block to start counting up from preload value and after reaching upper limit, counts down and after reaching lower limit, counts up again and then reaches its preload value.

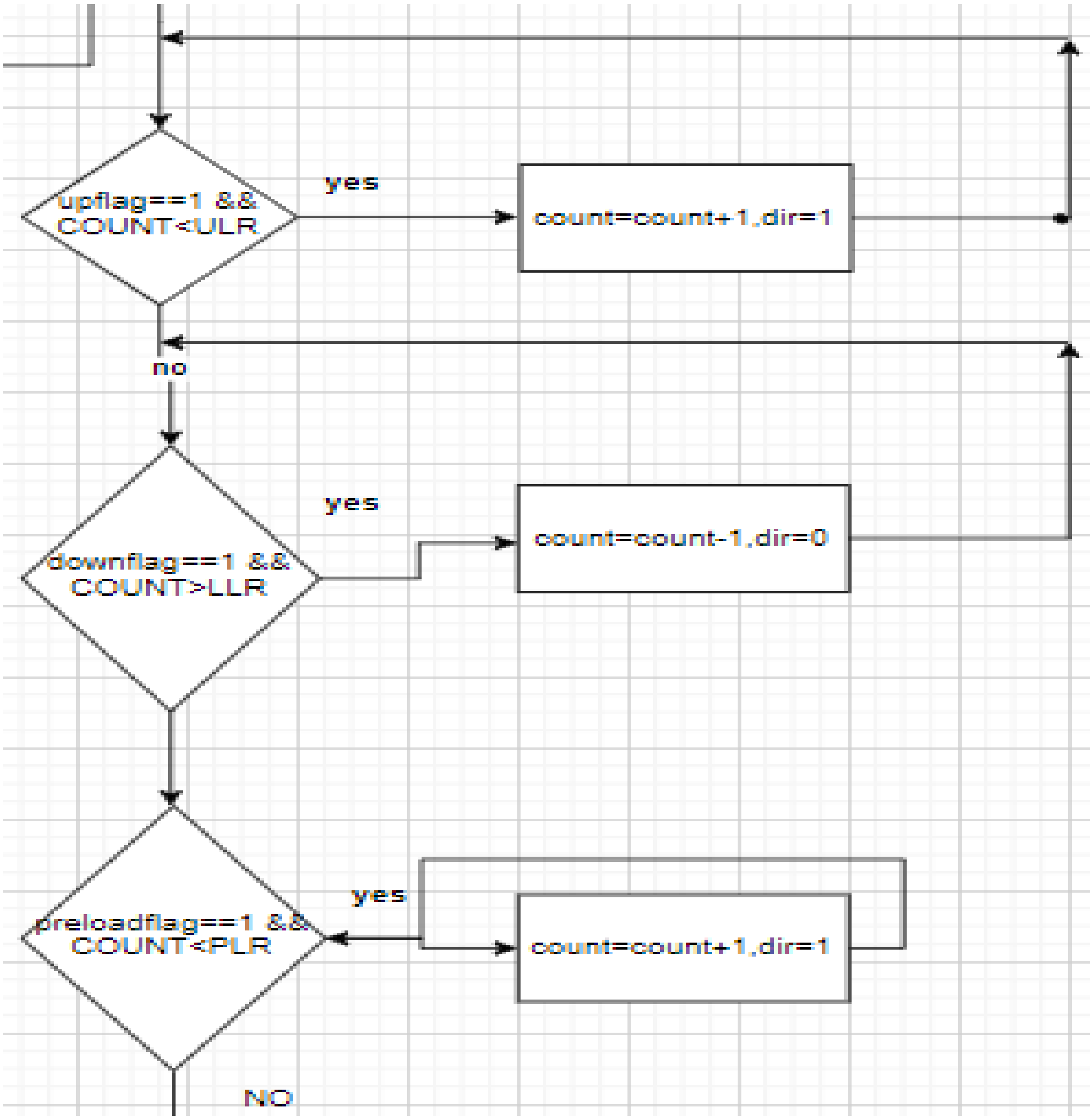
Operation	dir

cout=cout-1	0
cout=cout+1	1
cout=cout	stores previous dir value

8.1 Table : Count Block



8.1 Figure : .Count Block



8.1 Flowchart : Start block

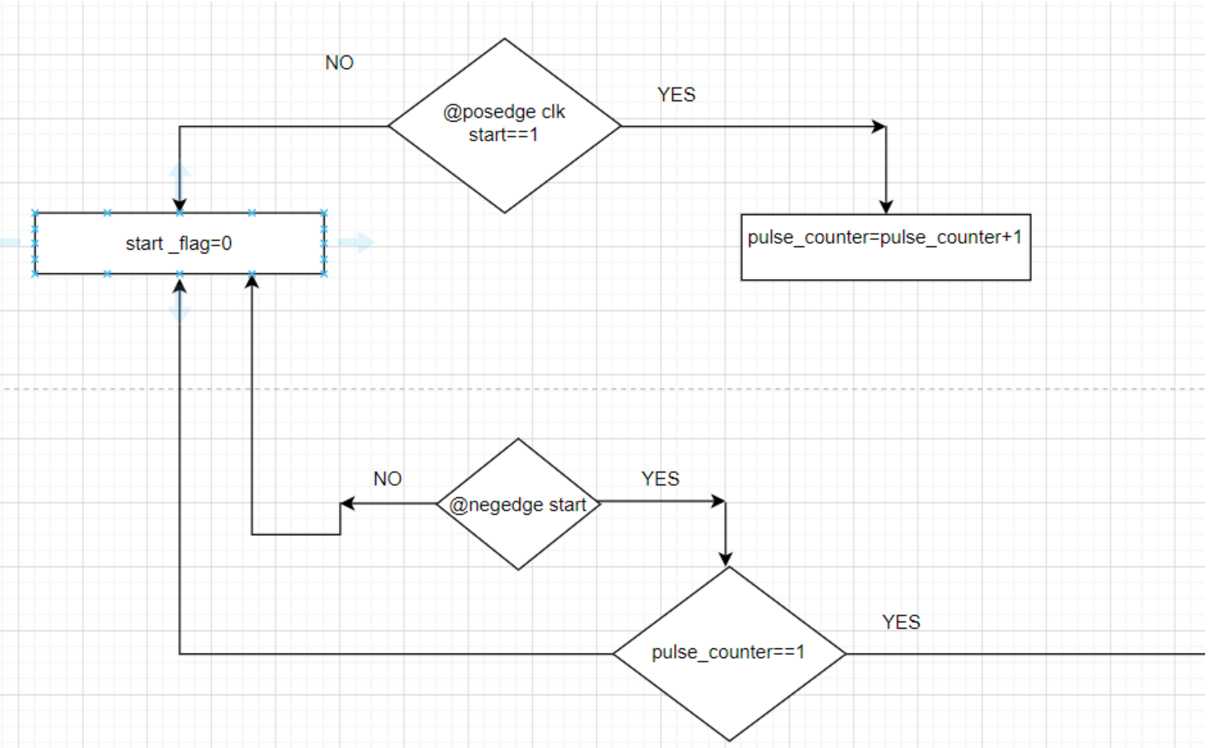
9.Start Block

