

گزارش آزمایش 2

ALU 74181

کامیاب عابدی - امین ساوه درودی - محمد اسکینی

این ALU همانطور که در دستور کار آزمایش ذکر شده ۳۲ عملیات محاسباتی انجام می‌دهد که این کار را با یک ورودی چهار بیتی [select] S و یک ورودی یک بیتی M انجام می‌دهد که جمعاً ۳۲ حالت دارند. برای پیاده سازی این مدار ما آمدم و از طراحی Gate Level استفاده کردیم و طبق شماتیک و جدول موجود در دستور کار ALU را پیاده سازی کردیم.

برای این کار دو ماژول Module 1, Module 2 و دو ماژول نهایی final module 1, final module 2 طراحی کردیم.

ماژول 1: ورودیهای A و B و S را میگیرد و gb و bnot را تولید میکند که در ماژول 2 لازم میشود.

```
module module1(
    input [3:0] A,
    input [3:0] B,
    input [3:0] S,
    output [3:0] gb,
    output [3:0] bnot
);

wire [3:0] ABS , ASBnot;

// bnot = ~B
not not1(bnot[0], B[0]);
not not2(bnot[1], B[1]);
not not3(bnot[2], B[2]);
not not4(bnot[3], B[3]);

// ABS=A & B & S[3]
and andabs1(ABS[0], A[0], B[0], S[3]);
and andabs2(ABS[1], A[1], B[1], S[3]);
and andabs3(ABS[2], A[2], B[2], S[3]);
and andabs4(ABS[3], A[3], B[3], S[3]);

// ASBnot = A & ~B & S[2]
and andasbnot1(ASBnot[0], A[0], bnot[0], S[2]);
and andasbnot2(ASBnot[1], A[1], bnot[1], S[2]);
```

```

and andasbnot3(ASBnot[2], A[2], bnot[2], S[2]);
and andasbnot4(ASBnot[3], A[3], bnot[3], S[2]);

//~(ABS|ASBnot)
nor norgb1(gb[0], ABS[0], ASBnot[0]);
nor norgb2(gb[1], ABS[1], ASBnot[1]);
nor norgb3(gb[2], ABS[2], ASBnot[2]);
nor norgb4(gb[3], ABS[3], ASBnot[3]);

endmodule

```

ماژول 2: ورودیهای A و B و bnot و S را میگیرد و خروجی pb را تولید میکند.

```

module module2(
    input [3:0] A,
    input [3:0] B,
    input [3:0] bnot,
    input [3:0] S,
    output [3:0] pb
);

    wire [3:0] Sbnot, BS;

    // ~B & S[1]
    and andsbnot1(Sbnot[0], bnot[0], S[1]);
    and andsbnot2(Sbnot[1], bnot[1], S[1]);
    and andsbnot3(Sbnot[2], bnot[2], S[1]);
    and andsbnot4(Sbnot[3], bnot[3], S[1]);

    // B & S[0]
    and andBS1(BS[0], B[0], S[0]);
    and andBS2(BS[1], B[1], S[0]);
    and andBS3(BS[2], B[2], S[0]);
    and andBS4(BS[3], B[3], S[0]);

    // ~(BbS1 | BS0 | A)
    nor norfinal1(pb[0], Sbnot[0], BS[0], A[0]);
    nor norfinal2(pb[1], Sbnot[1], BS[1], A[1]);
    nor norfinal3(pb[2], Sbnot[2], BS[2], A[2]);
    nor norfinal4(pb[3], Sbnot[3], BS[3], A[3]);

endmodule

```

ماژول نهایی 1 : خروجیهای دو ماژول قبلی یعنی gb و pb به همراه Cn را به عنوان ورودی میگیرد و خروجیهای اصلی و نهایی مدار یعنی G و P و Cn4 را تولید میکند.

```
module modulefinal1(
    input [3:0] gb,
    input [3:0] pb,
    input Cn,
    output [3:0] C,
    output G,
    output P,
    output Cn4
);

    //c[0] = ~Cn
    not C0(C[0], Cn);

    buf bufferpb0(pb0, pb[0]);
    and andcngb0(Cngb0, Cn, gb[0]);

    buf bufferpb1(pb1, pb[1]);
    and andpb0gb1(pb0gb1, pb[0], gb[1]);
    and andcngb01(Cngb01, Cn, gb[0], gb[1]);

    buf bufferpb2(pb2, pb[2]);
    and andpb1gb2(pb1gb2, pb[1], gb[2]);
    and andpb0gb1gb2(pb0gb1gb2, pb[0], gb[1], gb[2]);
    and andcngb012(Cngb012, Cn, gb[0], gb[1], gb[2]);

    buf bufferpb3(pb3, pb[3]);
    and andpb2gb3(pb2gb3, pb[2], gb[3]);
    and andpb1gb2gb3(pb1gb23, pb[1], gb[2], gb[3]);
    and andpb0gb123(pb0gb123, pb[0], gb[1], gb[2], gb[3]);

    // G
    nand nandG(G, gb[0], gb[1], gb[2], gb[3]);

    // P
    nor norP(P, pb3, pb2gb3, pb1gb23, pb0gb123);
    nand nandCnG(CnG, gb[0], gb[1], gb[2], gb[3], Cn);

    nand nandCn4(Cn4, P, CnG);
    // C
    nor norc3(C[3], pb2, pb1gb2, pb0gb1gb2, Cngb012);

    nor norc2(C[2], pb1, pb0gb1, Cngb01);
```

```

        nor norc1(C[1], pb0, Cngb0);

endmodule

