

طراحی ALU74181

طراحی این مدار به منظور پیاده‌سازی تمامی خروجی به صورت Structural انجام شده‌است. همچنین برای این مدار ساختار Active High انتخاب شده.

این سیستم متشکل از ۴ بیت برای Select نوع عملیات و

۴ بیت برای ورودی اول و ۴ بیت برای ورودی دوم می‌باشد همچنین بیت M برای انتخاب بین عملیات‌های منطقی و ریاضی تعبیه شده‌است. در هنگام استفاده از عملیات ریاضی نیز یک بیت برای CarryIn تعبیه شده‌است.

برای خروجی هم ۴ بیت F جواب عملیات است.

اگر هر چهار بیت عملیات برابر یک باشد حاصل بیت $A=B$ نیز یک می‌شود.

برای تسریع بخشیدن عملیات ریاضی بیت‌های G و P (بر اساس CarryLookAhead) تعبیه شده‌است و یک بیت Carryout (بر اساس Ripple) نیز برای حالت عادی مدار تعریف شده‌است.

جدول درستی مدار به شکل زیر می‌باشد :

SELECTION				ACTIVE-HIGH DATA	
				M = H	M = L; ARITHMETIC OPERATIONS
S3	S2	S1	S0	LOGIC FUNCTIONS	
					$\overline{C}_n = H$ (no carry)
L	L	L	L	$F = \overline{A}$	$F = A$
L	L	L	H	$F = \overline{A + B}$	$F = A + B$
L	L	H	L	$F = \overline{AB}$	$F = A + \overline{B}$
L	L	H	H	$F = 0$	$F = \text{MINUS 1 (2's COMPL)}$
L	H	L	L	$F = \overline{AB}$	$F = A \text{ PLUS } \overline{AB}$
L	H	L	H	$F = \overline{B}$	$F = (A + B) \text{ PLUS } \overline{AB}$
L	H	H	L	$F = A \oplus B$	$F = A \text{ MINUS } B \text{ MINUS } 1$
L	H	H	H	$F = \overline{AB}$	$F = \overline{AB} \text{ MINUS } 1$
H	L	L	L	$F = \overline{A + B}$	$F = A \text{ PLUS } AB$
H	L	L	H	$F = \overline{A \oplus B}$	$F = A \text{ PLUS } B$
H	L	H	L	$F = B$	$F = (A + \overline{B}) \text{ PLUS } AB$
H	L	H	H	$F = AB$	$F = AB \text{ MINUS } 1$
H	H	L	L	$F = 1$	$F = A \text{ PLUS } A$
H	H	L	H	$F = A + \overline{B}$	$F = (A + B) \text{ PLUS } A$
H	H	H	L	$F = A + B$	$F = (A + \overline{B}) \text{ PLUS } A$
H	H	H	H	$F = A$	$F = A \text{ MINUS } 1$
					$\overline{C}_n = L$ (with carry)
					$F = A \text{ PLUS } 1$
					$F = (A + B) \text{ PLUS } 1$
					$F = (A + \overline{B}) \text{ PLUS } 1$
					$F = \text{ZERO}$
					$F = A \text{ PLUS } \overline{AB} \text{ PLUS } 1$
					$F = (A + B) \text{ PLUS } \overline{AB} \text{ PLUS } 1$
					$F = A \text{ MINUS } B$
					$F = \overline{AB}$
					$F = A \text{ PLUS } AB \text{ PLUS } 1$
					$F = A \text{ PLUS } B \text{ PLUS } 1$
					$F = (A + \overline{B}) \text{ PLUS } AB \text{ PLUS } 1$
					$F = AB$
					$F = A \text{ PLUS } A \text{ PLUS } 1$
					$F = (A + B) \text{ PLUS } A \text{ PLUS } 1$
					$F = (A + \overline{B}) \text{ PLUS } A \text{ PLUS } 1$
					$F = A$

آزمایشگاه مدار منطقی - تمرین دوم
متین زیودار - محمد خدام
کدها :

```
module myand1(input a,
              output o);
    assign o = a;
endmodule

module myand2(input a, b,
              output o);
    assign o = a & b;
endmodule

module myand3(input a, b, c,
              output o);
    assign o = a & b & c;
endmodule

module myand4(input a, b, c, d,
              output o);
    assign o = a & b & c & d;
endmodule

module myand5(input a, b, c, d, e,
              output o);
    assign o = a & b & c & d & e;
endmodule
```

```
module myinv(input a,
             output o);
    assign o = ~a;
endmodule
```

```
module mynor2(input a, b,
              output o);
    assign o = ~a | ~b;
endmodule
```

```
module mynand2(input a, b,  
    output o);  
    assign o = ~(a & b);  
endmodule  
  
module mynand4(input a, b, c, d,  
    output o);  
    assign o = ~(a & b & c & d);  
endmodule  
  
module mynand5(input a, b, c, d, e,  
    output o);  
    assign o = ~(a & b & c & d & e);  
endmodule
```

```
module mynor2(input a,b,  
    output o);  
    assign o = ~(a | b);  
endmodule  
  
module mynor3(input a, b, c,  
    output o);  
    assign o = ~(a | b | c);  
endmodule  
  
module mynor4(input a, b, c, d,  
    output o);  
    assign o = ~(a | b | c | d);  
endmodule
```

```
module myxor2(input a, b,  
    output o);  
    assign o = a ^ b;  
endmodule
```

```
module ALU(input[3:0] A,B,S,  
    input M,Cn_,  
    output[3:0] F,  
    output Eq,G,P,Cnplus);
```

```
wire[3:0] Pnot, Gnot;
wire[3:0] Bnot;
wire[4:0] temp_0;
wire[4:0] temp_1;
wire[4:0] temp_2;
wire[4:0] temp_3;
wire Mnot;

myinv not_M(M, Mnot);

myinv inv_00(B[0], Bnot[0]);
myand1 and_00(A[0], temp_0[0]);
myand2 and_01(B[0], S[0], temp_0[1]);
myand2 and_02(S[1], Bnot[0], temp_0[2]);
myand3 and_03(Bnot[0], S[2], A[0], temp_0[3]);
myand3 and_04(A[0], B[0], S[3], temp_0[4]);
mynor3 nor_00(temp_0[0], temp_0[1], temp_0[2], Pnot[0]);
mynor2 nor_01(temp_0[3], temp_0[4], Gnot[0]);

myinv inv_10(B[1], Bnot[1]);
myand1 and_10(A[1], temp_1[0]);
myand2 and_11(B[1], S[0], temp_1[1]);
myand2 and_12(S[1], Bnot[1], temp_1[2]);
myand3 and_13(Bnot[1], S[2], A[1], temp_1[3]);
myand3 and_14(A[0], B[0], S[3], temp_1[4]);
mynor3 nor_10(temp_1[0], temp_1[1], temp_1[2], Pnot[1]);
mynor2 nor_11(temp_1[3], temp_1[4], Gnot[1]);

myinv inv_20(B[2], Bnot[2]);
myand1 and_20(A[2], temp_2[0]);
myand2 and_21(B[2], S[0], temp_2[1]);
myand2 and_22(S[1], Bnot[2], temp_2[2]);
myand3 and_23(Bnot[2], S[2], A[2], temp_2[3]);
myand3 and_24(A[2], B[2], S[3], temp_2[4]);
mynor3 nor_20(temp_2[0], temp_2[1], temp_2[2], Pnot[2]);
