تمرین عملی اول

درس معماری کامپیوتر

عرفان قصری9923061

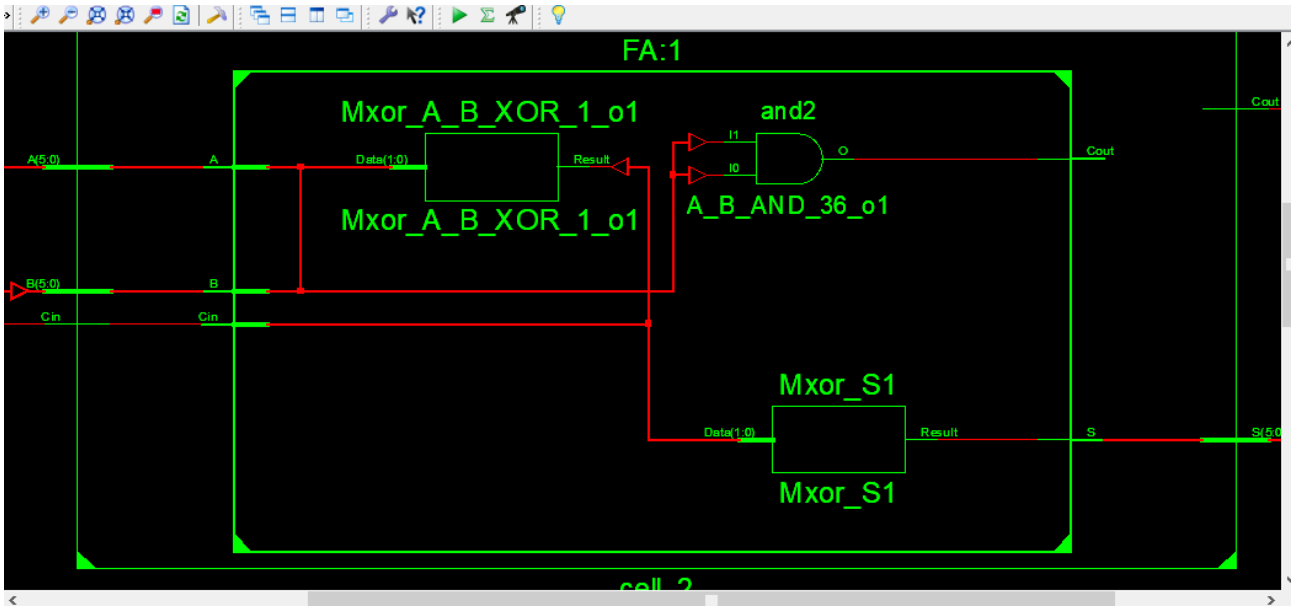
هدف از این تمرین اجرای سه نوع ضرب کننده شش بیتی و تست‌بنچ انها به کمک زبان VHDL است.

هر کدام از این ضرب‌کننده ها، دو باس شش بیتی (xوy) را به عنوان ورودی دریافت کرده و خروجی انها را در یک باس 12 بیتی p تحویل میدهند.

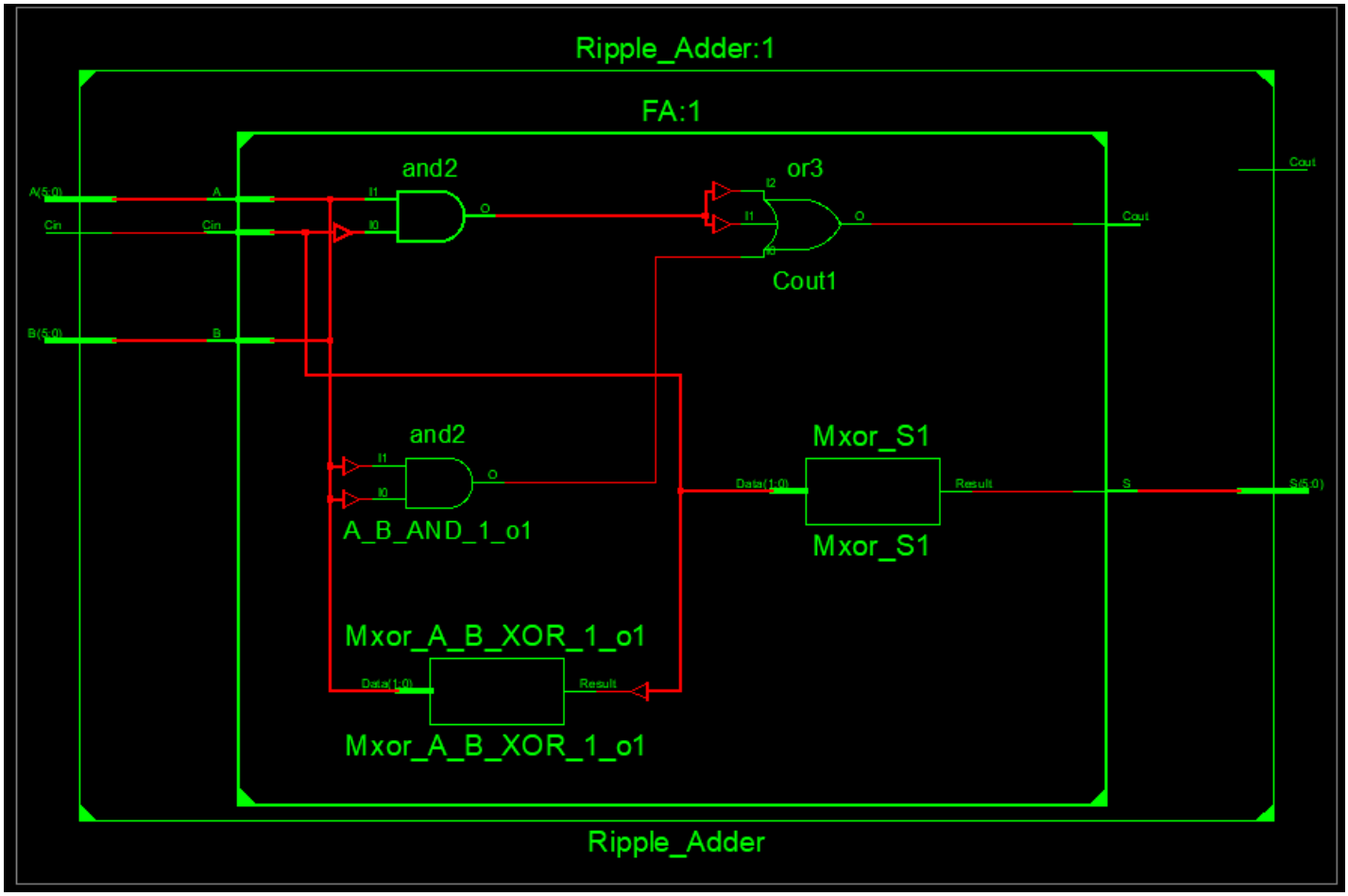
جهت جلوگیری از شلوغی فایل گزارش، تمامی کد ها در پوشه مجزا موجود هستند و در این فایل به انها اشاره نمیشود، همچنین تمامی فایل های شبیهسازی نیز به صورت ضمیمه ارسال شدند.

ضرب کننده اول:

در ابتدا نوع ساده ضرب کننده(simple multiplier) را پیاده سازی میکنیم که برای رسیدن به ان ابتدا نیاز داریم تا کامپوننت های مورد نیاز برای ساخت ساده تر ضرب کننده را پیاده سازی کنیم.



