



به نام خدا

دانشگاه تهران
دانشکده فنی
دانشکده مهندسی برق و کامپیوتر

مدار های منطقی و سیستم های دیجیتال

گزارش تمرین کامپیوتری دوم
CA 2

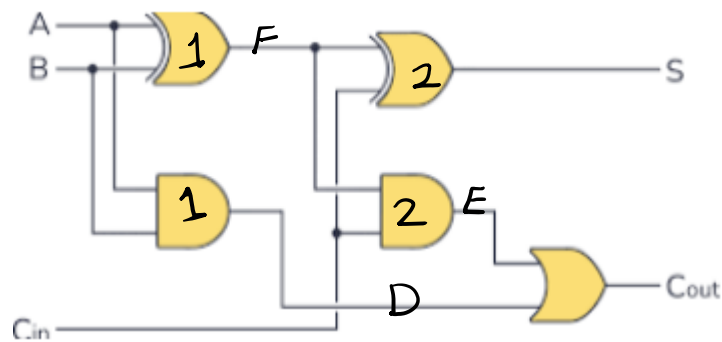
نام و نام خانوادگی:
نیلوفر مرتضوی

شماره دانشجویی:
220701096

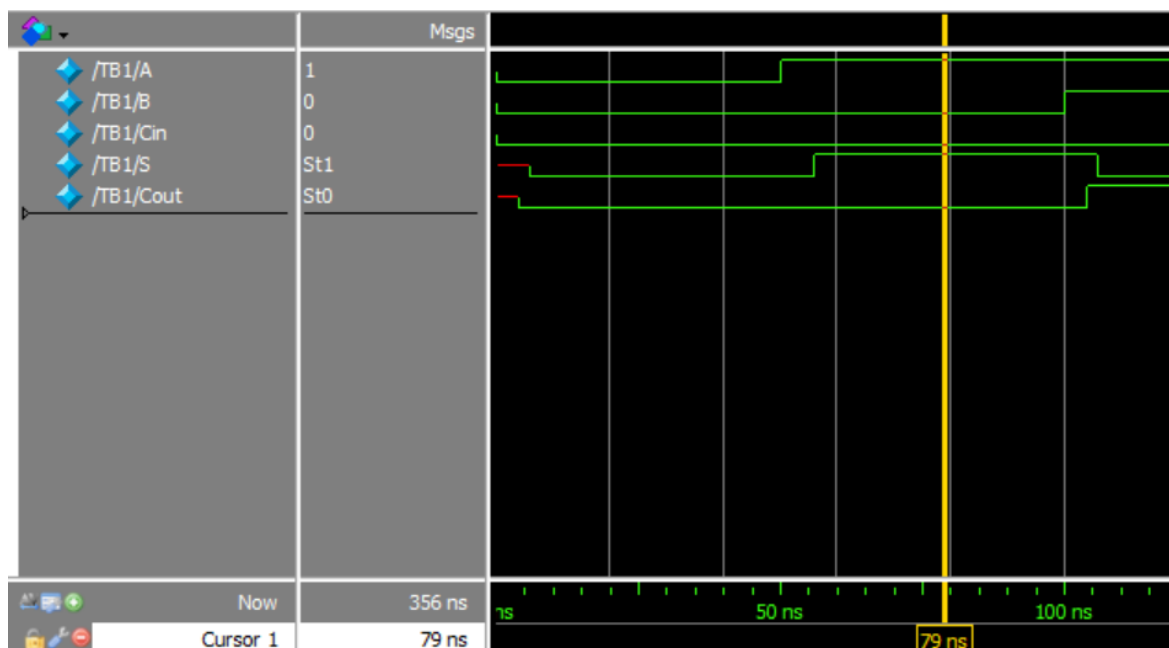
*از سه روز گریس برای
تحويل این تمرین استفاده شده

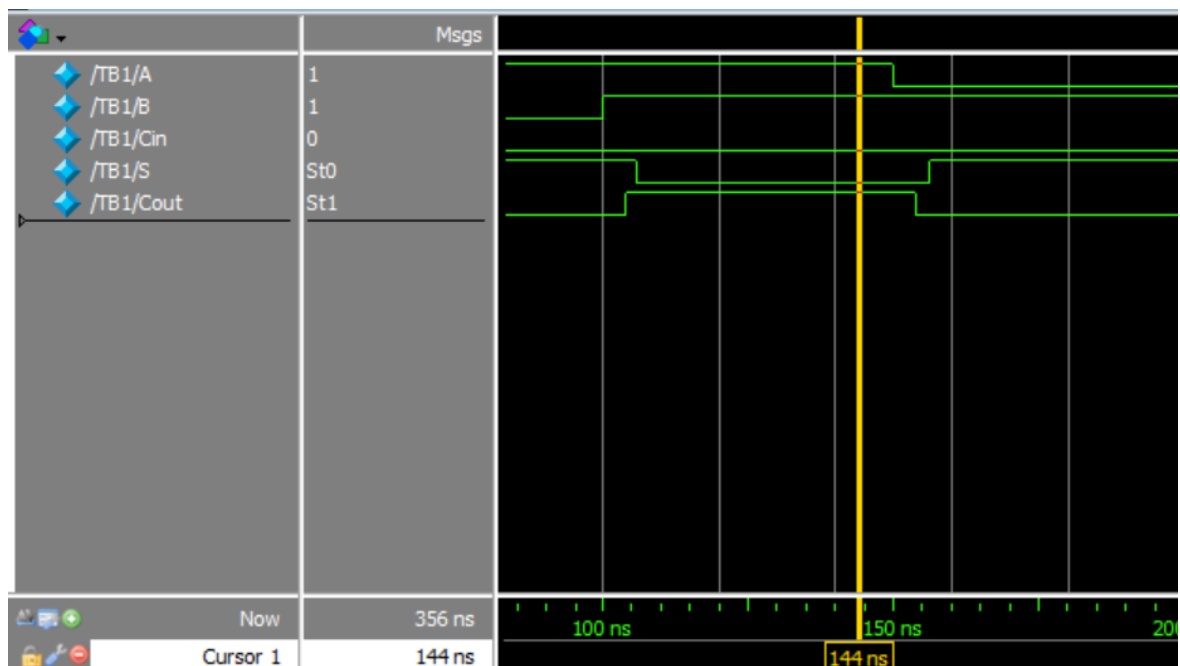
Step 1

طبق شکل مدار داده شده کد وریلاگ را با گیت های داده شده نوشتم (فایل وریلاگ 1) و برای آزمایش درستی fulladder نوشته شده یک تست بنچ برای آن طراحی کردم (فایل وریلاگ TB1).

