



دانشگاه صنعتی اصفهان
دانشکده برق و کامپیوتر

به نام خدا

درس : زبان های توصیف سخت افزار و مدارها

استاد درس : دکتر امیر خورسندی

گزارش پروژه تحت عنوان " عملیات ریاضی روی ماتریس ها در قالب سخت افزار "

اعضای تیم :

محمد جواد عرب

پویا بهزادی فر

علیرضا میرحسینی

معرفی

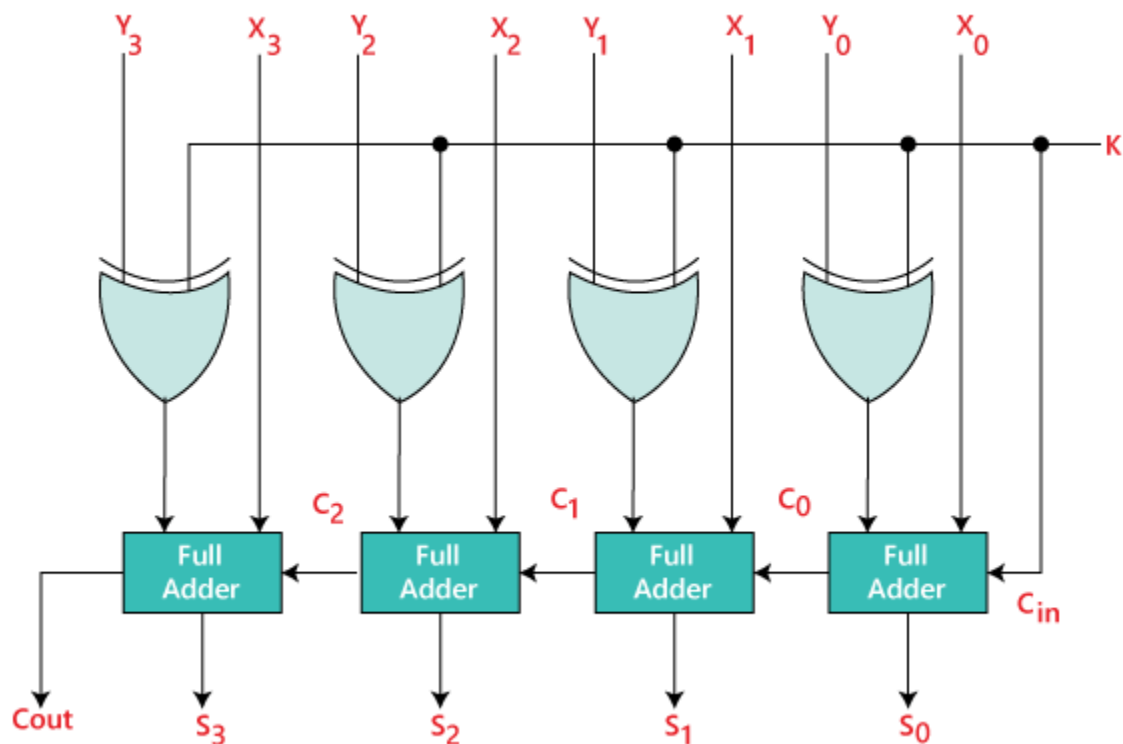
این پروژه در دو مرحله انجام شده است.

در مرحله اول عملیات ریاضی روی ماتریس ها به سبک زبان های برنامه نویسی سطح بالا و در مرحله دوم به سبک سلسله مراتبی و با دید سخت افزاری کد نویسی شده است.

طراحی سلسله مراتبی

جمع و تفریق

برای اعمال جمع و تفریق از ساختار جمع کننده توام استفاده شده است .



در مدارهای دیجیتال، جمع کننده توام مداری است که قادر به جمع و تفریق اعداد باینری در یک مدار واحد است. عملیات در حال انجام بستگی به مقدار دودویی سیگنال کنترل دارد.

این مدار یکی از اجزای ALU می باشد که از گیت های xor و full adder استفاده میکند.
برای مثال دو عدد باینری ۴ بیتی X و Y را به عنوان ورودی مدار دیجیتال برای عملیات با ارقام در نظر می گیریم.

X0 X1 X2 X3 FOR X

Y0 Y1 Y2 Y3 FOR Y

مدار شامل چهار FULL ADDER است زیرا در حال انجام عملیات روی اعداد ۴ بیتی هستیم.
همچنین یک خط کنترلی K وجود دارد که دارای مقدار دودویی ۰ یا ۱ می باشد که تعیین می کند عملیات در حال انجام جمع یا تفریق است.
همانطور که در شکل صفحه قبل نشان داده شده ، CIN اولین FULL ADDER مستقیماً به خط کنترلی متصل شده است. برای ورودی دوم X0 مستقیماً به FULL ADDER داده شده و برای ورودی سوم حاصل XOR خط کنترلی و Y0 داده شده است. در خروجی هم حاصل جمع یا تفریق (S0) و بیت نقلی (C0) حاصل میشود.

حال اگر مقدار خط کنترلی یک باشد، خروجی گیت XOR مکمل Y0 خواهد بود که در این صورت حاصل $X0+Y0'+K$ که برابر $X0-Y0$ است محاسبه خواهد شد.
به صورت مشابه اگر خط کنترلی صفر باشد، حاصل XOR برابر Y0 خواهد بود و عبارت $X0+Y0$ محاسبه خواهد شد.

