

طراحی و پیاده‌سازی ضرب‌کننده ماتریس توسط Verilog

احمد سلیمی^۱، کیمیا نوربخش^۱، ساعی سعادت^۱، علیرضا حسین‌پور^۱

^۱ دانشگاه صنعتی شریف، دانشکده مهندسی کامپیوتر

چکیده—

کلمات کلیدی—

۱. مقدمه

۲. معماری سیستم

معماری این سیستم، از سه لایه اصلی تشکیل شده است. در ادامه، معماری و جزئیات هر یک از این لایه‌ها، توضیح داده شده است.

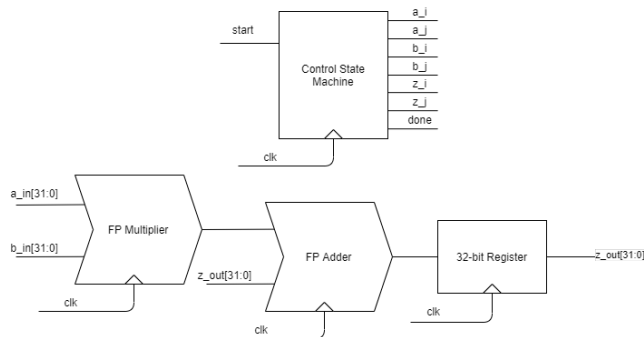
۲.۱. ضرب‌کننده ماتریس ترتیبی

در این ماژول مانند ضرب ماتریسی عادی، دو ماتریس $m \times m$ را در هم ضرب می‌کنیم. می‌دانیم که برای به دست آوردن درایه ij حاصلضرب، باید سطر i ام ماتریس اول را در ستون j ام ماتریس دوم ضرب کنیم. برای این موضوع به ازای هر $1 \leq i \leq m, 1 \leq j \leq m$ داریم:

$$R_{ij} = \sum_{k=1}^m A_{ik} \times B_{kj}$$

که در آن، R ماتریس $m \times m$ حاصل ضرب است. در این ماژول برای محاسبه جمع و ضرب‌ها، از ماژول‌های جمع‌کننده و ضرب‌کننده اعشاری^۱ استفاده می‌کنیم. ماژول ضرب‌کننده ماتریس ترتیبی^۲ این فرایند را در قالب یک ماشین حالت انجام می‌دهد. برای محاسبه درایه i, j ام، یک accumulator برای نگه‌داری جواب نهایی در نظر می‌گیریم و سپس به ازای هر k ، ابتدا با استفاده از ماژول FP_multiplier حاصل $A_{ik} \times B_{kj}$ را محاسبه می‌کنیم و با استفاده از ماژول FP_Adder، به ازای k های مختلف جواب را آپدیت می‌کنیم.

شکل ۱ بلوک دیاگرام ضرب‌کننده ماتریس ترتیبی را نشان می‌دهد. باید توجه کرد که حافظه‌ای که حاوی ماتریس‌های ورودی و ماتریس جواب است، در خارج این ماژول قرار دارد. در نتیجه، واحد کنترل در این ماشین



شکل ۱: بلوک دیاگرام ضرب‌کننده ماتریس ترتیبی.

حالت محدود^۳ اندیس‌های درایه‌های موردنیاز خود، یعنی $a_i, a_j, b_i, b_j, z_i, z_j$ را تعیین می‌کند، و مقادیر مربوط به هر درایه در ماتریس‌های ورودی در a_{in} و b_{in} قرار گرفته، و مقدار z_{out} نیز در ماتریس جواب قرار داده می‌شود.

۲.۲. ضرب‌کننده ماتریس سطری در ستونی

وظیفه این ماژول، این است که با استفاده از یک ماژول ضرب‌کننده ماتریس ترتیبی، حاصل ضرب یک ماتریس سطری $m \times n$ در یک ماتریس ستونی $n \times m$ را محاسبه کند. حاصل این ضرب، یک ماتریس $m \times m$ خواهد بود.

۲.۳. ضرب‌کننده ماتریس موازی

۳. شبیه‌سازی و نتایج

۴. سنتز و نتایج

۵. نتیجه‌گیری

توضیح پروسه‌ی stable و acknowledge: برای ارتباط بین تمامی ماژول‌ها، برای اطمینان از این که ورودی و خروجی‌ها هنگام استفاده شدن تغییر نمی‌کنند و مقدار صحیحی دارند، برای هر کدام دو سیگنال stable و acknowledge در نظر می‌گیریم. نحوه‌ی استفاده از آن‌ها بدین گونه است که ماژولی که مقدار را دارد

^۳Finite State Machine

^۱floating point adder and multiplier

^۲sequential matrix multiplier

و می‌خواهد آن‌را پاس بدهد، با استفاده از سیگنال stable به مازول گیرنده اعلام می‌کند که ورودی آماده‌ی استفاده است، سپس مازول گیرنده با استفاده از سیگنال acknowledge اعلام می‌کند که ورودی را با موفقیت دریافت کرده و مازول فرستنده می‌تواند آن را تغییر دهد.