Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

Multiplier-Less 3D Rotation and Orthographic Projection Architecture Using Cascaded CORDIC on Artix-7 FPGA

Pavithra P*, Prakruthi D R*, T Gyan Santosh*, Vikas Singh Lodhi*
Dept. of Electronics and Communication Engineering
Dr. Ambedkar Institute of Technology
Bengaluru, India
*(*All undergraduate authors contributed equally to this work)*

Dr. Nayana R Shenoy
Associate Professor, Dept. of Electronics and Communication Engineering
Dr. Ambedkar Institute of Technology
Bengaluru, India

Abstract* - Real-time three-dimensional wireframe rendering on embedded and edge-display platforms often suffers from power constraints, area overhead, and memory latencies of graphics processor units and FPGA soft core CPU implementations. This research paper presents a zero-processor, all Register-Transfer Level (RTL) 3D wireframe graphics rendering engine implemented on a Xilinx Artix-7 (xc7a100tcsg324-1) FPGA, without using any DSP48E1 multiplier blocks. A seven-state master finite state machine controls the entire datapath clocked at 100 MHz, where eight Q8.8 fixed point coordinates of the cube vertices are streamed from internal Read-Only Memory to a sequential three-axis Coordinate Rotation Digital Computer (CORDIC) engine followed by a single cycle orthographic projection engine and integer only Bresenham line drawing rasterizer, which ultimately renders the data to a standard 640 by 480 pixels VGA output at 60 Hz refresh rate. The trigonometric rotations, CORDIC gain correction and line error correction logic is performed using only carry chain additions and arithmetic bit shifts. The post route design in Vivado 2018.3 is capable of meeting timing constraints at 100 MHz with worst negative slack of 0.311 ns and consumes 2,797 slice LUTs (4.41%), 1,839 flip flops (1.45%), 0 DSP blocks and 0.154 W power consumption, which was validated on live Digilent Nexys A7 platform.

Keywords—Artix-7 FPGA, Bresenham line algorithm, CORDIC, DSP-free arithmetic, pure-RTL graphics engine, VGA

I. INTRODUCTION

In this paper, a novel design of a zero-processor, pure-RTL 3D wireframe graphics engine running on a Xilinx Artix-7 FPGA is presented. The goal is to create DSP-free arithmetic by making the geometry stage an iterative 3-axis CORDIC engine combined with orthographic projection, while the rasterization stage would consist of an integer-only Bresenham line drawing engine connected with a traditional VGA output. A simple hardware finite state machine (FSM) will coordinate all the stages of the pipeline directly in Verilog RTL and thus make a CPU completely unnecessary.

Background and Motivation: Embedded devices such as portable medical monitors, car HUDs, and industrial control panels often require real-time 3D visualization. While a typical GPU is capable of rendering high frames per second, its big driver stack, dynamic threads scheduling and high thermal design power (TDP) makes it unfeasible to be used in compact and power-constrained enclosure. Moreover, even specialized mobile 3D accelerators still require a host processor (e.g. RISC or application) to perform 3D geometry transformation and control the whole process, introducing extra memory bus traffic and power consumption [6], [7]. Thus, FPGAs become an alternative to enable designer to create a customized hardware pipeline specifically optimized for particular graphics task. However, many FPGA-based 3D graphics engines are still using a soft-core processor, large floating-point IP or a hardware multiplier (DSP slices) and BRAM table to calculate rotation matrix [8], [9]. Removing processor overhead and hardware multiplier is a feasible step towards compact low-power embedded 3D displays.

Contributions: This work has the following main contributions:
- A pure-RTL graphics pipeline where vertex fetching, 3D rotation, projection, line rasterization and VGA scan-out are coordinated by a single hardware FSM without using any processor.
- A DSP-free 3-axis iterative CORDIC engine in 16 bit signed Q8.8 fixed-point format with the two-phase shift-and-add datapath to reduce the critical path and shift-and-add gain compensation block, not requiring any DSP48 slices.
- An integer-only Bresenham rasterizer with standard VGA interface and demonstrated working in behavioral simulation and as a part of a live 3D cube rotation on an Artix-7 FPGA.

Organization: The rest of the paper is organized as follows. In section II, previous works on low-power 3D hardware, CORDIC engines on FPGA, and hardware rasterizers are surveyed. In section III, mathematical background and pure-RTL architecture are outlined. Section IV gives simulation waveforms, Artix-7 resource usage, timing, power estimates and comparison benchmarks. Section V concludes the paper.

II. RELATED WORK

Low-Power Embedded 3D Graphics Architectures

The design challenge of achieving tight power and area bounds for real-time 3D graphics has been one of the focal points in handheld and embedded hardware. Nam et al. [6] have designed the RamP-Lite 3D rendering engine for handheld terminals and have shown that switching from floating point to Q8.8 fixed point results in significant power savings and area reduction.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

However, RamP-Lite operates as an auxiliary chip that needs a PXA-255 processor to run software implementations of geometry transformations, clipping, and projection. Later on, Nam et al. [7] have designed a mobile GPU called RamP-VI that integrates an ARM10 compatible RISC core, logarithmic arithmetic units, and triple domain DVFS to operate at 52.4 mW with 60 frames per second. Even though processor-driven SoCs are useful in custom mobile ASIC applications, they are redundant in terms of software scheduling overhead and silicon complexity required for FPGA-based wireframe displays.

Hardware Rotation Engines Using CORDIC Algorithm

To eliminate power-hungry multipliers in computations involving heavy trigonometry, Volder [1] proposed CORDIC algorithm which rotates vectors by using shift-and-add operations. Li and Xin [3] have implemented 16-bit pipelined CORDIC on Altera Cyclone II FPGA by cutting down pipeline to nine stages and using carry-storage adder to save on logic and increase precision. Garrido et al. [4] have developed CORDIC II, which uses friend angles, uniformly-scaled redundant (USR) stages, and nano-rotations in place of standard micro-rotations, thus reducing number of adders by up to 30.4% and almost cutting latency of pipelined rotators in half. Li et al. [5] have proposed a pre-rotation CORDIC (PR-CORDIC) design on a Xilinx Kintex-7 FPGA which gets rid of redundant iterations to cut down latency in half without using any DSP slices or memory when input values are fixed point. As pointed out in [5], variable pre-rotation ruins the constant CORDIC scale factor and thus makes magnitude preserving rotations problematic unless there is a compensation hardware involved. Fully unrolled pipelined CORDIC cores are resource hungry as they use significant amounts of LUTs and Flip-Flops when applied to all three axes (X, Y, Z), making iterative constant-scaling 3-axis design more efficient for small FPGAs.

FPGA-based Rasterization and 3D Graphics Pipelines

Bresenham's algorithm [2] has proven to be the preferred choice for hardware line rasterization since it operates using integer addition and bit shifts, thus eliminating need for floating-point units and dividers. Younis and Sheet [8] have developed a modified 3D-Bresenham unit with dual-port BRAM framebuffer on a Xilinx Spartan-3E FPGA and have achieved operating frequency of 68.33 MHz consuming 266 LUTs and 541 Flip-Flops. In a later work, Younis and Sheet [9] have designed a dynamic three-dimensional graphics processor implemented on a Spartan-3E XC3S1600E FPGA, incorporating geometric transformation, scan conversion, and a VGA refresh controller. Though both previous designs were hardware driven, they still used dedicated hardware multipliers (MULT18X18SIO) and the transformation unit designed in [9] relied on dual-port BRAM lookup tables to store precomputed sine and cosine values.

The proposed architecture avoids usage of host/soft-core processors and dedicated hardware multipliers (0 DSPs). The combination of two-phase iterative 3-axis CORDIC engine with shift-and-add gain compensation and integer-only Bresenham rasterizer implemented in pure Verilog RTL allows to get a compact and deterministic 3D wireframe pipeline on Artix-7 FPGA.

III. PROPOSED PURE-RTL HARDWARE ARCHITECTURE

The suggested three-dimensional wireframe graphics core is designed as a completely synchronous, processor-less Register-Transfer Level (RTL) pipeline working at 100 MHz within a Xilinx Artix-7 FPGA device. Different from standard embedded graphics cores which rely either on a software processor core or DSP48E1 multipliers to compute matrix rotations, all geometric rotation, projection and rasterization processes are performed by integer-only hardware modules making use of shift-add operations. As seen in Figure 1, the datapath transfers eight vertexes of a cube stored within the internal read-only memory (vertex_rom) through a three-axes CORDIC rotation module (rotation_3d), a one-cycle orthographic projection process (orthographic_projection), a projected vertex register bank, a 12-edges Bresenham line rasterization process (edge_bresenham_top and bresenham_line), and a standard VGA output interface (vga_controller).

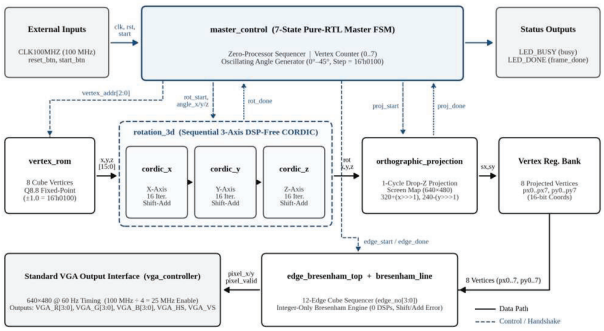

Fig. 1. The RTL-based hardware architecture of a zero processor and no digital signal processor three-dimensional wireframe graphics system. The datapath shown is the top level and shows the datapath starting from vertex_rom and passing through a sequential three-axis CORDIC rotation followed by orthographic projection, then integer Bresenham rasterization ending at the usual VGA interface

A. Zero-Processor Master Control FSM

Frame-level sequencing is controlled via a seven-state finite state machine (master_control) which removes the overhead of fetching instructions and processor bus latency. As shown in Figure 2, starting from the synchronous reset, the controller goes to S_IDLE and busy gets asserted on detecting the start signal. For each of the eight vertices (vertex_addr 0..7), the FSM goes from S_READ to fetch the signed 16-bit Q8.8 fixed point coordinates (+1.0=16'h0100, -1.0=16'hFF00) of the vertices stored in vertex_rom, S_ROTATE to invoke rotation_3d, S_PROJECT to wait for rotation_done and generate a projection_start signal and S_NEXT to hold the resulting screen coordinates. After all eight vertices have been transformed (vertex_addr==7), the FSM goes to S_EDGE and then to S_FRAME_DONE to generate all the twelve wireframe edges. To get smooth real time animation at 100 MHz, master_control updates the three axis Euler angles (angle_x, angle_y, angle_z) every fourth frame using a 2-bit

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

speed_counter, switching between 0° (16'h0000) and 45° (16'h2D00) with 1° (16'h0100) increments.

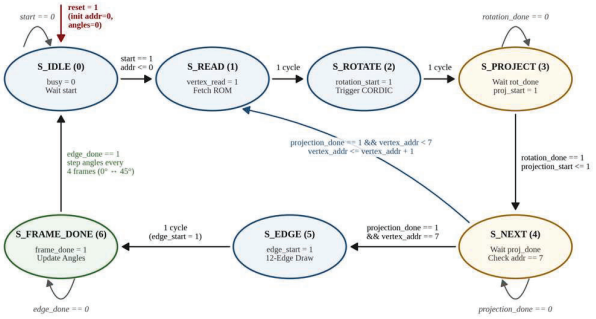

Fig. 2. Seven-state finite state machine (master_control) governing per-vertex transformation, sequential 12-edge rasterization, and oscillating rotation-angle updates without a soft-core processor.

