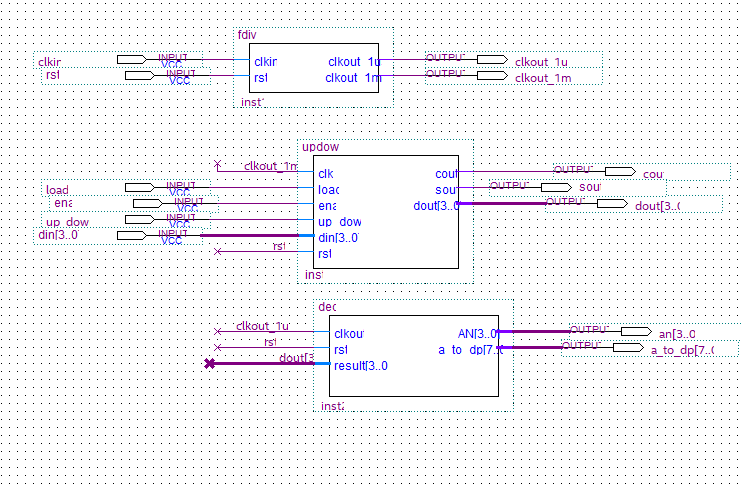
实验八**加减可控计数器及其应用实验结论**

（1）加减可控十进制计数器

1. 顶层文件



1. 计数器模块代码

module updown(clk,load,ena,cout,sout,up\_down,dout,din,rst);

input load,ena,clk,rst,up\_down;

input [3:0] din;

wire [3:0] din;

wire load,rst,up\_down;

output reg cout,sout;

output [3:0] dout;

wire [3:0] dout;

reg [3:0] Q;

assign dout =Q;

always @(posedge clk or posedge rst )

begin

if (rst==1)//异步清零，高电平清零

Q<=0;

else if (!load)//同步预置，低电平加载数据

Q<=din;

else if (ena)//同步使能，高电平使能

begin

if (up\_down==1&&Q==9)

Q<=0;

else if (up\_down==0&&Q==0)

Q<=9;

else//同步加减可控，高电平递增计数，低电平递减计数。

begin

if (up\_down==1) //加计数

Q=Q+1;

else //减计数

Q<=Q-1;

end

end

else

Q<=Q;

end

always

begin

if ((rst==0)&&(load==1)&&(up\_down==1)&&(ena==1))

cout<=1;

else

cout<=0;

end

always

begin

if ((rst==0)&&(load==1)&&(up\_down==0)&&(ena==1))

sout<=1;

else

sout<=0;

end

endmodule

1. 测试文件代码

`timescale 1 ns/ 1 ps

module cnt\_10\_updown\_vlg\_tst();

// constants

// general purpose registers

//reg eachvec;

// test vector input registers

reg clkin;

reg [3:0] din;

reg ena;

reg load;

reg rst;

reg up\_down;

// wires

wire [7:0] a\_to\_dp;

wire [3:0] an;

wire clkout\_1ms;

wire clkout\_1us;

wire cout;

wire [3:0] dout;

wire sout;

// assign statements (if any)

cnt\_10\_updown i1 (

// port map - connection between master ports and signals/registers

.a\_to\_dp(a\_to\_dp),

.an(an),

.clkin(clkin),

.clkout\_1ms(clkout\_1ms),

.clkout\_1us(clkout\_1us),

.cout(cout),

.din(din),

.dout(dout),

.ena(ena),

.load(load),

.rst(rst),

.sout(sout),

.up\_down(up\_down)

);

initial

begin

// code that executes only once

// insert code here --> begin

clkin=0;

rst=1;

ena=0;

load=1;

up\_down=0;

din=0;

#100000

rst=0;

load=0;

din=6;

#10000000

ena=1;

load=1;

#10000000

up\_down=1;

#100000

up\_down=0;

#10000000

up\_down=1;

// --> end

$display("Running testbench");

end

always

// optional sensitivity list

// @(event1 or event2 or .... eventn)

begin

// code executes for every event on sensitivity list

// insert code here --> begin

#10 clkin=~clkin;

//@eachvec;

