

به نام خدا

ارسلان فیروزی - ۹۷۱۰۲۲۲۵

پروژه فاز ۴ - FPGA

کد های وریلاگ به تمامی کامنت گذاری شده است و عملکرد آن در کد مشخص شده است.

Specification:

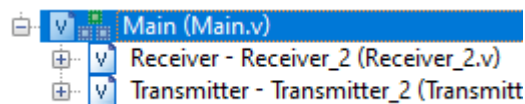
۱. ساختار Main: در این ماژول تنها ۲ ماژول Receiver و Transmitter instantiate

شده است و سیگنال های ورودی و خروجی این دو ماژول ورودی و خروجی گرفته شده

است. به همین دلیل توضیحی برای Design در مورد این ماژول وجود ندارد.

نحوه استفاده از این ماژول دقیقاً یکسان با نحوه استفاده توضیح داده شده در ۲ ماژول

بعدی است.



ورودی و خروجی های این ماژول:

```
// Instantiate the module
Main instance_name (
    .Clk(Clk),
    .Reset(Reset),
    .Start(Start),
    .TranAntenna(TranAntenna),
    .RecAntenna(RecAntenna),
    .TransFrame(TransFrame),
    .RecFrame(RecFrame),
    .Valid_Rec(Valid_Rec),
    .Valid_Tra(Valid_Tra),
    .num_pads_rec(num_pads_rec),
    .num_pads_tra(num_pads_tra)
);
```

مشخصات طراحی:

Device Utilization Summary (estimated values)				[+]
Logic Utilization	Used	Available	Utilization	
Number of Slice Registers	3530	93120	3%	
Number of Slice LUTs	9318	46560	20%	
Number of fully used LUT-FF pairs	2982	9866	30%	
Number of bonded IOBs	19	240	7%	
Number of BUFG/BUFGCTRLs	1	32	3%	
Number of DSP48E1s	6	288	2%	

Device	On-Chip	Power (W)	Used	Available	Utilization (%)
Family	xc6vxcx75t	0.000	1	---	---
Part	xc6vxcx75t	0.000	7803	46560	17
Package	ff484	0.000	9044	---	---
Temp Grade	Commercial	0.000	6	288	2
Process	Typical	0.000	19	240	8
Speed Grade	-2	1.293			
	Total	1.293			

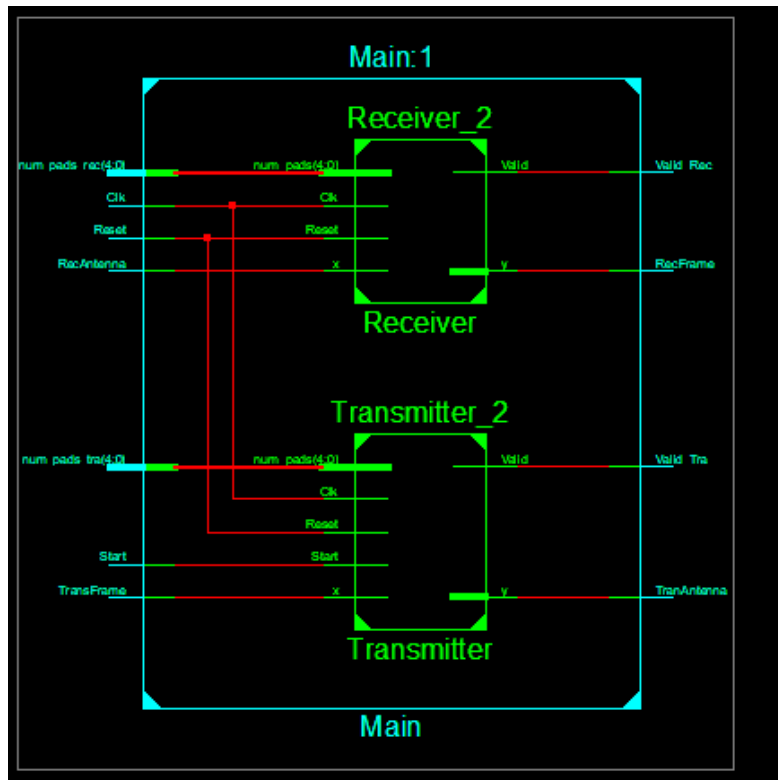
Supply	Summary	Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.000	0.619	0.000	0.619
Vccaux	2.500	0.045	0.000	0.045
Vcco25	2.500	0.001	0.000	0.001
MGTAVcc	1.000	0.303	0.000	0.303
MGTAVtt	1.200	0.213	0.000	0.213

Supply	Power (W)	Total	Dynamic	Quiescent
		1.293	0.000	1.293

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	2.7	81.5	53.5

Environment	
Ambient Temp (C)	50.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	250
Heat Sink	Medium Profile
Custom TSA (C/W)	NA
Board Selection	Medium (10"x10")
# of Board Layers	8 to 11
Custom TJB (C/W)	NA
Board Temperature (C)	NA

:RTL



۲. ساختار Transmitter_2:

در این ماژول بایستی بر روی قسمت های مختلف فریم ورودی پردازش مناسب انجام شود.

فیلد Signal بایستی تنها با ریت ۱/۲ انکود شود.

فیلد DATA بایستی با توجه به Rate و Length موجود در Signal پیاده سازی شود.

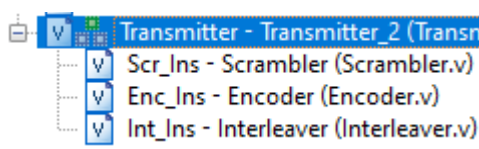
فیلد Preamble بایستی اضافه شود و پیش از بقیه بیت ها ارسال شود. (۱۲ بیت ۱)

البته تنها $\text{rate} = 6 \text{ MBits/s}$ در ویتربی دیکودر و انکودر پیاده سازی شد. اما در فاز چهارم یا همان Integration جوری پیاده سازی شده است که اگر ماژول ویتربی دیکودر و ماژول انکودر برای همه ی rate ها پیاده سازی شده بود، به طور کامل فریم به صورت مناسب انکود و اینترلیو شود و یا دیکود یا دی اینترلیو شود.

ورودی و خروجی های Transmitter_2:

```
// Instantiate the module
Transmitter_2 instance_name (
    .Clk(Clk),
    .Reset(Reset),
    .Start(Start),
    .x(x),
    .y(y),
    .Valid(Valid),
    .num_pads(num_pads)
);
```

در این ماژول برای ارسال یک فریم بایستی به صورت سریال از همان سیکلی که سیگنال Start یک می شود، بیت های ورودی داده شود. در خروجی پس از $152 + N_{cbps} + 2 \text{ PadBits} + \text{Length}$ (توضیح این اعداد در Design وجود دارد.) سایکل به صورت سریالی دیتا معتبر خواهیم داشت و سیگنال Valid در هنگام معتبر بودن خروجی یک خواهد بود.



با پیاده سازی این بخش به مشخصات زیر از لحاظ توان و RTL و میزان استفاده از منابع FPGA رسیدیم:

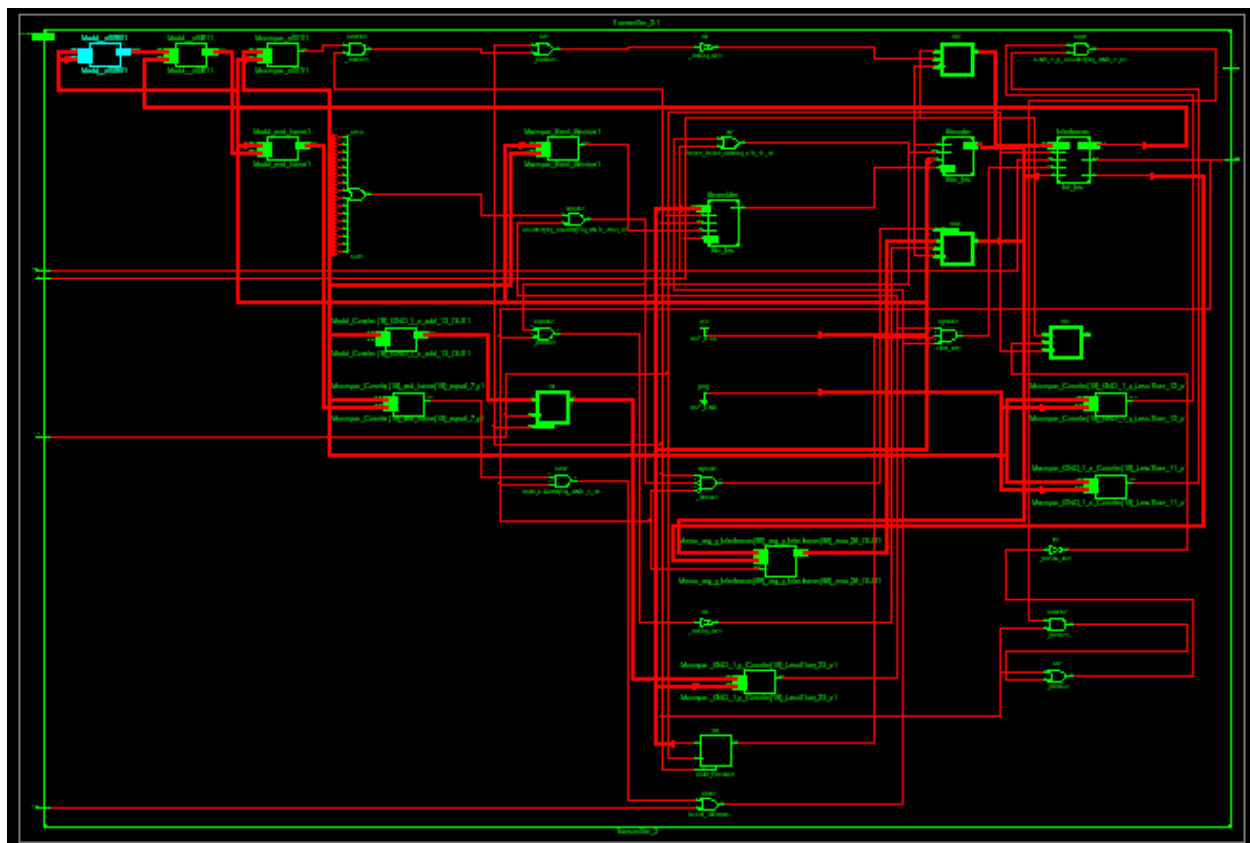
Device		On-Chip	Power (W)	Used	Available	Utilization (%)
Family	Virtex6	Clocks	0.000	1	---	---
Part	xc6vcx75t	Logic	0.000	3523	46560	8
Package	ff484	Signals	0.000	3884	---	---
Temp Grade	Commercial	DSPs	0.000	3	288	1
Process	Typical	IOs	0.000	11	240	5
Speed Grade	-2	Leakage	1.293			
		Total	1.293			

Supply Source	Summary Voltage	Total Current (A)	Dynamic Current (A)	Quiescent Current (A)
Vccint	1.000	0.619	0.000	0.619
Vccaux	2.500	0.045	0.000	0.045
Vcco25	2.500	0.001	0.000	0.001
MGTA Vcc	1.000	0.303	0.000	0.303
MGTA Vt	1.200	0.213	0.000	0.213

Supply Power (W)	Total	Dynamic	Quiescent
	1.293	0.000	1.293

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	2.7	81.5	53.5

Environment	
Ambient Temp (C)	50.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	250
Heat Sink	Medium Profile
Custom TSA (C/W)	NA
Board Selection	Medium (10"x10")
# of Board Layers	8 to 11
Custom TJB (C/W)	NA
Board Temperature (C)	NA



۳. ساختار Receiver_2:

در این ماژول بایستی بر روی قسمت های مختلف فریم ورودی پردازش مناسب انجام شود. با توجه به ورودی preamble را تشخیص دهد و در صورت تشخیص دادن فریم دریافت شده را پردازش کند.

فیلد Signal بایستی تنها با ریت ۱/۲ دیکود شود.

فیلد DATA بایستی با توجه به Rate و Length موجود در Signal دیکود و دی اینترلیو شود.

البته تنها $\text{rate} = 6 \text{ MBits/s}$ در ویتربی دیکودر و انکودر پیاده سازی شد. اما در فاز چهارم یا همان Integration جوری پیاده سازی شده است که اگر ماژول ویتربی دیکودر و ماژول انکودر برای همه ی rate ها پیاده سازی شده بود، به طور کامل فریم به صورت مناسب دیکود و دی اینترلیو شود.

ورودی و خروجی های Receiver_2:

```
// Instantiate the module
Receiver_2 instance_name (
    .Clk(Clk),
    .Reset(Reset),
    .x(x),
    .Y(Y),
    .Valid(Valid),
    .num_pads(num_pads)
);
```

در این ماژول برای دریافت یک فریم و پردازش آن بایستی به صورت سریال از همان سیکلی که سیگنال Start یک می شود، بیت های ورودی داده شود. در خروجی پس از $123 + N_{\text{cbps}} + 4N_{\text{DataBits}}$ (توضیح این اعداد در Design وجود دارد.) سایکل به صورت سریالی دیتا معتبر خواهیم داشت و سیگنال Valid در هنگام معتبر بودن خروجی یک خواهد بود.

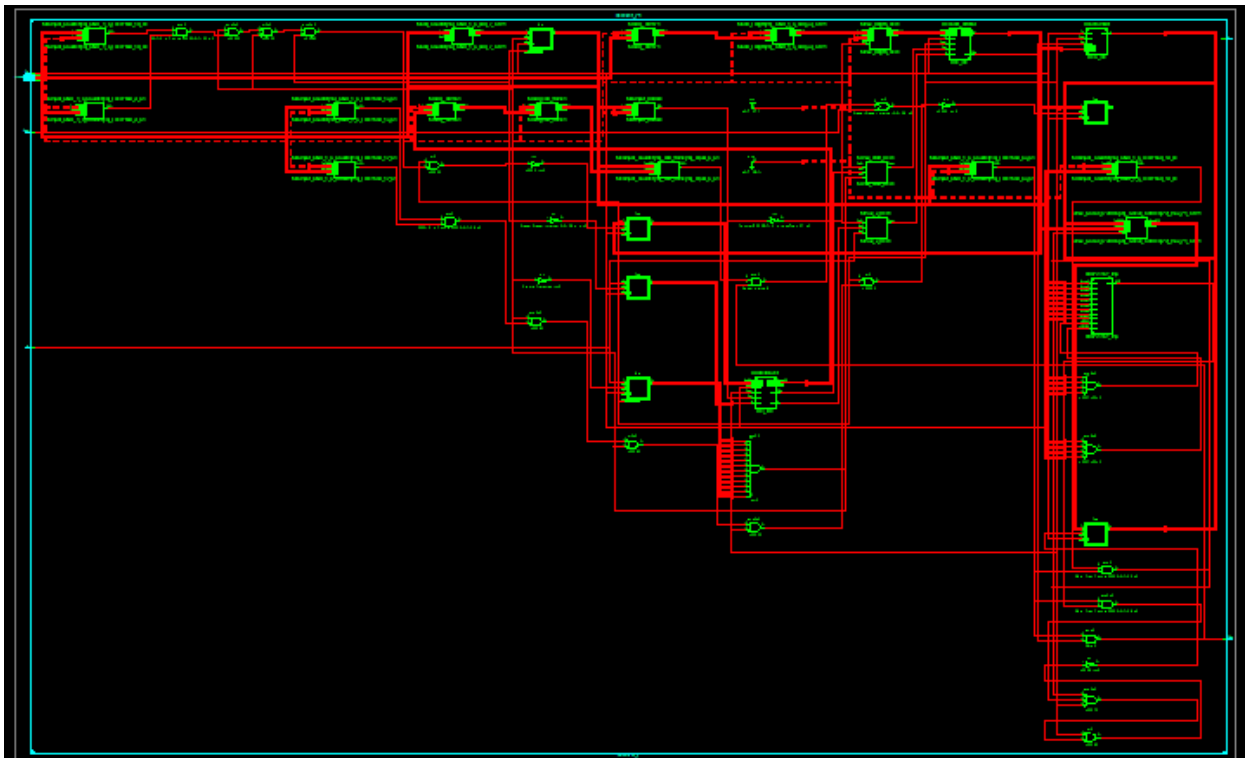


با پیاده سازی این بخش به مشخصات زیر از لحاظ توان و RTL و میزان استفاده از منابع FPGA رسیدم:

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	1816	93120	1%
Number of Slice LUTs	6173	46560	13%
Number of fully used LUT-FF pairs	1332	6657	20%
Number of bonded IOBs	10	240	4%
Number of BUFG/BUFFCTRLs	1	32	3%
Number of DSP48E1s	3	288	1%

Device						Supply Summary			
Family	Virtex6	On-Chip	Power (W)	Used	Available	Utilization (%)	Source	Voltage	Total Current (A)
Part	xc6vcx75t	Clocks	0.000	1	--	--	Vccint	1.000	0.619
Package	ff484	Logic	0.000	4952	46560	11	Vccaux	2.500	0.045
Temp Grade	Commercial	Signals	0.000	5439	--	--	Vcco25	2.500	0.001
Process	Typical	DSPs	0.000	3	288	1	MGTAVcc	1.000	0.303
Speed Grade	-2	IOs	0.000	10	240	4	MGTAVtt	1.200	0.213
		Leakage	1.293						
		Total	1.293						

Environment				Thermal Properties				Supply			
Ambient Temp (C)	50.0	Use custom TJA?	No	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)		Power (W)	Total	Dynamic	Quiescent
Custom TJA (C/W)	NA	Custom TJA (C/W)	NA	2.7	81.5	53.5			1.293	0.000	1.293
Airflow (LFM)	250	Heat Sink	Medium Profile								
Custom TSA (C/W)	NA	Board Selection	Medium (10"x10")								
# of Board Layers	8 to 11	Custom TJB (C/W)	NA								
Board Temperature (C)	NA										



Design:

۱. Transmitter_2: در این ماژول در ورودی به صورت سریال در همان سیکلی که سیگنال

Start یک می شود، ورودی داریم.

مراحل زیر به ترتیب طی می شود: (در زیر شاخه هرکدام تعداد ساینکل هایی که نیاز دارند مشخص شده است)

- پردازش توسط اسکرملر و اسکرمل کردن فیلد دیتا
 - تاخیر صفر ساینکل
- انکود کردن فیلد های دیتا و سیگنال
 - برای انکود کردن یک ساینکل تاخیر ایجاد می شود.
- اینترلیو کردن فیلد دیتا
 - ۴۸ ساینکل تاخیر
- ارسال پری امبل و بیت های پردازش شده در خروجی
 - ۱۲ ساینکل تاخیر برای ارسال پری امبل
 - ۴۸ بیت انکود شده سیگنال
 - ۳۲ بیت انکود شده سرویس
 - ۱۲ بیت انکود شده Tail Bits
 - ۲PadBits بیت انکود شده برای پد بیت
 - ۱۶Length بیت برای انکود شده بخش PSDU

این مراحل انجام می شود. برای اینکه به صورت خودکار پس از دریافت کامل یک فریم و خروجی دادن کامل، دوباره آماده دریافت فریم بعدی شود، با استفاده از اطلاعات تعداد ساینکل های مورد نیاز پایان فریم را تشخیص میدهم و با استفاده از یک ریست اینترنال به حالت ابتدایی بر میگردد.

برای اینکه بتوانم ۱۲ بیت پری امبل و ۴۸ بیت انکود شده سیگنال را بدون وقفه با بیت های دیتا ارسال کنم، نیاز به استفاده از یک صف ۶۰ بیتی داشتم.

۲. Receiver_2: در این ماژول در ورودی به صورت سریال در همان سیکلی که سیگنال Start

یک می شود، ورودی داریم.