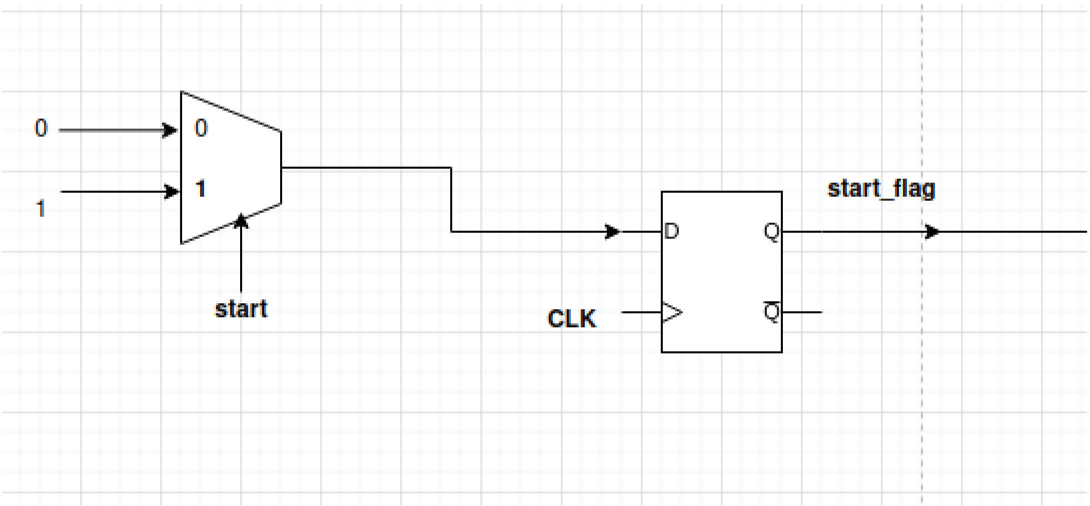
Description

Start block is used to make.

- The count operation starts after the start signal of only one clock pulse is applied.
- If the start pulse is not applied, the count operation will not start.
- The below design shows how the start block is modeled.

9.1 Flowchart : start block





9.1 Figure : start block

start one pulse	counting happens if ncs=0 err=0 ec=0
start more or less than one pulse	count wont happens

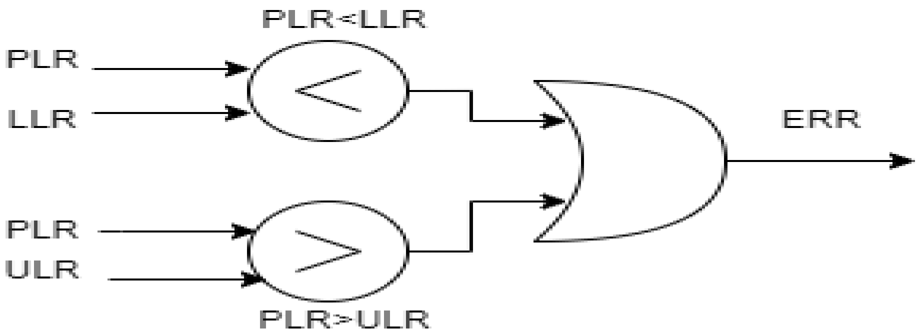
9.1Table : start block

10.Error Block

Description

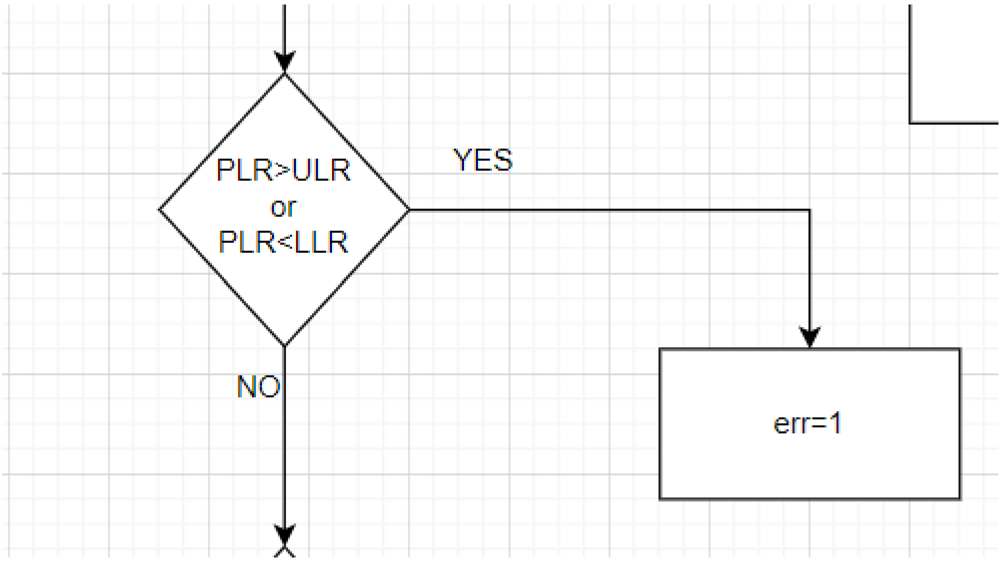
when error comes the counter won't work
errors will come in below cases listed in table
eg: when PLR is more than the LLR

10.1 Figure : error block



CONDITION	**ERROR STATUS**
PLR<LLR	err=1
PLR>ULR	err=1
LLR> ULR	err=1
LLR<PLR<ULR	err=0
LLR==PLR<ULR	err=0
LLR<PLR==ULR	err=0
LLR==PLR==ULR	err=0

10.1 Table : error conditions



err=0

10.1 Flowchart: error

11.End Count Block

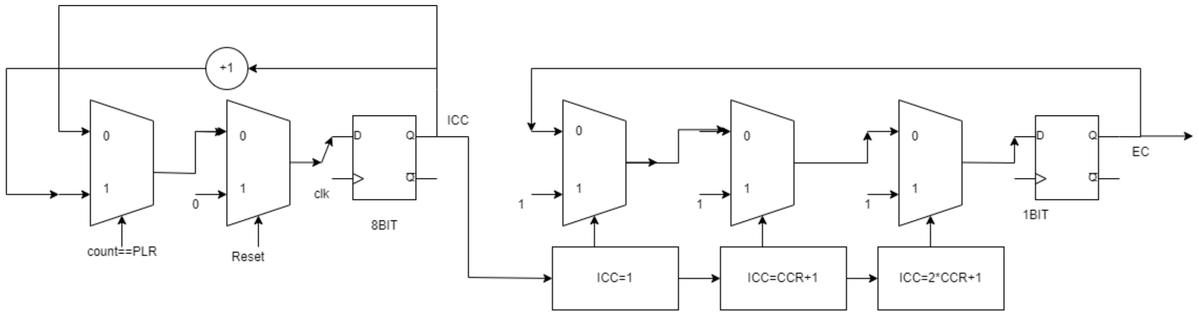
Description

A temporary register is taken and loaded with CCR

After completion of each cycle it is subtracted by 1.when it is equal to zero the end count is 1.

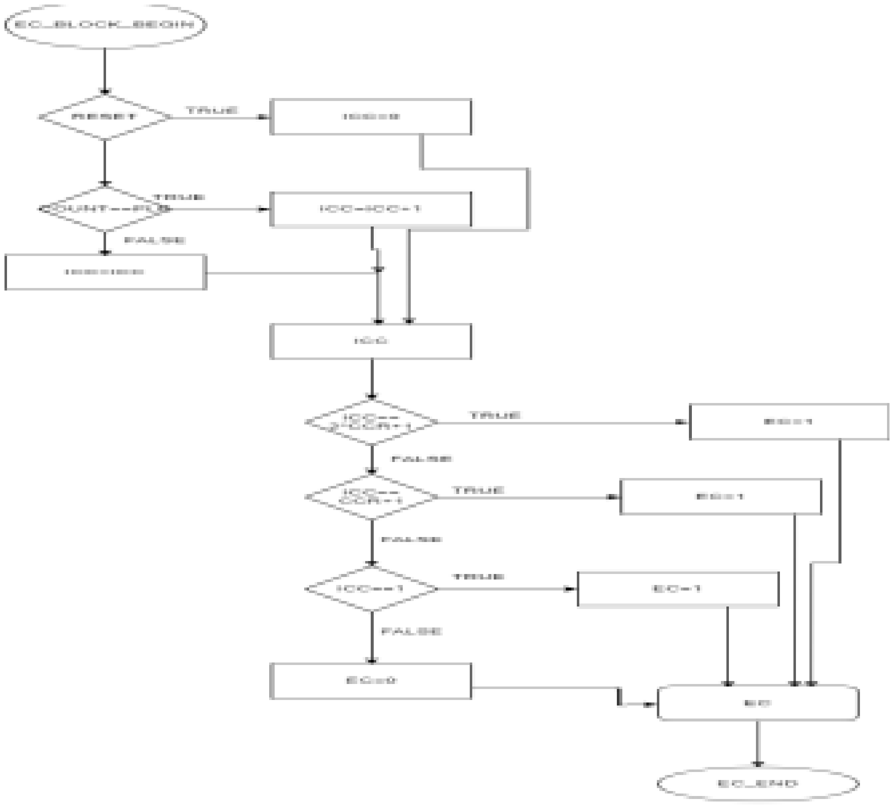
Conditions	Result
temporary_ccr ==0	ec=1

11.1 Table : end count



11.1 Figure : end count

11.1 flowchart end count

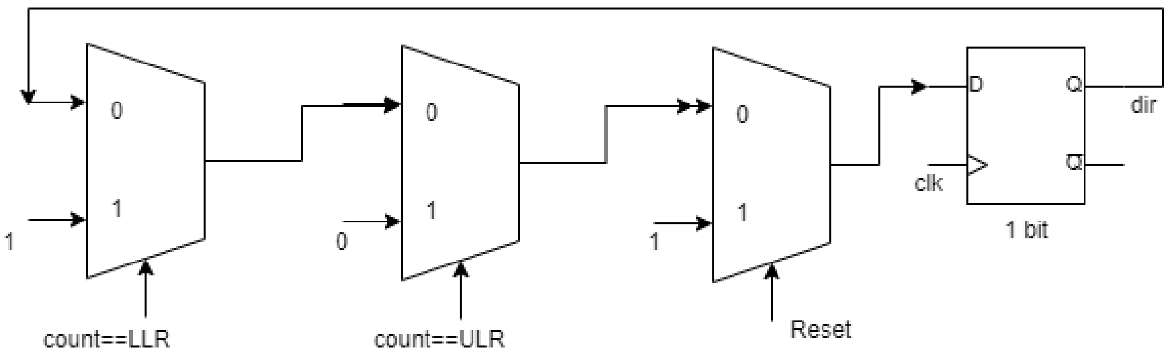


12.Direction Block

Direction block is used for giving signal direction for up counting and down counting based on direction (dir) .if dir=1 then counting process will be in up counting mode when dir changes to zero then count operation will be down counting process .

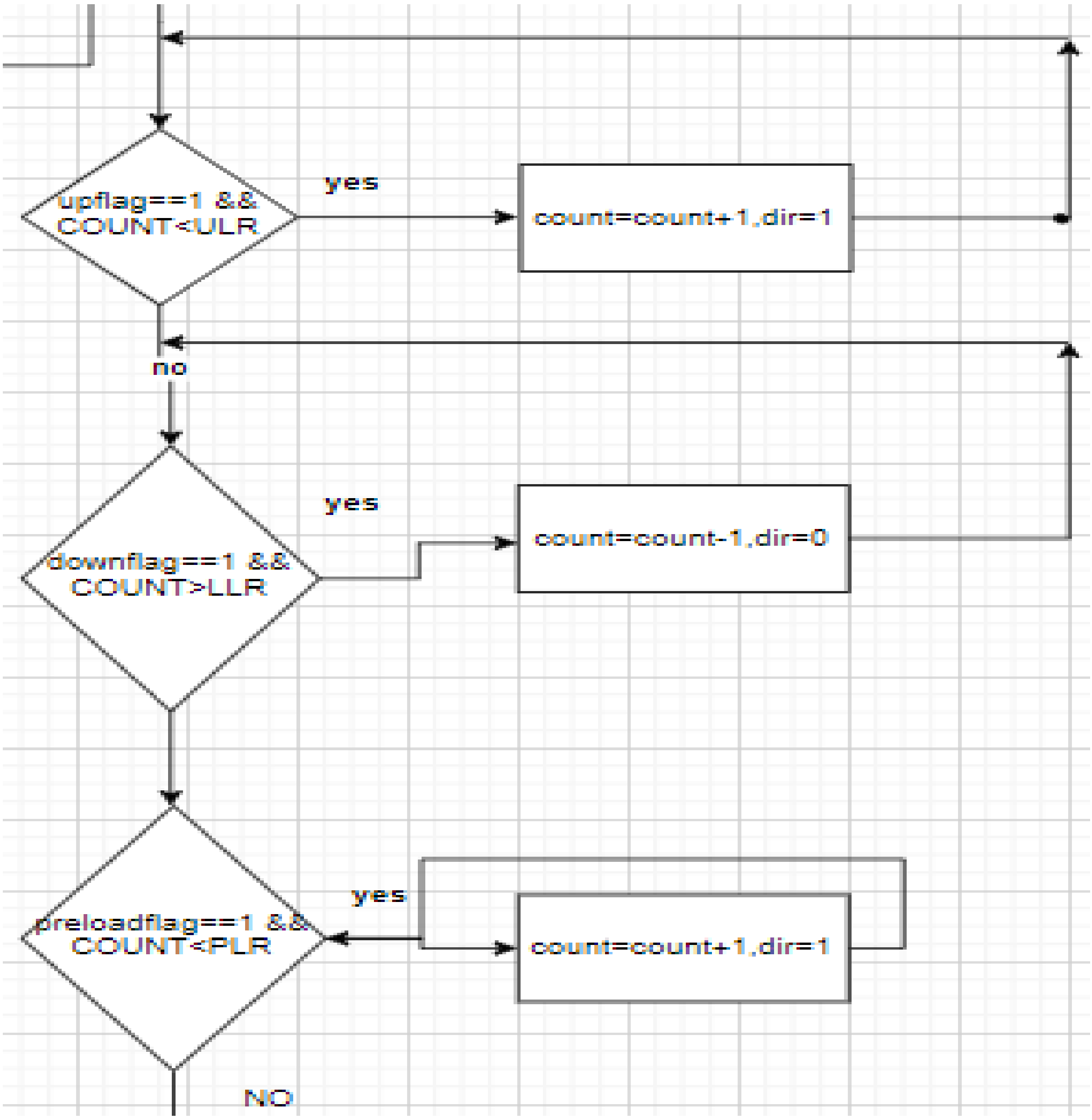
dir=1	when up counting
dir=0	when down counting

