

階層化設計

4I24 中川寛之

目標

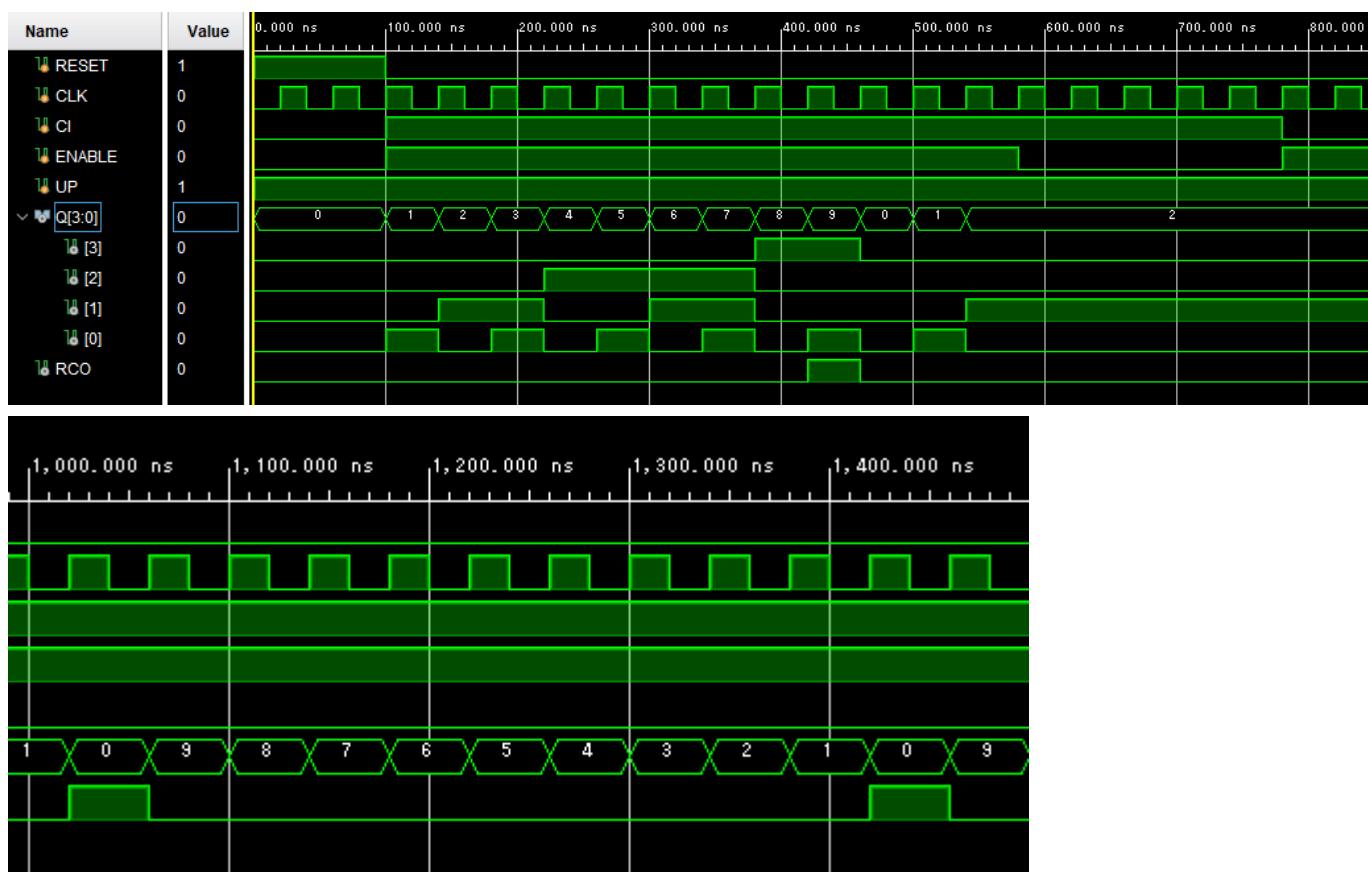
- 演習1：10進アップダウンカウンタを構成し、シミュレーションで動作を確認する
- 演習2：6進アップダウンカウンタを構成し、動作を確認する
- 課題：演習1と2で作成した10進カウンタと6進カウンタを組み合わせて60進アップダウンカウンタを構成する

演習課題

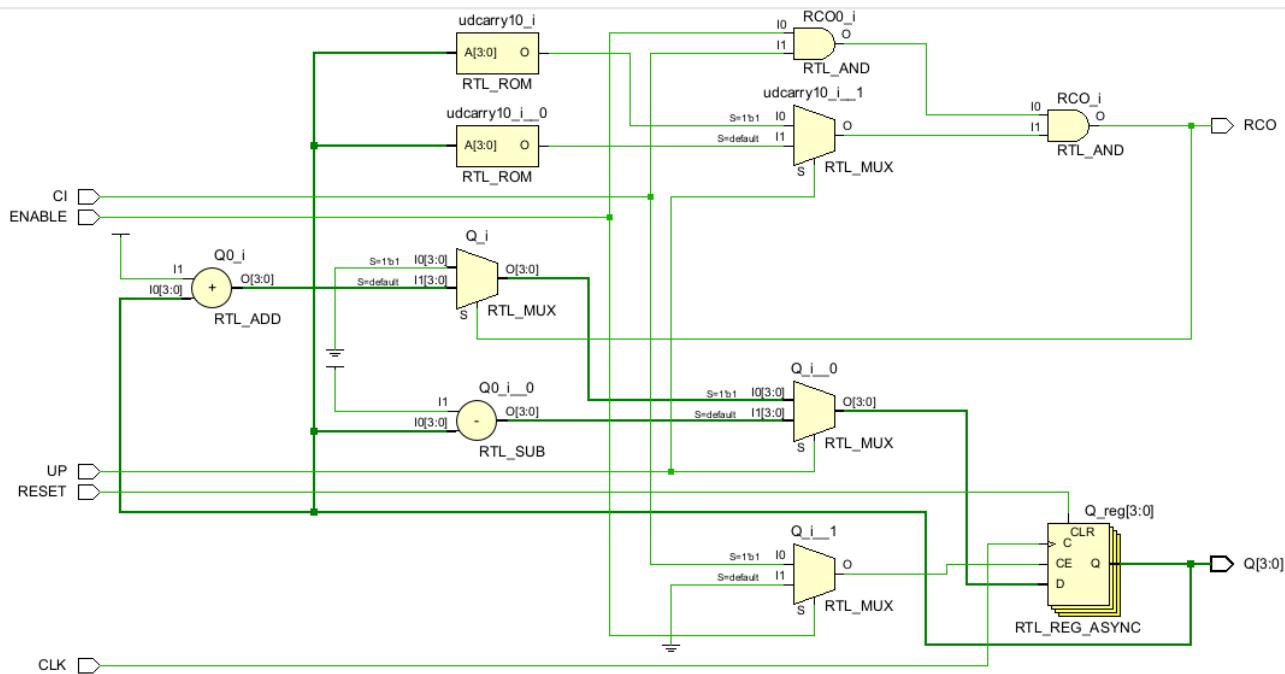
演習 1

10進アップダウンカウンタを構成し、シミュレーションで動作を確認しなさい

シミュレーション



回路図



Verilog HDLコード

```
<udcount10.v>
```

```
`timescale 1ns / 1ps

module count10(
    input RESET,
    input CLK,
    input CI,
    input ENABLE,
    input UP,
    output reg[3:0] Q = 4'd0,
    output RCO
);

function.udcarry10;
    input [3:0] Q;
    input UP;

    if(UP == 1'b1) begin //アップカウント
        if(Q == 4'd9) udcarry10 = 1'b1;
        else udcarry10 = 1'b0;
    end else begin
        if(Q == 4'd0) udcarry10 = 1'b1;
        else udcarry10 = 1'd0;
    end
endfunction

always @(posedge CLK or posedge RESET) begin
    if(RESET == 1'b1) begin
        Q <= 4'd0;
    end
end
```

```

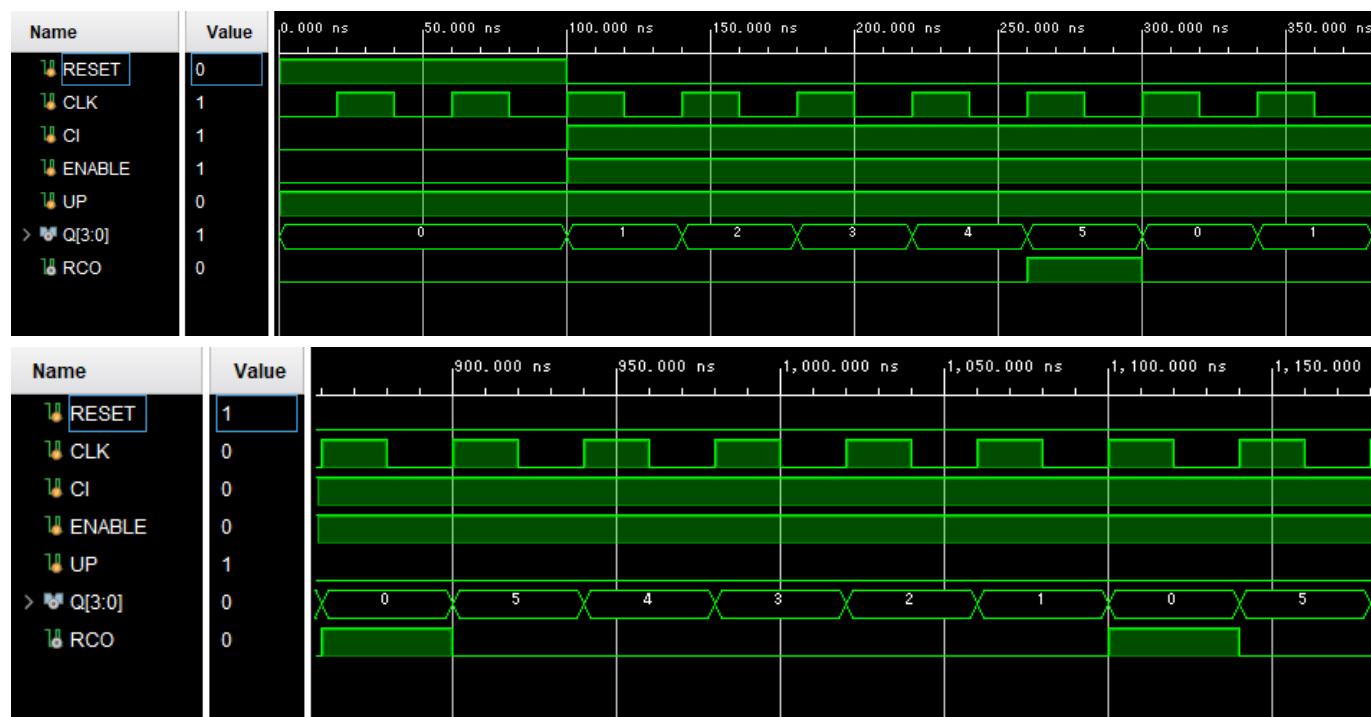
        else if(ENABLE == 1'b1 && CI == 1'b1) begin
            if(UP == 1'b1) begin
                if(Q == 4'd9) begin
                    Q <= 4'd0;
                end
                else begin
                    Q <= Q + 4'd1;
                end
            end
            else begin
                if(Q == 4'd0) begin
                    Q <= 4'd9;
                end
                else begin
                    Q <= Q - 4'd1;
                end
            end
        end
    end
    assign RCO = (ENABLE == 1'b1) && (CI == 1'b1) && udcarry10(Q, UP);
endmodule

