

۱ گام اول

در این گام، کد ضرب ماتریس‌هایی با حداکثر اندازه ۳۲ در ۳۲ را می‌نویسیم.

در صورتی که در ابتدای برنامه خط

```
#define DEBUG
```

قرار گیرد، نتایج محاسبات پس از اجرای کرنل و پیش از آزادسازی حافظه‌ها بر روی میزبان (CPU) به شکل تکنخی بررسی می‌شود. اگر ضرب ماتریس‌ها نادرست انجام شده باشد، در این قسمت پیغام wrong answer چاپ می‌شود.

۲ گام دوم

در این گام سه راه‌حل مختلف برای ضرب ماتریس‌ها پیاده‌سازی شده است. در روش اول تنها ۱۰۲۴ نخ در یک بلوک تمام محاسبات را انجام می‌دهند. این روش در کرنل matMulA1Kernel پیاده‌سازی شده است. روش دوم که در کرنل matMulA2Kernel پیاده شده است، به تعداد درایه‌های ماتریس‌ها نخ در بلوک‌های مختلف اختصاص می‌دهد. در روش سوم و آخر از تکنیک tiling با اندازه بلوک و کاشی‌های ۳۲ در ۳۲ استفاده شده است. روش سوم نیز در کرنل matMulA3Kernel آمده است.

در ابتدای برنامه با تغییر خط

```
#define APPROACH x
```

می‌توان برنامه را با راه‌حل شماره x اجرا کرد.

جدول ۱: زمان‌های اجرا (میلی ثانیه) به ازای اندازه ورودی‌های مختلف

تسریع	اندازه ورودی			موازی‌سازی
	2^{12}	2^{11}	2^{10}	
-	۵۳۸۶۵۱/۵	۵۶۲۸۰/۰	۱۷۵۳/۲	سریال
۹/۷۱	۵۳۴۵۵/۷	۶۰۱۱/۷	۶۸۷/۸	راه‌حل اول
۸۴۲/۶۰	۶۲۲/۰	۶۸/۷	۸/۵	راه‌حل دوم
۸۵۶/۹۰	۵۹۲/۸	۹.۶۹	۸.۸	راه‌حل سوم (کاشی‌کاری)

جدول ۲: زمان‌های اجرا (میلی ثانیه) به ازای اندازه ورودی‌های مختلف

تسریع	اندازه ورودی			موازی‌سازی
	2^{12}	2^{11}	2^{10}	
۹۸۰/۲۱	۵۱۵/۳	۶۱/۵	۸/۲	کاشی‌کاری با حافظه مشترک