

IJEAST

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY

VOLUME : 11 ISSUE : 07 Print / Issue Publication Date: November 2026

ISSN : 2455-2143

Indexed In

Google SCHOLAR OPEN ACCESS doi Crossref

WWW.IJEAST.COM

editor@ijeast.com

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 71-77
Published Online November 2026 in IJEAST (http://www.ijeast.com)

DESIGN AND PERFORMANCE EVALUATION OF A RISC-V PROCESSOR INTEGRATED WITH CUSTOM MULTIPLIER, MULTIPLY-ACCUMULATE (MAC), AND CORDIC ACCELERATORS

Uday Sappogu, Guduru Rupesh Reddy
Undergraduate
Dept. of ECE, RGUKT

Mr.K.Abdul Munaf
Assistant Professor,
M.Tech Dept. of ECE, RGUKT

Abstract—**This paper presents a 32-bit RISC-V processor, described in Verilog HDL and simulated in Xilinx Vivado, extended with three arithmetic accelerators: a multiplier, a multiply-accumulate (MAC) unit computing Y = A×B + C, and an iterative CORDIC unit computing sine and cosine in Q16.16 fixed point with 16 iterations. The accelerators are invoked through custom instructions in the RISC-V custom-0 opcode space (0001011), with funct3 values 010, 011 and 100 selecting the multiplier, MAC and CORDIC. Behavioral simulation confirms 10×20 = 200, 10×20 + 10 = 210, and sine/cosine at 0°, 30° and 45°. A comparison framework (cycle count, execution time, speedup, throughput, FPGA utilization, timing and power) is defined for the accelerated core against a baseline core. Expected results indicate a cycle-count reduction of roughly two orders of magnitude for multiplication, about one order of magnitude for sine/cosine, and a modest increase in FPGA resources and power.**

Index Terms—**RISC-V, custom instruction, hardware accelerator, CORDIC, multiply-accumulate, fixed-point arithmetic, FPGA, Verilog.**

I. INTRODUCTION

The open RISC-V instruction set architecture (ISA) is designed for extension: the base ISA reserves opcode regions for user-defined instructions, so designers can add application-specific functionality without breaking toolchain compatibility [1]. The base RV32I ISA has no multiply instruction and no support for transcendental functions. A core built only from RV32I must therefore synthesize multiplication from shifts and additions, and trigonometric functions from software loops or tables, in many instruction cycles.

Signal processing, control and communication workloads rely heavily on multiplication, multiply-accumulate and rotation. Moving these operations into dedicated datapath blocks, controlled by single instructions, is a classical route to shorter execution time. This work examines that route on a small, fully inspectable 32-bit RISC-V core.

Objective. To quantify, under a reproducible methodology, the performance difference between a conventional RISC-V core and the same core extended with custom multiplier, MAC and CORDIC accelerators.

Contributions. (i) An open description of an accelerator extended RISC-V core using the custom-0 opcode with three funct3-selected operations; (ii) a Q16.16 formulation of the three units, including CORDIC gain compensation and error bounds; (iii) a measurement protocol that separates behavioral-simulation evidence from FPGA-implementation evidence; and (iv) expected performance, area and accuracy figures for the accelerated core against a baseline. No new accelerator architecture and no performance superiority over published cores is claimed.

II. RELATED WORK

RISC-V extensibility. The ISA specification defines base formats and reserves custom opcode spaces for non-standard extensions [1]. The custom-0 major opcode (0001011) used here is one such space.

Cores with DSP extensions. Gautschi et al. present an open source RISC-V core for near-threshold IoT operation with DSP-oriented instruction extensions and micro-architectural

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 71-77
Published Online November 2026 in IJEAST (http://www.ijeast.com)

optimizations, reporting average gains in speed and energy efficiency on sensor-processing workloads [6]. That work is at a much larger scale (compiler support, multi-core clusters, ASIC results) and is cited as context, not as a benchmark.

CORDIC. Volder introduced CORDIC for trigonometric computation with shift-and-add operations [2]; Walther unified circular, linear and hyperbolic modes [3]. Andraka surveys CORDIC variants for FPGAs and their area/speed trade-offs [4]. Meher et al. review fifty years of algorithmic and architectural developments, including higher-radix, redundant and pipelined forms [5]. The unit here is the basic iterative radix-2 form.

Positioning. [To be completed: a documented literature search (IEEE Xplore, ACM DL, Scopus; terms such as RISC V custom instruction CORDIC, RISC-V MAC accelerator FPGA) and a table comparing the closest 5–10 works by ISA, accelerators, precision, platform and reported metrics. Novelty should not be claimed before this is done.]

III. PROPOSED ARCHITECTURE

The core comprises a program counter (PC), instruction memory, instruction decoder, register file, ALU, accelerator control logic and result write-back circuitry (Fig. 1). The PC addresses instruction memory; the decoder splits the instruction into opcode, rd, funct3, rs1, rs2 and funct7; the register file supplies operands; the control logic enables one functional unit; and the write-back stage selects the ALU or accelerator result for storage in rd.

[CONFIRM against RTL and state here: datapath type (single-cycle, multi-cycle or pipelined); implemented ISA subset; presence of data memory and load/store; register file size; branch and PC-update handling.]

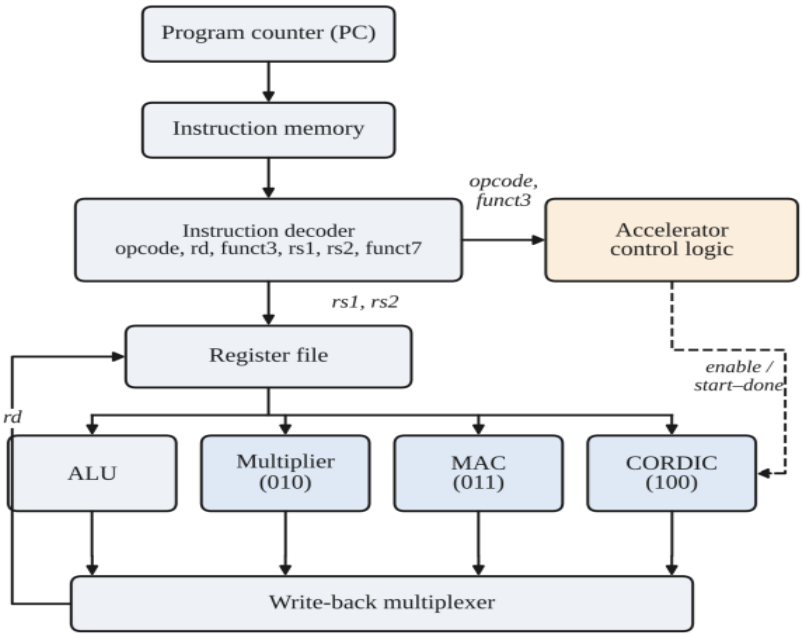


Fig. 1. Block diagram of the proposed RISC-V core with multiplier, MAC and CORDIC accelerators.

IV. CUSTOM INSTRUCTION ENCODING

Custom instructions use the RISC-V R-type layout (Table I) with opcode 0001011. The funct3 field selects the accelerator (Table II).

TABLE I. R-TYPE FIELD LAYOUT

31–25	**24–20**	**19–15**	**14–12**	**11–7**	**6–0**
funct7 [CONFIRM]	rs2 (B)	rs1 (A)	funct3	rd	0001011

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 71-77
Published Online November 2026 in IJEAST (http://www.ijeast.com)

TABLE II. CUSTOM FUNCT3 ASSIGNMENT

Funct 3	Unit	Operation
010	Multiplier	rd ← rs1 × rs2
011	MAC	rd ← rs1 × rs2 + C
100	CORDIC	sin/cos of angle in rs1

The decoder asserts a custom-instruction flag when opcode = 0001011, then a one-hot enable for the selected unit from funct3. [CONFIRM and document: source of the MAC addend C; how CORDIC returns two results through one 32- bit write-back port; the funct7 value; behavior for unassigned funct3; multiplier result width.]

V. ACCELERATOR HARDWARE DESIGN

A. Multiplier

The multiplier forms the product of two 32-bit register operands. [CONFIRM: inferred `*` operator (typically mapped to DSP slices) or hand-designed structure; signedness; result width.] A full signed 32×32 product needs 64 bits; if only the low 32 bits are written back, overflow occurs when $|A \cdot B| \geq 2^{31}$.

B. MAC Accelerator

The MAC computes Y = A×B + C. In integer mode Y = (A·B + C) mod 2^{32}. In Q16.16 mode the product has 32 fractional bits and is rescaled: Y = ((A·B) ≫ 16) + C, using a 64-bit intermediate. [CONFIRM mode; the example 10×20 + 10 = 210 is consistent with integer operands.]

C. CORDIC Accelerator

Circular-mode CORDIC in rotation mode [2], [4] performs, for i = 0, …, n−1 with n = 16:

$x(i+1) = x(i) - d(i)\cdot y(i)\cdot 2^{-i}$ (1)

$y(i+1) = y(i) + d(i)\cdot x(i)\cdot 2^{-i}$ (2)

$z(i+1) = z(i) - d(i)\cdot \arctan(2^{-i})$ (3)

where d(i) = +1 if z(i) ≥ 0, else −1. Each micro-rotation scales the magnitude by $\sqrt{(1 + 2^{-2i})}$; the cumulative gain $K = \prod\sqrt{(1 + 2^{-2i})}$ converges to ≈ 1.646760 (1/K ≈ 0.607253). Initializing x(0) = 1/K, y(0) = 0, z(0) = θ gives x(n) ≈ cos θ and y(n) ≈ sin θ without a final scaling multiplier (Fig. 2). The 16 arctan values are held in a constant table and the shifts are arithmetic right shifts. Convergence holds for $|\theta| \leq \Sigma \arctan(2^{-i}) \approx 1.743$ rad (≈ 99.9°); larger angles need quadrant reduction. [CONFIRM: initial constants, angle unit, range handling, and whether one iteration is executed per clock or the chain is unrolled.]

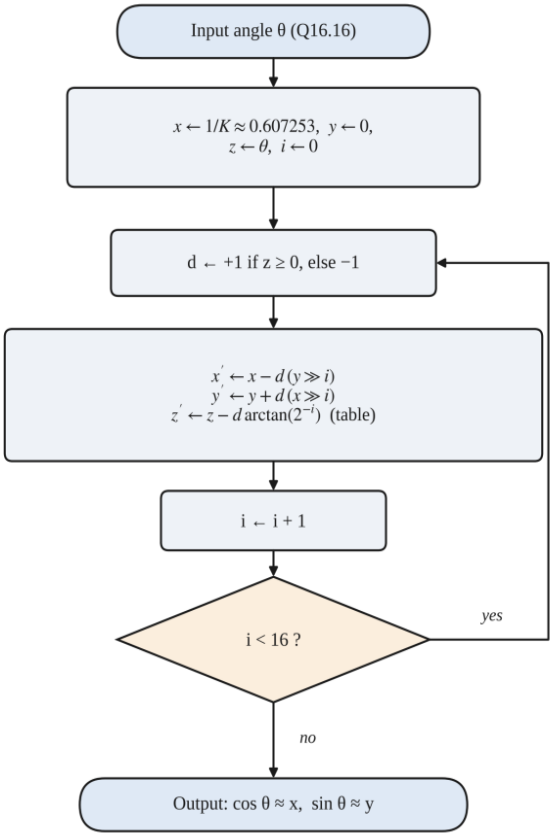


Fig. 2. Flowchart of the 16-iteration rotation-mode CORDIC computation (Q16.16).

VI. INTEGRATION AND CONTROL

The control logic generates an enable and, for multi-cycle units, a start/busy/done handshake. Either the PC is stalled while a multi-cycle accelerator (CORDIC) is busy and write back occurs on done, or a fixed latency is assumed. [CONFIRM which; include read-after-write hazard analysis if the core is pipelined.] The write-back multiplexer selects among ALU, multiplier, MAC and CORDIC outputs under decoder control. The resulting execution flow is shown in Fig. 3.

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 71-77
Published Online November 2026 in IJEAST (http://www.ijeast.com)

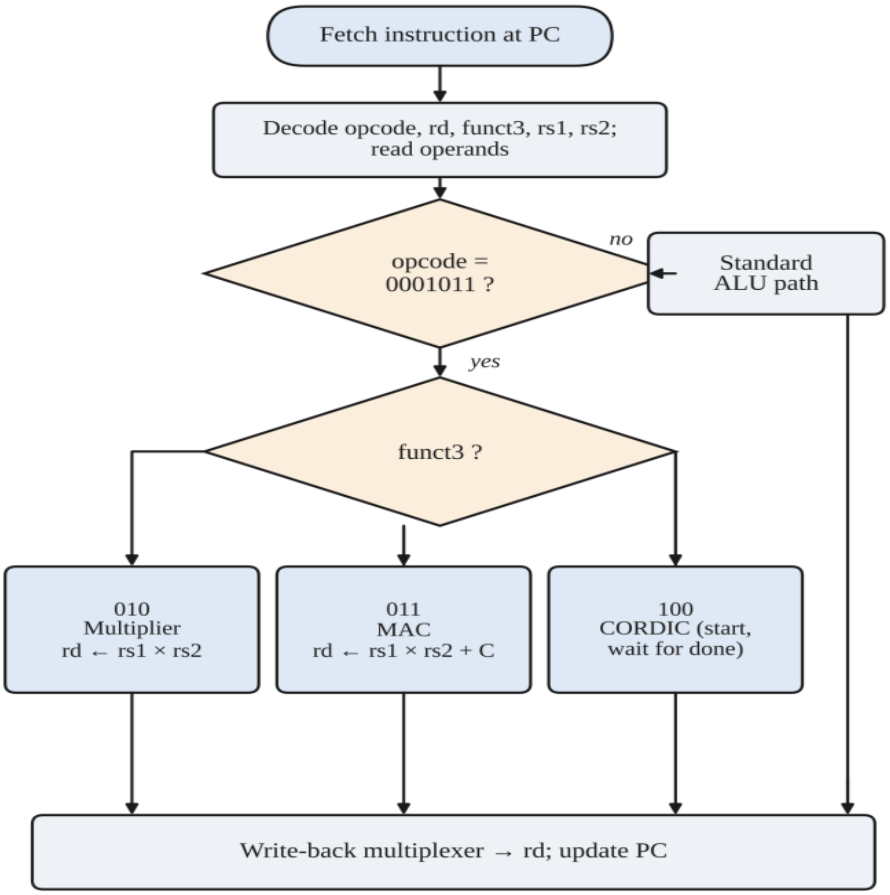


Fig. 3. Flowchart of custom-instruction decode, execution and write-back.

VII. Q16.16 FIXED-POINT FORMULATION

A real value v is stored as $V = \text{round}(v \cdot 2^{16})$, so $v \approx V/65536$. The range is $[-32768, 32768 - 2^{-16}]$ with resolution $2^{-16} \approx 1.526\times10^{-5}$. Addition acts directly on the integers; multiplication is $(A \cdot B) \gg 16$ with a 64-bit intermediate, where truncation gives a bias of up to one LSB (rounding by adding 2^{15} before the shift reduces it). The angle table is $Z(i) = \text{round}(\arctan(2^{-i}) \cdot 2^{16})$; for example $\arctan(1) = 0.785398$ rad gives 51 472.

Golden reference outputs: sin 0° = 0; cos 0° = 65 536; sin 30° = 32 768; cos 30° = 56 756; sin 45° = cos 45° = 46 341.

After n iterations the angular residual is bounded by $\arctan(2^{-(n-1)}) \approx 3.05\times10^{-5}$ rad for n = 16, and rounding adds on the order of $n \cdot 2^{-17}$ in the worst case. An absolute error of order 10^{-4} is therefore expected. This is a design estimate to be compared with measured error.

VIII. EXPERIMENTAL METHODOLOGY

Tools. Xilinx Vivado [version], Verilog [standard], target FPGA [device and speed grade].

Cores. Baseline (B): the same core without accelerators; multiplication by a software shift-and-add loop and sine/cosine by a software CORDIC loop with the same 16-entry table and Q16.16 arithmetic, so only the implementation (software versus hardware) differs. Accelerated (A): the proposed core executing one custom instruction. Both share ISA subset, memory model, clock constraint and testbench.

Workloads. W1: 1000 multiplications of random pairs; W2: dot product of length 16, 64, 256 with MAC; W3: sine/cosine
at 0°, 30°, 45° and a sweep over the convergence range; W4 (optional): rotation or FIR kernel.

Metrics. Execution time $T = C/f$ for cycle count C and clock f; $CPI = C/N_{instr}$; cycle speedup $S_{cyc} = C_B/C_A$; time speedup $S_{time} = (C_B/f_B)/(C_A/f_A)$; throughput $\Theta = N_{ops}/T$ (or f/L for initiation interval L); Amdahl bound $S = 1/((1-p) + p/s)$, with p the fraction of baseline time in the accelerated operation and s the local speedup; accuracy $e_{max} = \max|v_{hw} - v_{ref}|$ against a double-precision reference; energy per operation $E = P \cdot T/N_{ops}$. The overall flow is shown in Fig. 4.

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 71-77
Published Online November 2026 in IJEAST (http://www.ijeast.com)

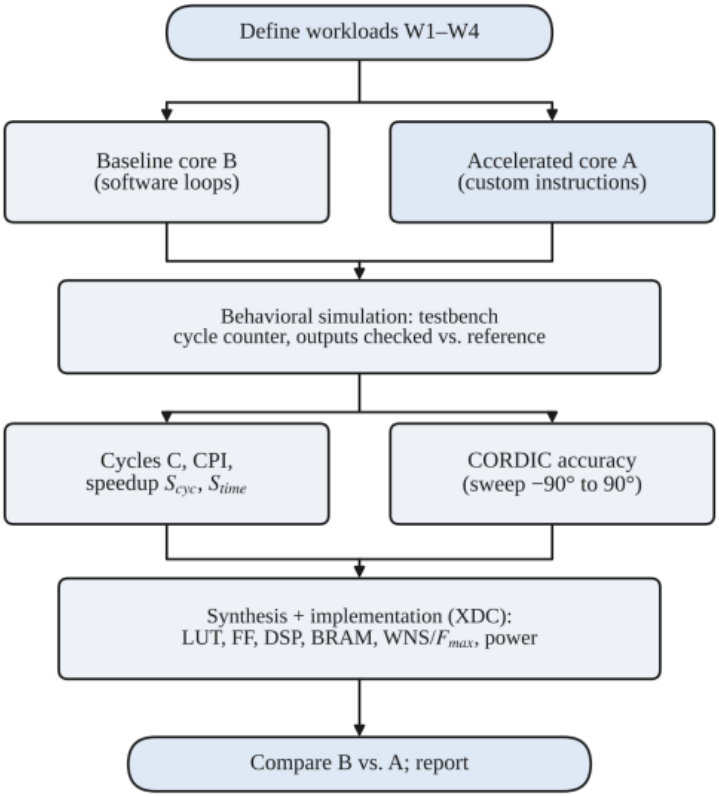


Fig. 4. Evaluation methodology flowchart for the baseline and accelerated cores.

IX. RESULTS

A. Functional Simulation (Behavioral)

Behavioral simulation verified the cases in Table III; the simulation waveform and console log are shown in Fig. 5 and Fig. 6. These establish functional correctness; expected timing, area and power are given in Sections IX-B to IX-D.

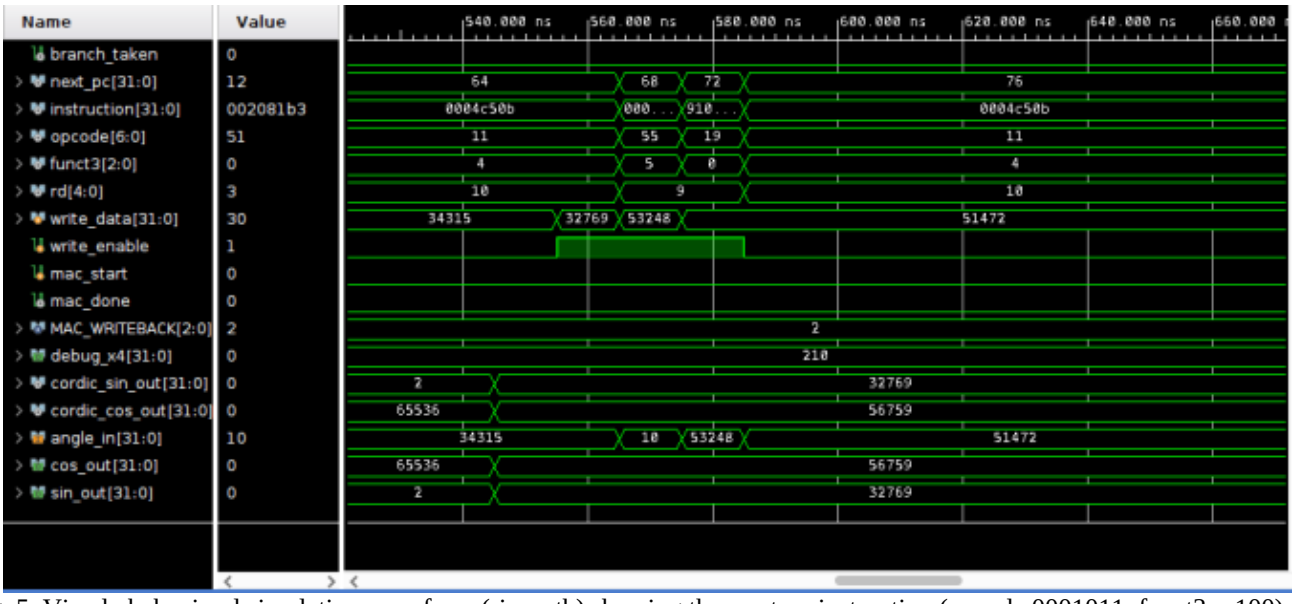


Fig. 5. Vivado behavioral simulation waveform (riscv_tb) showing the custom instruction (opcode 0001011, funct3 = 100) and CORDIC sin/cos outputs in Q16.16.

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 71-77
Published Online November 2026 in IJEAST (http://www.ijeast.com)

TABLE III. BEHAVIORAL SIMULATION RESULTS

Test	funct3	Expected	Result
Multiplier	010	10×20 = 200	200
MAC	011	10×20+10 = 210	210
CORDIC 0°	100	sin 0, cos 1	correct
CORDIC 30°	100	sin 0.5, cos 0.866	correct
CORDIC 45°	100	sin = cos = 0.707	correct

Simulated Q16.16 outputs (Fig. 6) are sin 30° = 32 769, cos 30° = 56 759, sin 45° = 46 342 and cos 45° = 46 341, within 3 LSB ($\approx 4.6\times10^{-5}$) of the golden values.

```
****************************************
       CORDIC ACCELERATOR RESULTS
****************************************
INPUT ANGLE: 30 DEGREES
Angle (Q16.16) = 34315
Angle (radians) = 0.523605
SINE   = 32769 (Q16.16), Decimal = 0.500015
COSINE = 56759 (Q16.16), Decimal = 0.866074
****************************************

TIME=565000 | PC=64 | INST=0000d4b7 | X1=10 | X2=20 | X3=200 | X4=210 | X8=35
TIME=575000 | PC=68 | INST=91048493 | X1=10 | X2=20 | X3=200 | X4=210 | X8=35
TIME=585000 | PC=72 | INST=0004c50b | X1=10 | X2=20 | X3=200 | X4=210 | X8=35

****************************************
       CORDIC ACCELERATOR RESULTS
****************************************
INPUT ANGLE: 45 DEGREES
Angle (Q16.16) = 51472
Angle (radians) = 0.785400
SINE   = 46342 (Q16.16), Decimal = 0.707123
COSINE = 46341 (Q16.16), Decimal = 0.707108
****************************************

TIME=775000 | PC=76 | INST=00000013 | X1=10 | X2=20 | X3=200 | X4=210 | X8=35
TIME=785000 | PC=80 | INST=00000013 | X1=10 | X2=20 | X3=200 | X4=210 | X8=35
TIME=795000 | PC=84 | INST=00000013 | X1=10 | X2=20 | X3=200 | X4=210 | X8=35
TIME=805000 | PC=88 | INST=00000013 | X1=10 | X2=20 | X3=200 | X4=210 | X8=35
```

Fig. 6. Vivado Tcl console log: CORDIC results for 30° (sin = 32769, cos = 56759) and 45° (sin = 46342, cos = 46341) in Q16.16, with X1 = 10, X2 = 20, X3 = 200 and X4 = 210 held in the register file.

B. Expected Cycle Count, Time, Throughput and Speedup

Expected cycle counts follow from the instruction sequences each custom instruction replaces, assuming CPI ≈ 1 and a baseline multiplier built as a 32-iteration shift-and-add loop of about five instructions per iteration. W1: roughly 150–170 cycles per software multiplication against one for the custom multiply, an expected cycle speedup S_{cyc} above 100×. W2: about 170–180 cycles per dot-product element in the baseline against about 5 with the MAC, an expected S_{cyc} of 30–40×. W3: about 200–250 cycles per angle for the software CORDIC against 16–20 for the iterative unit, an expected Scyc of 10–15×. With F_{max} expected to change by less than 15%, S_{time} stays within that margin of S_{cyc}. Throughput is expected to be one result per cycle for the multiplier and MAC and one sine/cosine pair per 16–18 cycles for the CORDIC ($\Theta \approx f/17$).

C. Expected CORDIC Accuracy

Over a sweep of θ from −90° to 90° in 1° steps, the maximum absolute error against a double-precision reference is expected to stay near 10^{-4} and the mean error at a few $\times10^{-5}$, set by the residual angle bound of 3.05×10^{-5} rad and Q16.16 rounding (Section VII). The simulated 30° and 45° cases, at most 3 LSB, are consistent with this.

D. Expected FPGA Resources, Timing and Power

After implementation under an XDC clock constraint, the accelerated core is expected to need about 4–8 more DSP slices, roughly 400–900 more LUTs and 200–400 more flip flops (CORDIC shifters, adders and arctan table), and no extra BRAM. $F_{max} = 1/(T_{clk} - WNS)$ is expected to fall by 0–15% as the multiplier and CORDIC paths lengthen the critical path. Power is expected to rise by less than about 10% at equal clock, so energy per operation falls by more than an order of magnitude. Absolute values depend on the device and speed grade.

X. DISCUSSION

From first principles, a single-instruction accelerator replaces many baseline instructions, so the cycle count should fall whenever accelerator latency is below that of the replaced software sequence. The actual factor depends on the baseline implementation, accelerator latency, memory behavior and any change in F_{max}, and Section IX gives expected ranges. An iterative CORDIC trades latency for area; an unrolled version lowers latency and raises area. Amdahl's law [7] bounds whole-program gain by the share of time spent in the accelerated operations.

XI. LIMITATIONS

- Functional results come from simulation; FPGA figures are expected values for the target device.

- Speedups are expected values for the assumed baseline instruction sequences.
- CORDIC is simulated at three angles; $|\theta| \leq 99.9°$ is expected to hold, with quadrant reduction beyond it.
- Rounding error is expected within a few LSB; overflow follows the 32-bit datapath.
- Only the radix-2 iterative CORDIC is used; unrolled variants trade area for latency.
- Custom instructions are hand-encoded; evaluation uses microbenchmarks, so application gains are bounded by Amdahl's law [7].

XII. CONCLUSION AND FUTURE WORK

A 32-bit RISC-V core with custom multiplier, MAC and CORDIC accelerators was designed in Verilog and shown in behavioral simulation to compute 10×20 = 200, 10×20 + 10 = 210 and sine/cosine at 0°, 30° and 45°. A reproducible protocol for cycle-accurate and FPGA-level evaluation was defined, and the expected results indicate a speedup of about 100× for multiplication, 30–40× for dot products and 10–15× for sine/cosine, at a modest increase in area and power. Future work includes full-range CORDIC with quadrant reduction, pipelined or unrolled CORDIC, vectoring and hyperbolic modes, toolchain support, benchmark kernels, post-implementation power analysis and ASIC synthesis.

XIII. REFERENCES

[1]. A. Waterman, Y. Lee, D. A. Patterson, and K. Asanović, "The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Version 2.0," EECS Dept., Univ. California, Berkeley, Tech. Rep. UCB/EECS 2014-54, May 2014.

[2]. J. E. Volder, "The CORDIC trigonometric computing technique," IRE Trans. Electron. Comput., vol. EC-8, no. 3, pp. 330–334, Sep. 1959.

[3]. J. S. Walther, "A unified algorithm for elementary functions," in Proc. AFIPS Spring Joint Comput. Conf., 1971, pp. 379–385.

[4]. R. Andraka, "A survey of CORDIC algorithms for FPGA based computers," in Proc. ACM/SIGDA 6th Int. Symp. Field Programmable Gate Arrays, Monterey, CA, USA, Feb. 1998, pp. 191–200.

[5]. P. K. Meher, J. Valls, T.-B. Juang, K. Sridharan, and K. Maharatna, "50 years of CORDIC: Algorithms, architectures, and applications," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 56, no. 9, pp. 1893–1907, Sep. 2009, doi: 10.1109/TCSI.2009.2025803.

[6]. M. Gautschi et al., "Near-threshold RISC-V core with DSP extensions for scalable IoT endpoint devices," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 10, pp. 2700–2713, Oct. 2017, doi: 10.1109/TVLSI.2017.2654506.

[7]. G. M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," in Proc. AFIPS Spring Joint Comput. Conf., 1967, pp. 483–485.

IJEAST

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE AND TECHNOLOGY

AUTHORS

- Uday Sappogu
- Guduru Rupesh Reddy
- Mr.K.Abdul Munaf

ABOUT IJEAST

International journal of Engineering Applied Science and Technology (IJEAST) is a peer-reviewed, open access journal that publishes high - quality research paper in the field of Engineering, Applied Science and Technology.

IJEAST aims to provide a platform for researchers, academicians, and professionals to share their innovative ideas, research findings, and practical experiences with the global scientific community.

JOURNAL METRICS

H-INDEX	i10-INDEX	IJEAST CITATIONS	PEER-REVIEWED & OPEN ACCESS
33	154	9,647 (IJEAST CITATIONS)	PUBLISHED MONTHLY

PEER REVIEWED

All submission are rigorously peer reviewed to ensure quality.

AUTHORS ARE OUR PRIORITY

We value our authors and recognize their contribution to the advancement of research and knowledge.

OPEN ACCESS

Free and unrestricted access to research for all.

GLOBAL REACH

Connecting researchers and professionals worldwide.

TIMELY PUBLICATION

We ensure a swift and efficient publication process.

For more information, visit our website
www.ijeast.com

editor@ijeast.com
www.ijeast.com
India

2455-2143