```

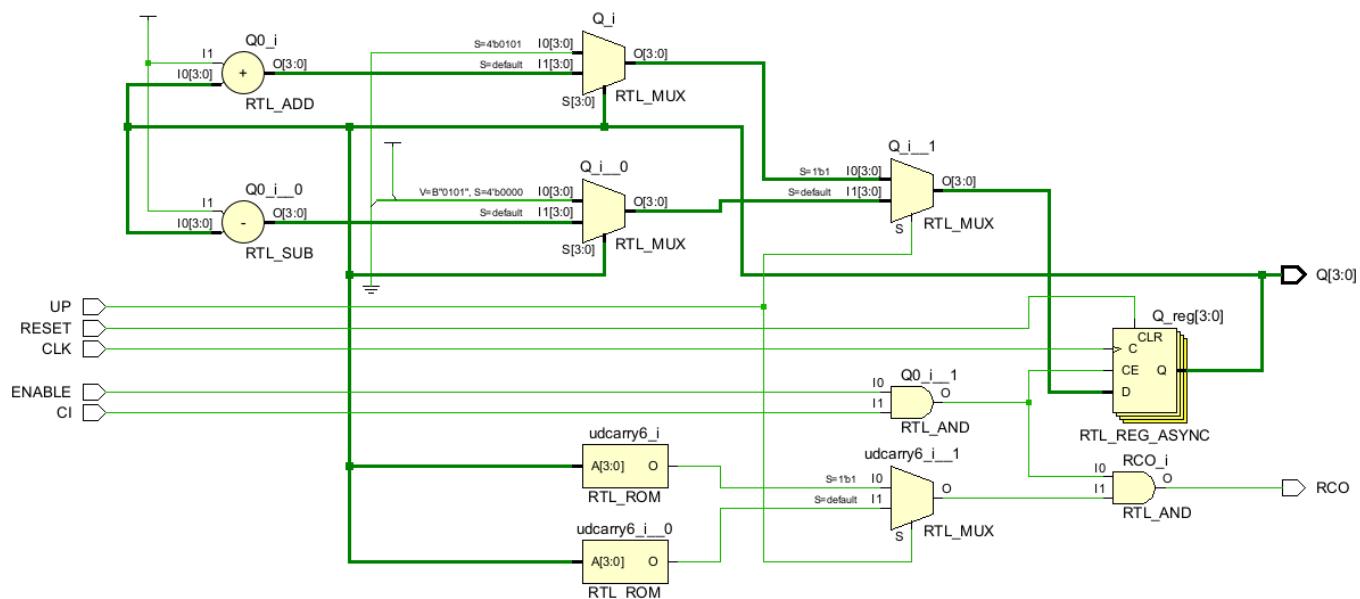
演習 2

6進アップダウンカウンタを構成し、動作を確かめなさい。

シミュレーション



回路図



Verilog HDLコード

<udcount6.v>

```

`timescale 1ns / 1ps

module count6(
    input RESET,
    input CLK,
    input CI,
    input ENABLE,
    input UP,
    output reg[3:0] Q = 4'd0,
    output RCO
);
function udcarry6;
    input [3:0] Q;
    input UP;

    if(UP == 1'b1) begin
        if(Q == 4'd5) udcarry6 = 1'b1;
        else udcarry6 = 1'b0;
    end else begin
        if(Q == 4'd0) udcarry6 = 1'b1;
        else udcarry6 = 1'd0;
    end
endfunction

always @(posedge CLK or posedge RESET) begin
    if(RESET == 1'b1) begin
        Q <= 4'd0;
    end
    else if(ENABLE == 1'b1 && CI == 1'b1) begin

```

```

        if(UP == 1'b1) begin
            if(Q == 4'd5) begin
                Q <= 4'd0;
            end
            else begin
                Q <= Q + 4'd1;
            end
        end
        else begin
            if(Q == 4'd0) begin
                Q <= 4'd5;
            end
            else begin
                Q <= Q - 4'd1;
            end
        end
    end
end
assign RC0 = (ENABLE == 1'b1) && (CI == 1'b1) && udcarry6(Q, UP);
endmodule

```

レポート課題

60進アップダウンカウンタを構成しなさい。

Verilog HDLコード

<udcount60.v>

```

`timescale 1ns / 1ps
module count60(
    input RESET,
    input CLK,
    input CI,
    input ENABLE,
    input UP,
    output [3:0] Q10,
    output [3:0] Q1,
    output RC0
);

    wire rco_1;
    wire rco_10;

    count10 U_10 (
        .RESET(RESET),
        .CLK(CLK),
        .CI(CI),
        .ENABLE(ENABLE),
        .UP(UP),
        .Q(Q1),

```

```

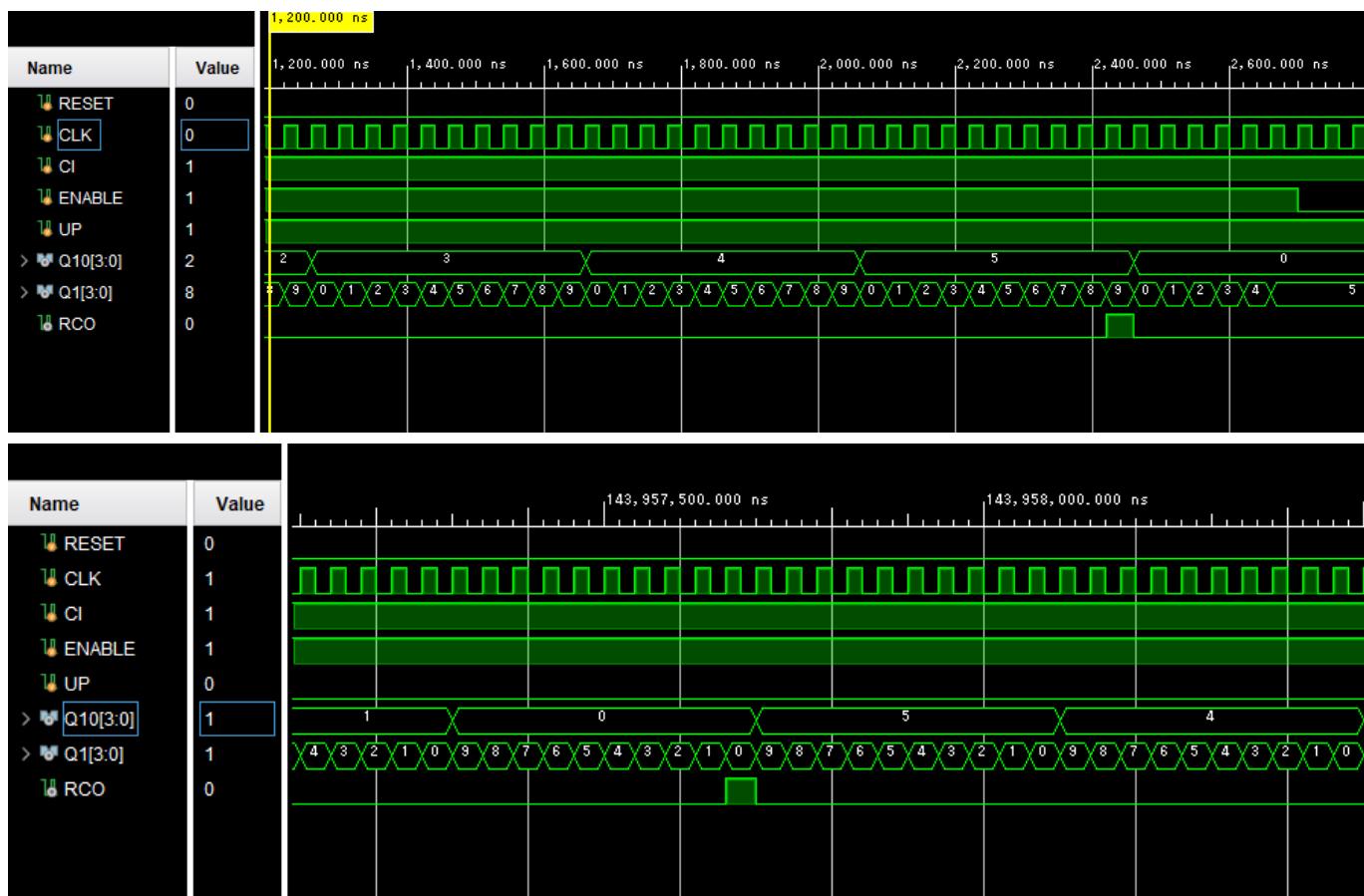
    .RCO(rco_1)
);

count6 U_6 (
