

باسمه تعالی

گزارش پروژه درس FPGA

اعضای گروه: احمدرضا یزدانی، مهدی مردانی

ماژول ها:

1. Image\_Converter

2. Encoder

3. UART\_Sender

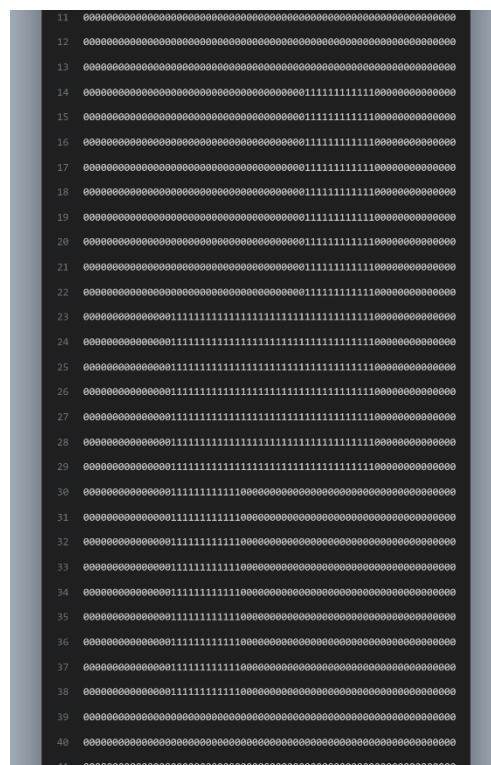
4. UART\_Receiver

5. Decoder

6. Main

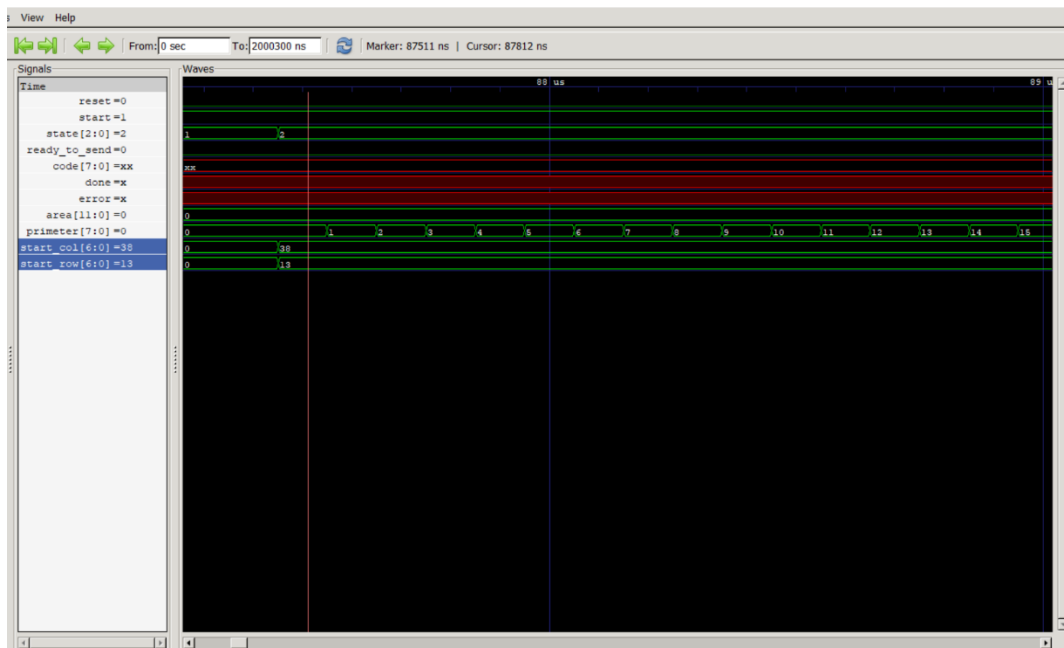
**:Image\_Converter**

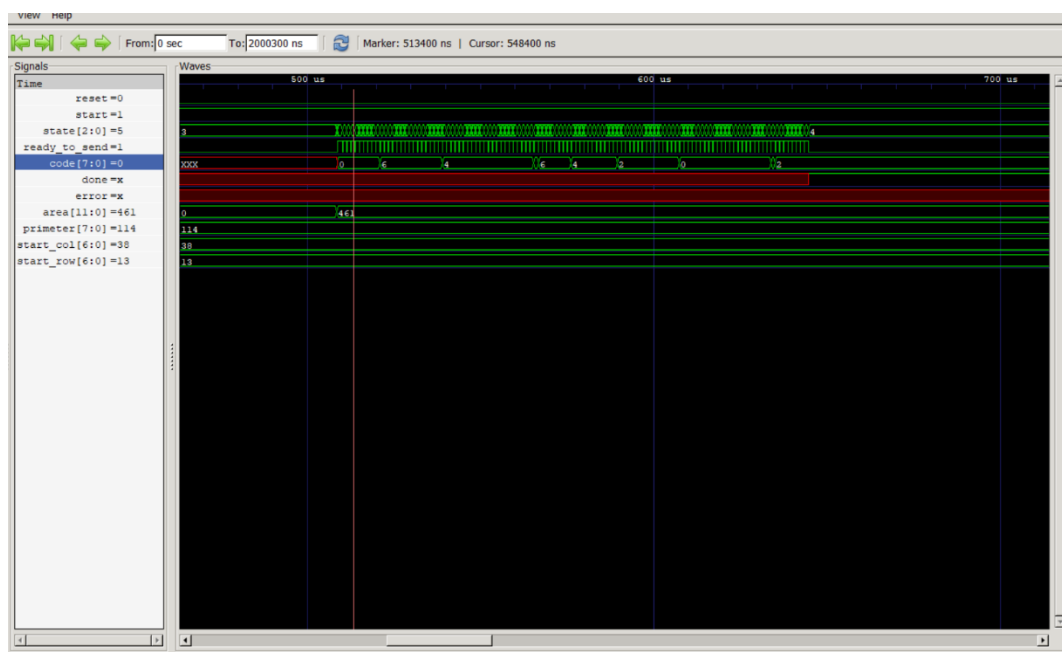
در این ماژول که با زبان پایتون نوشته شده است میتوان عکس مورد نظر را به صورت یک فایل باینری ذخیره کرد.



## :Encoder

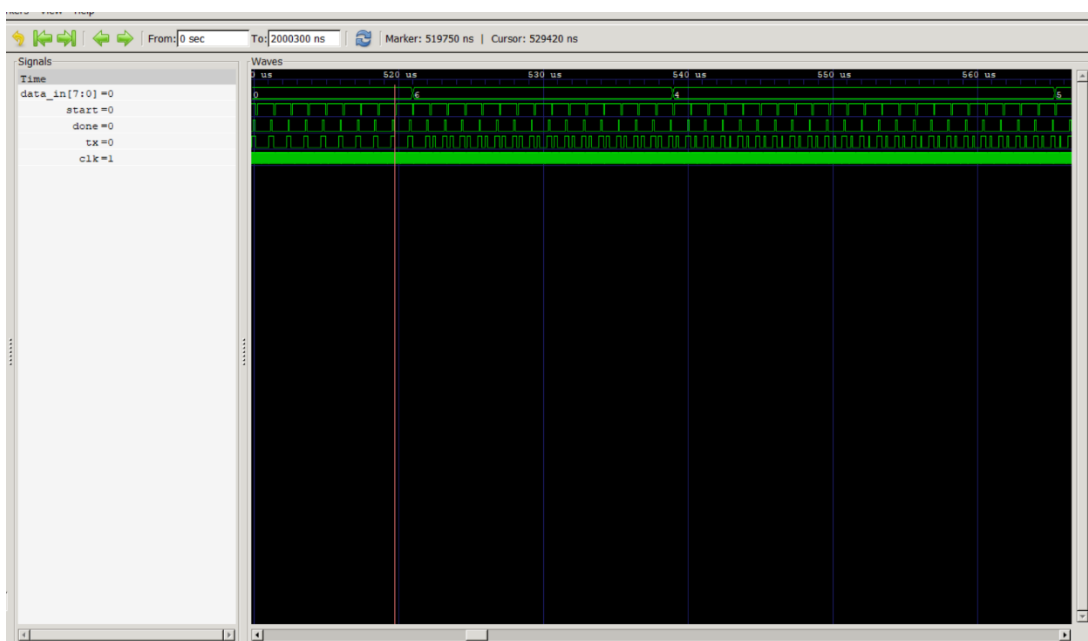
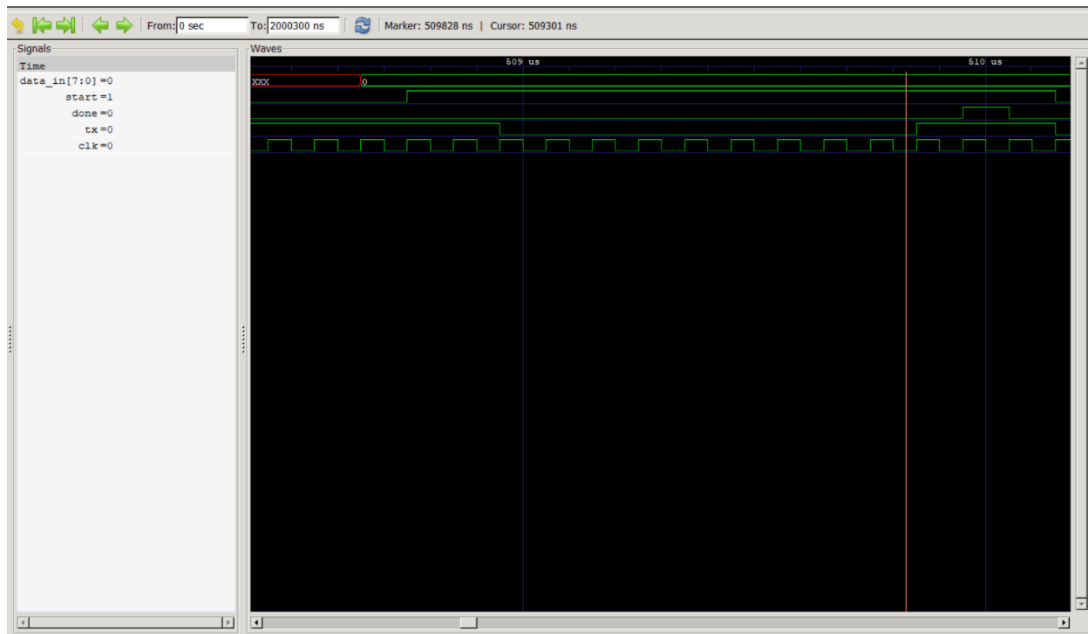
این ماژول به صورت FSM پیاده سازی شده است. به این صورت که بعد از ست شدن و سپس صفر شدن سیگنال ورودی reset تمام رجیستر های داخلی برابر صفر خواهند شد. سپس ماشین وارد حالت s1 شده و اطلاعات عکس را از فایل مربوطه میخواند. سپس هنگامی که سیگنال ورودی start ست شود ماشین وارد حالت s2 میشود. در s2 ماشین از پیکسل اول شروع کرده و به ترتیب هر ردیف را از چپ به راست و از بالا به پایین پیمایش میکند تا اولین پیکسل مشکی را پیدا کند. بعد از پیدا کردن پیکسل مشکی اول مقدار x,y آن را در رجیستر مربوطه ذخیره میکند و ماشین به حالت s3 میرود. در این حالت ماشین با توجه به الگوریتم توضیح داده شده در توضیحات پروژه دور تا دور شکل حرکت میکند و حرکات خود را کدگذاری میکند همچنین حین حرکت محیط شکل را نیز بدست می آورد. بعد از برگشتن به خانه اول ماشین به حالت s4 میرود. در حالت s4 ماشین از اول کل عکس را یکبار پیمایش میکند تا مساحت شکل را بدست آورد و آن را ذخیره کند. بعد از بدست آوردن مساحت شکل ماشین به حالت s5 میرود. در حالت s5 سیگنال ready\_to\_send صفر میشود تا ماشین کد های ذخیره شده را روی خروجی قرار دهد. سپس ماشین به حالت s6 رفته و در این حالت این سیگنال را یک میکند تا به UART\_Sender اطلاع دهد که دیتای خروجی را ارسال کند. تا زمانی که ارسال UART\_Sender تمام نیافته و سیگنال sender\_done یک نشده ماشین در این حالت متوقف میشود. پس از یک شدن سیگنال sender\_done ماشین به حالت s5 برمیگردد تا بقیه دیتا را روی خروجی قرار دهد.





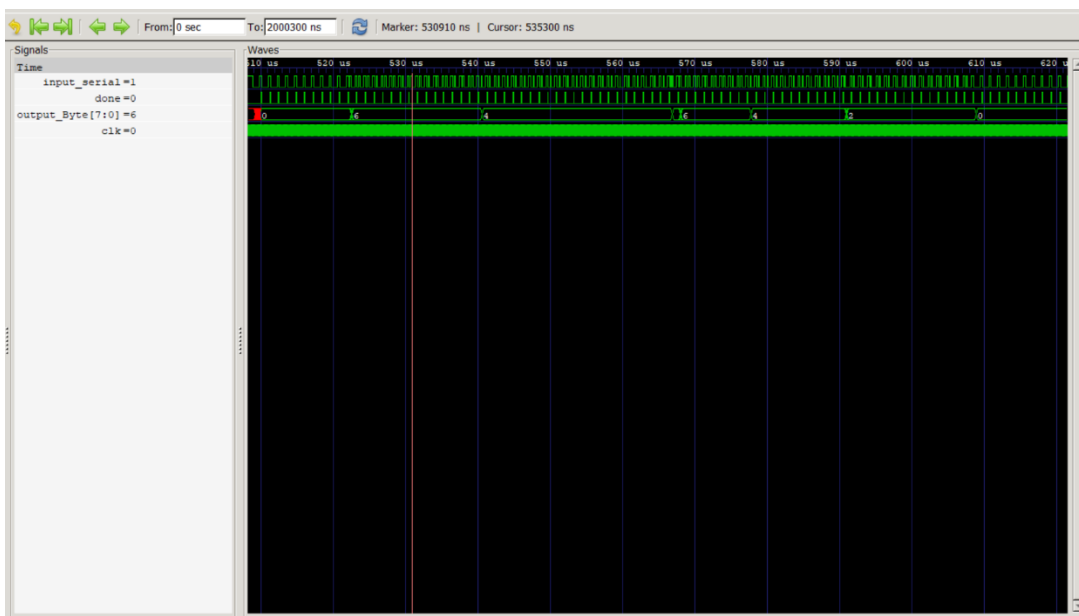
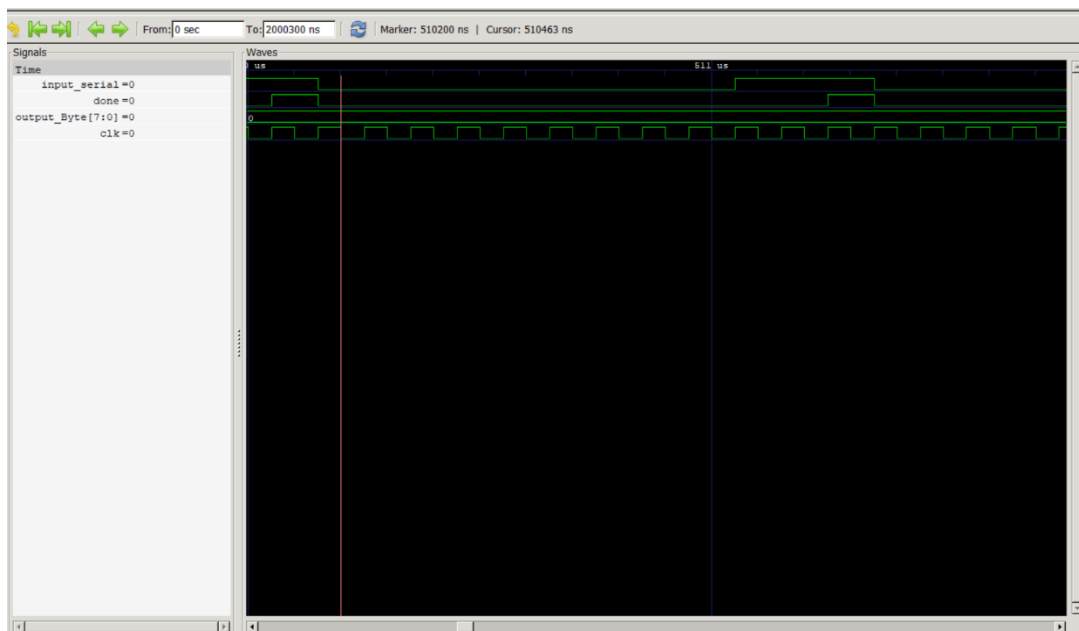
## :UART\_Sender

این ماژول نیز با FSM پیاده سازی شده. حالت پیشفرض و اولیه آن حالت IDLE است که در این حالت دیتایی ارسال نمیشود و خروجی در طول این مدت یک است. اگر سیگنال start در حالت IDLE فعال شود ماشین دیتا را از ورودی به صورت موازی میخواند و در خود ذخیره میکند. سپس به حالت START\_BIT میرود. در این حالت ماشین خروجی خود را برای لحظه ای صفر میکند. این صفر شدن در سمت گیرنده نشانه شروع یک فریم از دیتاست. فریم های دیتای ما ده بیت در نظر گرفته شده است که شامل یک بیت صفر شروع، هشت بیت دیتا و یک بیت یک پایانی میشود. سپس ماشین به حالت DATA\_BITS میرود. هشت بیت دیتا به ترتیب روی خروجی گذاشته میشوند. پس از آن ماشین به حالت STOP\_BIT میرود که در این حالت بیت یک پایانی روی خروجی گذاشته میشود. پس از آن ماشین به حالت CLEAN\_UP رفته که در این حالت سیگنال خروجی done را ست میکند که به معنای اتمام ارسال فریم است و پس از آن به حالت IDLE باز میگردد.



## :UART\_Receiver

این ماژول نیز مانند ماژول های قبل با FSM طراحی شده. در حالت اول که حالت IDLE است ماشین منتظر تغییر و صفر شدن دیتای ورودی است. پس از صفر شدن آن ماشین بلافاصله به حالت START\_BIT میرود و بیت صفر اولیه را دریافت میکند. پس از آن ماشین وارد حالت DATA\_BITS شده و دیتا های ورودی را به ترتیب میخواند و در رجیستر خود ذخیره میکند. پس از دریافت هشت بیت ماشین وارد حالت STOP\_BIT شده و بیت یک پایانی را دریافت میکند و سیگنال done خود را یک میکند تا اطلاع دهد که دیتا خوانده شده و روی پورت خروجی قرار گرفته است. سپس به حالت CLEANUP میرود و سیگنال done را صفر میکند و به حالت IDLE برمیگردد.



## :Decoder

FSM ما در این مازول چهار حالت را شامل میشود. در صورتی که بیت reset یک نباشد ماشین وارد حالت s1 میشود. در حالت s1 ماشین با کمک سیگنال ورودی کنترلی done\_receiver دیتا ها را از ورودی در وقت مناسب میخواند. سیگنال start زمانی برای این مازول ارسال میشود که آخرین ارسال از سمت فرستنده صورت گرفته باشد. در صورت یک شدن این سیگنال ماشین به حالت s2 میرود. در این حالت ماشین با کمک یک شمارنده مدتی منتظر دیتای جدید میماند تا اگر در این فاصله دیتایی در میان راه وجود داشته باشد فرصت کند به مقصد برسد. بعد از اتمام شمارنده ماشین به حالت s3 میرود. این مازول دارای یک حافظه جداگانه است که ابعاد این حافظه دقیقاً به اندازه عکس اولیه است. فرض بر این گرفته شده که این حافظه هیچ پیکسل سیاهی ندارد. در این حالت از ماشین، مازول با کمک  $x, y$  گرفته شده از فرستنده شروع به حرکت و ساخت شکل اولیه طبق الگوریتم میکند و آن را در حافظه مذکور ذخیره میکند. همچنین در این حین محیط شکل ساخته شده را بدست می آورد. پس از آن ماشین به حالت s4 رفته و مقدار محیط شکل اولیه که از سمت فرستنده دریافت شده را با محیطی که خود به دست آورده مقایسه میکند. اگر محیط ها متفاوت بود یعنی عکس ساخته شده همان عکس اولیه نیست پس سیگنال error فعال میشود. همچنین تمامی خانه های حافظه روی شصت و چهار پورت خروجی گذاشته شده تا نتیجه نهایی قابل مشاهده باشد.

