

فربد فولادی-98243045

عرفان رفیعی اسکویی-98243027

گزارش تمرین سری 4

در ضرب عدد اول در عدد دوم به تعداد بیت های دو عدد ما جمع عدد N بیتی داریم یعنی در اصل جمع N عدد N بیتی.

برای توضیح این گزارش فرض می شود که اعداد ما 4 بیتی هستند.

در multiplier entity با دانش قبلی که از adder ها داشتیم محاسبات جمع ضربکننده خود را پیاده سازی کردیم.

در ابتدا به علت این که ضرب دو عدد N بیتی داریم و تعداد مرحله های عملیات ما در ضرب N می باشد ما یک ارایه $N \times N$ بیتی برای سیگنال های IN و OUT خود در نظر میگیریم.

برای این که در هر مرحله ی ضرب که از بیت اول عدد دوم آغاز شده و تا بیت آخر ان ادامه می یابد حاصل هر جمع را نگه داریم باید از دو متغیر کمکی prodin و prodout کمک بگیریم که prodin بیت های carryout را یک عدد به سمت چپ شیفت داده و بیت سمت راست را 0 قرار می دهیم به علت ان که یک مرحله در ضرب جلو رفتیم وظیفه prodout این است که در هر ردیف ان جمع هر مرحله نگهداری می شود به عنوان مثال ردیف شماره دوم ارایه pout بیان کر این است که حاصل جمع ضرب بیت اول و بیت دوم عدد دوم در عدد اول چه عددی را حاصل می کند البته یادآور می شود در هر ردیف به مقدار عدد ان ردیف شیفت به چپ داریم. یعنی ردیف دوم prodout که عدد ان 1 می باشد بیت 5 تا 1 حاصل جمع ضرب دو بیت عدد دوم در عدد اول را به ما نشان می دهد.

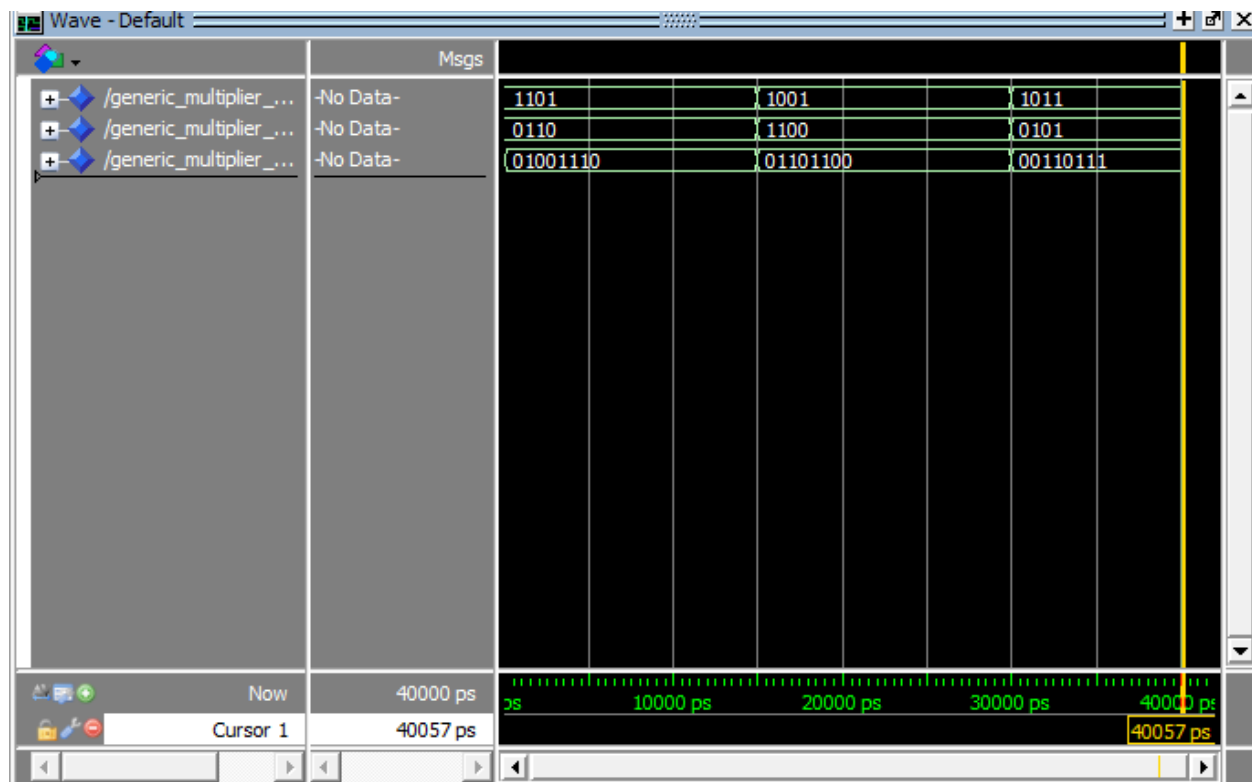
در ادامه cin داریم که 3 بیت آخر مرحله قبل prodout می باشند که یک شیفت به سمت راست داده شده اند و بیت چهارم ان ها سه ستون اول سه ردیف مرحله قبل cout می باشد که در اصل بیانگر این است که cin در اصل همان cout است که در مرحله قبل محاسبه شده است.

حال یک خروجی با عنوان answer که $N+N$ بیتی می باشد که در اصل بیانگر ان است که حداکثر تعداد بیت های خروجی answer ما همان عدد $2N$ است می باشد.

حال برای محاسبه ی خروجی می دانیم که ستون شماره 0 هر ردیف ارایه prodout بیانگر این است که بیت آخر خروجی جمع هر مرحله چه می باشد یادآور می شویم که در هر ردیف به اندازه شماره ان ردیف -1 شیفت به چپ داریم پس در نتیجه خروجی ستون صفر از بالا تا ردیف یکی مانده به آخر بیانگر 3 بیت آخر جواب ما است و ردیف آخر بیانگر بیت های 4 تا 7 جواب نهایی ماست و چون می دانیم جواب نهایی حداکثر 8 بیتی می باشد برای بیت آخر مقدار ستون و ردیف آخر ارایه ی cout که همان

cout آخرین جمع ما می باشد را قرار می دهیم که با توجه به مقدار آن بیت هشتم جواب نهایی می تواند دو مقدار 0 یا 1 را دارا باشد.

در ادامه تصاویری از خروجی testbench و توضیحات بالا در قالب مثال آورده شده است.



	Msgs
+ /generic_multiplier/A	1101
+ /generic_multiplier/B	0110
+ /generic_multiplier/answer	01001110
- /generic_multiplier/Cin	{0000} {0000} {0110} {1001}
+ (0)	0000
+ (1)	0000
+ (2)	0110
+ (3)	1001
- /generic_multiplier/Cout	{0000} {0000} {1100} {0000}
+ (0)	0000
+ (1)	0000
+ (2)	1100
+ (3)	0000
- /generic_multiplier/Prod_in	{0000} {0000} {1000} {0000}
+ (0)	0000
+ (1)	0000
+ (2)	1000
+ (3)	0000
- /generic_multiplier/Prod_out	{0000} {1101} {0011} {1001}
+ (0)	0000
+ (1)	1101
+ (2)	0011
+ (3)	1001

+ /generic_multiplier/A	1001
+ /generic_multiplier/B	0101
+ /generic_multiplier/answer	00101101
- /generic_multiplier/Cin	{0000} {0100} {0010} {0101}
+ (0)	0000
+ (1)	0100
+ (2)	0010
+ (3)	0101
- /generic_multiplier/Cout	{0000} {0000} {0000} {0000}
+ (0)	0000
+ (1)	0000
+ (2)	0000
+ (3)	0000
- /generic_multiplier/Prod_in	{0000} {0000} {0000} {0000}
+ (0)	0000
+ (1)	0000
+ (2)	0000
+ (3)	0000
- /generic_multiplier/Prod_out	{1001} {0100} {1011} {0101}
+ (0)	1001
+ (1)	0100
+ (2)	1011
+ (3)	0101