به نام خدا

محمدفرحان بهرامی

401105729

پروژه امتیازی DSD

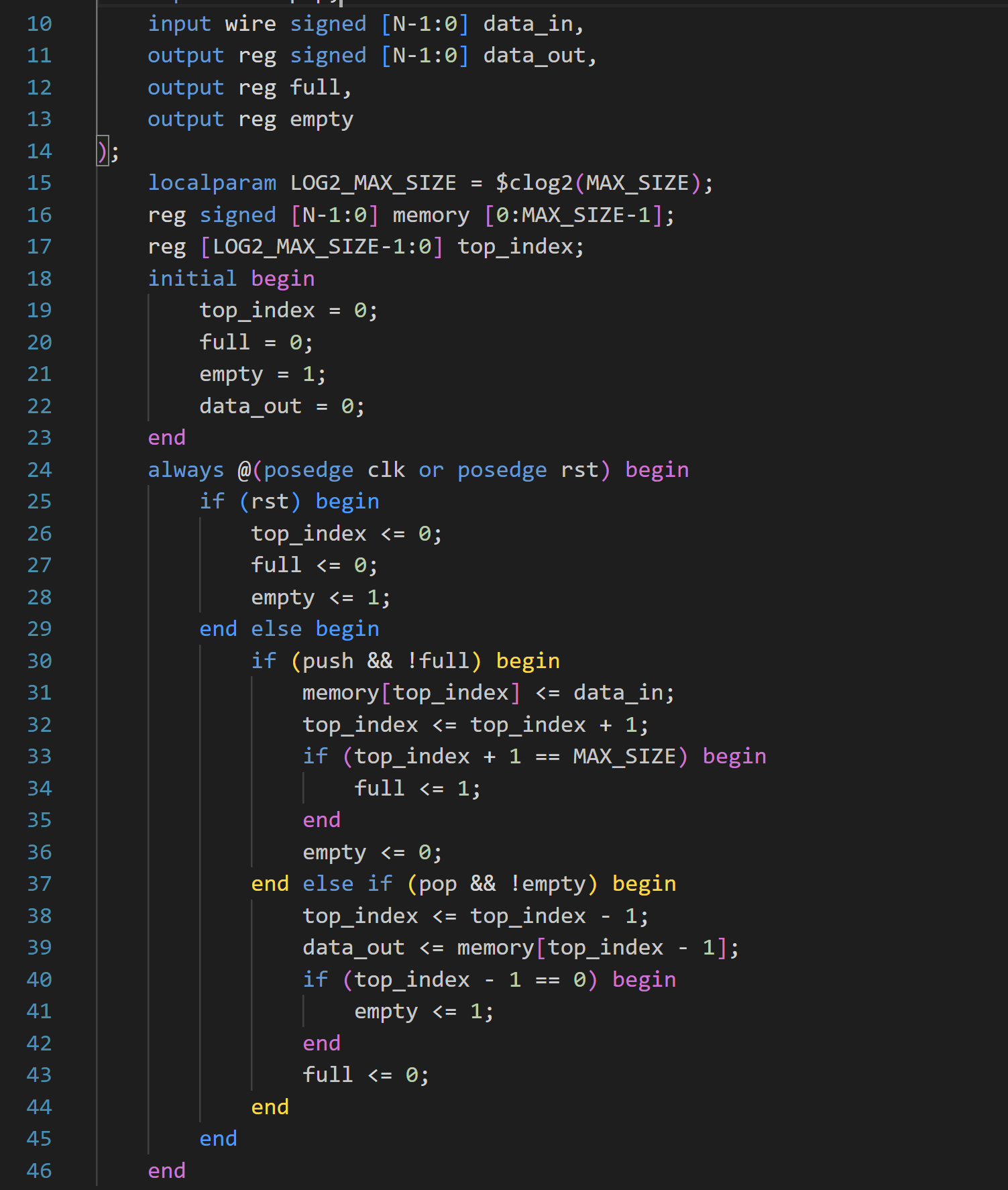
# سوال هفتم میانترم

## الف)

ماژول stack: در ابتدا، پشته از پارامترهای N و MAX\_SIZE استفاده می‌کند تا اندازه بیت‌های داده و حداکثر تعداد عنصرها رو مشخص کند. سیگنال‌های ورودی شامل کلاک (clk)، ریست (rst)، سیگنال‌ پوش و پاپ و همچنین داده‌های ورودی (data\_in) هستند. سیگنال‌های خروجی شامل داده‌های خروجی (data\_out)، پر بودن استک (full) و خالی بودن استک (empty) هستند.

استک از یک آرایه به نام memory برای ذخیره داده‌ها استفاده می‌کند که طول آن برابر با MAX\_SIZE است. یک اندیس به نام top\_index وجود دارد که به عنوان پوینتر به بالای استک عمل می‌کند. در ابتدا، top\_index برابر با صفر است و استک خالی است empty برابر با 1 و پر نیست full برابر با 0.

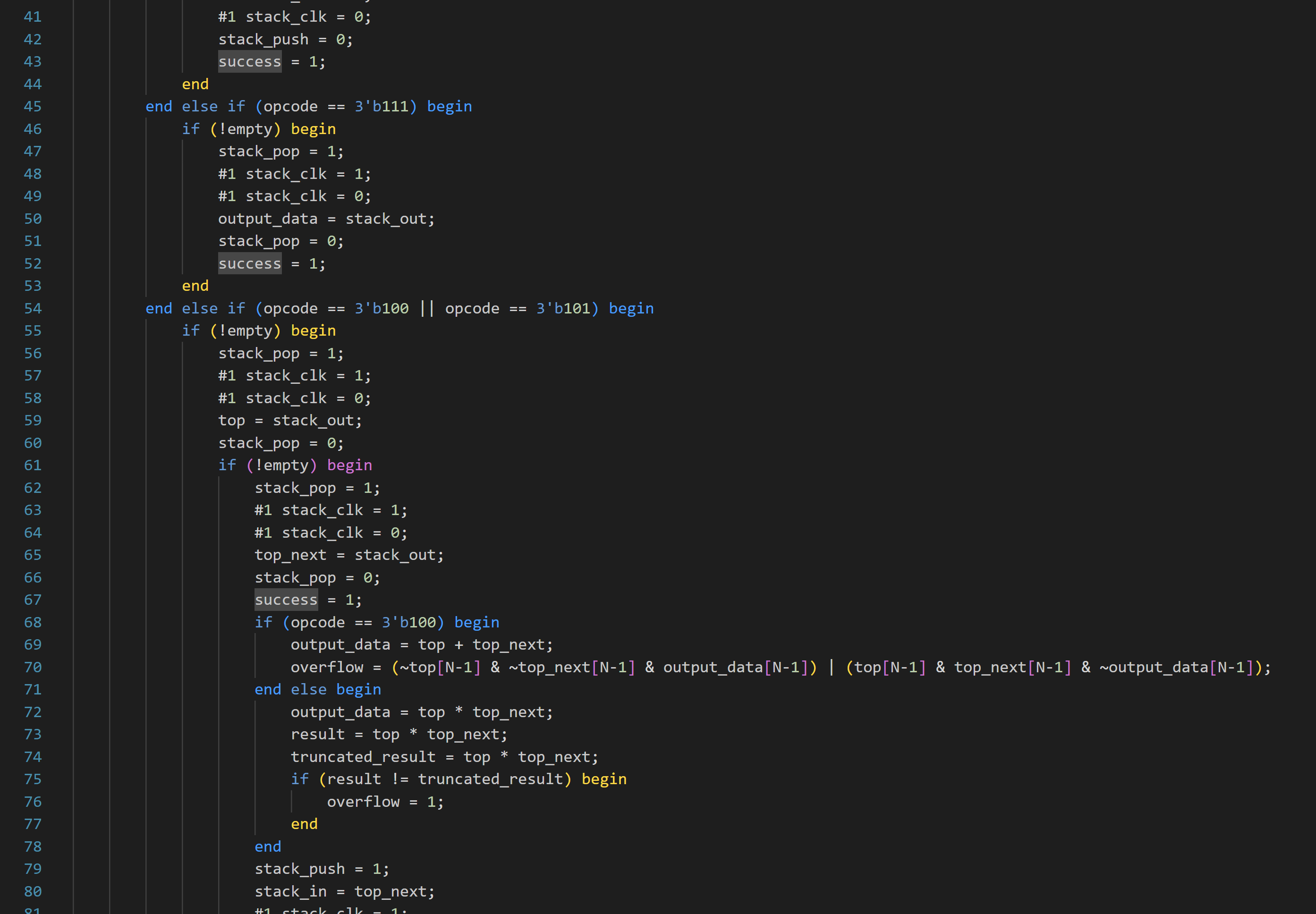
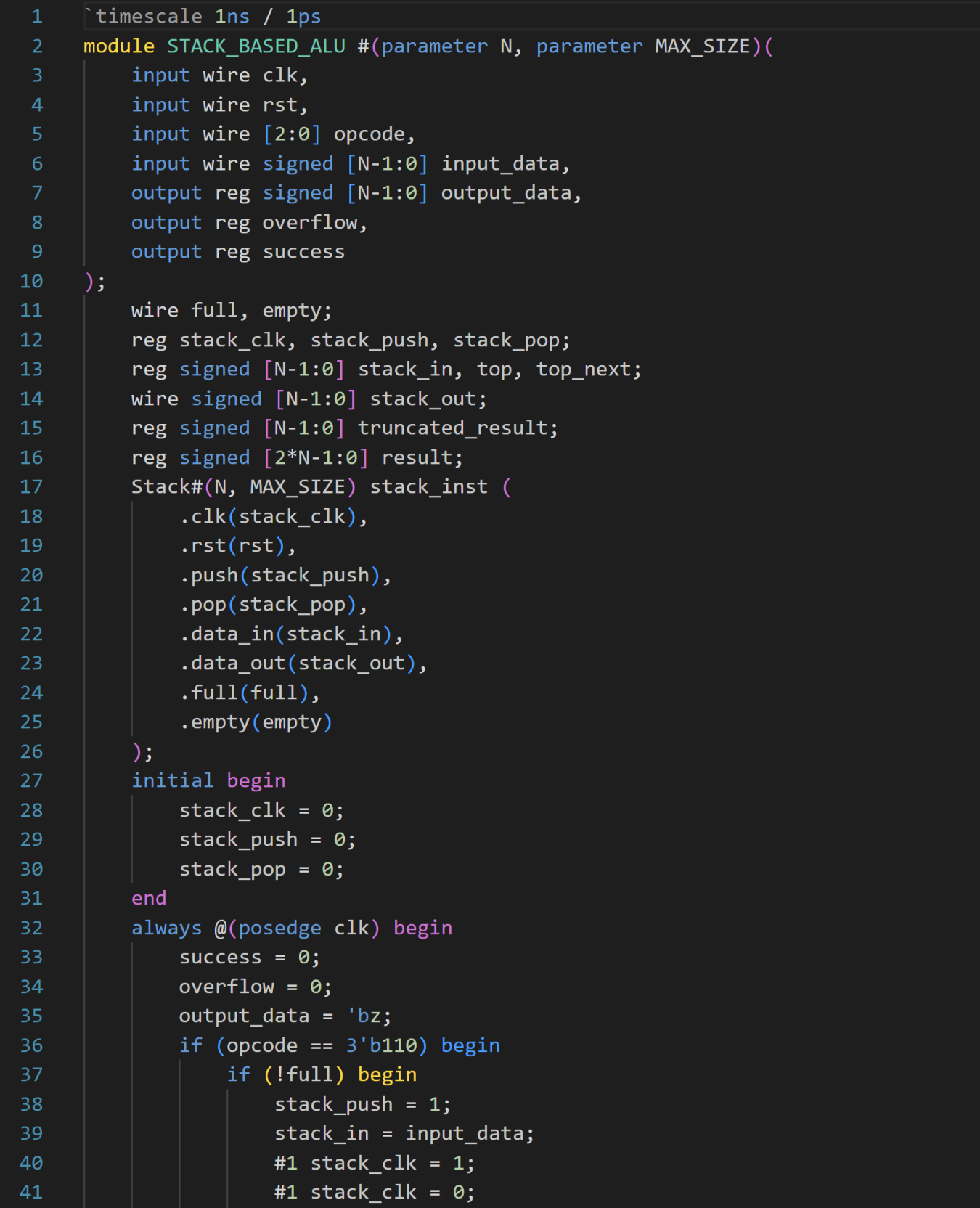
یک بلوک always وجود دارد که با هر کلاک مثبت یا ریست فعال می‌شود. اگر ریست فعال باشد، top\_index به صفر تنظیم شده و استک خالی و پر نبودن تنظیم می‌شود. اگر سیگنال push فعال باشد و استک پر نباشد، داده ورودی در بالای استک قرار می‌گیرد و top\_index یک واحد افزایش می‌یابد. اگر top\_index به حداکثر اندازه استک برسد، سیگنال full فعال می‌شود و سیگنال empty غیرفعال می‌گردد. اگر سیگنال pop فعال باشد و استک خالی نباشد، top\_index یک واحد کاهش می‌یابد و داده بالای استک به خروجی داده منتقل می‌شود. اگر top\_index به صفر برسد، سیگنال empty فعال می‌شود و سیگنال full غیرفعال می‌گردد.

