



به نام خدا
دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده مهندسی برق و کامپیوتر



سیستم‌های دیجیتال 1

--- ECE 894 ---

نیمسال دوم (99-00)

استاد: پروفسور نوابی

Computer Assignment 3

محمد مهدی عبدالحسینی

810 198 434



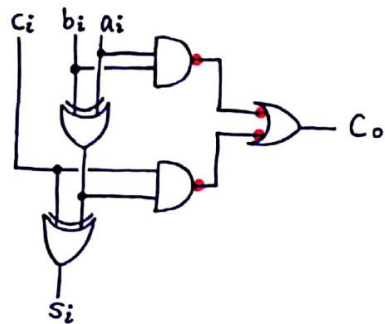
D I G I T A L S Y S T E M S

فہرست مطالب

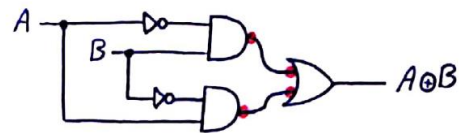
1	بخش اول:
4	بخش دوم:
4	بخش سوم:
4	بخش چهارم:
5	بخش پنجم:
5	بخش ششم:
5	بخش ہفتم:

بخش اول :

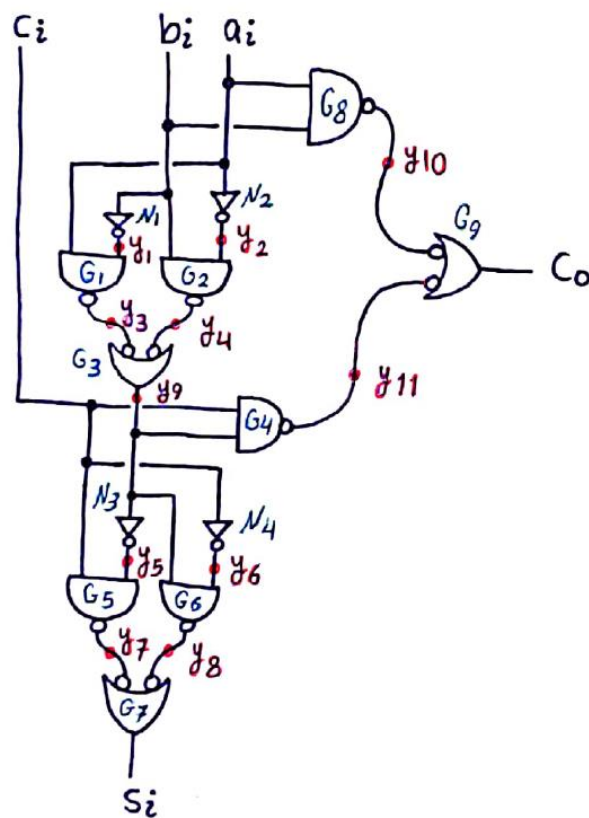
Full Adder : (gate level)



XOR : (using basic gates)



Full Adder : (using basic gates)



Calculating Delay Values and Worst Case Delay :

Delay Values from CA1:

2-input NAND #(10,8)

Inverter: NOT #(5,7)

Worst case delay :

$$S_i \text{ حالت } to0 \text{ برای } : \begin{matrix} c_i & a_i & b_i \\ 0 & 0 & 1 \end{matrix} \rightarrow \begin{matrix} c_i & a_i & b_i \\ 0 & 1 & 1 \end{matrix}$$

$$\Rightarrow to0 \text{ WCD} = 7 + 10 + 8 + 10 + 8 = \underline{45}$$

$$S_i \text{ حالت } to1 \text{ برای } : \begin{matrix} 0 & 0 & 0 \end{matrix} \rightarrow \begin{matrix} 0 & 0 & 1 \end{matrix}$$

$$\Rightarrow to1 \text{ WCD} = 8 + 10 + 8 + 10 = \underline{36}$$

$$C_o \text{ حالت } to0 \text{ برای } : \begin{matrix} 0 & 1 & 1 \end{matrix} \rightarrow \begin{matrix} 0 & 0 & 1 \end{matrix}$$

$$\Rightarrow to0 \text{ WCD} = 10 + 8 = \underline{18}$$

$$C_o \text{ حالت } to1 \text{ برای } : \begin{matrix} 0 & 0 & 1 \end{matrix} \rightarrow \begin{matrix} 0 & 1 & 1 \end{matrix}$$

$$\Rightarrow to1 \text{ WCD} = 10 + 8 = \underline{18}$$

n-Bit Full Adder using basic gates :

* sv-file مربوط به این قسمت با نام Part1.sv پیوست شده است.

```

1 module Part1_myFullAdder (input ai,bi,ci , output si,co);
2     wire [1:11] y;
3     nand #(10,8) G1(y[3],y[1],ai);
4     nand #(10,8) G2(y[4],y[2],bi);
5     nand #(10,8) G3(y[9],y[3],y[4]);
6     nand #(10,8) G4(y[11],y[9],ci);
7     nand #(10,8) G5(y[7],y[5],ci);
8     nand #(10,8) G6(y[8],y[6],y[9]);
9     nand #(10,8) G7(si,y[7],y[8]);
10    nand #(10,8) G8(y[10],bi,ai);
11    nand #(10,8) G9(co,y[10],y[11]);
12    not #(5,7) N1(y[1],bi);
13    not #(5,7) N2(y[2],ai);
14    not #(5,7) N3(y[5],y[9]);
15    not #(5,7) N4(y[6],ci);
16 endmodule

17 module Part1_nBitAdder #(parameter n=4) (input [n:1] A,B, input Cin , output [n:1] S , output Co);
18     wire [1:n+1] J;
19     assign J[1] = Cin;
20     assign Co = J[n+1];
21     genvar k;
22     generate
23     for (k = 1 ; k <= n ; k = k + 1) begin: Adders
24         Part1_myFullAdder nBA (A[k],B[k],J[k],S[k],J[k+1]);
25     end
26     endgenerate
27 endmodule

28 module Part1_TB ();
29     logic [3:0] A = 0;
30     logic [3:0] B = 0;
31     logic Cin = 0;
32     wire [3:0] S;
33     wire Co;
34     Part1_nBitAdder G1 (A,B,Cin,S,Co);
35     initial begin
36         #100 A[0] = 1;
37         #100 A[2] = 1;
38         #100 B[2] = 1;
39         #100 B[1] = 1;
40         #100 A[2] = 0;
41         #100 A[1] = 1;
42         #100 B[3] = 1;
43         #200 $stop;
44     end
45 endmodule

46
47
48
49

```

بخش دوم:

* sv-file مربوط به این قسمت با نام Part2.sv پیوست شده است.

```
1 module Part2_nBitAdder #(parameter n=4) (input [n:1] A,B, input Cin , output [n:1] S , output Co);
2     assign #40 {Co,S} = A + B + Cin;
3 endmodule
```

بخش سوم:

* sv-file مربوط به این قسمت با نام Part3.sv پیوست شده است.

```
1 module Part2_TB ();
2     logic [3:0] A = 0;
3     logic [3:0] B = 0;
4     logic Cin = 0;
5     wire [3:0] S;
6     wire Co;
7     Part2_nBitAdder #4 G1 (A,B,Cin,S,Co);
8     initial begin
9         repeat(10) #100 A = $random(); #100 B = $random(); #100 Cin = $random();
10        #200 $stop;
11    end
12 endmodule
```

بخش چهارم:

* sv-file مربوط به این قسمت با نام Part4.sv پیوست شده است.

```
1 module Part4_myOnesCounter (input [126:0] A , output[6:0] S);
2     logic [63:0] B1 = 0;
3     logic [47:0] B2 = 0;
4     logic [36:0] B3 = 0;
5     logic [19:0] B4 = 0;
6     logic [11:0] B5 = 0;
7     genvar k;
8     generate
9         for (k = 0 ; k <= 31 ; k = k + 1)begin: Adder1Bit
10             Part1_nBitAdder #1 G1(A[3*k],A[3*k+1],A[3*k+2],B1[2*k],B1[2*k+1]);
11         end
12         for (k = 0 ; k <= 15 ; k = k + 1)begin: Adder2Bit
13             Part1_nBitAdder #2 G2(B1[4*k+1:4*k],B1[4*k+3:4*k+2],A[k+96],B2[3*k+1:3*k],B2[3*k+2]);
14         end
15         for (k = 0 ; k <= 7 ; k = k + 1)begin: Adder3Bit
16             Part1_nBitAdder #3 G3(B2[6*k+2:6*k],B2[6*k+5:6*k+3],A[k+112],B3[4*k+2:4*k],B3[4*k+3]);
17         end
18         for (k = 0 ; k <= 3 ; k = k + 1)begin: Adder4Bit
19             Part1_nBitAdder #4 G4(B3[8*k+3:8*k],B3[8*k+7:8*k+4],A[k+120],B4[5*k+3:5*k],B4[5*k+4]);
20         end
21         for (k = 0 ; k <= 1 ; k = k + 1)begin: Adder5Bit
22             Part1_nBitAdder #5 G5(B4[10*k+4:10*k],B4[10*k+9:10*k+5],A[k+124],B5[6*k+4:6*k],B5[6*k+5]);
23         end
24     endgenerate
25     Part1_nBitAdder #6 G6(B5[5:0],B5[11:6],A[126],S[5:0],S[6]);
26 endmodule
```

بخش پنجم :

* sv-file مربوط به این قسمت با نام Part5.sv پیوست شده است.

```

1 module Part5_myOnesCounterTB ();
2     logic [126:0] a;
3     wire [6:0] w;
4     Part4_myOnesCounter GG (a,w);
5     initial begin
6         #500 a = 127'd114;
7         #500 a = 127'd993;
8         #500 a = 127'd331;
9         #500 a = 127'd556;
10        end
11    endmodule

```

```

12 module Part6_TB ();
13     logic [126:0] a;
14     wire [6:0] w;
15     Part6_myOnesCounter GG(a,w);
16     initial begin
17         #500 a = 127'd114;
18         #500 a = 127'd993;
19         #500 a = 127'd331;
20         #500 a = 127'd556;
21     end
22 endmodule
23

```

بخش ششم :

* sv-file مربوط به این قسمت با نام Part6.sv پیوست شده است.

```

1 module Part6_myOnesCounter (input [126:0] A , output reg [6:0] C);
2     integer i;
3     always @(A) begin
4         C = 7'd0;
5         for (i = 0 ; i <= 126 ; i = i + 1) begin
6             if (A[i] == 1'b1) C = C + 1;
7         end
8     end
9 endmodule

```

بخش هفتم :

* sv-file مربوط به این قسمت با نام Part6_Synthesis.sv و Part4_Synthesis.sv پیوست شده است.

با بررسی خروجی سنتز برای بخش‌های چهارم و ششم میتوان گفت ساختار always به مراتب delay و تعداد gate بیشتری دارد. بنابراین قطعه کد بخش چهارم بهینه‌تر خواهد بود.