

به نام خدا

دانشکده مهندسی برق و کامپیوتر

گزارشکار پروژه سوم سیستم های دیجیتال 1

دکتر نوابی

دانشجو : فاطمه نائینیان

شماره دانشجویی: 810198479

بهار 1400

سوال اول

حل دستی:

Nmos #(3,4,5)

Pmos #(5,6,7)

Not #(5,7)

Nand #(10,8)

Si

To0= ci=0 , a=1 ,b=0 -> ci=0 , a=1 ,b=1 ->>> 43ns

7ns not+ 2*10ns nand +2*8ns nand = 43ns

To1= ci=1 , a=0 ,b=1 -> ci=1 , a=1 ,b=1 ->>> 48ns

7ns not+5ns not+ 2*10ns nand +2*8ns nand = 48ns

Co

To0= ci=1 , a=0 ,b=1 -> ci=1 , a=0 ,b=0 ->>> 36ns

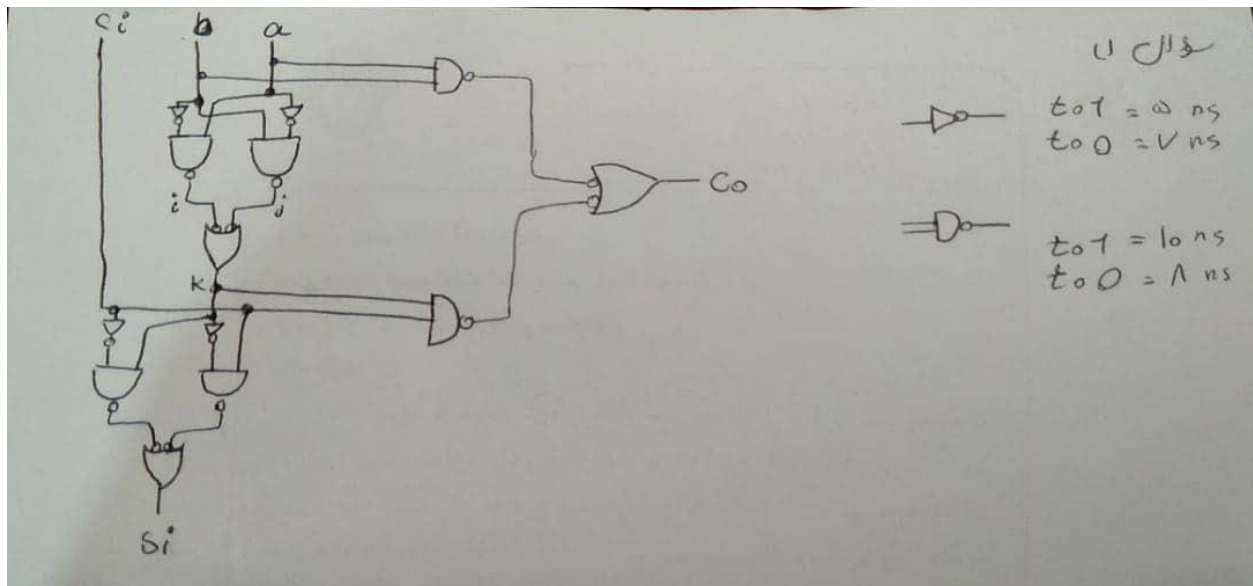
2*10ns nand +2*8ns nand = 36ns

To1= ci=1 , a=0 ,b=0 -> ci=1 , a=0 ,b=1 ->>> 36ns

2*10ns nand +2*8ns nand = 36ns

شماتیک مدار:

با توجه به اینکه سوال گفته از basic gate استفاده کنید xor ها را به شکل زیر جاگذاری کردم.