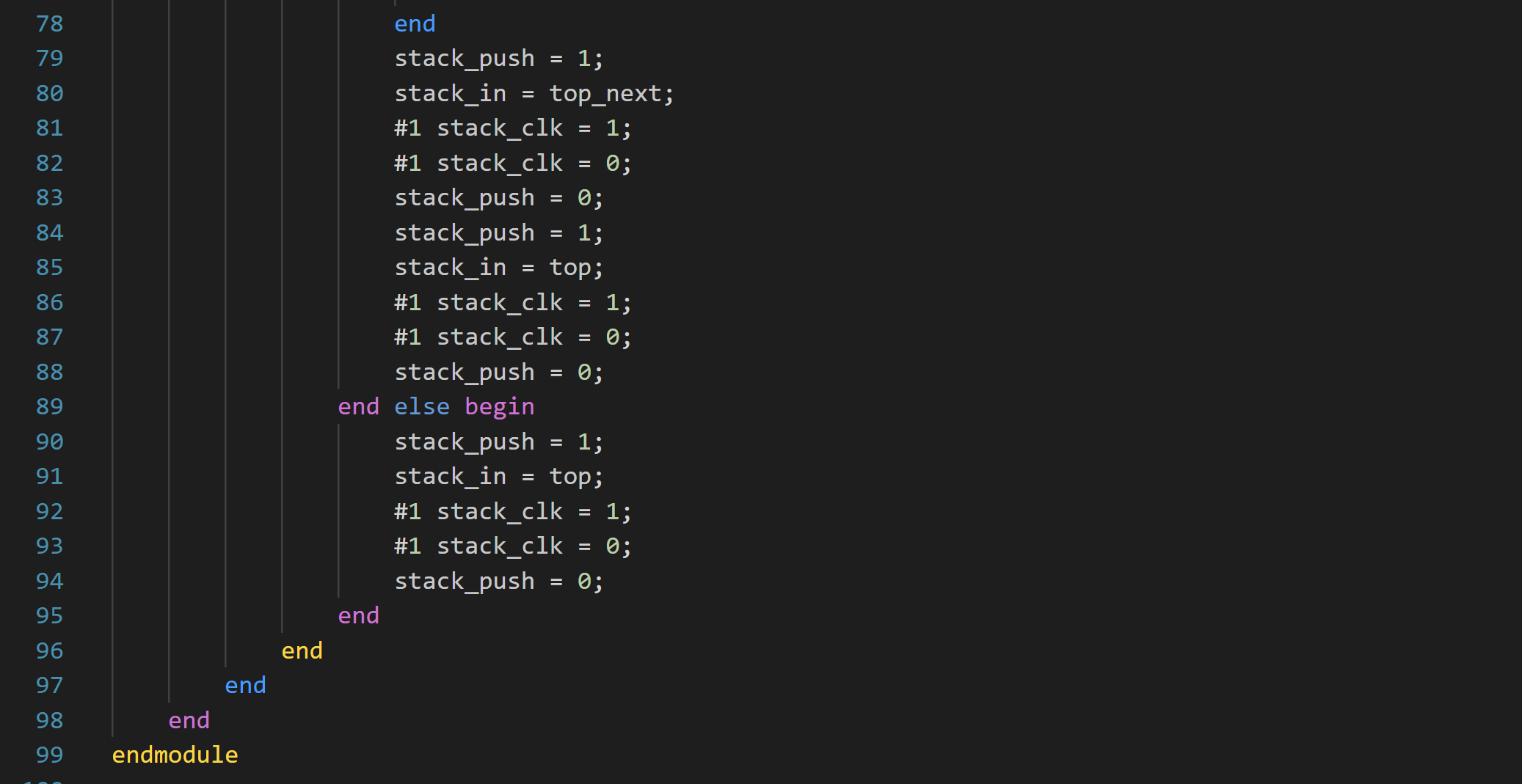


ماژول STACK\_BASED\_ALU: ورودی‌های این ماژول شامل سیگنال کلاک (clk)، ریست (rst)، (opcode) و داده‌های ورودی (input\_data) هستند. خروجی‌های ماژول شامل داده‌های خروجی (output\_data)، سیگنال (overflow) و سیگنال (success) می‌باشند. این ماژول از ماژول دیگری به نام Stack نمونه میگیرد و سیگنال های مربوطه را به این نمونه می دهیم.

اگر کد عملیاتی برابر با 3'b110 باشد، عملیات (push) انجام می‌شود. اگر استک پر نباشد، داده‌های ورودی به استک پوش می‌شوند و سیگنال موفقیت فعال می‌شود. اگر کد عملیاتی برابر با 3'b111 باشد، عملیات پاپ انجام می‌شود. اگر استک خالی نباشد، داده‌های بالای استک به خروجی منتقل می‌شوند و سیگنال موفقیت فعال می‌شود. اگر کد عملیاتی برابر با 3'b100 باشد، عملیات جمع انجام می‌شود و نتیجه به خروجی منتقل می‌شود. اگر اورفلو رخ دهد، سیگنال overflow فعال می‌شود. اگر کد عملیاتی برابر با 3'b101 باشد، عملیات ضرب انجام می‌شود. نتیجه کامل و نتیجه کوتاه شده به دست می‌آیند. اگر نتیجه کامل با نتیجه کوتاه شده برابر نباشد، سیگنال overflow فعال می‌شود.

عکس در صفحات بعدی میباشد...

