

- 5-6 把例 5-15 改成一个异步清零,同步时钟使能和异步数据加载型 8 位二进制加法计数器。
 5-7 对 5-6 的设计稍作修改,将其进位输出 COUT 与异步加载控制 LOAD 连在一起,构成一个自动加载型 16 位二进制计数器,即一个 16 位可控的分频器。设输入频率 $f_i=4\text{MHz}$,输出频率 $f_0=(516.5\pm 1)\text{Hz}$,求 16 位加载数值。
 5-10 用 Verilog 设计一个功能类似 74LS160 的计数器。
 5-11 给出含有异步清零和计数使能的 16 位二进制加减可控计数器的 Verilog 描述

解:

(1) 解题思路

5-6 解题思路:

参考书上例 5-15 的代码。需要修改的地方在于,输入 DATA 和输出 DOUT 宽度为 8,最大为 8b' 11111111.另外因为要求异步数据加载,所以 LOAD 加入 always 语句,取上升沿。**这里我取了上升沿,一般来说应该取下降沿。**

5-7 解题思路:

参考 5-6,将 LOAD 由 input 改为 wire, assign $\text{LOAD}=(Q1==0)$ 即可实现自动加载。另外需要注意将宽度改为 16. 这里 $\text{LOAD}=\text{COUT}$ 也可

$(\text{FFFF}-X)*T_i=T_o$ 得到 X 即加载数值为 $\text{FFFF}-7744=88\text{BB}$

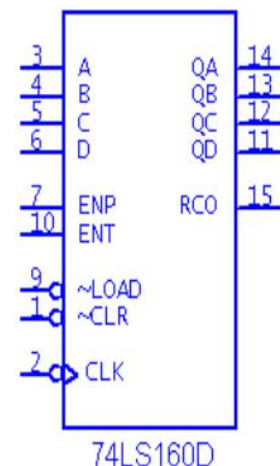
5-10 解题思路:

74LS160 为同步十进制加法器。

管脚图和功能如图:

1、74LS160 为可预置的十进制同步计数器,其管脚图如图所示:

- RCO 进位输出端
- ENP 计数控制端
- QA-QD 输出端
- ENT 计数控制端
- CLK 时钟输入端
- CLR 异步清零端(低电平有效)
- LOAD 同步并行置入端(低电平有效)



ENP,ENT 都为 1 时计数,否则保持。

LOAD 在时钟到来时才会加载。

根据其功能设计即可。

5-11 解题思路:

设置一个输入 `ctl`。`ctl=1` 进行加操作，为 0 减操作。当输出 `DOUT=FFFF` 且加一时输出 `COUT` 表示进位；`DOUT=0000` 且要-1 时 `COUT` 表示借位。其他的参考 5-7 的 16 位计数器即可。

(2) 核心模块代码

5_6

二进制 8 位加法计数器 CNT2_8

```
module CNT2_8(CLK, RST, EN, LOAD, COUT, DOUT, DATA);
input CLK, RST, EN, LOAD;
input[7:0] DATA;
output [7:0] DOUT;
output COUT;
reg[7:0] Q1;
reg COUT;
assign DOUT=Q1;
always @ (posedge CLK or negedge RST or posedge LOAD)
begin
if(!RST) Q1 <= 8'b00000000;
else if(LOAD) Q1 <= DATA;
else if(EN) begin
if(Q1<8'b11111111) Q1<=Q1+1;
else Q1<=8'b00000000; end //<=
end

always @ (Q1)
if(Q1==8'b11111111) COUT = 1'b1; // =
else COUT=1'b0;

endmodule
```

