Project 2: CORDIC

Author: Jason Yuan

Student ID: A69042479

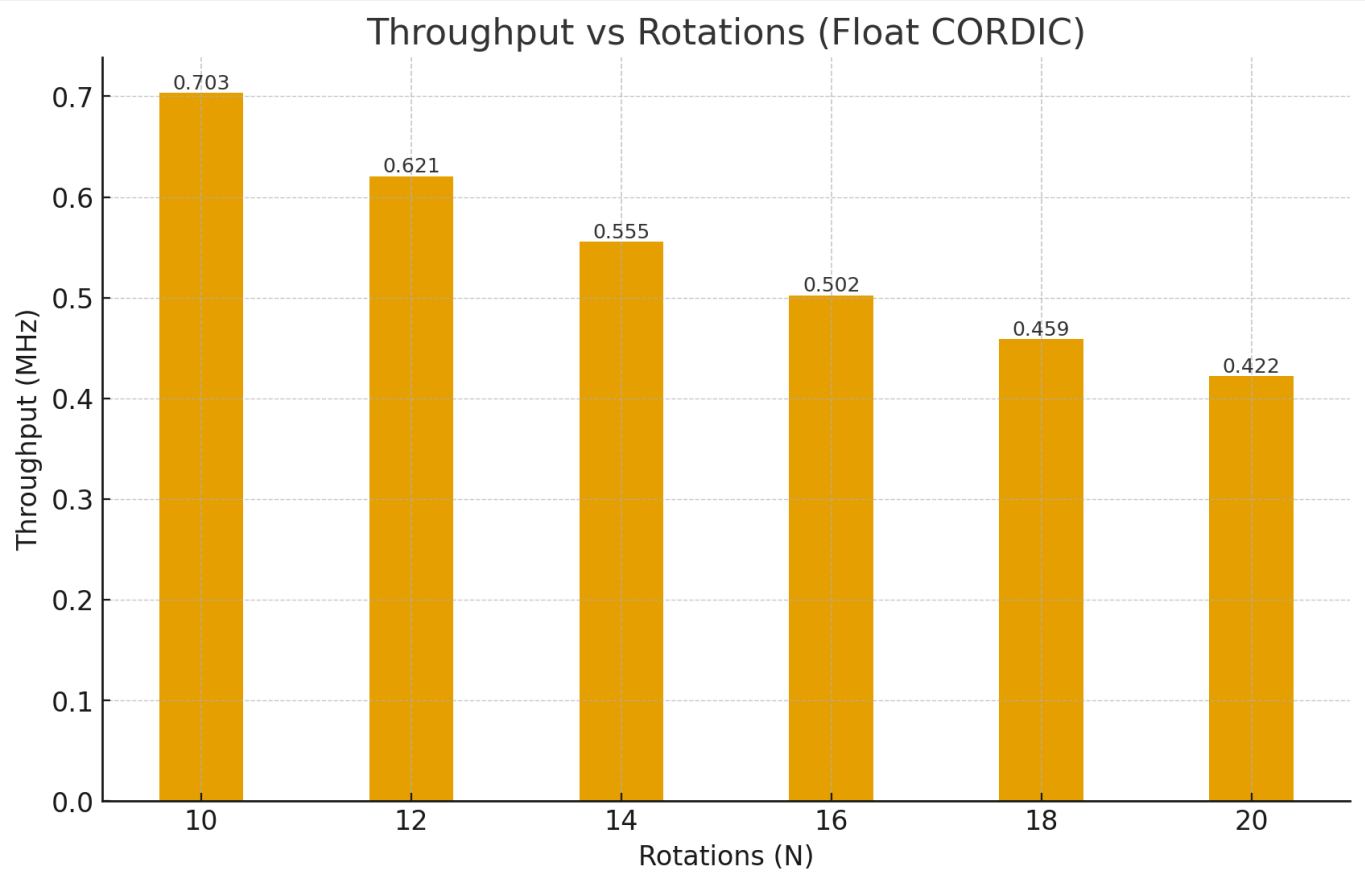
Github Repo: https://github.com/jas0xf/pp4fpgas-project2

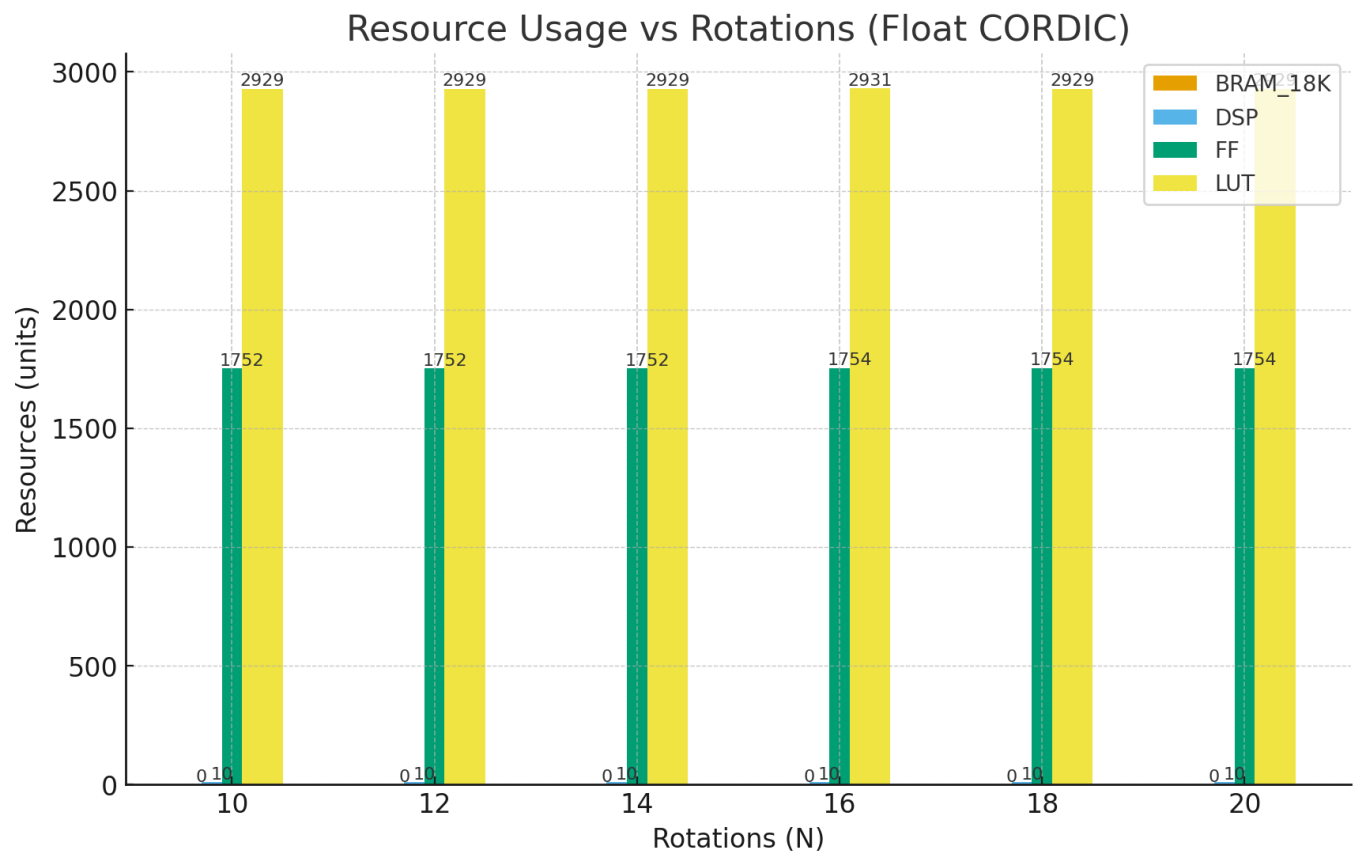
Q1

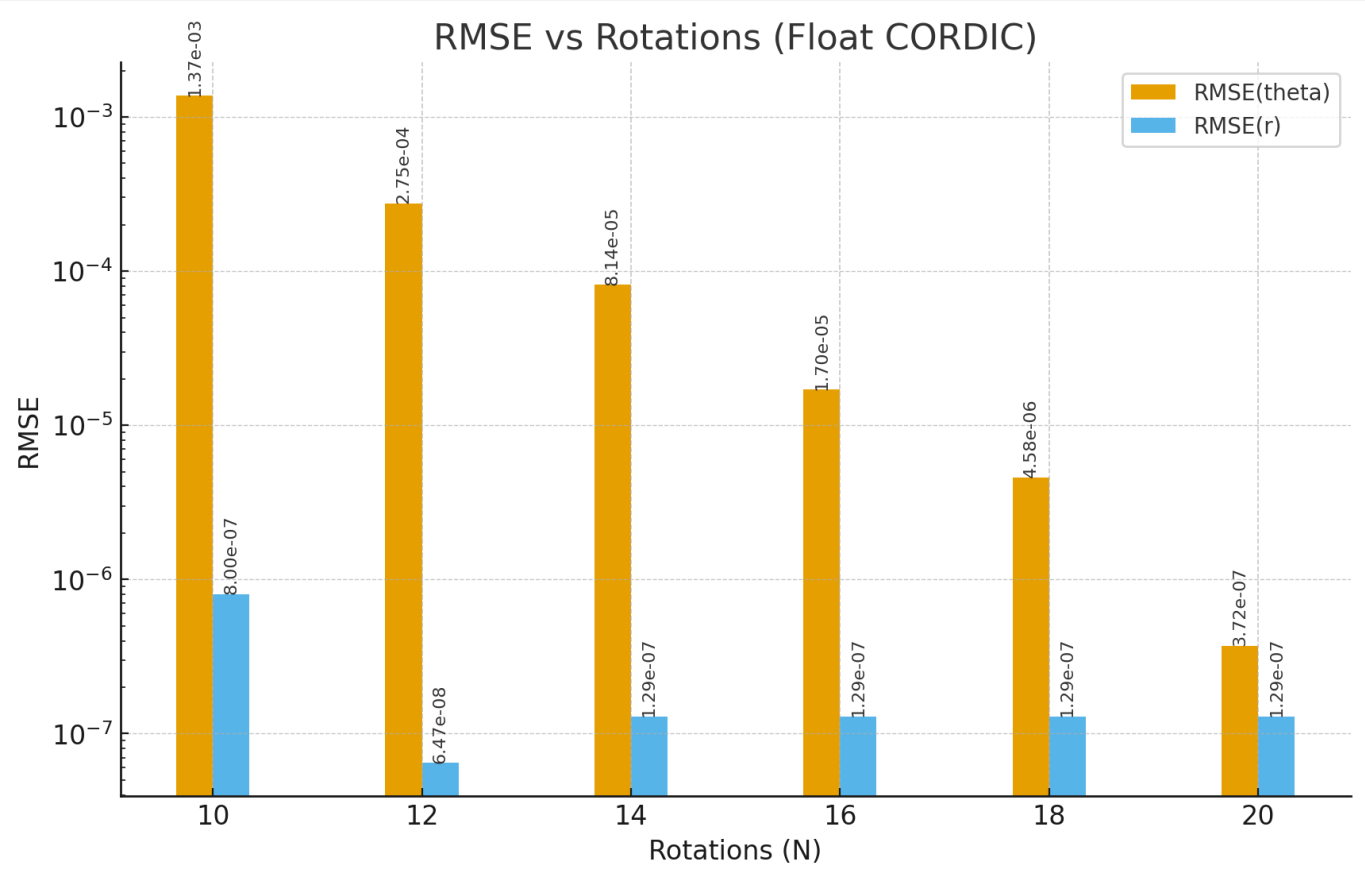
(a) — Floating-point CORDIC (vectoring mode), rotations = 10…20

ROTATIONS	CLOCK PERIOD (NS)	LATENCY (CYCLES)	INTERVAL (CYCLES)	THROUGHPUT (MHZ)	BRAM_18K	DSP	FF	LUT	RMSE(R)	RMSE(THETA)
10	7.29	194	195	0.703	0	10	1752	2929	0.000000800	0.001373779960
12	7.29	220	221	0.621	0	10	1752	2929	0.000000065	0.000274910999
14	7.29	246	247	0.555	0	10	1752	2929	0.000000129	0.000081368569
16	7.29	272	273	0.502	0	10	1754	2931	0.000000129	0.000017025930
18	7.29	298	299	0.459	0	10	1754	2929	0.000000129	0.000004583768
20	7.29	324	325	0.422	0	10	1754	2929	0.000000129	0.000000372390

(b) — Plot


Throughput vs Rotations (Float CORDIC)
Throughput (MHz)
Rotations (N)
0.703
0.621
0.555
0.502
0.459
0.422
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
10
12
14
16
18
20


Resource Usage vs Rotations (Float CORDIC)
Resources (units)
Rotations (N)
BRAM_18K
DSP
FF
LUT
2929
2929
2929
2931
2929
2929
1752
1752
1752
1754
1754
1754
0
0
0
0
0
0
0
500
1000
1500
2000
2500
3000
10
12
14
16
18
20



(c) At what number of rotations does the accuracy stop noticeably improving in the plot?

From the plots, **RMSE(r) stops noticeably improving at ~14 rotations** (it plateaus around ≈1.29×10^{-7} thereafter). **RMSE(θ) does not show a clear plateau up to 20 rotations**

Q2

(a)

VARIABLES	INTERGER BITS
x	3
y	3
r	2
theta	3

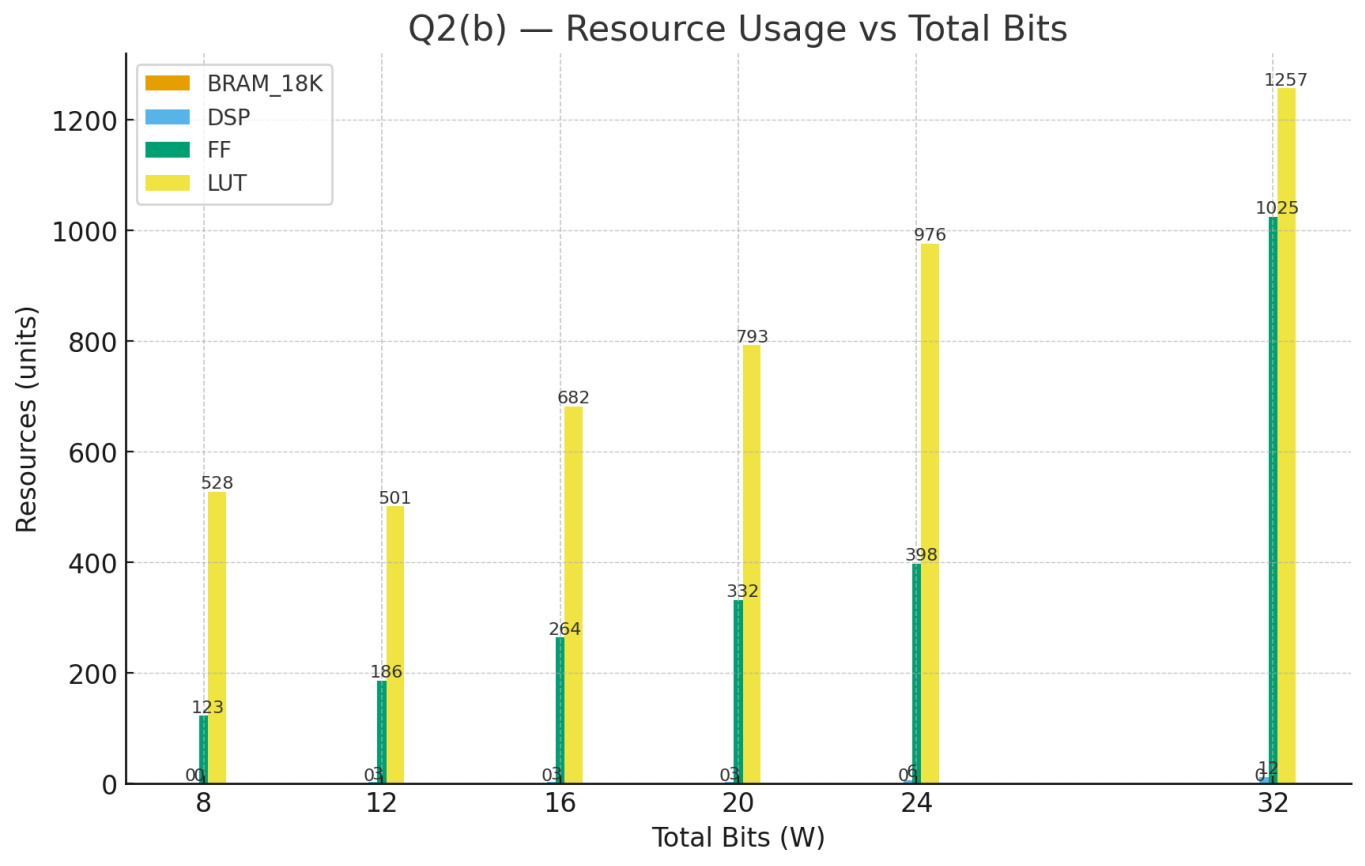
> *K converges to 1.647, r <= sqroot(2), so maximum x,y <= 1.647 * 1.414*
>
> *r <= sqroot(2), r <= 1.414*
>
> *theta ∈ [−π , π)*
>
> *cum_theta:*
>
> *∑arctan(2^−i)≈1.7433*
>
> *cum_theta ∈ [−π - 1.7433, π + 1.7433)≈±4.885*
>
> *but we can constraint cum_theta's range in the for loop to keep it within [−π , π)*

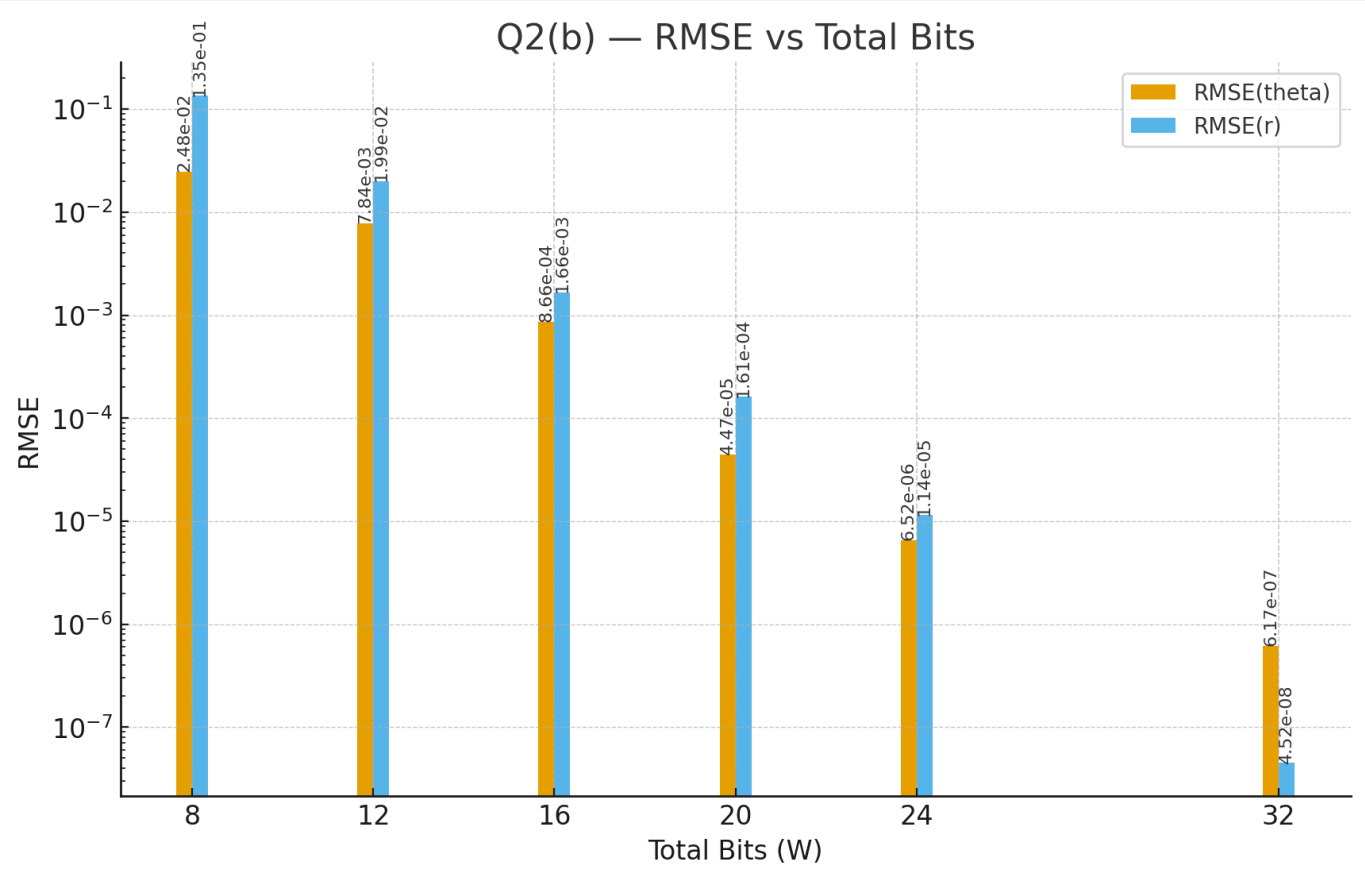
(b) — ap_fixed sweep (total bits = 8, 12, 16, 20, 24, 32)

Table

TOTAL BITS	CLOCK PERIOD (NS)	LATENCY (CYCLES)	INTERVAL (CYCLES)	THROUGHPUT (MHZ)	BRAM_18K	DSP	FF	LUT	RMSE(R)	RMSE(THETA)
8	7.073	45	46	3.074	0	0	123	528	1.349954e-01	2.476855e-02
12	6.163	45	46	3.527	0	3	186	501	1.990143e-02	7.838610e-03
16	6.792	46	47	3.133	0	3	264	682	1.664304e-03	8.657080e-04
20	6.814	46	47	3.122	0	3	332	793	1.607916e-04	4.470670e-05
24	6.918	46	47	3.076	0	6	398	976	1.144594e-05	6.520730e-06
32	7.160	66	67	2.085	0	12	1025	1257	4.515860e-08	6.172185e-07

Plots



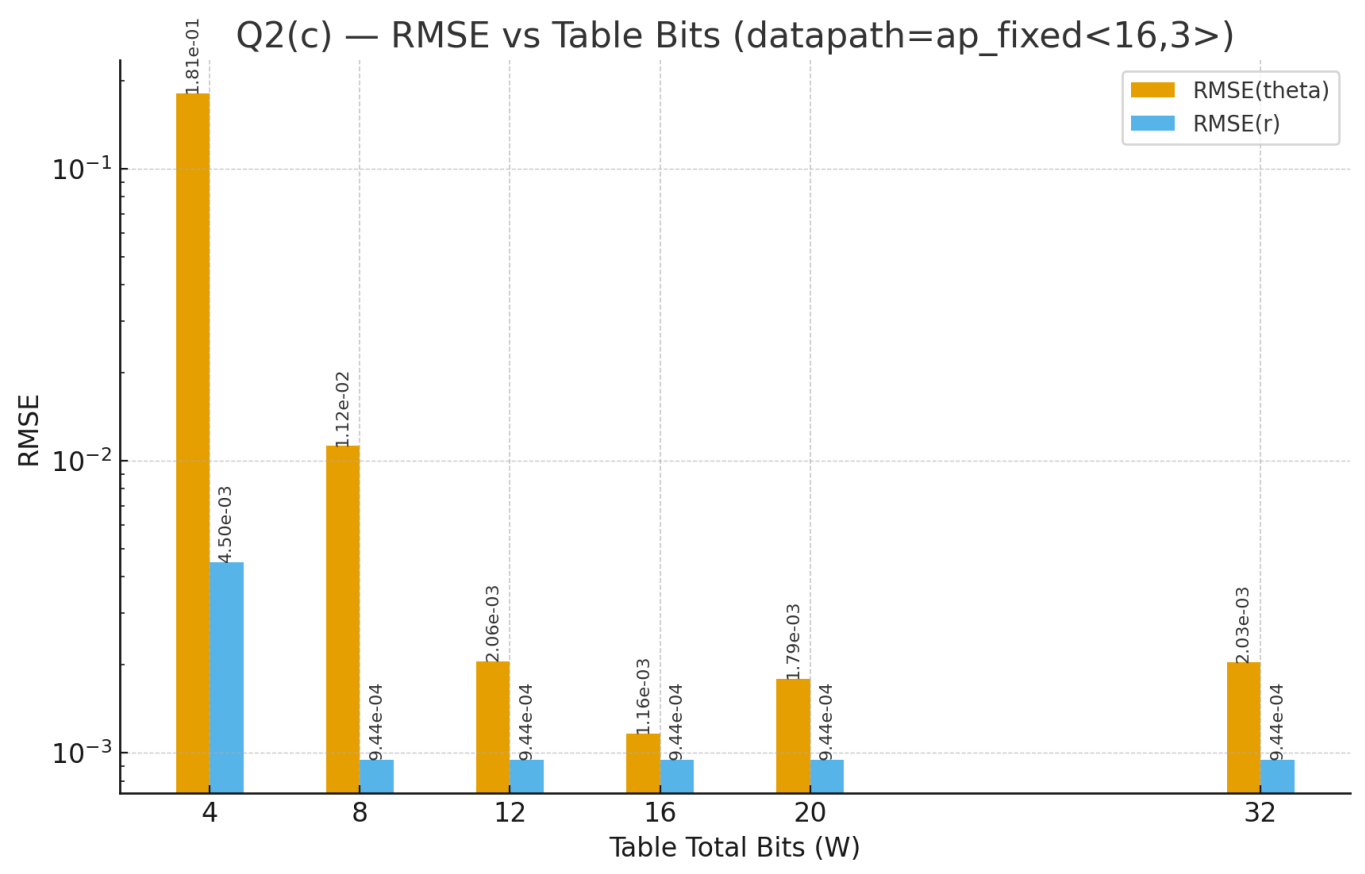


(c) — Tables quantization sweep (W = 4, 8, 12, 16, 20, 32)

Table.

TABLE BITS (W)	CLOCK PERIOD (NS)	LATENCY (CYCLES)	INTERVAL (CYCLES)	THROUGHPUT (MHZ)	BRAM_18K	DSP	FF	LUT	RMSE(R)	RMSE(THETA)
4	7.168	45	46	3.033	0	1	229	639	4.499e-03	1.806e-01
8	7.168	45	46	3.033	0	3	253	658	9.437e-04	1.123e-02
12	7.168	45	46	3.033	0	3	277	677	9.437e-04	2.056e-03
16	7.168	45	46	3.033	0	3	301	697	9.437e-04	1.163e-03
20	7.168	45	46	3.033	0	3	325	725	9.437e-04	1.791e-03
32	7.168	65	66	2.114	0	5	680	902	9.437e-04	2.033e-03

Plot

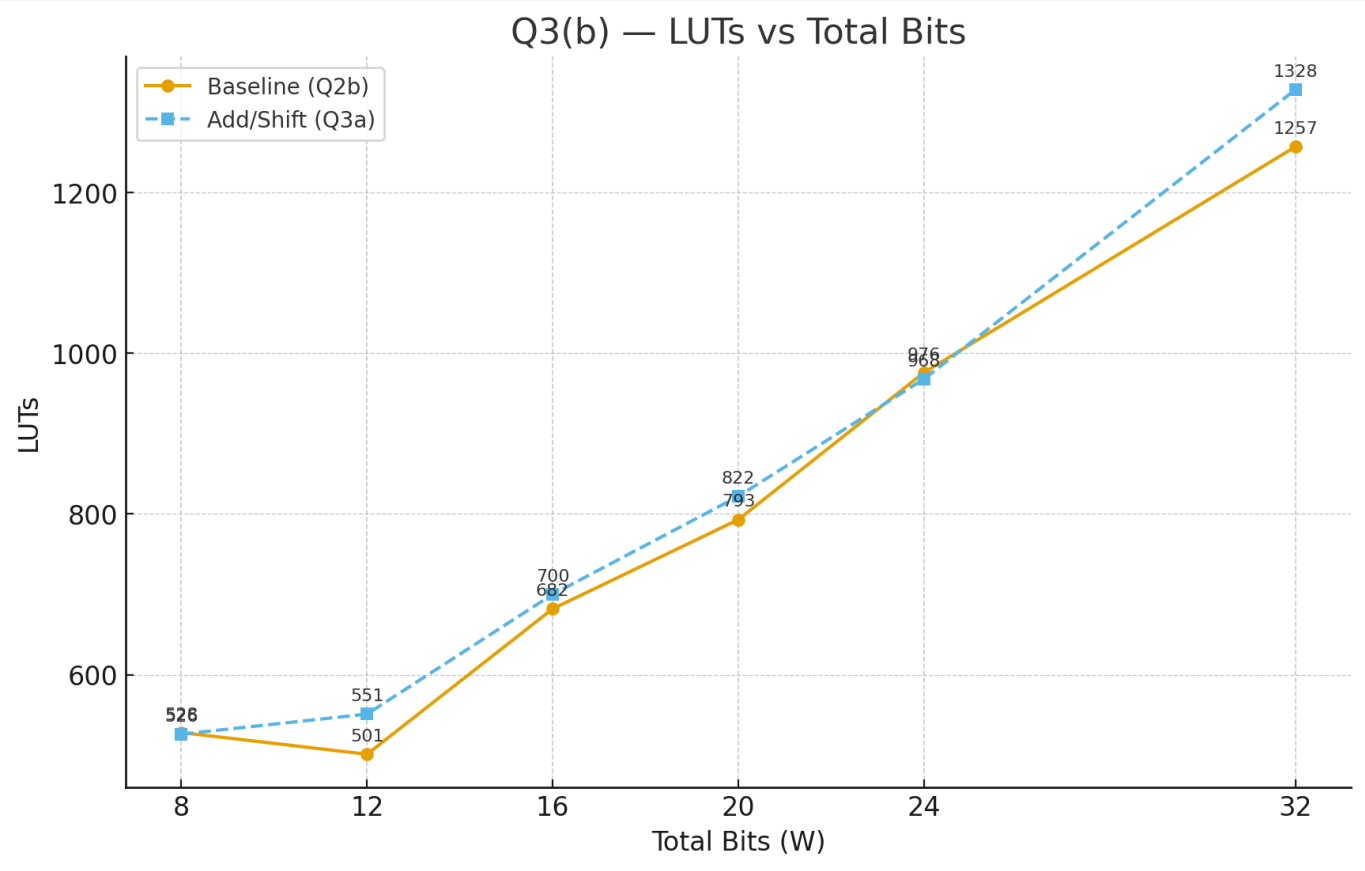


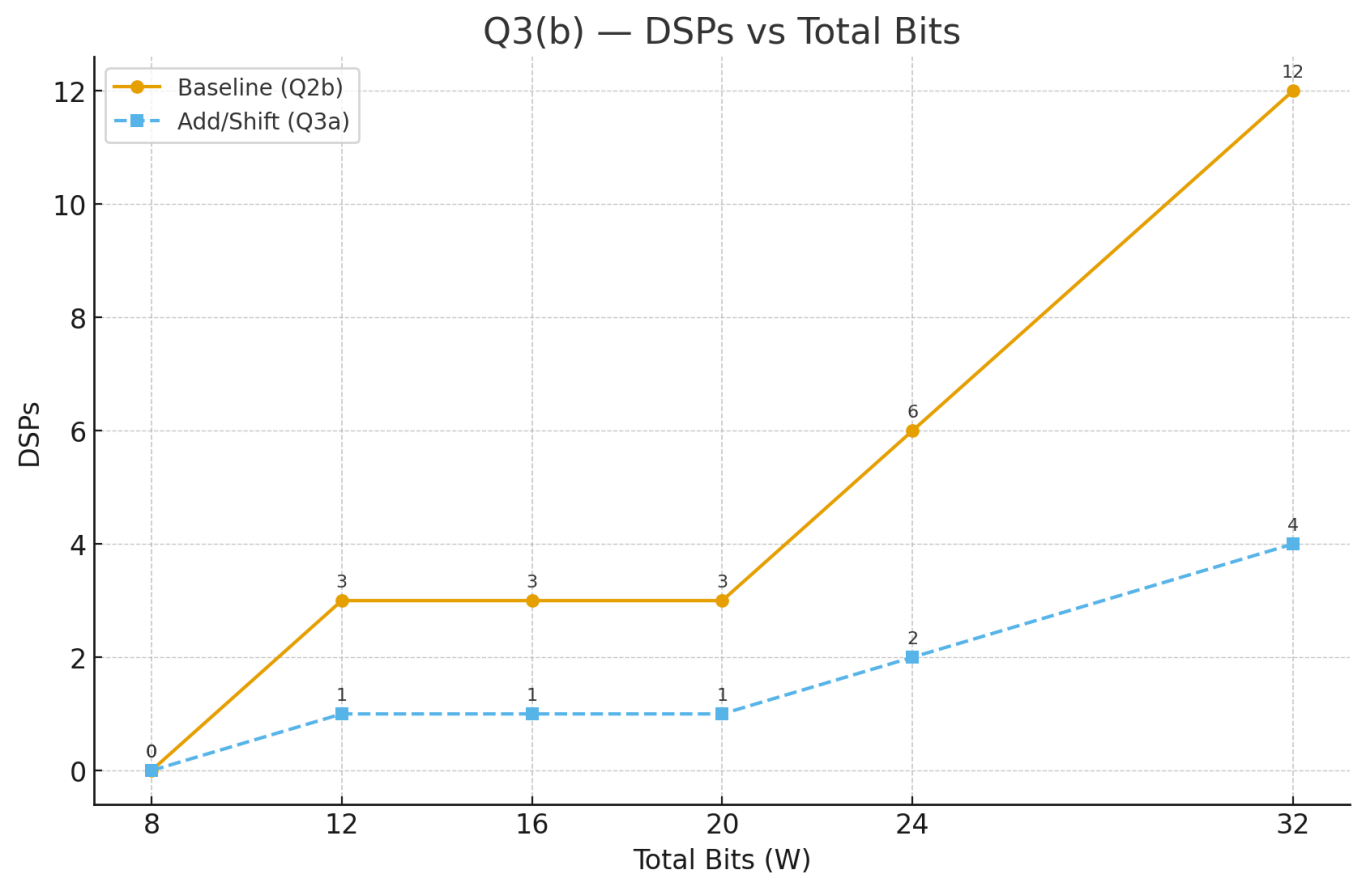
Q3

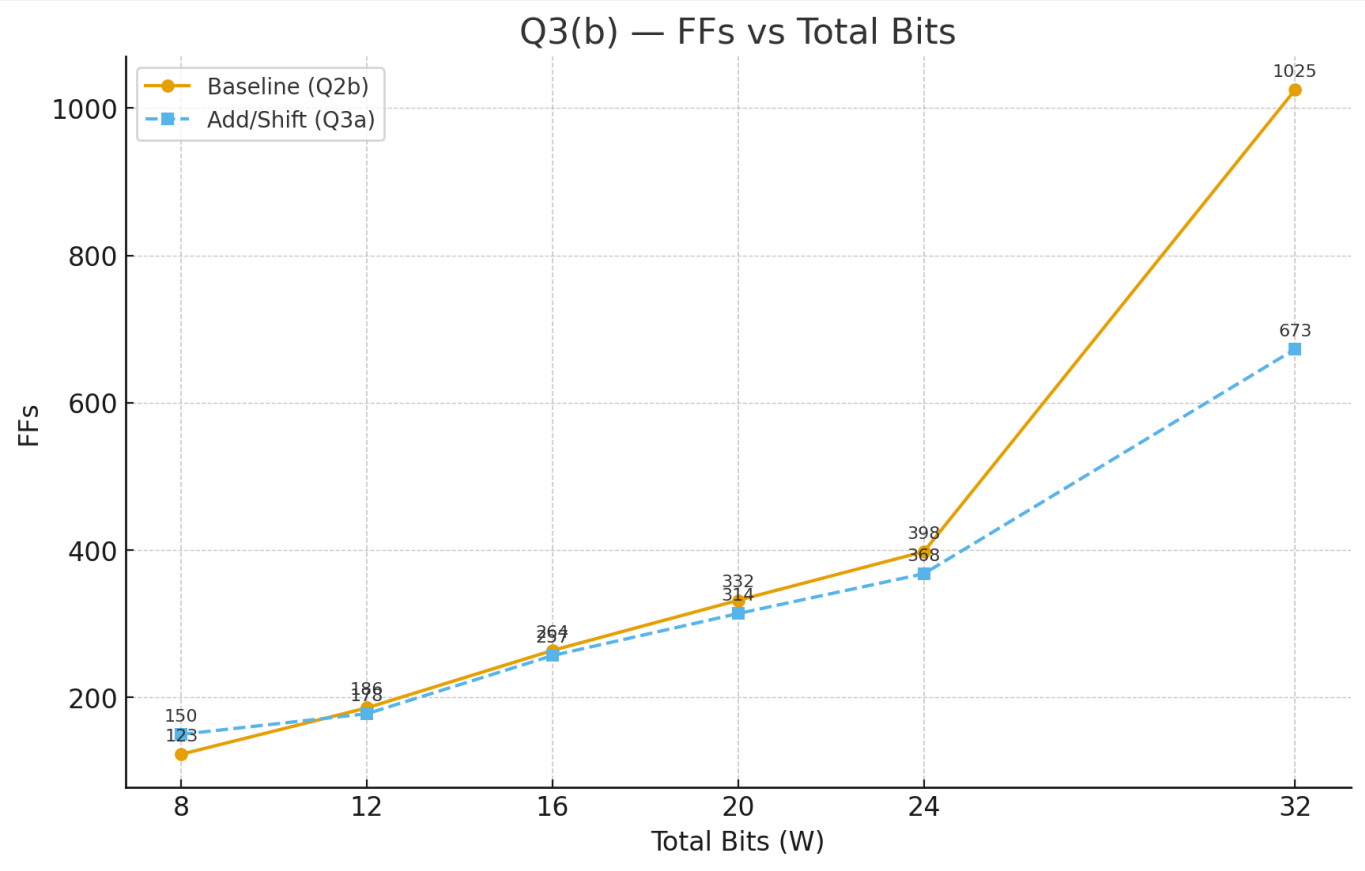
(a) — Add/Shift CORDIC

TOTAL BITS	CLOCK PERIOD (NS)	LATENCY (CYCLES)	INTERVAL (CYCLES)	THROUGHPUT (MHZ)	BRAM_18K	DSP	FF	LUT	RMSE(R)	RMSE(THETA)
8	6.518	46	47	3.264	0	0	150	526	2.269e-01	3.772e-01
12	7.208	45	46	3.016	0	1	178	551	1.297e-02	3.099e-02
16	7.168	45	46	3.033	0	1	257	700	5.884e-04	1.749e-03
20	5.860	46	47	3.631	0	1	314	822	8.365e-05	1.520e-04
24	6.540	46	47	3.253	0	2	368	968	6.530e-06	1.544e-04
32	6.912	47	48	3.014	0	4	673	1328	4.516e-08	1.540e-04

(b) **Plots**


Q3(b) — LUTs vs Total Bits
Baseline (Q2b)
Add/Shift (Q3a)
LUTs
Total Bits (W)
600
800
1000
1200
8
12
16
20
24
32
528
551
501
700
682
822
793
976
968
1328
1257


Q3(b) — DSPs vs Total Bits
Baseline (Q2b)
Add/Shift (Q3a)
DSPs
Total Bits (W)
0
2
4
6
8
10
12
8
12
16
20
24
32
0
3
3
3
6
12
0
1
1
1
2
4



Q4

(a) LUT Table Size & Input/Output datatype:

Entries (address space): LUT index is `concat(x_bits, y_bits)` → **address width** = `Wx + Wy` →

#entries =

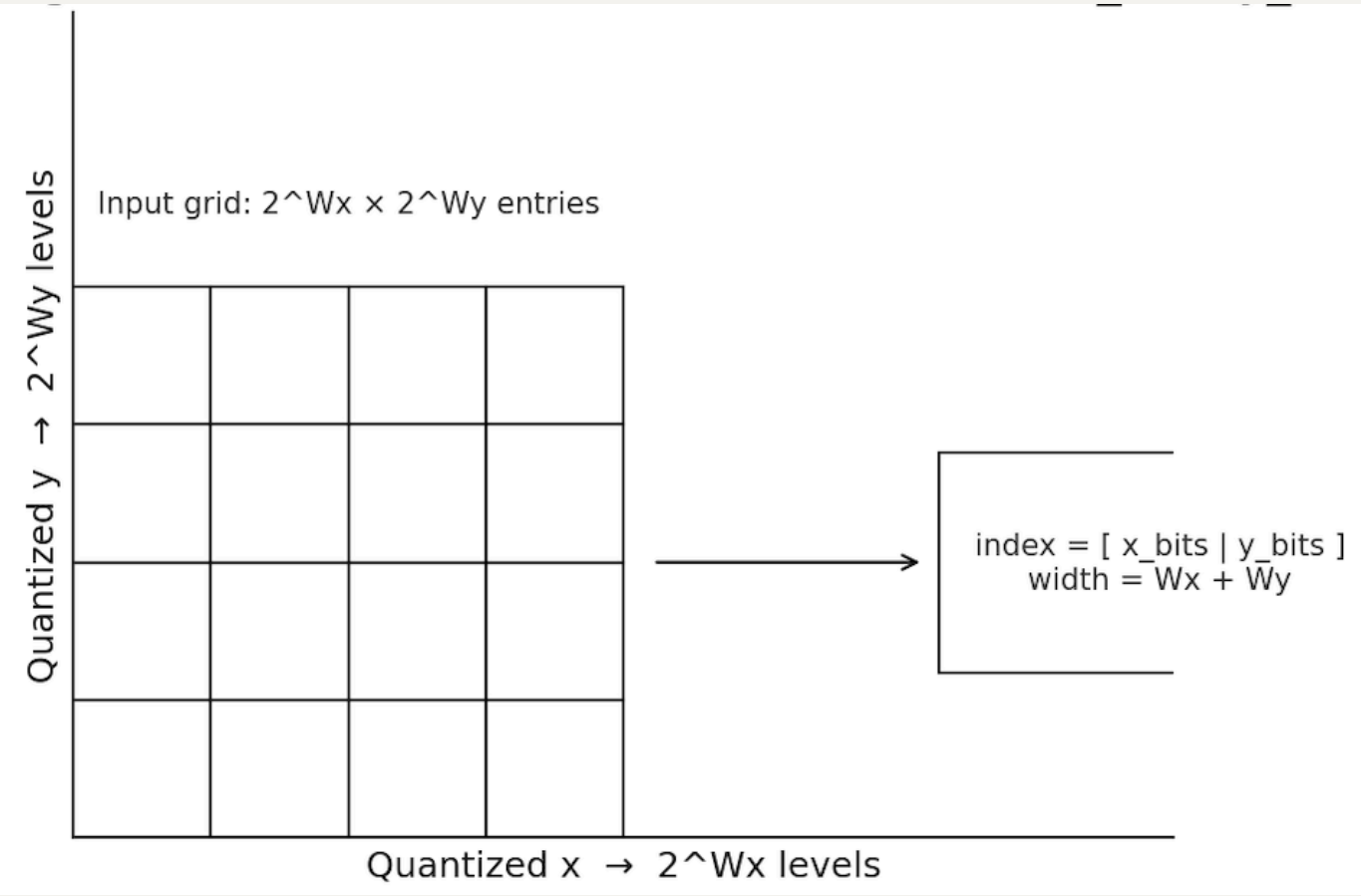
$$2^{(W_x+W_y)} \tag{1}$$

Data width per entry:

store **both outputs**: **data bits/entry** = `Wr + Wθ`

Total LUT bits:

$$\text{Mem\_bits} = 2^{(W_x+W_y)} \cdot (W_r + W_\theta) \tag{2}$$



(the code in CORDIC_LUT just serializes 2D array into 1D array)

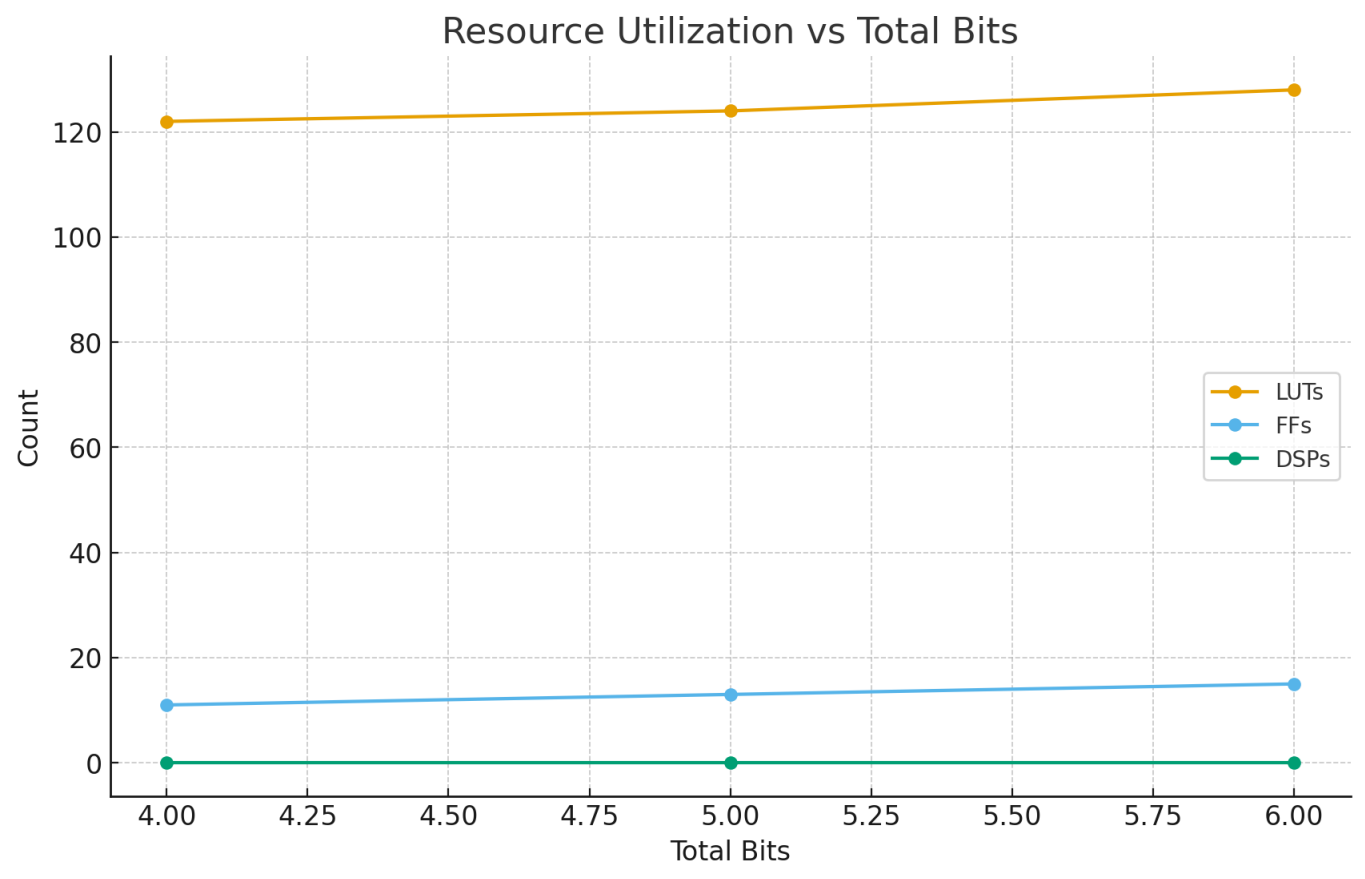
(b) Resource usage, throughput, latency, and error vs total bits

For 7- and 8-bit, only runtime RMSE was available (synthesis not completed), so resource/latency/throughput are N/A.

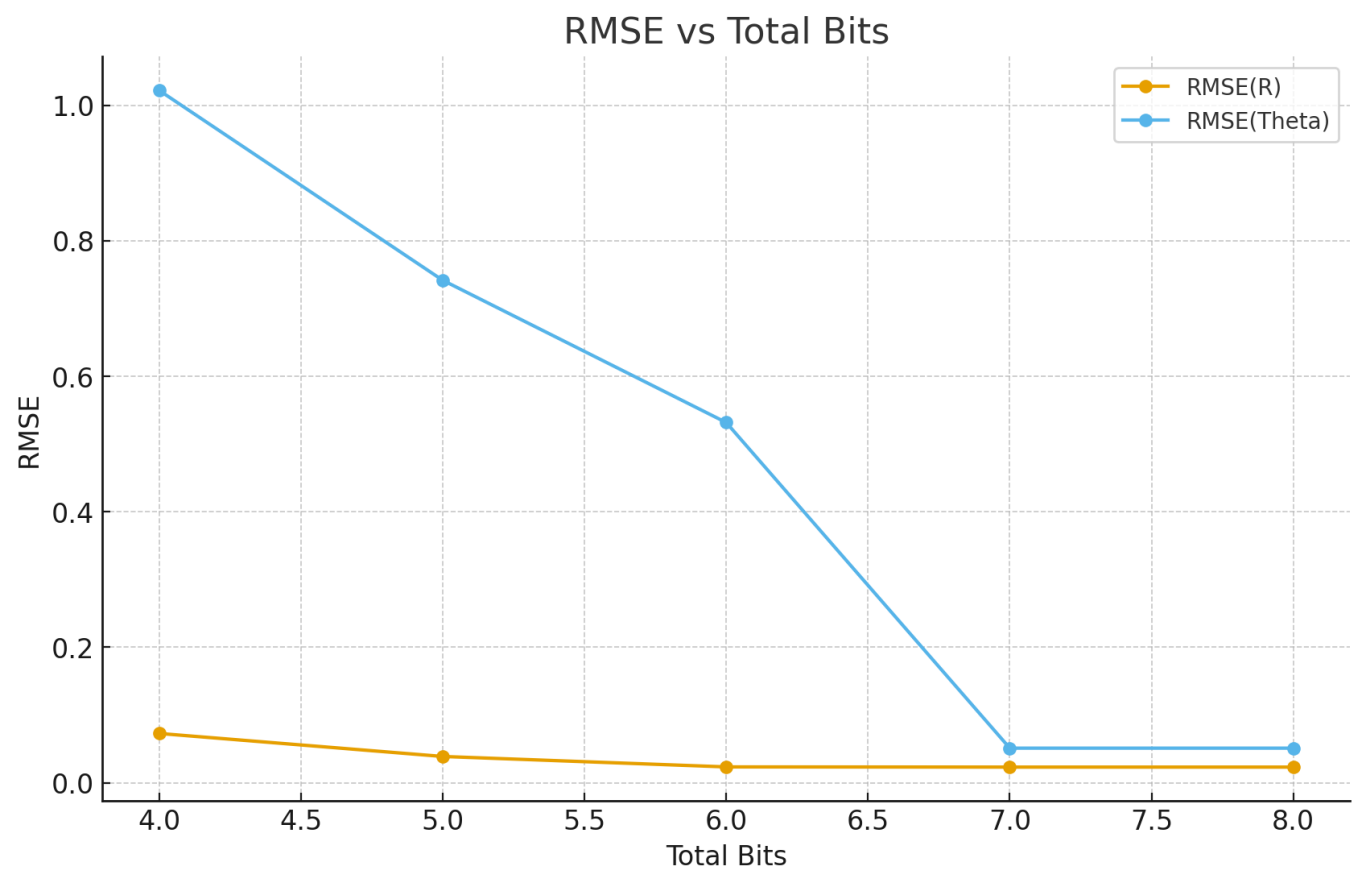
BITS	LUT	FF	DSP	BRAM18K	II	LATENCY (CYCLES)	THROUGHPUT (MSPS)	RMSE(R)	RMSE(Θ)
4	122	11	0	2	3	2	58.40	0.07276	1.02194
5	124	13	0	2	3	2	56.08	0.03879	0.74169
6	128	15	0	4	3	2	55.91	0.02338	0.53166
7	—	—	—	—	—	—	—	0.02309	0.05105
8	—	—	—	—	—	—	—	0.02309	0.05105

(c) Resource utilization vs total bits

[Only 4–6 bit points appear (no 7/8 synth).]



(d) RMSE vs total bits



(e) Advantages and disadvantages of CORDIC vs LUT

Advantages (CORDIC): precision improves with additional iterations rather than exponentially growing table size

Disadvantages (CORDIC): iterative nature creates a throughput–area trade-off (higher throughput needs deeper pipelining/unrolling), fixed-point micro-rotations accumulate quantization error that improves only as bit-width/iterations increase, and careful scaling/range handling is required; by contrast, a LUT can deliver one-cycle results if sufficient BRAM/ports are available for the target resolution.