```

ماژول نهایی 2: با گرفتن ورودیهای gb و pb و C و M قسمت دیگری از خروجیهای نهایی و اصلی یعنی F و AeqB را تولید میکند.

```

module modulefinal2(
    input [3:0] gb,
    input [3:0] pb,
    input [3:0] C,
    input M,
    output [3:0] F,
    output AeqB
);

    wire [3:0] gxp, candm;

    xor gxpgate1(gxp[0], gb[0], pb[0]);
    xor gxpgate2(gxp[1], gb[1], pb[1]);
    xor gxpgate3(gxp[2], gb[2], pb[2]);
    xor gxpgate4(gxp[3], gb[3], pb[3]);

    or CM0gate(candm[0], C[0], M);
    or CM1gate(candm[1], C[1], M);
    or CM2gate(candm[2], C[2], M);
    or CM3gate(candm[3], C[3], M);

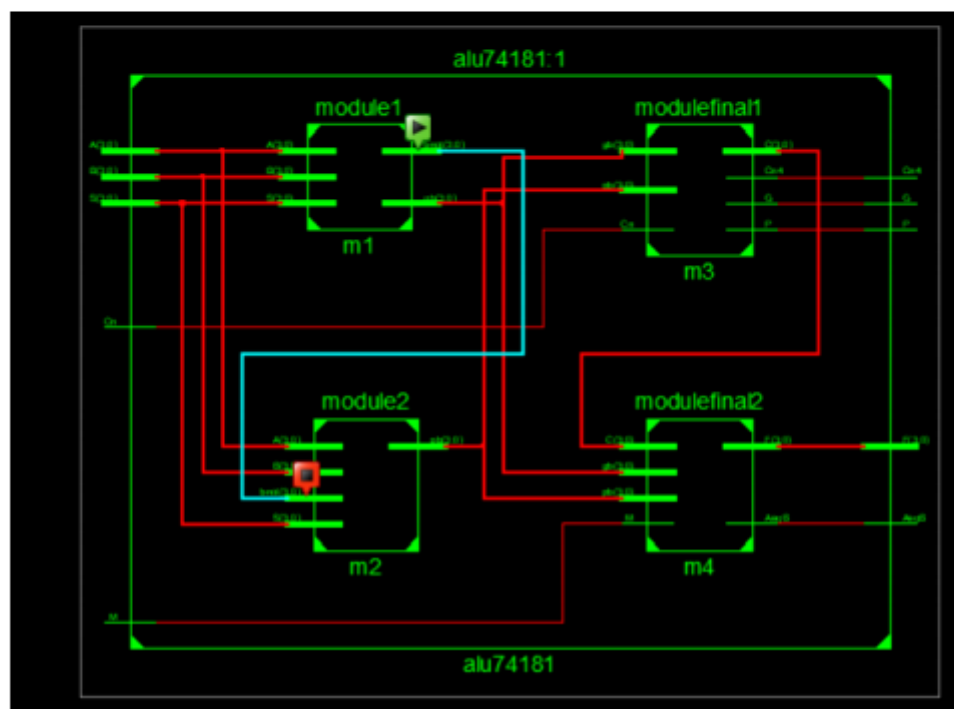
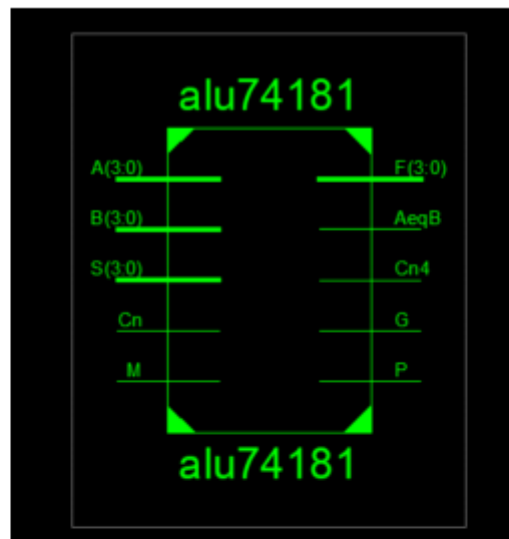
    xor xorf1(F[0], gxp[0], candm[0]);
    xor xorf2(F[1], gxp[1], candm[1]);
    xor xorf3(F[2], gxp[2], candm[2]);
    xor xorf4(F[3], gxp[3], candm[3]);

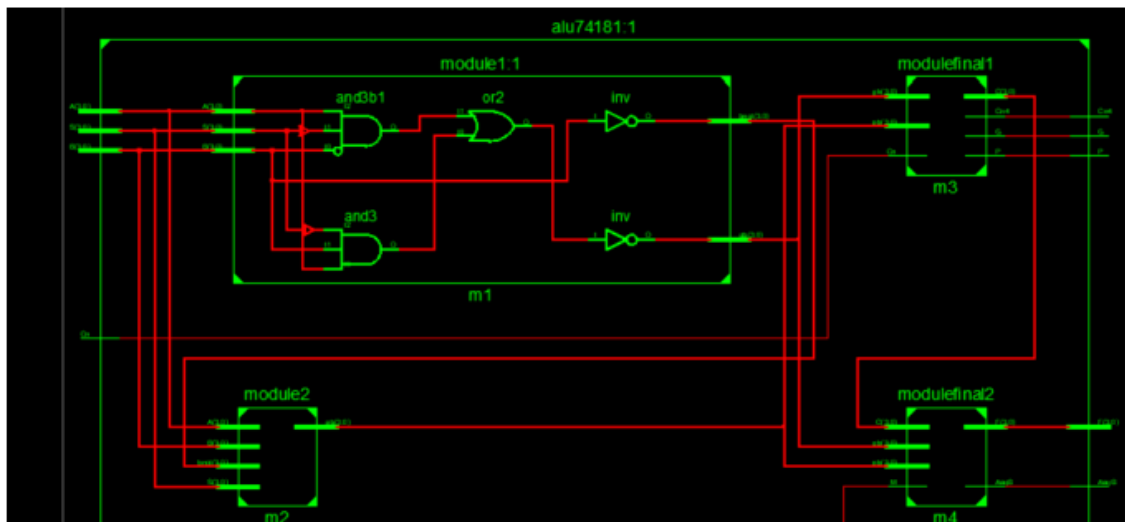
    //AeqB= &F
    and andAeqB(AeqB, F[0], F[1], F[2], F[3]);

endmodule