    .RESET(RESET),
    .CLK(CLK),
    .CI(rco_1),
    .ENABLE(ENABLE),
    .UP(UP),
    .Q(Q10),
    .RCO(rco_10)
);
assign RCO = rco_1 && rco_10;

endmodule

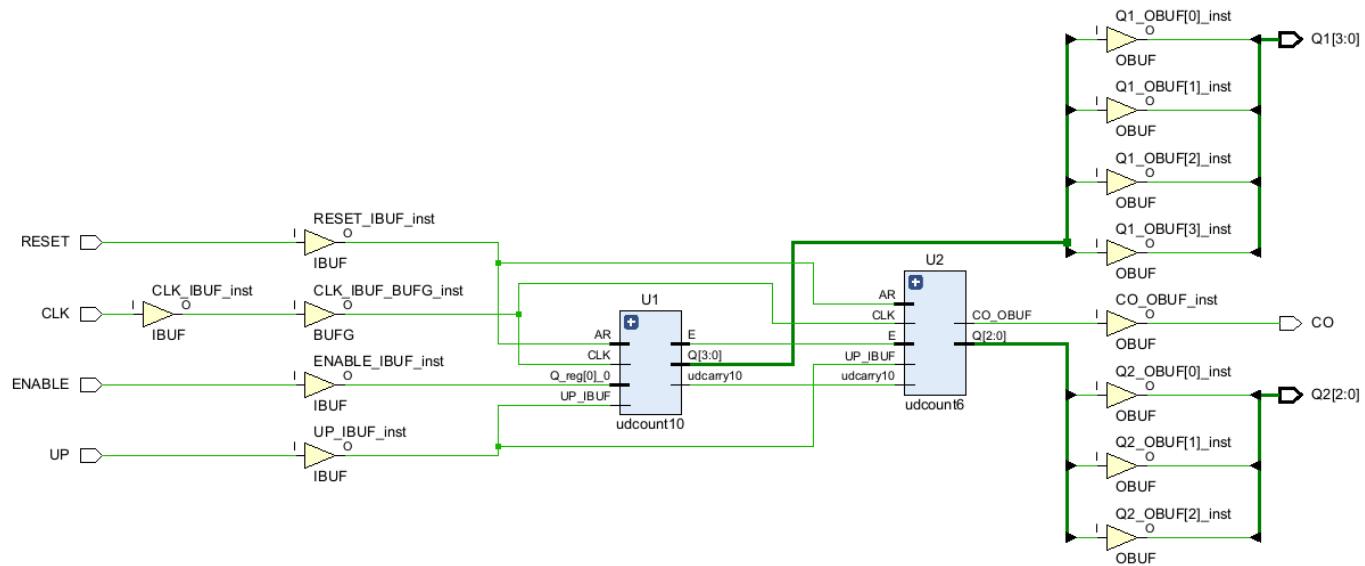
```

ビヘイビアシミュレーションで動作を確かめなさい。



シミュレーション結果から、キャリー出力が組み合 わせ回路で動作していることを確かめなさい。

回路図



udcount10/udcount6ともに回路図からも確認できているのでキャリー出力が演習で作成した2つの組み合わせ回路で動作していることが確認できた。

感想

今回の実験では10進と6進のアップダウンカウンタをVerilogで実装し、階層化して60進カウンタを構成、シミュレーションでキャリー出力による連鎖動作を確認した。

モジュール分割による設計の見通しの良さと再利用性が理解でき、リセットやイネーブル条件、タイミングに注意する重要性を実感した。