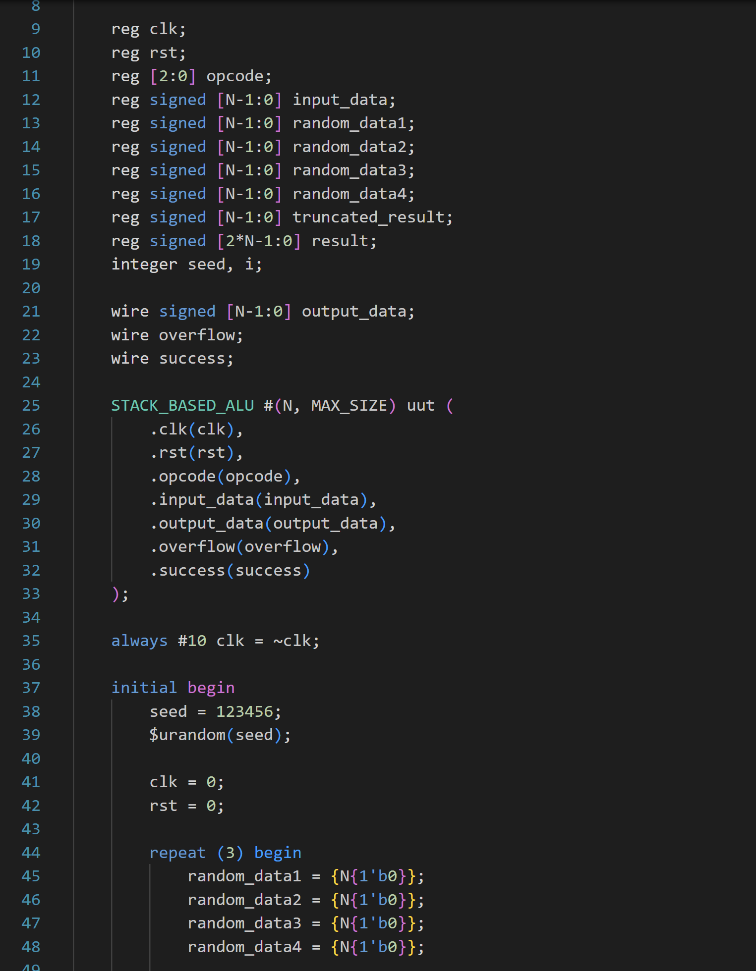


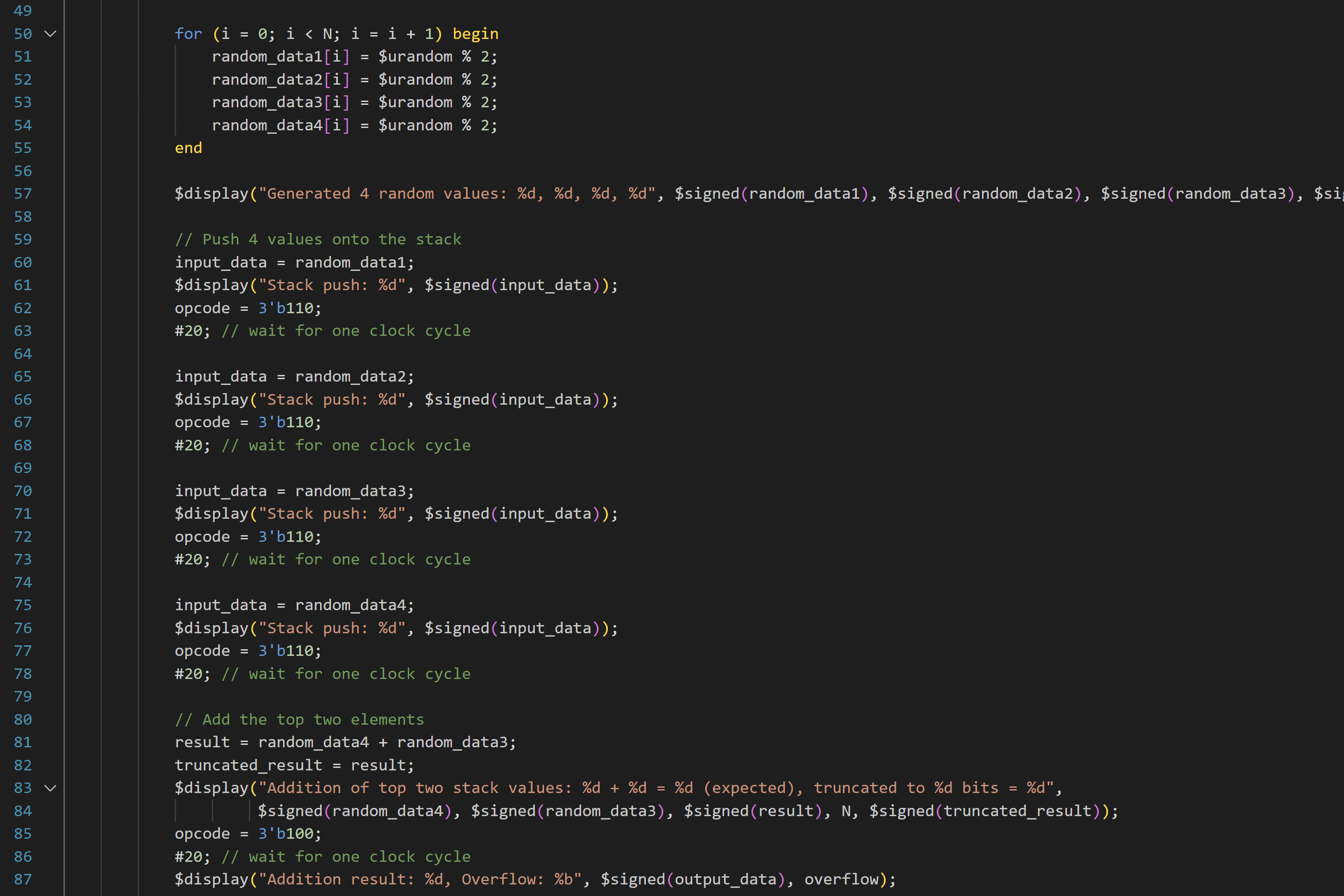


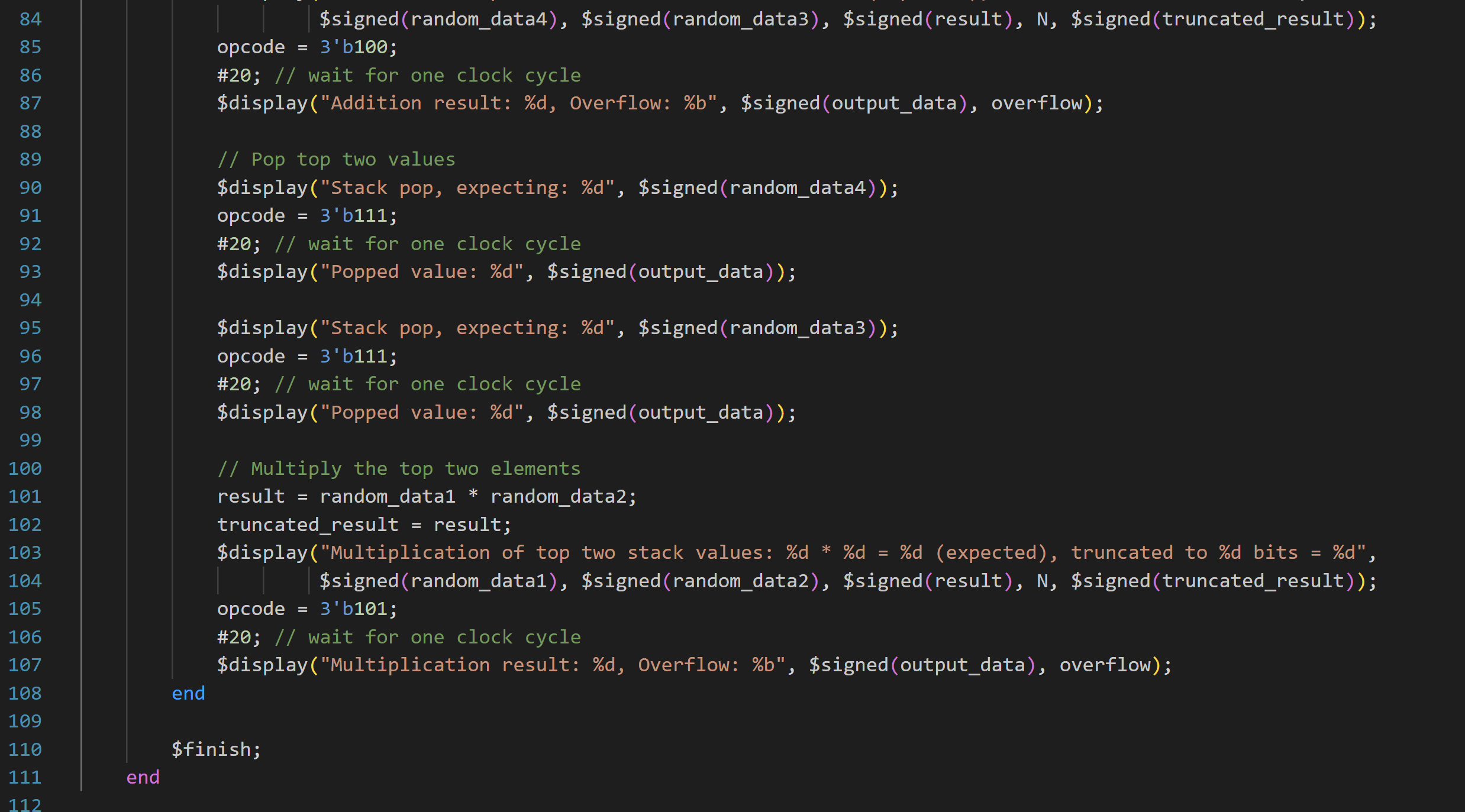
ماژول STACK\_BASED\_ALU\_tb: این کد یک تست بنچ برای ماژول ALU\_BASED\_STACK است. در این کد، پارامترهای N و MAX\_SIZE به ترتیب اندازه داده‌ها و حداکثر اندازه استک را مشخص می‌کنند. متغیرهای مختلفی از جمله سیگنال‌های کلاک (clk)، ریست (rst)، کد عملیاتی (opcode)، داده ورودی (input\_data)، و... را دارد.

در بخش همیشه (always)، سیگنال کلاک هر 10 نانوثانیه تغییر وضعیت می‌دهد. در بخش اولیه (initial)، سیگنال‌های کلاک و ریست مقداردهی اولیه می‌شوند و یک عدد برای تولید اعداد تصادفی تنظیم می‌شود. کد سه بار تکرار می‌شود، در هر تکرار چهار عدد تصادفی تولید می‌شوند و به استک افزوده می‌شوند. سپس عملیات جمع و ضرب روی دو عنصر بالای استک انجام می‌شود. در هر تکرار عملیات‌ها با تأخیر(20 نانوثانیه) انجام می‌شود. در ابتدا، داده‌های رندم تولید و به استک اضافه می‌شوند. سپس عملیات جمع دو عنصر بالای استک انجام می‌شود و نتیجه مورد انتظار با نتیجه واقعی مقایسه می‌شود. همچنین، دو عنصر بالای استک پاپ می‌شوند و مقدار واقعی پاپ شده. در نهایت، عملیات ضرب دو عنصر بالای استک انجام می‌شود.

نکته: برای تعیین کردن اندازه داده ها کافی است ما N را هر مقدار که بخواهیم مقدار دهی کنیم که ما در اینجا 4 بیتی در نظر میگیریم.





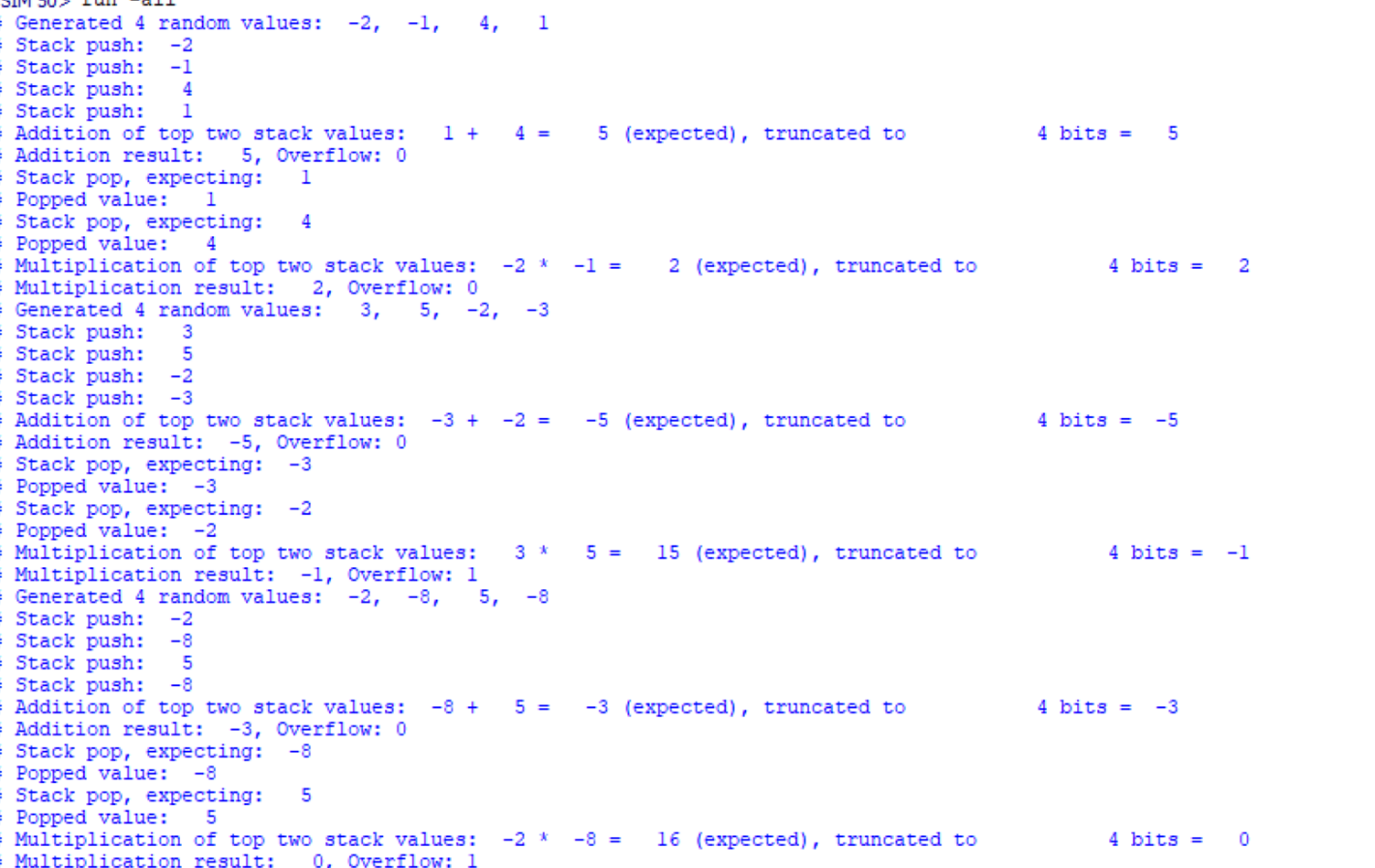


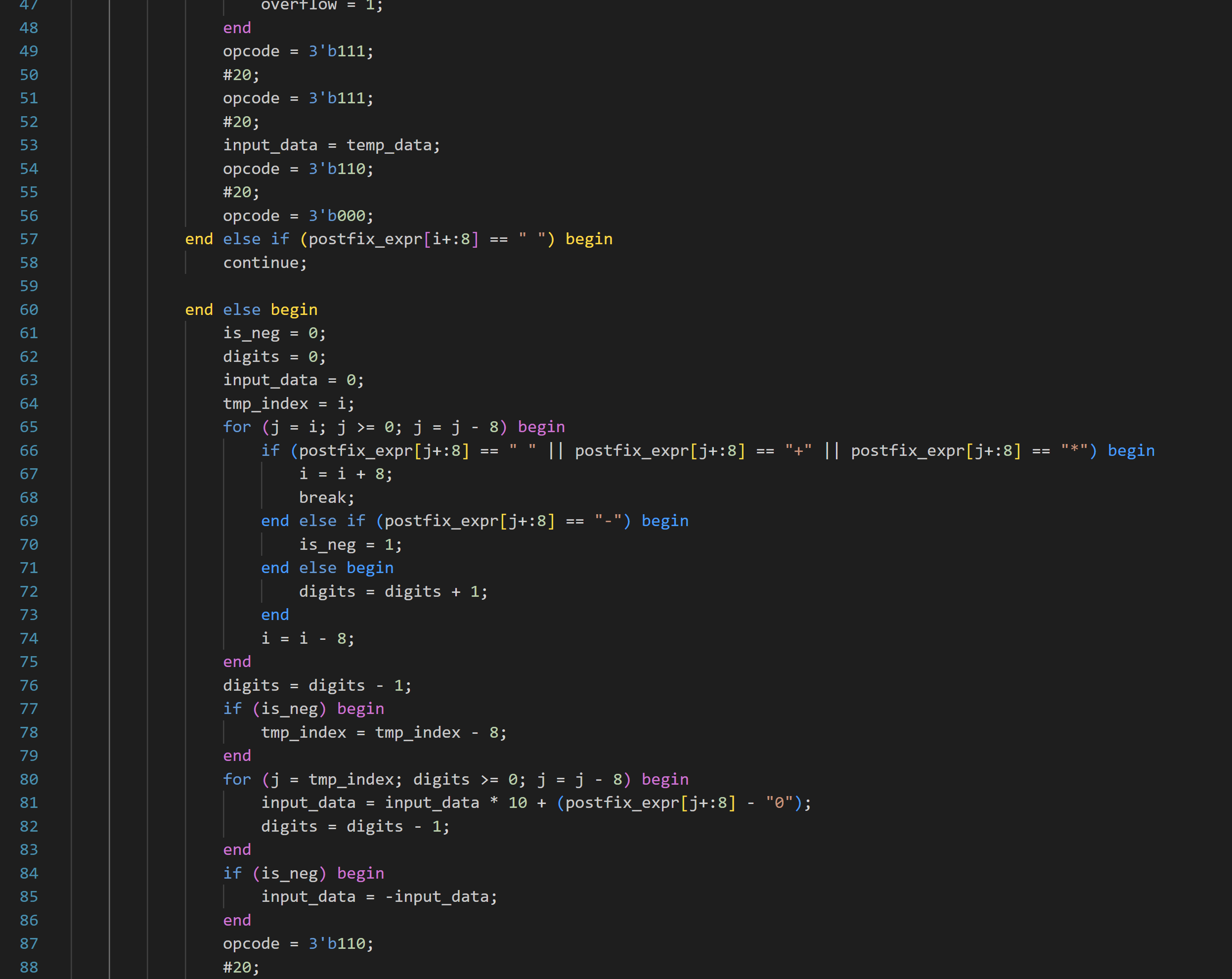
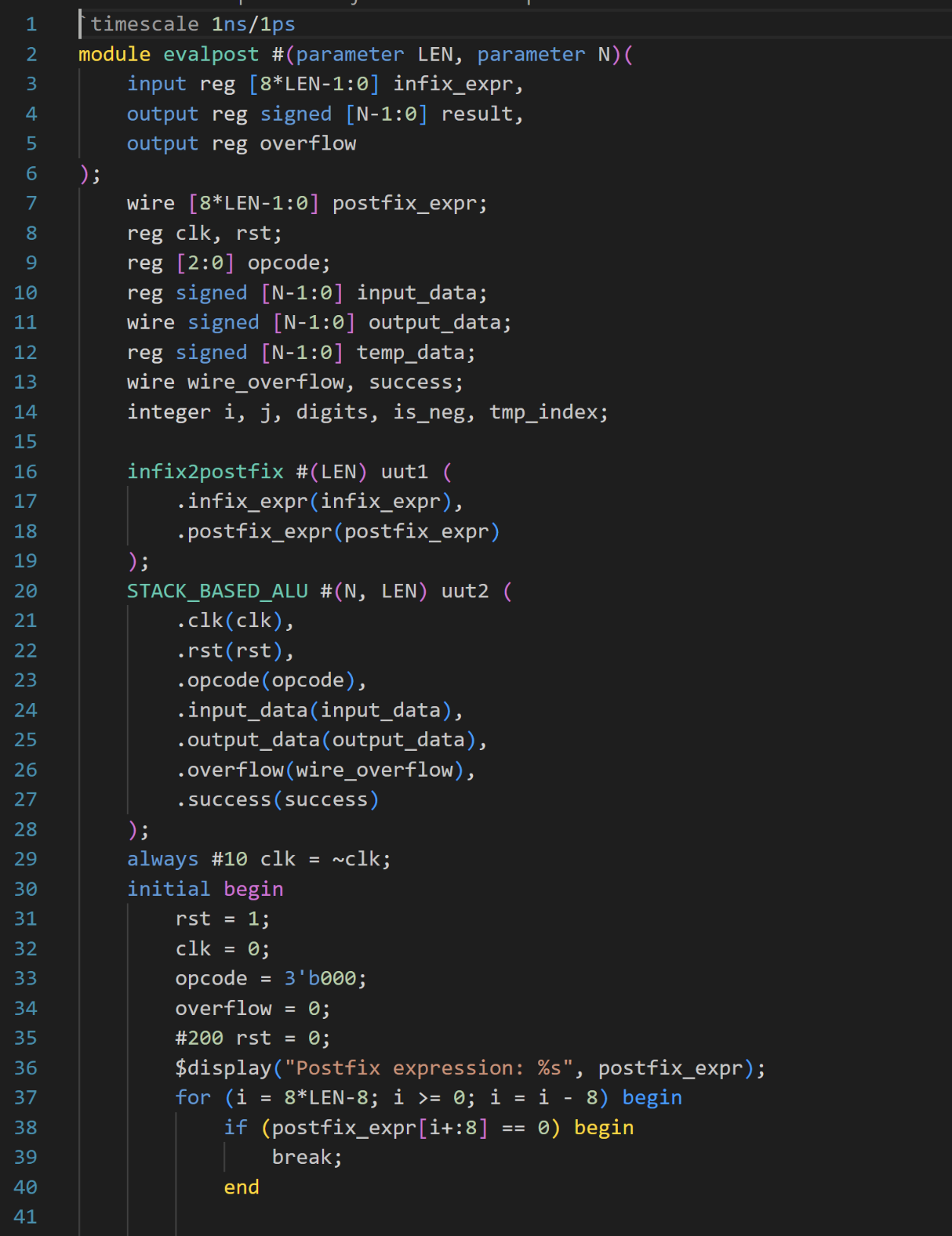
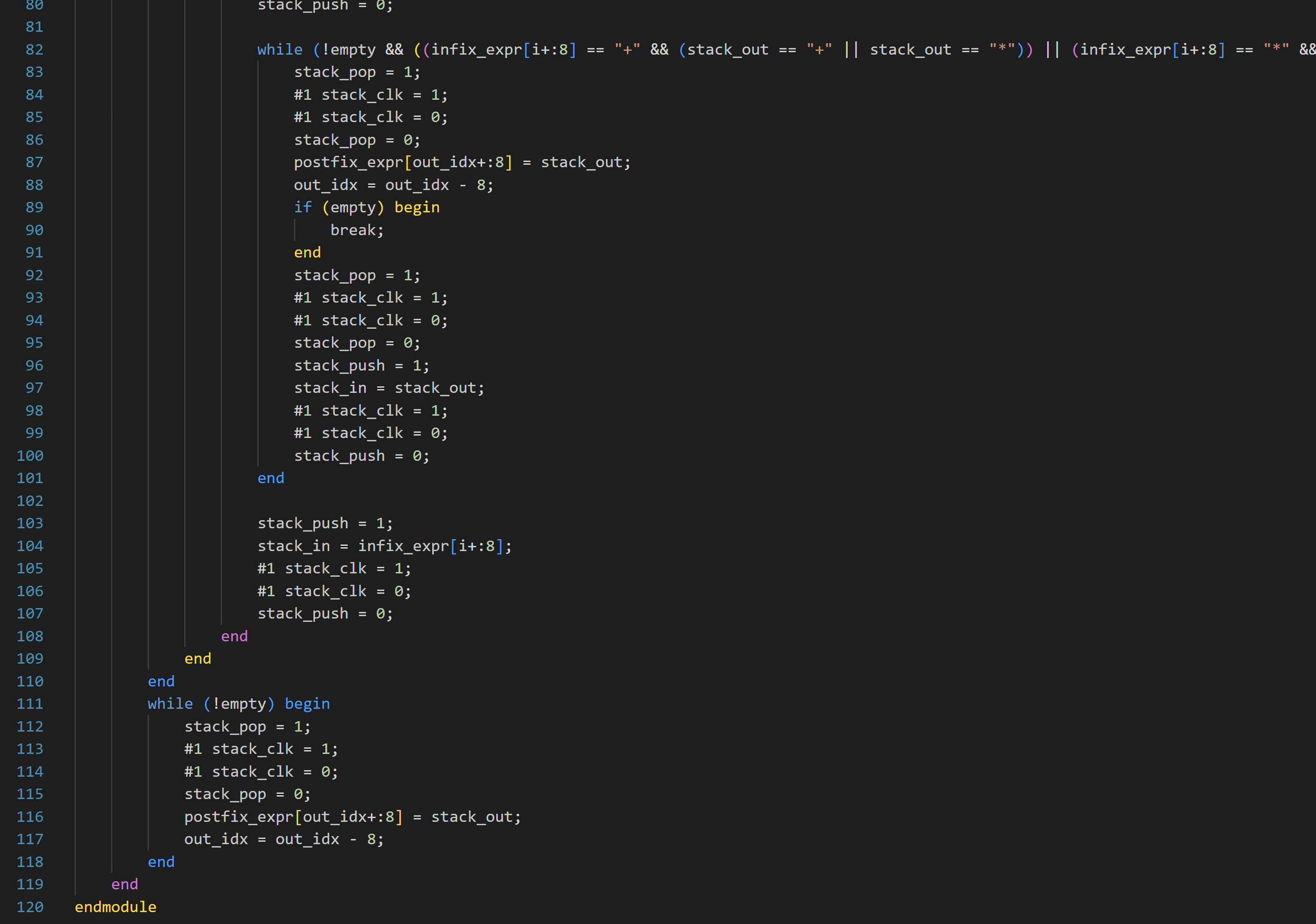
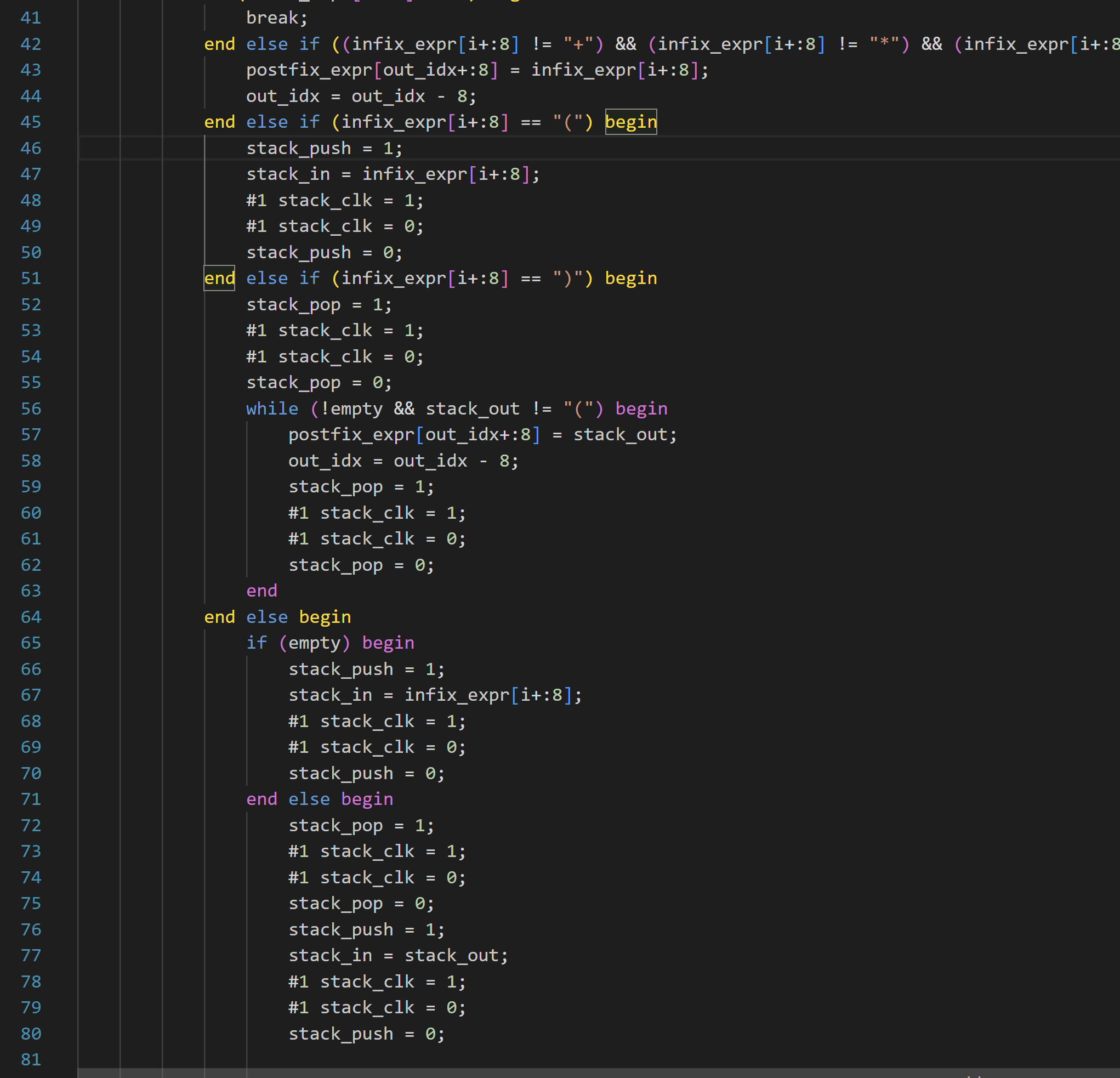
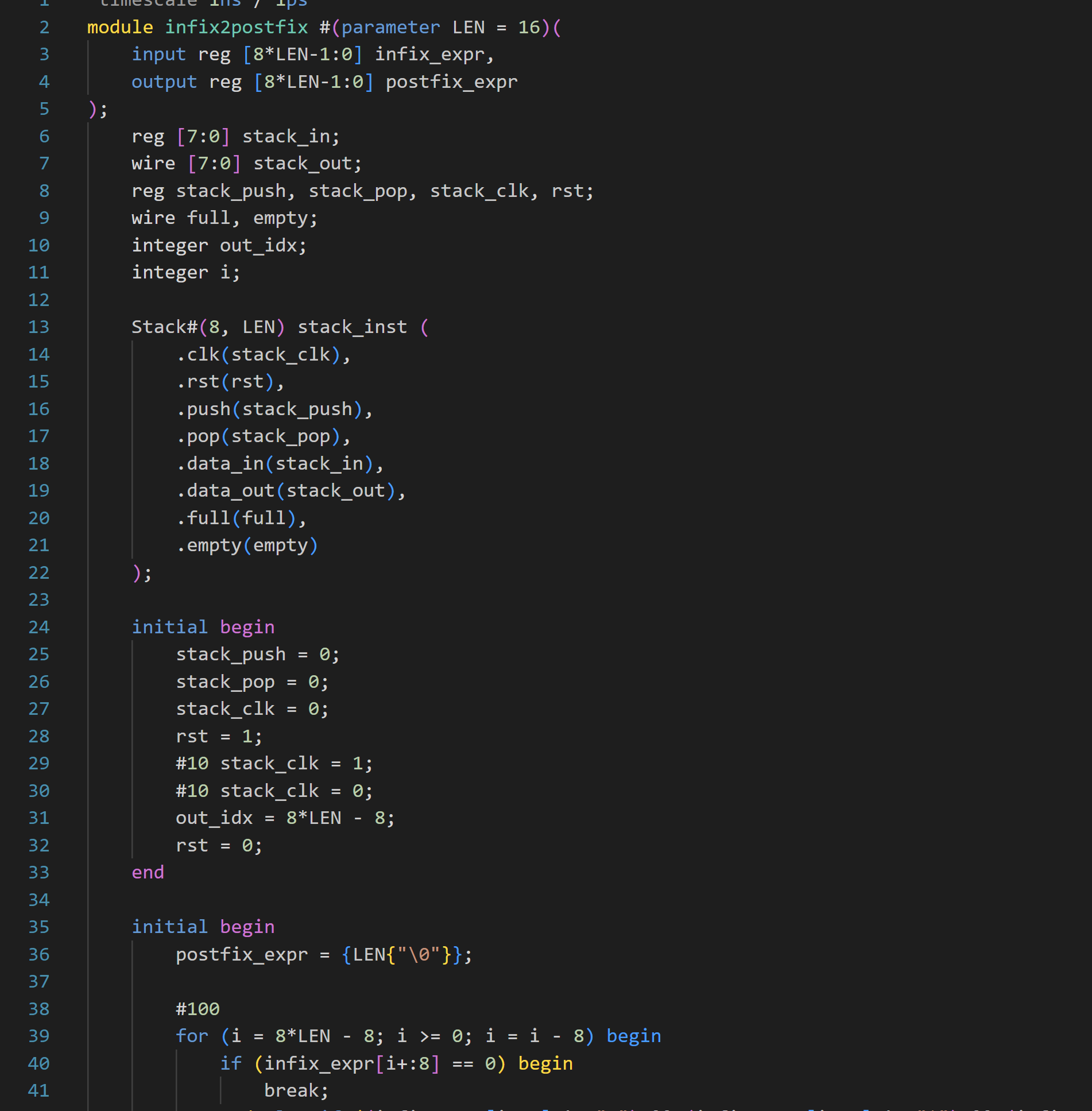
نتیجه دیتای 4 بیتی که 3 بار تکرار شده بصورت زیر است:

## ب)

ماژول infix2postfix: در بخش اولیه، سیگنال‌های مربوط به استک مقداردهی اولیه می‌شوند و استک ریست می‌شود. عملیات تبدیل عبارت اینفیکس به پست‌فیکس انجام می‌شود. یک حلقه از راست به چپ روی عبارت اینفیکس حرکت می‌کند و کاراکترها را بررسی می‌کند. اگر کاراکتر عددی باشد، به عبارت پست‌فیکس اضافه می‌شود. اگر کاراکتر پرانتز باز باشد، به پشته اضافه می‌شود. اگر کاراکتر پرانتز بسته باشد، کاراکترها از پشته خارج می‌شوند و به عبارت پست‌فیکس اضافه می‌شوند تا زمانی که به پرانتز باز برسیم. برای عملگرها، اگر استک خالی باشد، عملگر به استک اضافه می‌شود. در غیر این صورت، عملگرها از استک خارج شده و به عبارت پست‌فیکس اضافه می‌شوند تا زمانی که اولویت عملگر فعلی کمتر از عملگر درون استک باشد، سپس عملگر فعلی به استک اضافه می‌شود.

کد در صفحه بعد...



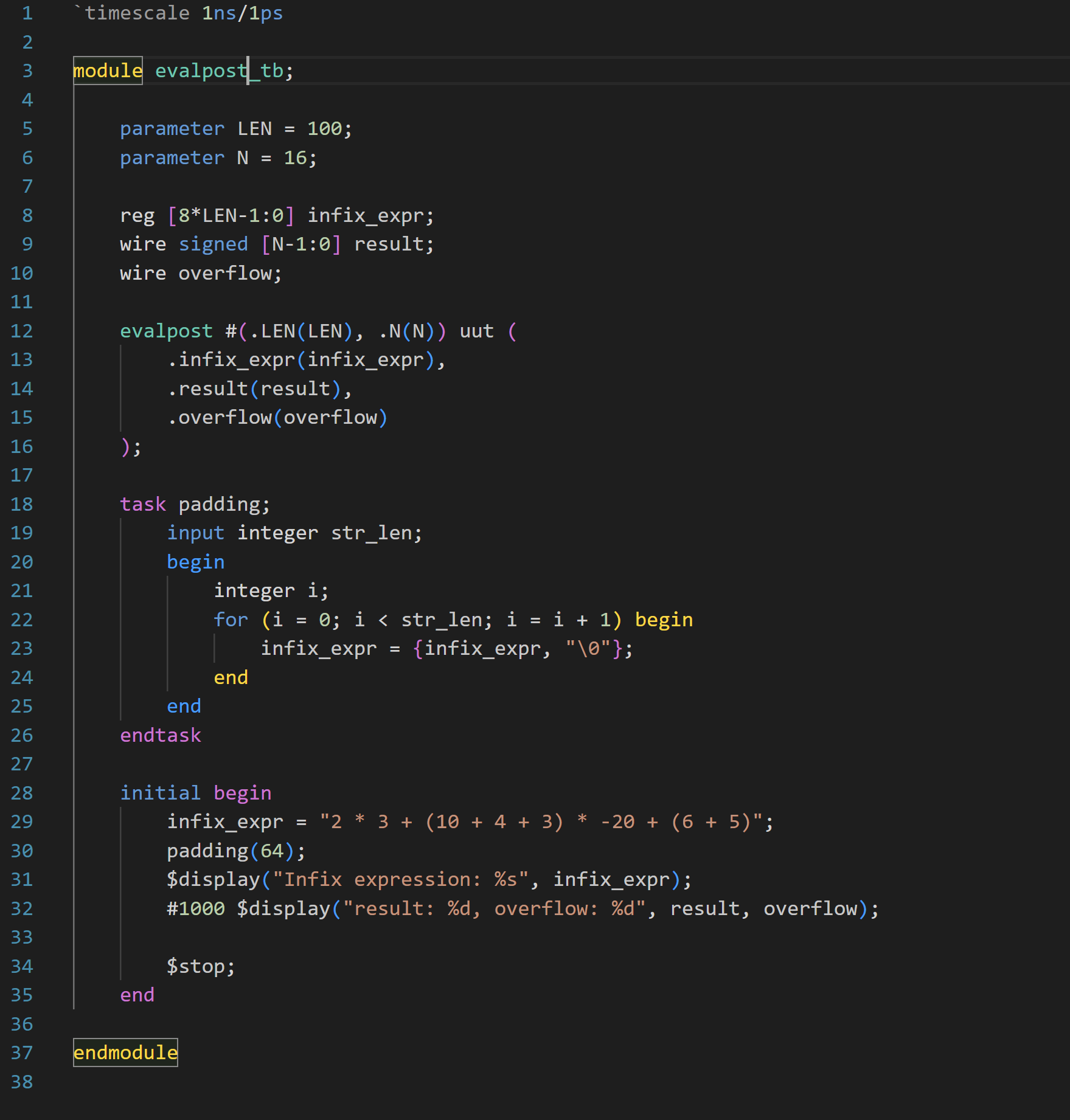


در ماژول evalpost: در ابتدای کد، سیگنال‌های مورد نیاز تعریف می‌شوند. این سیگنال‌ها شامل سیگنال‌های مربوط به ساعت (clk)، ریست (rst)، دستورالعمل (opcode)، داده ورودی (input\_data)، داده خروجی (output\_data)، و ... هستند.

در بخش اولیه، مقداردهی اولیه برای سیگنال‌های مورد نیاز انجام می‌شود. سیگنال ریست (rst) برای 200 واحد زمانی فعال می‌شود تا استک ریست شود. سپس، عبارت پست‌فیکس تولید شده نمایش داده می‌شود و یک حلقه برای پردازش هر کاراکتر از عبارت پست‌فیکس شروع می‌شود. اگر کاراکتر یک عملگر باشد (+ یا \*)، عملیات مربوطه با استفاده از ماژول ALU\_BASED\_STACK انجام می‌شود. در صورتی که عملگر + باشد، عملیات جمع (opcode = 3'b100) و در صورتی که \* باشد، عملیات ضرب (opcode = 3'b101) انجام می‌شود. اگر کاراکتر فضای خالی باشد، حلقه ادامه پیدا می‌کند. در غیر این صورت، کاراکتر به عنوان یک عدد پردازش می‌شود. ابتدا بررسی می‌شود که آیا عدد منفی است یا نه، سپس عدد خوانده شده و به داده ورودی (input\_data) اختصاص داده می‌شود.

در نهایت در ماژولevalpost\_tb ما یک عبارت ریاضی که در سوال داده شده را تست میکنیم و خروجی می دهیم.

عکس تست بنچ در صفحه بعد...



نتیجه بدست آمده:

