

## شرح آزمایش

در این آزمایش بنا به این است که یک ضرب کننده جامع (به تعداد بیت های دلخواه) با استفاده از الگوریتم مذکور در صورت پروژه، نوشته شود.

### نحوه اجرای الگوریتم

با بررسی الگوریتم به این نتیجه می رسیم که این جمع کننده شامل چند حالت می باشد که به راحتی با ماشین حالت متناهی (FSM) قابل نمایش است.

این حالات به شرح زیر می باشند:

(با راه اندازی این جمع کننده با سیگنال  $start = 1$  به استیت ۰ می رویم)

(حاصل در product نگه داری می شود و مقدار اولیه آن برابر صفر می باشد.)

(multiplicand به علت اینکه شیفت داده می شود برای حفظ تمامی بیت های اصلی باید به اندازه product اکستند شود.)

۰- استیت آغازین: اگر کم ارزش ترین بیت multiplier برابر یک بود product را با multiplicand جمع کن

#### برو به استیت ۱

۱- multiplicand را یک بیت به چپ شیفت بده.

#### برو به استیت ۲

۲- multiplier را یک بیت به راست شیفت بده.

#### برو به استیت ۳

۳- چک کن که آیا به انتهای multiplier رسیدی و آن را کامل خواندی

#### اگر بله برو به استیت ۴

در غیر این صورت برو به استیت ۰ (ورود دوباره به حلقه تا برآورده شدن شرط  $\leftarrow$  این کار توسط یک شبه شمارنده (با شیفت دادن counter) صورت می گیرد.)

۴- استیت پایانی (پایان عملیات ضرب)

۸- استیت بی اثر

## کد اصلی :

```
module mult
    #(parameter N = 32) (input reset, start, clk,
        input[N-1:0] A,B,
        output[2*N-1:0] O, output finish);
    logic[2:0] State;
    logic[2*N-1:0] multiplicand, product;
    logic[N-1:0] multiplier;
    logic[N-1:0] counter,zeroExt;
    assign zeroExt = 0;
    assign O = product;
    assign finish = (State == 4)? 1'b1 : 1'b0;

    always_ff @(posedge reset or posedge start or posedge clk)
    begin
        if(start) begin //Start
            State <= 0;
            counter <= 1;
            multiplicand <= {zeroExt,A};
            multiplier <= B;
            product <= 0;
        end
        else if(reset) begin
            State <= 8; //useless state that do nothing
            counter <= 1;
        end
        else if(State==0) begin //Test multiplier0
            if(multiplier[0]==1'b1) begin //Add multiplicand to product and place the result in product register
                product <= product + multiplicand;
            end
            State <= State + 1;
        end
        else if(State==1) begin //Shift the multiplicand register Left 1 bit
            multiplicand <= {multiplicand[2*N-2:0], 1'b0};
            State <= State + 1;
        end
        else if(State==2) begin //Shift the multiplier register right 1 bit
            multiplier <= {1'b0, multiplier[N-1:1]};
            State <= State + 1;
        end
        else if(State==3) begin //32nd repetition?
            if(counter[N-1] == 1) begin //No: < 32 repetitions
```

```
        State <= 4;
    end
    else begin //Yes: 32 repetitions
        State <= 0;
        counter <= {counter[N-2:0], 1'b0};
    end
end
end
endmodule
```

## تست بنچ :

```
module test();
// signals
    reg start,reset,clk;
    reg[31:0] A1,B1,A0,B0;
    reg[15:0] A2,B2;
    reg[7:0] A3,B3;
    reg[3:0] A4,B4;

// Outputs
    wire[63:0] O1,O0;
    wire[31:0] O2;
    wire[15:0] O3;
    wire[7:0] O4;
    wire Finish0,Finish1,Finish2,Finish3,Finish4;

// device under test

    mult dut0(reset,start,clk, A0,B0,O0,Finish0);
    mult dut1(reset,start,clk, A1,B1,O1,Finish1);
    mult #(.N(16)) dut2(reset,start,clk, A2,B2,O2,Finish2);
    mult #(.N(8)) dut3(reset,start,clk, A3,B3,O3,Finish3);
    mult #(.N(4)) dut4(reset,start,clk, A4,B4,O4,Finish4);

    initial begin
        reset=0; start=0; clk = 0;

        A0 = 32'd24221; B0 = 32'd14867; //expect = 360,093,607
        A1 = 32'd1878781947; B1 = 32'd2114709627; //expect = 3,973,078,270,354,703,76
9
        A2 = 16'd23291; B2 = 16'd32763; //expect = 763,083,033
        A3 = 8'd61; B3 = 8'd26; //expect = 1,586
```

آزمایشگاه مدار منطقی - تمرین دوم  
متین زیودار - محمد خدام

```
A4 = 4'd6; B4 = 4'd6; //expect = 36
end

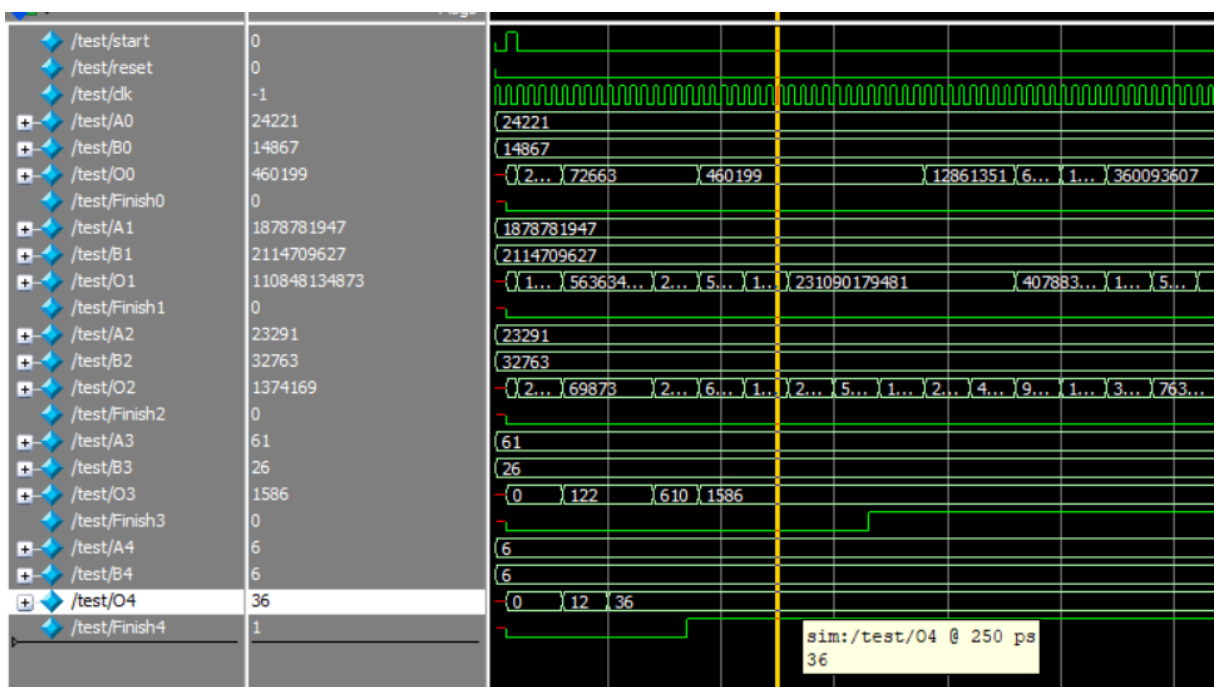
initial begin
#5 clk = !clk; #5 clk = !clk;
start = 1;
#5 clk = !clk; #5 clk = !clk;
start = 0;
end
always
begin
clk <= 1; # 5; clk <= 0; # 5;
end
endmodule
```

راستی آزمایی تست بنچ :

+ /test/O4	36	36
+ /test/O3	1586	1586
+ /test/O2	763083033	763083033
+ /test/O1	3973078270354703769	3973078270354703769
+ /test/O0	360093607	360093607

+ /test/start	0	
+ /test/reset	0	
+ /test/dk	-1	
+ /test/A0	24221	24221
+ /test/B0	14867	14867
+ /test/O0	360093607	360093607
+ /test/Finish0	1	
+ /test/A1	1878781947	1878781947
+ /test/B1	2114709627	2114709627
+ /test/O1	3973078270354703769	146622209804... {6...} {1...} {4...} {9...} {1...} {3973078270354703769}
+ /test/Finish1	1	
+ /test/A2	23291	23291
+ /test/B2	32763	32763
+ /test/O2	763083033	763083033
+ /test/Finish2	1	
+ /test/A3	61	61
+ /test/B3	26	26
+ /test/O3	1586	1586
+ /test/Finish3	1	
+ /test/A4	6	6
+ /test/B4	6	6
+ /test/O4	36	36
+ /test/Finish4	1	

آزمایشگاه مدار منطقی - تمرین دوم  
متین زیودار - محمد خدام



## سنتز و بررسی زمان

Command Info		Summary of Paths	
	Delay	From Node	To Node
1	12.300	A[4]	product[62]

## Slow 1200mV 85C Model Fmax Summary

	Fmax	Restricted Fmax	Clock Name	Note
1	130.31 MHz	130.31 MHz	clk	

آزمایشگاه مدار منطقی - تمرین دوم  
متین زیودار - محمد خدام

	Type	Count
1	boundary_port	132
2	▼ cycloneiii_ff	195
1	CLR	4
2	ENA CLR	191
3	▼ cycloneiii_lcell_comb	267
1	▼ arith	63
1	3 data inputs	63
2	▼ normal	204
1	1 data inputs	1
2	2 data inputs	67
3	3 data inputs	66
4	4 data inputs	70
4		
5	Max LUT depth	8.30
6	Average LUT depth	3.88

Resource	Usage
Estimated Total logic elements	328
Total combinational functions	267
▼ Logic element usage by number of LUT inputs	
-- 4 input functions	70
-- 3 input functions	129
-- <=2 input functions	68
▼ Logic elements by mode	
-- normal mode	204
-- arithmetic mode	63
▼ Total registers	195
-- Dedicated logic registers	195
-- I/O registers	0
I/O pins	132
Embedded Multiplier 9-bit elements	0
Maximum fan-out node	start~input
Maximum fan-out	289
Total fan-out	1774
Average fan-out	2.44

شکل RTL :