پیاده سازی شده.(فایلFA) full adder

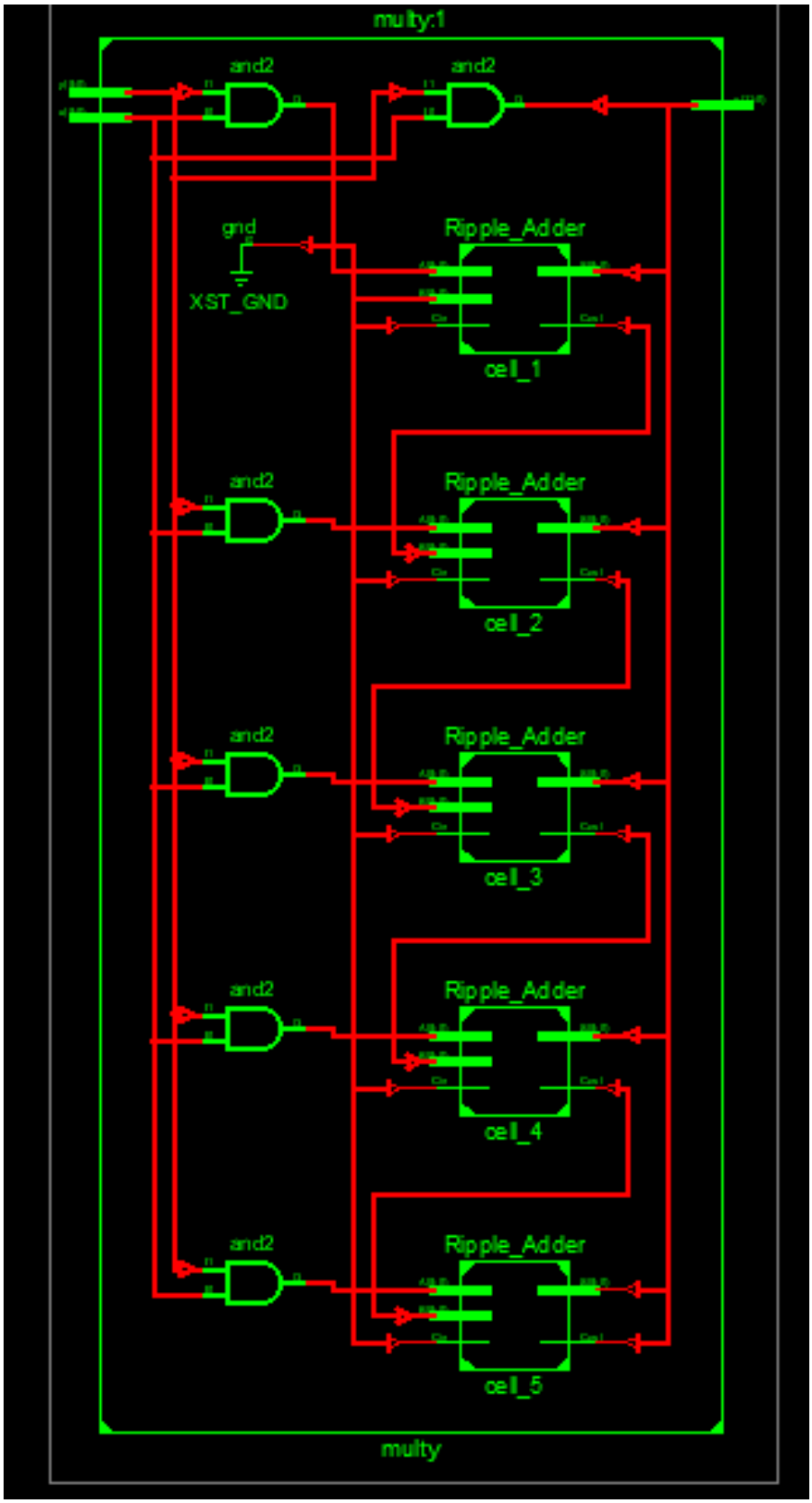


جمع کننده شش بیتی پیاده شده(فایلadder)

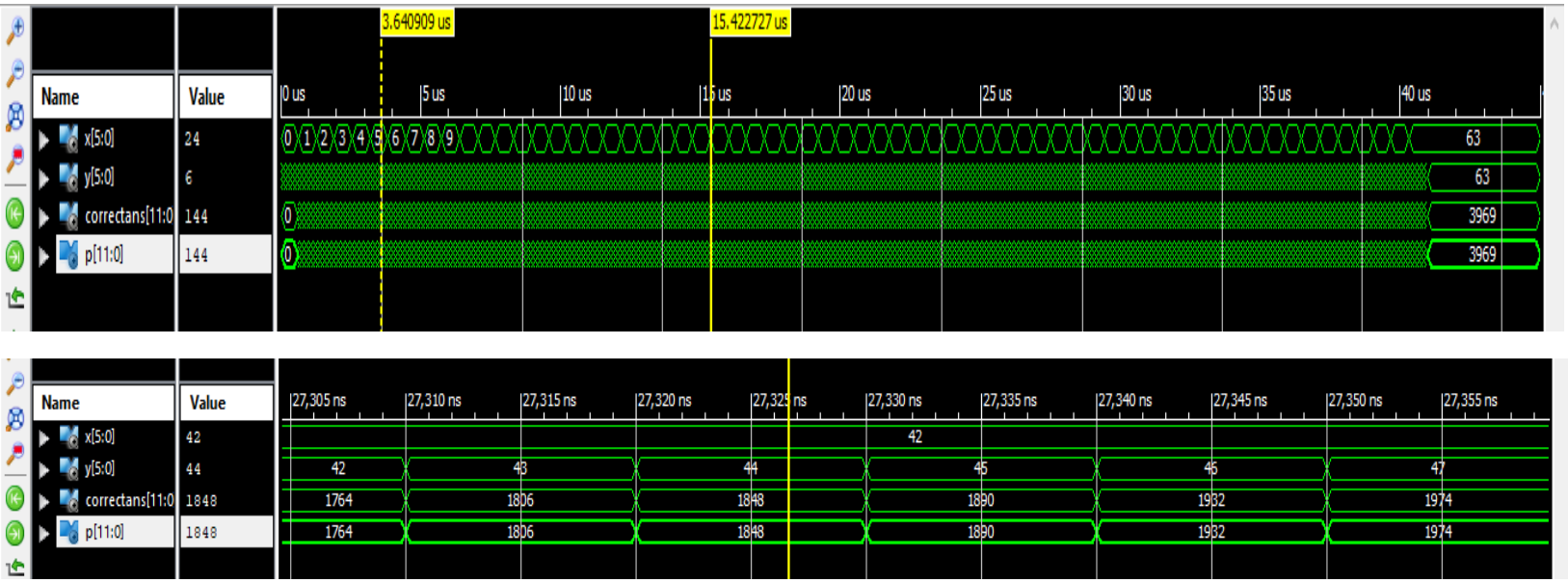
اکنون با استفاده از کامپوننت های پیاده شده به اجرای ضرب کننده می‌پردازیم.

(فایل

(Simplemulty



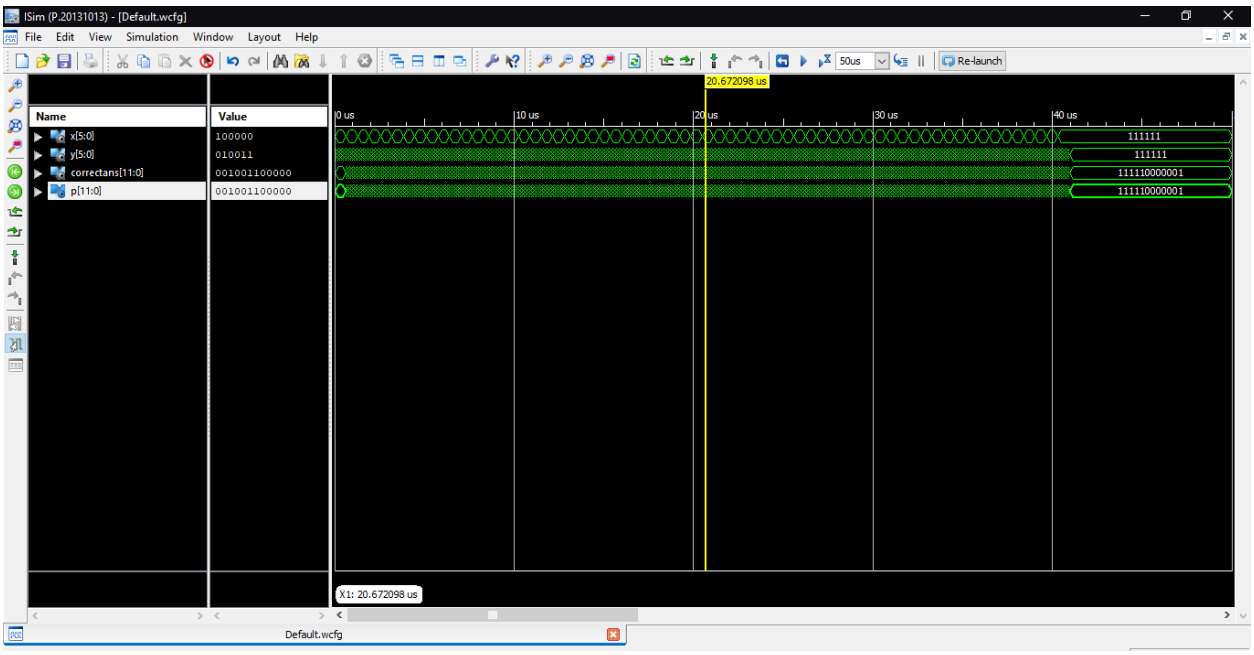
و در نهایت تست بنچ را روی ضرب کننده اعمال میکنیم تا نتایج به است امده راستی ازمایی شوند



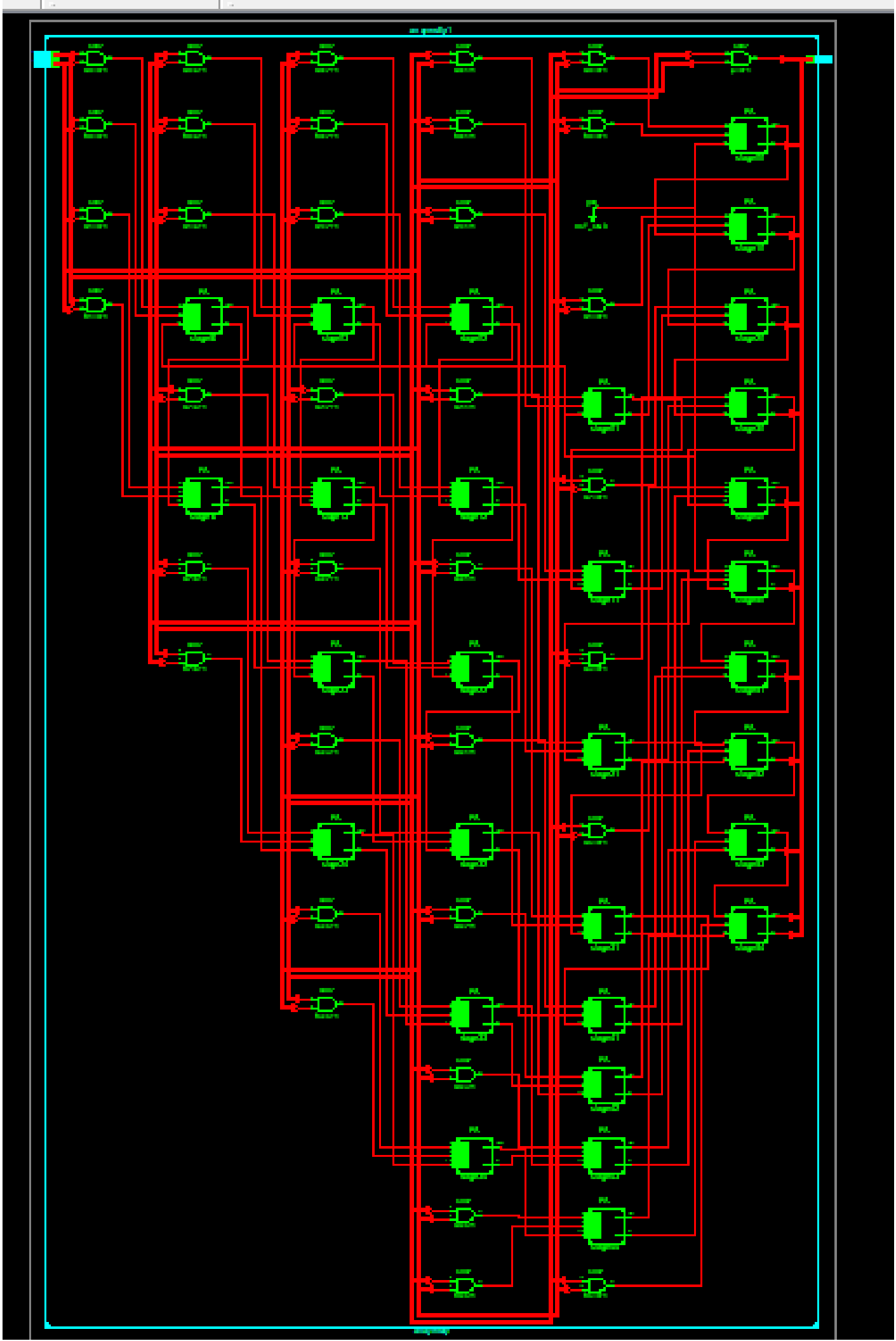
برای راحتی برسی،Radix ها بر روی unsigned decimal تنظیم شده‌اند و مقدار کمکی correctANS که از ضرب مبنای ده مقادیر تولید شده نیز در خط سوم اضافه شده.

ضرب کننده دوم:

با تجربه کسب شده از قسمت قبل و حتی تا حدی در این قسمت به این نتیجه رسیدم که هر چه کامپوننت استفاده شده کمتر، errorکمتر و زجر کمتر، پس این بخش را صرفا با استفاده از full adderها و گیت and پیاده میکنیم.



نتیجه مشابه مرحله قبل است و فایل تست بنچ نیز به سادگی با ان سازگار میشود.



ضرب کننده سوم:

در این ضرب کننده، تجربه کسب شده در قسمت قبل را بیخیال شدم و سعی کردم محض یادگیری و حل چالش هم که شده هر طبقه از ضرب کننده را به صورت یک کامپوننت در بیاورم.

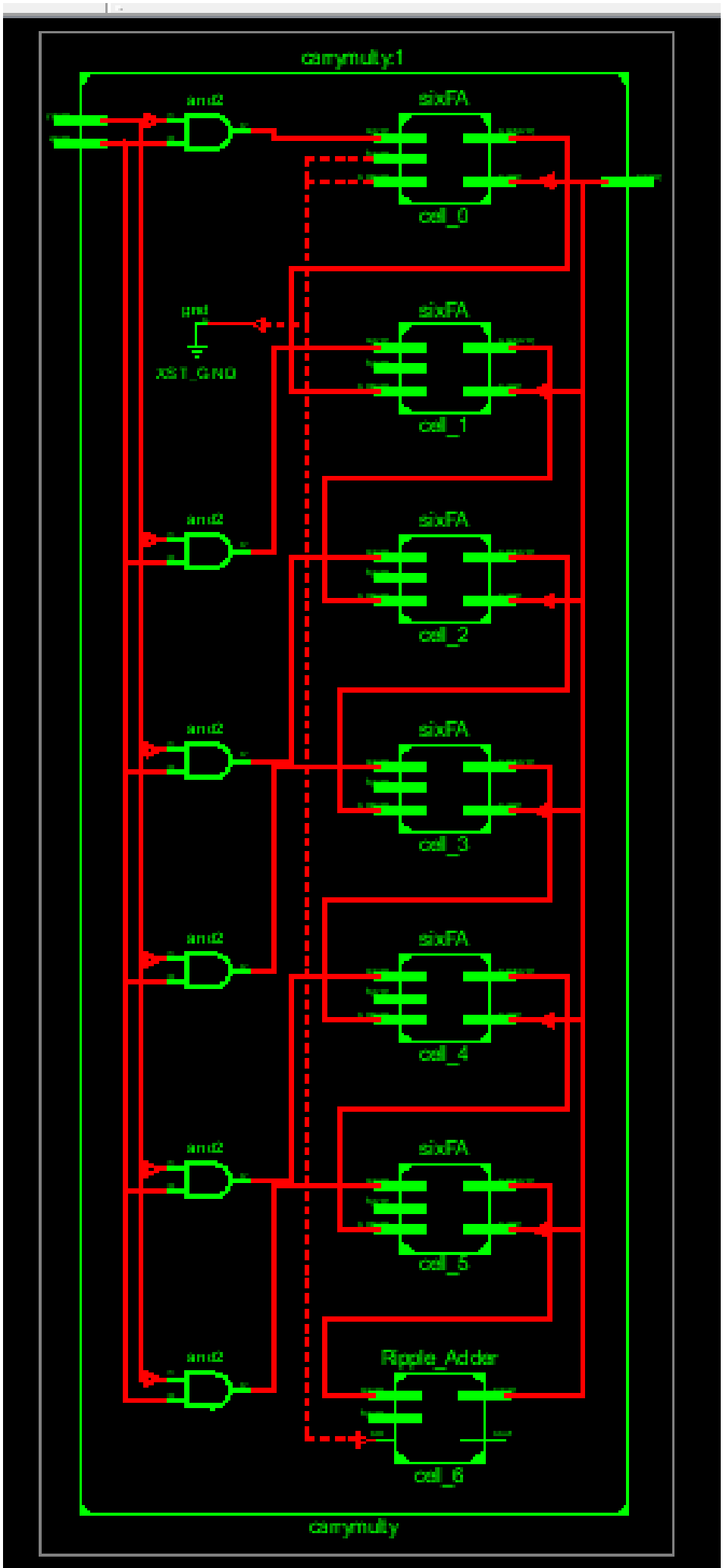
در ابتدا برای سادگی بیشتر پیاده سازی و پرهیز از ساخت کامپوننت جدید به جای قطعه ma از یک FA و یک گیت and در خارج ان استفاده کردم.

با کمی فکر کردن مشخص است که در طبقه اخر با کمی جا به جایی ورودی ها(که در نتیجه به دست امده از full adder تفاوتی ایجاد نمیکند) به جمع کننده شش بیتی که قبلا برای ضرب کننده اول نوشتیم میرسیم.

برای بقیه طبقات نیز کامپوننت جدید sixFA را نوشتم که از موازی کردن شش full adder به دست می‌آمد.

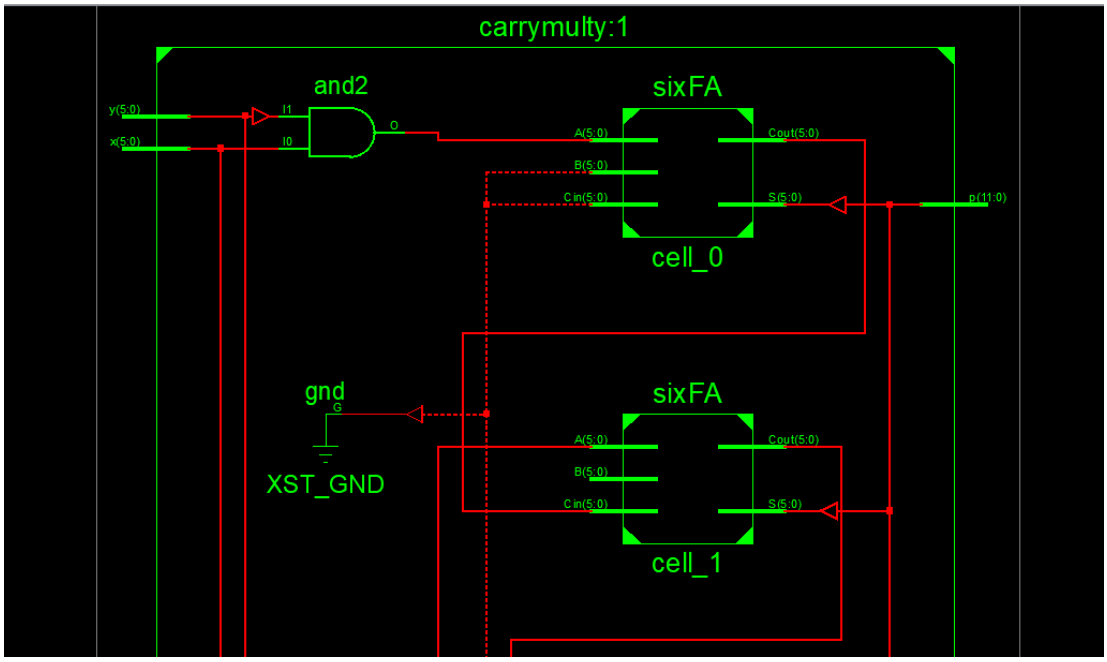
کمی توجه به نقشه داده شده در دستور کار نیز ما را به این نتیجه میرساند که این نوع ضرب کننده یک خروجی اضافی و dummy(الکی) دارد چراکه ضرب دو عدد شش بیتی،نهایا 12 بیت دارد و بیت p(12) قطعا صفر خواهد بود، ولی برای جلوگیری از مشکلات دستوری زبان،یک سیگنال dummy نیز به ان اختصاص دادم.

همانند بخش قبل نیز با کمی تغییر نام، تست بنچ نوشته شده برای این ضرب کننده نیز به کار رفت.

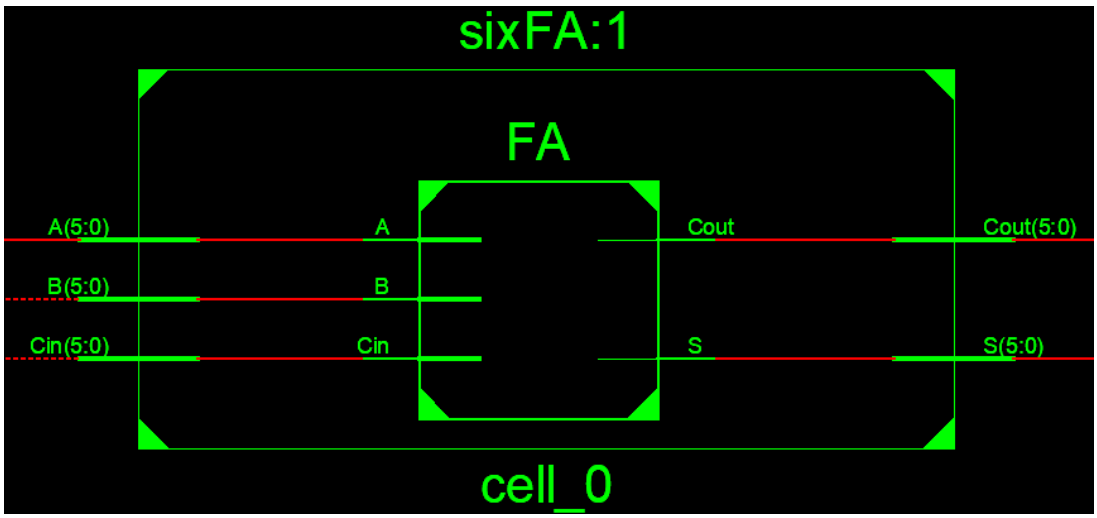


Carry save

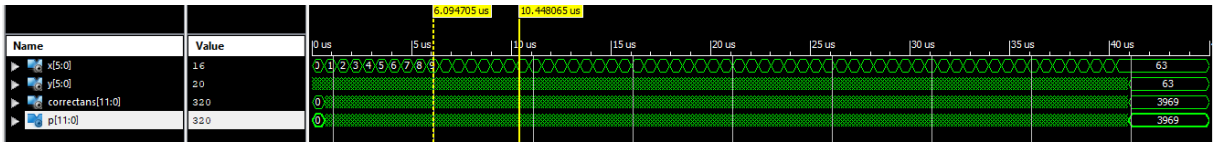
Multiplier



نمایی نزدیک تر از دو طبقه اول ضرب کننده



sixFA



نتیجه سیمولیشن(کاملا منطبق با قسمت های قبلی)