



دانشکده‌ی مهندسی برق و کامپیوتر

طراحی کامپیوتری سیستم‌های دیجیتال

تمرین سری اول

مدرس: سید محمد صدرالساداتی

۱ سوالات

۱. ضرب ماتریس با استفاده از آرایه systolic^۱ (۴۰ نمره)

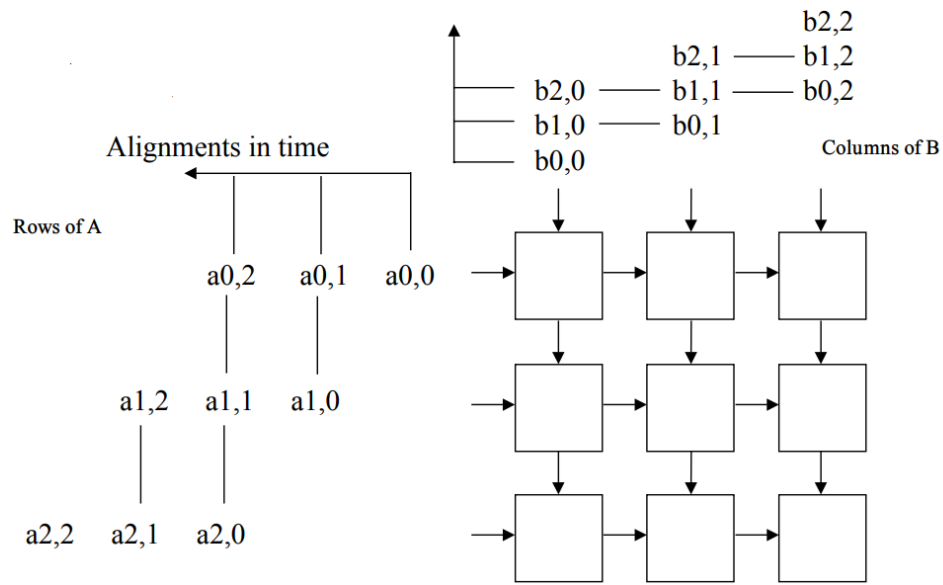
یک آرایه systolic، شبکه‌ای همگن از واحدهای پردازشی بهم متصل یا به اصطلاح سلول‌هاست. هر سلول به عنوان یک تابع عمل می‌کند به این صورت که از سلول‌های همسایه خود به عنوان ورودی داده‌هایی دریافت می‌کند و به صورت مستقل از بقیه‌ی سلول‌ها، بر روی این داده‌ها قسمتی از محاسبات را انجام داده و نتیجه را در سلول خود ذخیره می‌کند. این ساختار در بسیاری از الگوریتم‌های موازی کاربرد دارد. برای مثال، می‌توان از این ساختار در ضرب ماتریس در ماتریس استفاده کرد.

دو ماتریس A و B از سایز 3×3 را در نظر بگیرید. شکل ۱، آرایه systolic برای ضرب ماتریس را نشان می‌دهد.

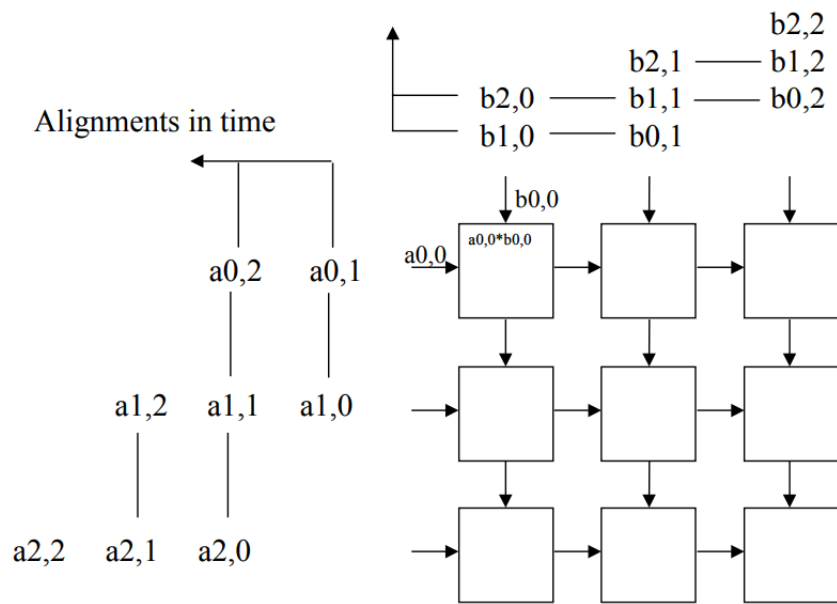
هر سلول مقادیر دریافتی از همسایه‌ها را در یکدیگر ضرب کرده و با مقدار ذخیره شده در سلول خود جمع می‌کند. برای مثال در مرحله اول، دو مقدار $a_{0,0}$ و $b_{0,0}$ وارد اولین سلول شده و حاصل ضرب آنها در سلول ذخیره می‌شود.

شکل ۲ اولین مرحله الگوریتم را نشان می‌دهد. بعد از انجام ۷ مرحله نتیجه نهایی

¹Systolic Array Matrix Multiplication

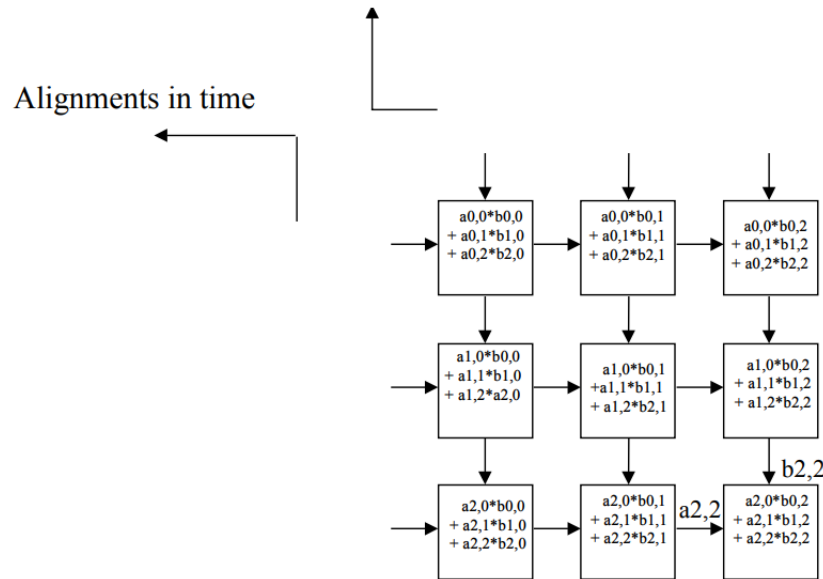


شکل ۱: آرایه systolic ضرب ماتریس



شکل ۲: مرحله اول الگوریتم ضرب ماتریس

بدست می آید. شکل ۳ حاصل ضرب دو ماتریس را نشان می دهد.



شکل ۳: حاصل ضرب دو ماتریس

این الگوریتم را برای دو ماتریس از سایز $N \times N$ در سطح dataflow پیاده سازی کنید (فرض کنید داده های ماتریس ها عدد صحیح مثبت M بیتی هستند). prototype ماژول را به صورت زیر تعریف کنید:

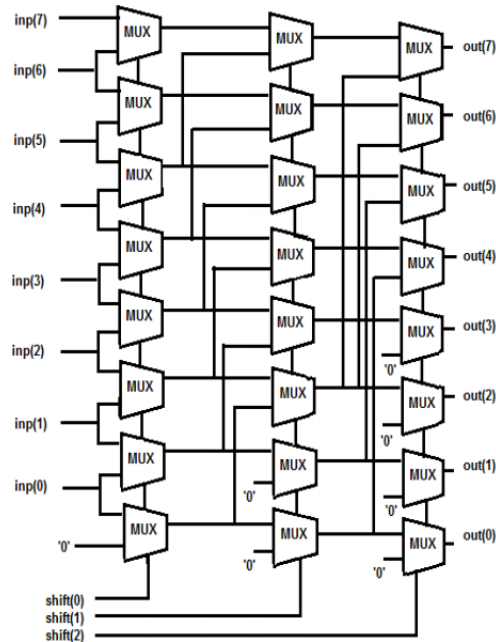
```
module mux-mul #(parameter N=3, parameter log_N=1, parameter M=4)
(data-left, data-top, reset, clk, result)
input  [N*M-1:0] data-left, data-top;
input  reset, clk;
output [(N*N*(M*2+log_N+1))-1:0] result;
```

۲. شیفت دهنده بشکه ای^۲ (۳۰ نمره)

یک شیفت دهنده بشکه ای یک مدار منطقی است که داده ورودی را بر اساس یک ورودی کنترل کننده به تعداد بیت دلخواه شیفت می دهد. این شیفت دهنده از چند لایه مالتی پلکسر تشکیل شده است بطوریکه خروجی یک مالتی پلکسر به ورودی

²Barrel Shifter

مالتی پلکسر بعدی متصل است. به عبارت دیگر، یک N بیت شیفت دهنده از N مالتی پلکسر ۲ به ۱ در $\log_2 N$ لایه تشکیل شده است. بنابراین، برای طراحی یک ۸ بیت شیفت دهنده بشکه‌ای به سه لایه و در هر لایه به ۸ مالتی پلکسر ۲ به ۱ نیاز است. شکل ۴، مدار شیفت دهنده راست بشکه‌ای را نشان می‌دهد. $inp(j)$ ، $out(j)$ و $shift(k)$



شکل ۴: شیفت دهنده بشکه‌ای

متناظر با j امین بیت ورودی، j امین بیت خروجی و k امین بیت انتخاب است و مقادیر k در بازه ۰ تا ۷ و ۰ تا ۲ می‌باشد. در هر لایه i ، اگر بیت انتخاب آن برابر ۱ باشد، به میزان 2^{i-1} بیت شیفت صورت می‌گیرد. برای مثال اگر ورودی شیفت برابر ۱۰۰ باشد، دو لایه اول، ورودی inp را بدون انجام شیفت به لایه سوم انتقال داده و در لایه سوم، ورودی به اندازه ۴ بیت شیفت داده می‌شود. در حالت کلی، یک شیفت دهنده بشکه‌ای ۸ بیتی با ترکیب شیفت‌ها در سه لایه، می‌تواند یک کلمه را به اندازه ۰ تا ۷ بیت شیفت دهد.

یک شیفت دهنده N بیتی در سطح dataflow طراحی کنید. prototype این ماژول را به صورت زیر تعریف کنید:

```

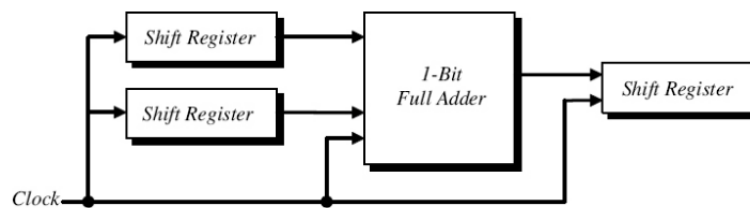
module barrel_shifter #(parameter N=8, parameter logN=3)
    (in, shift_selection, out)

input  [N-1:0] in;
input  [logN-1:0] shift_selection;
output [N-1:0] out;

```

۳. جمع کننده سریال^۳ (۳۰ نمره)

یک جمع کننده سریال شامل یک جمع کننده کامل یک بیتی و چندین شیفت دهنده است. برای مثال، فرض کنید می خواهیم دو مقدار N بیتی A و B را باهم جمع کنیم. در این جمع کننده، دو شیفت دهنده راست برای ذخیره دو مقدار A و B و یک شیفت دهنده راست برای ذخیره حاصل جمع استفاده می شود. شکل ۵، دیاگرام کلی یک جمع کننده سریال را نشان می دهد.



شکل ۵: جمع کننده سریال

این جمع کننده را برای دو عدد N بیتی در سطح dataflow پیاده سازی کنید. prototype ماژول را به صورت زیر تعریف کنید:

```

module serial_adder #(parameter N=4)
(a, b, reset, load, clk, sum, cout)

input  [N-1:0] a,b;
input  reset, load, clk;
output [N-1:0] sum;
output cout;

```

³Serial Adder

۴. فایل ثبات^۴ (۲۰ نمره اضافی)

یک فایل ثبات در پردازنده‌های مدرن شامل چندین بانک است و هر بانک شامل یک پورت ورودی برای نوشتن، یک پورت خروجی برای خواندن، دو ورودی آدرس یکی برای خواندن و یکی برای نوشتن و سیگنال فعالسازی نوشتن است. آدرس‌دهی ثبات‌ها بین بانک‌ها به صورت low order interleaving است. به این معنا که آدرس‌های متوالی ثبات‌ها در بانک‌های مختلف قرار می‌گیرند. برای مثال، فرض کنید ۶۴ ثبات ۸ بیتی و ۴ بانک داریم. یعنی هر بانک شامل ۱۶ ثبات است. آدرس‌دهی ثبات‌ها بین بانک‌ها با low order interleaving به صورت زیر است:

Bank0: Registers 00,04,08,...,60

Bank1: Registers 01,05,09,...,61

Bank2: Registers 02,06,10,...,62

Bank3: Registers 03,07,11,...,63

فایل ثبات مورد نظر را در سطح behavioral پیاده‌سازی کنید. prototype ماژول را به صورت زیر تعریف کنید:

```
module Register_File #(parameter bank_num=4, parameter log_bank_num=2,
parameter reg_num=64, parameter log_reg_num=6,
parameter reg_width=64)
(write-data, addr-write, addr-read, reset, write-en, clk, read-data)
input [reg_width-1:0] write-data;
input [log_reg_num-1:0] addr-write, addr-read;
input reset, write-en, clk;
output reg [reg_width-1:0] read-data;
```

⁴Register file

۲ نحوه ارسال و زمان تحویل

۱. فایل پروژه به همراه تمامی testbench ها به صورت یک فایل zip در سامانه CECM بارگذاری شود.

۲. مهلت نهایی ارسال پروژه تا پایان روز ۱۹ اسفند ۹۷ می باشد و به هیچ عنوان تمدید نخواهد شد.

۳ یادآوری‌های عمومی

۱. در صورت مشاهده تقلب در تکالیف، بار اول تمام نمره آن تمرین و بار دوم تمام نمرات کل تکالیف برای تقلب دهنده و گیرنده صفر منظور خواهد شد.
۲. در صورتی که فایل پروژه ارسالی تنها کامپایل شده و خطایی نداشته باشد، حداکثر ۵۰ درصد نمره آن تمرین را دریافت خواهید کرد.
۳. هر فرد می‌تواند تا پایان ترم در مجموع با حداکثر ۴۸ ساعت تاخیر تکالیف‌های خود را ارسال کند.
۴. ابهامات و سوالات خود را می‌توانید در forum ایجاد شده در سامانه CECM مطرح نمایید.
۵. لطفا در راستای بهبود هرچه بهتر روال طرح تمرین‌ها، نظرات و انتقادات خود را به ایمیل m.sadr89@gmail.com ارسال نمایید.