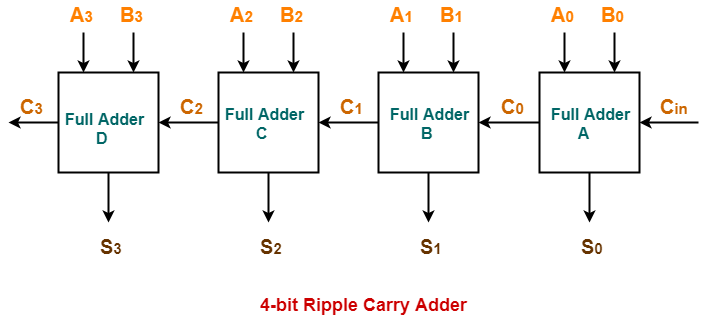
**به نام خدا**

**آزمایش اول**

**امین ساوه دورودی – محمدرضا اسکینی – کامیاب عابدی**

**پیاده سازی یک جمع کننده چهاربیتی Ripple Carry Adder**

**توضیحات اولیه** 

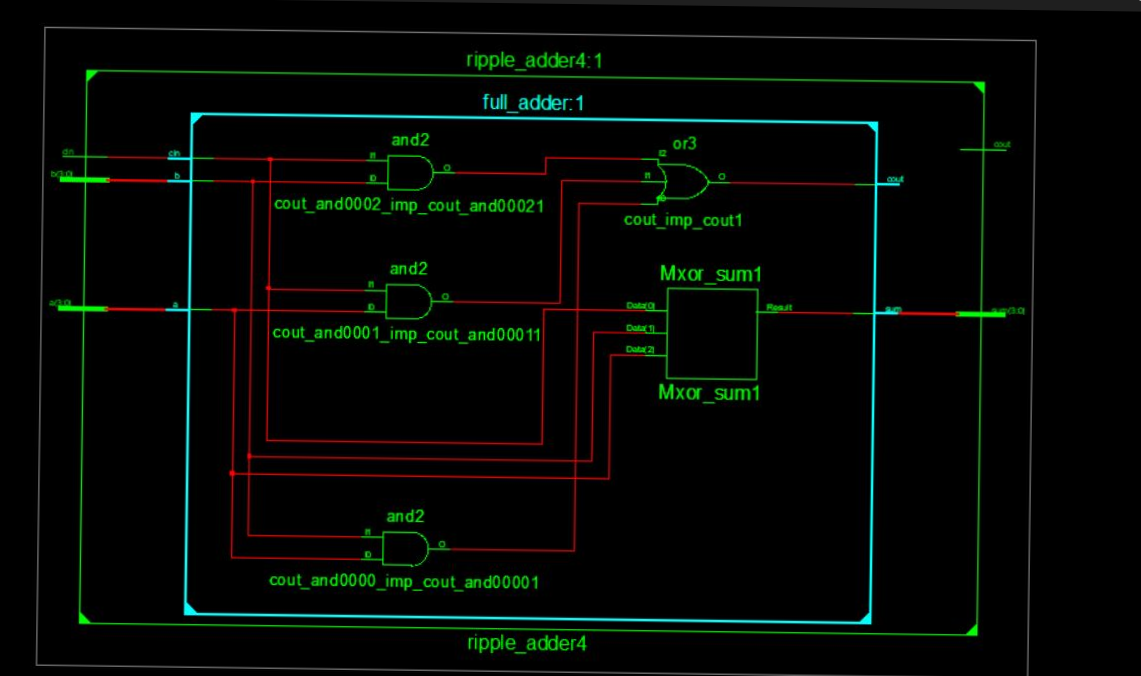
شماتیک یک جمع کننده چهاربیتی Ripple Carry Adder

در یک جمع کننده Ripple Carry Adder , همانند شکل بالا هر جمع کننده کامل به حمع کننده کامل قبلی خود وابسته است و خروجی Cout هر جمع کننده Cin جمع کننده بعدی خواهد بود. پس هر جمع کننده باید صبر کند تا محاسبات جمع کننده قبلی تمام شود و سپس کار خود را شروع کند که از این نظر , برای محاسبات بزرگ, به زمان زیادی نیاز داریم.

