

سوال 1 (طراحی ALU)

```
module testcode1(A,B,S,C,co
);

    input [15:0]A,B;
    output [15:0]S;
    output C;
    input [2:0]co;
    wire [16:0]W;

    assign W=(co==0)? A+B:
    (co==1) ? A-B:
    (co==2) ? A>>1:
    (co==3) ? A<<1:
    (co==4) ? A&B:
    (co==5) ? A|B:
    (co==6) ? {A[0],A[15:1]}:{A[14:0],A[15]};

    assign S=W[15:0];
    assign C=W[16];

endmodule
```

سوال (مقایسه عملکرد 2 قطعه کد مطرح شده)

قطعه کد شماره یک

```
module testcode2(IN,Clk,Reset,Out);
```

```
    input [15:0]IN;
```

```
        input Reset,Clk;
```

```
        output reg[31:0]Out;
```

```
        always @ (posedge Clk or  
negedge Reset)
```

```
            if(!Reset)
```

```
                Out<=0;
```

```
            else
```

```
                Out<=Out+IN;
```

```
endmodule
```

TEST BENCH

```
initial begin
```

```
    IN=0;
```

```
    Clk=0;
```

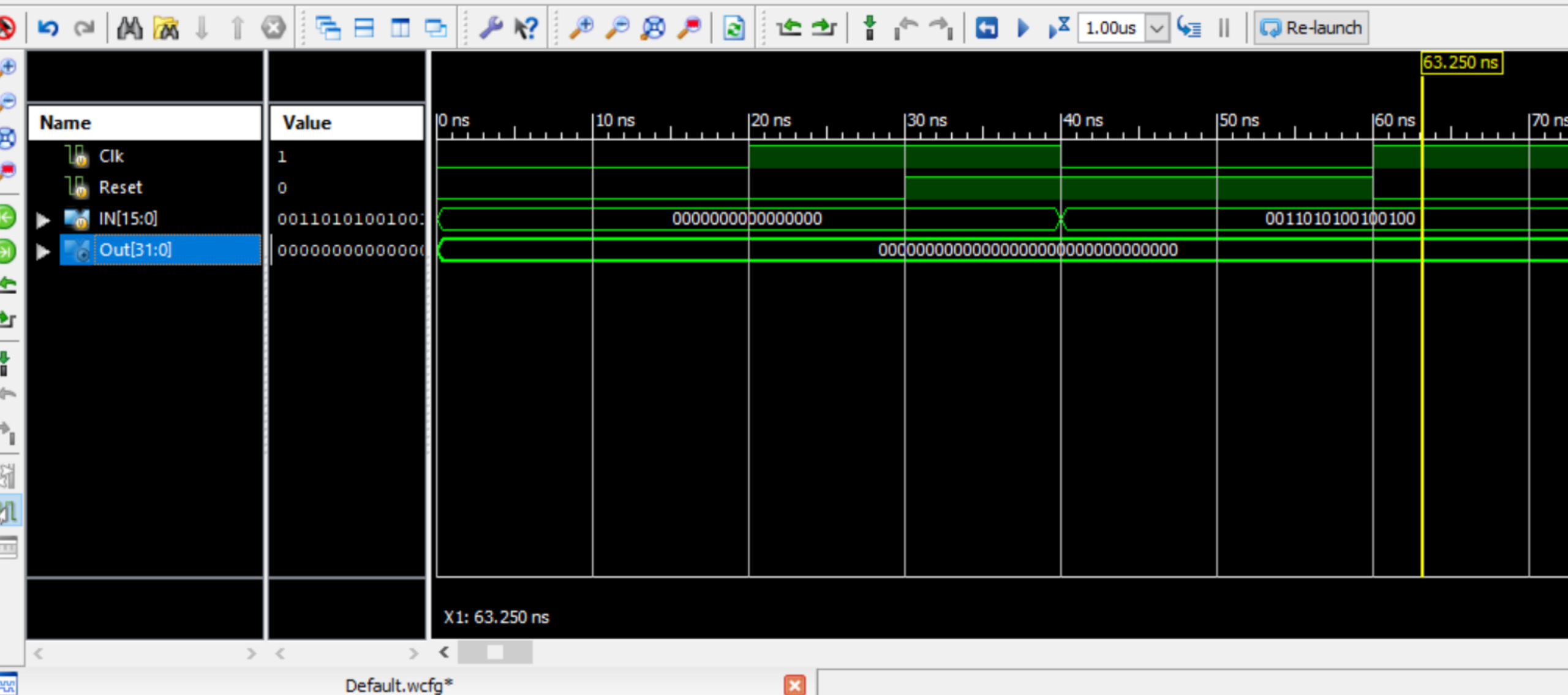
```
    Reset=0;
```

```
end
```

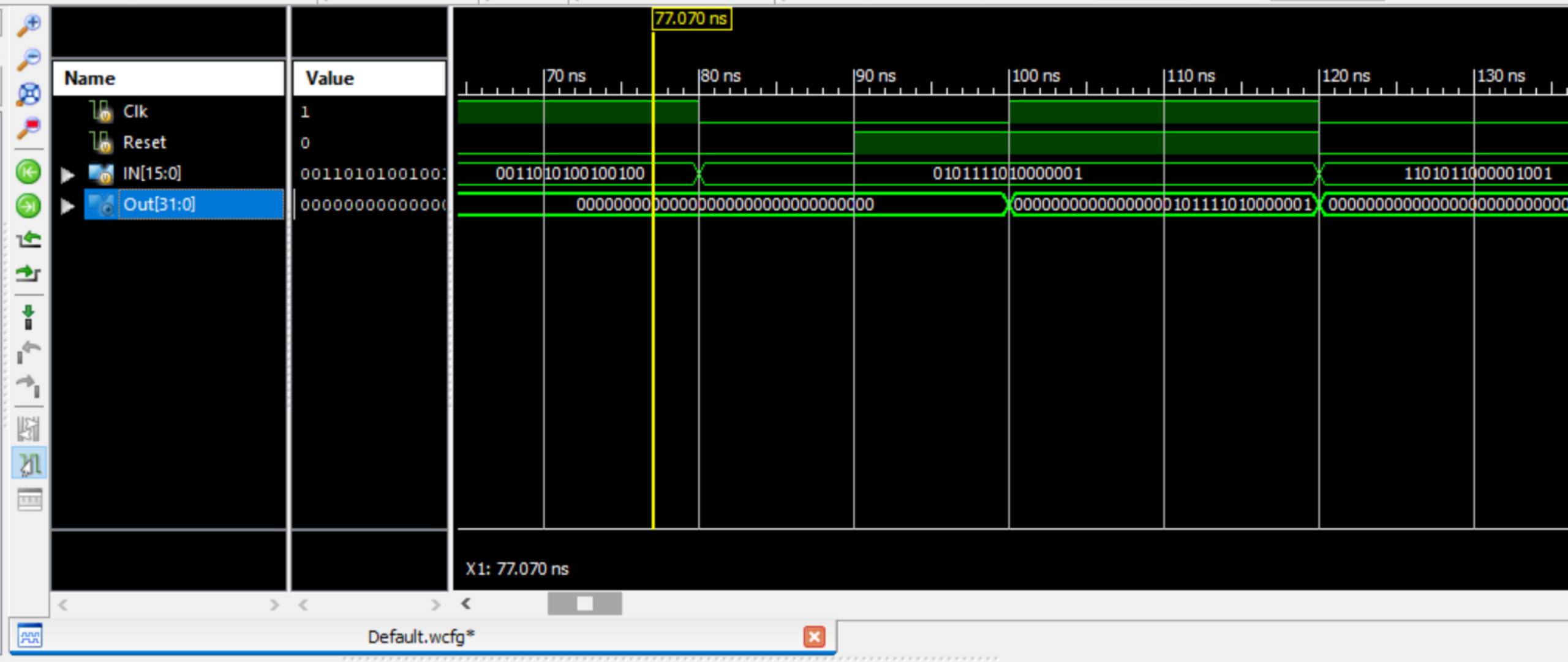
```
initial forever #40 IN=$random();
```

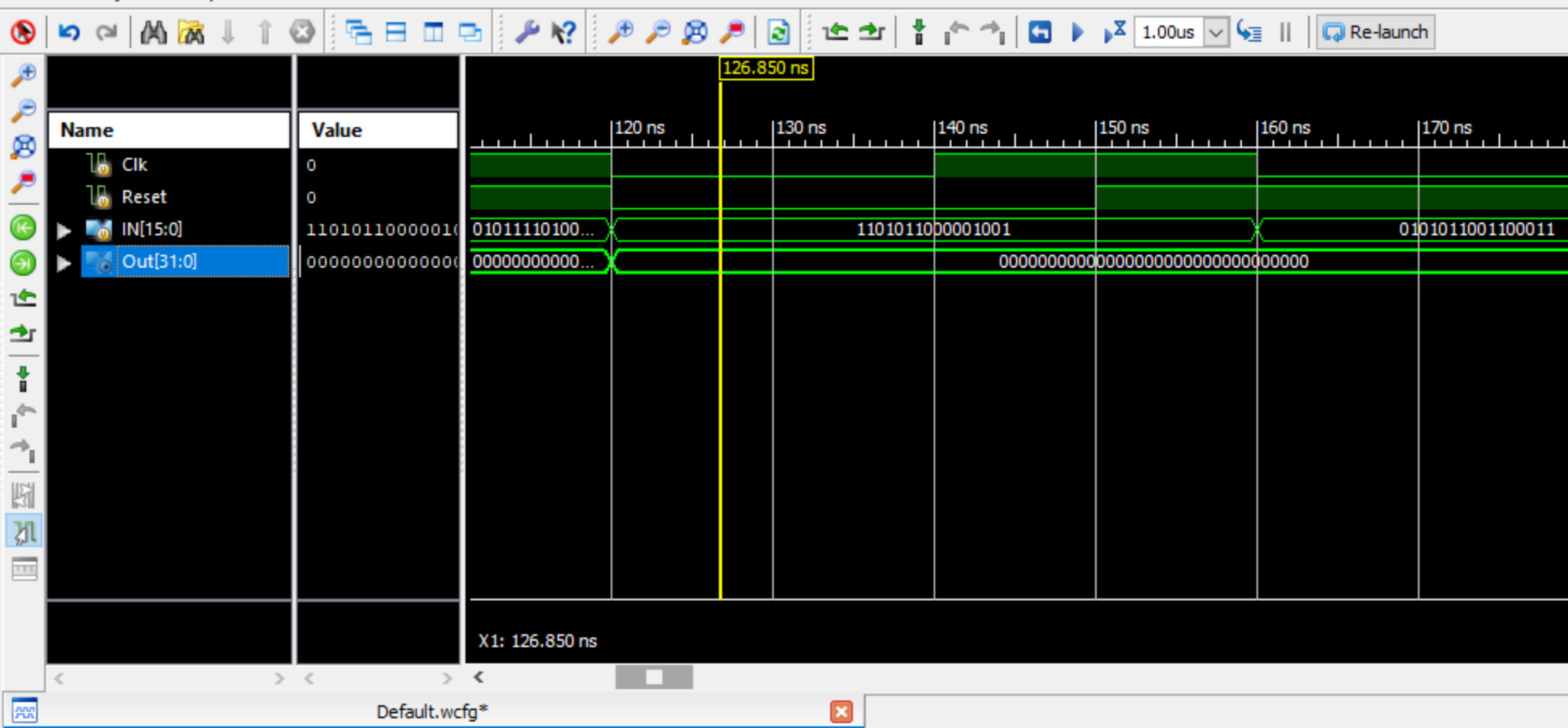
```
initial forever #20 Clk=~Clk;
```

```
initial forever #30 Reset=~Reset;
```



Default.wcfg*





in the same directory. Using database file name isim1.wdb.

قطعه کد شماره دو

```
module testcode3(IN,Clk,Reset,Out
);
    input [15:0]IN;
    input Reset;
    input Clk;
    output [31:0]Out;
    reg [31:0]R;
    always @(posedge Clk or negedge Reset)

    if(!Reset)
        R<=0;
    else

        R<=Out;

    assign Out=IN+R;
endmodule
```

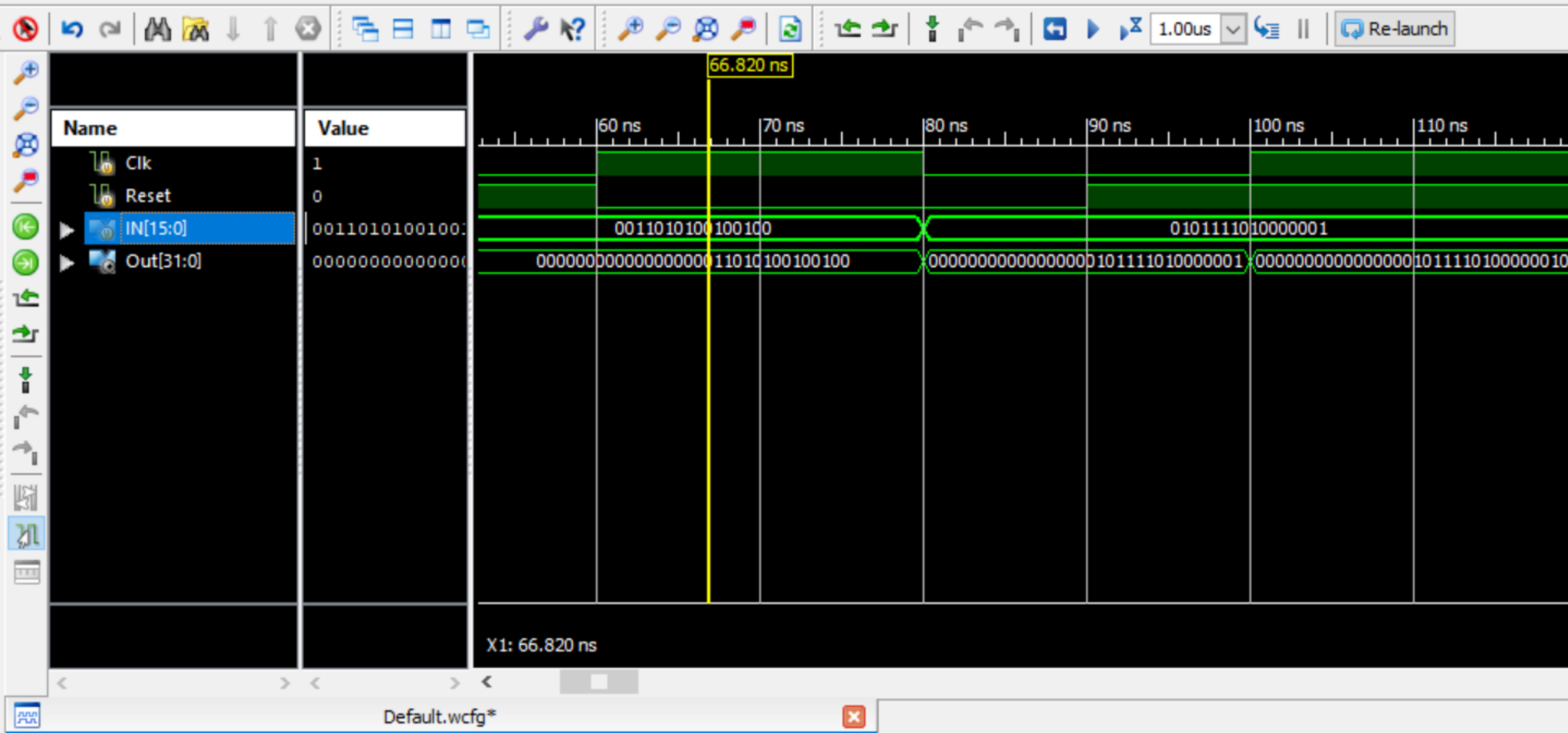
TEST BENCH

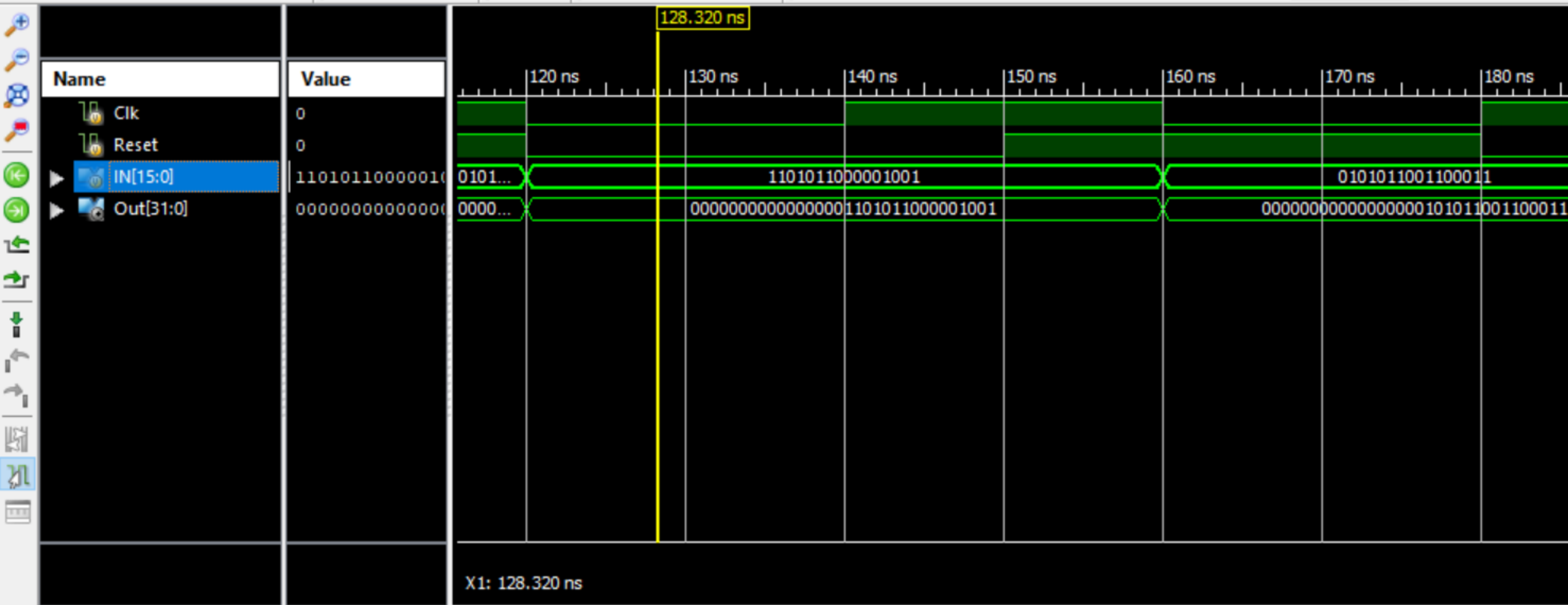
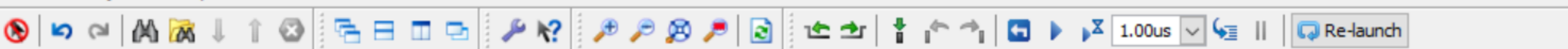
```
initial begin

    IN=0;
    Clk=0;
    Reset=0;

    end
    initial forever #40 IN=$random();
    initial forever #20 Clk=~Clk;
    initial forever #30 Reset=~Reset;
```







X1: 128.320 ns



مقایسه ی دو قطعه کد

در قطعه کد شماره یک هنگامیکه در پایه ی مثبت کلاک یا پایه ی منفی کلاک باشیم اگر Reset برابر صفر باشد که خروجی صفر می شود در غیر اینصورت خروجی حالت قبل با ورودی فعلی جمع شده و در خروجی فعلی ریخته می شود در قطعه کد شماره دو روند قطعه کد شماره یک تکرار شده با این تفاوت که در این کد از یک رجیستر استفاده شده که اگر Reset برابر با صفر باشد خروجی صفر می شود و در رجیستر ریخته می شود در غیر اینصورت خروجی حالت قبل در رجیستر ریخته می شود و رجیستر با ورودی فعلی جمع شده و در خروجی فعلی ریخته می شود

```
module testcode4(I,valid,Q
);
    input [3:0]I;
    output reg valid;
    output reg[1:0]Q;

    always @(*)

    if(I[3]==1) begin

        valid<=1;
        Q<=3;

    end
    else

    if(I[2]==1) begin
        valid<=1;
        Q<=2;
    end

    else
    if(I[1]==1) begin
        valid<=1;
        Q<=1;
    end
    else
    if (I[0]==1) begin
        valid<=1;
        Q<=0;
    end
    else begin
        valid<=0;
        Q<=0;
    end

endmodule
```

سوال 3)طراحی یک Encoder با اولویت

سوال 4 قسمت (الف)(طراحی یک مقسم)

```
module kitchen(clk,out
);
    input clk;

    output out;
    reg[3:0]count=0;
    reg f;

    always @(posedge clk)
    if(count>10) begin
        count<=1;
        f<=0;
    end

    else if(count>5) begin
        f<=1;
        count<=count+1;
    end
    else begin
        f<=0;
        count<=count+1;
    end
    assign out=f;

endmodule
```

سوال 4 قسمت (ب) (طراحی یک مقسم)

```
module lastversion(cl,ou
);
    input cl;
    output ou;
    wire w;
    kitchen inso(.clk(cl),.out(w));
    kitchen ins1(.clk(w),.out(ou));

endmodule
```