```

شماتیک:





گزارش تاخیر مدارها:

Data Sheet report:

All values displayed in nanoseconds (ns)

Pad to Pad

Source Pad	Destination Pad	Delay
A<0>	AeqB	7.937
A<0>	Cn4	7.225
A<0>	F<0>	5.801
A<0>	F<1>	6.513
A<0>	F<2>	6.542
A<0>	F<3>	7.225
A<0>	G	5.801
A<0>	P	6.513
A<1>	AeqB	7.937
A<1>	Cn4	7.225
A<1>	F<1>	5.801
A<1>	F<2>	6.542
A<1>	F<3>	7.225

A<1>	G		5.801
A<1>	P		6.513
A<2>	AeqB		7.225
A<2>	Cn4		7.225
A<2>	F<2>		6.079
A<2>	F<3>		6.513
A<2>	G		5.801
A<2>	P		6.513
A<3>	AeqB		6.513
A<3>	Cn4		6.513
A<3>	F<3>		5.801
A<3>	G		5.801
A<3>	P		5.801
B<0>	AeqB		7.937
B<0>	Cn4		7.225
B<0>	F<0>		5.801
B<0>	F<1>		6.513
B<0>	F<2>		6.542
B<0>	F<3>		7.225
B<0>	G		5.801
B<0>	P		6.513
B<1>	AeqB		7.937
B<1>	Cn4		7.225
B<1>	F<1>		5.801
B<1>	F<2>		6.542
B<1>	F<3>		7.225
B<1>	G		5.801
B<1>	P		6.513
B<2>	AeqB		7.225
B<2>	Cn4		7.225
B<2>	F<2>		6.079
B<2>	F<3>		6.513
B<2>	G		5.801
B<2>	P		6.513
B<3>	AeqB		6.513
B<3>	Cn4		6.513
B<3>	F<3>		5.801
B<3>	G		5.801
B<3>	P		5.801
Cn	AeqB		7.225
Cn	Cn4		5.801
Cn	F<0>		5.089
Cn	F<1>		5.801
Cn	F<2>		5.830
Cn	F<3>		6.513
M	AeqB		6.079
M	F<0>		5.089
M	F<1>		5.089

شکل موج:

