

طراحی نوع ساختمان داده‌های FIFO و Stack با استفاده از مدل - سازی رفتاری در VHDL

پارسا حجابی

چکیده

در این نوشتار گزارشی کامل از نحوه طراحی و پیاده‌سازی نوع ساختمان داده‌های FIFO همانند یک صف و Stack با استفاده از مدل‌سازی رفتاری یا Behavioral در زبان توصیف سخت‌افزار VHDL داده می‌شود و در نهایت نتیجه اجرا در قالب Waveها نشان داده می‌شود.

کلمات کلیدی

FIFO، ساختمان داده، Stack، VHDL

۱- مقدمه

در لبه بالارونده CLK و زمانی که EN برابر با یک باشد ابتدا خانه پانزدهم را روی خط خروجی DOut قرار می‌دهیم و سپس به وسیله یک FOR از صفر تا چهارده مقدار هر خانه آرایه‌مان را برابر با خلنه قبلی می‌گذاریم و در نهایت خلنه صفرم را از خط ورودی Din می‌گیریم. [۱]

برای نوشتن یک تست بنچ ۲ Process جدا یکی برای CLK و دیگری برای Process اصلی برنامه می‌نویسیم. Process کلاک حساس به خط کلاک است به دلیل اینکه اگر حساس نباشد کلاک فقط یکبار در کل زمان اجرای برنامه تغییر می‌کند. Process اصلی نیز در ابتدا RST مساوی یک شده تا همه خانه‌ها صفر شود. در ادامه یکبار مقادیر ۰ تا ۱۵ و بار دیگر مقادیر ۱۵ تا ۰ را در Din ریختیم و هر بار ۱۰ نانوثانیه صبر کردیم یعنی دقیقاً به اندازه زمان کلاک.

به علت اینکه نوع نمی‌توانیم int را در وکتور قرار دهیم این کار را به کمک تابع to_unsigned کردیم که ابتدا داده int را به unsigned تبدیل می‌کند و در خروجی یک vector تحویل می‌دهد. همچنین برای استفاده از این تابع مجبوریم تا ieee.numeric_std.all را در بالای برنامه استفاده کنیم. در نهایت یک WAIT گذاشتیم تا برنامه به درستی اجرا شود چون اگر نگذاریم Process به صورت دائم اجرا شده و برنامه از زمان صفر جلوتر نخواهد رفت. [۲]

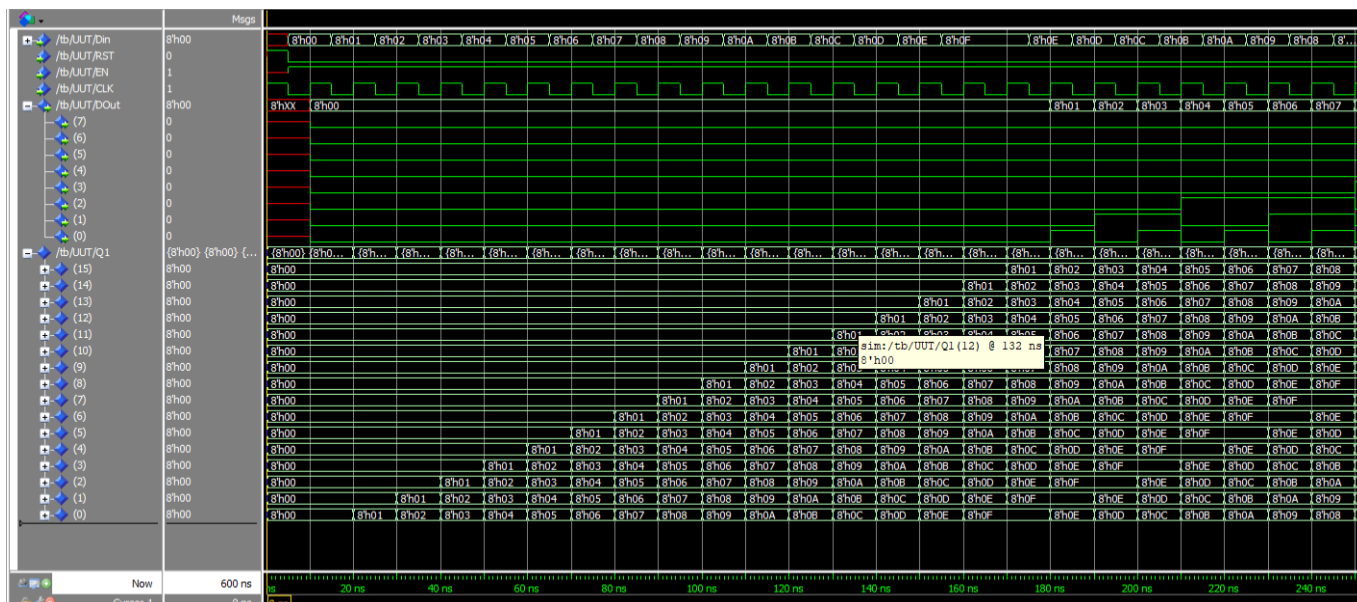
یک FIFO ساختمان داده‌ای است که دقیقاً همانند یک صف انسانی عمل می‌کند. داده‌ها از سمتی وارد آن شده و از سمت دیگر آن خارج می‌شوند و داده‌ای که زودتر از همه آمده زودتر از همه خارج می‌شود. Stack هم ساختمان داده‌ای می‌باشد که به نوعی برعکس حالت قبل عمل می‌کند یعنی داده‌ای که زودتر از همه به آن وارد می‌شود دیرتر از همه از آن خارج خواهد شد. این ساختمان داده‌ها اکثراً مفاهیمی نرم افزاری هستند و برای پیاده سازی آن‌ها معمولاً از مدل‌سازی رفتاری که یک مدل‌سازی سطح بالا در زبان VHDL می‌باشد استفاده می‌شود.