در مرحله بعد C0 به صورت سریالی به FULL ADDER دوم پاس داده و S0 نیز به عنوان کم ارزش ترین بیت جواب نهایی ذخیره خواهد شد. A1 A2 A3 به صورت مستقیم و B1 B2 B3 با خط کنترلی XOR شده و به FULL ADDER های بعدی داده خواهد شد. C1 C2 به صورت سریالی به FULL ADDER های بعدی پاس داده خواهد شد و C3 نیز بیت نقلی نهایی جمع/تفریق خواهد بود.
به صورت کلی برای جمع/تفریق کننده N-بیتی توام به FULL ADDER N نیاز داریم.

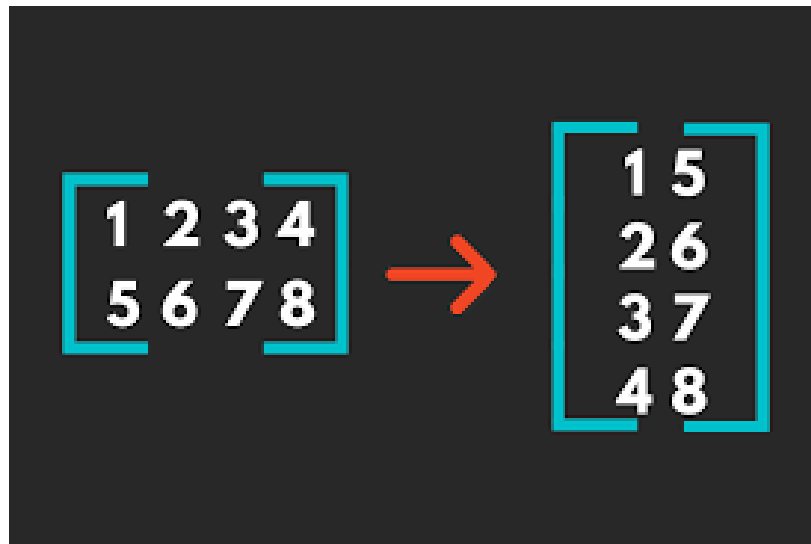
ضرب کننده

برای عملیات ضرب از جمع و شیفت استفاده میکنیم.

	b	1011	
	a	1110	
<hr/>			
m0	0000	→	1011 * 0
m1	1011X	→	1011 * 1 , یک بیت شیفت به سمت چپ
m2	1011XX	→	1011 * 1 , دو بیت شیفت به سمت چپ
m3	1011XXX	→	1011 * 1 , سه بیت شیفت به سمت چپ
<hr/>			
	1011010		

مثلا برای ضرب دو عدد چهار بیتی A و B از کم ارزش ترین بیت A شروع میکنیم اگر صفر بود کاری انجام نمی دهیم ولی اگر یک بود B را به اندازه ی جایگاه بیت A شیفت میدهیم. برای مثال در مرحله ی اول صفر بار، در مرحله ی دوم یک بار و ... B را شیفت میدهیم. در مرحله ی آخر حاصل همه ی ارقام شیفت پیدا کرده را با یکدیگر جمع میکنیم. نتیجه برابر حاصل ضرب دو عدد A و B خواهد بود.

ترانهاده



برای محاسبه ی ترانهاده یک ماتریس به طول و عرض $M \times N$ ابتدا ماتریسی به طول و عرض $N \times M$ تشکیل می‌دهیم و هر بار $Matrice[i][j] = MatriceTranspose[j][i]$ قرار می‌دهیم.

برای این کار از `generate` استفاده شده است.

به تعداد درایه های ماتریس، ماژول `transpose` فراخوانی شده و درایه را در مکان درست قرار می‌دهد.

طراحی به سبک زبان های برنامه نویسی سطح بالا

جمع

در دو حلقه تو در تو (nested) درایه های نظیر به نظیر با هم جمع میشوند.

```
//sumation
for(i=0 ; i<3 ; i = i + 1)
    for(j=0 ; j<3 ; j = j + 1)
        sum[i][j] = a[i][j]+b[i][j];
```

تفریق

در دو حلقه تو در تو (nested) درایه های نظیر به نظیر با هم تفریق میشوند.

```
//Subtract
for(i=0 ; i<3 ; i = i + 1)
    for(j=0 ; j<3 ; j = j + 1)
        sub[i][j] = a[i][j]-b[i][j];
```

ترانهاده

در دو حلقه تو در تو (nested) مقادیر درایه های $[i][j]$ ماتریس اولیه در درایه ی $[j][i]$ ماتریس ترانهاده قرار میگیرند.

```
7
8      //Transpose
9      for(i=0 ; i<3 ; i = i + 1)
10         for(j=0 ; j<3 ; j = j + 1)
11            transpose[j][i] = a[i][j];
```

ضرب

در سه حلقه تو در تو (nested) مقادیر حاصل ضرب را در ماتریس resault ذخیره میکنیم.

```
6
7
8      //Multiplication
9      for(i=0 ; i<3 ; i = i + 1)
10         for(j=0 ; j<3 ; j = j + 1)
11            begin
12               result[i][j] = 8'd0;
13               for(k=0 ; k<3 ; k = k + 1)
14                  result[i][j] = result[i][j] + (a[i][k]*b[k][j]);
15            end
```

نمونه ای از RTL SCHEMATIC کد سلسله مراتبی