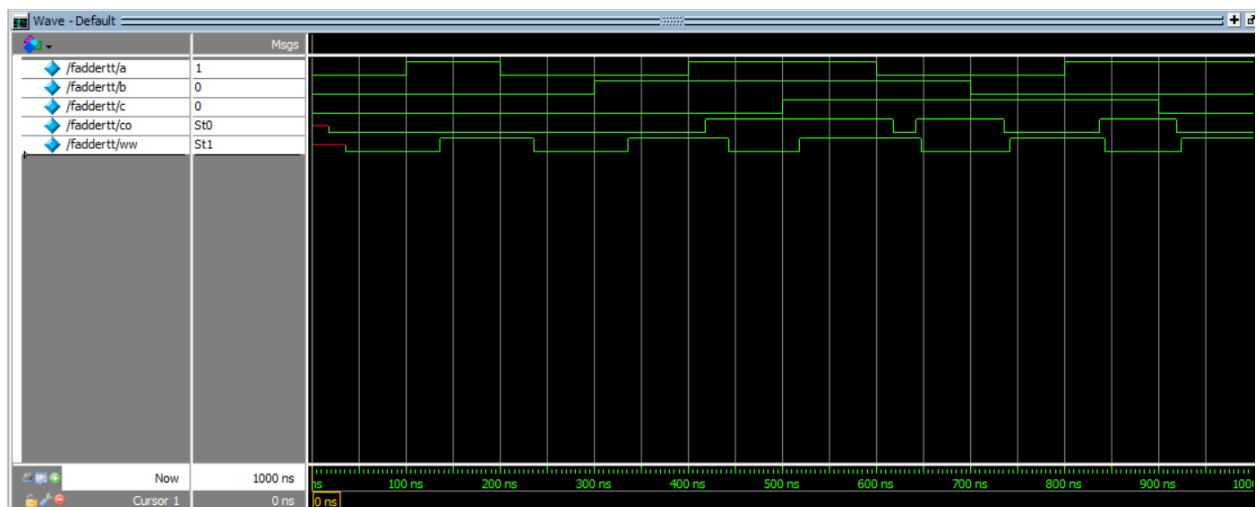
کد های سوال:

Ln#	
1	<code>`timescale 1ns/1ns</code>
2	<code>module fulladder (input ai,bi,ci,output co,si);</code>
3	<code>wire n,m,k,i,j,w,t,o,p,u,l;</code>
4	<code>not #(5,7) T1(m,ai);</code>
5	<code>not #(5,7) T2(w,bi);</code>
6	<code>nand #(10,8) T3(i,ai,w);</code>
7	<code>nand #(10,8) T4(j,m,bi);</code>
8	<code>nand #(10,8) T5(k,i,j);</code>
9	
10	<code>not #(5,7) T6(n,ci);</code>
11	<code>not #(5,7) T7(t,k);</code>
12	<code>nand #(10,8) T8(l,t,ci);</code>
13	<code>nand #(10,8) T9(u,n,k);</code>
14	<code>nand #(10,8) T10(si,u,l);</code>
15	
16	<code>nand #(10,8) T11(p,ai,bi);</code>
17	<code>nand #(10,8) T12(o,k,ci);</code>
18	<code>nand #(10,8) T13(co,p,o);</code>
19	
20	<code>endmodule</code>
21	

Testbench

Ln#	
1	<code>`timescale 1ns/1ns</code>
2	<code>module faddertt();</code>
3	<code>logic a=0,b=0,c=0;</code>
4	<code>fulladder UT(a,b,c,co,ww);</code>
5	<code>initial begin</code>
6	<code> #100 a=1;</code>
7	<code> #100 a=0;</code>
8	<code> #100 b=1;</code>
9	<code> #100 a=1;</code>
10	<code> #100 c=1;</code>
11	<code> #100 a=0;</code>
12	<code> #100 b=0;</code>
13	<code> #100 a=1;</code>
14	<code> #100 c=0;</code>
15	<code> #100 \$stop;</code>
16	<code>end</code>
17	<code>endmodule</code>
18	

شكل موج :



سوال دوم

در این سوال نیاز داریم از پارامتر استفاده کنیم پس ساختار را به شکل زیر می‌کنیم.

کدهای سوال:

Ln#	
1	<code>`timescale 1ns/1ns</code>
2	<code>module fadder #(parameter k) (a,b,ci,s,co);</code>
3	
4	<code>input [k-1:0]a,b;</code>
5	<code>input ci;</code>
6	<code>output [k-1:0]s;</code>
7	<code>output co;</code>
8	<code>assign #(48+36*(k-1)) {co,s}=a+b+ci;</code>
9	<code>endmodule</code>
10	
11	

سوال سوم

کد های سوال:

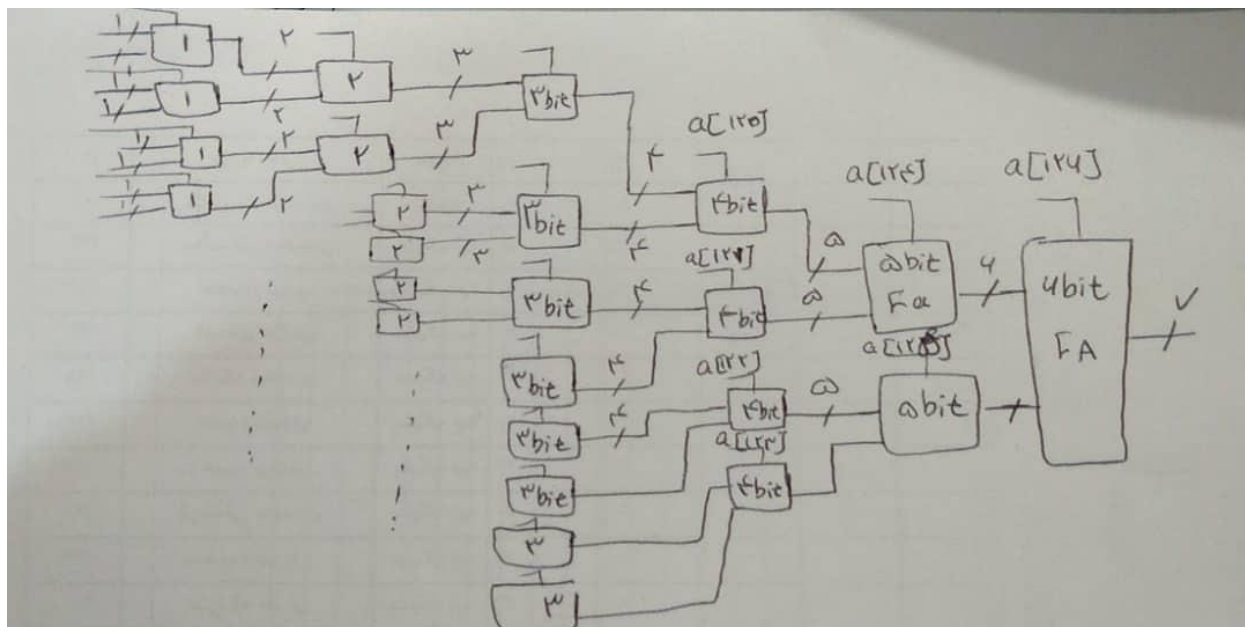
Ln#	
1	<code>`timescale 1ns/1ns</code>
2	<code>module tbfa();</code>
3	<code>logic [3:0]aa=4'b0000;</code>
4	<code>logic [3:0]bb=4'b0000;</code>
5	<code>logic ci=0;</code>
6	<code>wire [3:0]si;</code>
7	<code>wire co;</code>
8	<code>fadder #4 U(aa,bb,ci,si,co);</code>
9	<code>initial begin</code>
10	<code>#100 aa=4'b0001;</code>
11	<code>#100 bb=4'b0001;</code>
12	<code>repeat(10) begin #100 aa=\$random;</code>
13	<code>bb=\$random;</code>
14	<code>ci=\$random;</code>
15	<code>end</code>
16	<code>#100 \$stop;</code>
17	<code>end</code>
18	<code>endmodule</code>
19	

شکل موج :



سوال چهارم

شماتیک مدار:



کد های سوال:

```

Ln# 1 module adder127taei(input [126:0]a,output[6:0]w);
    2
    3   genvar i;
    4   wire [63:0]add1;
    5   generate
    6       for(i=0;i<32;i=i+1) begin
    7           fadder #1 Y(a[3*i],a[3*i+1],a[3*i+2],add1[2*i],add1[2*i+1]);
    8       end
    9   endgenerate
    10
    11
    12   genvar j;
    13   wire [47:0]add2;
    14   generate
    15       for(j=0;j<16;j=j+1) begin
    16           fadder #2 Y((add1[4*j],add1[4*j+1]),(add1[4*j+2],add1[4*j+3]),a[j+96],(add2[3*j],add2[3*j+1],add2[3*j+2]));
    17       end
    18   endgenerate
    19
    20   genvar h;
    21   wire [31:0]add3;
    22   generate
    23       for(h=0;h<8;h=h+1) begin
    24           fadder #3 Y((add2[6*h],add2[6*h+1],add2[6*h+2]),(add2[6*h+3],add2[6*h+4],add2[6*h+5]),a[h+112],(add3[4*h],add3[4*h+1],add3[4*h+2],add3[4*h+3]));
    25       end
    26   endgenerate
    27
    28
    29
    30
    31
    32
    33
    34
    35
    36
    37
    38
    39
    40
    41
    42
    43
    44
    45
    46
    47
    48
    49
    50
    51
    52
    53
    54
    55
    56
    57
    58
    59
    60
    61
    62
    63
    64
    65
    66
    67
    68
    69
    70
    71
    72
    73
    74
    75
    76
    77
    78
    79
    80
    81
    82
    83
    84
    85
    86
    87
    88
    89
    90
    91
    92
    93
    94
    95
    96
    97
    98
    99
    100
    
```

```

26 endgenerate
27
28 genvar p;
29 wire [19:0]add4;
30 generate
31     for(p=0;p<4;p=p+1) begin
32         fadder #4 Y((add3[8*p],add3[8*p+1],add3[8*p+2],add3[8*p+3]),{add3[8*p+4],add3[8*p+5],add3[8*p+6],add3[8*p+7]},a[p+120],{add4[5*p],add4[5*p+1],add4[5*p+2],add4[5*p+3]},add4[5*p+4]);
33     end
34 endgenerate
35
36
37 genvar z;
38 wire [11:0]add5;
39 generate
40     for(z=0;z<2;z=z+1) begin
41         fadder #5 Y((add4[10*z],add4[10*z+1],add4[10*z+2],add4[10*z+3],add4[10*z+4]),{add4[10*z+5],add4[10*z+6],add4[10*z+7],add4[10*z+8],add4[10*z+9]},a[z+124],{add5[6*z],add5[6*z+1],
42             add5[6*z+2],add5[6*z+3],add5[6*z+4]},add5[6*z+5]);
43     end
44 endgenerate
45
46 wire [6:0]add6;
47 fadder #6 Y((add5[0],add5[1],add5[2],add5[3],add5[4],add5[5]),{add5[6],add5[7],add5[8],add5[9],add5[10],add5[11]},a[126],{add6[0],add6[1],add6[2],add6[3],add6[4],add6[5]},add6[6]);
48
49 assign w=add6;
50
51 endmodule

```

<

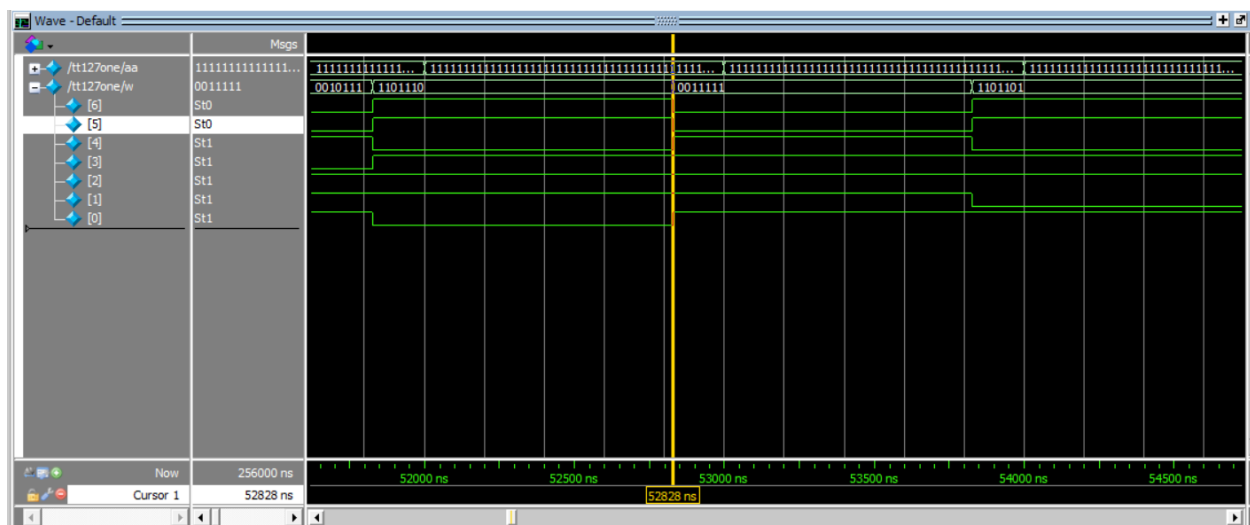
Transcript

سوال پنجم

کد های سوال:

```
Ln#
1  `timescale 1ns/1ns
2  module tt127one();
3      logic [126:0]aa=127'b0;
4      wire [6:0]w;
5      adder127taei u(aa,w);
6      initial begin
7          #400 aa[0]=1;
8
9          repeat(254) begin #400 aa=~aa[0],aa[126:1]];
10     end
11     #400 $stop;
12 end
13 endmodule
```

شکل موج : همانطور که مشاهده می شود بیشترین دیلی 828ns است.



سوال ششم

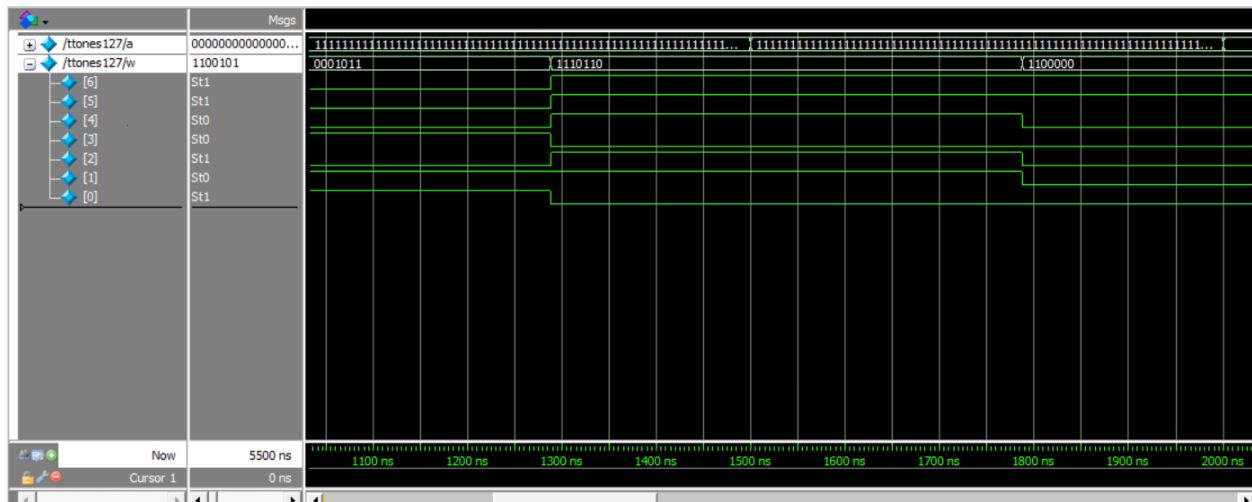
کد های سوال:

Ln#	
1	<code>`timescale 1ns/1ns</code>
2	<code>module ones127(input [126:0]a, output [6:0]w);</code>
3	<code>integer x=0;</code>
4	<code>integer i=0;</code>
5	<code>always @(a) begin</code>
6	<code> for(i=0;i<127;i=i+1)begin</code>
7	<code> x=x+a[i];</code>
8	<code> end</code>
9	<code>end</code>
10	
11	<code>assign #828 w=x;</code>
12	<code>endmodule</code>
13	

testbench

Ln#	
1	<code>`timescale 1ns/1ns</code>
2	<code>module ttones127();</code>
3	<code>logic [126:0]a=127'b0;</code>
4	<code>wire [6:0]w;</code>
5	<code>ones127 u(a,w);</code>
6	<code>initial begin</code>
7	<code> repeat(10) #1000 a=\$random();</code>
8	<code> #1000 \$stop;</code>
9	<code>end</code>
10	<code>endmodule</code>
11	

شکل موج :



سوال هفتم

توضیحات : تعداد سلولهای سوال 4 کمتر سلولهای سوال 6 است زیرا در سوال 6 از procedural statement استفاده کردیم که همان always است که در این سوال استفاده کرده ایم و این باعث غیر بهینه شدن کد میشود و این باعث بیشتر شدن سلول ها نسبت به سوال 4 می شود. در سوال 4 رویکرد سخت افزاری داشتیم و در سوال 6 رویکرد نرم افزاری. همچنین در سوال 4 دلیلی های کمتر از 828 هم داریم اما در سوال 6 به ازای همه تغییر ها 828 خواهیم داشت.

تعداد سلول های بخش 4 : 552 سلول |||| تعداد سلول های بخش 6 : 711 سلول

```
=== ones127 ===
Number of wires:          707
Number of wire bits:      870
Number of public wires:    3
Number of public wire bits: 166
Number of memories:        0
Number of memory bits:     0
Number of processes:       0
Number of cells:          711
  $ _AND_                  11
  $ _AOI3_                  51
  $ _NAND_                   6
  $ _NOR_                   254
  $ _NOT_                    98
  $ _OAI3_                   8
  $ _OR_                     8
  $ _XNOR_                  243
  $ _XOR_                    32
```

```
Number of wires:          692
Number of wire bits:      1172
Number of public wires:    323
Number of public wire bits: 803
Number of memories:        0
Number of memory bits:     0
Number of processes:       0
Number of cells:          552
  $ _AND_                   46
  $ _AOI3_                   46
  $ _NAND_                   83
  $ _NOR_                    3
  $ _NOT_                    46
  $ _OAI3_                   82
  $ _OR_                     6
  $ _XNOR_                  128
  $ _XOR_                   112
```

تعداد خط های یوسیس بخش 4 : 2208 خط |||| تعداد خط های یوسیس بخش 6 : 11200 خط

```
11194    assign _2415_ = _2585_;
11195    assign _2417_ = _2586_;
11196    assign _2419_ = _2587_;
11197    assign _2421_ = _2588_;
11198    assign _2423_ = _2589_;
11199    endmodule
11200
```

```
2203    assign s[4] = _048_;
2204    assign _049_ = b[5];
2205    assign _051_ = a[5];
2206    assign s[5] = _056_;
2207    assign co = _064_;
2208    endmodule
2209
```