

به نام خدا

ارسلان فیروزی - ۹۷۱۰۲۲۲۵

پروژه فاز ۳ - FPGA

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

## Specification:

۱. ساختار Interleaver: به صورت شکل زیر است:

در این ماژول با این ترتیب در بلوک های به اندازه  $N_{cbps}$  با توجه به روابط زیر هر بیت در جایگاه  $k$  در ورودی را به جایگاه  $j$  تبدیل می شود:

$$i = (N_{CBPS}/16) (k \bmod 16) + \text{floor}(k/16) \quad k = 0, 1, \dots, N_{CBPS} - 1$$

$$j = s \times \text{floor}(i/s) + (i + N_{CBPS} - \text{floor}(16 \times i/N_{CBPS})) \bmod s \quad i = 0, 1, \dots, N_{CBPS} - 1$$

ورودی و خروجی های Interleaver:

```
// Instantiate the module
Interleaver instance_name (
    .Clk(Clk),
    .Reset(Reset),
    .Start(Start),
    .x(x),
    .Y(Y),
    .Rate(Rate),
    .Valid(Valid),
    .Ncbps(Ncbps)
);
```

در این ماژول برای اینترلیو کردن ورودی، بایستی به صورت سریال از همان سیکلی که سیگنال Start یک می شود، ورودی ها داده شود. خروجی با توجه به Rate تعیین شده، به اندازه مقدار  $N_{cbps}$ ، Latency دارد و سپس با نرخ Throughput یک در خروجی دیتا موثق خواهیم داشت.

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

```

Slice Logic Utilization:
Number of Slice Registers:          588 out of 93120    0%
Number of Slice LUTs:              1246 out of 46560    2%
    Number used as Logic:          1246 out of 46560    2%

Slice Logic Distribution:
Number of LUT Flip Flop pairs used: 1255
    Number with an unused Flip Flop: 667 out of 1255    53%
    Number with an unused LUT:       9 out of 1255      0%
    Number of fully used LUT-FF pairs: 579 out of 1255   46%
    Number of unique control sets:    4

IO Utilization:
Number of IOs:                      19
Number of bonded IOBs:              17 out of 240      7%

Specific Feature Utilization:
Number of BUFG/BUFGCTRLs:          1 out of 32         3%
Number of DSP48E1s:                 3 out of 288        1%

```

<b>Device</b>		<b>On-Chip</b>	<b>Power (W)</b>	<b>Used</b>	<b>Available</b>	<b>Utilization (%)</b>	<b>Supply Summary</b>		<b>Total</b>	<b>Dynamic</b>	<b>Quiescent</b>
Family	Virtex6	Clocks	0.000	1	--	--	<b>Source</b>	<b>Voltage</b>	<b>Current (A)</b>	<b>Current (A)</b>	<b>Current (A)</b>
Part	xc6vcx75t	Logic	0.000	1174	46560	3	Vccint	1.000	0.619	0.000	0.619
Package	ff484	Signals	0.000	1337	--	--	Vccaux	2.500	0.045	0.000	0.045
Temp Grade	Commercial	DSPs	0.000	3	288	1	Vcca25	2.500	0.001	0.000	0.001
Process	Typical	I/Os	0.000	17	240	7	MGTAVcc	1.000	0.303	0.000	0.303
Speed Grade	-2	Leakage	1.293				MGTAVtt	1.200	0.213	0.000	0.213
		<b>Total</b>	<b>1.293</b>								
<b>Environment</b>									<b>Total</b>	<b>Dynamic</b>	<b>Quiescent</b>
Ambient Temp (C)	50.0	<b>Thermal Properties</b>		<b>Effective TJA (C/W)</b>	<b>Max Ambient (C)</b>	<b>Junction Temp (C)</b>	<b>Supply Power (W)</b>		<b>1.293</b>	<b>0.000</b>	<b>1.293</b>
Use custom TJA?	No			<b>2.7</b>	<b>81.5</b>	<b>53.5</b>					
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										

۲. ساختار دی اینترلیور: (DeInterleaver)

ورودی و خروجی ها:

```
// Instantiate the module
DeInterleaver instance_name (
    .Clk(Clk),
    .Reset(Reset),
    .Start(Start),
    .x(x),
    .y(y),
    .Rate(Rate),
    .Valid(Valid),
    .Ncbps(Ncbps)
);
```

در این ماژول برای دی اینترلیو کردن ورودی، بایستی به صورت سریال از همان سیکلی که سیگنال Start یک می شود، ورودی ها داده شود. خروجی با توجه به Rate تعیین شده، به اندازه مقدار  $N_{cbps}$ ، Latency دارد و سپس با نرخ Throughput یک در خروجی دیتا موثق خواهیم داشت.