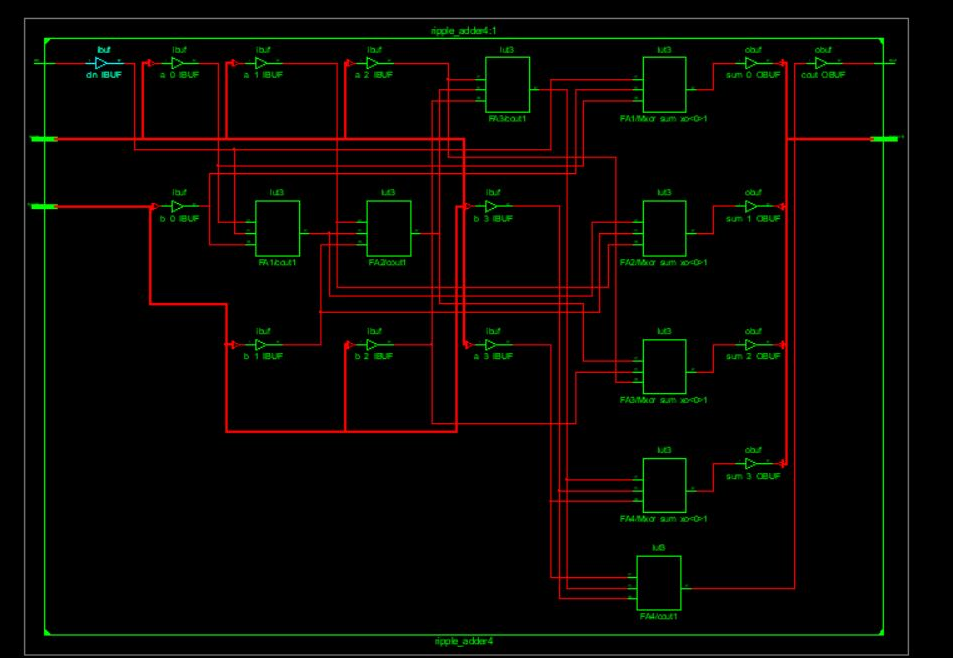
**پیاده سازی**

برای پیاده سازی به یک نمونه جمع کننده کامل یک بیتی (Full adder ) نیاز داریم و چهار نمونه از این جمع کننده میسازیم و مانند تصویر بالا , ورودی ها را به Full adder میدهیم و در آخر انتظار داریم محاسبه به درستی انجام شود.

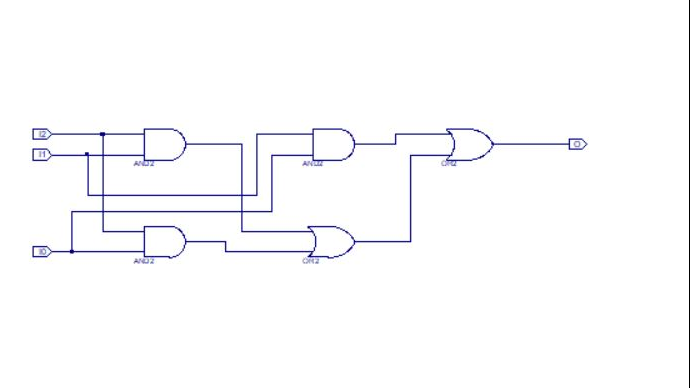
**مستندات**



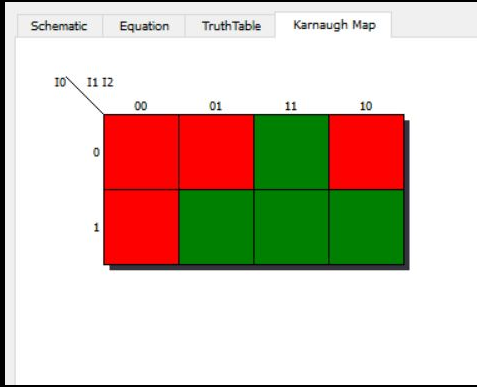
عکس ۱ ( RTL )



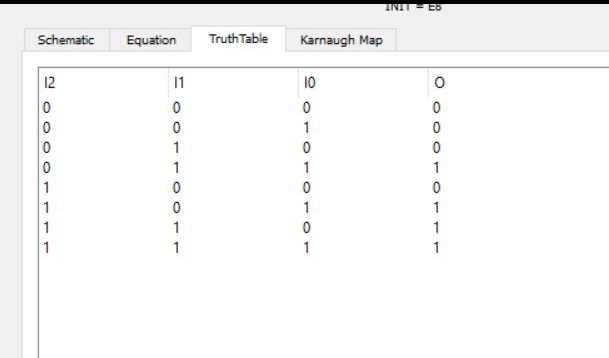
عکس ۲ (Technology)



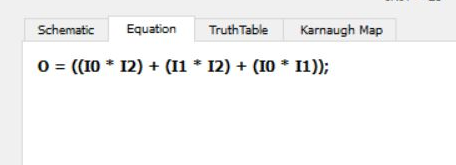
عکس ۳ ( Schematic)



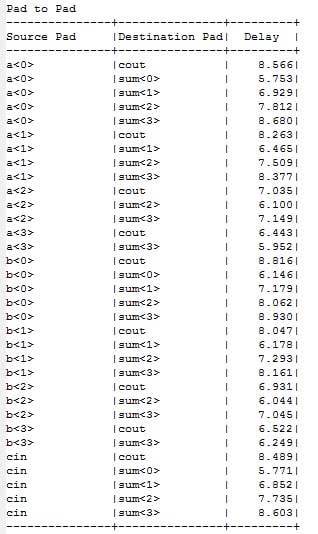
عکس ۴ (Karnaugh Map )



عکس ۵ ( Truth Table )



عکس ۶ ( Equation )

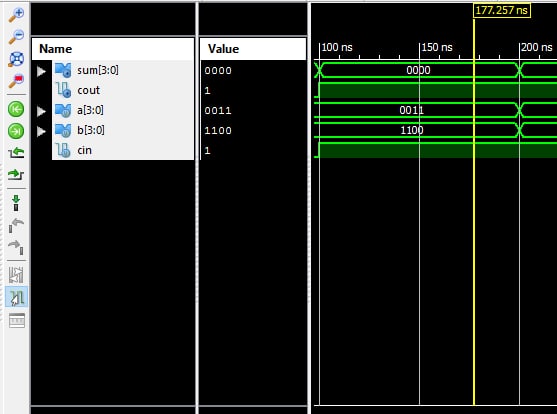


عکس ۷ ( جدول تاخیر)

**تست**



عکس ۸



عکس ۹



عکس ۱۰

تست اول :

A=0011 B= 1100 Cin=0

S = 0011 + 1100 = 1111 Cout=0 انتظار ما :

همانطور که در عکس ۸ مشاهده میشود در بازه زمانی ۰ تا ۱۰۰ نانوثانیه مقدار خروجی برابر انتظارات ما میباشد.

تست دوم :

A=0011 B= 1100 Cin=1

S = 0011+1100 = 0000 Cout=1 انتظار ما :

بعد از ۱۰۰ نانوثانیه مقدار ورودی ها تغییر میکنند.

همانطور که در عکس ۹ مشاهده میشود در بازه زمانی ۱۰۰ تا ۲۰۰ نانوثانیه مقدار خروجی برابر انتظارات ما میباشد.

تست سوم :

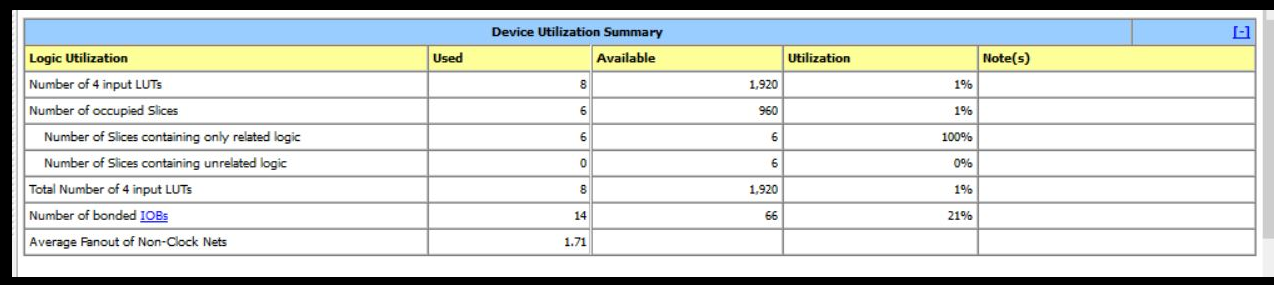
A=0001 B= 1111 Cin=1

S = 0001 + 1111 = 0001 Cout=1 : انتظار ما

بعد از ۱۰۰ نانوثانیه مقدار ورودی ها تغییر میکنند.

همانطور که در عکس ۱۰ مشاهده میشود در بازه زمانی ۲۰۰ تا ۳۰۰ نانوثانیه مقدار خروجی برابر انتظارات ما میباشد.

**خلاصه طراحی**



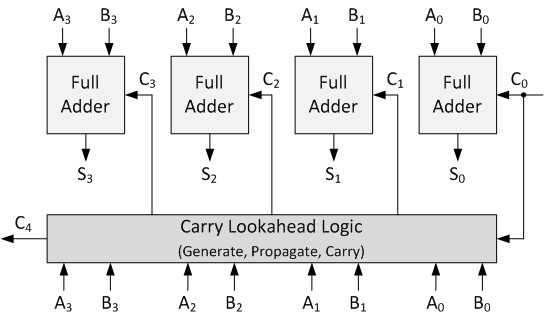
عکس ۱۱

طبق عکس بالا, خلاصه ای از کمیت های طراحی در قالب جدولی نشان داده شده است.

همانطور که مشاهده میشود , این نوع طراحی ۸ LUT را شامل میشود که در ۶ Slice در حال انجام هستند.

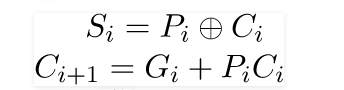
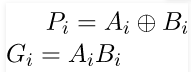
**پیاده سازی یک جمع کننده چهاربیتی Carry Look Ahead Adder**

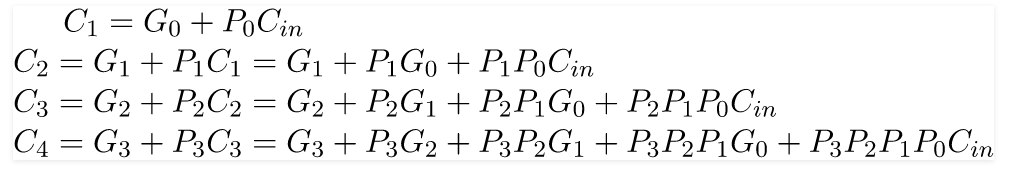
**توضیحات اولیه**



شماتیک یک جمع کننده چهاربیتی Carry Look Ahead

در یک جمع کننده چهاربیتی Carry Look Ahead Adder برخلاف Ripple Carry Adder, هر جمع کننده کامل یک بیتی به جمع کننده قبلی خود وابسته نیست. مقدار cin هر جمع کننده کامل یک بیتی طبق فرمول هایی سریعتر از محاسبات جمع کننده کامل به دست آورده میشود و در اختیار جمع کننده های یک بیتی قرار میگیرد. به همین دلیل , فرآیند محاسبه سریعتر خواهد بود.

-  

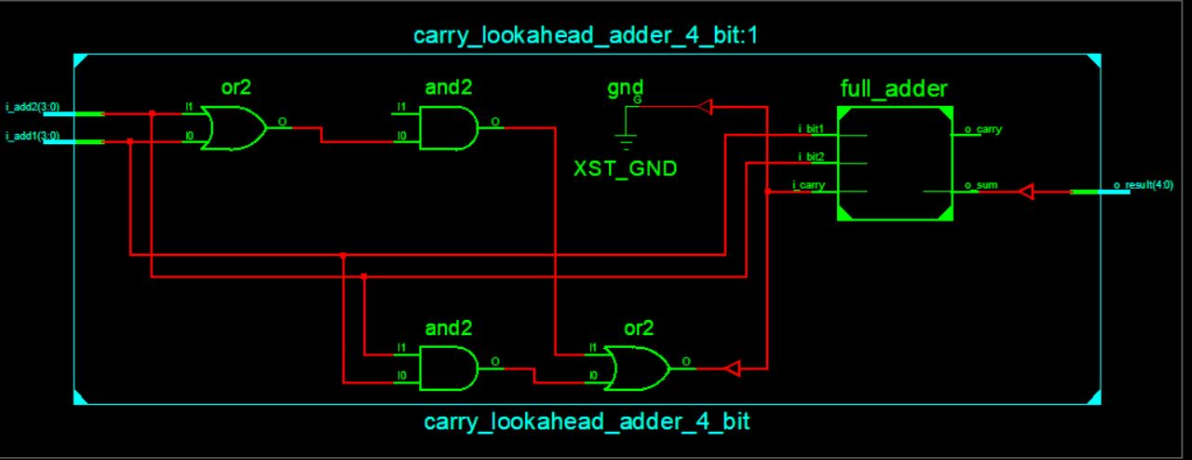


تصاویر بالا نحوه محاسبه سریعتر cin و cout را نشان میدهد که بدون وابستگی به جمع کننده های کامل میتوانیم آن ها را بدست آوریم و به جمع کننده های کامل دهیم تا مجموع بیت های متناظر را محاسبه کنند.

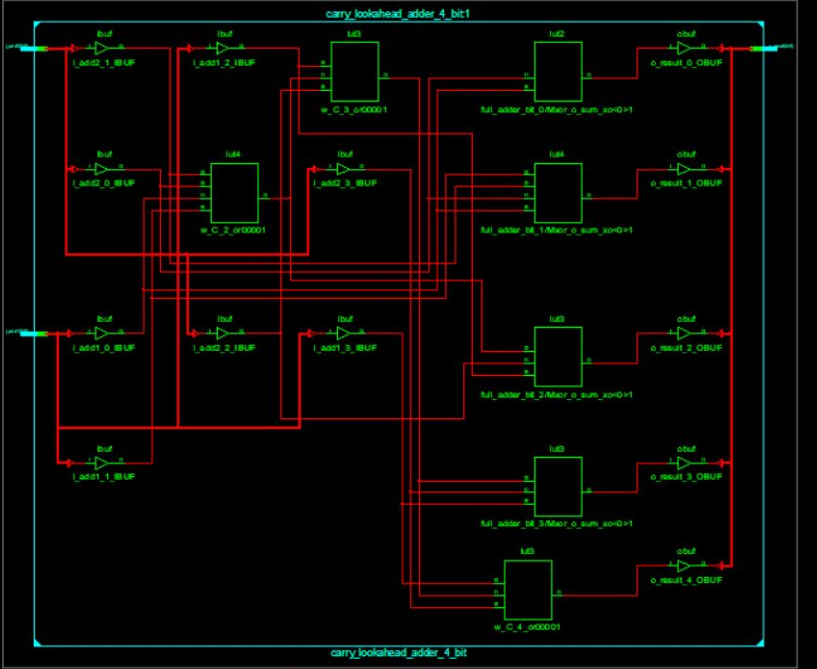
**پیاده سازی**

برای پیاده سازی به یک نمونه جمع کننده کامل یک بیتی (Full adder ) نیاز داریم و چهار نمونه از این جمع کننده میسازیم و با استفاده از فرمول های بالا, ورودی ها را به Full adder میدهیم و در آخر انتظار داریم محاسبه به درستی انجام شود.

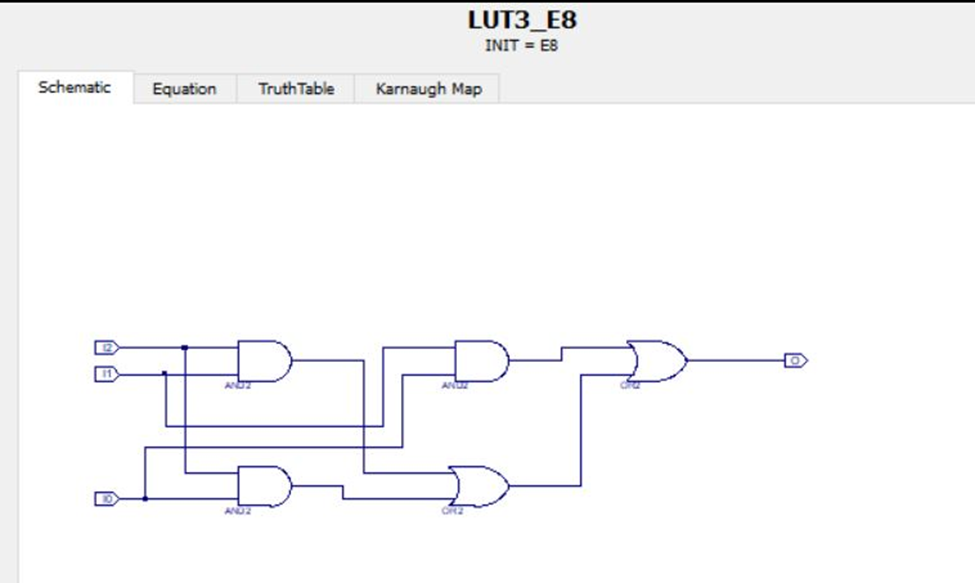
**مستندات**



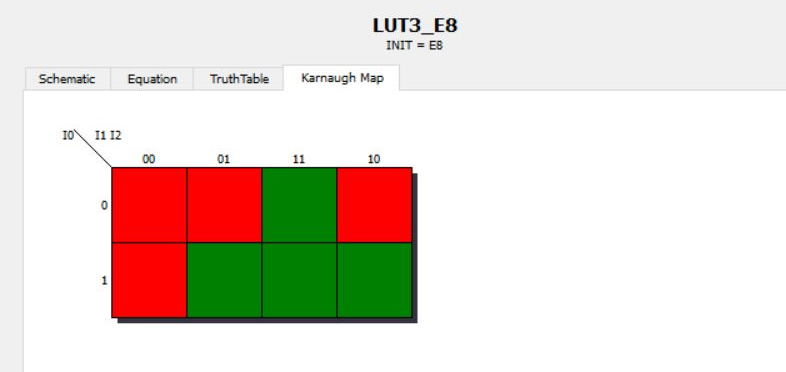
عکس ۱ ( RTL )



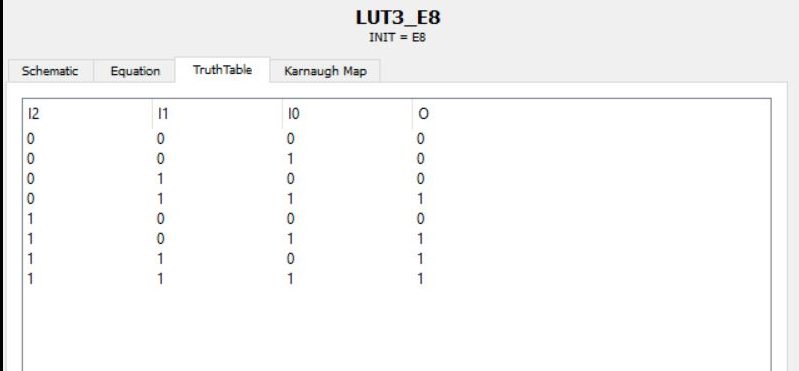
عکس ۲ (Technology)



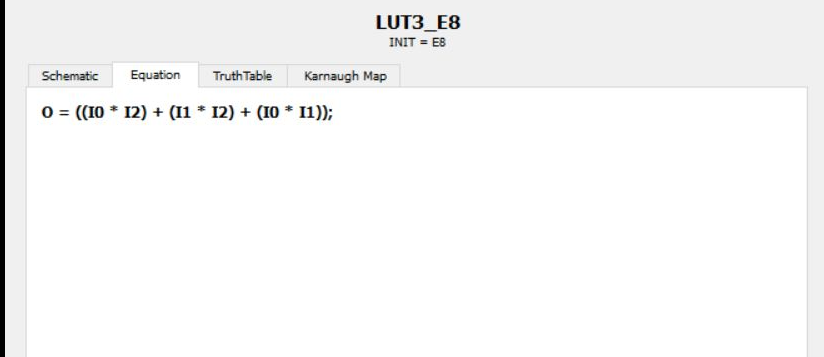
عکس ۳ ( Schematic)



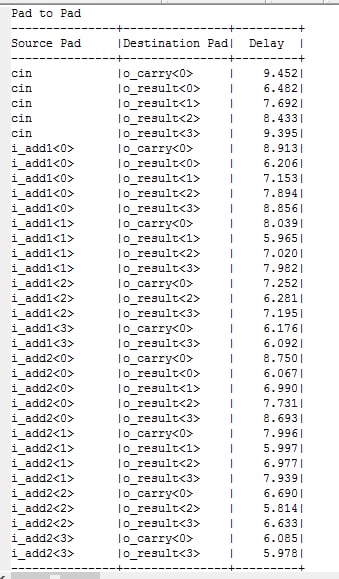
عکس ۴ (Karnaugh Map )



عکس ۵ ( Truth Table )

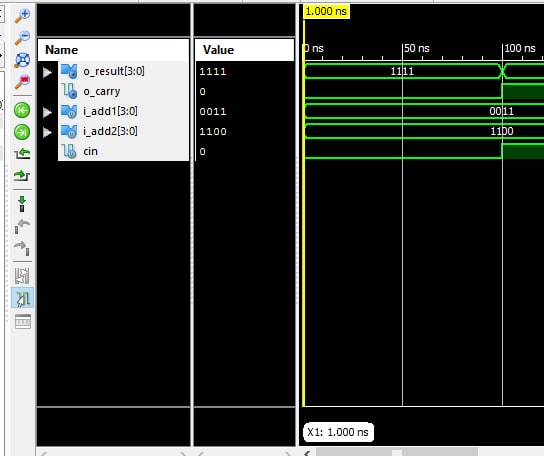


عکس ۶ ( Equation )



عکس ۷ ( جدول تاخیر)

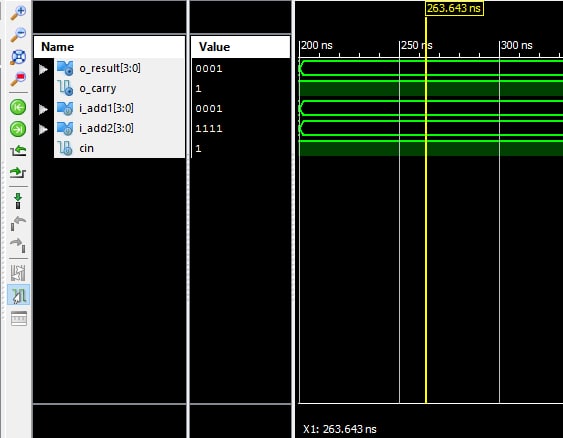
**تست**



عکس ۸



عکس ۹



عکس ۱۰

تست اول :

A=0011 B= 1100 Cin=0

S = 0011 + 1100 = 1111 Cout=0 انتظار ما :

همانطور که در عکس ۸ مشاهده میشود در بازه زمانی ۰ تا ۱۰۰ نانوثانیه مقدار خروجی برابر انتظارات ما میباشد.

تست دوم :

A=0011 B= 1100 Cin=1

S = 0011+1100 = 0000 Cout=1 انتظار ما :

بعد از ۱۰۰ نانوثانیه مقدار ورودی ها تغییر میکنند.

همانطور که در عکس ۹ مشاهده میشود در بازه زمانی ۱۰۰ تا ۲۰۰ نانوثانیه مقدار خروجی برابر انتظارات ما میباشد.

تست سوم :

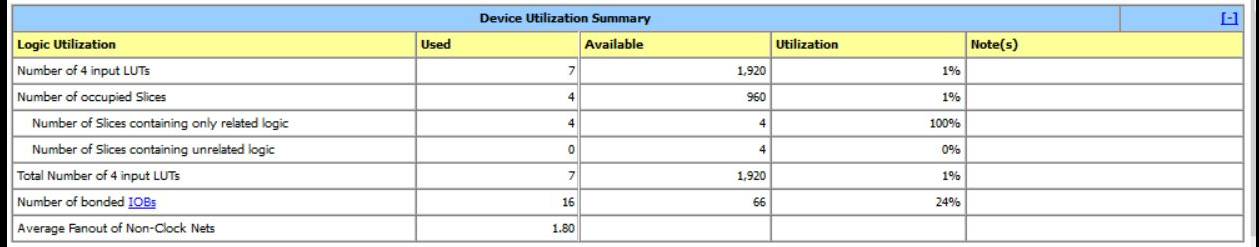
A=0001 B= 1111 Cin=1

S = 0001 + 1111 = 0001 Cout=1 : انتظار ما

بعد از ۱۰۰ نانوثانیه مقدار ورودی ها تغییر میکنند.

همانطور که در عکس ۱۰ مشاهده میشود در بازه زمانی ۲۰۰ تا ۳۰۰ نانوثانیه مقدار خروجی برابر انتظارات ما میباشد.

**خلاصه طراحی** :



عکس ۱۱

طبق عکس بالا, خلاصه ای از کمیت های طراحی در قالب جدولی نشان داده شده است.

همانطور که مشاهده میشود , این نوع طراحی ۷ LUT را شامل میشود که در ۴ Slice در حال انجام هستند.

**نتیجه گیری**

طبق مقایسه جدول های خلاصه طراحی هر دو جمع کننده یعنی تصویر های ۱۱ هر دو بخش , طبق انتظارات ما همانطور که در توضیحات اولیه آورده شده است, جمع کننده Carry Look Ahead Adder تعداد کمتری LUT در طراحی خود دارد و همینطور slice های کمتری نسبت به Ripple Carry Adder اشغال میکند. و هم چنین طبق توضیحات اولیه Carry Look Ahead Adder در محاسبات سنگین به دلیل انجام محاسبات به صورت موازی در جمع کننده های کامل آن , سریعتر میباشد که با مقایسه تصویر های ۷ هر دو بخش به این نتیجه به صورت عملی خواهیم رسید.