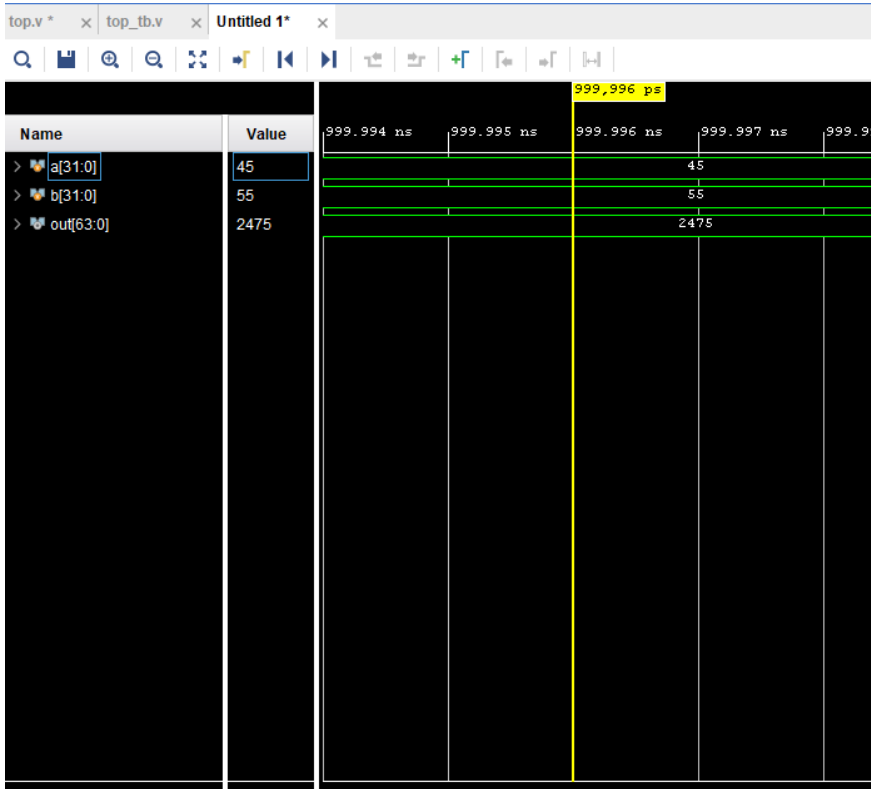


ASSİGNMENT 4

1)

4. Ödev de bizden 128 bitlik 2 sayıyı çarpmamız istendi. Uygulamanın bize yardımcı olabileceği (*) operatörünü kullanarak basit bir 128 bitlik çarpma işlemi gerçekleştirdim.



```

module top_tb();
    reg [31:0]a;
    reg [31:0]b;
    wire [63:0]out;

    top dut (.a(a), .b(b), .out(out));

    initial begin
        a=32'd45;
        b=32'd55;

    end

endmodule

```

```

top.v x top_tb.v x Untitled 1* x
C:/Users/Can/Desktop/homework_4/top/srcs/sources_1/new/top.v

1 module top(
2     input [31:0] a,
3     input [31:0] b,
4     output [63:0]out
5 );
6
7     assign out = a*b;
8
9
10
11 endmodule
12

```

- 2) Bu maddede ise bizden Karatsuba algoritması kullanarak 128 bitlik bir çarpma işlemini yapmamız istendi, öncelikli olarak basit bir çarpma modülü oluşturup ardından Karatsuba algoritmasını kullanabileceğim başka bir modül sayfası açıp bu iki modülü birbirine bağladım ve çarpa işlemini gerçekleştirdim.

```

module top(
input  [31:0] x,
input  [31:0] y,
output [63:0] out
);

wire [31:0] xlyl, xlyr, xryl, xryr;

karatsuba s1 (.X(x[31:16]), .Y(y[31:16]), .out (xlyl));
karatsuba s2 (.X(x[31:16]), .Y(y[15:0]), .out (xryl));
karatsuba s3 (.X(x[15:0]), .Y(y[31:16]), .out (xlyr));
karatsuba s4 (.X(x[15:0]), .Y(y[15:0]), .out (xryr));

assign out= (32'd4_294_967_296)*xlyl + (16'd65_536)*((xlyr)+(xryl))+xryr;

endmodule

```

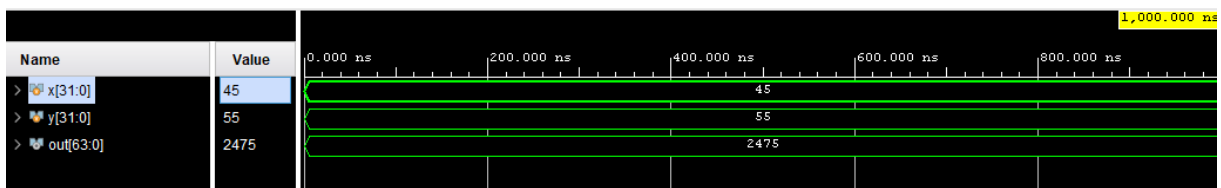
```

module karatsuba(
input wire [15:0] X,
input wire [15:0] Y,
output wire [63:0] out
);

assign out = X*Y;

endmodule

```



- 3) Bu maddede bizden d-flip-flop kullanarak kullanacağımız devreyi belli bir Hz değeri aralığında kullanabileceğimiz şekilde tasarlamamız istendi. Input girişlerine birer d flip flop eş zamanlı çalışacak şekilde konumlandırdıktan sonra bir adet de modül çıkışına yerleştirip 20 ns olacak şekilde bir delay elde.

$$f_{max} = \frac{1}{T} = \frac{1}{3,981ns} = 251MHz$$

```

C:/Users/Can/Desktop/homework_4/toplop.srcs/sources_1/newtop.v
1 module top(
2   input wire clk,
3   input wire [63:0] x,
4   input wire [63:0] y,
5   output [127:0] out
6 );
7
8
9   wire [63:0] xlyl, xlyr, xryl, xryr, out_xlyl, out_xryl, out_xlyr, out_xryr;
10  wire [127:0] fout;
11
12  karatsuba s1 (.X(x[63:32]), .Y(y[63:32]), .out (xlyl));
13  karatsuba s2 (.X(x[63:32]), .Y(y[31:0]), .out (xlyr));
14  karatsuba s3 (.X(x[31:0]), .Y(y[63:32]), .out (xryl));
15  karatsuba s4 (.X(x[31:0]), .Y(y[31:0]), .out (xryr));
16
17  dff s5 (.clk(clk), .D(xlyl), .Q(out_xlyl));
18  dff s6 (.clk(clk), .D(xlyr), .Q(out_xlyr));
19  dff s7 (.clk(clk), .D(xryl), .Q(out_xryl));
20  dff s8 (.clk(clk), .D(xryr), .Q(out_xryr));
21
22  assign fout = (32'd4294967296)*out_xlyl + (16'd5536)*((out_xlyr)+(out_xryl))+out_xryr;
23
24  dff s9 (.clk(clk), .D(fout), .Q(out));
25
26 endmodule
27
28

```

```

module karatsuba(
input wire [31:0] X,
input wire [31:0] Y,
output wire [127:0] out
);

assign out = X*Y;

endmodule