۲- مطالب اصلی

۲-۱- پیاده‌سازی FIFO

برای پیاده‌سازی این ساختمان داده از تایپ آرایه استفاده کردیم و یک آرایه از std_logic_vectorها به طول ۱۶ که هر کدام ۸ خانه دارند استفاده کردیم و نام آن را Q گذاشتیم و از آن یک نمونه به نام Q1 ایجاد کردیم.

سپس Process آن را نوشتیم که باید به CLK و RST چون این ریست آسنکرون می‌باشد حساس باشد. در همان ابتدا چک می‌کنیم که اگر RST یک باشد به وسیله یک FOR تمام خانه‌های آرایه Q1 مان را صفر کنیم. و در غیر این صورت فقط



تصویر شماره (۱): اجرای برنامه صف با طول ۶۰۰ نانوثانیه

int تبدیل کنیم. برای این کار از تابع conv_integer کمک می‌گیریم که برای استفاده آن باید std_logic_arith و std_logic_unsigned را در بالای برنامه استفاده کنیم.

ابتدا خنده‌ی SPTEMP آرایه‌مان را مساوی خط ورودی قرار می‌دهیم سپس چک می‌کنیم که اگر به ۷ رسیده باشد سیگنال FUL را برابر با ۱ قرار دهیم و در غیر این صورت آن را برابر با ۰ قرار دهیم. و در pop هم دقیقاً برعکس عملیات را انجام می‌دهیم.

برای تست آن نیز همان طور که کلاک را در FIFO پیاده کردیم اینجا هم پیاده می‌کنیم و ۸ عدد صفر تا ۷ را درون آن می‌ریزیم.

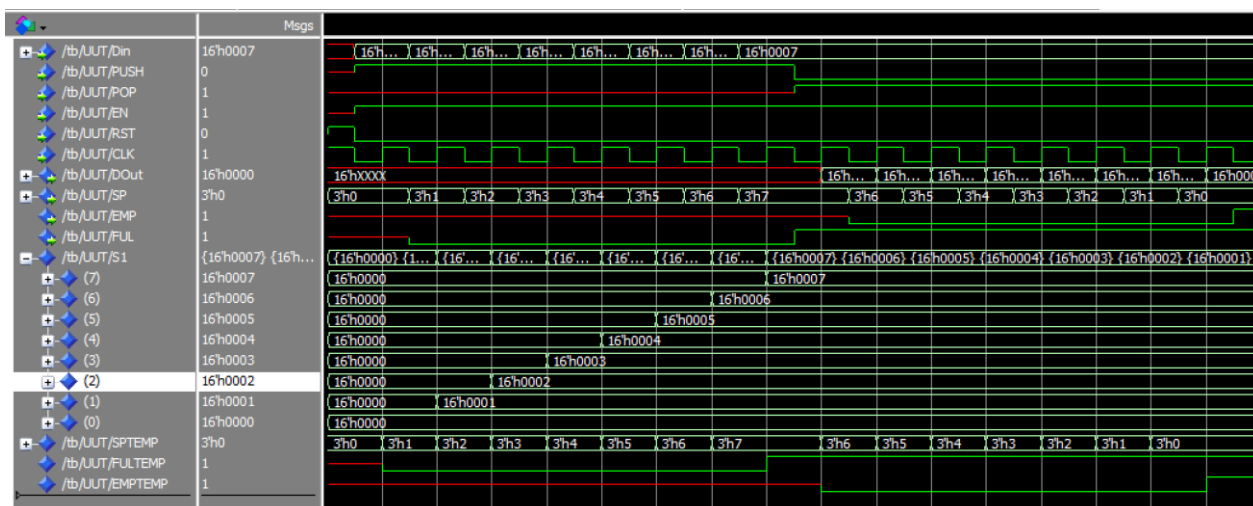
تصویر شماره (۱): ویو مربوط به ساختمان داده استک که ۶۰۰ نانوثانیه اجرا شده.

۲-۲- پیاده‌سازی STACK

برای پیاده‌سازی این ساختمان داده هم از آرایه استفاده کردیم و چون به مقادیر پورتهایی همانند SPT و FUL و EMP نیاز داریم اما همه از نوع OUT هستند مجبور هستیم سیگنال‌هایی درونی تعریف کنیم و در انتهای Process مقادیر تغییر یافته آن‌ها را در پورتهای اصلی بریزیم. [۳]

برای برقرار کردن شرط‌های گفته شده در صورت سوال از ۳ if استفاده کردیم. اولین شرط برای زمانی است که هم خط push و هم خط pop یک باشند اما ما در ابتدا باید عملیات pop را انجام دهیم و سپس عملیات push.

If بعدی برای زمانی است که stack ما پر نشده باشد و خط push آن فعال باشد. در این حالت و در حالتی که می‌خواهیم pop کنیم نیاز است تا از SPTEMP به عنوان index آرایه‌مان استفاده کنیم اما به دلیل اینکه نوع داده vector نمی‌تواند به عنوان index مورد استفاده قرار بگیرد باید به نحوی آن را به



۳- نتیجه

در سخت‌افزار هم همانند نرم‌افزار برای پیاده‌سازی اکثر ساختمان‌های داده از آرایه‌ها استفاده می‌کنیم با این تفاوت که این کدها به دلیل سطح بالا بودن به سختی سنتز خواهند شد.

مراجع

- [1] fifo.vhd file lines: 14 to 29
- [2] fifo_tb.vhd file lines: 33 to 52
- [3] stack.vhd file lines: 79 to 81