

به نام خدا



گزارش آزمایش شماره 5

محدثه غفوری (9632133)

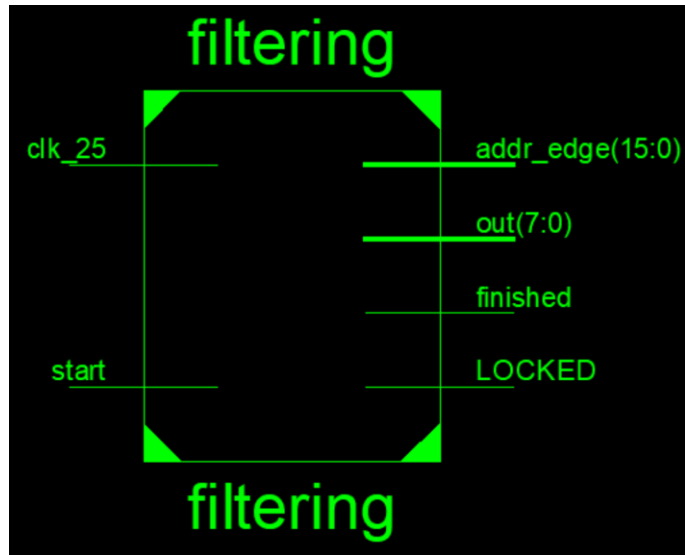
گروه چهارشنبه عصر

در اولین قدم میخواهیم پنجره ای بسازیم که روی تصویر حرکت کند و پردازش های تصویر را روی این پنجره متحرک به مرور روی کل تصویر اعمال کنیم که نام این پنجره را **matrix** که ارایه ی حافظه دو بعدی با 9 عنصر است میگذاریم . برای مقداردهی اولیه به این ماتریس از یک حلقه **for** که با حرکت روی سطر و ستون های خانه های این حافظه، آنرا را با مقدار 0 پر میکند

برای اینکه بتوانیم در یک کلاک 25 مگاهرتز سه پیکسل از تصویر را بخوانیم و در این ماتریس قرار دهیم نیاز داریم تا کلاک ورودی 25 مگ را سه برابر کنیم که بوسیله یک **DCM** با خروجی 75MHz کلاک سه برابر میشود ، حال در هر کلاک اصلی و سه کلاک 75MHz سه پیکسل از تصویر را (مثلا در شروع پیکسل اول از سطر اول که در خانه 0 رام قرار دارد و پیکسل اول از سطر دوم که در خانه 224 رام قرار دارد و پیکسل سوم از سطر سوم که در خانه 448 رام قرار دارد ، را میخوانیم) خوانده و در ستون سمت راست **matrix** قرار میدهیم یعنی بوسیله ی متغیر **row_matrix** که نشان دهنده شماره سطر پنجره است اگر در سطر صفرم بودیم از سطر صفرم تصویر مقدار میگیریم و اگر در حال پر کردن سطر یکم پنجره بودیم از سطر یکم تصویر مقدار میگیریم و هنگام پر کردن سطر ها در یک ستون ثابت میمانیم پس سطر های رام (**row**) را با متغیر **row_matrix** که در هر کلاک برای پر کردن یکی از سطر های یک ستون ثابت ، عوض میشود و متغیر **col** که ستون های تصویر اصلی را نشان میدهد مقدار دهی میکنیم و به پایه ادرس رام که تصویر اصلی در آن قرار دارد میدهیم . توجه شود پس از پر کردن یک ستون از ماتریس یک واحد به **col** اضافه میکنیم تا در ستون بعدی از تصویر سه سطر متوالی را بخواند و در ستون سمت راست پنجره قرار دهد ، همچنین ستون های پنجره پس از سه کلاک 75MHz که سه داده جدید وارد پنجره شد به سمت چپ شیفت پیدا میکنند و بدین طریق میتوان پنجره را با همپوشانی بصورت انلاین روی تصویر حرکت داد و از دوباره خوانی از رام جلوگیری کرد



قسمت قرمز رنگ بخشی از پنجره است
که به سمت فلش روی قسمت ابی رنگ
که بخشی از تصویر است حرکت میکند



