به نام خدا

تکلیف سری هفتم fpga

محدثه غفوری(9632133)

سوال 1) برای کاهش تاخیر های بین رجیستر ها از پایپ لاین استفاده میکنیم اما با افزایش لایه های پایپ لاین منابع مصرفی (حافظه ها) در fpga زیاد میشود و میتواند کل فضای fpga را پر کند و برای بقیه ی طرح نتوان از fpga استفاده کرد ، از طرفی با افزایش رجیستر ها زمان setup time , hold time فیلیپ فلاپ ها زیاد میشود و دیگر قابل صرف نظر نیستند و میتوانند باعث کاهش فرکانس شوند . همچنین با زیاد کردن لایه ها latency طرح زیاد میشود و برای سیستم هایی که به زود پاسخ دادن به هر ورودی حساس باشد پایپ لاین کردن مضراتی دارد . با اجرای پایپ لاین باید یک توازن بین قسمت های طرح ایجاد کرد تا کلاک ورودی همزمان به تمامی قسمت های طرح برسد تا عملیات ها بتوانند بصورت موازی انجام شوند ، بنابراین برقراری این توازن با زیاد شدن لایه ها برای fpga سخت تر میشود و باعث ایجاد اختلال در خروجی سیستم میشود

برای سیستم هایی که بین اجزای مختلف طرح های پیاده شده در fpga وابستگی وجود دارد و شکستن طرح به تکه های مستقل از هم سخت یا ممکن نباشد امکان پایپ لاین کردن وجود ندارد و درصورت اجرا باعث کاهش throughput میشود زیرا بعلت وابستگی اجزا مختلف طرح نمیتوان مدام داده های جدید را با سرعت به طرح وارد کرد و باید زمانی را برای محاسبات قسمت های وابسته در نظر گرفت . همچنین درصورتی که با اعمال پایپ لاین به طرح به جایی برسیم که تاخیر مربوط به لاجیک ها و مدار های ترکیبی مانع از رسیدن به فرکانس مطلوب شوند ، انگاه پایپ لاین کردن نه تنها موثر نیست بلکه با زیاد شدن رجیستر ها و زمان setup time , hold time انها تاخیر ها زیادتر میشوند و فرکانس کاهش میابد

اما در سیستم هایی که وابستگی زیادی بین تکه های مختلف طرح نیست یا امکان عملیات های موازی روی داده ها فراهم است و تاخیر مسیر های بحرانی زیاد است ، از طریق شکستن مسیر های بحرانی بین رجیستر ها و عملیات combinational بین انها میتوان بوسیله ی پایپ لاین هم گذردهی و هم فرکانس را افزایش داد

سوال 2) ساده ترین شکل نوشتن کد جمع کننده این است که ابتدا ورودی ها را رجیستر کنیم (متغیر های a , b را در a_reg , b_reg قرار دهیم) و سپس out<= a_reg * b_reg را در خروجی داشته باشیم که فرکانس کاری در این حالت 1 / 12.389 ns = 80.71 MHz میباشد

	Met	Constraint	Check	Worst Case Slack	Best Case Achievable	Timing Errors	Timing Score
1	Yes	Autotimespec constraint for clock net clk_BUFGP	SETUP HOLD	 1.313ns	12.389ns	 0	0 0

در این حالت با توجه به یکای گذرهی (bps) میتوان فرمول گذردهی را برابر فرکانس کاری * تعداد بیت خروجی ، در نظر گرفت بنابراین در این حالت 32 * 80.71 = 2582.72 Mbps می باشد

اما بعد از انجام عملیات پایپ لاین فرکانس کاری برابر 1 / 8.977ns = 111.39MHz است

		Constraint	Check	Worst Case Slack	Best Case Achievable	Timing Errors	Timing Score
1	Yes	Autotimespec constraint for clock net clk_BUFGP	SETUP HOLD	 0.677ns	8.977ns	 0	0 0

پس گذردهی برابر با 32 * 111.39 = 3564.48 Mbps است

ب : میزان لاجیک مصرفی در حالت اول یعنی ساده ترین کد مربوط به جمع کننده بصورت زیر است

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	64	1,536	4%	
Number of 4 input LUTs	271	1,536	17%	
Number of occupied Slices	161	768	20%	
Number of Slices containing only related logic	161	161	100%	
Number of Slices containing unrelated logic	0	161	0%	
Total Number of 4 input LUTs	287	1,536	18%	
Number used as logic	271			
Number used as a route-thru	16			
Number of bonded IOBs	65	124	52%	
Number of BUFGMUXs	1	8	12%	
Average Fanout of Non-Clock Nets	1.98			

در حالت دوم و پایپ لاین شده میزان لاجیک مصرفی بصورت زیر است

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	153	1,536	9%	
Number of 4 input LUTs	284	1,536	18%	
Number of occupied Slices	173	768	22%	
Number of Slices containing only related logic	173	173	100%	
Number of Slices containing unrelated logic	0	173	0%	
Total Number of 4 input LUTs	317	1,536	20%	
Number used as logic	212			
Number used as a route-thru	33			
Number used as Shift registers	72			
Number of bonded IOBs	65	124	52%	
Number of BUFGMUXs	1	8	12%	
Average Fanout of Non-Clock Nets	1.80			

بعلت پایپ لاین میزان لاجیک مصرفی در قسمت دوم مشاهده میشود که دو برابر شده است بطور مثال تعداد اسلایس های فیلیپ فلاپ میبینیم بیش از دو برابر شده است بنابرابن با پایپ لاین در ازای مصرف منابع بیشتر فرکانس طرح را افزایش میدهیم

ج : حداکثر فرکانس کاری در ساده ترین حالت برابر 81.5 مگاهرتز است

	Met	Constraint	Check	Worst Case Slack	Best Case Achievable	Timing Errors	Timing Score
1	Yes	TS_clk = PERIOD TIMEGRP "clk" 81.5 MHz HIGH 50%	SETUP HOLD	0.009ns 1.237ns	12.260ns	0 0	0 0

ساده ترین شکل نوشتن کد دارای بیشترین تاخیر در مسیر زیر است

```
--------------------------------------------------------------------------------
Timing constraint: TS_clk = PERIOD TIMEGRP "clk" 81.5 MHz HIGH 50%;
For more information, see Period Analysis in the Timing Closure User Guide (UG612).
 7203422 paths analyzed, 120 endpoints analyzed, 0 failing endpoints
 0 timing errors detected. (0 setup errors, 0 hold errors, 0 component switching limit errors)
 Minimum period is  12.260ns.
--------------------------------------------------------------------------------

Paths for end point Mmult_out_mult0000_0 (SLICE_X16Y24.CIN), 862164 paths
--------------------------------------------------------------------------------
Slack (setup path):     0.009ns (requirement - (data path - clock path skew + uncertainty))
  Source:               y_reg_12 (FF)
  Destination:          Mmult_out_mult0000_0 (FF)
  Requirement:          12.269ns
  Data Path Delay:      12.260ns (Levels of Logic = 14)
  Clock Path Skew:      0.000ns
  Source Clock:         clk_BUFGP rising at 0.000ns
  Destination Clock:    clk_BUFGP rising at 12.269ns
  Clock Uncertainty:    0.000ns

  Maximum Data Path: y_reg_12 to Mmult_out_mult0000_0
    Location             Delay type         Delay(ns)  Physical Resource
                                                       Logical Resource(s)
    -------------------------------------------------  -------------------
```

حداکثر فرکانس کاری در حالت پایپ لاین شده با اعمال constraint برابر MHz111.5 است

	Met	Constraint	Check	Worst Case Slack	Best Case Achievable	Timing Errors	Timing Score
1	Yes	TS_clk = PERIOD TIMEGRP "clk" 111.5 MHz HIGH 50%	SETUP HOLD	0.056ns 0.709ns	8.912ns	0 0	0 0

با اعمال پایپ لاین بیشترین تاخیر در مسیر زیر است

```
================================================================================
Timing constraint: TS_clk = PERIOD TIMEGRP "clk" 111.5 MHz HIGH 50%;
For more information, see Period Analysis in the Timing Closure User Guide (UG612).
 70831 paths analyzed, 281 endpoints analyzed, 0 failing endpoints
 0 timing errors detected. (0 setup errors, 0 hold errors, 0 component switching limit errors)
 Minimum period is   8.912ns.
--------------------------------------------------------------------------------

Paths for end point Mmult_v_mult0000_0 (SLICE_X5Y10.CIN), 5996 paths
--------------------------------------------------------------------------------
Slack (setup path):     0.056ns (requirement - (data path - clock path skew + uncertainty))
  Source:               B_5 (FF)
  Destination:          Mmult_v_mult0000_0 (FF)
  Requirement:          8.968ns
  Data Path Delay:      8.899ns (Levels of Logic = 6)
  Clock Path Skew:      -0.013ns (0.314 - 0.327)
  Source Clock:         clk_BUFGP rising at 0.000ns
  Destination Clock:    clk_BUFGP rising at 8.968ns
  Clock Uncertainty:    0.000ns

  Maximum Data Path: B_5 to Mmult_v_mult0000_0
    Location             Delay type         Delay(ns)  Physical Resource
                                                       Logical Resource(s)
    -------------------------------------------------  -------------------
```

سوال 3) الف : با استفاده از constraint گذاری و چک کردن design summary میبینیم حداکثر فرکانس کاری کد سوال 87.5 MHz است

	Met	Constraint	Check	Worst Case Slack	Best Case Achievable	Timing Errors	Timing Score
1	Yes	TS_Clk = PERIOD TIMEGRP "Clk" 87.5 MHz HIGH 50%	SETUP HOLD	0.003ns 3.096ns	11.425ns	0 0	0 0

ب : گذردهی برابر حاصل ضرب فرکانس کاری در تعداد بیت خروجی است بنابراین کد صورت سوال بدون پایپ لاین چون حداکثر فرکانس 87.5 مگاهرتز دارد بنابراین حداکثر گذردهی برابر 87.5 * 34 = 2975 Mbps میباشد

ج: برای افزایش فرکانس کاری از تکنیک پایپ لاین استفاده میکنیم . با مشاهده ی کد سوال میبینیم خروجی out از یک جمع کننده 4 ورودی و 4 تا ضرب کننده دو ورودی تشکیل شده است که میتوان این عملیات را شکست تا تاخیر مدار را کاهش داد و فرکانس را بیشتر کرد زیرا فرکانس به تاخیر بین رجیستر های طرح بستگی دارد . بنابراین این 4 عمل ضرب را جداگانه انجام داده و در رجیستر های prod1,prod2,prod3,prod4 قرار میدهیم . در مرحله ی بعد بجای استفاده از یک جمع کننده 4 ورودی از دو جمع کننده دو ورودی استفاده میکنیم تا تاخیر لاجیک کمتری متحمل شویم و یک لایه رجیستر های y1,y2 سر راه خروجی جمع کننده ها قرار میدهیم و در انتها مجموع y1,y2 را در out میریزیم . با توجه به design summary مشاهده میشود تاخیر بین رجیستر ها به 5.541ns کاهش یافته است یعنی این طرح بدون constraint گذاری به فرکانس 1 / 5.541ns = 180.42 MHz رسیده است که نسبت به حالت الف که با اعمال constraint به 87.5 رسیده بود افزایش قابل توجهی داشته است

	Met	Constraint	Check	Worst Case Slack	Best Case Achievable	Timing Errors	Timing Score
1	Yes	Autotimespec constraint for clock net clk_BUFGP	SETUP HOLD	1.010ns	5.541ns	0	0 0

د : بله با تعریف constraint و قرار دادن ان روی فرکانس 200 مگاهرتز میتوان فرکانس این طرح را به این مقدار رساند

	Met	Constraint	Check	Worst Case Slack	Best Case Achievable	Timing Errors	Timing Score
1	Yes	TS_clk = PERIOD TIMEGRP "clk" 200 MHz HIGH 50%	SETUP HOLD	0.180ns 1.055ns	4.820ns	0 0	0 0

ه : در صورت ممنوع کردن استفاده از dsp block بدون اعمال شرط timing به فرکانس = 1 / 9.504ns

105.21 MHz میرسیم

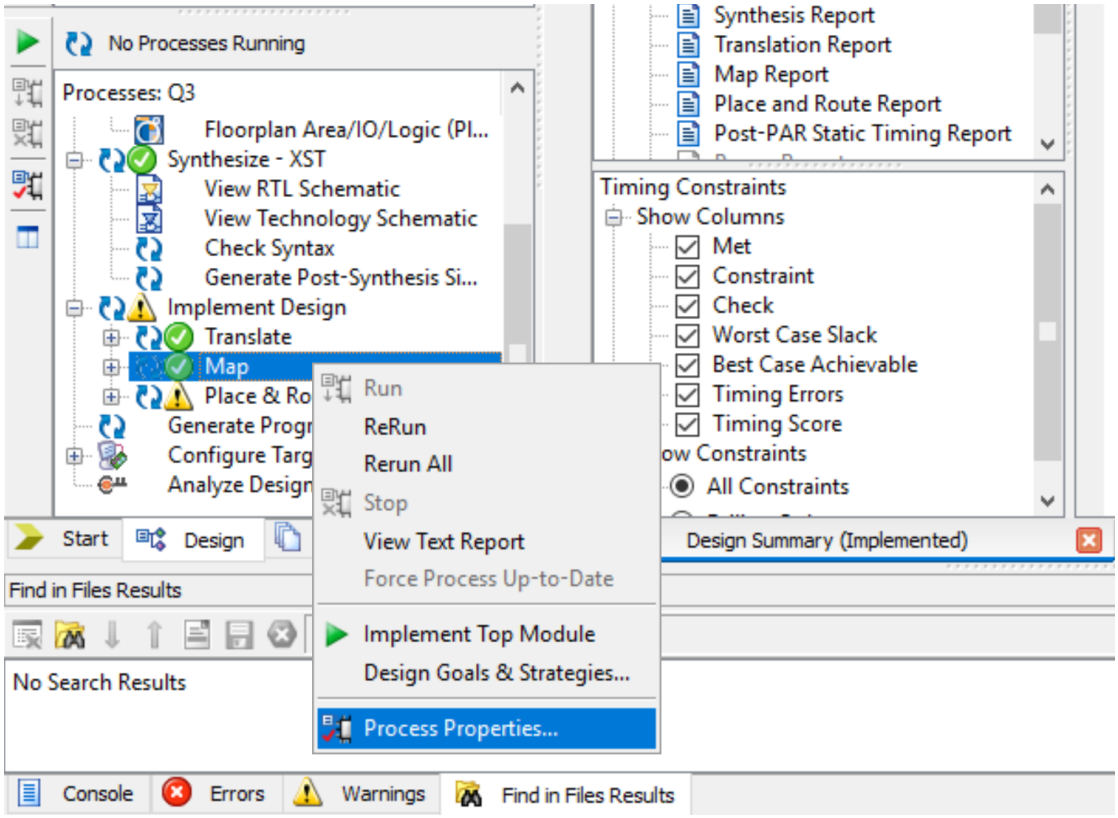
	Met	Constraint	Check	Worst Case Slack	Best Case Achievable	Timing Errors	Timing Score
1	Yes	Autotimespec constraint for clock net clk_BUFGP	SETUP HOLD	 0.580ns	9.504ns	 0	0 0

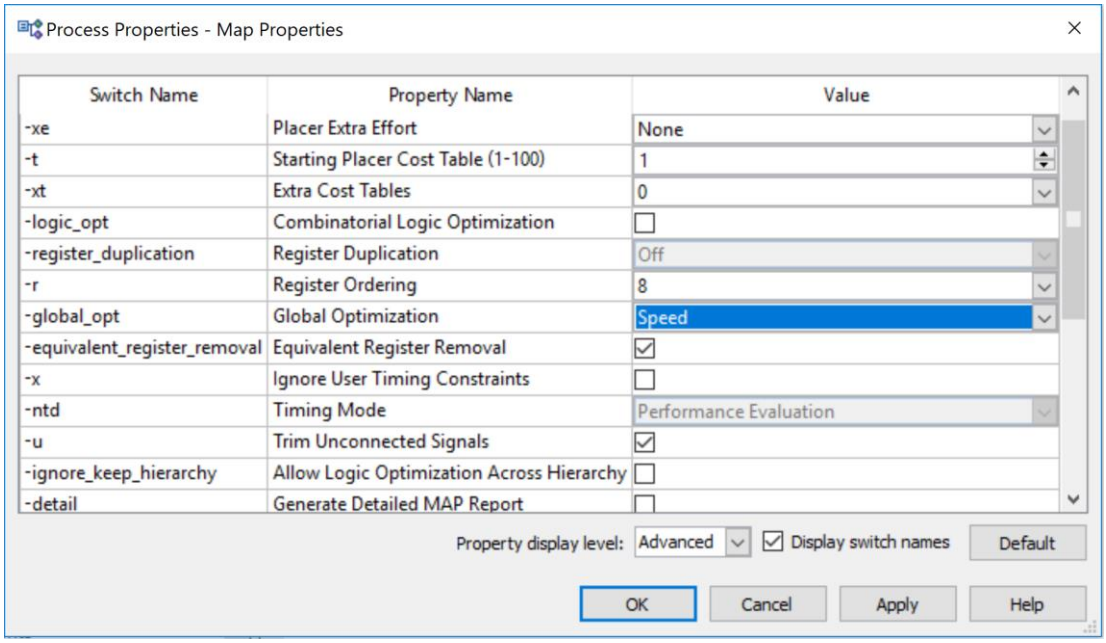
حال با اعمال constraint در بهترین حالت داریم

