VLSI DPS HW#4

電機四 **4109064119** 陳柏淳

1. Calculate the Frobenius Distance

```
Matrix A :
   -95  -125   -53  -108
   113  -117    62   109
   116   -85   -80    70
    19    38    47    -4
  -113    59   -82   -17
   -68    37   -34   -14
   -38   -13    32   -50
    82    12    71     2
```

- Generate a random matrix A

```
Matrix Q_float :
  -0.3841   0.4569   0.4690   0.0768  -0.4569  -0.2749  -0.1536   0.3315
  -0.7889  -0.4293  -0.2600   0.2247   0.1308   0.0864  -0.1251   0.1873
   0.0765   0.3000  -0.7361   0.2164  -0.3511  -0.1076   0.3423   0.2520
  -0.3412   0.6795  -0.2002  -0.1280   0.3468   0.2077  -0.1314  -0.4297
   0.3284   0.1393  -0.1124   0.4462   0.2219   0.1268  -0.6965   0.3325
        0   0.1891   0.2463   0.1124   0.5477   0.1913   0.5293   0.5236
        0        0   0.2393   0.7462  -0.2118   0.3381   0.2361  -0.4135
        0        0        0   0.3344   0.3695  -0.8338   0.0720  -0.2264

Matrix Q_fix :
  -0.3828   0.4561   0.4697   0.0752  -0.4609  -0.2783  -0.1582   0.3291
  -0.7939  -0.4307  -0.2637   0.2217   0.1270   0.0830  -0.1270   0.1777
   0.0703   0.3018  -0.7363   0.2188  -0.3545  -0.1074   0.3516   0.2422
  -0.3506  -0.0820   0.0967  -0.1367   0.3350   0.1953  -0.0889  -0.4463
   0.3164  -0.6748   0.1904   0.4541   0.2139   0.1182  -0.6777   0.3076
        0  -0.1846  -0.2178   0.0420   0.5234   0.1914   0.4834   0.5400
        0        0   0.2695   0.7607  -0.1572   0.3066   0.2988  -0.3828
        0        0        0   0.2969   0.3965  -0.8467   0.0889  -0.1816
```

```
Matrix R_float :
  247.3297  -73.5415   80.2087  143.7676
   -0.0000  194.2618   42.1888   22.5000
    0.0000    0.0000  144.8919  -37.0943
    0.0000   -0.0000   -0.0000   94.3326
    0.0000    0.0000   -0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000
   -0.0000    0.0000    0.0000    0.0000
   -0.0000    0.0000    0.0000   -0.0000

Matrix R_fix :
  248.0000  -74.7500   79.5000  143.0000
         0  193.7500   41.0000   21.5000
         0         0  146.0000  -37.2500
         0         0         0   93.7500
         0         0         0         0
         0         0         0         0
         0         0         0         0
         0         0         0         0
```

- Q_float and R_float are generated from Q*[A|I] =[R|Q]
- Q_fix and R_fix are generated from Cordic given's rotation
- The Frobenius Distance (Euclidean Norm) is sqrt(trace((float-fix)*(float-fix)'))

```
------------------------------------------------------------------
------------------- Hardware implementation loss -----------------
------------------------------------------------------------------
Q Fix Point Loss :
    1.3433

R Fix Point Loss :
    2.6959
```

Q_loss : 1.3433
R_loss : 2.6959

- Fix-point precision

Data	Sign bit	Integer part	Fraction part	Length
A (input)	1	7	0	8
Q (output)	1	1	10	12
R (output)	1	9	2	12
K (parameter)	1	0	9	10

2. Timing diagram

1	CYCLE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
2	RD_ADDR	81	82	83	84	71	72	73	74	61	62	63	64	51	52	53	54	41	42	43	44	31	32	33	34	21	22	23	24	11	12	13	14																												
3	RD_DATA		81	82	83	84	71	72	73	74	61	62	63	64	51	52	53	54	41	42	43	44	31	32	33	34	21	22	23	24	11	12	13	14																											
4	GG1		N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N																											
5	GR11			N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N																										
6	GR12				N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N																									
7	GR13					N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N																								
8	MK1									K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K																								
9	Q11						N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N																							
10	Q12							N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N																						
11	Q13								N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N																					
12	Q14									N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N																				
13	Q15										N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N																			
14	Q16											N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N																		
15	Q17												N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N																	
16	Q18													N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N																
17	MK1_Q_former													K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K																				
18	MK1_Q_later																	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K																
19	GG2											N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N																						
20	GR21												N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N																					
21	GR22													N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N																				
22	MK2																		K	K	K		K	K	K		K	K	K		K	K	K		K	K	K		K	K	K																				
23	Q21														N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N																			
24	Q22															N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N																		
25	Q23																N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N																	
26	Q24																	N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N																
27	Q25																		N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N															
28	Q26																			N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N														
29	Q27																				N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N													
30	Q28																					N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N												
31	MK2_Q_former																					K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K																
32	MK2_Q_later																									K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K												
33	GG3																					N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N																
34	GR31																						N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N															
35	MK3																												K	K			K	K			K	K			K	K			K	K															
36	Q31																						N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N															
37	Q32																							N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N														
38	Q33																								N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N													
39	Q34																									N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N												
40	Q35																										N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N											
41	Q36																											N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N										
42	Q37																												N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N									
43	Q38																													N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N								
44	MK3_Q_former																													K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K												
45	MK3_Q_later																																	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K								
46	GG4																														N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N											
47	MK4																																					K				K				K				K											
48	Q41																														N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N											
49	Q42																															N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N										
50	Q43																																N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N									
51	Q44																																	N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N								
52	Q45																																		N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N							
53	Q46																																			N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N						
54	Q47																																				N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N					
55	Q48																																					N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N				
56	MK4_Q_former																																					K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K								
57	MK4_Q_later																																									K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K				
58	WR_ADDR																																		11					22	12				33	23	13			44	34	24	14								
59	WR_ADDR_Q1																																					18	17	16	15	14	13	12	11																
60	WR_ADDR_Q2																																									28	27	26	25	24	23	22	21												
61	WR_ADDR_Q3																																													38	37	36	35	34	33	32	31								
62	WR_ADDR_Q4																																																	48	47	46	45	44	43	42	41				
63	WR_ADDR_Q5																																																	58	57	56	55	54	53	52	51				
64	WR_ADDR_Q6																																													68	67	66	65	64	63	62	61								
65	WR_ADDR_Q7																																									78	77	76	75	74	73	72	71												
66	WR_ADDR_Q8																																					88	87	86	85	84	83	82	81																
67	NEXT_IN_VALID																																	IN																											
68	Finish																																																									F			

- N : No Operation (Idle)
- MK : Multiplied by K
- 78 78 78 78 N : Perform rotation on 7th and 8th row
- 18 17 16 15 14 13 12 11 : The matrix index of data write back to tb
- MK1_Q_former : process Q11~Q14
- MK1_Q_later : process Q15~Q18

3. Clock cycles needed to complete one QR factorization

```
------------------------------------------------------------------------------
-------------------------    S U M M A R Y    -------------------------------
******************************************************************************
**    Congratulations!! R matrix data have been generated successfully!     **
**    Congratulations!! Q matrix data have been generated successfully!     **
******************************************************************************
**    The simulation results are all Pass!!                                 **
**    Get finish at cycle: 57                                               **
******************************************************************************
```

57 cycles (As the timing diagram)

4. The initiation interval of two successive QR factorizations

1	CYCLE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
2	RD_ADDR	81	82	83	84	71	72	73	74	61	62	63	64	51	52	53	54	41	42	43	44	31	32	33	34	21	22	23	24	11	12	13	14				
3	RD_DATA		81	82	83	84	71	72	73	74	61	62	63	64	51	52	53	54	41	42	43	44	31	32	33	34	21	22	23	24	11	12	13	14			
4	GG1		N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N			
5	GR11			N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N			
6	GR12				N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N		
7	GR13					N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N	
8	MK1									K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K

When GG1 performs all vectoring over, the next input is valid.
The initiation interval is 33 cycles.

Matlab result

```
Matrix A :
   -95  -125   -53  -108
   113  -117    62   109
   116   -85   -80    70
    19    38    47    -4
  -113    59   -82   -17
   -68    37   -34   -14
   -38   -13    32   -50
    82    12    71     2
```

- Random matrix A
- Hardware loss

```
------------------------------------------------------------------
-------------------- Hardware implementation loss ----------------
------------------------------------------------------------------
Q Fix Point Loss :
    1.3477

R Fix Point Loss :
    2.3847
```

Actual values

```
------------------------------------------------------------------
--------------------------- Original scale -----------------------
------------------------------------------------------------------
Matrix Q_float :
   -0.3841    0.4569    0.4690    0.0768   -0.4569   -0.2749   -0.1536    0.3315
   -0.7889   -0.4293   -0.2600    0.2247    0.1308    0.0864   -0.1251    0.1873
    0.0765    0.3000   -0.7361    0.2164   -0.3511   -0.1076    0.3423    0.2520
   -0.3412    0.6795   -0.2002   -0.1280    0.3468    0.2077   -0.1314   -0.4297
    0.3284    0.1393   -0.1124    0.4462    0.2219    0.1268   -0.6965    0.3325
         0    0.1891    0.2463    0.1124    0.5477    0.1913    0.5293    0.5236
         0         0    0.2393    0.7462   -0.2118    0.3381    0.2361   -0.4135
         0         0         0    0.3344    0.3695   -0.8338    0.0720   -0.2264

Matrix Q_fix :
   -0.3828    0.4561    0.4678    0.0752   -0.4590   -0.2783   -0.1592    0.3320
   -0.7881   -0.4268   -0.2578    0.2256    0.1328    0.0947   -0.1260    0.1914
    0.0762    0.3018   -0.7344    0.2168   -0.3545   -0.1094    0.3457    0.2441
   -0.3477   -0.0918    0.0977   -0.1357    0.3340    0.1914   -0.0967   -0.4404
    0.3223   -0.6748    0.1914    0.4375    0.2178    0.1328   -0.6846    0.3213
         0   -0.1846   -0.2324    0.0781    0.5205    0.2012    0.5020    0.5146
         0         0    0.2500    0.7598   -0.1924    0.3154    0.2549   -0.4063
         0         0         0    0.3193    0.3926   -0.8369    0.0762   -0.1963

Matrix R_float :
  247.3297  -73.5415   80.2087  143.7676
   -0.0000  194.2618   42.1888   22.5000
    0.0000    0.0000  144.8919  -37.0943
    0.0000   -0.0000   -0.0000   94.3326
    0.0000    0.0000   -0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000
   -0.0000    0.0000    0.0000    0.0000
   -0.0000    0.0000    0.0000   -0.0000

Matrix R_fix :
  248.0000  -74.2500   79.7500  143.0000
         0  195.2500   42.0000   22.7500
         0         0  146.5000  -37.5000
         0         0         0   94.0000
         0         0         0         0
         0         0         0         0
         0         0         0         0
         0         0         0         0
```