در قسمت اول میخواهیم بوسیله دو ماتریس Gx, Gy دستور کار لبه یابی کنیم . برای این منظور باید این دو ماتریس جداگانه روی پنجره ای که روی تصویر حرکت میکند ضرب شوند و حاصل در دو متغیر Ax, Ay که علامت دار میباشند ، ذخیره میشود . توجه شود عملیات محاسبه ی Ax, Ay زمانی معتبر است که پنجره کاملاً شکل گرفته است و در حال حرکت روی تصویر است بنابراین حداقل باید تا رسیدن به انتهای ستون سوم یعنی $col=2$ در هر سطر ، صبر کنیم تا پنجره کاملاً شکل شود و سپس محاسبات را آغاز کنیم و پس از آن باید پنجره به اندازه ی حداقل یک ستون یعنی $row_matrix=2$ جلو برود تا سه دیتا از تصویر بگیرد و در خود بریزد تا اطلاعات پنجره اپدیت شود و بتوانیم با شیفت دادن ستون های پنجره ، محاسبات را انجام دهیم

برای تشخیص تعداد بیت های مورد نیاز برای Ax, Ay دقت میکنیم که با توجه به باز کردن فرمول شان در بدترین حالت اگر درایه های $a11, a21, a31$ برابر 255 باشند و درایه های دیگر صفر باشند Ax بیشترین مقدار مثبت خود را اختیار میکند که برابر $255*4$ است که این مقدار ده بیتی است به طور مشابه اگر درایه های $a13, a23, a33$ همگی بیشترین مقدار مثبت یعنی 255 باشند Ax در منفی ترین حالت خود یعنی $4*255$ - که ده بیتی است قرار میگیرد حال برای آنکه Ax هر دو بازه مثبت و منفی $255*4$ را پوشش دهد از یازده بیت استفاده میکنیم و دقیقاً بطور مشابه برای Ay ، از 11 بیت . از انجایی که در نهایت باید قدر مطلق Ax, Ay با هم جمع شوند و قدر مطلق هر کدام حداکثر $255*4$ است پس abs_Ax, abs_Ay هر کدام ده بیتی و مجموع شان یعنی A یازده بیتی خواهد بود

در ادامه با مقایسه A با یک استانه اگر از آن بیشتر بود لبه است و در تصویری که لبه را نشان میدهد این پیکسل مقدار 0 و در غیر این صورت لبه نبوده و مقدار یک خواهد گرفت

توجه شود که با هر بار حرکت افقی یا عمودی این پنجره روی تصویر در صورتی که سه داده جدید از تصویر خوانده شد و پنجره کل 9 داده مورد نیاز را دریافت کرده بود باید عملیات لبه یابی و ضرب در گرادیان ها انجام شود یعنی هر سه کلاک 75 مگ یکبار پس در واقع عملیات ضرب گرادیان روی پنجره و تولید خروجی با همان کلاک اصلی یعنی 25 مگاهرتز انجام شده است

خروجی locked از DCM به خارج مازول داده میشود تا کاربر از فعال شدن DCM و کلاک 75 مگ با خبر شود و قبل از ان عملیات را شروع نکند

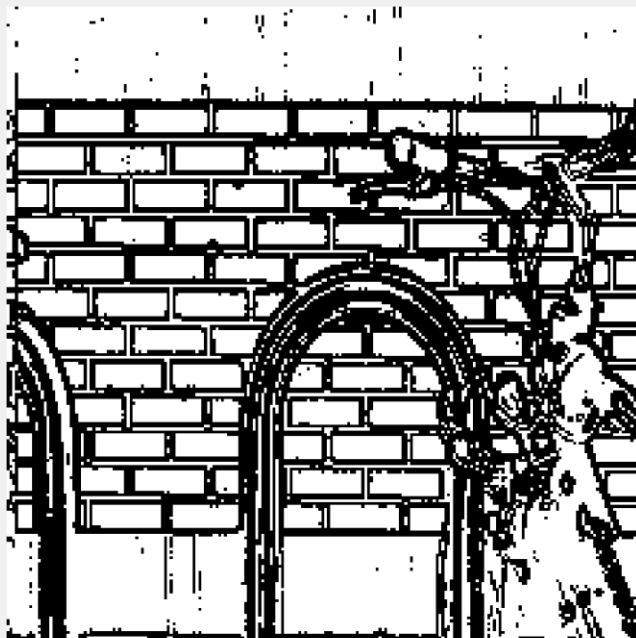
برای اینکه میخواهیم در شبیه سازی در هر کلاک خروجی مازول را روی حافظه ی outfile قرار دهیم و این حافظه نیاز باید در کلاک های مختلفی مقدار مناسبی در خانه های مختلفش قرار گیرند ، متغیر addr_edge را بعنوان خروجی تعیین میکنیم تا با هر بار تولید یک دیتای 0 یا 255 که در نهایت نشان دهنده لبه های تصویر است یک واحد به addr_edge اضافه شود و در شبیه سازی حافظه ی outfile به درستی مقدار دهی شود

همچنین از انجایی که از zeropading استفاده نکردیم ابعاد تصویر به علت حرکت پنجره روی ان و جایگزاری مقدار مناسب با پیکسل وسطی از پنجره ، بجای $224*224$ به $222*222$ کاهش میابد پس در نهایت 49284 پیکسل یا خانه در outfile خواهیم داشت و هنگامی که ادرس addr_edge به 49284 رسید یعنی تمامی این پیکسل ها در تصویر جدید مقدار دهی شدند عملیات خاتمه یافته و finished فعال میشود تا به کاربر اماده بودن تصویر را اطلاع دهد

threshold 50



threshold 100



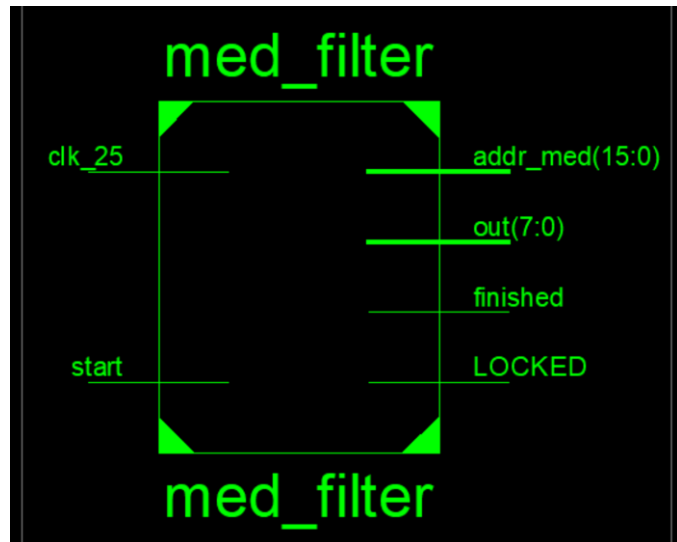
threshold 150



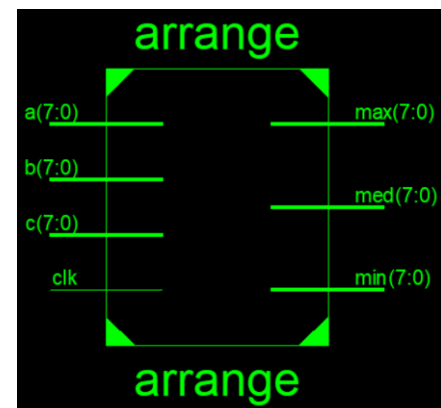
threshold 200



با توجه به تصاویر بالا به نظر میرسد استانه ی پایین تغییرات بیشتر و جزئی تر هم اشکار میکند و حتی تغییراتی که لبه هم نیستند را نمایش میدهد که مناسب نیست ، اما استانه های بیشتر لبه ها را کاملاً تشخیص داده اند و تغییراتی که مربوط به لبه ها نبوده است اشکار نشده اند اما اگر استانه زیادتر شود مانند 200 لبه ها کمتر و ناواضح تر تشخیص داده میشوند و اطلاعات تصویر حذف میشود به نظر استانه ی 150 مناسب ترین حالت است



در قسمت دوم آزمایش می‌خواهیم نویز نمک و فلفل را از تصویر حذف کنیم . عملیات تشکیل پنجره و حرکت آن بر روی تصویر دقیقاً مانند قسمت قبل است اما اینجا از یک زیر ماژول برای میانه یابی استفاده می‌کنیم



زیر ماژول بالا سه عدد از ورودی دریافت میکند و بزرگترین آنها را در **max** و کمترین شان را در **min** و متغیری که مقداری ما بین ماکس و مینیمم دارد را در **med** که همان **median** است میریزد در واقع سه عدد **3!** جایشت دارند و از شش شرط برای تشخیص ارزش هر کدام و خروجی ها استفاده می‌کنیم

در ماژول اصلی هنگامی که پنجره به طور کامل تشکیل شد و دیتا های آن معتبر بود در ابتدا سه سطر از پنجره یعنی **matrix** را به ماژول **arrange** بطور جداگانه می‌دهیم تا هر سطر مرتب شود و عناصر هر سطر از کوچک به بزرگ قرار گیرند . هنگامی که سه سطر از پنجره مرتب شد ، در ستون اول **min** و در ستون دوم **med** و در ستون سوم **max** های نامرتب قرار دارند پس باید ستون ها هم مینیمم های مرتب و مدین و ماکسیمم های مرتبی داشته باشند بنابراین خروجی سطر های مرتب شده را دوباره به ماژول **arrange** می‌دهیم تا ستون ها را هم مرتب کند

حال سطر ها و ستون ها مرتب میشوند و باید از بین عناصر قطر فرعی با ماژول **arrange** میانه را پیدا کنیم که این میانه کل پنجره است و بعنوان خروجی به بیرون از ماژول داده میشود

از انجایی که با هر بار جلو رفتن پنجره روی تصویر اطلاعات ان تغییر میکند و عملیات میانه گیری مجددا انجام میشود برای ذخیره اطلاعات در جای مناسب از ادرس **adrr_med** به عنوان خروجی ماژول استفاده میکنیم تا در خانه های مناسب **outfile** خروجی قرار داده شود

توجه شود که به علت نبود عملیات **zeropading** تصویر نهایی ابعاد $222*222$ خواهد داشت و هنگامی که کل **49284** خانه از تصویر با دیتا های مناسب پر شد **finished** فعال میشود

دقت شود که طبق گفته سوال مرتب شدن هر سطر یا ستون در یک کلاک اصلی انجام میشود بنابراین ماژول **arrange** با کلاک **25** مگ کار میکند که چون در طرح از **DCM** استفاده شده است نمیتوان مستقیما این **25** مگ را از ورودی ماژول اصلی گرفت و باید دو خروجی از **DCM** بگیریم یکی همان کلاک ورودی یعنی **25** مگاهرتز و یکی کلاک **75** مگاهرتز

تصویر نهایی بصورت زیر است



در این قسمت میخواهیم حداکثر فرکانس طرح را مشخص کنیم

برای قسمت لبه یابی با توجه به تصویر زیر حداکثر فرکانس طرح برابر 137.15MHz است

	Constraint	Check	Worst Case Slack	Best Case Achievable	Timing Errors	Timing Score
1 Yes	Autotimespec constraint for clock net clk_75	SETUP HOLD	0.439ns	7.291ns	0	0

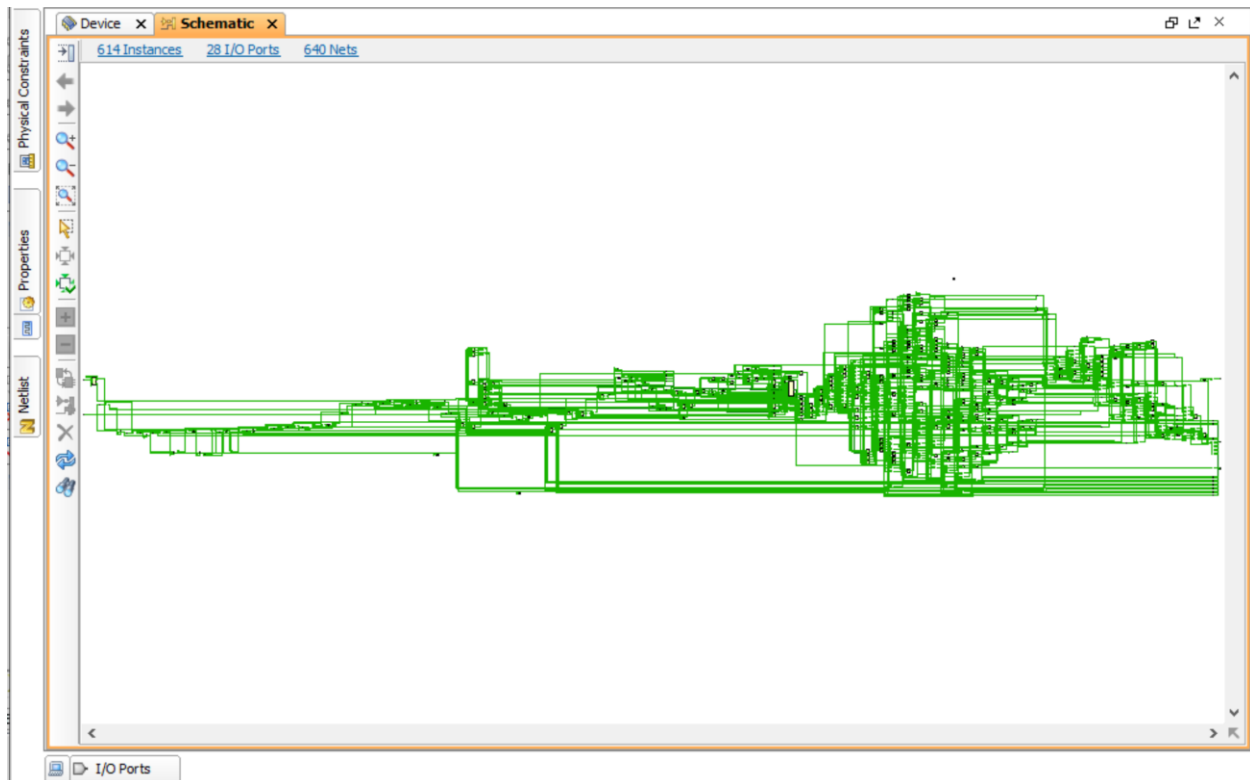
منابع مصرف شده بصورت زیر است

Device Utilization Summary					
Slice Logic Utilization	Used	Available	Utilization	Note(s)	
Number of Slice Registers	182	11,440	1%		
Number used as Flip Flops	165				
Number used as Latches	0				
Number used as Latch-thrus	0				
Number used as AND/OR logics	17				
Number of Slice LUTs	211	5,720	3%		
Number used as logic	204	5,720	3%		
Number using O6 output only	127				
Number using O5 output only	28				
Number using O5 and O6	49				
Number used as ROM	0				
Number used as Memory	2	1,440	1%		
Number used as Dual Port RAM	0				
Number used as Single Port RAM	0				
Number used as Shift Register	2				
Number using O6 output only	2				
Number using O5 output only	0				
Number using O5 and O6	0				
Number used exclusively as route-thrus	5				
Number with same-slice register load	0				
Number with same-slice carry load	5				
Number with other load	0				
Number of occupied Slices	69	1,430	4%		

Number of occupied Slices	69	1,430	4%
Number of MUXCYs used	108	2,860	3%
Number of LUT Flip Flop pairs used	238		
Number with an unused Flip Flop	62	238	26%
Number with an unused LUT	27	238	11%
Number of fully used LUT-FF pairs	149	238	62%
Number of unique control sets	7		
Number of slice register sites lost to control set restrictions	9	11,440	1%
Number of bonded IOBs	28	102	27%
Number of RAMB16BWERs	24	32	75%
Number of RAMB8BWERs	1	64	1%
Number of BUFIO2/BUFIO2_2CLKs	1	32	3%
Number used as BUFIO2s	1		
Number used as BUFIO2_2CLKs	0		
Number of BUFIO2FB/BUFIO2FB_2CLKs	1	32	3%
Number used as BUFIO2FBs	1		
Number used as BUFIO2FB_2CLKs	0		
Number of BUFG/BUFGMUXs	2	16	12%
Number used as BUFGs	2		
Number used as BUFGMUX	0		
Number of DCM/DCM_CLKGENs	1	4	25%
Number used as DCMs	1		
Number used as DCM_CLKGENs	0		
Number of ILOGIC2/ISERDES2s	0	200	0%
Number of IODELAY2/IODRP2/IODRP2_MCBs	0	200	0%

Number used as BUFIO2FBs	1		
Number used as BUFIO2FB_2CLKs	0		
Number of BUFG/BUFGMUXs	2	16	12%
Number used as BUFGs	2		
Number used as BUFGMUX	0		
Number of DCM/DCM_CLKGENs	1	4	25%
Number used as DCMs	1		
Number used as DCM_CLKGENs	0		
Number of ILOGIC2/ISERDES2s	0	200	0%
Number of IODELAY2/IODRP2/IODRP2_MCBs	0	200	0%
Number of OLOGIC2/OSERDES2s	0	200	0%
Number of BSCANs	0	4	0%
Number of BUFHs	0	128	0%
Number of BUFPLLs	0	8	0%
Number of BUFPLL_MCBs	0	4	0%
Number of DSP48A1s	1	16	6%
Number of ICAPs	0	1	0%
Number of MCBs	0	2	0%
Number of PCILOGICSEs	0	2	0%
Number of PLL_ADVs	0	2	0%
Number of PMVs	0	1	0%
Number of STARTUPs	0	1	0%
Number of SUSPEND_SYNCS	0	1	0%
Average Fanout of Non-Clock Nets	3.74		

تصویر ساختار قسمت لبه یابی بصورت زیر است



همچنین برای قسمت نویز نمک و فلفل با توجه به تصویر زیر حداکثر فرکانس کاری برابر 144.07MHz است

		Constraint	Check	Worst Case Slack	Best Case Achievable	Timing Errors	Timing Score
1	Yes	Autotimespec constraint for clock net clk_75	SETUP HOLD	0.415ns	6.829ns	0	0
2	Yes	Autotimespec constraint for clock net clk25	SETUP HOLD	0.624ns	6.941ns	0	0

منابع مصرفی قسمت نويز نمک فلفل

Device Utilization Summary					
Slice Logic Utilization	Used	Available	Utilization	Note(s)	
Number of Slice Registers	292	11,440	2%		
Number used as Flip Flops	292				
Number used as Latches	0				
Number used as Latch-thrus	0				
Number used as AND/OR logics	0				
Number of Slice LUTs	516	5,720	9%		
Number used as logic	509	5,720	8%		
Number using O6 output only	438				
Number using O5 output only	26				
Number using O5 and O6	45				
Number used as ROM	0				
Number used as Memory	2	1,440	1%		
Number used as Dual Port RAM	0				
Number used as Single Port RAM	0				
Number used as Shift Register	2				
Number using O6 output only	2				
Number using O5 output only	0				
Number using O5 and O6	0				
Number used exclusively as route-thrus	5				
Number with same-slice register load	2				
Number with same-slice carry load	3				
Number with other load	0				
Number of occupied Slices	179	1,430	12%		

Number of occupied Slices	179	1,430	12%	
Number of MUXC's used	32	2,860	1%	
Number of LUT Flip Flop pairs used	557			
Number with an unused Flip Flop	280	557	50%	
Number with an unused LUT	41	557	7%	
Number of fully used LUT-FF pairs	236	557	42%	
Number of unique control sets	17			
Number of slice register sites lost to control set restrictions	10	11,440	1%	
Number of bonded IOBs	28	102	27%	
Number of RAMB16BWERS	24	32	75%	
Number of RAMB8BWERS	1	64	1%	
Number of BUFIO2/BUFIO2_2CLKs	1	32	3%	
Number used as BUFIO2s	1			
Number used as BUFIO2_2CLKs	0			
Number of BUFIO2FB/BUFIO2FB_2CLKs	1	32	3%	
Number used as BUFIO2FBs	1			
Number used as BUFIO2FB_2CLKs	0			
Number of BUFG/BUFGMUXs	2	16	12%	
Number used as BUFGs	2			
Number used as BUFGMUX	0			
Number of DCM/DCM_CLKGENs	1	4	25%	
Number used as DCMs	1			
Number used as DCM_CLKGENs	0			
Number of ILOGIC2/ISERDES2s	0	200	0%	
Number of IODELAY2/IODRP2/IODRP2_MCBs	0	200	0%	

Number used as BUFIO2FBs	1		
Number used as BUFIO2FB_2CLKs	0		
Number of BUFG/BUFGMUXs	2	16	12%
Number used as BUFGs	2		
Number used as BUFGMUX	0		
Number of DCM/DCM_CLKGENs	1	4	25%
Number used as DCMs	1		
Number used as DCM_CLKGENs	0		
Number of ILOGIC2/ISERDES2s	0	200	0%
Number of IODELAY2/IODRP2/IODRP2_MCBs	0	200	0%
Number of OLOGIC2/OSERDES2s	0	200	0%
Number of BSCANs	0	4	0%
Number of BUFHs	0	128	0%
Number of BUFPLLs	0	8	0%
Number of BUFPLL_MCBs	0	4	0%
Number of DSP48A1s	1	16	6%
Number of ICAPs	0	1	0%
Number of MCBs	0	2	0%
Number of PCILOGICSEs	0	2	0%
Number of PLL_ADVs	0	2	0%
Number of PMVs	0	1	0%
Number of STARTUPs	0	1	0%
Number of SUSPEND_SYNCs	0	1	0%
Average Fanout of Non-Clock Nets	4.42		

ساختار طرح قسمت نويز نمک فلفل