```

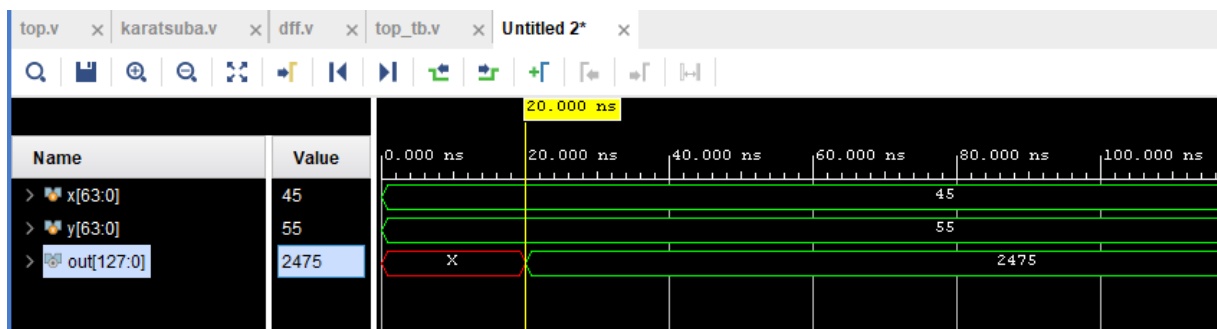
```

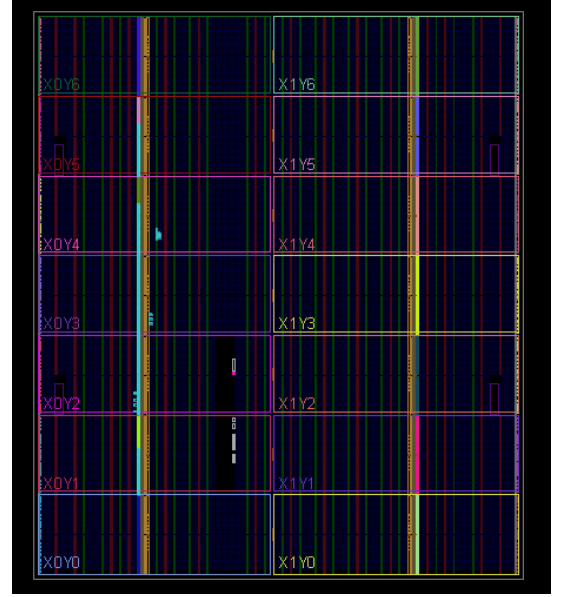
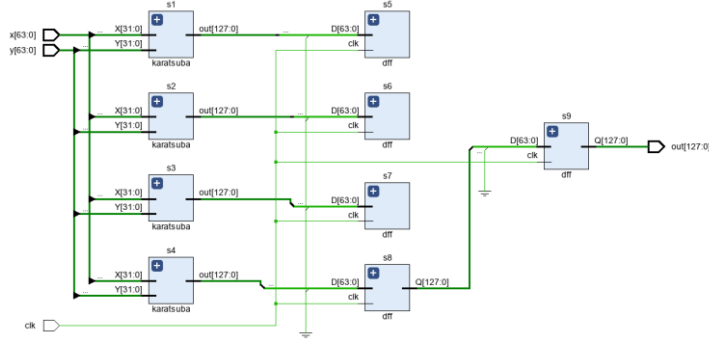
module dff(
input clk,
input [63:0] D,
output reg [127:0] Q
);

always @(posedge clk) begin
Q <= D;

end
endmodule

```





Unconstrained Paths - NONE - NONE - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	∞	2	3	1	x[16]	s8/Q_reg[0]_0/D	5.148	3.981	1.167	∞	input port clock			0.000
Path 2	∞	2	3	1	x[16]	s8/Q_reg[10]_0/D	5.148	3.981	1.167	∞	input port clock			0.000
Path 3	∞	2	3	1	x[16]	s8/Q_reg[11]_0/D	5.148	3.981	1.167	∞	input port clock			0.000
Path 4	∞	2	3	1	x[16]	s8/Q_reg[12]_0/D	5.148	3.981	1.167	∞	input port clock			0.000
Path 5	∞	2	3	1	x[16]	s8/Q_reg[13]_0/D	5.148	3.981	1.167	∞	input port clock			0.000
Path 6	∞	2	3	1	x[16]	s8/Q_reg[14]_0/D	5.148	3.981	1.167	∞	input port clock			0.000
Path 7	∞	2	3	1	x[16]	s8/Q_reg[15]_0/D	5.148	3.981	1.167	∞	input port clock			0.000

Unconstrained Paths - NONE - NONE - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	S
Path 11	∞	1	1	1	s8/Q_reg[0]_0/C	s9/Q_reg[0]_0/D	0.202	0.100	0.102	-∞	
Path 12	∞	1	1	1	s8/Q_reg[10]_0/C	s9/Q_reg[10]_0/D	0.202	0.100	0.102	-∞	
Path 13	∞	1	1	1	s8/Q_reg[11]_0/C	s9/Q_reg[11]_0/D	0.202	0.100	0.102	-∞	
Path 14	∞	1	1	1	s8/Q_reg[12]_0/C	s9/Q_reg[12]_0/D	0.202	0.100	0.102	-∞	
Path 15	∞	1	1	1	s8/Q_reg[13]_0/C	s9/Q_reg[13]_0/D	0.202	0.100	0.102	-∞	
Path 16	∞	1	1	1	s8/Q_reg[14]_0/C	s9/Q_reg[14]_0/D	0.202	0.100	0.102	-∞	
Path 17	∞	1	1	1	s8/Q_reg[15]_0/C	s9/Q_reg[15]_0/D	0.202	0.100	0.102	-∞	

4) 3 farklı modülün de yaşadıkları gecikmelere bakarak PIPELINE modülünün diğerlerine nazaran çok daha hızlı çalıştığını görmekteyiz.

	DELAY	FREQUENCY
MULTİPLİER	12,185	82MHz
KARATSUBA	9,001	111MHz
PIPELINE	3,981	251MHz