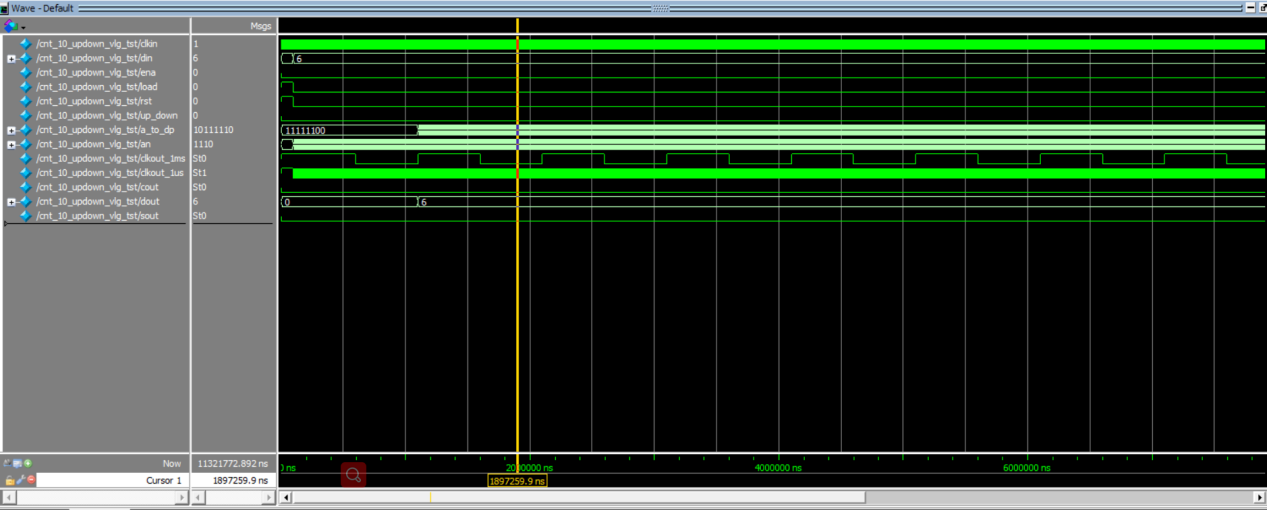
// --> end

end

endmodule

1. 仿真波形

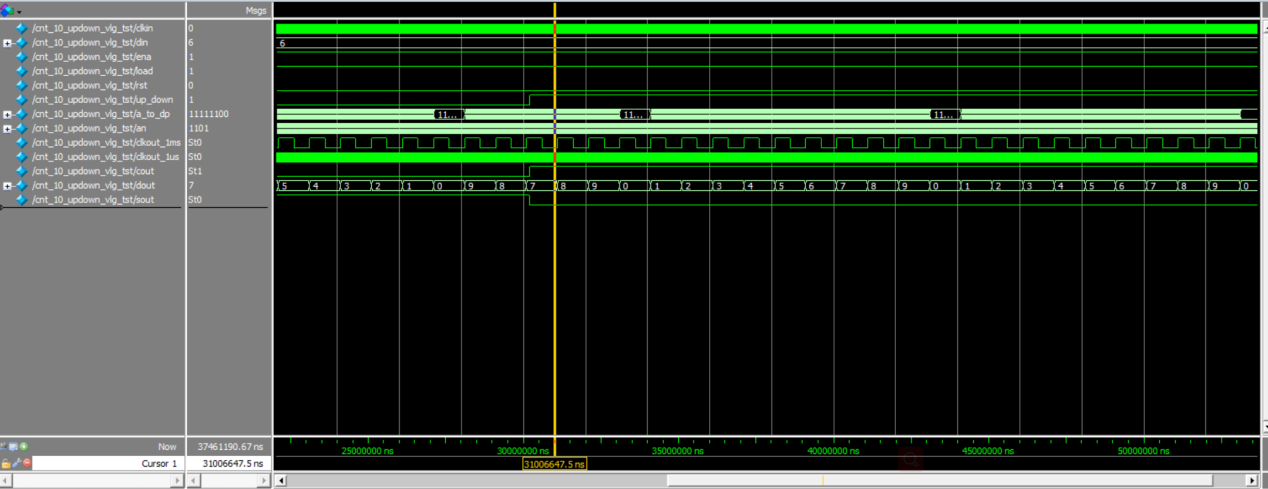
1.数据加载有效时(rst=0,load=0,din=6）



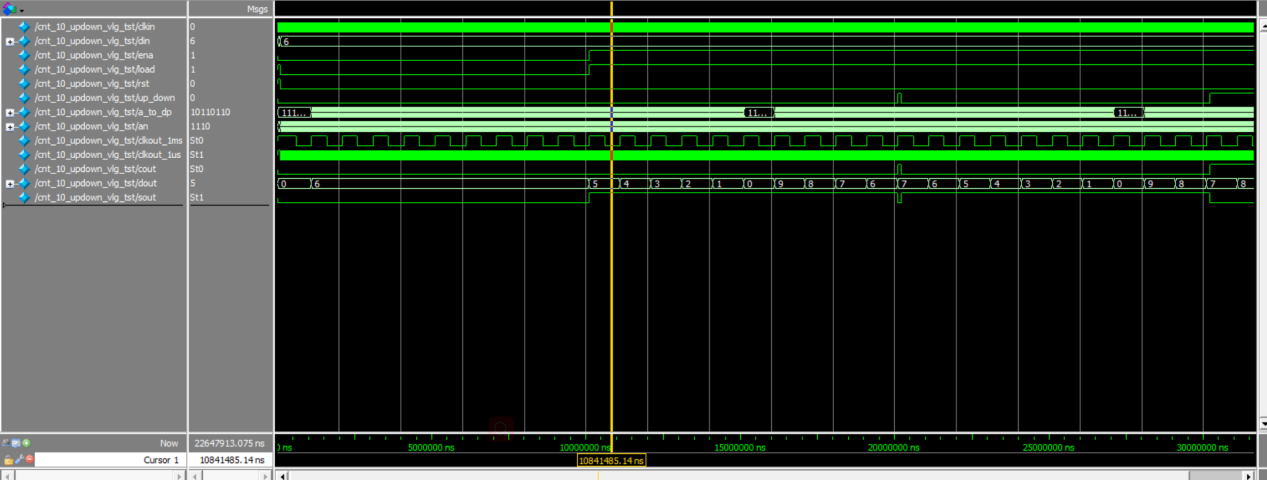
2.使能有效时（rst=0,load=1,ena=1）



3.递增计数cout为1时（rst=0,load=1,ena=1,up\_down=1.）

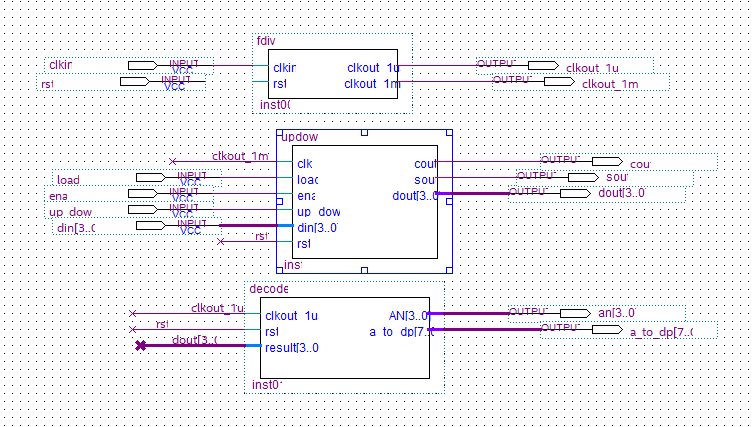


1. 递减计数sout为1时（rst=0,load=1,ena=1,up\_down=0.）



1. 加减可控十六进制计数器

1. 顶层文件



2. 计数器模块代码

module updown(clk,load,ena,cout,sout,up\_down,dout,din,rst);

input load,ena,clk,rst,up\_down;

input [3:0] din;

wire [3:0] din;

wire load,rst,up\_down;

output reg cout,sout;

output [3:0] dout;

wire [3:0] dout;

reg [3:0] Q;

assign dout =Q;

always @(posedge clk or posedge rst )

begin

if (rst==1)//异步清零，高电平清零

Q<=0;

else if (!load)//同步预置，低电平加载数据

Q<=din;

else if (ena)//同步使能，高电平使能

begin

if (up\_down==1&&Q==15)

Q<=0;

else if (up\_down==0&&Q==0)