5-7

自动加载型 16 位 2 进制加法计数器

```
module CNT16(CLK, RST, EN, COUT, DOUT, DATA);
input CLK, RST, EN;
input[15:0] DATA;
output [15:0] DOUT;
output COUT;
reg[15:0] Q1;
reg COUT;
wire LOAD;
assign DOUT=Q1;
```

```

always @ (posedge CLK or negedge RST or posedge LOAD)
begin
if(!RST) Q1 <= 16'b0000000000000000;
else if(LOAD) Q1 <= DATA;
else if(EN) begin
    if(Q1<16'b1111111111111111) Q1<=Q1+1;
    else Q1<=16'b0000000000000000; end    //<=

end

assign LOAD=(Q1==0);

always @ (Q1)
if(Q1==16'b1111111111111111) COUT = 1'b1;    // =
else COUT=1'b0;

Endmodule

```

5-10

类似 74LS160

```

module LS160(CLK, RST, ENP, ENT, LOAD, COUT, DOUT, DATA) ;
input CLK, RST, ENP, ENT, LOAD;
input[3:0] DATA;
output [3:0] DOUT;
output COUT;
reg[3:0] Q1;
reg COUT;
assign DOUT=Q1;
always @ (posedge CLK or negedge RST)
begin
if(!RST) Q1 <= 4'b0000;
else if(ENP && ENT) begin
if(!LOAD) Q1 <= DATA;
else if(Q1<4'b1001) Q1<=Q1+1;
    else Q1<=4'b0000; end    //<=

end

always @ (Q1)
if(Q1==4'b1001) COUT = 1'b1;    // =
else COUT=1'b0;

Endmodule

```

5-11

16 位加减可控二进制计数器

```
module ADDSUB16(ct1, CLK, RST, EN, LOAD, COUT, DOUT, DATA);
input CLK, RST, EN, LOAD, ct1;
input[15:0] DATA;
output [15:0] DOUT;
output COUT;
reg[15:0] Q1;
reg COUT;
assign DOUT=Q1;
always @ (posedge CLK or negedge RST)
begin
if(!RST) Q1 <= 16'b0000000000000000;
else if(EN)
begin
if(!LOAD) Q1=DATA;
else if(ct1)
begin
if(Q1<16'b1111111111111111) Q1<=Q1+1;
else Q1<=16'b0000000000000000;
end
else begin
if(Q1>16'b0000000000000000) Q1<=Q1-1;
else Q1=16'b1111111111111111;
end
end
end

always @ (Q1)
if((Q1==16'b1111111111111111)&&ct1) COUT = 1'b1; // =
else if((Q1==16'b0000000000000000)&&!ct1) COUT=1'b1;
else COUT=1'b0;

endmodule
```

(3) 测试模块代码

5_6

二进制 8 位加法计数器 CNT2_8

```
`timescale 1ns/1ps
module CNT2_8_test_tb;
```

```

reg CLK, RST, EN, LOAD;
reg [7:0] DATA;
wire [7:0] DOUT;
wire COUT;

CNT2_8 DUT (
    .CLK(CLK),
    .RST(RST),
    .EN(EN),
    .LOAD(LOAD),
    .COUT(COUT),
    .DOUT(DOUT),
    .DATA(DATA)
);

initial begin

    DATA = 0;
    CLK=1'b0;
    EN=1'b0;
    RST=1'b1;
    LOAD=1'b0;

    #60;
    DATA = 8'b00000001;
    #60;
    DATA = 8'b00000010;
    #60;

    DATA = 8'b00000100;
    #60;
    RST=~RST;
    #20
    RST=~RST;
    DATA = 8'b00001000;
    #60;
    LOAD=~LOAD;
    #20
    LOAD=~LOAD;
    #10
    DATA = 8'b00010000;
    #60;

```

```

        DATA = 8'b00100000;
        #60;

        DATA = 8'b01000000;
        #60;

        DATA = 8'b10000000;
    end
    always #10 CLK=~CLK;
    //always #200 RST=~RST;
    always #30 EN=~EN;
    //always #300 LOAD=~LOAD;

Endmodule

```

5-7

自动加载型 16 位 2 进制加法计数器

```

`timescale 1ns/1ps
module CNT16_test_tb;

```

```

    reg CLK,RST,EN;
    reg [15:0] DATA;
    wire [15:0] DOUT;
    wire COUT;