12.1 Table : direction block

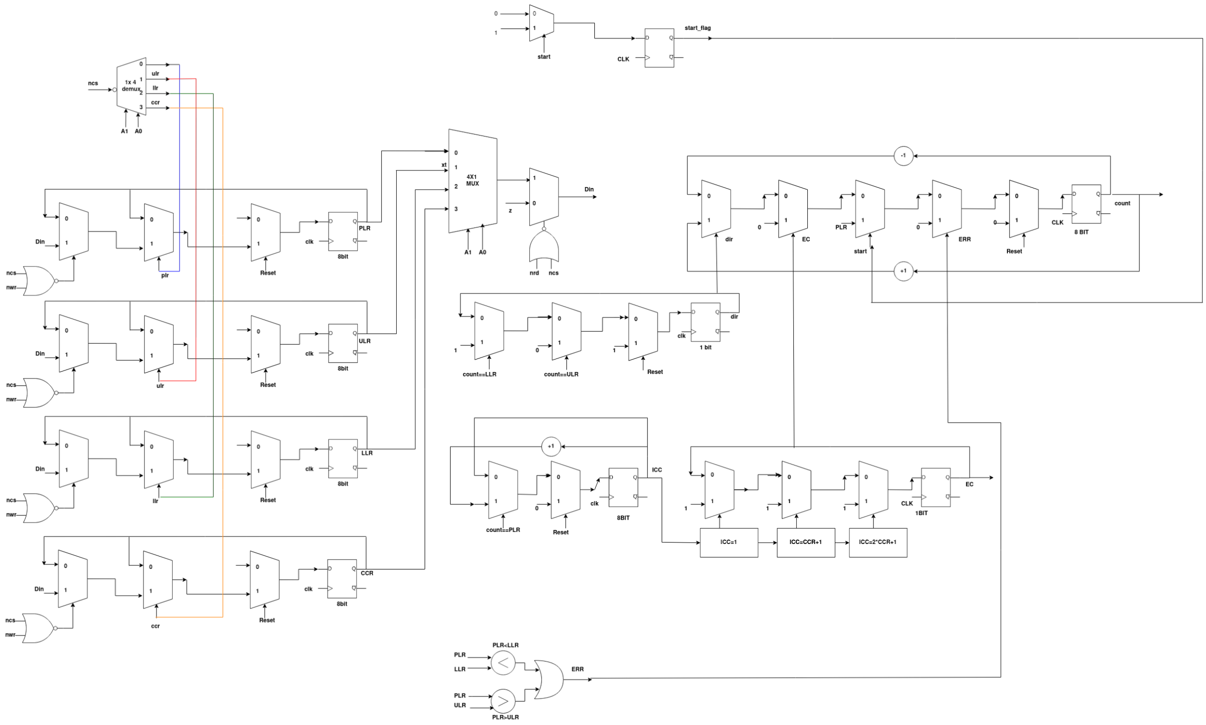


12.1 Figure : direction block

12.1 Flowchart : direction block



13.Design Diagram



13.1 Figure : design diagram

14.Reference