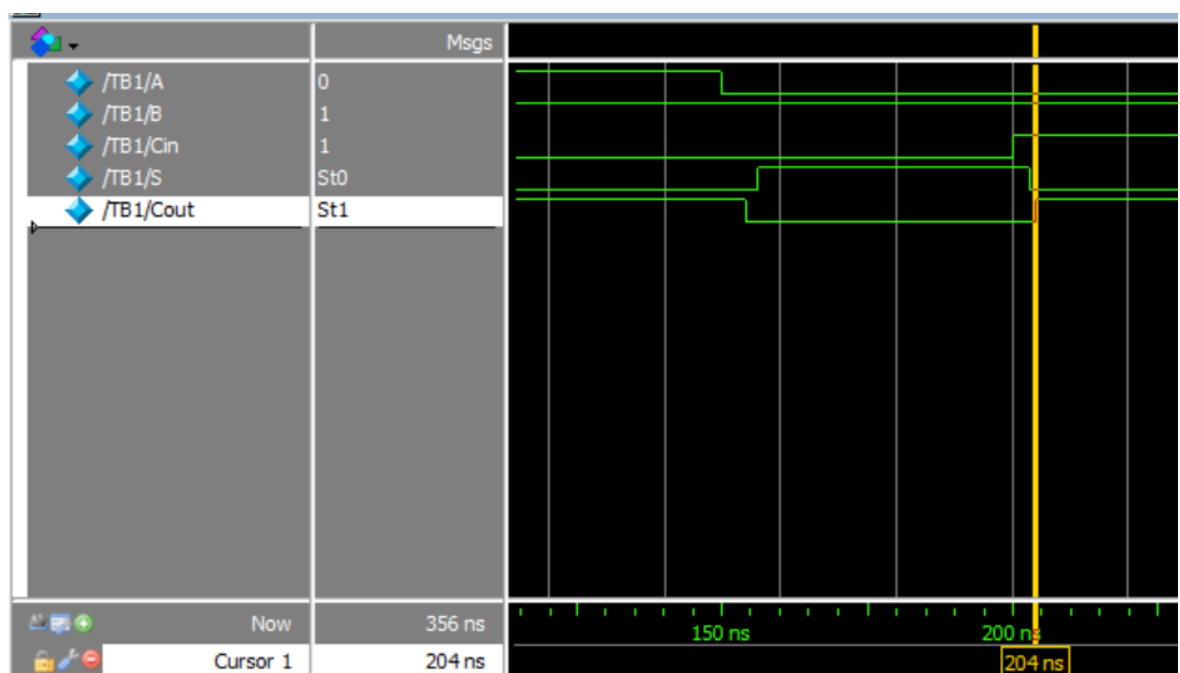


اگر waveform بدست آمده را بررسی کنیم میبینیم که این ماژول به درستی کار می کند:





میبینیم خروجی مدار با تاخیر کمی از عوض شدن مقادیر، تغییر می کند!
 دلیل این تاخیر پیش آمده به دلیل وجود تاخیر گیت ها است که خروجی Cout باید از سه گیت
 متفاوت (worth path) باید عبور کند.



Step 2

در این قسمت با چهار بار صدا زدن ماژول fulladder تک بیتی، یک ماژول 4 ripple carry adder نوشتم (فایل ورپلاگ 2) و برای آزمایش درستی آن تست پنجم نوشتم (فایل ورپلاگ TB2).

قسمتی از waveform بدست آمده برای این قسمت به شکل زیر است که درستی کارکرد ماژول را نشان می دهد:



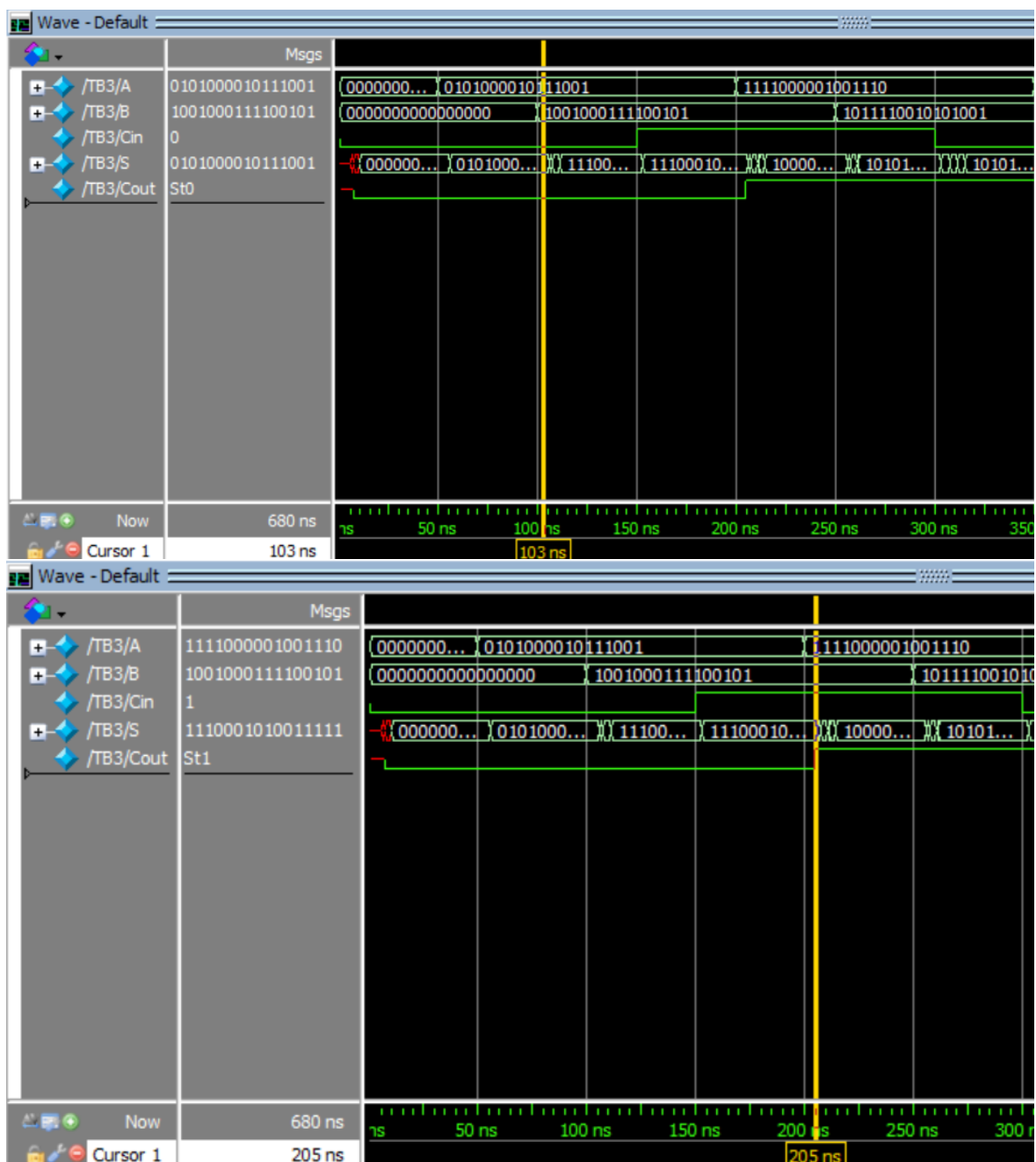


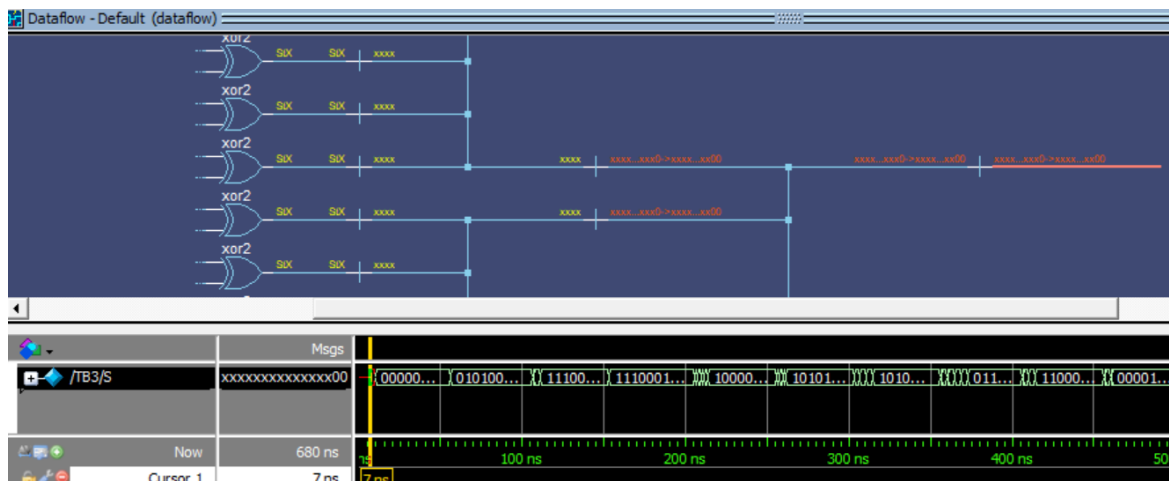
همچنین در دو شکل بالا مشاهده می شود در ابتدای waveform مقادیر مبهم داریم که به دلیل تاخیر گیت ها به وجود می آیند.

Step 3

در این گام با کنار هم صدا کردن 4 بار ماژول ripple carry adder 4 در یک ماژول، یک ماژول برای 16 ripple carry adder نوشتم (فایل وریلاگ 3) و برای تست درستی آن یک تست بنچ طراحی کردم (فایل وریلاگ TB3).

قسمتی از waveform بدست آمده برای این قسمت به شکل زیر است که درستی کارکرد ماژول را نشان می دهد:





همچنین مانند گام قبل، تاخیری که در ابتدای dataflow مشاهده میشود به دلیل همان تاخیر گیت هایمان می باشد.

اگر بخواهیم تاخیر را به صورت دستی محاسبه کنیم:

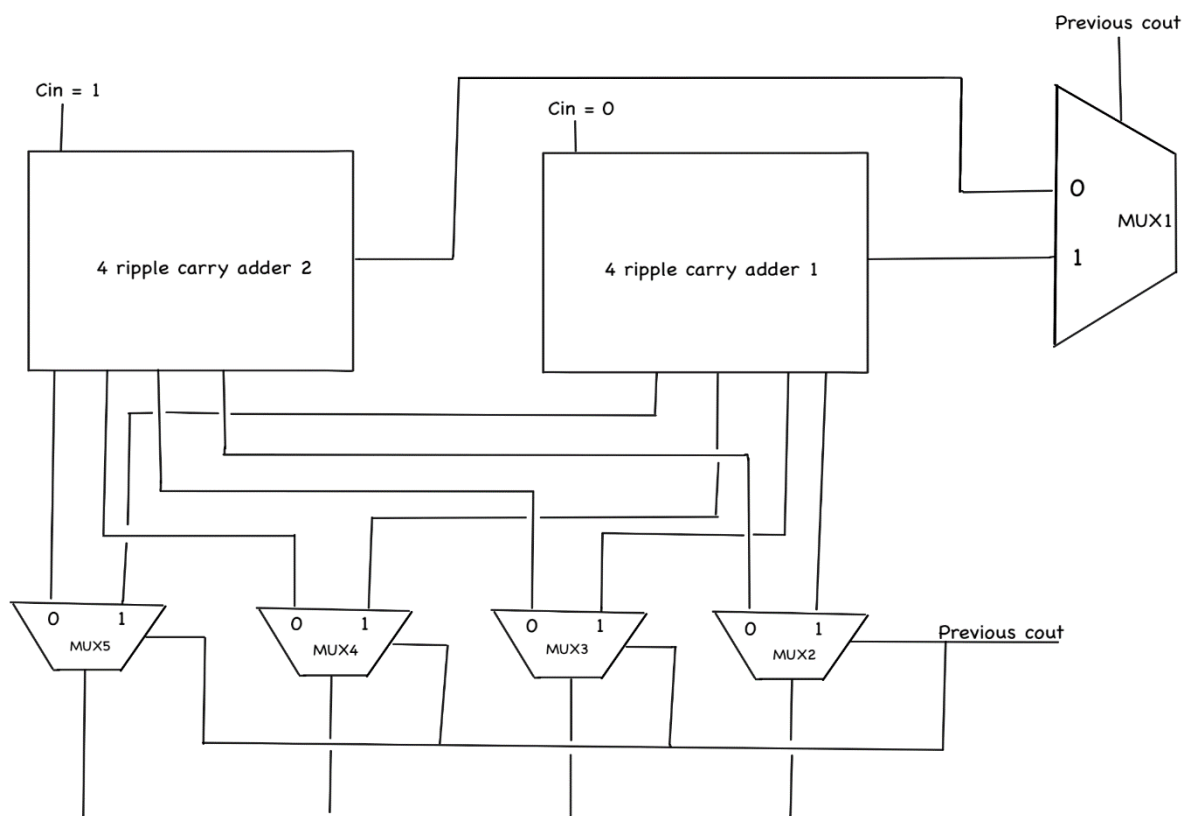
$$15 * (\text{carry delay}) + \text{max delay sum \& carry} = 15 * (3 + 2 + 2 = 7) + (2 * 3 = 6) = 112 \text{ ns}$$

اگر در waveform به بررسی تاخیر پردازیم مشاهده می شود ماکسیمم تاخیر اتفاق افتاده 112 نانوثانیه است و تاخیر های پیش آمده ممکن است کمتر هم باشند.

Step 4

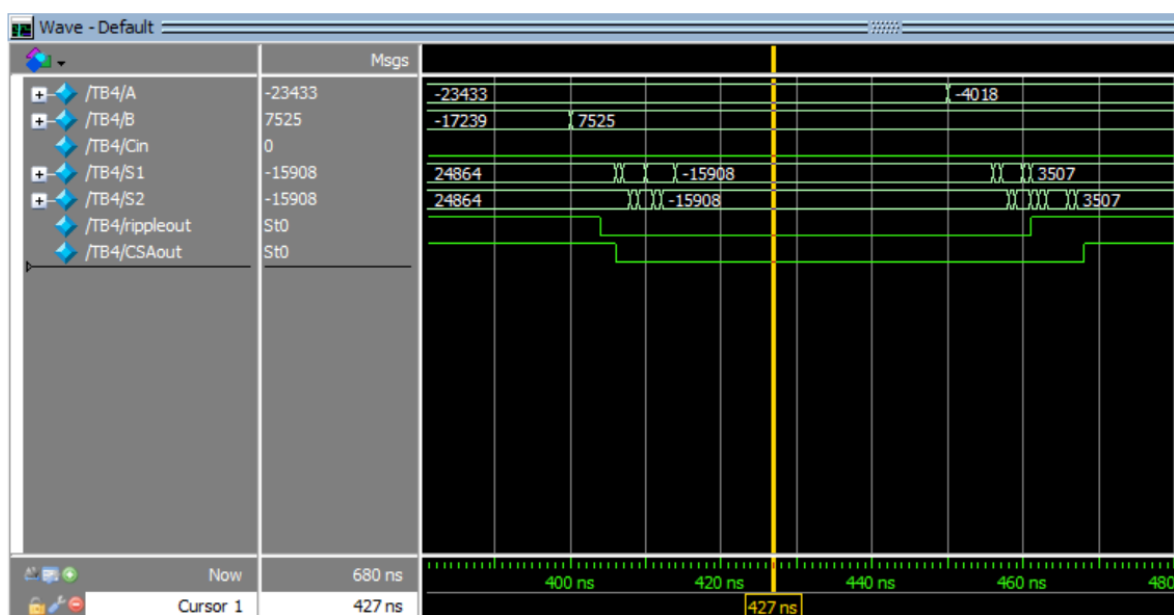
در این قسمت ابتدا ماژول یک مولتی پلکسر 2 به 1 را نوشتیم. طراحی من برای این قسمت به شکل بلاک زیر است که از 5 مولتی پلکسر و 2 تا ripple carry adder استفاده شده و سپس با کنار هم گذاشتن 4 تا از این بلاک ها یک CSA ساخته می شود:

(فایل وریلاگ 4)



$$\text{Worst case delay} = \text{carry delay} + 15 * \text{delay MUX} = 7 + 15 * 2 = 37 \text{ ns}$$

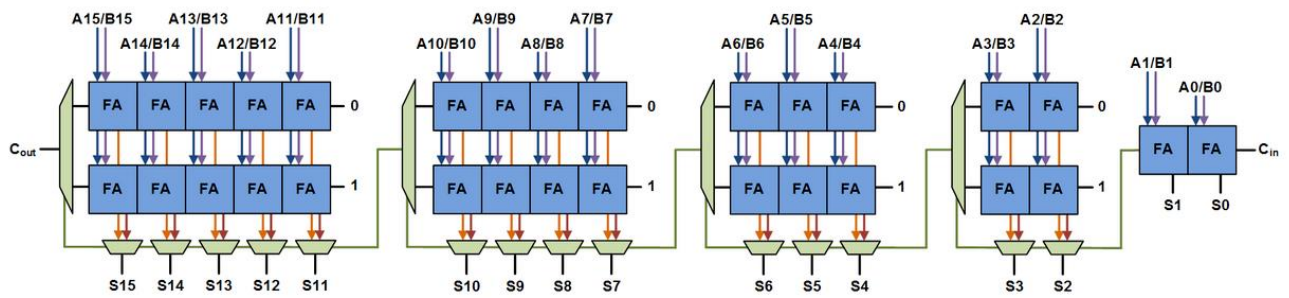
با نوشتن یک تست بنچ (فایل وریلاگ TB4) برای طراحی گام 4 و گام 3 رفتار آنها را مقایسه میکنیم:



تفاوتی که در زمان عوض شدن خروجی های این دو مشاهده می شود به دلیل 2 نانو ثانیه تاخیر مولتی پلکسر هاست که در CSA استفاده کردیم. اما در نهایت هر دوی آنها به درستی کار می کنند.

Step 5

برای سرعت بخشیدن به بخش قبل می‌توانیم برای جمع کردن 16 بیت به ترتیب از 5-4-3-2 جمع کننده استفاده کنیم. برای هر کدام بلاک های جداگانه نوشته و آنها را کنار هم قرار می دهیم:



Questions

- با کم کردن تعداد منابع در هر بلاک می توان در زمان صرفه جویی کرد چون اگر در هر بلاک جمع کننده 4تایی داشته باشیم از بلاک دوم به بعد قسمتی از آن ها خالی و بیکار باقی می مانند و فقط رد شدن دیتا از آنها زمان میبرد.
- در بخش پنجم ما با استفاده از متغیر سازی اندازه بلاک ها توانستیم سرعت نتیجه گیری را افزایش دهیم اما در عوض وادار به ساخت بلاک هایی با اندازه هایی متفاوت شدیم که منابع بیشتری از ما میگیرد.
پس به ازای کاهش زمان شبیه سازی مدار ما منابع بیشتری مصرف کردیم، درواقع tradeoff ما در این مثال، جایگزین کردن زمان با منابع بود.