Q<=15;

else//同步加减可控，高电平递增计数，低电平递减计数。

begin

if (up\_down==1) //加计数

Q=Q+1;

else //减计数

Q<=Q-1;

end

end

else

Q<=Q;

end

always

begin

if ((rst==0)&&(load==1)&&(up\_down==1)&&(ena==1))

cout<=1;

else

cout<=0;

end

always

begin

if ((rst==0)&&(load==1)&&(up\_down==0)&&(ena==1))

sout<=1;

else

sout<=0;

end

endmodule

3. 测试文件代码

`timescale 1 ns/ 1 ps

module cnt\_16\_updown\_vlg\_tst();

// constants

// general purpose registers

//reg eachvec;

// test vector input registers

reg clkin;

reg [3:0] din;

reg ena;

reg load;

reg rst;

reg up\_down;

// wires

wire [7:0] a\_to\_dp;

wire [3:0] an;

wire clkout\_1ms;

wire clkout\_1us;

wire cout;

wire [3:0] dout;

wire sout;

// assign statements (if any)

cnt\_16\_updown i1 (

// port map - connection between master ports and signals/registers

.a\_to\_dp(a\_to\_dp),

.an(an),

.clkin(clkin),

.clkout\_1ms(clkout\_1ms),

.clkout\_1us(clkout\_1us),

.cout(cout),

.din(din),

.dout(dout),

.ena(ena),

.load(load),

.rst(rst),

.sout(sout),

.up\_down(up\_down)

);

initial

begin

// code that executes only once

// insert code here --> begin

clkin=0;

rst=1;

ena=0;

load=1;

up\_down=0;

din=0;

#100000

rst=0;

load=0;

din=6;

#10000000

ena=1;

load=1;

#10000000

up\_down=1;

#100000

up\_down=0;

#10000000

up\_down=1;

// --> end

$display("Running testbench");

end

always

// optional sensitivity list

// @(event1 or event2 or .... eventn)

begin

// code executes for every event on sensitivity list

// insert code here --> begin

#10 clkin=~clkin;

//@eachvec;

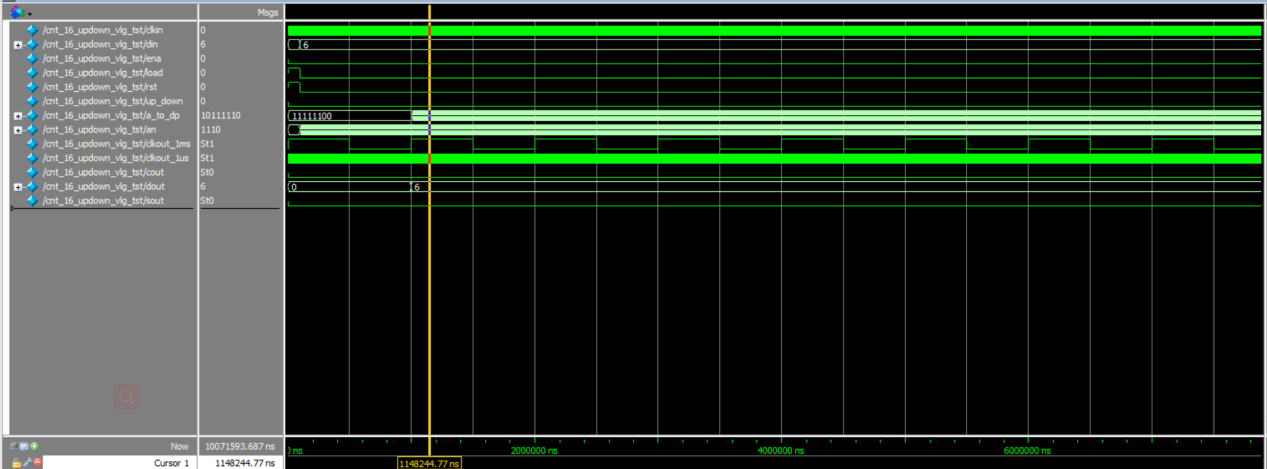
// --> end

end

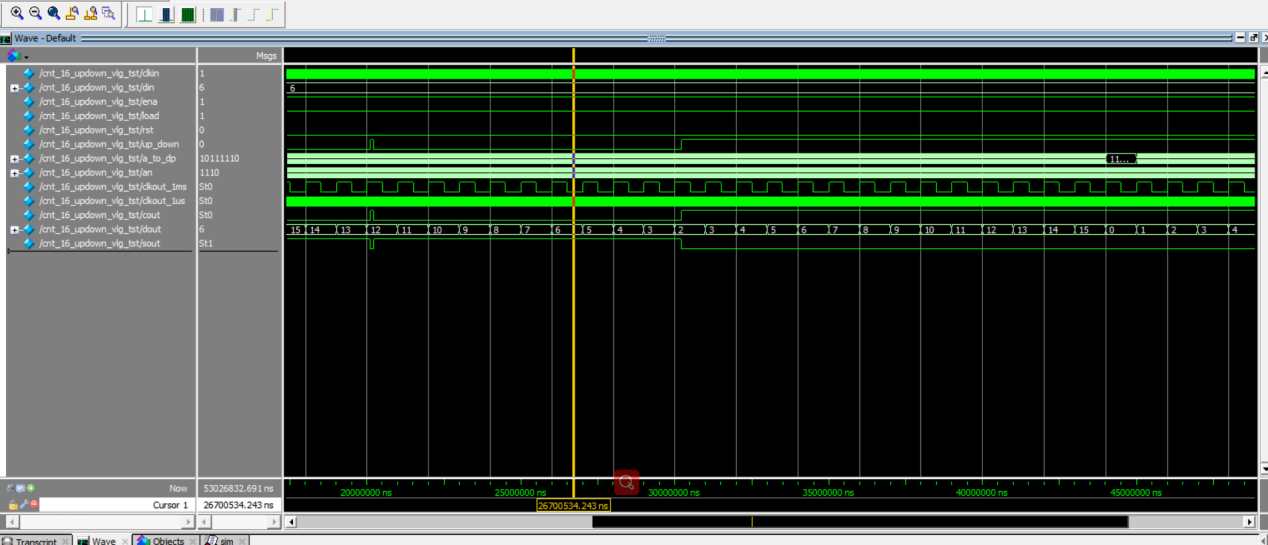
endmodule

4.仿真波形

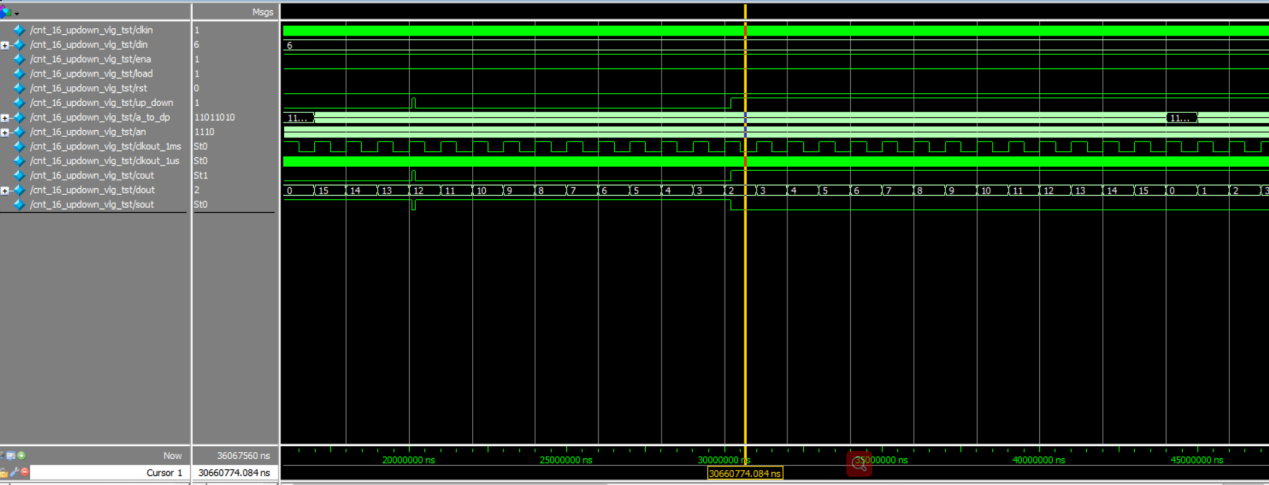
1.数据加载有效时(rst=0,load=0,din=6）



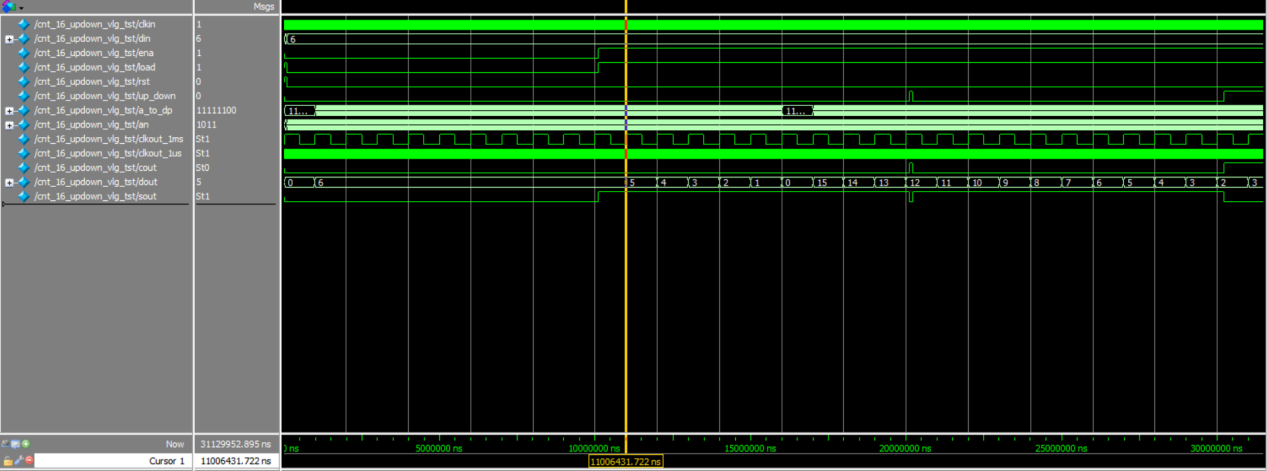
1. 使能有效时（rst=0,load=1,ena=1）



3.递增计数cout为1时（rst=0,load=1,ena=1,up\_down=1.）



4.递减计数sout为1时（rst=0,load=1,ena=1,up\_down=0.）



1. 加减可控双十进制计数器

1. 顶层文件

2. 计数器模块代码

3. 测试文件代码

4.仿真波形

1.数据加载有效时(rst=0,load=0,din=6）

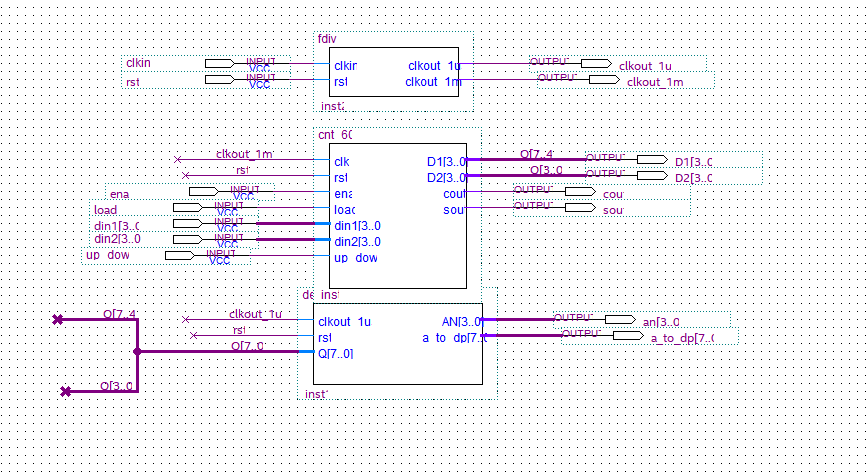
2.使能有效时（rst=0,load=1,ena=1）

3.递增计数cout为1时（rst=0,load=1,ena=1,up\_down=1.）

4.递减计数sout为1时（rst=0,load=1,ena=1,up\_down=0.）

（5）实验扩展部分：加减可控60进制计数器的设计（秒表）

1. 顶层文件



2. 计数器模块代码

3. 测试文件代码

4.仿真波形

1.数据加载有效时(rst=0,load=0,din=6）

2.使能有效时（rst=0,load=1,ena=1）

3.递增计数cout为1时（rst=0,load=1,ena=1,up\_down=1.）

4.递减计数sout为1时（rst=0,load=1,ena=1,up\_down=0.）