

گزارش دستورکار هفتم آزمایشگاه معماری کامپیوتر

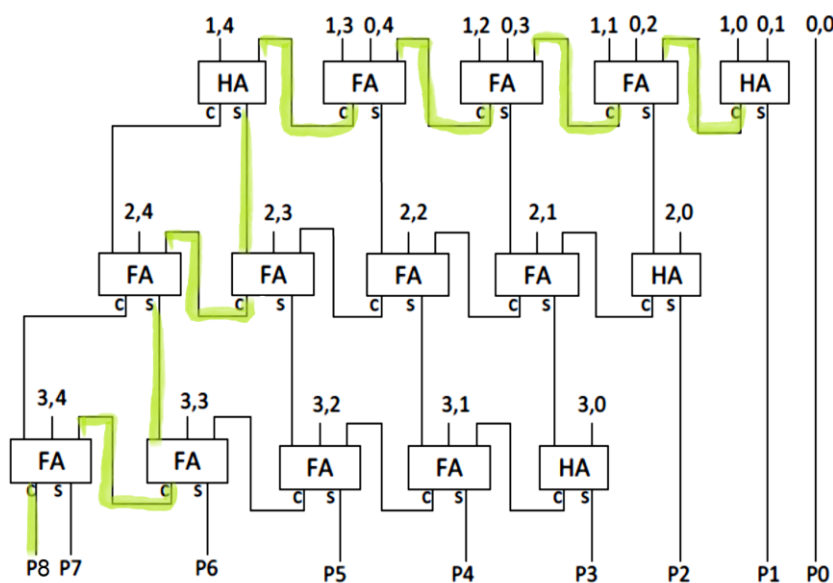
نگار موقتیان، ۹۸۳۱۰۶۲

ماژول simple_multiplier

در این قسمت از آزمایش می‌خواهیم یک ضرب کننده معمولی ۴ در ۵ بیتی طراحی کنیم. ابتدا بررسی می‌کنیم پاسخ خروجی چند بیتی خواهد بود. این محاسبات برای تمام ضرب کننده‌ها در آزمایش‌های بعدی نیز معتبر است و تعداد بیت‌های خروجی همین مقدار خواهد بود. برای این کار بزرگ‌ترین عدد ۴ بیتی را در بزرگ‌ترین عدد ۵ بیتی ضرب می‌کنیم تا حداکثر حاصل ضرب را بیابیم.

$$(1111)_2 \times (1\ 1111)_2 = 15 \times 31 = 465 = (1\ 1101\ 0001)_2$$

بنابراین حاصل حداکثر ۹ بیتی خواهد بود. مدار این ضرب کننده ۴ در ۵ به صورت زیر می‌باشد.



همانطور که در این شکل دیده می‌شود نیاز به ۴ نیم جمع کننده و ۱۱ تمام جمع کننده داریم که در سه طبقه (۱ - ۴) قرار دارند. در این ضرب کننده مانند ضرب دهدهی هر بیت از عدد ۴ بیتی A را در بیت‌های عدد ۵ بیتی B ضرب کرده و با یک شیفت با حاصل مرحله بعد جمع می‌زنیم.

بنابر توضیحات گفته شده هزینه ساخت این ضرب کننده برابر است با:

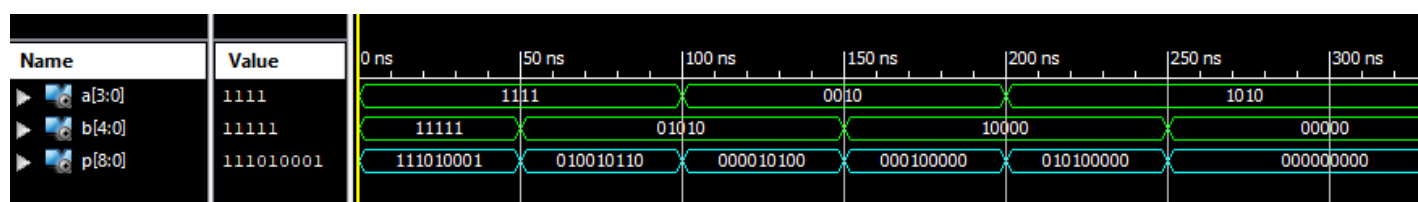
$$\text{Cost} = 20 \text{ AND} + 4 \text{ HA} + 11 \text{ FA} = 20g + 4(2g) + 11(5g) = 83g$$

همچنین تاخیر این مدار، تاخیر مسیر بحرانی آن است که در شکل با رنگ سبز مشخص شده است و برابر است با:

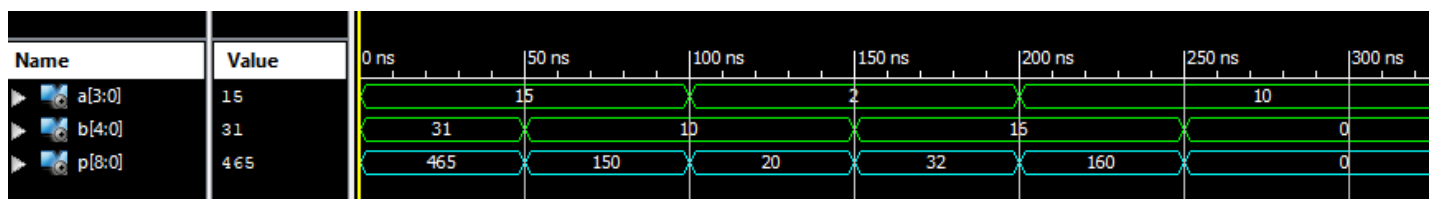
$$\text{Delay} = \text{delay AND} + \text{delay(carry) HA} + 6 \text{ delay(carry) FA} + \text{delay(sum) HA} + \text{delay(sum)}$$

$$\text{FA} = d + d + 6(2d) + d + d = 16d$$

پس از پیاده سازی مدار شبیه سازی آن توسط test bench نوشته شده و به ازای مقادیر مختلف A و B انجام شد، تا از درستی رفتار مدار ساخته شده اطمینان حاصل شود.



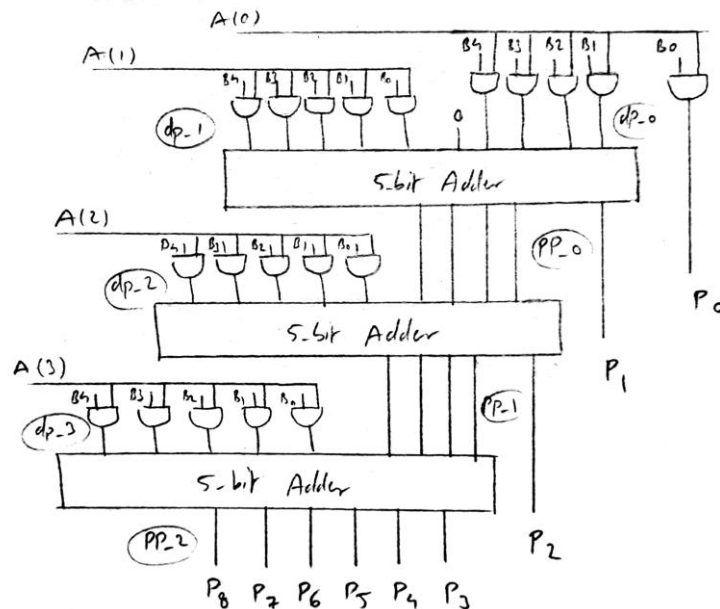
برای سهولت بیشتر در بررسی سیگنال های ورودی و خروجی تمام اعداد چند بیتی را با فرض بی علامت بودن به مبنای ۱۰ می بریم. نتیجه تنظیمات گفته شده به صورت زیر است.



همانطور که مشاهده می شود تمامی حاصل ضرب ها به درستی انجام شده اند.

ماژول array_multiplier

حال در این قسمت از آزمایش می‌خواهیم یک ضرب کننده آرایه‌ای ۴ در ۵ بیتی طراحی کنیم. همانطور که در قسمت قبل توضیح دادیم خروجی این ضرب کننده باید ۹ بیتی باشد. مدار این ضرب کننده ۴ در ۵ به صورت زیر می‌باشد.



این ضرب کننده درست مانند ضرب کننده قبلی عمل می‌کند، با این تفاوت که به جای استفاده از تمام جمع‌کننده‌ها و نیم جمع‌کننده‌ها برای جمع هر مرحله با مرحله قبل از یک جمع کننده ۵ بیتی مانند یک جمع کننده آبشاری استفاده می‌کند.

هزینه ساخت این جمع کننده به سادگی برابر است با:

$$\text{Cost} = 20 \text{ AND} + 3 \text{ RA:5-bit} = 20g + 3(25g) = 85g$$

همچنین هزینه این مدار را می‌توان به دو صورت تفسیر کرد. زمانی که هر جمع کننده آبشاری را به صورت یک Black Box بینیم داریم:

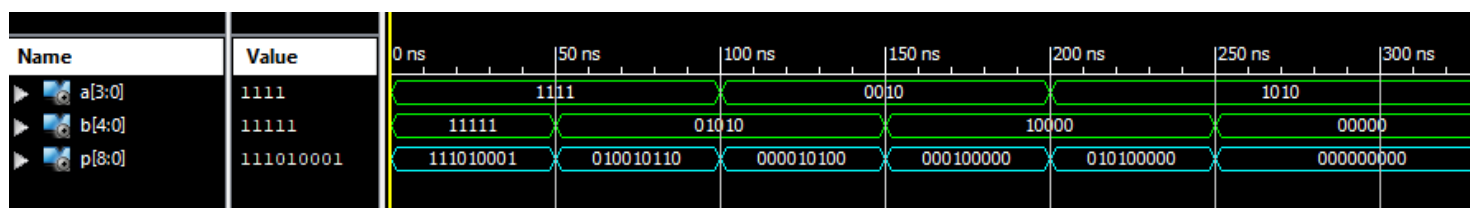
$$\text{Delay} = \text{delay AND} + 3(\text{delay (carry) RA}) = d + 3(10d) = 31d$$

اما در عمل چنین نیست و محاسبات بیت‌های کم ارزش‌تر می‌توانند سریع‌تر حاضر شوند. اگر نگاه جزئی‌تری داشته باشیم و هر جمع کننده آبشاری را به صورت مجموعه‌ای از تمام جمع کننده‌ها بینیم داریم:

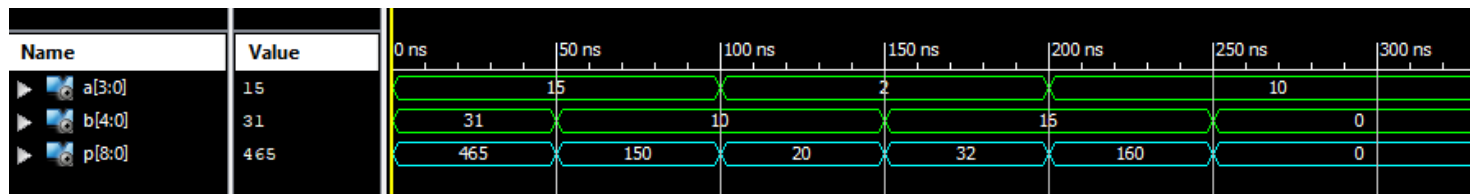
$$\text{Delay} = \text{delay AND} + 7 \text{ delay(carry) FA} + 2 \text{ delay(sum) HA} = d + 7(2d) + 2d = 17d$$

لذا هزینه و تاخیر این جمع کننده از جمع کننده معمولی کمی بیش‌تر است (به دلیل اینکه به جای HA های موجود در جمع کننده معمولی از FA استفاده می‌شود)، اما به دلیل اینکه جمع کننده‌ها به صورت آماده موجود هستند پیاده‌سازی این مدار از مدار قبلی ساده‌تر است.

پس از پیاده سازی مدار شبیه سازی آن توسط test bench نوشته شده و به ازای مقادیر مختلف A و B انجام شد، تا از درستی رفتار مدار ساخته شده اطمینان حاصل شود.



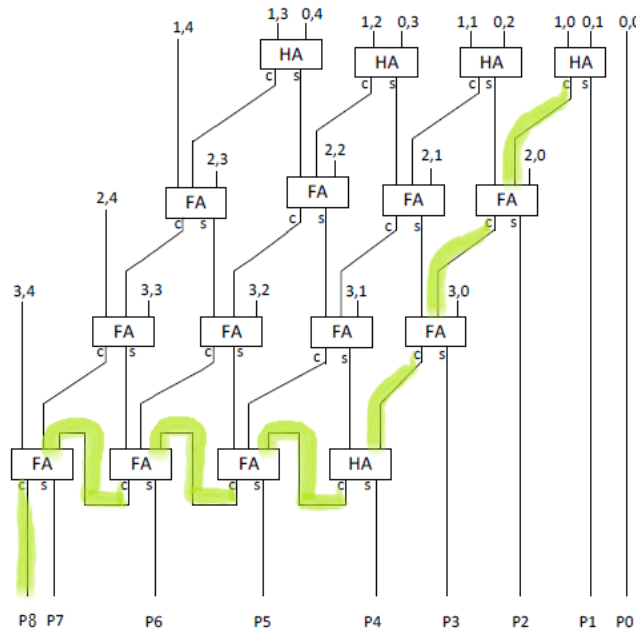
برای سهولت بیش تر در بررسی سیگنال های ورودی و خروجی تمام اعداد چند بیتی را با فرض بی علامت بودن به مبنای ۱۰ می بریم. نتیجه تنظیمات گفته شده به صورت زیر است.



همانطور که مشاهده می شود تمامی حاصل ضرب ها به درستی انجام شده اند.

ماژول save_adder_multiplier

در نهایت می‌خواهیم یک ضرب کننده ۴ در ۵ بیتی با استفاده از جمع کننده ذخیره‌گر رقم نقلی طراحی کنیم. همانطور که در قسمت قبل توضیح دادیم خروجی این ضرب کننده باید ۹ بیتی باشد. مدار این ضرب کننده ۴ در ۵ به صورت زیر می‌باشد.



روش کار این ضرب کننده کمی متفاوت با ضرب کننده‌های قبلی و مبتنی بر روش ذخیره رقم نقلی است.

هزینه ساخت این جمع کننده به سادگی برابر است با:

$$\text{Cost} = 20 \text{ AND} + 4 \text{ HA} + 11 \text{ FA} = 20g + 4(2g) + 11(5g) = 83g$$

که با هزینه ضرب کننده اول برابر است.

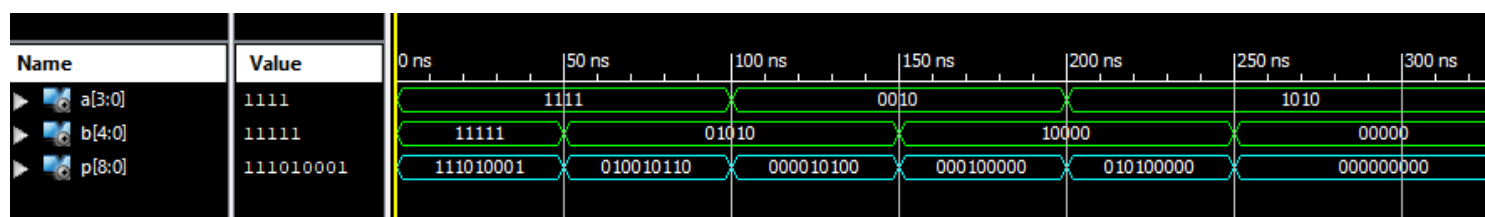
همچنین تاخیر این مدار، تاخیر مسیر بحرانی آن است که در شکل با رنگ سبز مشخص شده است و برابر است با:

$$\text{Delay} = \text{delay AND} + 2 \text{ delay(carry) HA} + 5 \text{ delay(carry) FA} = d + 2d + 5(2d) = 13d$$

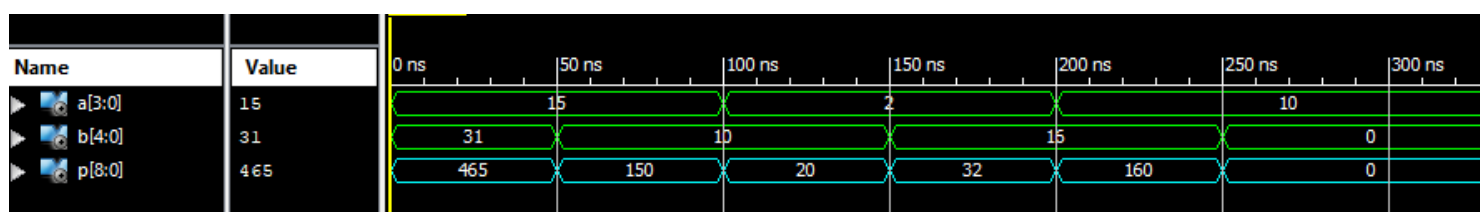
که در مقایسه با ضرب کننده‌های قبلی تاخیر بسیار خوبی است.

در کل نیز به نظر می‌رسد این ضرب کننده بهترین performance per cost میان این چند ضرب کننده را داراست، زیرا کمترین هزینه و کمترین تاخیر را دارد.

پس از پیاده سازی مدار شبیه سازی آن توسط test bench نوشته شده و به ازای مقادیر مختلف A و B انجام شد، تا از درستی رفتار مدار ساخته شده اطمینان حاصل شود.



برای سهولت بیش تر در بررسی سیگنال های ورودی و خروجی تمام اعداد چند بیتی را با فرض بی علامت بودن به مبنای ۱۰ می بریم. نتیجه تنظیمات گفته شده به صورت زیر است.



همانطور که مشاهده می شود تمامی حاصل ضرب ها به درستی انجام شده اند.

در نهایت برای جمع بندی می‌توان جدول زیر را ارائه داد.

	ضرب کننده عادی	ضرب کننده آرایه‌ای	ضرب کننده ذخیره‌گر نقلی
هزینه	83g	85g	83g
تاخیر	16d	17d	13d
روش کار	مانند ضرب دهمی هر رقم عدد اول در ارقام عدد دوم ضرب شده و در هر مرحله با یک شیفت با مرحله بعد جمع می‌شود	همانند ضرب کننده عادی	مانند جمع کننده ذخیره‌گر نقلی و بر خلاف جمع کننده آبشاری، رقم نقلی هر مرحله را نگه داشته و مانند یک عدد آن را با اعداد اصلی جمع می‌زند
مزایا	هزینه و تاخیر نسبتاً خوب	به دلیل استفاده از جمع کننده‌های آماده پیاده سازی نسبتاً ساده‌ای دارد	سریع‌ترین و کم هزینه‌ترین ضرب کننده میان این سه است
معایب	نسبت به ضرب کننده آرایه‌ای پیاده سازی دشوارتری دارد و نسبت به ضرب کننده ذخیره‌گر نقلی کندتر است	نسبت به دیگر ضرب کننده‌ها کند است و هزینه بیشتری دارد	نسبت به ضرب کننده آرایه‌ای پیاده سازی دشوارتری دارد