	Met	Constraint	Check	Worst Case Slack	Best Case Achievable	Timing Errors	Timing Score
1	Yes	TS_clk = PERIOD TIMEGRP "clk" 139.9 MHz HIGH 50%	SETUP HOLD	0.074ns 0.500ns	7.073ns	0 0	0 0

بنابراین به فرکانس 200 مگاهرتز قسمت قبل دست نیافتیم

و : برای رسیدن به فرکانس مورد نظر (200 مگاهرتز) میتوان از تنظیمات مربوط به map بصورت مراحل نشان داده شده در تصویر استفاده کرد (باید optimize _ global را روی هدف speed قرار دهیم)





حال با اعمال constraint این بار میبینیم براحتی به فرکانس 200 مگاهرتز میرسیم

	Met	Constraint	Check	Worst Case Slack	Best Case Achievable	Timing Errors	Timing Score
1	Yes	TS_clk = PERIOD TIMEGRP "clk" 200 MHz HIGH 50%	SETUP HOLD	0.180ns 1.055ns	4.820ns	0 0	0 0

سوال 4) برای ساخت تبدیل کننده 4 بیتی به نمایشگر 7-seg با استفاده از ROM میتوان یک حافظه به طول 7 بیت (کد های متناظر اعداد برای نمایش روی سون سگمنت 7 بیتی هستند) و به عمق 16 خانه در نظر گرفت که در خانه ی 0 کد نمایش عدد صفر قرار دارد ، در خانه ی یک کد نمایش مربوط به عدد یک قرار دارد و ... بنابراین با در نظر گرفتن ورودی in_num کاربر عدد مورد نظر را که میخواهد روی سون سگمنت مشاهده کند به ورودی ماژول میدهد و در خروجی ماژول کد متناظر برای نمایش عدد مورد نظر کاربر دریافت میشود و میتواند عدد را روی سون سگمنت مشاهده کند پس با دریافت ورودی به خانه ای از ROM اشاره میکنیم و مقدار انرا روی خروجی قرار میدهیم که ادرس ان خانه با عدد وارد شده توسط کاربر یکسان باشد بطور مثال کاربر برای نمایش عدد صفر ، صفر را به ورودی ماژول میدهد و ماژول پایه ادرس حافظه را روی صفر تنظیم میکند و مقدار نوشته شده روی این خانه را در خروجی قرار میدهد .

کد های متناظر اعداد از 0 تا 15 را بصورت فایل COE. ساخته و هنگام ساختن ROM این فایل را بعنوان مقدار اولیه روی حافظه قرار میدهیم

کد های متناظر هر عدد را بصورت دسیمال به فایل COE. میدهیم

```
memory_initialization_radix=10;
memory_initialization_vector=
63   ,
48   ,
91   ,
79   ,
```

102 ,

109 ,

125 ,

7 ,

127 ,

111 ,

119 ,

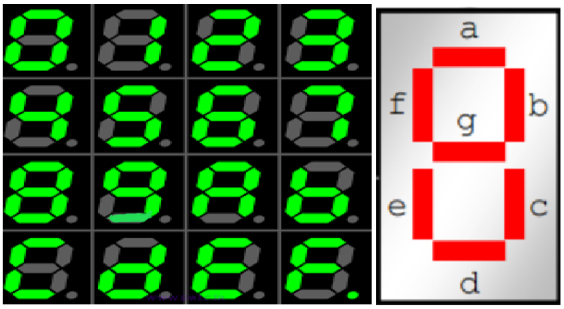
124 ,

57 ,

94 ,

121 ,

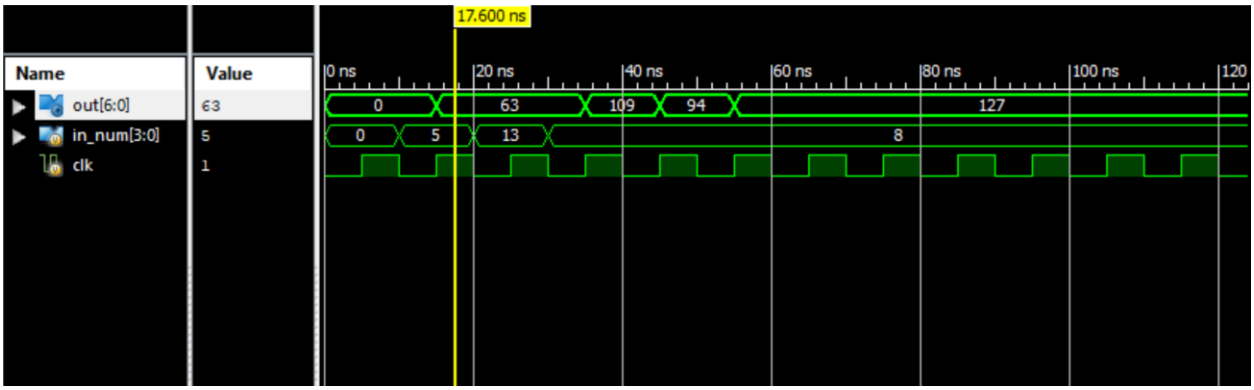
113 ;



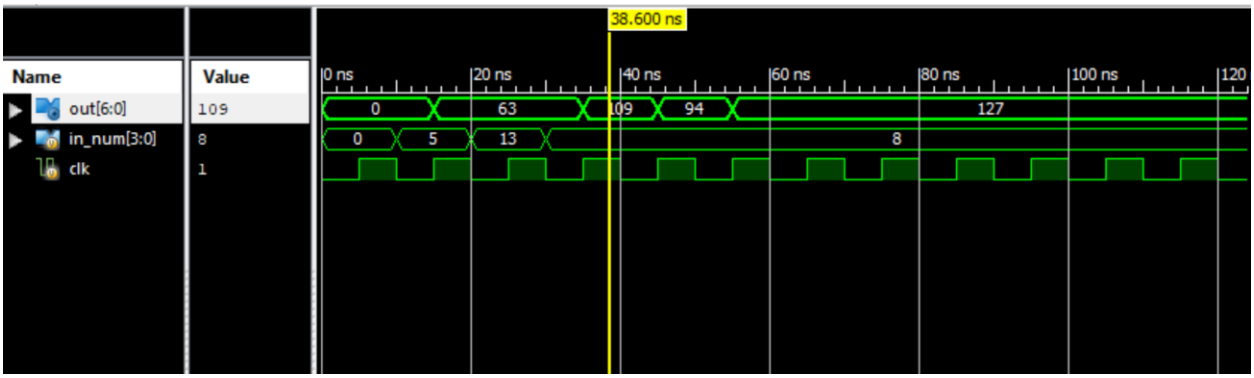
	g	f	e	d	c	b	a
0	0	1	1	1	1	1	1
1	0	1	1	0	0	0	0
2	1	0	1	1	0	1	1
3	1	0	0	1	1	1	1
4	1	1	0	0	1	1	0
5	1	1	0	1	1	0	1
6	1	1	1	1	1	0	1
7	0	0	0	0	1	1	1
8	1	1	1	1	1	1	1
9	1	1	0	1	1	1	1
10=A	1	1	1	0	1	1	1
11=b	1	1	1	1	1	0	0
12=c	0	1	1	1	0	0	1
13=d	1	0	1	1	1	1	0
14=E	1	1	1	1	0	0	1
15=F	1	1	1	0	0	0	1

نتایج شبیه سازی و waveform ها

برای وردی in_num =0 >> با کد متناظر 63



برای وردی in_num =5 >> با کد متناظر 109



برای وردی in_num =8 >> با کد متناظر 127