Shift to integer to simulate hardware performance

```
------------------------------------------------------------------
---------------------------- INT scale ---------------------------
------------------------------------------------------------------
Matrix Q_float :
 -393.3211  467.8451  480.2657   78.6642 -467.8451 -281.5351 -157.3284  339.4982
 -807.8038 -439.6231 -266.2414  230.0868  133.8913   88.4553 -128.0858  191.7783
   78.3759  307.1939 -753.7280  221.6229 -359.5194 -110.1942  350.5433  258.0019
 -349.4265  695.8533 -204.9684 -131.0403  355.1718  212.6709 -134.5897 -439.9903
  336.2825  142.6095 -115.1396  456.8971  227.2753  129.8618 -713.2458  340.4449
         0  193.6781  252.2366  115.1114  560.8470  195.9207  542.0406  536.1423
         0         0  245.0265  764.1410 -216.8794  346.2087  241.7700 -423.4107
         0         0         0  342.4072  378.3236 -853.8199   73.7731 -231.8482

Matrix Q_fix :
  -392   467   479    77  -470  -285  -163   340
  -807  -437  -264   231   136    97  -129   196
    78   309  -752   222  -363  -112   354   250
  -356   -94   100  -139   342   196   -99  -451
   330  -691   196   448   223   136  -701   329
     0  -189  -238    80   533   206   514   527
     0     0   256   778  -197   323   261  -416
     0     0     0   327   402  -857    78  -201

          DataTypeMode: Fixed-point: binary point scaling
            Signedness: Signed
            WordLength: 12
        FractionLength: 0

Matrix R_float :
  989.3190 -294.1660  320.8349  575.0704
   -0.0000  777.0472  168.7551   89.9999
    0.0000    0.0000  579.5677 -148.3771
    0.0000   -0.0000   -0.0000  377.3306
    0.0000    0.0000   -0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000
   -0.0000    0.0000    0.0000    0.0000
   -0.0000    0.0000    0.0000   -0.0000

Matrix R_fix :
   992  -297   319   572
     0   781   168    91
     0     0   586  -150
     0     0     0   376
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0

          DataTypeMode: Fixed-point: binary point scaling
            Signedness: Signed
            WordLength: 12
        FractionLength: 0
```

- **Matlab result**

Record the values during iteration, which can be used for debugging in hardware implementation.

```
k = 1 row78:

d = 1 1 -1 1
GG1  Iteration 4 times: input: X =   152; Y =  -328 output: X =   594; Y =    -5
GR11 Iteration 4 times: input: X =    52; Y =   -48 output: X =   108; Y =    43
GR12 Iteration 4 times: input: X =  -128; Y =  -284 output: X =   331; Y =  -390
GR13 Iteration 4 times: input: X =   200; Y =    -8 output: X =   153; Y =   291
Q11  Iteration 4 times: input: X =     0; Y =     0 output: X =     0; Y =     0
Q12  Iteration 4 times: input: X =     0; Y =     0 output: X =     0; Y =     0
Q13  Iteration 4 times: input: X =     0; Y =     0 output: X =     0; Y =     0
Q14  Iteration 4 times: input: X =     0; Y =     0 output: X =     0; Y =     0
Q15  Iteration 4 times: input: X =     0; Y =     0 output: X =     0; Y =     0
Q16  Iteration 4 times: input: X =     0; Y =     0 output: X =     0; Y =     0
Q17  Iteration 4 times: input: X = -1024; Y =     0 output: X =  -720; Y = -1520
Q18  Iteration 4 times: input: X =     0; Y = -1024 output: X =  1520; Y =  -720

d = 1 -1 -1 -1
GG1  Iteration 8 times: input: X =   594; Y =    -5 output: X =   596; Y =     1
GR11 Iteration 8 times: input: X =   108; Y =    43 output: X =   107; Y =    45
GR12 Iteration 8 times: input: X =   331; Y =  -390 output: X =   334; Y =  -388
GR13 Iteration 8 times: input: X =   153; Y =   291 output: X =   150; Y =   293
Q11  Iteration 8 times: input: X =     0; Y =     0 output: X =     0; Y =     0
Q12  Iteration 8 times: input: X =     0; Y =     0 output: X =     0; Y =     0
Q13  Iteration 8 times: input: X =     0; Y =     0 output: X =     0; Y =     0
Q14  Iteration 8 times: input: X =     0; Y =     0 output: X =     0; Y =     0
Q15  Iteration 8 times: input: X =     0; Y =     0 output: X =     0; Y =     0
Q16  Iteration 8 times: input: X =     0; Y =     0 output: X =     0; Y =     0
Q17  Iteration 8 times: input: X =  -720; Y = -1520 output: X =  -711; Y = -1528
Q18  Iteration 8 times: input: X =  1520; Y =  -720 output: X =  1528; Y =  -708

d = -1 0 0 0
GG1  Iteration 12 times: input: X =   832; Y =     3 output: X =   832; Y =     0
MK:   X = 505; Y = 0
GR11 Iteration 12 times: input: X =  -711; Y =  -384 output: X =  -713; Y =  -381
MK:   X = 492; Y = -344
GR12 Iteration 12 times: input: X =    33; Y =   758 output: X =    35; Y =   758
MK:   X = 428; Y = -460
GR13 Iteration 12 times: input: X =   499; Y =   355 output: X =   500; Y =   354
MK:   X = 492; Y = -204
Q11  Iteration 12 times: input: X =     0; Y =     0 output: X =     0; Y =     0
MK:   X = 0; Y = 0
Q12  Iteration 12 times: input: X =     0; Y =     0 output: X =     0; Y =     0
MK:   X = 0; Y = 0
Q13  Iteration 12 times: input: X =     0; Y =     0 output: X =     0; Y =     0
MK:   X = 0; Y = 0
Q14  Iteration 12 times: input: X =     0; Y =     0 output: X =     0; Y =     0
MK:   X = 0; Y = 0
Q15  Iteration 12 times: input: X =     0; Y =     0 output: X =     0; Y =     0
MK:   X = 0; Y = 0
Q16  Iteration 12 times: input: X =     0; Y =     0 output: X =     0; Y =     0
MK:   X = 0; Y = 0
Q17  Iteration 12 times: input: X =  -247; Y =  1669 output: X =  -241; Y =  1670
MK:   X = -147; Y = 1014
Q18  Iteration 12 times: input: X = -1669; Y =  -246 output: X = -1670; Y =  -239
MK:   X = -1015; Y = -146
```

```
k = 4 row45:

d = 1 -1 1 1
GG4  Iteration 4 times: input: X =   346; Y =  -331 output: X =   784; Y =   -57
Q41  Iteration 4 times: input: X =   759; Y =     0 output: X =   960; Y =   794
Q42  Iteration 4 times: input: X =    97; Y =   321 output: X =  -212; Y =   507
Q43  Iteration 4 times: input: X =  -201; Y =    53 output: X =  -309; Y =  -144
Q44  Iteration 4 times: input: X =   186; Y =  -615 output: X =   880; Y =  -584
Q45  Iteration 4 times: input: X =   235; Y =   127 output: X =   165; Y =   406
Q46  Iteration 4 times: input: X =    86; Y =  -710 output: X =   853; Y =  -809
Q47  Iteration 4 times: input: X =  -134; Y =   157 output: X =  -334; Y =    58
Q48  Iteration 4 times: input: X =   -57; Y =   143 output: X =  -221; Y =   120

d = 1 1 -1 -1
GG4  Iteration 8 times: input: X =   784; Y =   -57 output: X =   789; Y =    -2
Q41  Iteration 8 times: input: X =   960; Y =   794 output: X =   904; Y =   862
Q42  Iteration 8 times: input: X =  -212; Y =   507 output: X =  -248; Y =   492
Q43  Iteration 8 times: input: X =  -309; Y =  -144 output: X =  -299; Y =  -166
Q44  Iteration 8 times: input: X =   880; Y =  -584 output: X =   921; Y =  -522
Q45  Iteration 8 times: input: X =   165; Y =   406 output: X =   136; Y =   418
Q46  Iteration 8 times: input: X =   853; Y =  -809 output: X =   910; Y =  -749
Q47  Iteration 8 times: input: X =  -334; Y =    58 output: X =  -338; Y =    35
Q48  Iteration 8 times: input: X =  -221; Y =   120 output: X =  -230; Y =   104

d = 1 -1 0 0
GG4  Iteration 12 times: input: X =   789; Y =    -2 output: X =   790; Y =     0
MK:   X = 479; Y = 0
Q41  Iteration 12 times: input: X =   904; Y =   862 output: X =   902; Y =   864
MK:   X = 547; Y = 524
Q42  Iteration 12 times: input: X =  -248; Y =   492 output: X =  -249; Y =   492
MK:   X = -152; Y = 298
Q43  Iteration 12 times: input: X =  -299; Y =  -166 output: X =  -299; Y =  -167
MK:   X = -182; Y = -102
Q44  Iteration 12 times: input: X =   921; Y =  -522 output: X =   922; Y =  -520
MK:   X = 560; Y = -316
Q45  Iteration 12 times: input: X =   136; Y =   418 output: X =   135; Y =   418
MK:   X = 82; Y = 253
Q46  Iteration 12 times: input: X =   910; Y =  -749 output: X =   911; Y =  -747
MK:   X = 553; Y = -454
Q47  Iteration 12 times: input: X =  -338; Y =    35 output: X =  -338; Y =    34
MK:   X = -206; Y = 20
Q48  Iteration 12 times: input: X =  -230; Y =   104 output: X =  -230; Y =   104
MK:   X = -140; Y = 63
```

Matlab code (only core part):

- Floating-point (Q*[A|I] = [R|Q])

```
% Floating point QR factorization using Q*[A|I] = [R|Q]
function R_Q = float_QR(row_R, col_R, row_Q, Q, A)
    % Q*[A|I] = [R|Q]
    R_Q = [A Q];
    % Perform Givens rotations
    for p_float = 1 : col_R
        for q_float = (row_R-1) : (-1) : p_float
            Q     = eye(row_Q);
            theta = atan2(R_Q(q_float+1, p_float), R_Q(q_float, p_float));
            % Givens Q
            Q(q_float  , q_float  ) = cos(theta);
            Q(q_float  , q_float+1) = sin(theta);
            Q(q_float+1, q_float  ) = -sin(theta);
            Q(q_float+1, q_float+1) = cos(theta);
            % Q*[A|I] = [R|Q]
            R_Q = Q * R_Q;
        end
    end
end
```

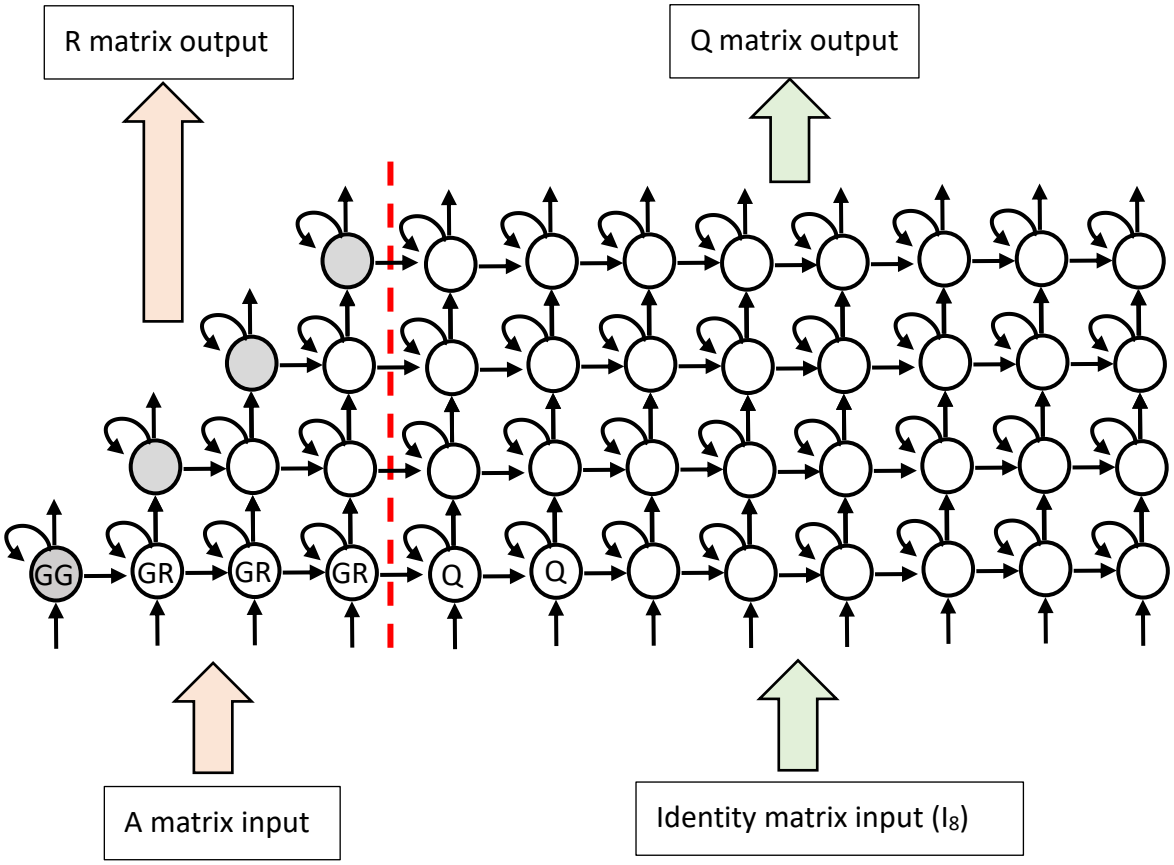
- Fixed-point (Cordic Given's rotation)

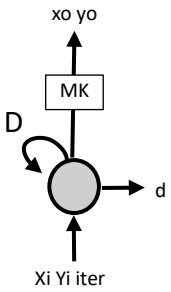
```
% Fixed point QR factorization with the CORDIC
function [Q_cordic, R_cordic] = Cordic_QR(K_cordic, Q_scaled, R_scaled, row_R, col_R, col_Q, iter_num,
R_sign, R_len, R_frac, Q_sign, Q_len, Q_frac)
    F = fimath('RoundingMethod','Floor');
    Q_cordic = Q_scaled;
    R_cordic = R_scaled;

    % Eliminate A(q+1,p) by A(q,p)
    for p_fix = 1 : col_R
        for q_fix = (row_R-1) : (-1) : p_fix
            % Column q and column q+1 are rotated 180 degrees
            if R_cordic(q_fix,p_fix) < 0
                for reverse_R = p_fix : col_R
                    R_cordic(q_fix   ,reverse_R) = -R_cordic(q_fix   ,reverse_R);
                    R_cordic(q_fix+1,reverse_R) = -R_cordic(q_fix+1,reverse_R);
                end
                for reverse_Q = p_fix : col_Q
                    Q_cordic(q_fix   ,reverse_Q) = -Q_cordic(q_fix   ,reverse_Q);
                    Q_cordic(q_fix+1,reverse_Q) = -Q_cordic(q_fix+1,reverse_Q);
                end
            end
            x_pre = zeros(col_Q, 1);
            y_pre = zeros(col_Q, 1);
            for iter = 0 : iter_num-1
                % vectoring mode
                x_vect = R_cordic(q_fix   , p_fix);
                y_vect = R_cordic(q_fix+1, p_fix);
                [X_vect, Y_vect, d] = GG(x_vect, y_vect, R_len, R_frac, iter);

                if iter == iter_num-1
                    R_cordic(q_fix, p_fix) = fi((X_vect*K_cordic),R_sign,R_len,R_frac, F);
                    R_cordic(q_fix+1,p_fix)= fi((Y_vect*K_cordic),R_sign,R_len,R_frac, F);
                else
                    R_cordic(q_fix   , p_fix) = X_vect;
                    R_cordic(q_fix+1, p_fix) = Y_vect;
                end

                % rotation mode
                for rot_R = 1 : (col_R-p_fix)
                    x_rot_R = R_cordic(q_fix   , p_fix+rot_R);
                    y_rot_R = R_cordic(q_fix+1, p_fix+rot_R);
                    [X_rot_R, Y_rot_R] = GR(x_rot_R, y_rot_R, d, R_len, R_frac, iter);

```

```
            if iter == iter_num-1
                R_cordic(q_fix,p_fix+rot_R)=fi((X_rot_R*K_cordic),R_sign,R_len, R_frac,F);
                R_cordic(q_fix+1,p_fix+rot_R)=fi((Y_rot_R*K_cordic),R_sign,R_len,R_frac,F);
                else
                R_cordic(q_fix    , p_fix+rot_R) = X_rot_R;
                R_cordic(q_fix+1, p_fix+rot_R) = Y_rot_R;
            end
            end
            % compute Q (As the processing of R)
            for rot_Q = 1 : col_Q
                x_rot_Q = Q_cordic(q_fix    , rot_Q);
                y_rot_Q = Q_cordic(q_fix+1, rot_Q);
                [X_rot_Q, Y_rot_Q] = GR(x_rot_Q, y_rot_Q, d, Q_len, Q_frac, iter);

                if iter == iter_num-1
                    Q_cordic(q_fix,rot_Q)=fi((X_rot_Q*K_cordic),Q_sign,Q_len,Q_frac,F);
                    Q_cordic(q_fix+1,rot_Q)=fi((Y_rot_Q*K_cordic),Q_sign,Q_len,Q_frac,F);
                else
                    Q_cordic(q_fix    , rot_Q) = X_rot_Q;
                    Q_cordic(q_fix+1, rot_Q) = Y_rot_Q;
                end
            end
            end
        end
    end
end

% GG : vectoring mode
function [X, Y, d] = GG(x, y, len, frac, iter)
    d = -sign(x * y);
    X = x - d * bitsra(y, iter);
    Y = y + d * bitsra(x, iter);

    F = fimath('RoundingMethod','Floor');
    X = fi(X, 1, len, frac, F);
    Y = fi(Y, 1, len, frac, F);
end

% GR : rotation mode
function [X, Y] = GR(x, y, d, len, frac, iter)
    X = x - d * bitsra(y, iter);
    Y = y + d * bitsra(x, iter);

    F = fimath('RoundingMethod','Floor');
    X = fi(X, 1, len, frac, F);
    Y = fi(Y, 1, len, frac, F);
end
```

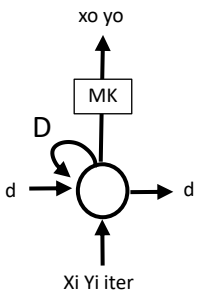
- **Hardware implementation**

  1. Hardware architecture ($s=[1\ 1\ 0]^t$, $d=[1\ 0\ 0]^t$)





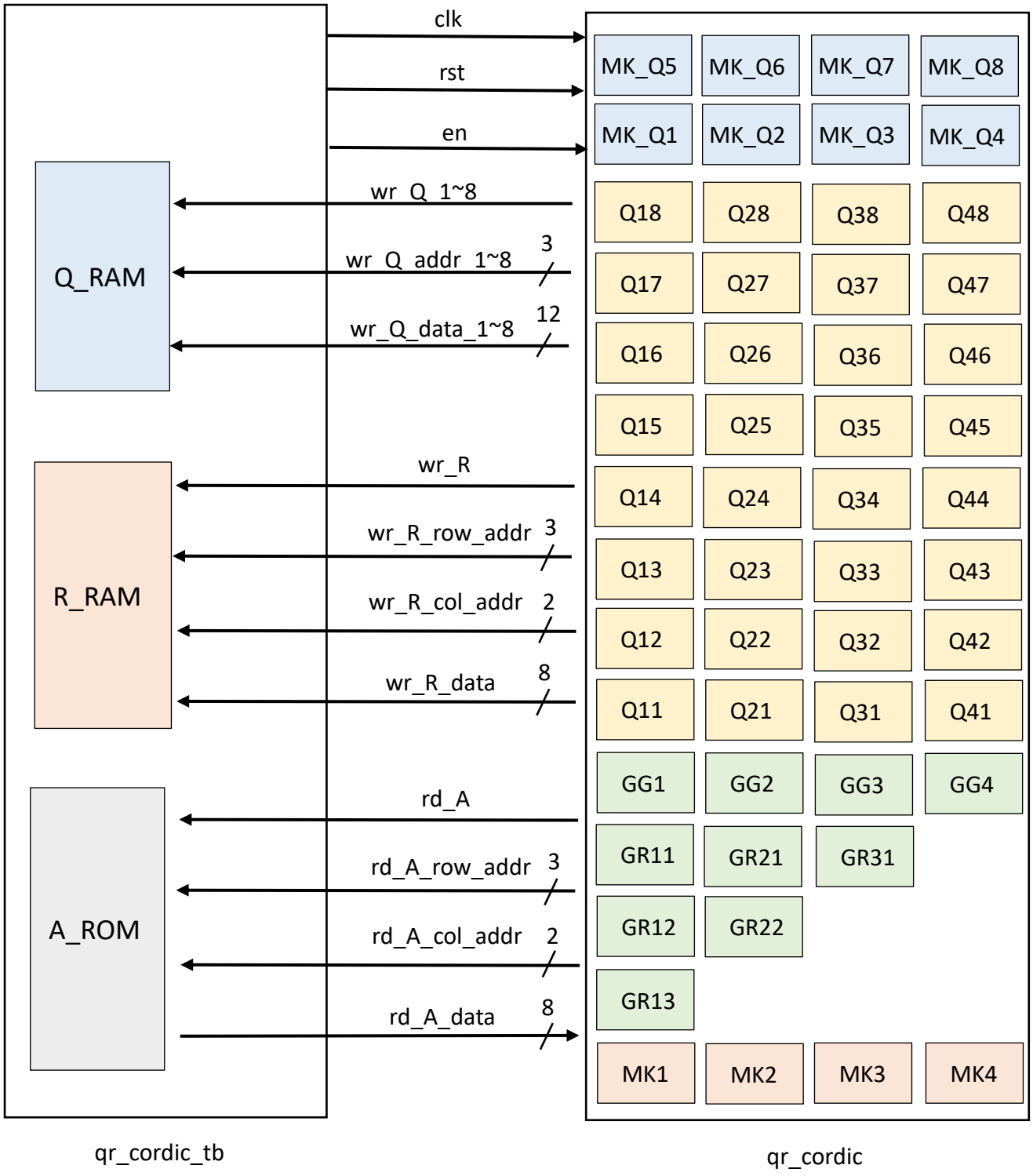
GG(vectoring mode) : One GG contains 4 micro-rotation modules, when iteration over, it outputs to above and multiplied by K(MK module), the rest of data will be transmitted to the right GR and Q modules.



GR(rotation mode) : GR is almost the same module as GG, the only difference is rotation direction d, d will input from the left module(GG or GR).

- There are 4 GG (vectoring mode) and (6+32) GR (rotation mode)
- GG output the direction of vector rotation and transmits it to GR. In each iteration, the rotation angle is halved to approach the target angle.

2. System diagram



3. Verilog code (only core part):

a. GG module

```
1  module GG_one_iter #(
2      parameter R_LEN    = 12,
3      parameter R_FRAC = 2
4  )(
5      input            signed [R_LEN-1:0]       xi,
6      input            signed [R_LEN-1:0]       yi,
7      input                   [3:0]              iter,
8      output                  [1:0]              d,
9      output   reg     signed [R_LEN-1:0]       xo,
10     output   reg     signed [R_LEN-1:0]       yo
11 );
12
13 assign d = (yi == 'd0) ? 'd2 : xi[R_LEN-1] ^ yi[R_LEN-1];
14
15 always @(*) begin
16     if(d == 'd2) begin
17         xo = xi;
18         yo = yi;
19     end
20     else if(d == 'd1) begin
21         xo = xi - (yi >>> iter);
22         yo = yi + (xi >>> iter);
23     end
24     else begin
25         xo = xi + (yi >>> iter);
26         yo = yi - (xi >>> iter);
27     end
28 end
29
30 endmodule
```

b. GR module

```
module GR_one_iter #(
    parameter R_LEN   = 12,
    parameter R_FRAC = 2
)
(   input           signed [R_LEN-1:0]      xi,
    input           signed [R_LEN-1:0]      yi,
    input                  [3:0]            iter,
    input                  [1:0]            d,
    output  reg    signed [R_LEN-1:0]      xo,
    output  reg    signed [R_LEN-1:0]      yo
);

always @(*) begin
    if(d == 'd2) begin
        xo = xi;
        yo = yi;
    end
    else if(d == 'd1) begin
        xo = xi - (yi >>> iter);
        yo = yi + (xi >>> iter);
    end
    else begin
        xo = xi + (yi >>> iter);
        yo = yi - (xi >>> iter);
    end
end

endmodule
```

c. Q module is the same module as GR

d. MK (multiplied by K)

```
module MK #(
    parameter R_LEN  = 12,
    parameter R_FRAC = 2,
    parameter K_LEN  = 10,
    parameter K_FRAC = 9
)(
    input  signed [R_LEN-1:0]  xi,
    input  signed [R_LEN-1:0]  yi,
    output signed [R_LEN-1:0]  xo,
    output signed [R_LEN-1:0]  yo
);

localparam signed  K = 10'b0_100110111; // K = 0.607421875

wire signed [R_LEN+K_LEN-1:0]  xo_0;
wire signed [R_LEN+K_LEN-1:0]  yo_0;


assign xo_0 = xi * K;
assign yo_0 = yi * K;

// truncate to R_LEN bits
assign xo = xo_0[R_LEN+K_FRAC-1:K_FRAC];
assign yo = yo_0[R_LEN+K_FRAC-1:K_FRAC];

endmodule

```

e. GG iteration control

```
// iteration times
always @(posedge clk or posedge rst) begin
    if (rst) begin
        iter_gg1 <= 'd0;
    end
    else if(ROT_wire) begin
        if(nop_gg1) begin
            iter_gg1 <= 'd0;
        end
        else if(iter_last_gg1) begin
            iter_gg1 <= iter_gg1 + 'd1;
        end
        else begin
            iter_gg1 <= iter_gg1 + ITER_ONE_CYCLE;
        end
    end
    else begin
        iter_gg1 <= 'd0;
    end
end

// GG1 input data xi, yi
always @(posedge clk or posedge rst) begin
    if (rst) begin
        xi_gg1 <= 'd0;
        yi_gg1 <= 'd0;
    end
    else if(OP_wire) begin
        case(iter_gg1)
            0: begin
                if(start_gg1) begin
                    xi_gg1 <= 'd0;
                    yi_gg1 <= rd_A_data_ext;
                end
                else if(nop_gg1 && !finish_gg1) begin
                    xi_gg1 <= rd_A_data_ext;
                    yi_gg1 <= yo_gg1;
                end
                else begin
                    xi_gg1 <= xo_gg1;
                    yi_gg1 <= yo_gg1;
                end
            end
            ITER_K: begin
                if(finish_gg1) begin
```

```
                        xi_gg1 <= xo_gg1;
                        yi_gg1 <= yo_gg1;
                    end
                    else begin
                        xi_gg1 <= rd_A_data_ext;
                        yi_gg1 <= xo_mk1;
                    end
                end
                default: begin
                    xi_gg1 <= xo_gg1;
                    yi_gg1 <= yo_gg1;
                end
            endcase
        end
        else begin
            xi_gg1 <= 'd0;
            yi_gg1 <= 'd0;
        end
end

// GG1 mk_count
always @(posedge clk or posedge rst) begin
        if (rst) begin
            mk_count_gg1 <= 'd0;
        end
        else if(multk_gg1) begin
            mk_count_gg1 <= mk_count_gg1 + 'd1;
        end
end
```

f. GR iteration control

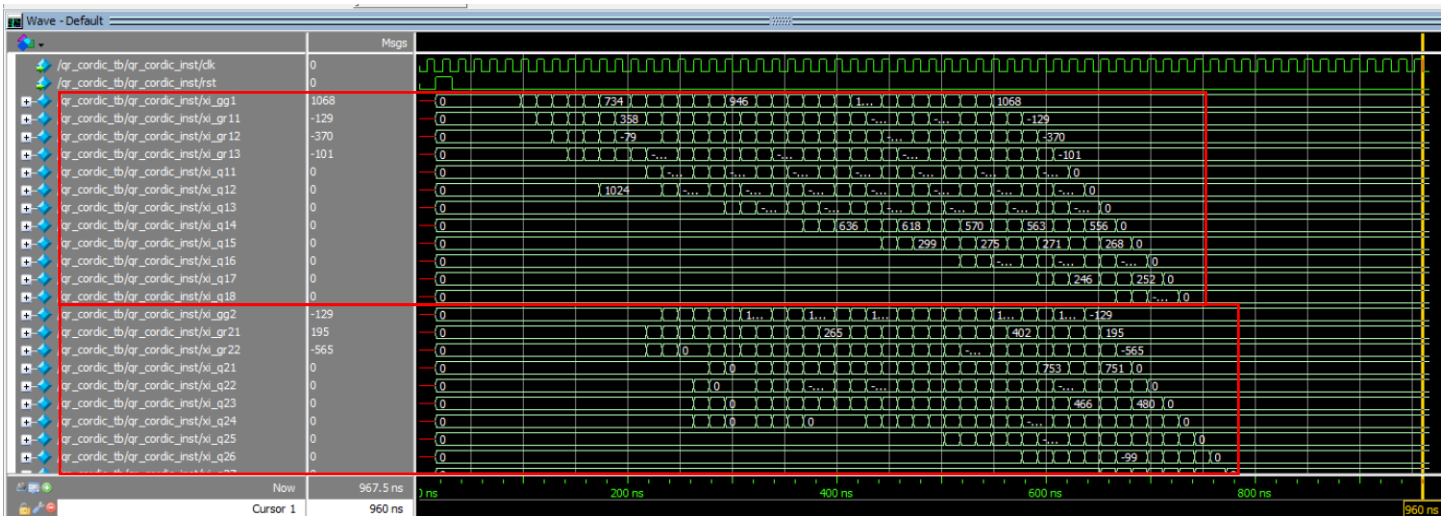
```
// data propagated from left to right
always @(posedge clk or posedge rst) begin
    if (rst) begin
        iter_gr11     <= 'd0;
        nop_gr11      <= 'd0;
        d1_gr11       <= 'd0;
        d2_gr11       <= 'd0;
        d3_gr11       <= 'd0;
        d4_gr11       <= 'd0;
        neg_gr11      <= 'd0;
        mk_count_gr11 <= 'd0;
    end
    else begin
        iter_gr11     <= iter_gg1;
        nop_gr11      <= nop_gg1;
        d1_gr11       <= d1_gg1;
        d2_gr11       <= d2_gg1;
        d3_gr11       <= d3_gg1;
        d4_gr11       <= d4_gg1;
        neg_gr11      <= neg_gg1;
        mk_count_gr11 <= mk_count_gg1;
    end
end

// GR11 input data xi, yi
always @(posedge clk or posedge rst) begin
    if (rst) begin
        xi_gr11 <= 'd0;
        yi_gr11 <= 'd0;
    end
    else if(OP_wire) begin
        case(iter_gr11)
            0: begin
                if(start_gr11_reg) begin
                    xi_gr11 <= 'd0;
                    yi_gr11 <= rd_A_data_ext;
                end
                else if(nop_gr11 && !finish_gr11) begin
                    xi_gr11 <= rd_A_data_ext;
                    yi_gr11 <= yo_gr11;
                end
                else begin
                    xi_gr11 <= xo_gr11;
                    yi_gr11 <= yo_gr11;
                end
```

```
46                  end
47                  ITER_K: begin
48                      if(finish_gr11) begin
49                          xi_gr11 <= xo_mk1;
50                          yi_gr11 <= yo_mk1;
51                      end
52                      else begin
53                          xi_gr11 <= rd_A_data_ext;
54                          yi_gr11 <= xo_mk1;
55                      end
56                  end
57                  default: begin
58                      xi_gr11 <= xo_gr11;
59                      yi_gr11 <= yo_gr11;
60                  end
61              endcase
62      end
63      else begin
64          xi_gr11 <= 'd0;
65          xi_gr11 <= 'd0;
66      end
67  end
```

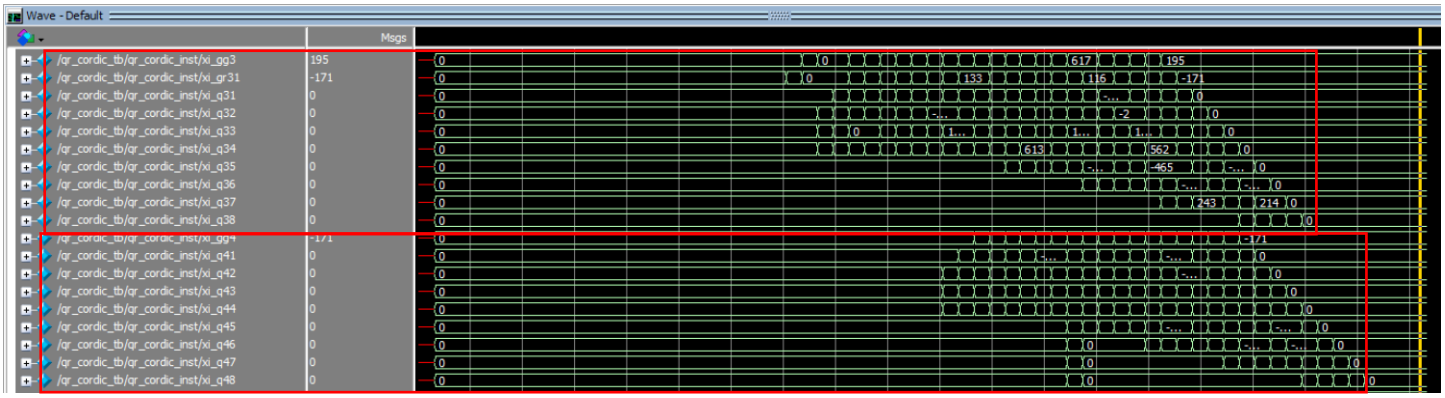
4. RTL Simulation

a. Simulation waveform (Data propagate form left to right PE (GG, GR, Q))

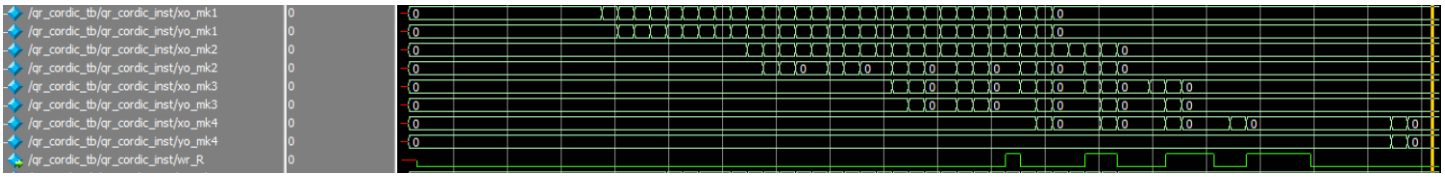
- Data propagate : GG1 →Q18, GG2 →Q28



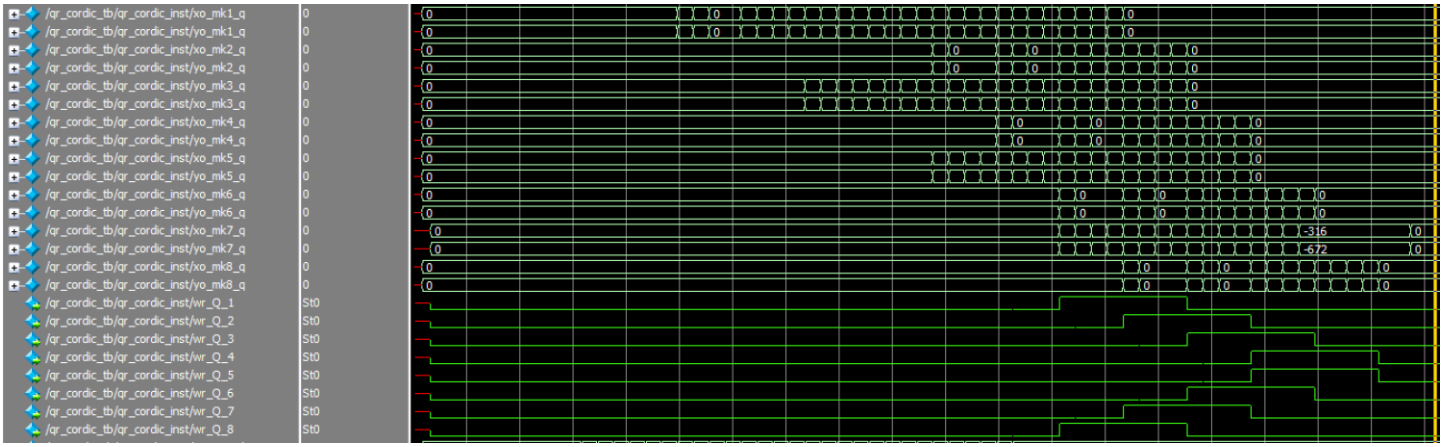
- Data propagate : GG3 →Q38, GG4 →Q48



- Output (MK output R)



- Output (MK output Q)



- Data flow & data scheduling

  GG1→GR11→GR12→GR13→Q11→Q12→Q13→…..→Q18

  -------- GG2→GR21→GR22→Q21→Q22→Q23→…..→Q28

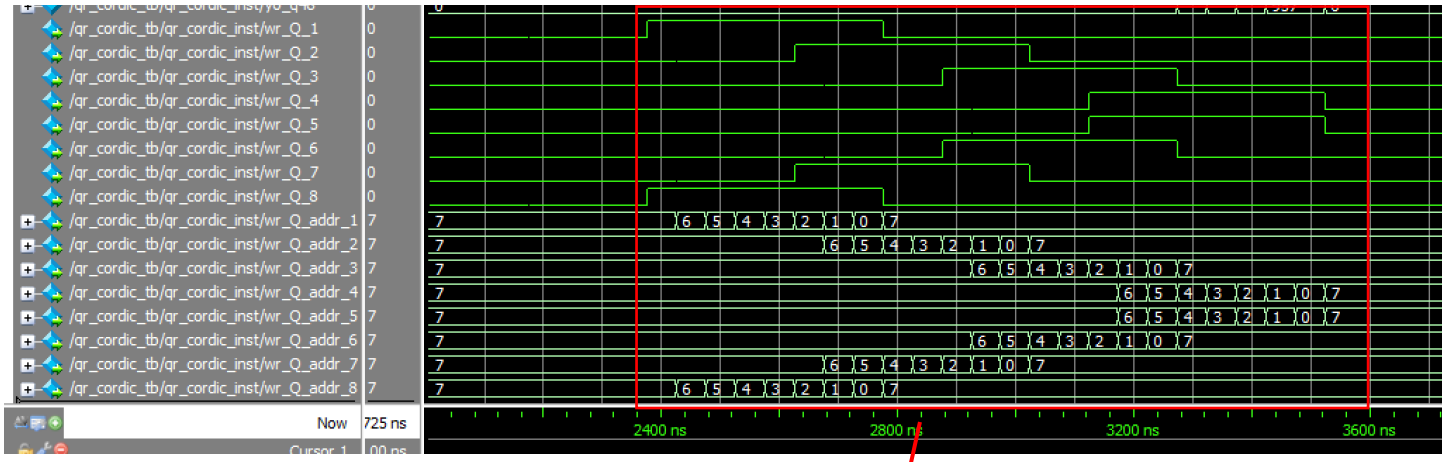
  ---------- GG3→GR31→Q31→Q32→Q33→…..→Q38

  ---------- GG4→Q41→Q42→Q43→…..→Q48

- Corresponding Timing diagram

1	CYCLE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
2	RD_ADDR	81	82	83	84	71	72	73	74	61	62	63	64	51	52	53	54	41	42	43	44	31	32	33	34	21	22	23	24	11	12	13	14													
3	RD_DATA		81	82	83	84	71	72	73	74	61	62	63	64	51	52	53	54	41	42	43	44	31	32	33	34	21	22	23	24	11	12	13	14													
4	GG1		N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N													
5	GR11			N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N													
6	GR12				N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N													
7	GR13					N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N													
8	MK1									K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K													
9	Q11						N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N									
10	Q12							N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N								
11	Q13								N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N							
12	Q14									N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N						
13	Q15										N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N					
14	Q16											N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N				
15	Q17												N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N			
16	Q18													N	N	N	N	78	78	78	N	67	67	67	N	56	56	56	N	45	45	45	N	34	34	34	N	23	23	23	N	12	12	12	N	
17	MK1_Q_former													K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K					
18	MK1_Q_later																	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K	K

- Write Q matrix data back



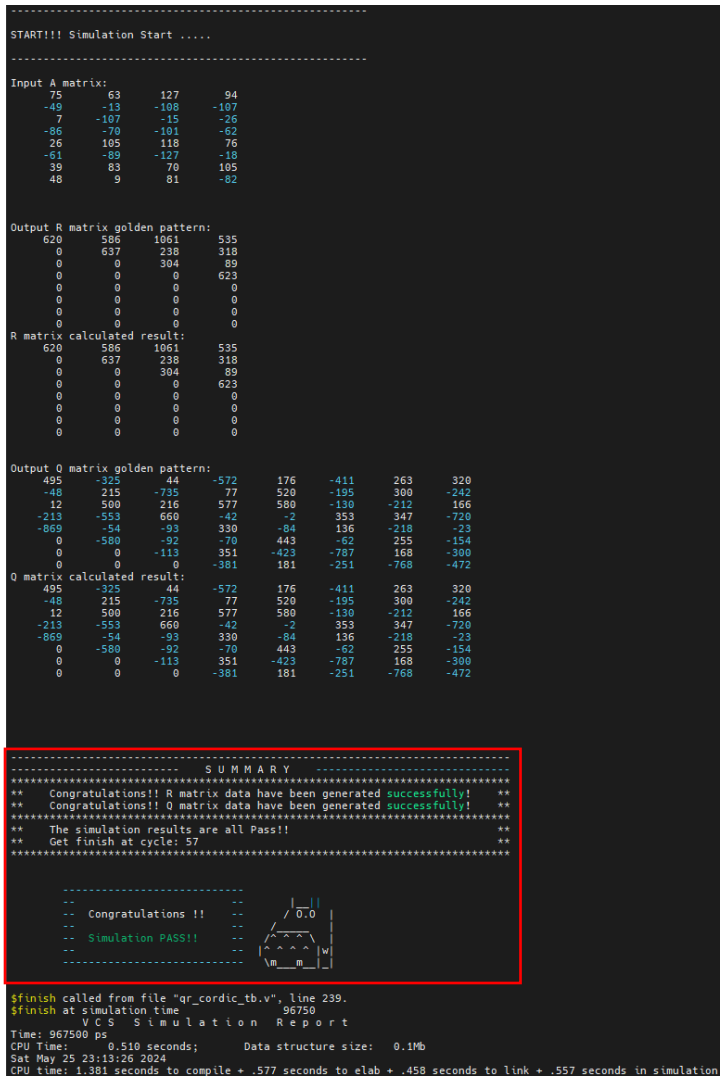
Each row has an independent write enable signal, since multiple data outputs from Q occur simultaneously in the systolic array.

- Corresponding Timing diagram

18	17	16	15	14	13	12	11												
				28	27	26	25	24	23	22	21								
								38	37	36	35	34	33	32	31				
												48	47	46	45	44	43	42	41
												58	57	56	55	54	53	52	51
								68	67	66	65	64	63	62	61				
				78	77	76	75	74	73	72	71								
88	87	86	85	84	83	82	81												

b. Simulation transcript (4 patterns)

Pattern 1:



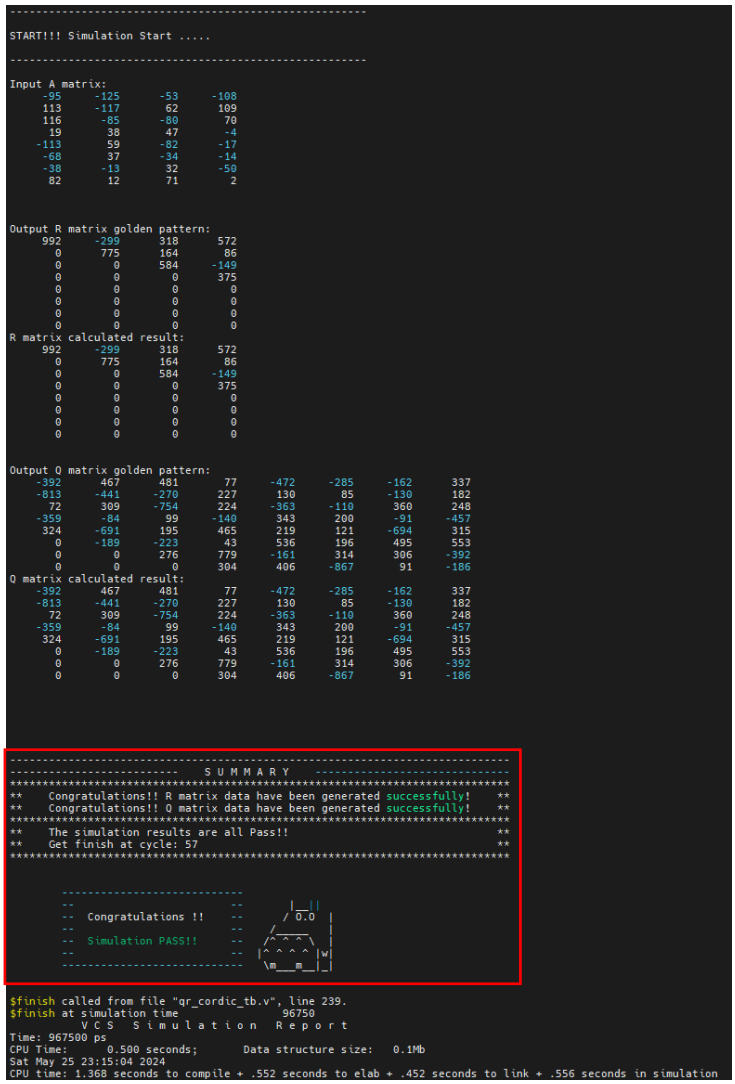
```
----------------------------------------------------

START!!! Simulation Start .....

----------------------------------------------------

Input A matrix:
      75       63      127       94
     -49      -13     -108     -107
       7     -107      -15      -26
     -86      -70     -101      -62
      26      105      118       76
     -61      -89     -127      -18
      39       83       70      105
      48        9       81      -82


Output R matrix golden pattern:
     620      586     1061      535
       0      637      238      318
       0        0      304       89
       0        0        0      623
       0        0        0        0
       0        0        0        0
       0        0        0        0
       0        0        0        0
R matrix calculated result:
     620      586     1061      535
       0      637      238      318
       0        0      304       89
       0        0        0      623
       0        0        0        0
       0        0        0        0
       0        0        0        0
       0        0        0        0


Output Q matrix golden pattern:
     495     -325       44     -572      176     -411      263      320
     -48      215     -735       77      520     -195      300     -242
      12      500      216      577      580     -130     -212      166
    -213     -553      660      -42       -2      353      347     -720
    -869      -54      -93      330      -84      136     -218      -23
       0     -580      -92      -70      443      -62      255     -154
       0        0     -113      351     -423     -787      168     -300
       0        0        0     -381      181     -251     -768     -472
Q matrix calculated result:
     495     -325       44     -572      176     -411      263      320
     -48      215     -735       77      520     -195      300     -242
      12      500      216      577      580     -130     -212      166
    -213     -553      660      -42       -2      353      347     -720
    -869      -54      -93      330      -84      136     -218      -23
       0     -580      -92      -70      443      -62      255     -154
       0        0     -113      351     -423     -787      168     -300
       0        0        0     -381      181     -251     -768     -472
```



```
--------------------------------------------------------------------------------
------------------------    S U M M A R Y    ------------------------------------
********************************************************************************
**    Congratulations!! R matrix data have been generated successfully!     **
**    Congratulations!! Q matrix data have been generated successfully!     **
********************************************************************************
**    The simulation results are all Pass!!                                 **
**    Get finish at cycle: 57                                               **
********************************************************************************


        ----------------------------
        --                        --       |__||
        --  Congratulations !!    --      / 0.0  |
        --                        --     /_____   |
        --  Simulation PASS!!     --   /^ ^ ^ ^ \  |
        --                        --  |^ ^ ^ ^ |w|
        ----------------------------   \m___m__|_|
```

```
$finish called from file "qr_cordic_tb.v", line 239.
$finish at simulation time                96750
           V C S   S i m u l a t i o n   R e p o r t
Time: 967500 ps
CPU Time:      0.510 seconds;       Data structure size:   0.1Mb
Sat May 25 23:13:26 2024
CPU time: 1.381 seconds to compile + .577 seconds to elab + .458 seconds to link + .557 seconds in simulation
```

Pattern 2:



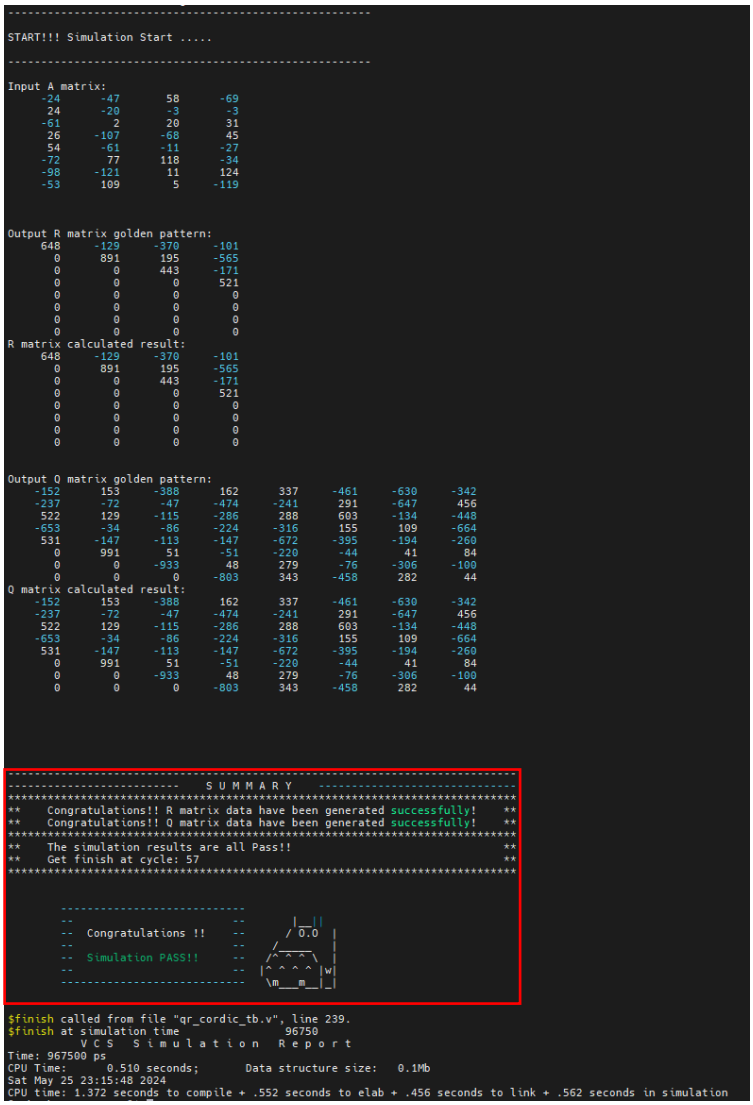
```
----------------------------------------------------

START!!! Simulation Start .....

----------------------------------------------------

Input A matrix:
     -95     -125      -53     -108
     113     -117       62      109
     116      -85      -80       70
      19       38       47       -4
    -113       59      -82      -17
     -68       37      -34      -14
     -38      -13       32      -50
      82       12       71        2


Output R matrix golden pattern:
     992     -299      318      572
       0      775      164       86
       0        0      584     -149
       0        0        0      375
       0        0        0        0
       0        0        0        0
       0        0        0        0
       0        0        0        0
R matrix calculated result:
     992     -299      318      572
       0      775      164       86
       0        0      584     -149
       0        0        0      375
       0        0        0        0
       0        0        0        0
       0        0        0        0
       0        0        0        0


Output Q matrix golden pattern:
    -392      467      481       77     -472     -285     -162      337
    -813     -441     -270      227      130       85     -130      182
      72      309     -754      224     -363     -110      360      248
    -359      -84       99     -140      343      200      -91     -457
     324     -691      195      465      219      121     -694      315
       0     -189     -223       43      536      196      495      553
       0        0      276      779     -161      314      306     -392
       0        0        0      304      406     -867       91     -186
Q matrix calculated result:
    -392      467      481       77     -472     -285     -162      337
    -813     -441     -270      227      130       85     -130      182
      72      309     -754      224     -363     -110      360      248
    -359      -84       99     -140      343      200      -91     -457
     324     -691      195      465      219      121     -694      315
       0     -189     -223       43      536      196      495      553
       0        0      276      779     -161      314      306     -392
       0        0        0      304      406     -867       91     -186
```



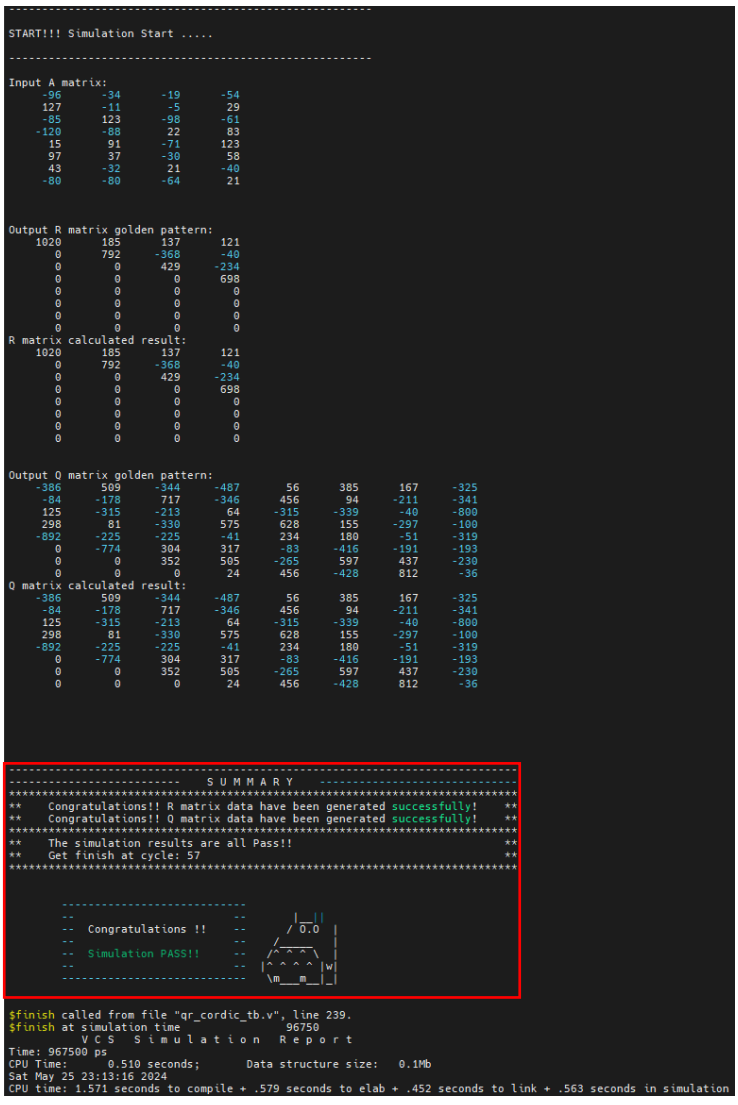
```
--------------------------------------------------------------------------------
------------------------    S U M M A R Y    ------------------------------------
********************************************************************************
**    Congratulations!! R matrix data have been generated successfully!     **
**    Congratulations!! Q matrix data have been generated successfully!     **
********************************************************************************
**    The simulation results are all Pass!!                                 **
**    Get finish at cycle: 57                                               **
********************************************************************************


        ----------------------------
        --                        --       |__||
        --  Congratulations !!    --      / 0.0  |
        --                        --     /_____   |
        --  Simulation PASS!!     --   /^ ^ ^ ^ \  |
        --                        --  |^ ^ ^ ^ |w|
        ----------------------------   \m___m__|_|
```

```
$finish called from file "qr_cordic_tb.v", line 239.
$finish at simulation time                96750
           V C S   S i m u l a t i o n   R e p o r t
Time: 967500 ps
CPU Time:      0.500 seconds;       Data structure size:   0.1Mb
Sat May 25 23:15:04 2024
CPU time: 1.368 seconds to compile + .552 seconds to elab + .452 seconds to link + .556 seconds in simulation
```

Pattern 3:



Pattern4:



- Both R and Q are all pass in RTL simulation
- Clock rate : 15ns
- Cycles : 57
- Run time : 967.5ns

RTL simulation Commands (VCS):

vcs -full64 -R -sverilog qr_cordic_tb.v qr_cordic.v +access+r +vcs+fsdbon +fsdb+mda +fsdbfile+qr_cordic.fsdb

5. Synthesis

a. Synopsys Design Constraints (SDC)

```
# operating conditions and boundary conditions #

set cycle 15.0
create_clock -name clk  -period $cycle   [get_ports  clk]

set_dont_touch_network      [all_clocks]
set_fix_hold                [all_clocks]
set_clock_uncertainty  0.1  [all_clocks]
set_clock_latency      0.5  [all_clocks]
set_ideal_network           [get_ports clk]


#Don't touch the basic env setting as below
set_input_delay  1   -clock clk [remove_from_collection [all_inputs] [get_ports clk]]
set_output_delay 1    -clock clk [all_outputs]

set_load         1   [all_outputs]
set_drive        0.1   [all_inputs]

set_operating_conditions -max_library slow -max slow
set_wire_load_model -name tsmc13_wl10 -library slow
set_max_fanout 20 [all_inputs]

```

- Clock rate = 15ns
- I/O delay = 1ns

b. Design compiler synthesis .tcl

```
#Read All Files
read_file -format verilog  qr_cordic.v
current_design qr_cordic
link

#Setting Clock Constraints
source -echo -verbose qr_cordic.sdc
check_design
set high_fanout_net_threshold 0
uniquify
set_fix_multiple_port_nets -all -buffer_constants [get_designs *]
#set_max_area 0
#Synthesis all design
#compile -map_effort high -area_effort high
#compile -map_effort high -area_effort high -inc
#compile_ultra
#compile_ultra -area
compile

write -format ddc     -hierarchy -output "qr_cordic_syn.ddc"
write_sdf -version 1.0  qr_cordic_syn.sdf
write -format verilog -hierarchy -output qr_cordic_syn.v
report_area > area.log
report_timing > timing.log
report_qor    >  qr_cordic_syn.qor

```

- Use Standard synthesis run without specifying additional optimization efforts for mapping or area.

- Synthesis on the CBDK_IC_Contest_v2.5 and TSMC 130nm process

c. Timing report

```
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : qr_cordic
Version: U-2022.12
Date   : Sat May 25 21:21:51 2024
****************************************

Operating Conditions: slow   Library: slow
Wire Load Model Mode: top

  Startpoint: iter_q28_reg[0]
              (rising edge-triggered flip-flop clocked by clk)
  Endpoint: yi_mk4_q_reg[8]
            (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  qr_cordic          tsmc13_wl10           slow

  Point                                                   Incr       Path
  ---------------------------------------------------------------------------
  clock clk (rise edge)                                   0.00       0.00
  clock network delay (ideal)                             0.50       0.50
  iter_q28_reg[0]/CK (DFFRX4)                             0.00       0.50 r
  iter_q28_reg[0]/Q (DFFRX4)                              0.61       1.11 f
  Q28_inst/iter[0] (Q_16)                                 0.00       1.11 f
  Q28_inst/Q_one_iter_inst_1/iter[0] (Q_one_iter_67)      0.00       1.11 f
  Q28_inst/Q_one_iter_inst_1/U162/Y (INVX4)               0.17       1.27 r
  Q28_inst/Q_one_iter_inst_1/U122/Y (NAND2X8)             0.20       1.48 f
  Q28_inst/Q_one_iter_inst_1/U45/Y (INVX4)                0.30       1.78 r
  Q28_inst/Q_one_iter_inst_1/U11/Y (MX2X4)                0.33       2.11 f
  Q28_inst/Q_one_iter_inst_1/U61/Y (OAI221X4)             0.48       2.59 r
  Q28_inst/Q_one_iter_inst_1/sub_100/B[6] (Q_one_iter_67_DW01_sub_3)
                                                          0.00       2.59 r
  Q28_inst/Q_one_iter_inst_1/sub_100/U126/Y (CLKINVX1)
                                                          0.24       2.83 f
  Q28_inst/Q_one_iter_inst_1/sub_100/U128/Y (NOR2X2)      0.43       3.26 r
  Q28_inst/Q_one_iter_inst_1/sub_100/U163/Y (NOR2X4)      0.19       3.46 f
  Q28_inst/Q_one_iter_inst_1/sub_100/U166/Y (AOI21X4)     0.18       3.63 r
  Q28_inst/Q_one_iter_inst_1/sub_100/U165/Y (OAI21X4)     0.15       3.78 f
  Q28_inst/Q_one_iter_inst_1/sub_100/U135/Y (AOI21X1)     0.27       4.06 r
  Q28_inst/Q_one_iter_inst_1/sub_100/U185/Y (XOR2X2)      0.21       4.27 r
  Q28_inst/Q_one_iter_inst_1/sub_100/DIFF[10] (Q_one_iter_67_DW01_sub_3)
                                                          0.00       4.27 r
  Q28_inst/Q_one_iter_inst_1/U144/Y (AO22X2)              0.26       4.53 r
  Q28_inst/Q_one_iter_inst_1/U180/Y (AO21X4)              0.28       4.81 r
  Q28_inst/Q_one_iter_inst_1/xo[10] (Q_one_iter_67)       0.00       4.81 r
  Q28_inst/Q_one_iter_inst_2/xi[10] (Q_one_iter_66)       0.00       4.81 r
  Q28_inst/Q_one_iter_inst_2/U111/Y (INVX8)               0.11       4.92 f
  Q28_inst/Q_one_iter_inst_2/U28/Y (OAI222X4)             0.55       5.47 r
  Q28_inst/Q_one_iter_inst_2/U27/Y (INVX4)                0.08       5.55 f
  Q28_inst/Q_one_iter_inst_2/U168/Y (AOI2BB1X2)           0.22       5.77 f
  Q28_inst/Q_one_iter_inst_2/U21/Y (OAI221X4)             0.34       6.11 r
  Q28_inst/Q_one_iter_inst_2/sub_105/B[1] (Q_one_iter_66_DW01_sub_2)
                                                          0.00       6.11 r
  Q28_inst/Q_one_iter_inst_2/sub_105/U128/Y (INVX3)       0.10       6.21 f
  Q28_inst/Q_one_iter_inst_2/sub_105/U185/Y (NAND2X1)     0.26       6.46 r
  Q28_inst/Q_one_iter_inst_2/sub_105/U132/Y (OA21X2)      0.36       6.83 r
  Q28_inst/Q_one_iter_inst_2/sub_105/U154/Y (OA21X4)      0.23       7.06 r
  Q28_inst/Q_one_iter_inst_2/sub_105/U143/Y (OAI21X4)     0.15       7.20 f
  Q28_inst/Q_one_iter_inst_2/sub_105/U131/Y (AOI21X1)     0.28       7.48 r
  Q28_inst/Q_one_iter_inst_2/sub_105/U166/Y (XOR2X2)      0.21       7.69 r
  Q28_inst/Q_one_iter_inst_2/sub_105/DIFF[10] (Q_one_iter_66_DW01_sub_2)
                                                          0.00       7.69 r
  Q28_inst/Q_one_iter_inst_2/U130/Y (AO22X2)              0.22       7.92 r
  Q28_inst/Q_one_iter_inst_2/U177/Y (AO21X4)              0.28       8.20 r
  Q28_inst/Q_one_iter_inst_2/yo[10] (Q_one_iter_66)       0.00       8.20 r
  Q28_inst/Q_one_iter_inst_3/yi[10] (Q_one_iter_65)       0.00       8.20 r
  Q28_inst/Q_one_iter_inst_3/U19/Y (CLKINVX8)             0.16       8.35 f
  Q28_inst/Q_one_iter_inst_3/U16/Y (OR2X1)                0.36       8.71 f
  Q28_inst/Q_one_iter_inst_3/U18/Y (NAND3X4)              0.15       8.86 r
  Q28_inst/Q_one_iter_inst_3/U22/Y (INVX3)                0.10       8.96 f
  Q28_inst/Q_one_iter_inst_3/U62/Y (OAI221X4)             0.47       9.44 r
  Q28_inst/Q_one_iter_inst_3/sub_100/B[0] (Q_one_iter_65_DW01_sub_3)
                                                          0.00       9.44 r
  Q28_inst/Q_one_iter_inst_3/sub_100/U116/Y (INVX4)       0.09       9.52 f
  Q28_inst/Q_one_iter_inst_3/sub_100/U132/Y (NOR2X2)      0.21       9.73 r
  Q28_inst/Q_one_iter_inst_3/sub_100/U130/Y (OA21X4)      0.25       9.98 r
  Q28_inst/Q_one_iter_inst_3/sub_100/U170/Y (OA21X4)      0.21      10.20 r
  Q28_inst/Q_one_iter_inst_3/sub_100/U155/Y (INVX6)       0.10      10.30 f
  Q28_inst/Q_one_iter_inst_3/sub_100/U121/Y (AOI21X2)     0.19      10.49 r
  Q28_inst/Q_one_iter_inst_3/sub_100/U133/Y (XOR2X4)      0.17      10.66 f
  Q28_inst/Q_one_iter_inst_3/sub_100/DIFF[5] (Q_one_iter_65_DW01_sub_3)
                                                          0.00       10.66 f
  Q28_inst/Q_one_iter_inst_3/U8/Y (AO22X4)                0.25       10.91 f
  Q28_inst/Q_one_iter_inst_3/U183/Y (AO21X4)              0.31       11.23 f
  Q28_inst/Q_one_iter_inst_3/xo[5] (Q_one_iter_65)        0.00       11.23 f
  Q28_inst/Q_one_iter_inst_4/xi[5] (Q_one_iter_64)        0.00       11.23 f
  Q28_inst/Q_one_iter_inst_4/U66/Y (INVX1)                0.30       11.52 r
  Q28_inst/Q_one_iter_inst_4/U151/Y (OA22X1)              0.33       11.85 r
  Q28_inst/Q_one_iter_inst_4/U45/Y (OAI221X4)             0.32       12.17 f
  Q28_inst/Q_one_iter_inst_4/U44/Y (INVX4)                0.07       12.25 r
  Q28_inst/Q_one_iter_inst_4/U108/Y (OAI221X4)            0.36       12.60 f
  Q28_inst/Q_one_iter_inst_4/sub_105/B[1] (Q_one_iter_64_DW01_sub_2)
                                                          0.00       12.60 f
  Q28_inst/Q_one_iter_inst_4/sub_105/U136/Y (INVX3)       0.13       12.73 r
  Q28_inst/Q_one_iter_inst_4/sub_105/U162/Y (NOR2X4)      0.10       12.84 f
  Q28_inst/Q_one_iter_inst_4/sub_105/U137/Y (OAI21X2)     0.35       13.18 r
  Q28_inst/Q_one_iter_inst_4/sub_105/U165/Y (AOI21X4)     0.21       13.39 f
  Q28_inst/Q_one_iter_inst_4/sub_105/U164/Y (OAI21X4)     0.27       13.66 r
  Q28_inst/Q_one_iter_inst_4/sub_105/U113/Y (XNOR2X2)     0.24       13.90 f
  Q28_inst/Q_one_iter_inst_4/sub_105/DIFF[8] (Q_one_iter_64_DW01_sub_2)
                                                          0.00       13.90 f
  Q28_inst/Q_one_iter_inst_4/U78/Y (AO22X4)               0.21       14.11 f
  Q28_inst/Q_one_iter_inst_4/U171/Y (AO21X1)              0.30       14.42 f
  Q28_inst/Q_one_iter_inst_4/yo[8] (Q_one_iter_64)        0.00       14.42 f
  Q28_inst/U117/Y (AO22X1)                                0.42       14.84 f
  Q28_inst/yo[8] (Q_16)                                   0.00       14.84 f
  U5086/Y (CLKINVX3)                                      0.14       14.98 r
  U7140/Y (OAI221X2)                                      0.16       15.13 f
  yi_mk4_q_reg[8]/D (DFFSRHQX1)                           0.00       15.13 f
  data arrival time                                                  15.13

  clock clk (rise edge)                                  15.00       15.00
  clock network delay (ideal)                             0.50       15.50
  clock uncertainty                                      -0.10       15.40
  yi_mk4_q_reg[8]/CK (DFFSRHQX1)                          0.00       15.40 r
  library setup time                                     -0.27       15.13
  data required time                                                 15.13
  ---------------------------------------------------------------------------
  data required time                                                 15.13
  data arrival time                                                 -15.13
  ---------------------------------------------------------------------------
  slack (MET)                                                        0.00
```

- Clock rate = 15ns
- Slack ≧ 0ns

d. Area report

```
****************************************
Report : area
Design : qr_cordic
Version: U-2022.12
Date   : Sat May 25 21:21:51 2024
****************************************

Library(s) Used:

    slow (File: /home/cell_library/CBDK_IC_Contest_v2.5/SynopsysDC/db/slow.db)

Number of ports:                        39008
Number of nets:                        145079
Number of cells:                       107729
Number of combinational cells:         103742
Number of sequential cells:              2096
Number of macros/black boxes:               0
Number of buf/inv:                      25537
Number of references:                     191

Combinational area:            1187614.757948
Buf/Inv area:                   176923.394568
Noncombinational area:           79423.041847
Macro/Black Box area:                0.000000
Net Interconnect area:        11491491.069855

Total cell area:               1267037.799795
Total area:                   12758528.869650
1
```

- Area: 12567037.8 μm² (clock rate:15ns, standard compile)

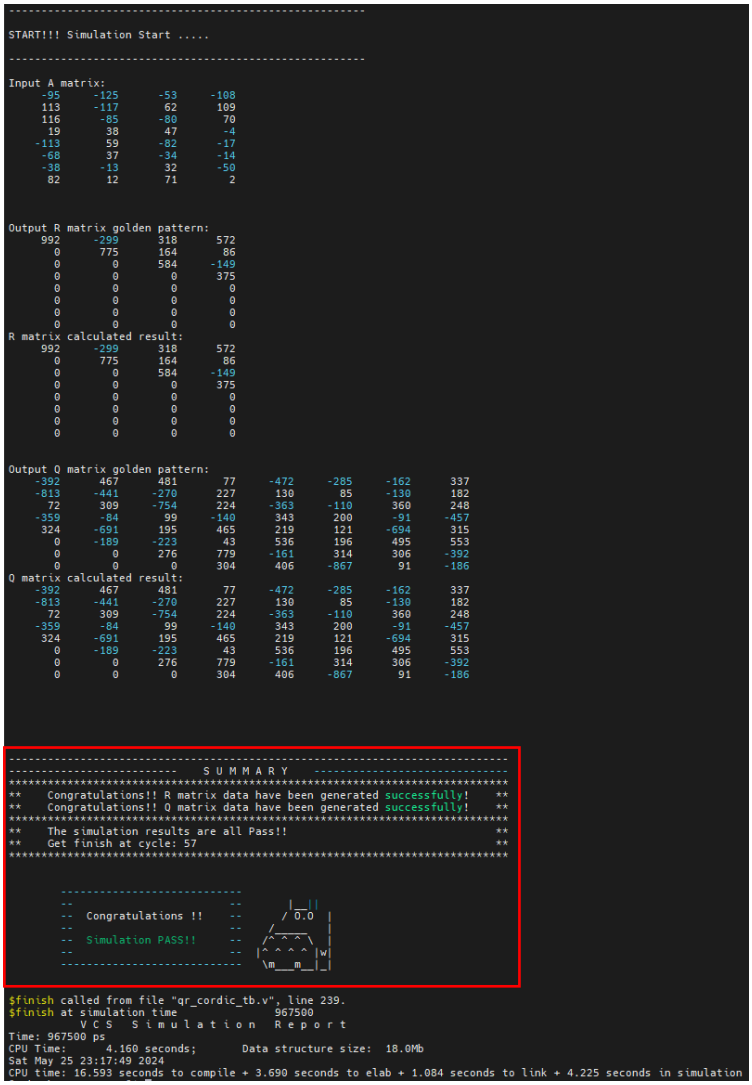
e. Power

```
Operating Conditions: slow   Library: slow
Wire Load Model Mode: top

Design        Wire Load Model            Library
------------------------------------------------
qr_cordic             tsmc13_wl10       slow


Global Operating Voltage = 1.08
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000pf
    Time Units = 1ns
    Dynamic Power Units = 1mW    (derived from V,C,T units)
    Leakage Power Units = 1pW


Attributes
----------
i - Including register clock pin internal power


  Cell Internal Power  =   4.4245 mW   (63%)
  Net Switching Power  =   2.5960 mW   (37%)
                         ---------
Total Dynamic Power    =   7.0205 mW  (100%)

Cell Leakage Power     =   1.0694 mW


                 Internal         Switching           Leakage            Total
Power Group      Power            Power               Power              Power   (   %    )  Attrs
--------------------------------------------------------------------------------------------------
io_pad             0.0000           0.0000              0.0000             0.0000  (   0.00%)
memory             0.0000           0.0000              0.0000             0.0000  (   0.00%)
black_box          0.0000           0.0000              0.0000             0.0000  (   0.00%)
clock_network      3.5668           0.0000              0.0000             3.5668  (  44.09%)  i
register       6.8604e-02       8.5020e-02        7.0488e+07             0.2241  (   2.77%)
sequential         0.0000           0.0000              0.0000             0.0000  (   0.00%)
combinational      0.7891           2.5108        9.9895e+08             4.2990  (  53.14%)
--------------------------------------------------------------------------------------------------
Total              4.4245 mW        2.5959 mW     1.0694e+09 pW          8.0899 mW
1
```

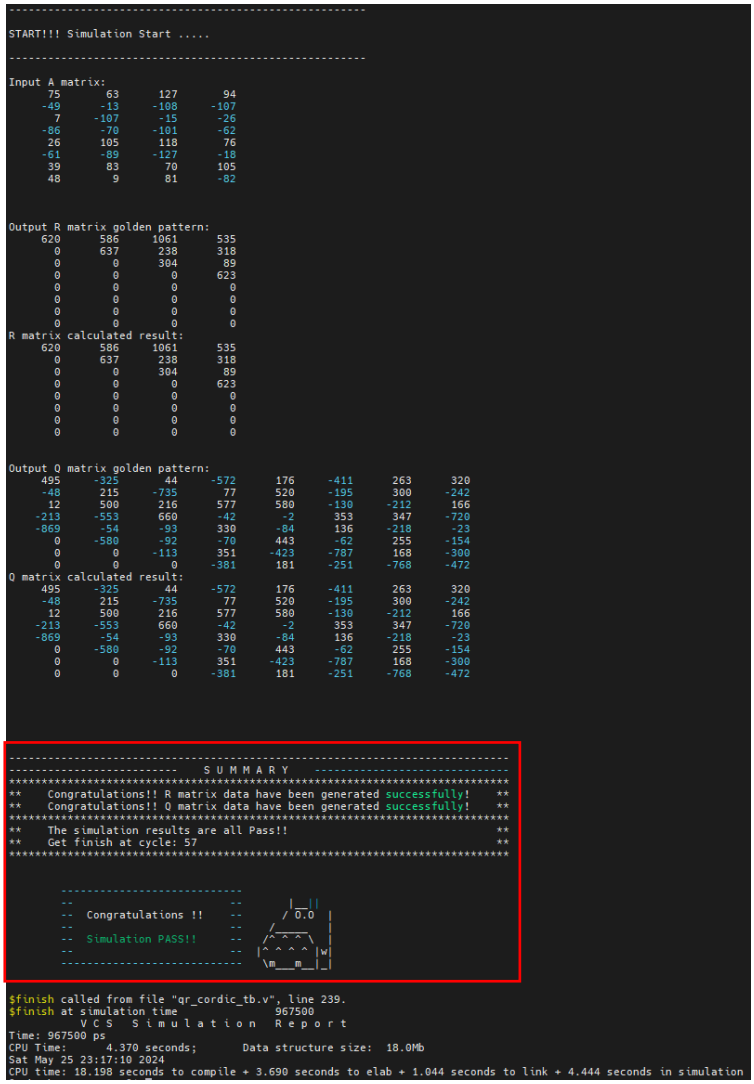
- Total Power : 8.0899mW

f. Gate-level simulation (Run time : 1117.5ns)

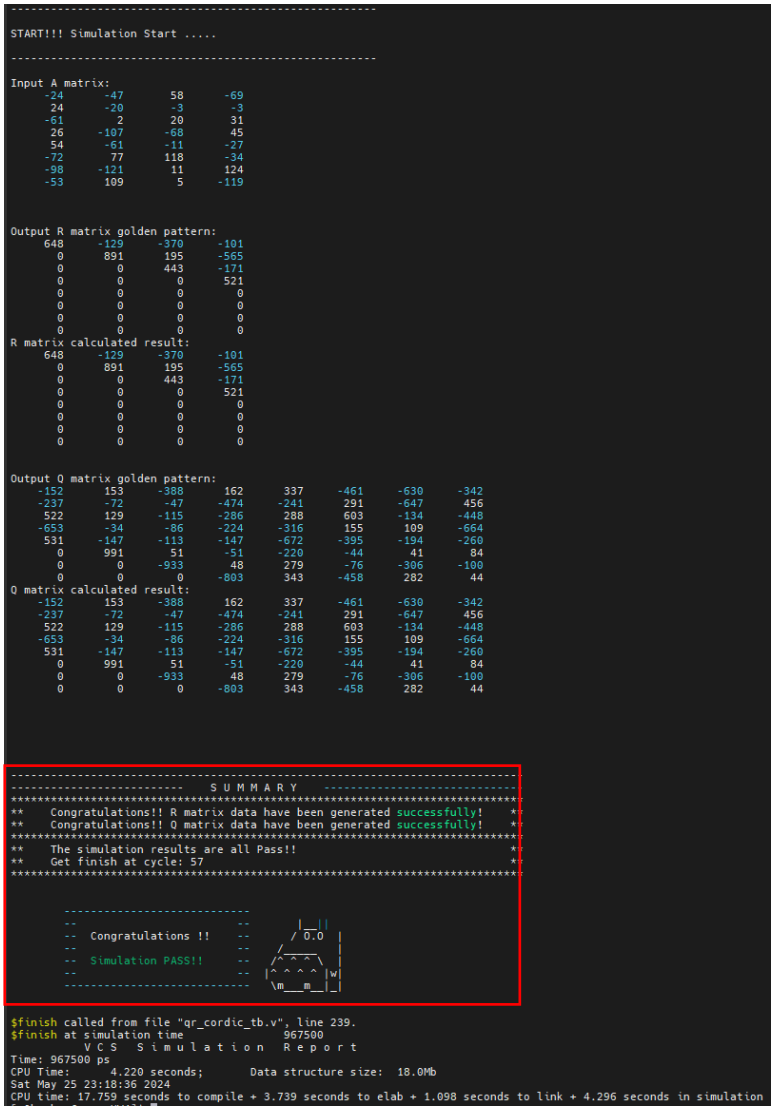
Pattern 1:



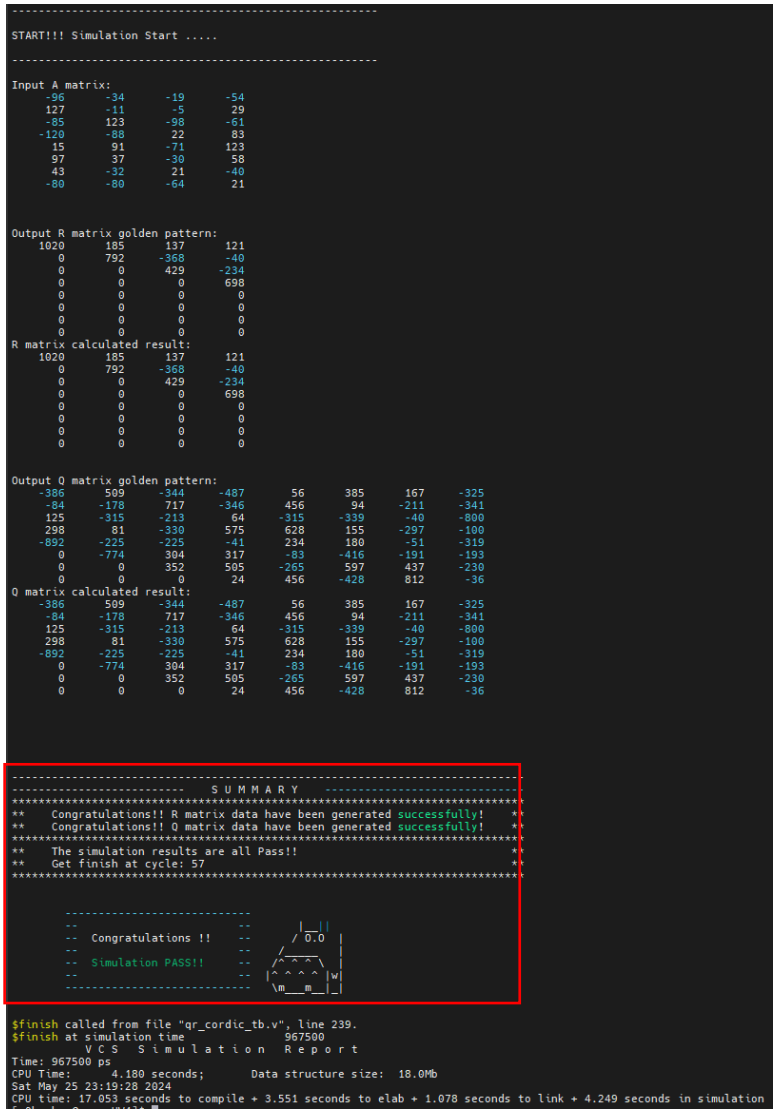
Pattern 2:



Pattern 3:



Pattern4:



- Both R and Q are all pass in gate-level simulation
- Clock rate : 15ns
- Cycles : 57
- Run time : 967.5ns

Gate-level simulation Commands (VCS):
vcs -full64 -R -sverilog qr_cordic_tb.v qr_cordic_syn.v +define+SDF +access+r +neg_tchk +vcs+fsdbon +fsdb+mda +fsdbfile+qr_cordic.fsdb -v /home/cell_library/CBDK_IC_Contest_v2.5/Verilog/tsmc13_neg.v +maxdelays

- The complete code and related data are placed on [GitHub]