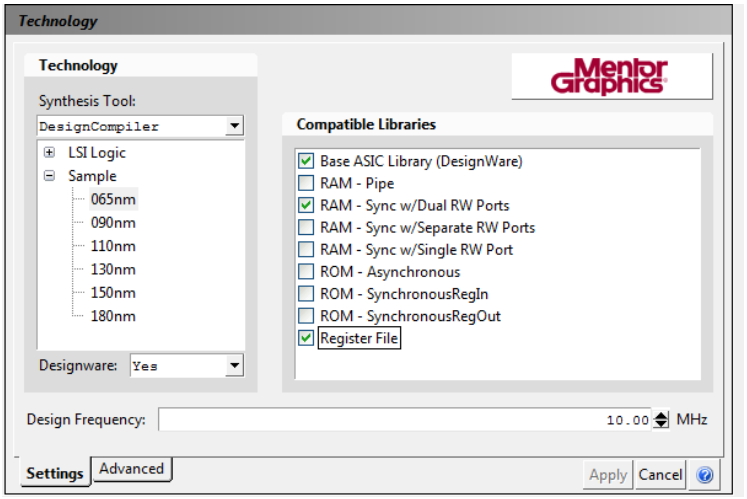
علی مومنی / 810100215 / پروژه اول طراحی در سطح سیستم

گام اول)

تغییرات پارامترها در کد سطح بالا انجام شد و تنظیمات در نرم افزار کتپلت صورت گرفت.

```
#ifndef MATRIX_MULT_H_
#define MATRIX_MULT_H_

#include "ac_int.h"

#define M 13
#define N 10
#define K 12
#define INPUT_BITS 16
#define OUTPUT_BITS (2*INPUT_BITS+ac::log2_ceil<K>::val)

#define INPUT_WIDTH INPUT_BITS

void matrix_mult(ac_int<INPUT_WIDTH,false> input_matrix1[N][K], ac_int<INPUT_WIDTH,false> input_matrix2[K][M],
                 ac_int<OUTPUT_BITS,false> *output, ac_int<1,false> *output_valid, ac_int<ac::log2_ceil<N*M>::val,false> *addr);

#endif /* MATRIX_MULT_H_ */
```

تغییر پارامتر ها در کد سطح بالا



گام دوم)

فرکانس کاری مدار در گام قبل روی 10 مگاهرتز قرار داده شد و نتایج استخراج شد :

Solution	Latency Cy...	Latency Time	Throughpu...	Throughpu...	Slack	Total Area
matrix_mult.v1	3378	337800.00	3391	339100.00	95.01	5384.46

همانطور که مشاهده میشود مقادیر به دست می اید که شرح انها به اینگونه است :

Latency cycle: تعداد کلاک های زده شده تا به دست امدن اولین خروجی پس از اعمال اولین ورودی.

Latency time: مقدار زمان طی شد بر حسب نانو ثانیه تا به دست امدن اولین خروجی پس از اعمال اولین ورودی . در اینجا چون فرکانس کاری مدار روی 10 مگا هرتز تنظیم شده است پس T برای با 100 نانو ثانیه میشود که این مقدار ضربدر latency cycle به ما مقدار latency time را میدهد.

Thoughput cycle: تعداد کلاک هایی زده تا اتمام کامل کار مدار است و نشان میدهد بعد از این میتوان دوباره از ان استفاده کرد.

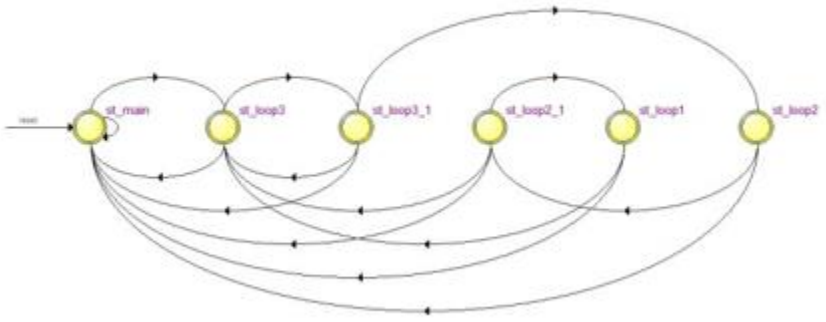
Throuput time: دوباره مانند latency time است که مقدار ان هم به صورت مشابه به دست می اید .

Slack : میزان تایم آزاد یا استفاده نشده هر کلاک را نشان میدهد .

بعد از سنتز در کوارتس مقدار فرکانس ماکسیمم به دست می اید :

	Fmax	Restricted Fmax	Clock Name	Note
1	193.57 MHz	193.57 MHz	clk	

نتیجه سنتز :



گام سوم)

matrix_mult.v14	3378	17464.26	3391	17531.47	0.18	5384.46

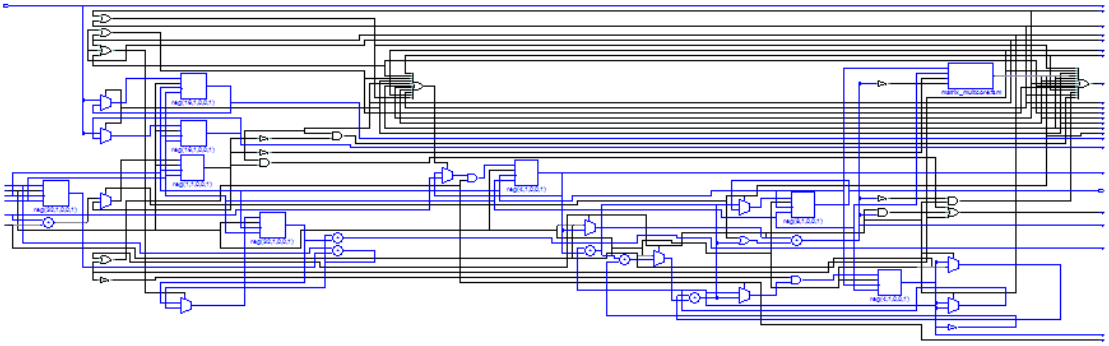
همانطور که مشاهده میشود با قرار دادن فرکانس در به عددی که کوارتس داده است میزان slack ما به شدت کم میشود که این نشان میدهد که ما باید تایمینگ دقیق تری داشته باشیم تا مشکلی پیش نیاید پس میتوان نتیجه گرفت با زیاد کردن فرکانس کلاک ها سریعتر زده شده و میزان ازادی هر کلاک کمتر میشود و از انطرف چون T کمتر شده throughput time نیز مقداری کمتری میگیرد و مدار در حالت کلی سریعتر میشود .

گام چهارم)

آنرول کردن با استفاده از نرم افزار :

matrix_mult.v2	1819	181900.00	1832	183200.00	89.55	18382.98

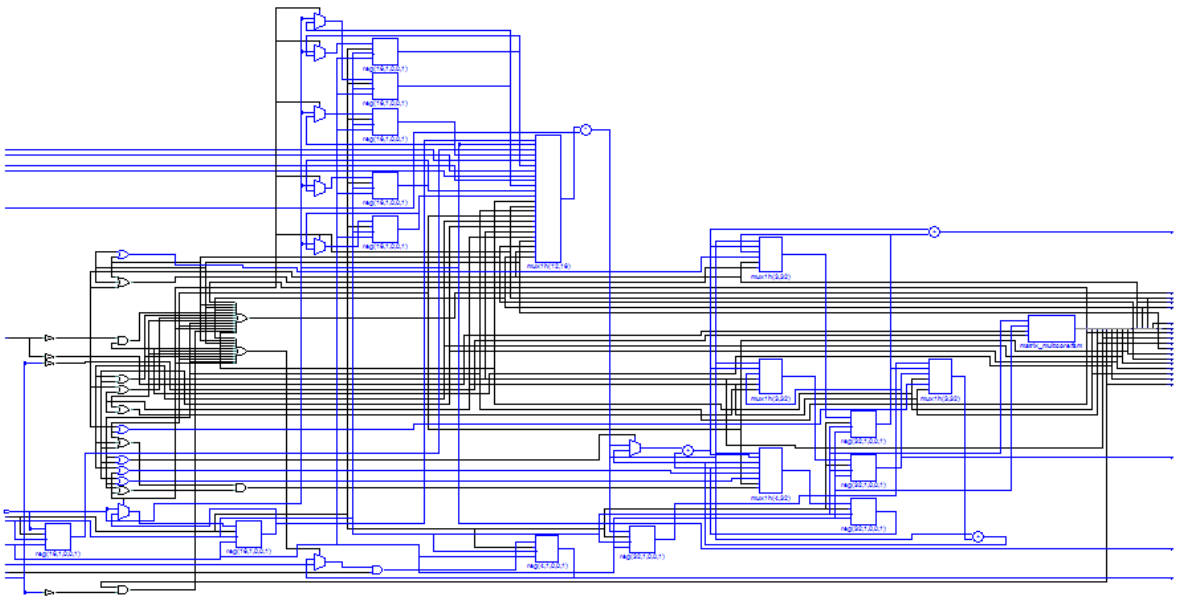
دیتاپس:



آنرول کردن به صورت دستی :

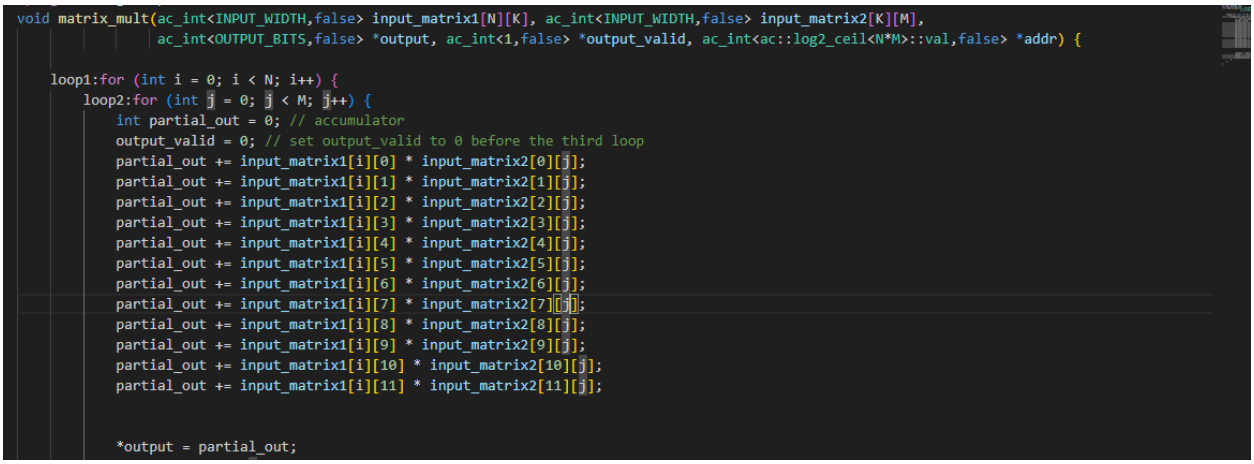
matrix_mult.v3	1819	181900.00	1832	183200.00	89.46	18890.86

دیتا پس :



کد سطح بالا :

کاری که برای آنرول کردن دستی باید انجام داد این است مه loop را پاک کنیم و به تعداد تکرار های لوپ کد را کپی کنیم :



```
void matrix_mult(ac_int<INPUT_WIDTH,false> input_matrix1[N][K], ac_int<INPUT_WIDTH,false> input_matrix2[K][M],
                 ac_int<OUTPUT_BITS,false> *output, ac_int<1,false> *output_valid, ac_int<ac::log2_ceil<N*M>::val,false> *addr) {

  loop1:for (int i = 0; i < N; i++) {
    loop2:for (int j = 0; j < M; j++) {
      int partial_out = 0; // accumulator
      output_valid = 0; // set output_valid to 0 before the third loop
      partial_out += input_matrix1[i][0] * input_matrix2[0][j];
      partial_out += input_matrix1[i][1] * input_matrix2[1][j];
      partial_out += input_matrix1[i][2] * input_matrix2[2][j];
      partial_out += input_matrix1[i][3] * input_matrix2[3][j];
      partial_out += input_matrix1[i][4] * input_matrix2[4][j];
      partial_out += input_matrix1[i][5] * input_matrix2[5][j];
      partial_out += input_matrix1[i][6] * input_matrix2[6][j];
      partial_out += input_matrix1[i][7] * input_matrix2[7][j];
      partial_out += input_matrix1[i][8] * input_matrix2[8][j];
      partial_out += input_matrix1[i][9] * input_matrix2[9][j];
      partial_out += input_matrix1[i][10] * input_matrix2[10][j];
      partial_out += input_matrix1[i][11] * input_matrix2[11][j];


      *output = partial_out;
```

همانطور که مشاهده میشود در هر دو حالت مقادیر latency cycle و thoughput cycle مقدار یکسانی میشود و نسبت به حالتی که آنرول نکرده بودیم هر دو مقدار کمتر میشود ولی در رابطه به slack و area برعکس است با آنرول کردن area به شدت افزایش یافته است . در حالت دستی area افزایش بیشتری دارد .

نتایج به دست امده از بخش reports:

Area Scores

	Post-Scheduling		Post-DP & FSM		Post-Assignment	
Total Area Score:	17062.5		43553.1		18890.9	
Total Reg:	11269.5	(66%)	11453.6	(26%)	11453.6	(61%)
DataPath:	17062.5	(100%)	43385.1	(100%)	18722.9	(99%)
MUX:	0.0		2345.9	(5%)	3217.3	(17%)
FUNC:	5793.0	(34%)	29358.8	(68%)	3795.5	(20%)
LOGIC:	0.0		389.8	(1%)	419.4	(2%)
BUFFER:	0.0		0.0		0.0	
MEM:	0.0		0.0		0.0	
ROM:	0.0		0.0		0.0	
REG:	11269.5	(66%)	11290.6	(26%)	11290.6	(60%)
FSM:	0.0		168.0	(0%)	168.0	(1%)
FSM-REG:	0.0		163.0	(97%)	163.0	(97%)
FSM-COMB:	0.0		5.0	(3%)	5.0	(3%)

Area Scores

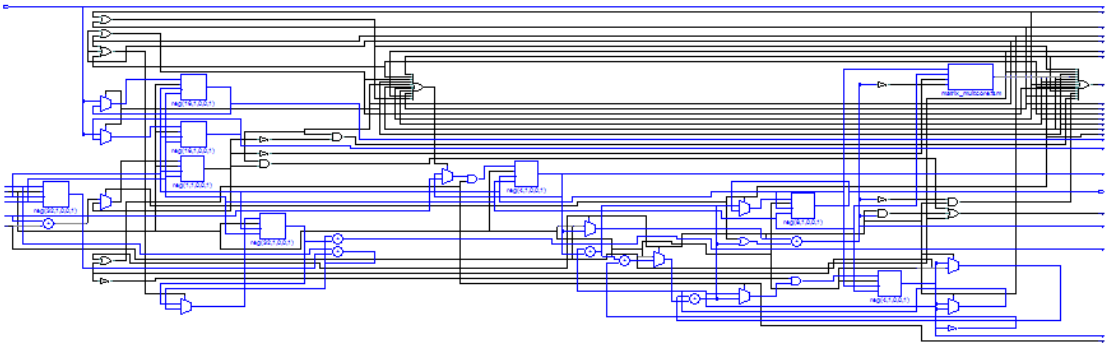
	Post-Scheduling		Post-DP & FSM		Post-Assignment	
Total Area Score:	17318.1		43645.9		18383.0	
Total Reg:	11460.1	(66%)	11453.6	(26%)	11453.6	(62%)
DataPath:	17318.1	(100%)	43477.9	(100%)	18215.0	(99%)
MUX:	0.0		2412.4	(6%)	2674.7	(15%)
FUNC:	5858.0	(34%)	29379.5	(68%)	3836.7	(21%)
LOGIC:	0.0		395.4	(1%)	413.0	(2%)
BUFFER:	0.0		0.0		0.0	
MEM:	0.0		0.0		0.0	
ROM:	0.0		0.0		0.0	
REG:	11460.1	(66%)	11290.6	(26%)	11290.6	(62%)
FSM:	0.0		168.0	(0%)	168.0	(1%)
FSM-REG:	0.0		163.0	(97%)	163.0	(97%)
FSM-COMB:	0.0		5.0	(3%)	5.0	(3%)

گام پنجم)

آنرول کردن لوپ سوم :

matrix_mult.v4	1819	181900.00	1832	183200.00	89.55	18382.98

دیتاپس:



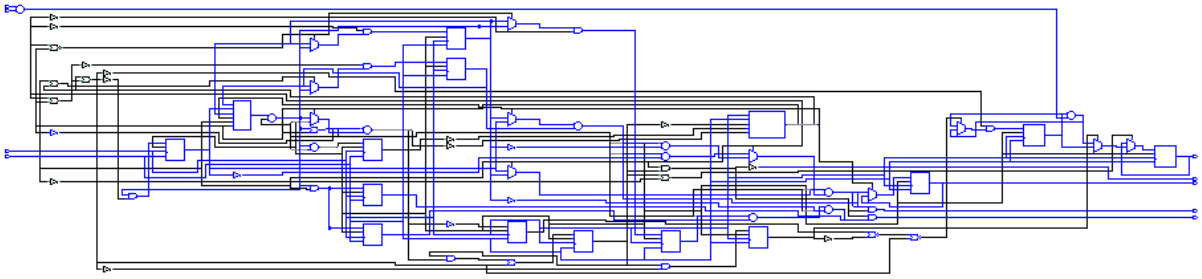
با اینجام اینکار مقدار فضای اشغالی به شدت زیاد میشود ولی latency cycle کم میشود .

پایپ لاین لوپ سوم :

با پایپ لاین کردن این لوپ مقدار latency و throughput به شدت کاهش یافته و مقدار area بع میزان کمی زیاد میشود (به دلیل اضافه شدن رجیسترها)

matrix_mult.v5	1948	194800.00	1961	196100.00	94.96	5723.98

دیتاپس:

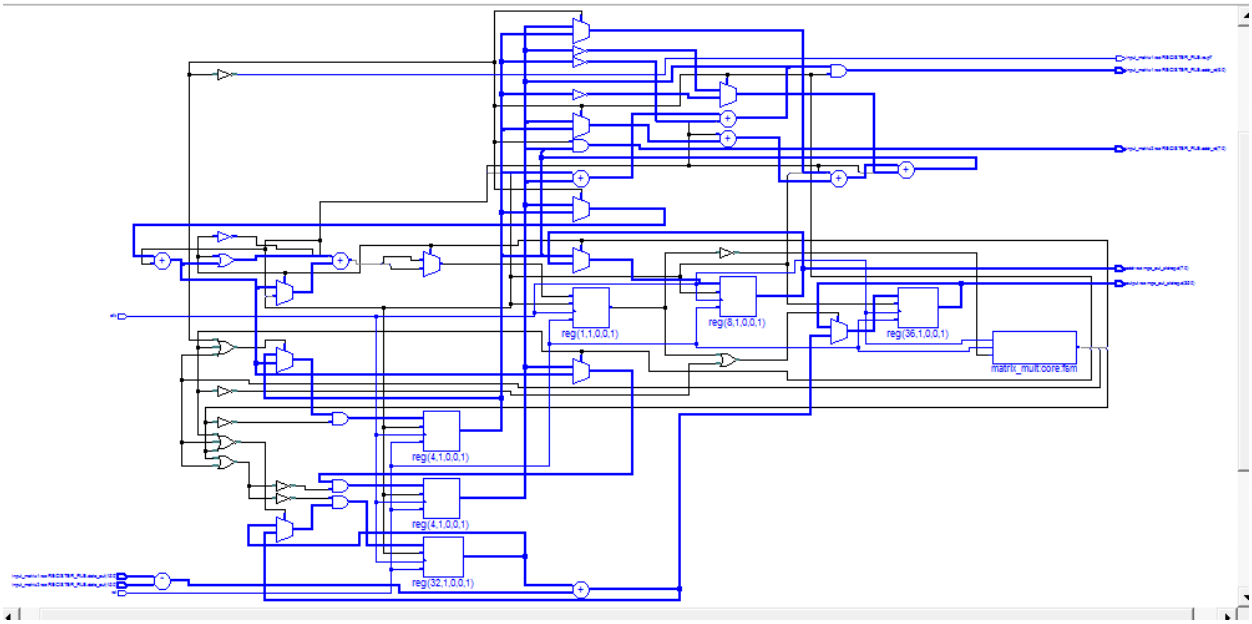


انرول کردن لوپ دوم :

آنرول کردن این لوپ باعث میشود که ضرب ها به صورت همزمان انجام شود یعنی به صورت همزمان حلقه های سوم ران میشنود که این باعث میشود latency به صورت قابل توجهی کمتر شود همینطور نرم افزار با استفاده از قابلیت resource sharing مقدار areaرا نیز کمتر میکند :

matrix_mult.v6	258	25800.00	261	26100.00	95.01	5110.94

دیتاپس:

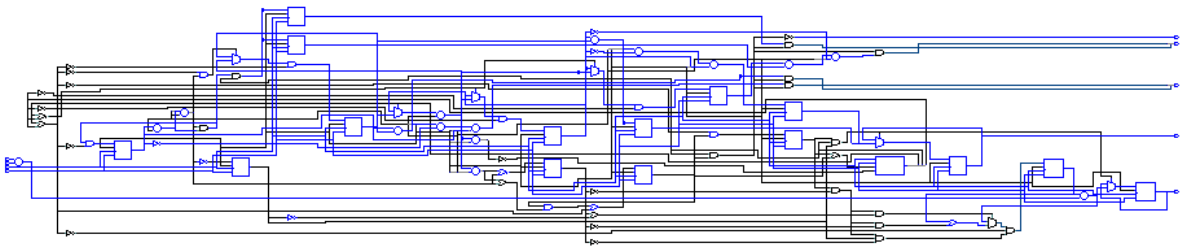


پایپ لاین لوپ دوم :

در اینجا latency کم میشود ولی نه به اندازه انرول کردن و همینطور area افزایش میابد :

matrix_mult.v7	1578	157800.00	1591	159100.00	95.00	5912.99

دیتاپس:

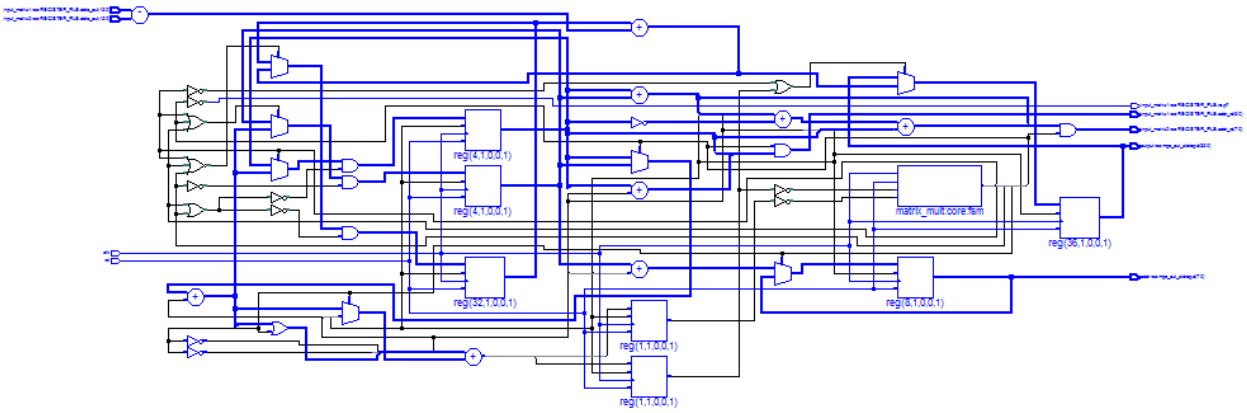


آنرول کردن لوپ اول :

مانند آنرول کردن حلقه دوم به دلیل همزمان انجام شدن حلقه دوم و سوم latencyبه صورت قابل توجهی کاهش پیدا میکند و دوباره نرم افزار با استفاده از resource sharing مقدار فضا را هم کمتر میکند .

matrix_mult.v8	336	33600.00	339	33900.00	95.01	5088.48

دیتاپس:

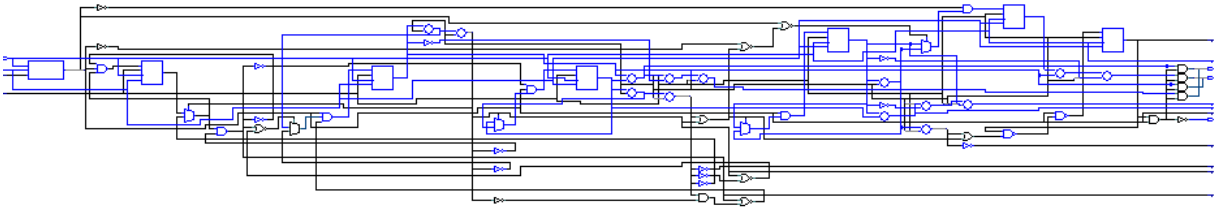


پایپلاین لوپ اول :

مانند دو لوپ دیگر باعث کاهش latency میشود ولی چون در سطح بالاتر است کاهش نیز بیشتر میشود.

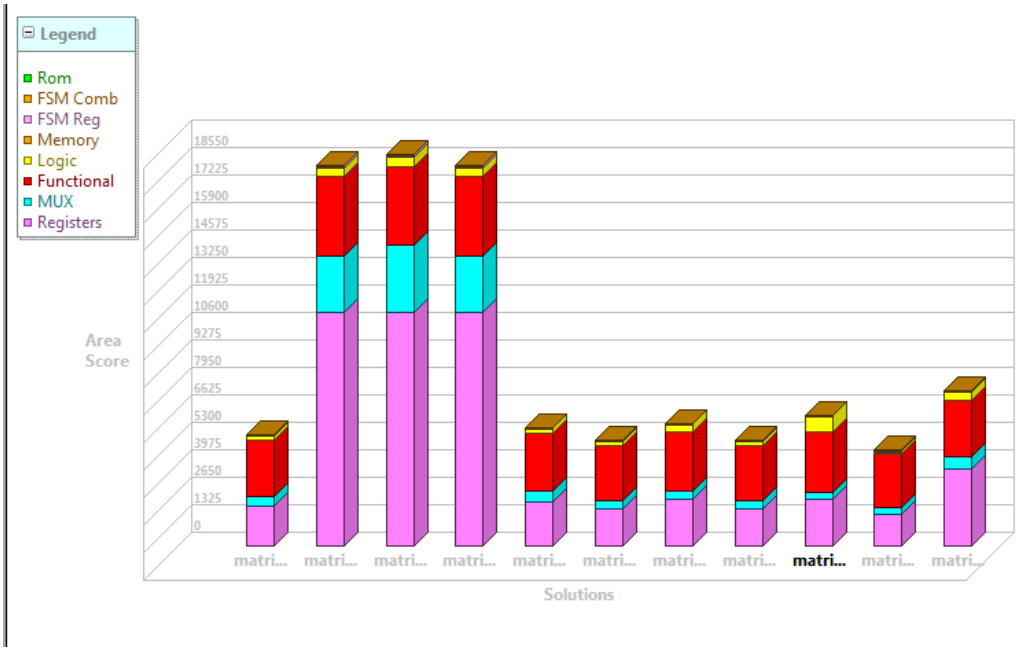
matrix_mult.v9	1560	156000.00	1563	156300.00	94.92	6324.56

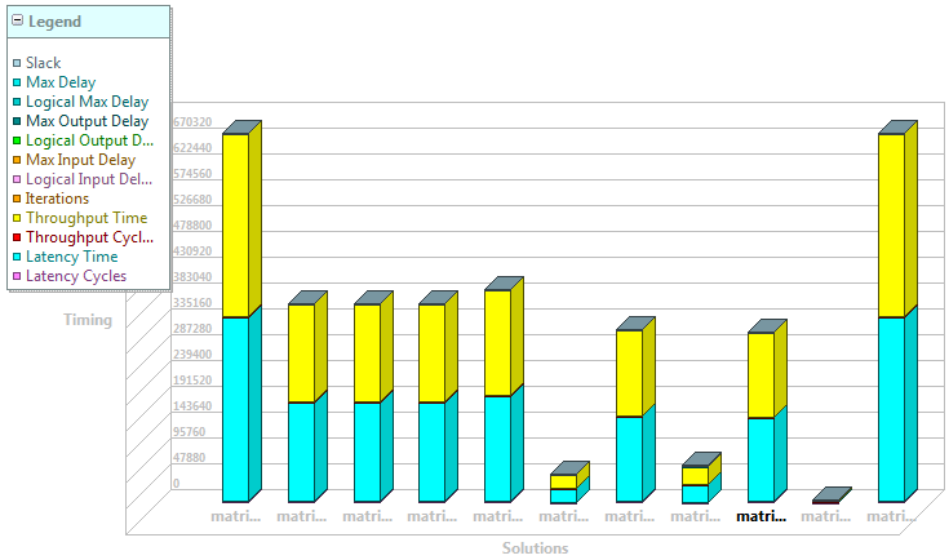
دیتا پس :



باید توجه داشت که در تمامی حالات به دلیل ثابت بودن critical path در مدار مقدار slack تغییر زیادی نمیکند:

نمودار ها :





در حالت کلی پایپلاین کردن در این طراحی به اندازه انرول کردن خوب نیست برای همین گزینه مناسبی نمیباشد همچنین انرول کردن لوپ سوم به دلیل افزایش زیاد area کارامد نیست .

بهترین گزینه در این طراحی آنرول کردن لوپ اول یا دوم است که هردو latencyرا به شدت کم کرده همینطوری area کمتری نیست دارند ولی در کل عملکرد نزدیک به هم دارند ولی انرول لوپ اول میتواند گزینه بهتری باشد .

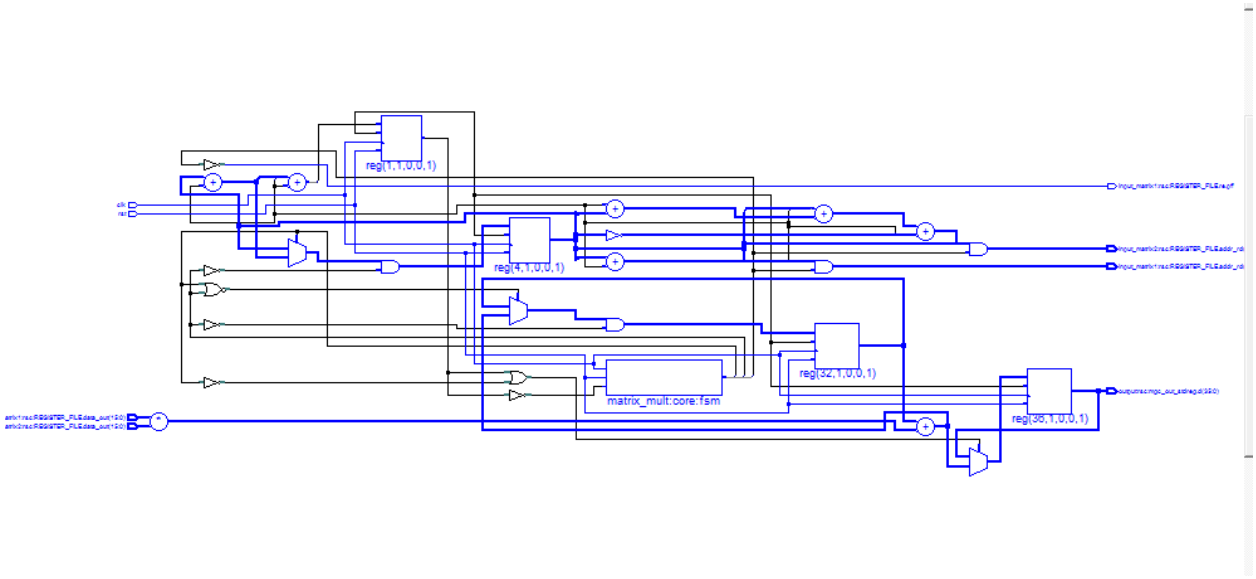
گام ششم)

با استفاده از نتایج به دست امده از بخش قبلی میفهمیم پایپلاین کردن لوپ ها باعث کاهش کم در latency شده و همینطور باعث افزایش فضا میشود برای همین آنرول کردن انتخاب بهتری است همینطور دیدیم که آنرول کردن لوپ سوم باعث افزایش بسیار زیاد فضا میشود برای همین میتوان نتیجه گیری کرد که آنرول لوپ اول و دوم بهترین انتخال است :

matrix_mult.v10	23	2300.00	26	2600.00	95.01	4604.86

همانطور که مشاهده میشود هم latency و هم area کاهش چشمگیری دارند .

دیتاپس:

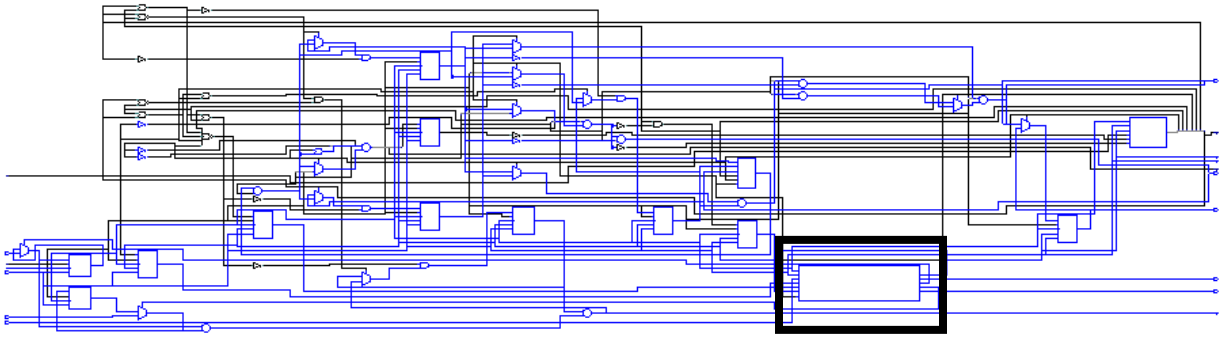


گام هفتم)

matrix_mult.v12	3378	337800.00	3393	339300.00	99.09	7504.63

همانطور که مشاهده میشود تعداد کلاک های ما دوتا بیشتر شده (یکی برای سیگنال start و یکی برای done) طبق انتظار و همینطور به دلیل به وجود امدن یک ماجول جدید برای handshaking مقدار area هم زیادتر میشود :

دیتاپس:



همانطور که مشاهده میشود ماجول اضافه شده در دیتاپس با مستطیل مشکی نشان داده شده است که اگر به درون ان نگاه کنیم متوجه میشویم که تعدادی رجیستر و ... به مدار اضافه میشود که دلیل افزایش area است :