```

```

    CNT16 DUT (
        .CLK(CLK),
        .RST(RST),
        .EN(EN),
        .COUT(COUT),
        .DOUT(DOUT),
        .DATA(DATA)
    );

```

```

    initial begin

```

```

        DATA = 0;
        CLK=1'b0;
        EN=1'b0;
        RST=1'b1;

```

```

        #60;
        DATA = 16'b0000000000000001;
        #60;
        DATA = 16'b0000000000000010;
        #60;

        DATA = 16'b0000000000000100;
        #60;
        RST=~RST;
        #20
        RST=~RST;
        DATA = 16'b0000000000001000;
        #60;
        DATA = 16'b0000000000010000;
        #60;

        DATA = 16'b0000000000100000;
        #60;

        DATA = 16'b0000000001000000;
        #60;

        DATA = 16'b0000000010000000;
end
    always #10 CLK=~CLK;
    //always #200 RST=~RST;
    always #30 EN=~EN;
    //always #300 LOAD=~LOAD;

endmodule

```

5-10

类似 74LS160

```

`timescale 1ns/1ps
module LS160_test_tb;

```

```

    reg CLK,RST,ENP,ENT,LOAD;
    reg [3:0] DATA;
    wire [3:0] DOUT;

```

```

wire COUT;

LS160 DUT (
    .CLK(CLK),
    .RST(RST),
    .ENP(ENP),
    .ENT(ENT),
    .LOAD(LOAD),
    .COUT(COUT),
    .DOUT(DOUT),
    .DATA(DATA)
);

initial begin
    DATA = 0;
    CLK=1'b0;
    ENP=1'b1;
    ENT=1'b1;
    RST=1'b1;
    LOAD=1'b1;
    #60;
    DATA = 4'b0001;
    #60;
    DATA = 4'b0100;
    #60;
    RST=~RST;
    #20
    RST=~RST;
    DATA = 4'b0011;
    #60;
    LOAD=~LOAD;
    #250
    LOAD=~LOAD;
    #10
    DATA = 4'b0100;
    #60;
    DATA = 4'b0101;
end

always #10 CLK=~CLK;
always #30 ENP=~ENP;
always #20 ENT=~ENT;

endmodule

```


5-11

16 位加减可控二进制计数器

```
`timescale 1ns/1ps
module ADDSUB16_test_tb;

    reg CLK, RST, EN, ctl, LOAD;
    reg [15:0] DATA;
    wire [15:0] DOUT;
    wire COUT;

    ADDSUB16 DUT (
        .ctl(ctl),
        .CLK(CLK),
        .RST(RST),
        .EN(EN),
        .LOAD(LOAD),
        .COUT(COUT),
        .DOUT(DOUT),
        .DATA(DATA)
    );

    initial begin
        ctl=1'b0;
        DATA = 0;
        CLK=1'b0;
        EN=1'b0;
        RST=1'b1;
        LOAD=1'b1;

        #60;
        DATA = 16'b0000000000000001;
        #60;
        DATA = 16'b0000000000000010;
        #60;

        DATA = 16'b0000000000000100;
        #60;
        RST=~RST;
        #20
        RST=~RST;
        DATA = 16'b0000000000011000;
```

```

        #60;
        LOAD=~LOAD;
        #60
        LOAD=~LOAD;
        #10
        DATA = 16'b00000000000010000;
        #6000;
        ctl=~ctl;
        #40000
        ctl=~ctl;

    end

    always #10 CLK=~CLK;
    //always #200 RST=~RST;
    always #30 EN=~EN;
    //always #300 LOAD=~LOAD;

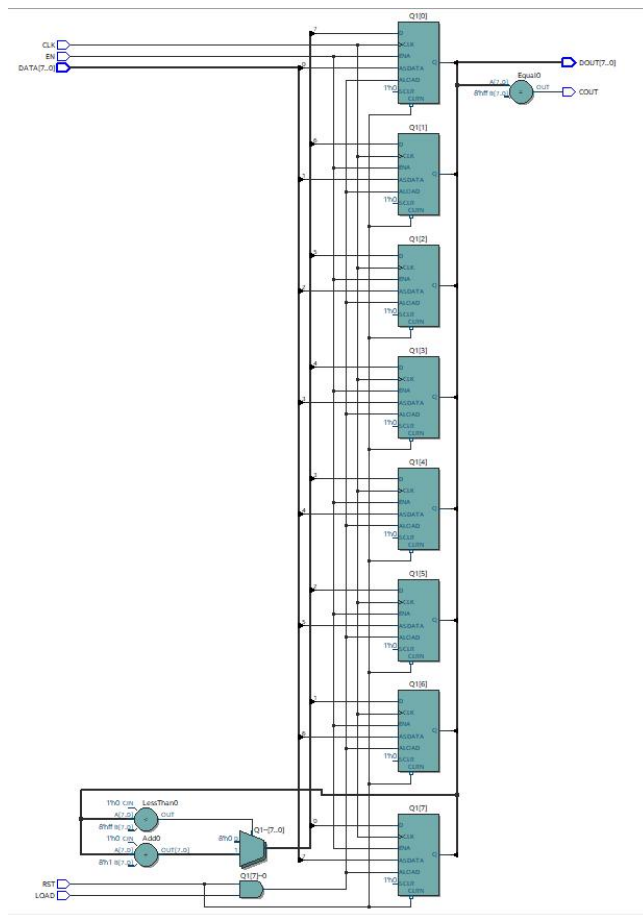
endmodule

```

(4) RTL View 的网表图

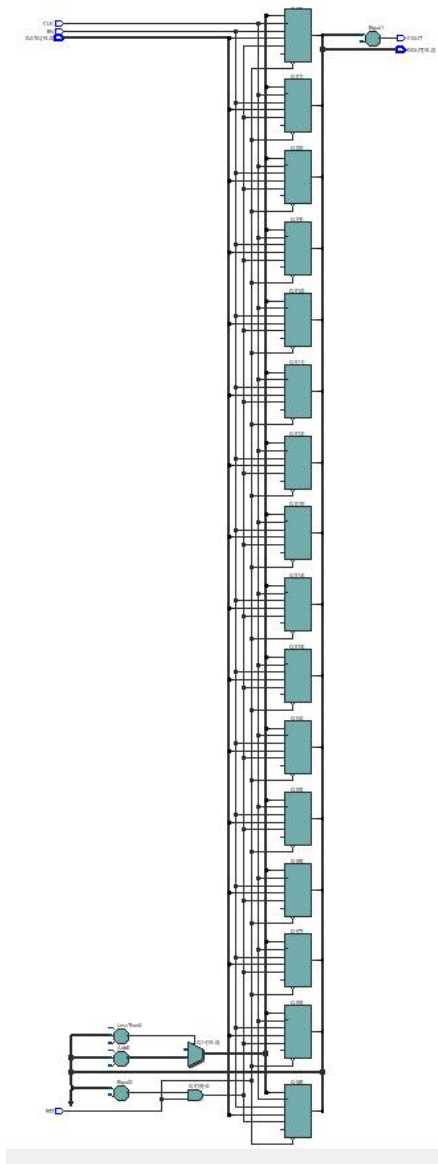
5_6

二进制 8 位加法计数器 CNT2_8



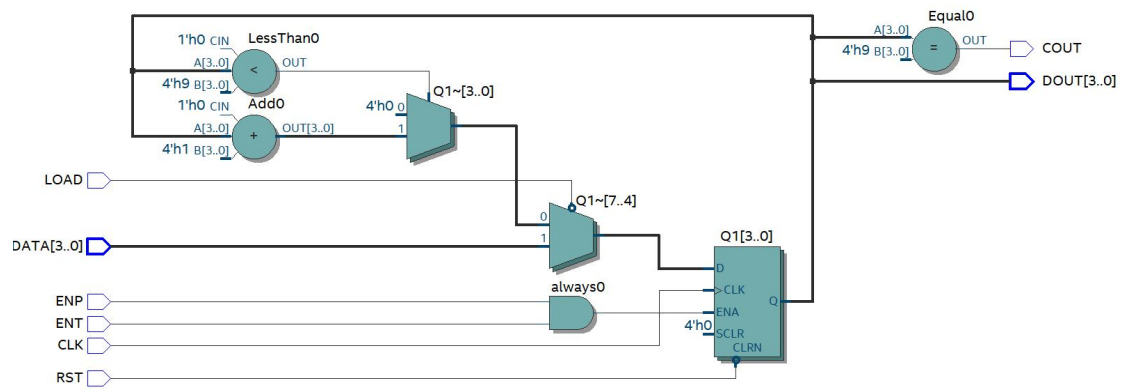
5-7

自动加载型 16 位 2 进制加法计数器



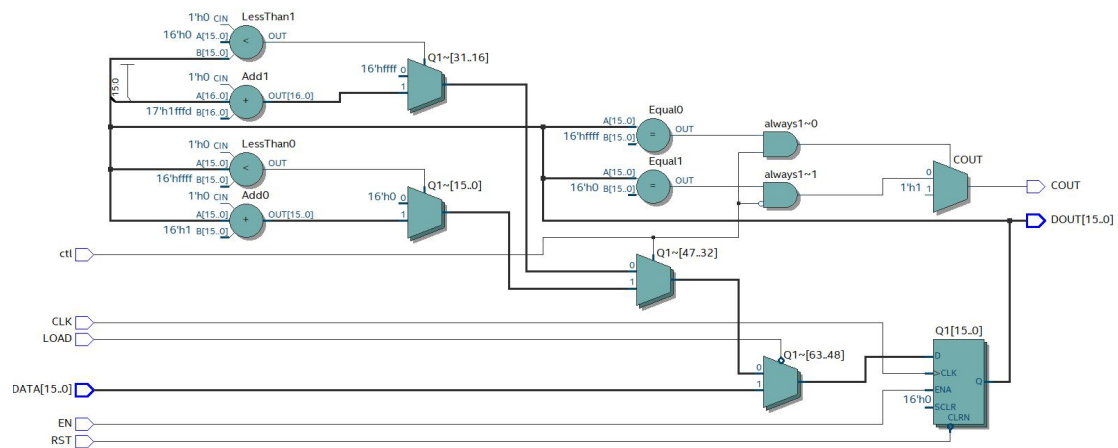