mynor2 nor_21(temp_2[3], temp_2[4], Gnot[2]);

myinv inv_30(B[3], Bnot[3]);
myand1 and_30(A[3], temp_3[0]);
myand2 and_31(B[3], S[0], temp_3[1]);
myand2 and_32(S[1], Bnot[3], temp_3[2]);
myand3 and_33(Bnot[3], S[2], A[3], temp_3[3]);
myand3 and_34(A[3], B[3], S[3], temp_3[4]);
mynor3 nor_30(temp_3[0], temp_3[1], temp_3[2], Pnot[3]);
```

```
mynor2 nor_31(temp_3[3], temp_3[4], Gnot[3]);

wire[15:0] temp_lv13;

mynand2 nand_lv13_0(Cn_, Mnot, temp_lv13[0]);
myand2 and_lv13_0(Mnot, Pnot[0], temp_lv13[1]);
myand3 and_lv13_1(Mnot, Gnot[0], Cn_, temp_lv13[2]);
myand2 and_lv13_2(Mnot, Pnot[1], temp_lv13[3]);
myand3 and_lv13_3(Mnot, Pnot[0], Gnot[1], temp_lv13[4]);
myand4 and_lv13_4(Mnot, Cn_, Gnot[0], Gnot[1], temp_lv13[5]);
myand2 and_lv13_5(Mnot, Pnot[2], temp_lv13[6]);
myand3 and_lv13_6(Mnot, Pnot[1], Gnot[2], temp_lv13[7]);
myand4 and_lv13_7(Mnot, Pnot[0], Gnot[1], Gnot[2], temp_lv13[8]);
myand5 and_lv13_8(Mnot, Cn_, Gnot[0], Gnot[1], Gnot[2], temp_lv13[9]);
mynand4 nand_lv13_1(Gnot[0], Gnot[1], Gnot[2], Gnot[3], temp_lv13[10]);
mynand5 nand_lv13_2(Cn_, Gnot[0], Gnot[1], Gnot[2], Gnot[3], temp_lv13[11]);
myand4 and_lv13_9(Pnot[0], Gnot[0], Gnot[2], Gnot[3], temp_lv13[12]);
myand3 and_lv13_10(Pnot[1], Gnot[2], Gnot[3], temp_lv13[13]);
myand2 and_lv13_11(Pnot[2], Gnot[3], temp_lv13[14]);
myand1 and_lv13_12(Pnot[3], temp_lv13[15]);

wire[7:0] temp_lv14;

myxor2 xor_lv14_0(Pnot[0], Gnot[0], temp_lv14[0]); //sig
ma0
mynor2 nor_lv14_0(temp_lv13[1], temp_lv13[2], temp_lv14[1]);
//C1
myxor2 xor_lv14_1(Pnot[1], Gnot[0], temp_lv14[2]); //sig
ma1
mynor3 nor_lv14_1(temp_lv13[3], temp_lv13[4], temp_lv13[5], temp_lv14[3]);
//c2
myxor2 xor_lv14_2(Pnot[2], Gnot[1], temp_lv14[4]); //sig
ma2
mynor4 nor_lv14_2(temp_lv13[6], temp_lv13[7], temp_lv13[8], temp_lv13[9], temp_lv14[5]); //c3
myxor2 xor_lv14_3(Pnot[3], Gnot[3], temp_lv14[6]); //sig
ma3
mynor4 nor_lv14_3(temp_lv13[12], temp_lv13[13], temp_lv13[14], temp_lv13[15], temp_lv14[7]); //c4

wire[4:0] temp_lv15;
```

```

myxor2 xor_lv15_0(temp_lv13[0], temp_lv14[0], temp_lv15[0]);
myxor2 xor_lv15_1(temp_lv14[1], temp_lv14[2], temp_lv15[1]);
myxor2 xor_lv15_2(temp_lv14[3], temp_lv14[4], temp_lv15[2]);
myxor2 xor_lv15_3(temp_lv14[5], temp_lv14[6], temp_lv15[3]);
mymor2 mor_lv15_4(temp_lv13[11], temp_lv14[7], temp_lv15[4]);

myand4 and_lv16_0(temp_lv15[0], temp_lv15[1], temp_lv15[2], temp_lv15[3], Eq)
;

assign F[0] = temp_lv15[0];
assign F[1] = temp_lv15[1];
assign F[2] = temp_lv15[2];
assign F[3] = temp_lv15[3];
assign G = temp_lv13[10];
assign Cnplus = temp_lv15[4];
assign P = temp_lv14[7];
endmodule

```

تست‌بنج :

```

module testbench();

reg[3:0] A;
reg[3:0] B;
reg[3:0] S;
reg M;
reg Cn_;

wire[3:0] F;
wire Eq;
wire G;
wire P;
wire Cnplus;

ALU alu0(A,B,S,
        M,Cn_,
        F,
        Eq,G,P,Cnplus);

```

```
initial
begin
    S = 4'b0000;
    A = 4'b0010;
    B = 4'b0110;
    M = 1'b0;
    Cn_ = 1'b1;

    #15;
    S = 4'b0001;
    A = 4'b1010;
    B = 4'b0110;
    M = 1'b0;
    Cn_ = 1'b0;

    #15;
    S = 4'b0010;
    A = 4'b1110;
    B = 4'b1010;
    M = 1'b0;
    Cn_ = 1'b0;

    #15;
    S = 4'b0011;
    A = 4'b1110;
    B = 4'b0001;
    M = 1'b1;
    Cn_ = 1'b1;

    #15;
    S = 4'b0101;
    A = 4'b1111;
    B = 4'b1111;
    M = 1'b0;
    Cn_ = 1'b1;
```

```
#15;
S = 4'b0111;
A = 4'b1111;
B = 4'b1111;
M = 1'b0;
Cn_ = 1'b0;

#15;
S = 4'b1000;
A = 4'b1111;
B = 4'b0111;
M = 1'b0;
Cn_ = 1'b0;

#15;
S = 4'b1010;
A = 4'b0111;
B = 4'b1000;
M = 1'b0;
Cn_ = 1'b1;

#15;
S = 4'b1100;
A = 4'b1111;
B = 4'b1111;
M = 1'b0;
Cn_ = 1'b1;

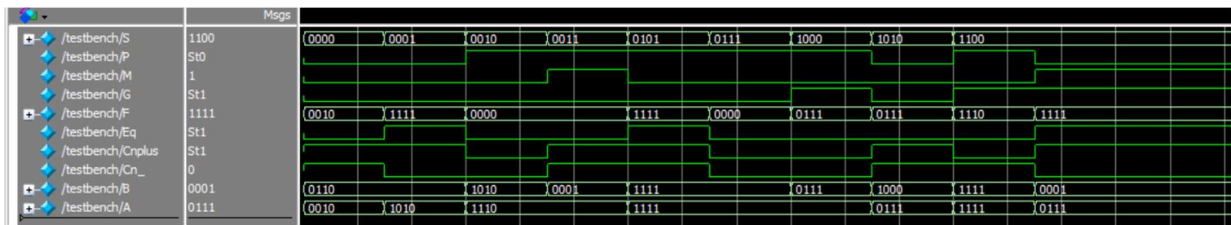
#15;
S = 4'b1100;
A = 4'b0111;
B = 4'b0001;
M = 1'b1;
Cn_ = 1'b0;

$finish;
end
endmodule
```


آزمایشگاه مدار منطقی - تمرین دوم

متین زیودار - محمد خدام

سیمپولیشن :



کد Behavioral :

```
module ALU(  
    output[3:0] F,  
    output AeB,  
    output G,  
    output Cn4,  
    output P,  
  
    input[3:0] S,  
    input[3:0] A,  
    input[3:0] B,  
    input M,  
    input Cn  
  
);  
  
reg[3:0] Result;  
assign F = Result;  
//assign Cn4 = OutRipple;  
  
always @(*)  
    begin  
        case(S)  
            4'b0000:  
                Result = M ? ~A : (Cn ? A : A + 1) ;  
            4'b0001:  
                Result = M ? ~(A | B) : (Cn ? A | B : (A | B) + 1) ;  
            4'b0010:  
                Result = M ? ~A & B : (Cn ? A | ~B : (A | ~B) + 1) ;  
            4'b0011:  
                Result = M ? 0 : (Cn ? -1 : 0) ;  
            4'b0100:  
                Result = M ? ~A : (Cn ? A : A + 1) ;  
            4'b0101:  
                Result = M ? ~(A | B) : (Cn ? A | B : (A | B) + 1) ;  
            4'b0110:  
                Result = M ? ~A & B : (Cn ? A | ~B : (A | ~B) + 1) ;  
            4'b0111:  
                Result = M ? 0 : (Cn ? -1 : 0) ;  
            4'b1000:  
                Result = M ? ~A : (Cn ? A : A + 1) ;  
            4'b1001:  
                Result = M ? ~(A | B) : (Cn ? A | B : (A | B) + 1) ;  
            4'b1010:  
                Result = M ? ~A & B : (Cn ? A | ~B : (A | ~B) + 1) ;  
            4'b1011:  
                Result = M ? 0 : (Cn ? -1 : 0) ;  
            4'b1100:  
                Result = M ? ~A : (Cn ? A : A + 1) ;  
            4'b1101:  
                Result = M ? ~(A | B) : (Cn ? A | B : (A | B) + 1) ;  
            4'b1110:  
                Result = M ? ~A & B : (Cn ? A | ~B : (A | ~B) + 1) ;  
            4'b1111:  
                Result = M ? 0 : (Cn ? -1 : 0) ;  
        endcase  
    end
```

```

    Result = M ? ~(A & B) : (Cn ? A + (A & ~B) : (A + (A & ~B)) + 1) ;
4'b0101:
    Result = M ? ~B : (Cn ? (A | B) + (A & ~B) : ((A | B) + (A & ~B)) + 1)
;

4'b0110:
    Result = M ? ~(A & B) & (A | B) : (Cn ? A - B - 1 : A - B) ;
4'b0111:
    Result = M ? A & ~B : (Cn ? (A & ~B) - 1 : A & ~B) ;
4'b1000:
    Result = M ? ~A | B : (Cn ? A + (A & B) : (A + (A & B)) + 1) ;
4'b1001:
    Result = M ? ~(~(A & B) & (A | B)) : (Cn ? A + B : (A + B) + 1) ;
4'b1010:
    Result = M ? B : (Cn ? (A | ~B) + (A & B) : ((A | ~B) + (A & B)) + 1)
;

4'b1011:
    Result = M ? A & B : (Cn ? (A & B) - 1 : A & B) ;
4'b1100:
    Result = M ? 1 : (Cn ? A + A : (A + A) + 1) ;
4'b1101:
    Result = M ? A | ~B : (Cn ? (A | B) + A : ((A | B) + A) + 1) ;
4'b1110:
    Result = M ? A | B : (Cn ? (A | ~B) + A : ((A | ~B) + A) + 1) ;
4'b1111:
    Result = M ? A : (Cn ? A - 1 : A) ;
default: Result = A + B ;
endcase
end

assign AeB = (F[3] & F[2] & F[1] & F[0]);

endmodule

```

آزمایشگاه مدار منطقی - تمرین دوم

متین زیودار - محمد خدام

سنتز و بررسی زمانی مدار :

	Type	Count
1	boundary_port	22
2	cycloneiii_lcell_comb	30
1	normal	30
1	1 data inputs	5
2	2 data inputs	3
3	3 data inputs	2
4	4 data inputs	20
3		
4	Max LUT depth	6.00
5	Average LUT depth	4.64

Command Info		Summary of Paths	
	Delay	From Node	To Node
1	11.695	S[3]	Eq