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

Selected Device : 6vcx75tff484-2

Slice Logic Utilization:				
Number of Slice Registers:	588	out of	93120	0%
Number of Slice LUTs:	1198	out of	46560	2%
Number used as Logic:	1198	out of	46560	2%

Slice Logic Distribution:				
Number of LUT Flip Flop pairs used:	1210			
Number with an unused Flip Flop:	622	out of	1210	51%
Number with an unused LUT:	12	out of	1210	0%
Number of fully used LUT-FF pairs:	576	out of	1210	47%
Number of unique control sets:	3			

```
IO Utilization:
Number of IOs:          19
Number of bonded IOBs:  17 out of 240  7%
```

Specific Feature Utilization:			
Number of BUFG/BUFGCTRLs:	1	out of	32 3%
Number of DSP48E1s:	3	out of	288 1%

[illegible]



موثق از زمان یک شدن Start باید داشته باشد، را دارد. در خروجی پس  $N_{cbps}$  سایکل کلاک خروجی موثق خواهیم داشت.

جهت محاسبه در مدار ترکیبی متناظر با زیر:

$$j = s \times \text{floor}(i/s) + (i + N_{CBPS} - \text{floor}(16 \times i/N_{CBPS})) \bmod s \quad i = 0, 1, \dots, N_{CBPS} - 1$$

نیاز به تقسیم بود که من با استفاده از جدول زیر به طور تقریبی با توجه به جدول زیر در یک عدد ضرب میکنم و شیفِت به راست میدم:

Ncpbs	Denominator	1/Denominator		Equivalent Mult.	Precsie Denominator
48	3	0.333333333	85.33333333	86	2.976744186
96	6	0.166666667	42.66666667	43	5.953488372
192	12	0.083333333	21.33333333	22	11.63636364
288	18	0.055555556	14.22222222	15	17.06666667

Factor	256
--------	-----

۲. DeInterleaver: در این ماژول در ورودی به صورت سریال در همان سیکلی که سیگنال Start یک می شود، ورودی داریم. طراحی پایپ لاین است. به این صورت که با توجه به سیگنال Rate در ورودی  $N_{cbps}$  تعیین می شود، و دقیقاً به تعداد Latency  $N_{cbps}$  داریم و بعد از این تعداد سیکل کلاک در خروجی مقدار موثق خواهیم داشت. برای پایپ لاین کردن پردازش از ۲ رجیستر Out1 و Out2 استفاده کردم. از رجیستر Out1 برای پر کردن در جایگاه هایی که بیت ها قرار است قرار بگیرند استفاده کردم. و از Out2 برای خروجی دادن استفاده کردم. (یک شیفِت رجیستر است که هر بار به تعداد  $N_{cbps}$  سایکل می گذرد پر می شود).

برای پیاده سازی این مدار از یک رجیستر Counter برای state مدار استفاده کردم که تعیین می کند در این سیکل چندمین بیت در فریمی به اندازه  $N_{cbps}$  را داریم بررسی می کنیم. از

طریق مدار ترکیبی جایگاه برعکس جایگشت اول و برعکس جایگشت دوم را بدست می آورم و در همان سیکل در Out1 بیت را در جایگاه جایگشت دوم ذخیره می کنم.

پس از Ncbps سیکل کلاک، مقادیر Out2 را با Out1 جایگزین می کنم.

این ماژول در ورودی سیگنال های Reset, Start, Clk را برای قسمت ترتیبی مدار و ورودی Rate را برای تعیین Ncbps و Nbpsc از طریق LUT و ورودی X که به صورت سریال ورودی های موثق از زمان یک شدن Start باید داشته باشد، را دارد. در خروجی پس Ncbps سیکل کلاک خروجی موثق خواهیم داشت.

جهت محاسبه در مدار ترکیبی متناظر با زیر:

$$i = s \times \text{floor}(j/s) + (j + \text{floor}(16 \times j/N_{CBPS})) \bmod s \quad j = 0, 1, \dots, N_{CBPS} - 1$$

$$k = 16 \times i - (N_{CBPS} - 1) \text{floor}(16 \times i/N_{CBPS}) \quad i = 0, 1, \dots, N_{CBPS} - 1$$

نیاز به تقسیم بود که من با استفاده از جدول زیر به طور تقریبی با توجه به جدول زیر در یک عدد ضرب میکنم و شیفیت به راست میدم:

Ncpbs	Denominator	1/Denominator		Equivalent Mult.	Precsie Denominator
48	3	0.333333333	85.33333333	86	2.976744186
96	6	0.166666667	42.66666667	43	5.953488372
192	12	0.083333333	21.33333333	22	11.63636364
288	18	0.055555556	14.22222222	15	17.06666667

Factor	256
--------	-----

## User Documents:

۱. Interleaver: برای شروع اینترلیو کردن بایستی در یک سیکل سیگنال Start یک شود و در

همان سیکل ورودی به صورت سریال به پورت X داده شود. به اندازه مقدار  $N_{cbps}$  ، Latency دارد و سپس با نرخ Throughput یک در خروجی دیتا موثق خواهیم داشت.

۲. DeInterleaver: برای شروع اینترلیو کردن بایستی در یک سیکل سیگنال Start یک شود و

در همان سیکل ورودی به صورت سریال به پورت X داده شود. به اندازه مقدار  $N_{cbps}$  ، Latency دارد و سپس با نرخ Throughput یک در خروجی دیتا موثق خواهیم داشت.

## Testing:

تست بنچ **Interleaver** ➔ **Interleaver\_tb** :

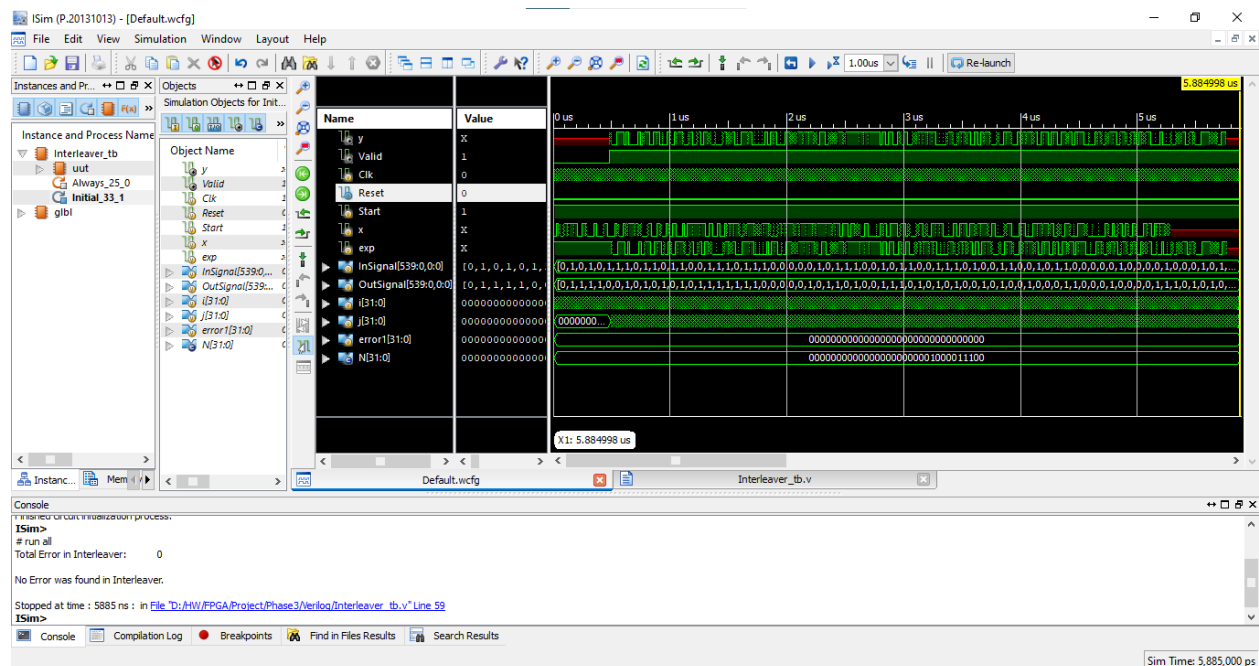
در این تست بنچ من از دو فایل که در کد متلب ساخته شده میخوانم:

۱. شامل کل بیت های یک فریم که به جز Preamble انکود شده است: Out\_Encoder.txt

۲. شامل خروجی اینترلیو در متلب است. Out\_Interleaver.txt

خروجی ماژول را با Out\_Interleaver مقایسه میکنم.

با توجه به عکس زیر به تطابق ۱۰۰٪ رسیدم:



## تست بنچ DeInterleaver → DeInterleaver\_tb :

در این تست بنچ من از دو فایل که در کد متلب ساخته شده میخوانم:

۱. شامل خروجی اینترلیو در متلب است. Out\_Interleaver.txt

۲. شامل کل بیت های یک فریم که به جز Preamble انکود شده است: Out\_Encoder.txt

خروجی ماژول را با Out\_Encoder مقایسه میکنم.

با توجه به عکس زیر به تطابق ۱۰۰٪ رسیدم:

