VLSI DPS HW#4

電機四 **4109064119** 陳柏淳

1. Calculate the Frobenius Distance

```
Matrix A :
  -104    -4  -119   -85
   -57    76    89    52
    12   -92   111  -120
   117   -21    45   -58
   119   106    65  -117
   -88    74    62  -104
   120   117   -28    82
   117    39    39    49
```

- Generate a random matrix A

```
Matrix Q_float :
   -0.3726   -0.2042    0.0430    0.4192    0.4264   -0.3153    0.4300    0.4192
    0.0918    0.4283   -0.4575   -0.2266    0.3851    0.4517    0.4372    0.0634
   -0.4671    0.4719    0.5443    0.1017    0.1681    0.3738   -0.2847    0.0594
   -0.5104    0.3772   -0.2673   -0.2421   -0.5552   -0.3311    0.1386    0.1732
    0.6116    0.4864    0.2875    0.1650   -0.1330   -0.2509    0.0945    0.4358
         0    0.4163   -0.2190    0.3501    0.2810   -0.4499   -0.1521   -0.5930
         0         0    0.5390   -0.3943    0.0328   -0.1821    0.5809   -0.4271
         0         0         0   -0.6314    0.4854   -0.3856   -0.3927    0.2505

Matrix Q_fix :
   -0.3750   -0.2031    0.0361    0.4150    0.4258   -0.3213    0.4258    0.4189
   -0.0918    0.4287   -0.4551   -0.2295    0.3867    0.4561    0.4365    0.0615
   -0.4697   -0.4756    0.5391    0.0977    0.1680    0.3701   -0.2900    0.0508
   -0.5117   -0.3867   -0.0898   -0.2422   -0.5537   -0.3271    0.1426    0.1729
    0.6084   -0.4941    0.4355    0.1631   -0.1367   -0.2598    0.0908    0.4287
         0   -0.4121   -0.0605    0.3584    0.2900   -0.4424   -0.1523   -0.5879
         0         0   -0.5449   -0.3916    0.0332   -0.1836    0.5840   -0.4258
         0         0         0   -0.6318    0.4863   -0.3857   -0.3916    0.2441
```

```
Matrix R_float :
  279.0914   61.7289   62.2807   30.2840
    0.0000  206.9023    9.4802   29.4226
   -0.0000   -0.0000  206.9619  -85.9580
   -0.0000    0.0000   -0.0000  228.3668
    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 :
  280.5000   61.0000   61.0000   30.0000
         0  208.2500    9.0000   28.5000
         0         0  207.7500  -86.7500
         0         0         0  230.2500
         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)'))

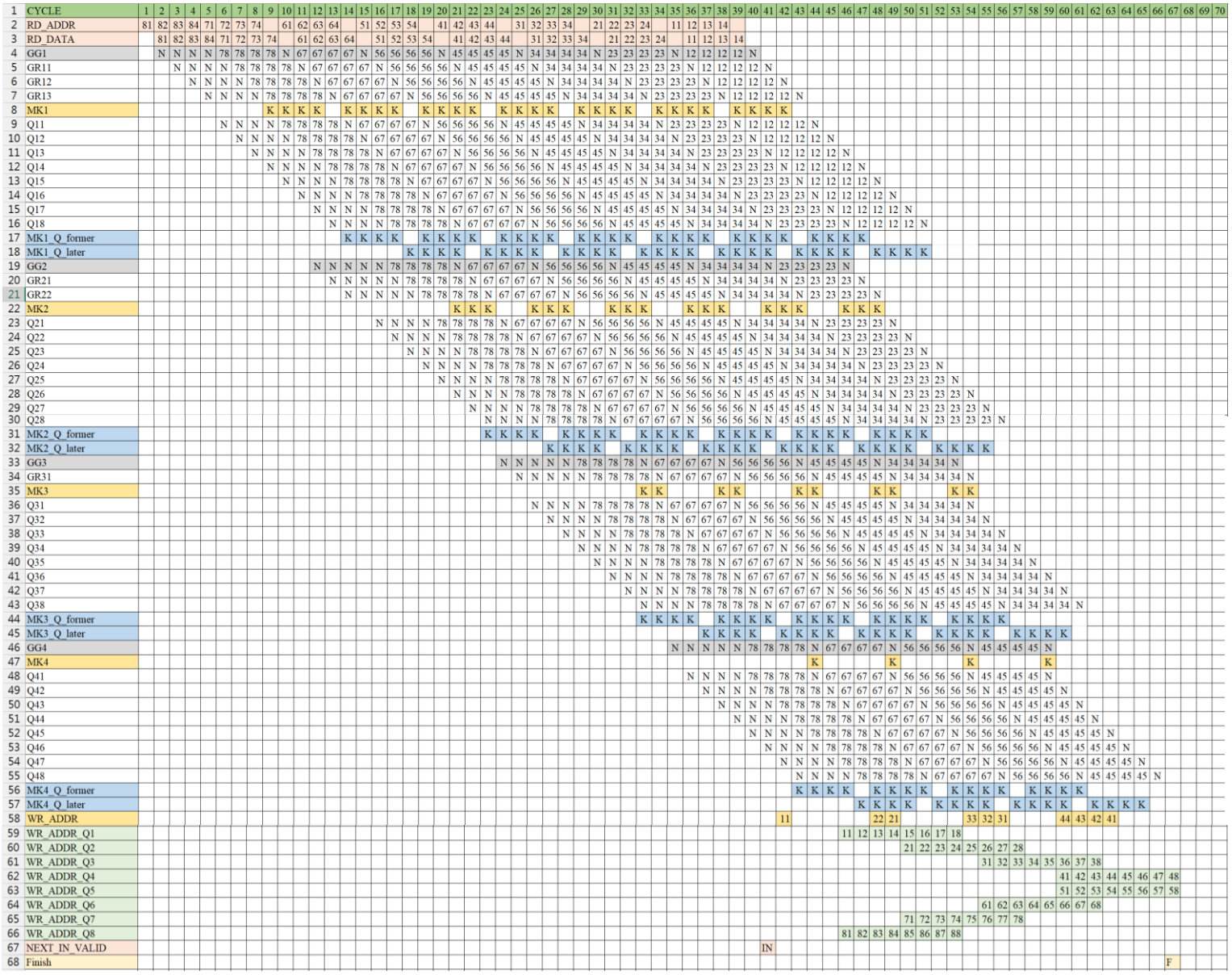
```
------------------------------------------------------------
------------------- Hardware implementation loss -----------
------------------------------------------------------------
Q Fix Point Loss :
    2.1017

R Fix Point Loss :
    3.4539
```

Q_loss : 2.1017
R_loss : 3.4539

- 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



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

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: 67                                                 **
********************************************************************************
```

67 cycles

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	37	38	39	40	41	42	43	44
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	78	N	67	67	67	67	N	56	56	56	56	N	45	45	45	45	N	34	34	34	34	N	23	23	23	23	N	12	12	12	12	N				
5	GR11			N	N	N	N	78	78	78	78	N	67	67	67	67	N	56	56	56	56	N	45	45	45	45	N	34	34	34	34	N	23	23	23	23	N	12	12	12	12	N				
6	GR12				N	N	N	N	78	78	78	78	N	67	67	67	67	N	56	56	56	56	N	45	45	45	45	N	34	34	34	34	N	23	23	23	23	N	12	12	12	12	N				
7	GR13					N	N	N	N	78	78	78	78	N	67	67	67	67	N	56	56	56	56	N	45	45	45	45	N	34	34	34	34	N	23	23	23	23	N	12	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				

When GG1 performs all vectoring over, the next input is valid.
The initiation interval is 41 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: X = 594; Y = -5594
GR11 Iteration 4 times: X = 108; Y = 43
GR12 Iteration 4 times: X = 331; Y = -390
GR13 Iteration 4 times: 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: X = 596; Y = 1596
GR11 Iteration 8 times: X = 107; Y = 45
GR12 Iteration 8 times: X = 334; Y = -388
GR13 Iteration 8 times: 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 1 0 0
GG1  Iteration 12 times: X = 597; Y = 0597
GR11 Iteration 12 times: X = 107; Y = 45
GR12 Iteration 12 times: X = 333; Y = -389
GR13 Iteration 12 times: X = 151; Y = 293
Q11  Iteration 12 times: input: X =     0; Y =     0 output: X =     0; Y =     0
Q12  Iteration 12 times: input: X =     0; Y =     0 output: X =     0; Y =     0
Q13  Iteration 12 times: input: X =     0; Y =     0 output: X =     0; Y =     0
Q14  Iteration 12 times: input: X =     0; Y =     0 output: X =     0; Y =     0
Q15  Iteration 12 times: input: X =     0; Y =     0 output: X =     0; Y =     0
Q16  Iteration 12 times: input: X =     0; Y =     0 output: X =     0; Y =     0
Q17  Iteration 12 times: input: X =  -711; Y = -1528 output: X =  -714; Y = -1527
Q18  Iteration 12 times: input: X =  1528; Y =  -708 output: X =  1527; Y =  -711

GG1  Multiplied by K:   X = 362; Y = 0
GR11 Multiplied by K:   X = -500; Y = -468
GR12 Multiplied by K:   X = -212; Y = 248
GR13 Multiplied by K:   X = -432; Y = 436
Q11 Multiplied by K:   X = 0; Y = 0
Q12 Multiplied by K:   X = 0; Y = 0
Q13 Multiplied by K:   X = 0; Y = 0
Q14 Multiplied by K:   X = 0; Y = 0
Q15 Multiplied by K:   X = 0; Y = 0
Q16 Multiplied by K:   X = 0; Y = 0
Q17 Multiplied by K:   X = -434; Y = -928
Q18 Multiplied by K:   X = 927; Y = -432
k = 1 row67:
```

```
k = 4 row45:

d = -1 1 -1 -1
GG4  Iteration 4 times: X = 608; Y = 40608
Q41  Iteration 4 times: input: X =  -488; Y =     0 output: X =  -618; Y =   511
Q42  Iteration 4 times: input: X =   399; Y =  -567 output: X =   -89; Y = -1134
Q43  Iteration 4 times: input: X =   -62; Y =   213 output: X =   144; Y =   335
Q44  Iteration 4 times: input: X =  -411; Y =   233 output: X =  -277; Y =   726
Q45  Iteration 4 times: input: X =    96; Y =   397 output: X =   536; Y =   403
Q46  Iteration 4 times: input: X =    49; Y =   233 output: X =   305; Y =   245
Q47  Iteration 4 times: input: X =   402; Y =  -577 output: X =   -96; Y = -1151
Q48  Iteration 4 times: input: X =  -557; Y =   -66 output: X =  -775; Y =   499

d = -1 -1 1 1
GG4  Iteration 8 times: X = 612; Y = -4612
Q41  Iteration 8 times: input: X =  -618; Y =   511 output: X =  -582; Y =   555
Q42  Iteration 8 times: input: X =   -89; Y = -1134 output: X =  -169; Y = -1129
Q43  Iteration 8 times: input: X =   144; Y =   335 output: X =   167; Y =   324
Q44  Iteration 8 times: input: X =  -277; Y =   726 output: X =  -225; Y =   746
Q45  Iteration 8 times: input: X =   536; Y =   403 output: X =   565; Y =   365
Q46  Iteration 8 times: input: X =   305; Y =   245 output: X =   323; Y =   223
Q47  Iteration 8 times: input: X =   -96; Y = -1151 output: X =  -177; Y = -1145
Q48  Iteration 8 times: input: X =  -775; Y =   499 output: X =  -739; Y =   554

d = 1 1 1 1
GG4  Iteration 12 times: X = 616; Y = -1616
Q41  Iteration 12 times: input: X =  -582; Y =   555 output: X =  -585; Y =   548
Q42  Iteration 12 times: input: X =  -169; Y = -1129 output: X =  -158; Y = -1133
Q43  Iteration 12 times: input: X =   167; Y =   324 output: X =   166; Y =   324
Q44  Iteration 12 times: input: X =  -225; Y =   746 output: X =  -228; Y =   742
Q45  Iteration 12 times: input: X =   565; Y =   365 output: X =   564; Y =   368
Q46  Iteration 12 times: input: X =   323; Y =   223 output: X =   323; Y =   224
Q47  Iteration 12 times: input: X =  -177; Y = -1145 output: X =  -166; Y = -1149
Q48  Iteration 12 times: input: X =  -739; Y =   554 output: X =  -742; Y =   547

GG4  Multiplied by K:   X = 376; Y = -1
Q41 Multiplied by K:   X = -356; Y = 330
Q42 Multiplied by K:   X = -94; Y = -691
Q43 Multiplied by K:   X = 100; Y = 196
Q44 Multiplied by K:   X = -139; Y = 448
Q45 Multiplied by K:   X = 342; Y = 223
Q46 Multiplied by K:   X = 196; Y = 136
Q47 Multiplied by K:   X = -99; Y = -701
Q48 Multiplied by K:   X = -451; Y = 329
```

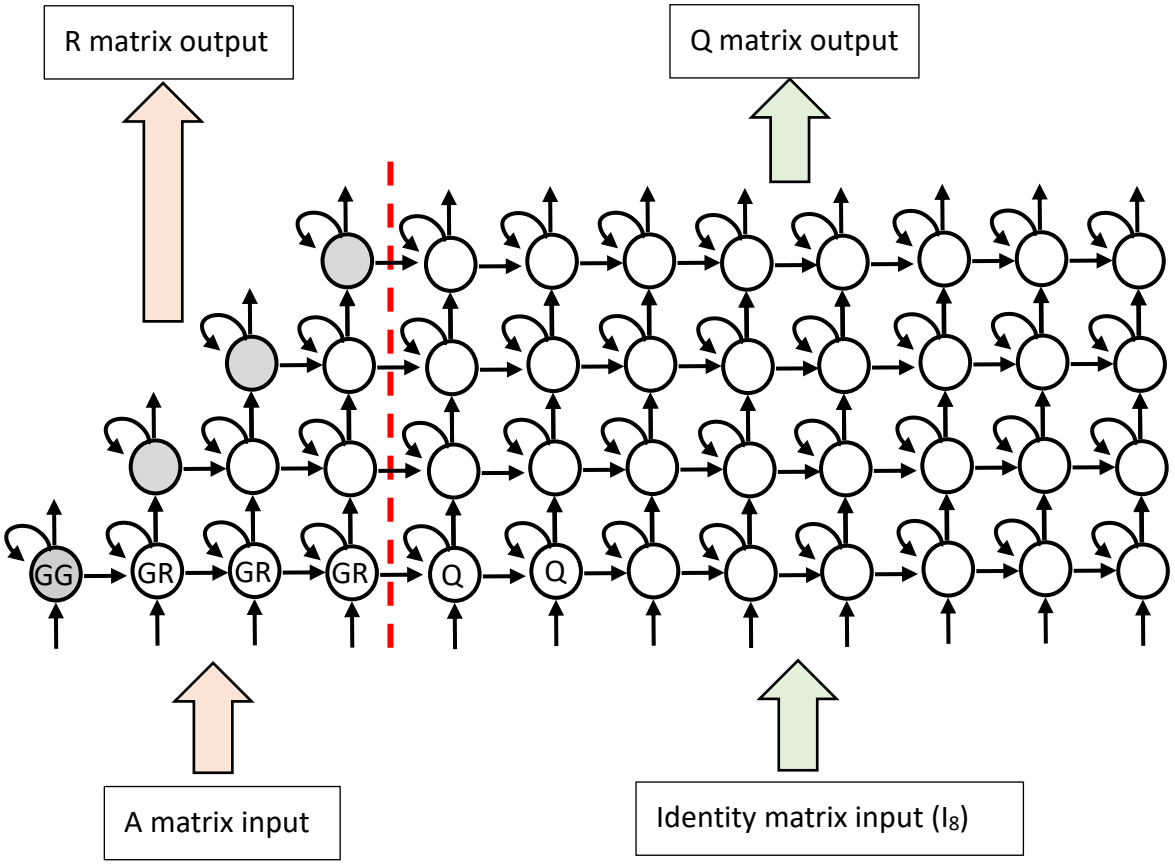
Matlab code (only core part):

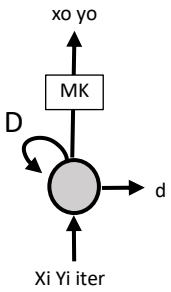
```
% Fixed point QR factorization with the CORDIC
function [Q_cordic, R_cordic] = Cordic_QR(K_cordic, Q_scaled, R_scaled, row_R, col_R, col_Q, iter_num, R_sign, R_len, R_frac,
Q_sign, Q_len, Q_frac)
    F = fimath('RoundingMethod','Floor');
    Q_cordic = Q_scaled;
    R_cordic = R_scaled;

    % 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
```

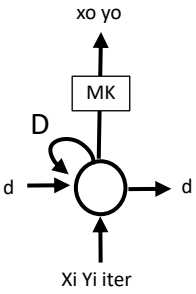
- **Hardware implementation**

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





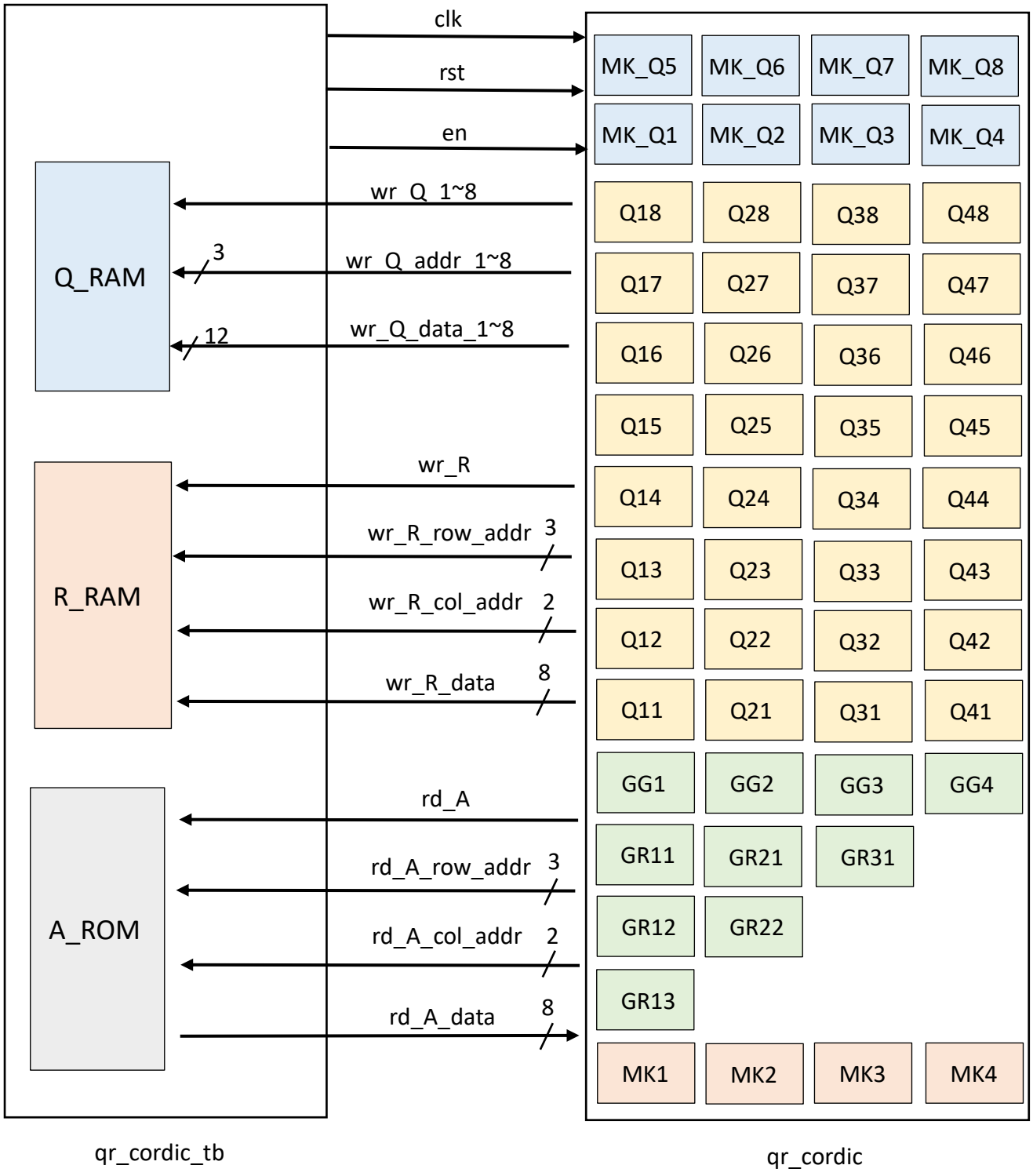
GG(vectoring mode) : One GG contains 4 micro-rotation modules, when iteration over, it outputs to above and multiplied by K(MK module), the rest of data will be transmitted to the right GR and Q modules.



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

```
module GG_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,
    output                 [1:0]                d,
    output   reg    signed [R_LEN-1:0]      xo,
    output   reg    signed [R_LEN-1:0]      yo
);

assign d = (yi == 'd0) ? 'd2 : xi[R_LEN-1] ^ yi[R_LEN-1];

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
```

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 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

```
1   // iteration times
2   always @(posedge clk or posedge rst) begin
3       if (rst) begin
4           iter_gg1 <= 'd0;
5       end
6       else if(ROT_wire) begin
7           if(nop_gg1) begin
8               iter_gg1 <= 'd0;
9           end
10          else if(iter_last_gg1) begin
11              iter_gg1 <= iter_gg1 + 'd1;
12          end
13          else begin
14              iter_gg1 <= iter_gg1 + ITER_ONE_CYCLE;
15          end
16      end
17      else begin
18          iter_gg1 <= 'd0;
19      end
20  end
21
22  // GG1 input data xi, yi
23  always @(posedge clk or posedge rst) begin
24      if (rst) begin
25          xi_gg1 <= 'd0;
26          yi_gg1 <= 'd0;
272     end
829     else if(OP_wire) begin
303         case(iter_gg1)
132             0: begin
333                 if(start_gg1) begin
435                     xi_gg1 <= 'd0;
363                     yi_gg1 <= rd_A_data_ext;
738                 end
394                 else if(nop_gg1 && !finish_gg1) begin
041                     xi_gg1 <= rd_A_data_ext;
424                     yi_gg1 <= yo_gg1;
344                 end
454                 else begin
647                     xi_gg1 <= xo_gg1;
484                     yi_gg1 <= yo_gg1;
950                 end
515             end
253             ITER_K: begin
545                 if(finish_gg1) begin
```

```
556                         xi_gg1 <= xo_gg1;
575                         yi_gg1 <= yo_gg1;
859                     end
606                     else begin
162                         xi_gg1 <= rd_A_data_ext;
636                         yi_gg1 <= xo_mk1;
465                     end
666                 end
768                 default: begin
697                     xi_gg1 <= xo_gg1;
071                     yi_gg1 <= yo_gg1;
727                 end
374             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

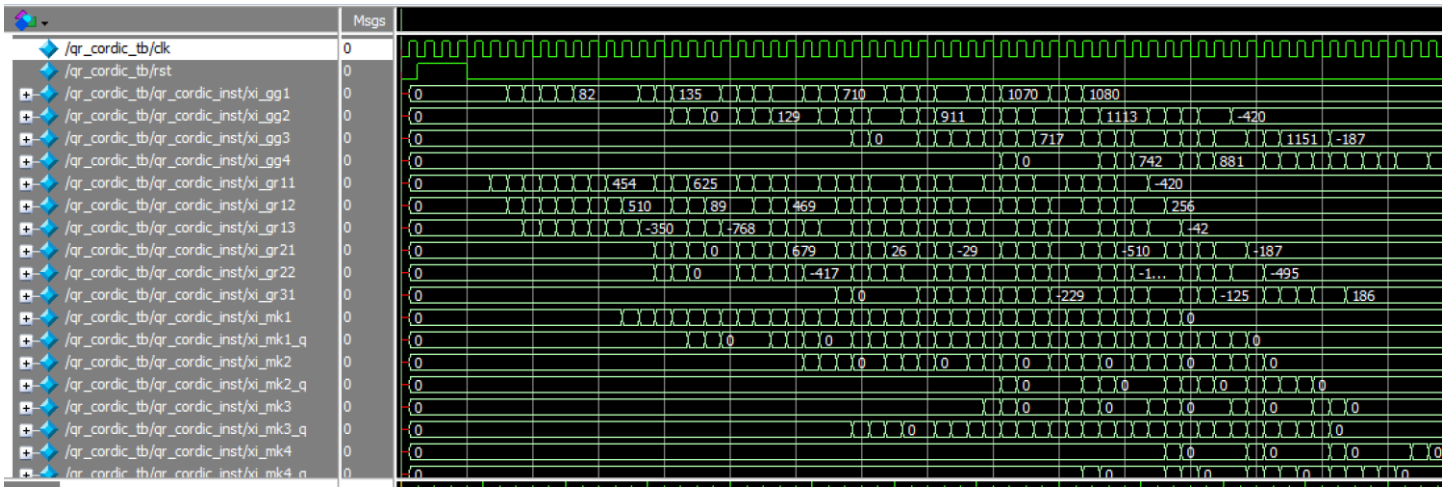
```
1   // data propagated from left to right
2   always @(posedge clk or posedge rst) begin
3       if (rst) begin
4           iter_gr11     <= 'd0;
5           nop_gr11      <= 'd0;
6           d1_gr11       <= 'd0;
7           d2_gr11       <= 'd0;
8           d3_gr11       <= 'd0;
9           d4_gr11       <= 'd0;
10          neg_gr11      <= 'd0;
11          mk_count_gr11 <= 'd0;
12      end
13      else begin
14          iter_gr11     <= iter_gg1;
15          nop_gr11      <= nop_gg1;
16          d1_gr11       <= d1_gg1;
17          d2_gr11       <= d2_gg1;
18          d3_gr11       <= d3_gg1;
19          d4_gr11       <= d4_gg1;
20          neg_gr11      <= neg_gg1;
21          mk_count_gr11 <= mk_count_gg1;
22      end
23  end
24
25  // GR11 input data xi, yi
26  always @(posedge clk or posedge rst) begin
272     if (rst) begin
829         xi_gr11 <= 'd0;
303         yi_gr11 <= 'd0;
132     end
333     else if(OP_wire) begin
435         case(iter_gr11)
363             0: begin
738                 if(start_gr11_reg) begin
394                     xi_gr11 <= 'd0;
041                     yi_gr11 <= rd_A_data_ext;
424                 end
344                 else if(nop_gr11 && !finish_gr11) begin
454                     xi_gr11 <= rd_A_data_ext;
647                     yi_gr11 <= yo_gr11;
484                 end
950                 else begin
515                     xi_gr11 <= xo_gr11;
253                     yi_gr11 <= yo_gr11;
545                 end
```

```
556                 end
575                 ITER_K: begin
859                     if(finish_gr11) begin
606                         xi_gr11 <= xo_mk1;
162                         yi_gr11 <= yo_mk1;
636                     end
465                     else begin
666                         xi_gr11 <= rd_A_data_ext;
  7                         yi_gr11 <= xo_mk1;
                        end
                end
                default: begin
                        xi_gr11 <= xo_gr11;
                        yi_gr11 <= yo_gr11;
                end
            endcase
        end
        else begin
            xi_gr11 <= 'd0;
            xi_gr11 <= 'd0;
        end
    end
```

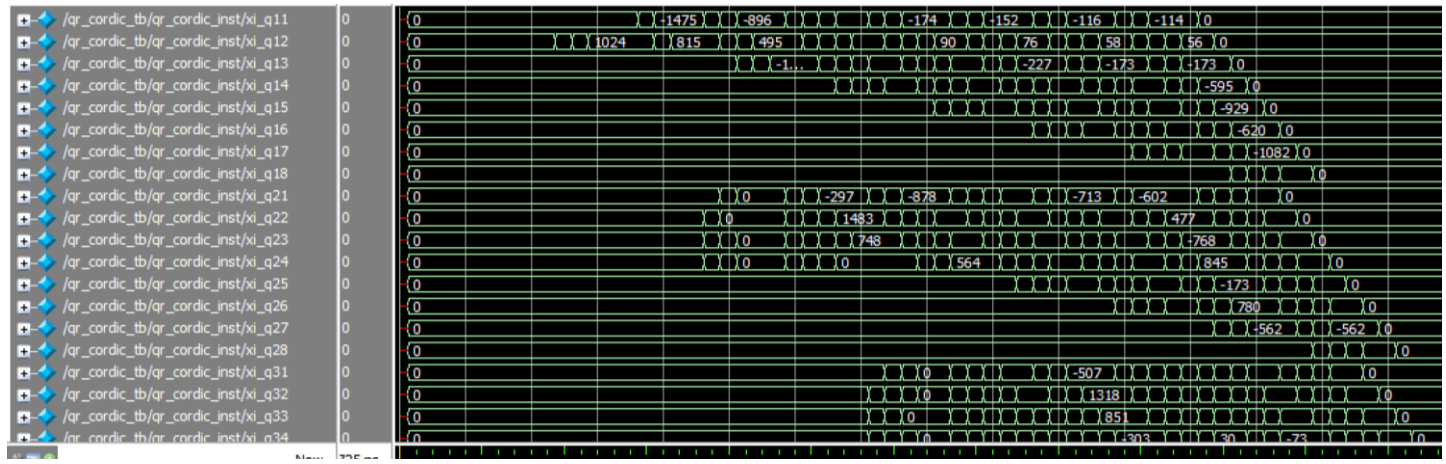
4. RTL Simulation

a. Simulation waveform

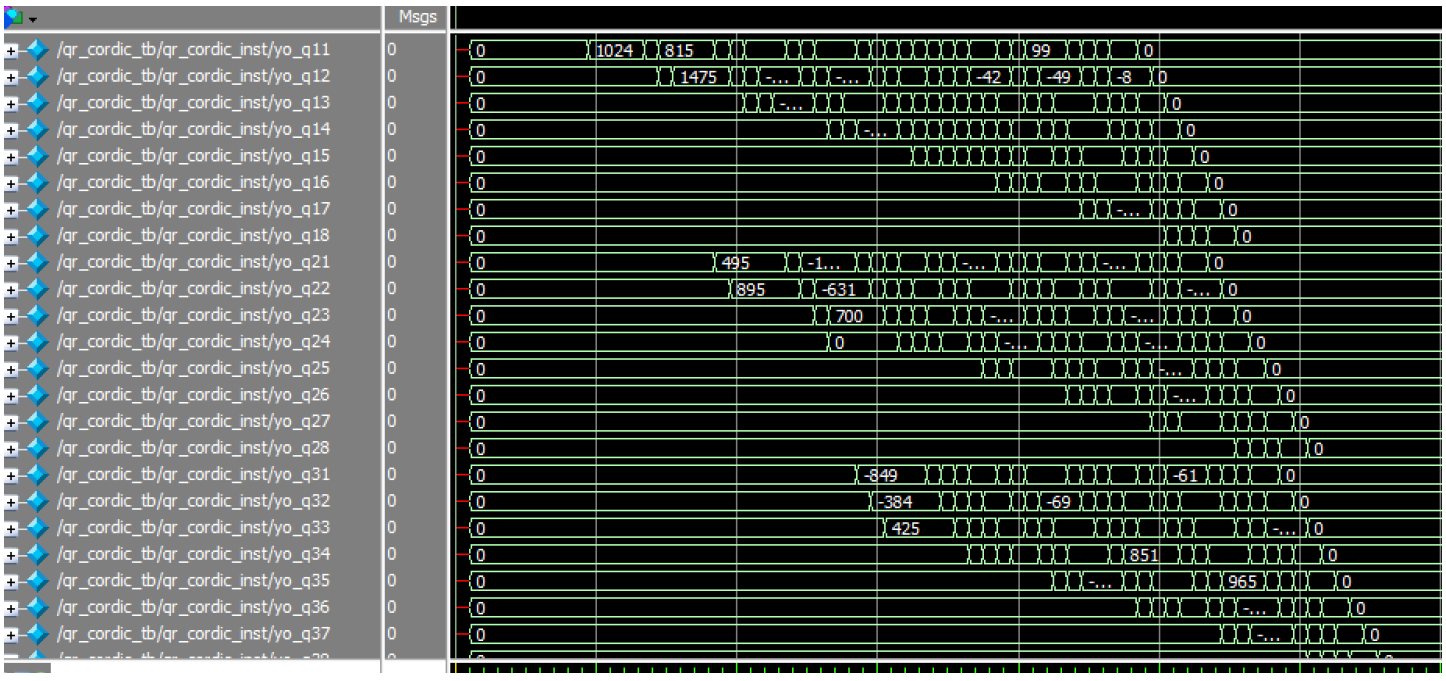
- Data propagate form left to right PE(GG, GR, Q)



- Process matrix R

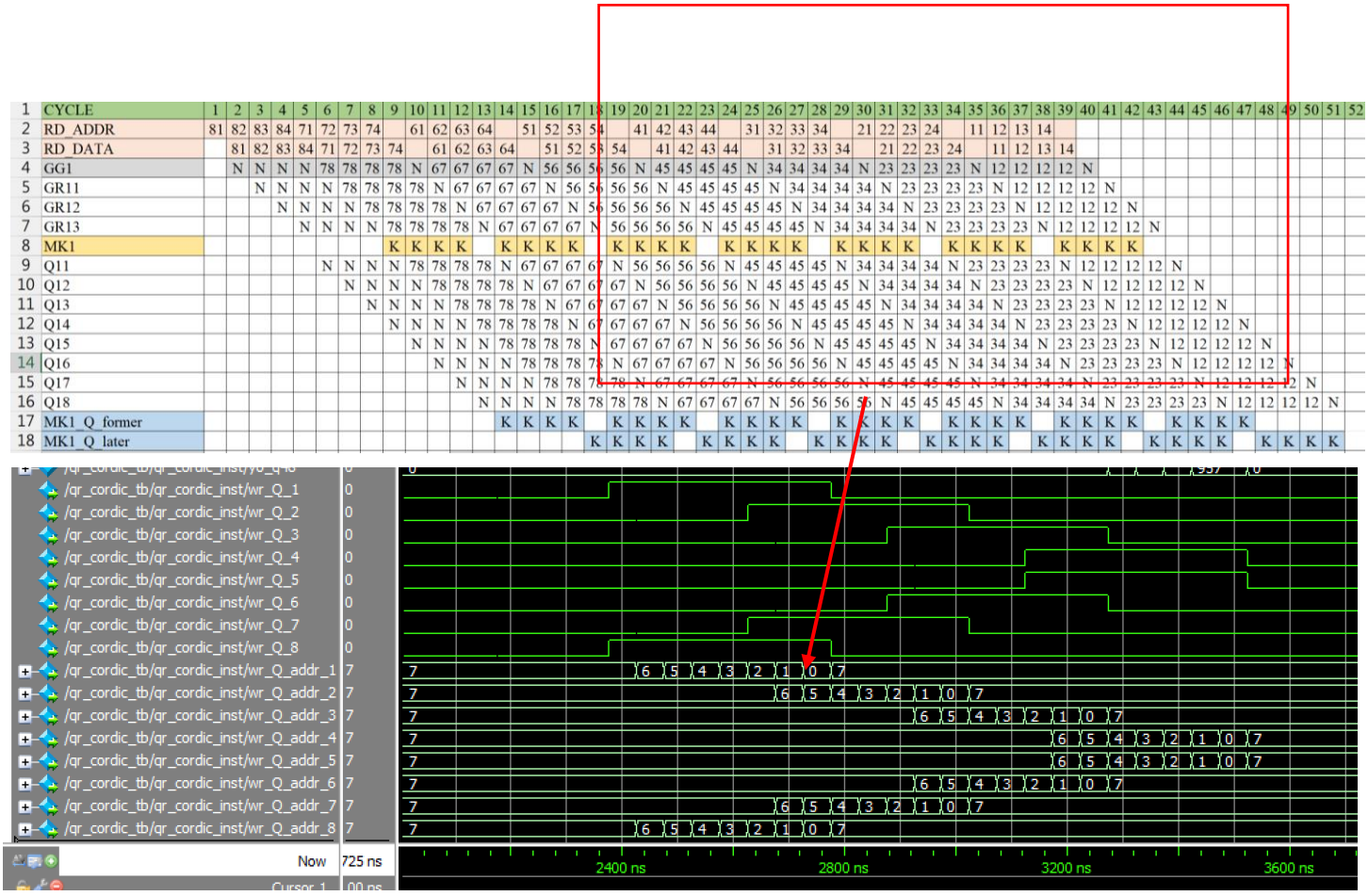


- Process matrix Q



- 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
- timing diagram

- 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.

- Timing diagram

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

b. Simulation transcript (4 patterns)

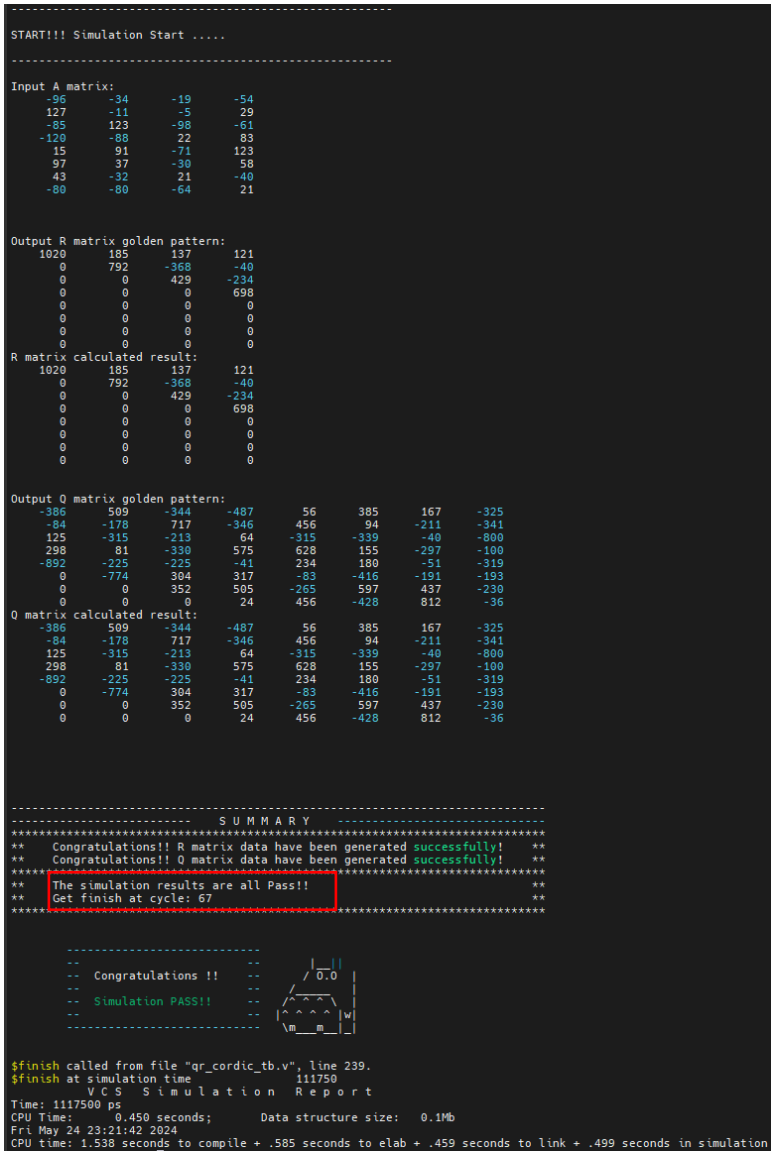
Pattern 1:

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

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: 67                                               **
******************************************************************************

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


$finish called from file "qr_cordic_tb.v", line 239.
$finish at simulation time              111750
           V C S   S i m u l a t i o n   R e p o r t
Time: 1117500 ps
CPU Time:      0.510 seconds;       Data structure size:   0.1Mb
Fri May 24 23:14:06 2024
CPU time: 1.348 seconds to compile + .556 seconds to elab + .452 seconds to link + .558 seconds in simulation
```

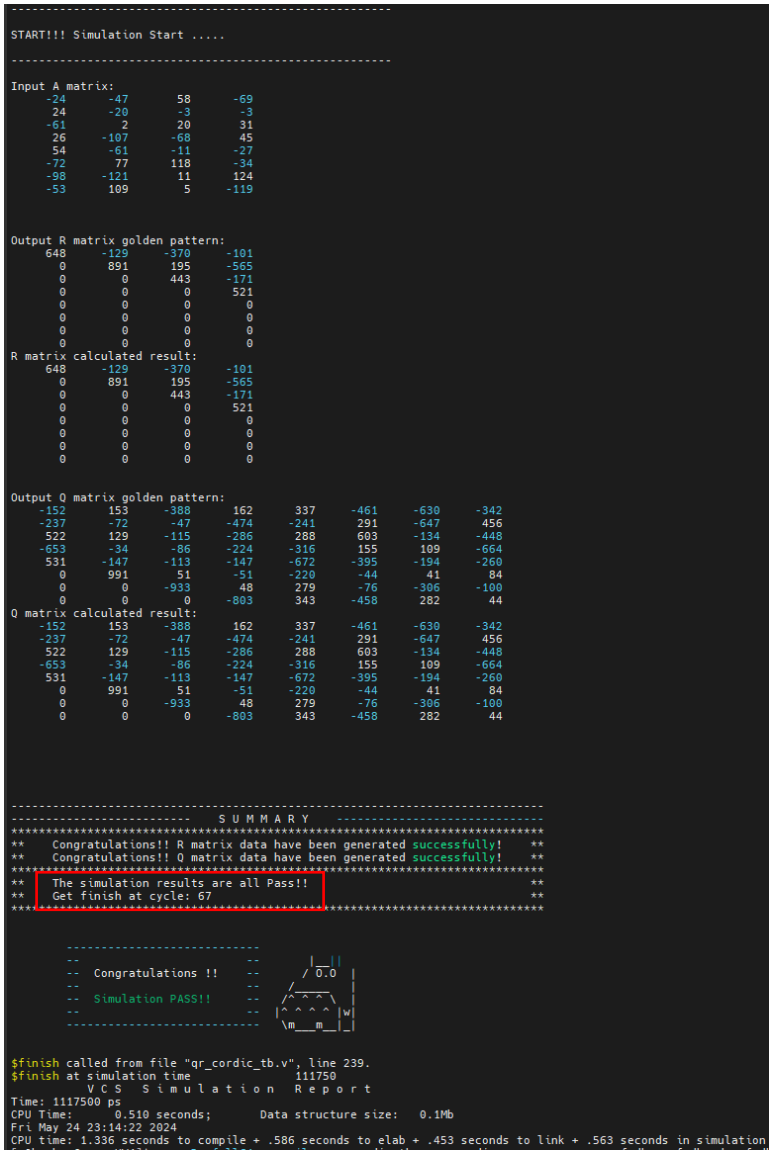
Pattern 2:

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

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: 67                                               **
******************************************************************************

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


$finish called from file "qr_cordic_tb.v", line 239.
$finish at simulation time              111750
           V C S   S i m u l a t i o n   R e p o r t
Time: 1117500 ps
CPU Time:      0.510 seconds;       Data structure size:   0.1Mb
Fri May 24 23:14:13 2024
CPU time: 1.365 seconds to compile + .552 seconds to elab + .436 seconds to link + .560 seconds in simulation
```

Pattern 3:



Pattern4:



- Both R and Q are all pass in RTL simulation
- Clock rate : 15ns
- Cycles : 67
- Run time : 1117.5ns

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   : Fri May 24 23:07:08 2024
****************************************

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

  Startpoint: xi_q18_reg[0]
              (rising edge-triggered flip-flop clocked by clk)
  Endpoint: xi_q18_reg[5]
            (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
  xi_q18_reg[0]/CK (DFFRX4)                               0.00       0.50 r
  xi_q18_reg[0]/Q (DFFRX4)                                0.52       1.02 f
  Q18_inst/xi[0] (Q_24)                                   0.00       1.02 f
  Q18_inst/U124/Y (NAND2BX4)                              0.18       1.20 f
  Q18_inst/U85/Y (NAND2BX4)                               0.16       1.36 f
  Q18_inst/U69/Y (NAND2BX4)                               0.16       1.53 f
  Q18_inst/U55/Y (NAND2BX4)                               0.17       1.70 f
  Q18_inst/U54/Y (NAND2BX4)                               0.17       1.87 f
  Q18_inst/U66/Y (NAND2X1)                                0.20       2.07 r
  Q18_inst/U38/Y (CLKXOR2X4)                              0.47       2.54 r
  Q18_inst/Q_one_iter_inst_1/xi[6] (Q_one_iter_99)        0.00       2.54 r
  Q18_inst/Q_one_iter_inst_1/U110/Y (INVX6)               0.12       2.66 f
  Q18_inst/Q_one_iter_inst_1/U119/Y (OA22XL)              0.47       3.13 f
  Q18_inst/Q_one_iter_inst_1/U74/Y (OAI221X1)             0.22       3.35 r
  Q18_inst/Q_one_iter_inst_1/U30/Y (CLKINVX1)             0.24       3.59 f
  Q18_inst/Q_one_iter_inst_1/U162/Y (OAI221X4)            0.56       4.15 r
  Q18_inst/Q_one_iter_inst_1/sub_105/B[6] (Q_one_iter_99_DW01_sub_3)
                                                          0.00       4.15 r
  Q18_inst/Q_one_iter_inst_1/sub_105/U150/Y (INVX3)       0.12       4.27 f
  Q18_inst/Q_one_iter_inst_1/sub_105/U220/Y (NAND2X1)     0.31       4.58 r
  Q18_inst/Q_one_iter_inst_1/sub_105/U179/Y (OAI21X4)     0.21       4.79 f
  Q18_inst/Q_one_iter_inst_1/sub_105/U158/Y (CLKINVX1)
                                                          0.21       5.01 r
  Q18_inst/Q_one_iter_inst_1/sub_105/U186/Y (OAI21X1)     0.20       5.21 f
  Q18_inst/Q_one_iter_inst_1/sub_105/U194/Y (AOI21X4)     0.16       5.37 r
  Q18_inst/Q_one_iter_inst_1/sub_105/U208/Y (XOR2X2)      0.17       5.54 r
  Q18_inst/Q_one_iter_inst_1/sub_105/DIFF[8] (Q_one_iter_99_DW01_sub_3)
                                                          0.00       5.54 r
  Q18_inst/Q_one_iter_inst_1/U75/Y (AO22X4)               0.21       5.75 r
  Q18_inst/Q_one_iter_inst_1/U174/Y (AO21X4)              0.28       6.03 r
  Q18_inst/Q_one_iter_inst_1/yo[8] (Q_one_iter_99)        0.00       6.03 r
  Q18_inst/Q_one_iter_inst_2/yi[8] (Q_one_iter_98)        0.00       6.03 r
  Q18_inst/Q_one_iter_inst_2/U41/Y (INVX3)                0.20       6.24 f
  Q18_inst/Q_one_iter_inst_2/U96/Y (OA22X4)               0.23       6.47 f
  Q18_inst/Q_one_iter_inst_2/U97/Y (OAI221X4)             0.31       6.78 r
  Q18_inst/Q_one_iter_inst_2/U121/Y (AO21X4)              0.24       7.02 r
  Q18_inst/Q_one_iter_inst_2/sub_100/B[7] (Q_one_iter_98_DW01_sub_3)
                                                          0.00       7.02 r
  Q18_inst/Q_one_iter_inst_2/sub_100/U130/Y (INVX8)       0.07       7.09 f
  Q18_inst/Q_one_iter_inst_2/sub_100/U184/Y (NAND2XL)     0.47       7.56 r
  Q18_inst/Q_one_iter_inst_2/sub_100/U121/Y (OAI21X2)     0.20       7.76 f
  Q18_inst/Q_one_iter_inst_2/sub_100/U159/Y (AOI21X2)     0.23       7.99 r
  Q18_inst/Q_one_iter_inst_2/sub_100/U179/Y (OAI21X4)     0.20       8.18 f
  Q18_inst/Q_one_iter_inst_2/sub_100/U140/Y (AOI21X2)     0.22       8.40 r
  Q18_inst/Q_one_iter_inst_2/sub_100/U173/Y (XOR2X4)      0.17       8.57 f
```

```
  Q18_inst/Q_one_iter_inst_2/sub_100/U140/Y (AOI21X2)     0.22       8.40 r
  Q18_inst/Q_one_iter_inst_2/sub_100/U173/Y (XOR2X4)      0.17       8.57 f
  Q18_inst/Q_one_iter_inst_2/sub_100/DIFF[11] (Q_one_iter_98_DW01_sub_3)
                                                          0.00       8.57 f
  Q18_inst/Q_one_iter_inst_2/U204/Y (AO22X4)              0.24       8.81 f
  Q18_inst/Q_one_iter_inst_2/U109/Y (AO21X4)              0.24       9.05 f
  Q18_inst/Q_one_iter_inst_2/xo[11] (Q_one_iter_98)       0.00       9.05 f
  Q18_inst/Q_one_iter_inst_3/xi[11] (Q_one_iter_97)       0.00       9.05 f
  Q18_inst/Q_one_iter_inst_3/U24/Y (CLKINVX12)            0.08       9.12 r
  Q18_inst/Q_one_iter_inst_3/U26/Y (INVX12)               0.06       9.18 f
  Q18_inst/Q_one_iter_inst_3/U66/Y (NAND2X4)              0.16       9.34 r
  Q18_inst/Q_one_iter_inst_3/U65/Y (INVX3)                0.10       9.44 f
  Q18_inst/Q_one_iter_inst_3/U136/Y (CLKAND2X8)           0.17       9.62 f
  Q18_inst/Q_one_iter_inst_3/U42/Y (AOI2BB1X2)            0.14       9.76 r
  Q18_inst/Q_one_iter_inst_3/U31/Y (OAI221X4)             0.35      10.11 f
  Q18_inst/Q_one_iter_inst_3/sub_105/B[1] (Q_one_iter_97_DW01_sub_3)
                                                          0.00      10.11 f
  Q18_inst/Q_one_iter_inst_3/sub_105/U130/Y (INVX2)       0.17      10.28 r
  Q18_inst/Q_one_iter_inst_3/sub_105/U141/Y (NOR2X4)      0.10      10.38 f
  Q18_inst/Q_one_iter_inst_3/sub_105/U145/Y (OA21X4)      0.27      10.65 f
  Q18_inst/Q_one_iter_inst_3/sub_105/U176/Y (OA21X4)      0.27      10.92 f
  Q18_inst/Q_one_iter_inst_3/sub_105/U166/Y (OAI21X4)     0.27      11.18 r
  Q18_inst/Q_one_iter_inst_3/sub_105/U143/Y (AOI21X1)     0.21      11.39 f
  Q18_inst/Q_one_iter_inst_3/sub_105/U185/Y (XOR2X2)      0.18      11.58 r
  Q18_inst/Q_one_iter_inst_3/sub_105/DIFF[9] (Q_one_iter_97_DW01_sub_3)
                                                          0.00      11.58 r
  Q18_inst/Q_one_iter_inst_3/U119/Y (AO22X4)              0.20      11.78 r
  Q18_inst/Q_one_iter_inst_3/U182/Y (AO21X4)              0.15      11.93 r
  Q18_inst/Q_one_iter_inst_3/yo[9] (Q_one_iter_97)        0.00      11.93 r
  Q18_inst/U10/Y (BUFX20)                                 0.13      12.06 r
  Q18_inst/Q_one_iter_inst_4/yi[9] (Q_one_iter_96)        0.00      12.06 r
  Q18_inst/Q_one_iter_inst_4/U97/Y (INVX12)               0.06      12.12 f
  Q18_inst/Q_one_iter_inst_4/U103/Y (OAI222X2)            0.35      12.48 r
  Q18_inst/Q_one_iter_inst_4/U48/Y (INVX6)                0.13      12.61 f
  Q18_inst/Q_one_iter_inst_4/U132/Y (AOI2BB1X4)           0.21      12.82 f
  Q18_inst/Q_one_iter_inst_4/U125/Y (OAI221X4)            0.33      13.15 r
  Q18_inst/Q_one_iter_inst_4/sub_100/B[1] (Q_one_iter_96_DW01_sub_3)
                                                          0.00      13.15 r
  Q18_inst/Q_one_iter_inst_4/sub_100/U149/Y (INVX3)       0.11      13.27 f
  Q18_inst/Q_one_iter_inst_4/sub_100/U152/Y (NAND2X2)     0.17      13.44 r
  Q18_inst/Q_one_iter_inst_4/sub_100/U180/Y (OAI21X4)     0.15      13.59 f
  Q18_inst/Q_one_iter_inst_4/sub_100/U186/Y (AOI21X4)     0.20      13.79 r
  Q18_inst/Q_one_iter_inst_4/sub_100/U122/Y (INVX4)       0.14      13.93 f
  Q18_inst/Q_one_iter_inst_4/sub_100/U121/Y (AO21X2)      0.31      14.24 f
  Q18_inst/Q_one_iter_inst_4/sub_100/U120/Y (XNOR2X4)     0.13      14.37 f
  Q18_inst/Q_one_iter_inst_4/sub_100/DIFF[5] (Q_one_iter_96_DW01_sub_3)
                                                          0.00      14.37 f
  Q18_inst/Q_one_iter_inst_4/U162/Y (AO22X4)              0.25      14.62 f
  Q18_inst/Q_one_iter_inst_4/U127/Y (AO21X4)              0.18      14.80 f
  Q18_inst/Q_one_iter_inst_4/xo[5] (Q_one_iter_96)        0.00      14.80 f
  Q18_inst/U9/Y (MX2X4)                                   0.19      15.00 f
  Q18_inst/xo[5] (Q_24)                                   0.00      15.00 f
  U6901/Y (AO22X4)                                        0.19      15.19 f
  xi_q18_reg[5]/D (DFFRX1)                                0.00      15.19 f
  data arrival time                                                 15.19

  clock clk (rise edge)                                  15.00      15.00
  clock network delay (ideal)                             0.50      15.50
  clock uncertainty                                      -0.10      15.40
  xi_q18_reg[5]/CK (DFFRX1)                               0.00      15.40 r
  library setup time                                     -0.21      15.19
  data required time                                                15.19
  ---------------------------------------------------------------------------
  data required time                                                15.19
  data arrival time                                                -15.19
  ---------------------------------------------------------------------------
  slack (MET)                                                       0.00


1
```

- Clock rate = 15ns
- Slack ≧ 0ns

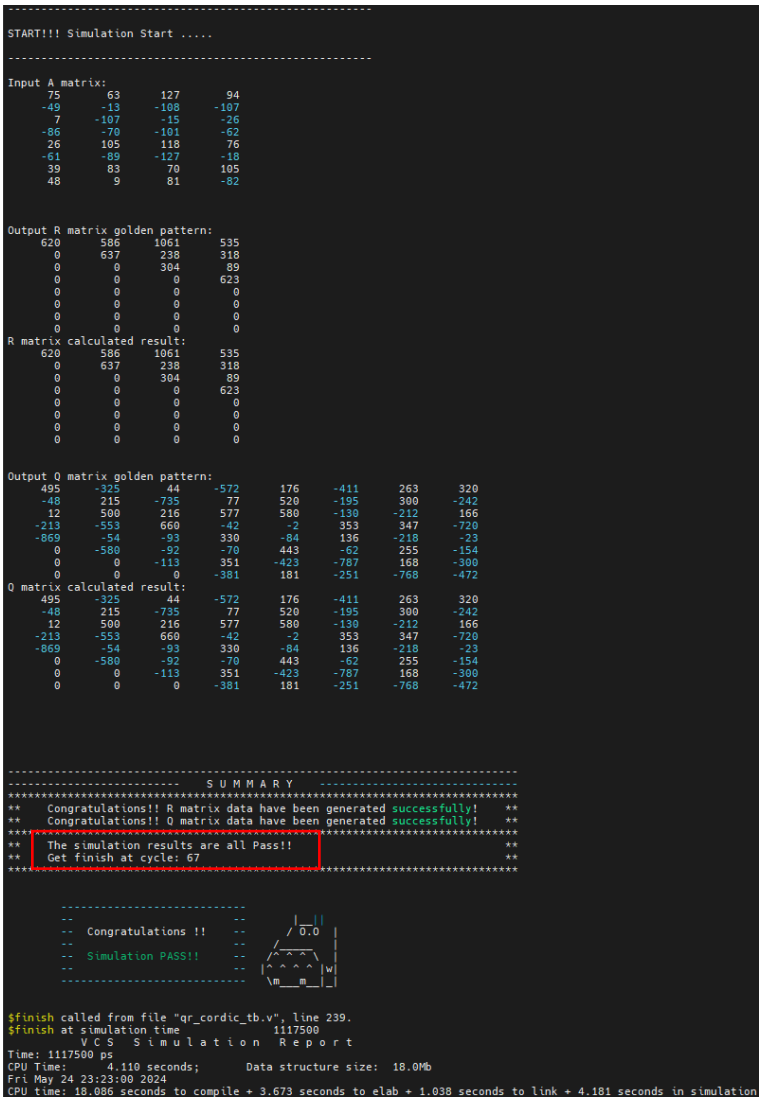
d. Area report

```
****************************************
Report : area
Design : qr_cordic
Version: U-2022.12
Date   : Fri May 24 23:07:08 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:                        145611
Number of cells:                       107787
Number of combinational cells:         103929
Number of sequential cells:              2092
Number of macros/black boxes:               0
Number of buf/inv:                      25314
Number of references:                     187

Combinational area:            1175906.092861
Buf/Inv area:                   169627.969362
Noncombinational area:           79005.481339
Macro/Black Box area:                0.000000
Net Interconnect area:        11484024.363495

Total cell area:               1254911.574200
Total area:                   12738935.937695
1
```

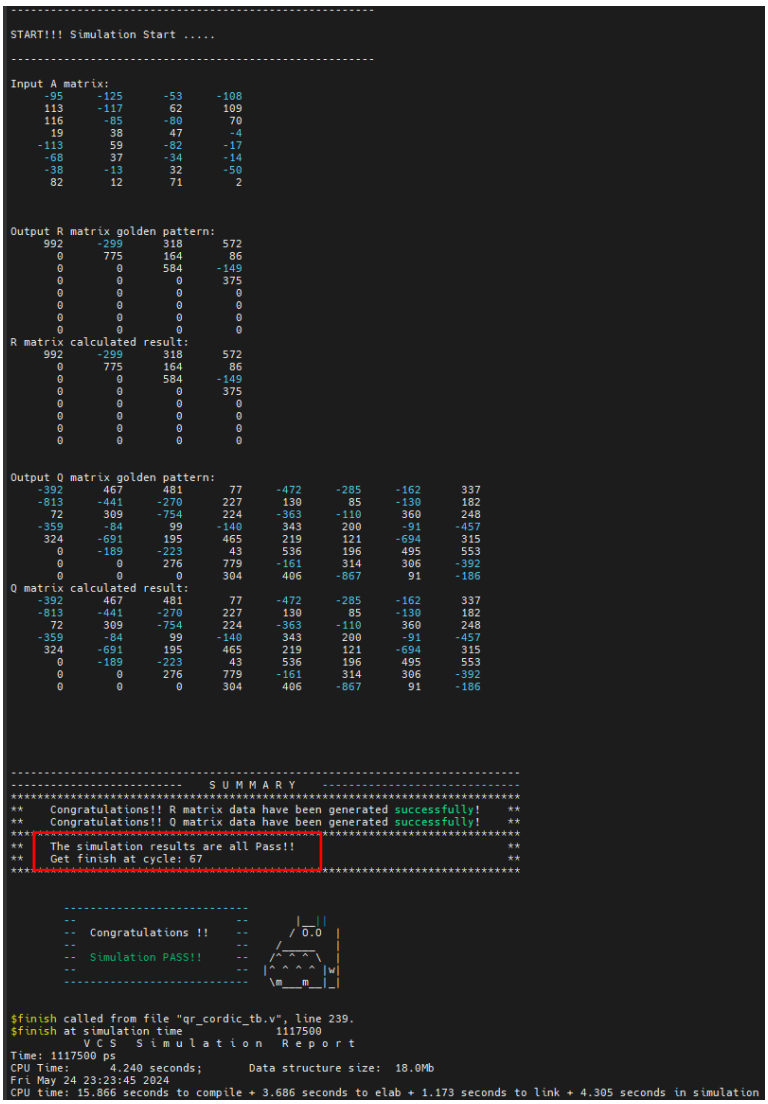
- Area: 1254911.5742 μm² (clock rate:15ns, standard compile)

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

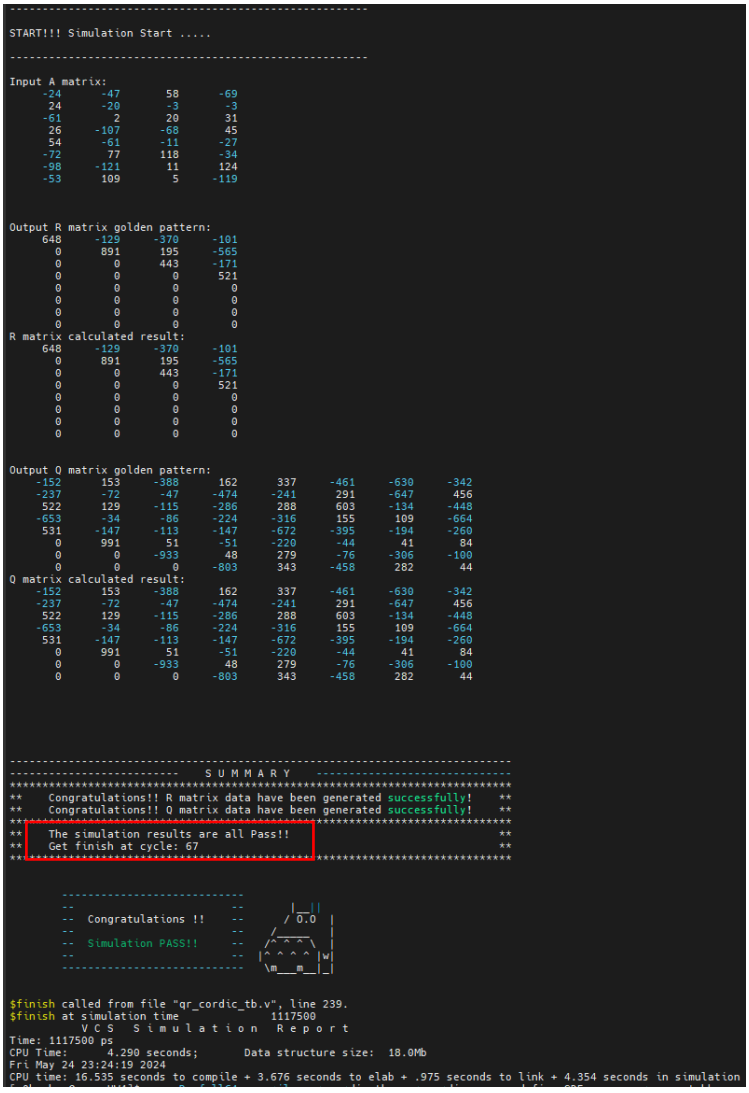
Pattern 1:



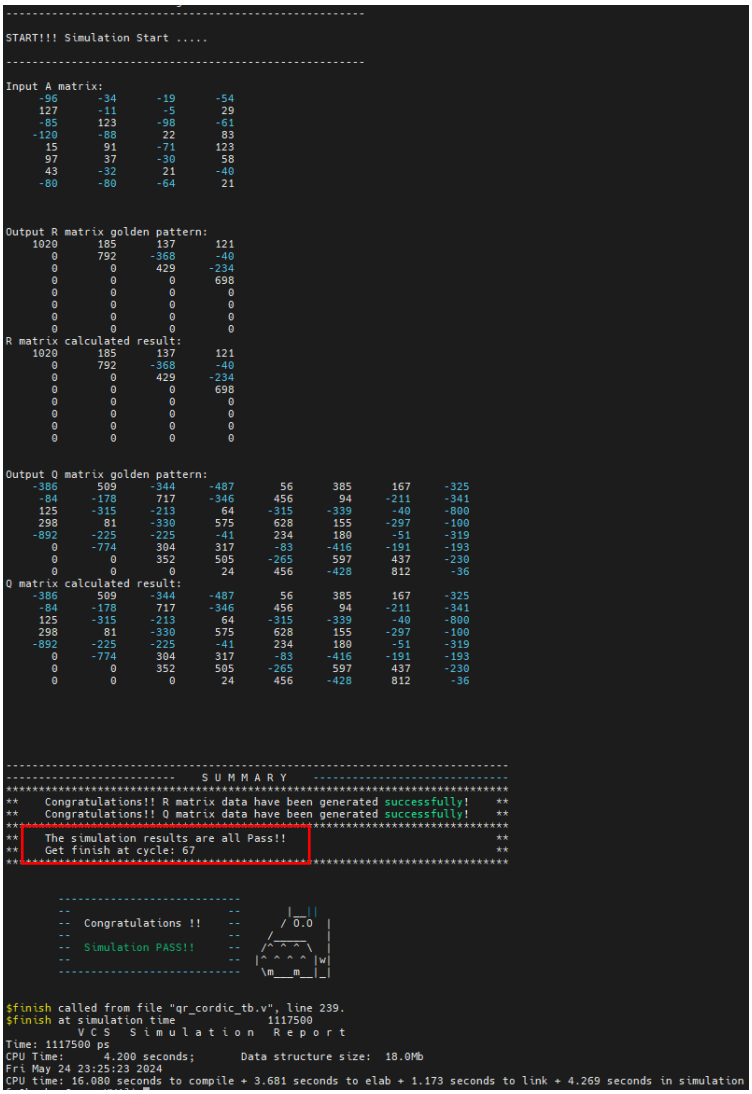
Pattern 2:



Pattern 3:



Pattern4:



- Both R and Q are all pass in gate-level simulation
- Clock rate : 15ns
- Cycles : 67
- Run time : 1117.5ns

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