5-10

类似 74LS160



5-11

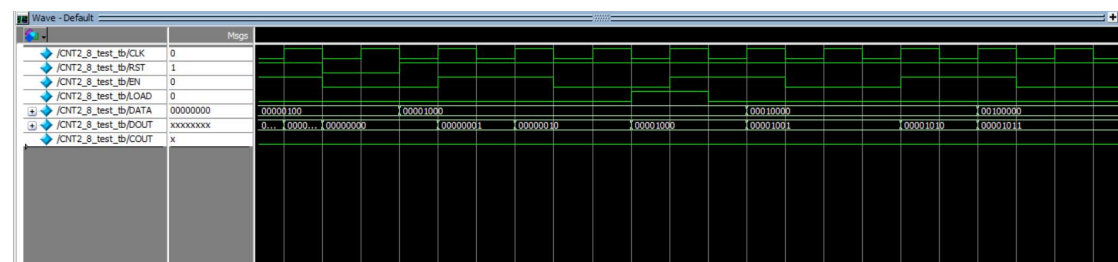
16 位加减可控二进制计数器



(5) ModelSim 仿真的结果图

5_6

二进制 8 位加法计数器 CNT2_8



5-7

自动加载型 16 位 2 进制加法计数器

两个图，分别展示 RST 和 LOAD(COUT)