مراحل زیر به ترتیب طی می شود: (در زیر شاخه هر کدام تعداد ساینکل هایی که نیاز دارند مشخص شده است)

- تشخیص ۱۲ بیت پری امبل
 - چون پس از تشخیص این بیت ها پردازش داده شروع می شود، این بخش تاخیری ایجاد نمی کند.
- دیکود کردن ۴۸ بیت فیلد سیگنال و استخراج Length و Rate و نگه داری ۲۴ بیت حاصل برای خروجی دادن
 - ۲۸۴۸ ساینکل برای دیکود کردن
 - ۲۴ ساینکل برای خروجی دادن فیلد سیگنال
- دی اینترلیو کردن فیلد دیتا
 - N_{cbps} ساینکل تاخیر برای دی اینترلیو
- دیکود کردن فیلد دیتا
 - Length ۳۲ برای دیکود کردن PSDU
 - ۶۴ ساینکل برای دیکود بخش سرویس
 - ۲۴ ساینکل برای دیکود بخش TailBits
 - ۴PadBits ساینکل برای دیکود بخش پدبیت
- دی اسکرمبل کردن فیلد دیتا
- خروجی دادن بیت های پردازش شده
 - ۳ ساینکل تاخیر برای جا به جایی بین state ها

جهت Resource Sharing از یک ماژول دیکودر برای استفاده شد.

این مراحل انجام می شود. برای اینکه به صورت خودکار پس از دریافت کامل یک فریم و خروجی دادن کامل، دوباره آماده دریافت فریم بعدی شود، با استفاده از اطلاعات تعداد ساینکل

های مورد نیاز پایان فریم را تشخیص میدهم و با استفاده از یک ریست اینترنال به حالت ابتدایی بر میگردد.

برای تشخیص پری امبل نیاز به یک صف ۱۲ بیتی دارم.

برای اینکه بتوانم ۲۴ بیت سیگنال را بدون وقفه با بیت های دیتا ارسال کنم، نیاز به استفاده از یک صف ۲۴ بیتی داشتم. و به صورت یک شیفت رجیستر از آن استفاده کردم.

با توجه به این که دیکود بخش سیگنال تاخیر زیادی ایجاد می کند، جهت از دست ندادن بیت های ورودی نیاز به یک صف ۲+۴۸ بیتی (۲ بیت برای تاخیر تغییر بین state ها) از ورودی ها وجود دارد.

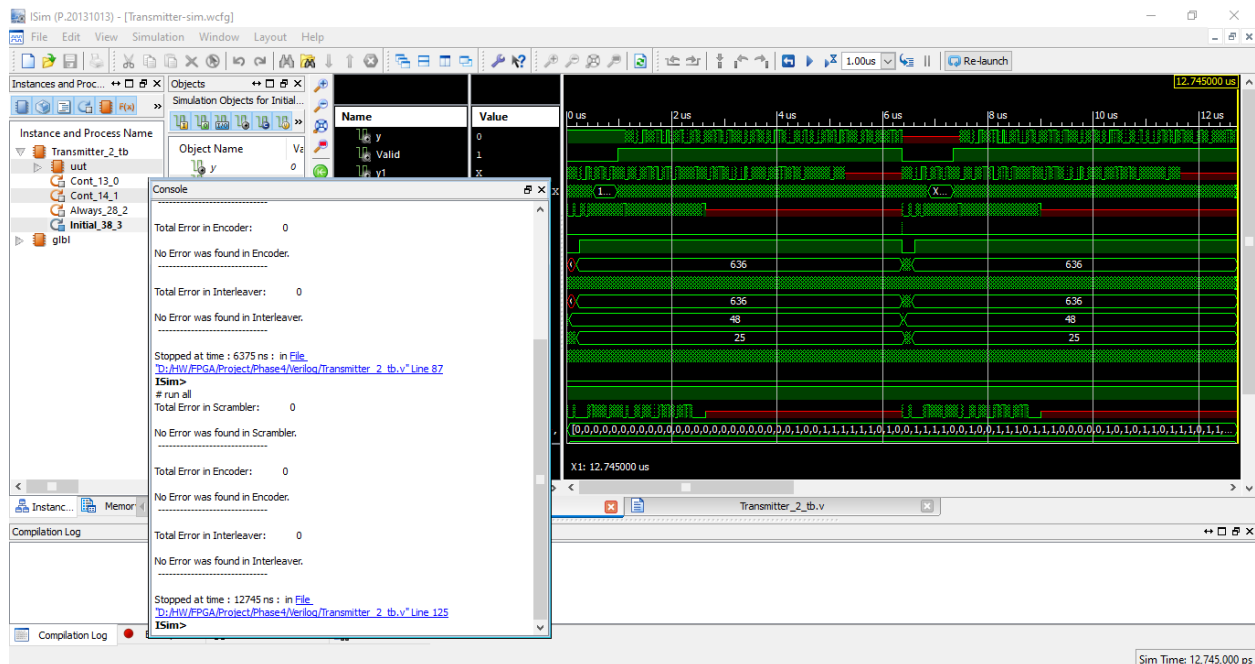
User Documents:

۱. Main: اصول حاکم بر دو ماژول زیر برای این ماژول باید رعایت شود.
۲. Transmitter_2: در این ماژول برای ارسال یک فریم بایستی به صورت سریال از همان سیکلی که سیگنال Start یک می شود، بیت های ورودی داده شود. در خروجی پس از $152 + N_{cbps} + 2 \text{ PadBits} + 8 \text{ Length}$ (توضیح این اعداد در Design وجود دارد.) سایکل به صورت سریالی دیتا معتبر خواهیم داشت و سیگنال Valid در هنگام معتبر بودن خروجی یک خواهد بود.
۳. Receiver_2: در این ماژول برای دریافت یک فریم و پردازش آن بایستی به صورت سریال از همان سیکلی که سیگنال Start یک می شود، بیت های ورودی داده شود. در خروجی پس از $123 + N_{cbps} + 4 \text{ NDataBits}$ (توضیح این اعداد در Design وجود دارد.) سایکل به صورت سریالی دیتا معتبر خواهیم داشت و سیگنال Valid در هنگام معتبر بودن خروجی یک خواهد بود.

Testing:

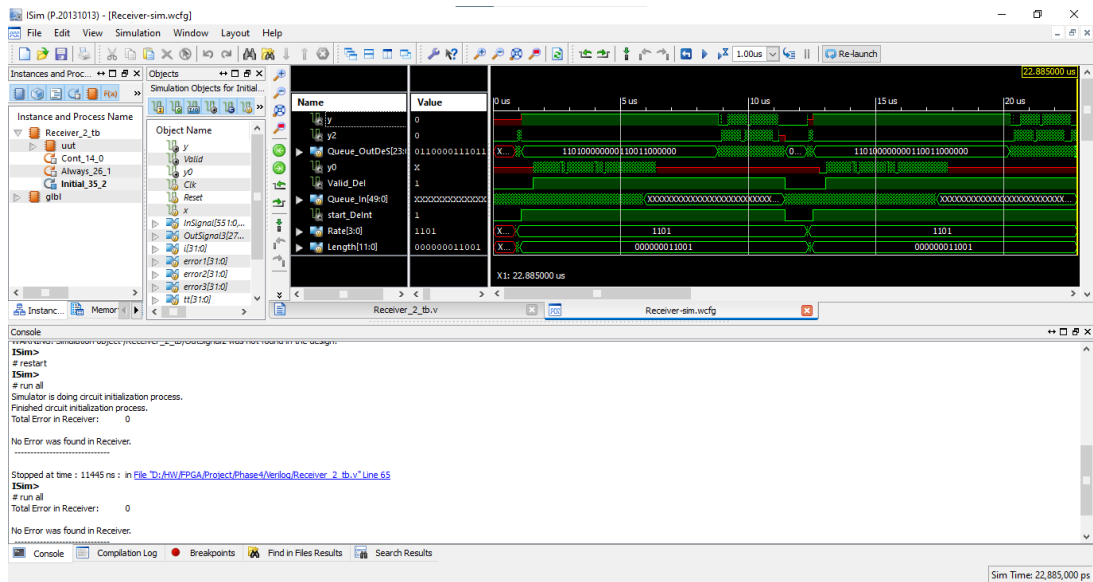
تست بنچ `Transmitter_2` ➔ `Transmitter_2_tb.v` :

در این تست بنچ من فریم `ln.txt` را ارسال میکنم و هر مرحله از انکود و اینترلیو و اسکرمبل را با فایل های `Out_Encoder.txt` و `Out_Interleaver.txt` و `Out_Scramble.txt` که در کد متلب ساخته شده مقایسه میکنم. به تطابق صد در صد رسیدم. (با استفاده از این تست بنچ دو فریم را پشت سر هم ارسال کردم و به نتیجه مطلوب رسیدم.)



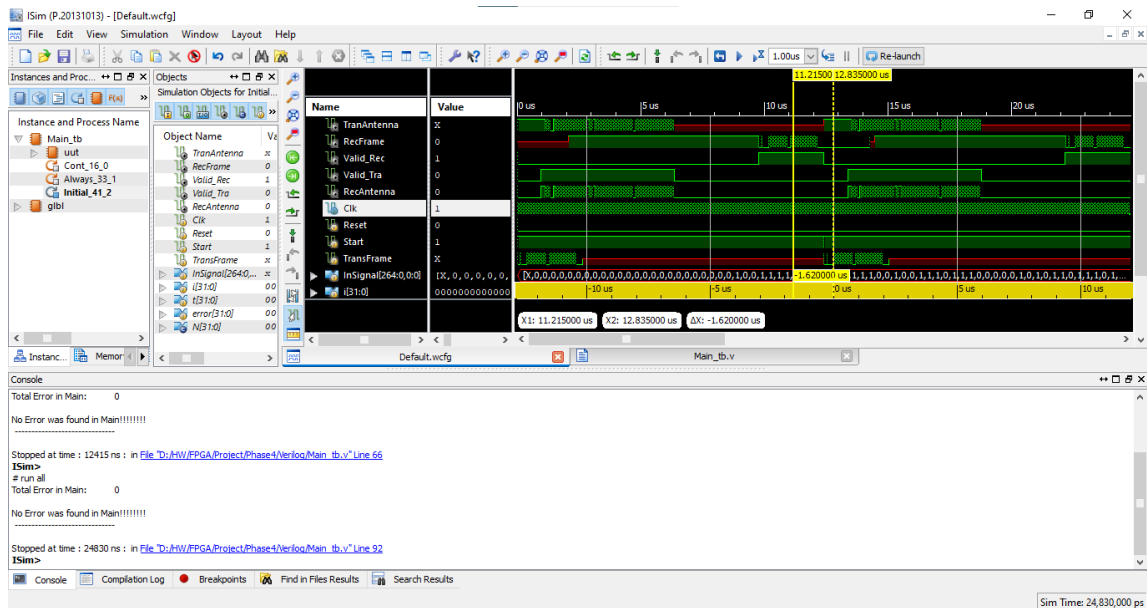
تست بنچ `Receiver_2` ➔ `Receiver_2_tb.v` :

در این تست بنچ من فریم `Out_Interleaver.txt` را ارسال میکنم و با `ln.txt` که در کد متلب ساخته شده مقایسه میکنم. به تطابق صد در صد رسیدم. (با استفاده از این تست بنچ دو فریم را پشت سر هم ارسال کردم و به نتیجه مطلوب رسیدم.)



تست بنچ \rightarrow Main : Main_tb.v

در این تست بنچ من فریم In.txt را ارسال میکنم و در حالیکه RecAntenna را با TranAntenna اتصال کوتاه کردم در خروجی فریم تشخیص داده شده با همان فریم In.txt مقایسه میکنم. به تطابق صد در صد رسیدم.(با استفاده از این تست بنچ دو فریم را پشت سر هم ارسال کردم و به نتیجه مطلوب رسیدم.)



توجه: من از همان مازول های طراحی شده در فاز های قبلی استفاده کردم. جهت برطرف کردن وارنینگ ها واجب شد که طول سیگنال **Length** را در مازول فاز ۲ تغییر دهم:

```
input [14:0]Length; // Length Indicates number of bits to be decoded
```