B. DSP-Free Three-Axis CORDIC Rotation

Rotation of an object in three dimensions is accomplished in a cascade sequence of X, Y, and Z axis rotations by using three CORDIC cores (cordic_x, cordic_y, and cordic_z) in the rotation_3d module. Rotating in circular rotation mode in 16 micro-rotations (i = 0 … 15), each core uses 32-bit internal registers to maintain high precision and substitutes multiplication with right shift and addition operations. At iteration i, the coordinate pair (u, v) and angle θ are updated as a function of the sign of the θi angle as indicated by (1) and (2):

$$u_{i+1} = u_i \mp (v_i \gg i), \quad v_{i+1} = v_i \pm (u_i \gg i) \qquad (1)$$

$$\theta_{i+1} = \theta_i \mp \arctan(2^{-i}) \qquad (2)$$

In the above equation set, elementary angles arctan(2−i) values are looked up from a 16-entry combinational lookup table in Q8.8 fixed-point representation ranging from 16'h2D00 (45°) at i = 0 to 0 at i = 15. To address the inherent CORDIC scaling factor (K ≈ 0.60725) without using FPGA DSP blocks, each core uses a multiplier-free shift-and-add algorithm to approximate 311/512 (≈ 0.60742) as follows:

$$Y_{corr} = ((Y \ll 8)+(Y \ll 5)+(Y \ll 4)+(Y \ll 2)+(Y \ll 1)+Y) \gg 9 \qquad (3)$$

The shift-and-add network is accurate to a gain-compensation ratio of 0.028% by using only carry-chain adders and fixed bit-wiring shifts, while having 0% DSP48E1 usage in all three rotation stages.

C. Orthographic Projection and Integer Bresenham Rasterizer

Following the 3D rotation process, orthographic_projection does a single-cycle drop-Z projection and forwards the rotated coordinates (x, y) to a screen space mapper based on the combinational function of a 640 × 480 display as shown in (4):

$$x_{scr} = 320 + (x_{proj} \gg 1), \quad y_{scr} = 240 - (y_{proj} \gg 1) \qquad (4)$$

Eight mapped vertices (px0–px7, py0–py7) are registered and sent to edge_bresenham_top that sequentially issues all twelve edges of the cube (four back-face edges, four front-face edges, and four depth-connecting edges) to bresenham_line. The line drawing routine calculates integer deltas (dx, dy), unit step directions (sx, sy), and the decision parameter err = dx − dy; the error term is doubled by using a left shift operation (err <<< 1), resulting in pixel coordinates (pixel_x, pixel_y) at one pixel per clock cycle without multipliers.

D. Standard VGA Output Interface

The vga_controller module sends a standard 640 × 480 at 60 Hz VGA signal directly from the 100 MHz system clock with a 2-bit divide-by-four counter (clk_div == 2'd3) to produce a 25 MHz pixel-enable pulse without the need of using a separate MMCM or PLL tile. HORIZONTAL and VERTICAL counters create active-low synchronization pulses (VGA_HS, VGA_VS) and gate the 12-bit RGB signals (VGA_R[3:0], VGA_G[3:0], VGA_B[3:0]).

IV. EXPERIMENTAL RESULTS AND DISCUSSION

Functional Simulation and Verification

The developed 3D wireframe graphics engine was simulated using behavioral simulation in Verilog HDL in Xilinx Vivado before synthesis. Specific testbenches were used for the 3-axis CORDIC rotation engine and Bresenham line rasterizer to check numerical accuracy and handshake timing. During CORDIC behavioral simulation, the 16-bit signed coordinates of the vertex were rotated through arbitrary Q8.8 fixed-point angles, which showed that each axis finishes 16 two-phase rotations in 32 cycles (33 including input sampling) and uses gain compensation factor of 311/512 corresponding to shift-and-add algorithm with great accuracy. Bresenham rasterizer was tested for all octants and slopes of lines and it was shown to produce pixels in one cycle without divisions and floating point operations. Fig. 3 shows the results of behavioral simulation waveform for 3-axis rotation and line rasterization steps.

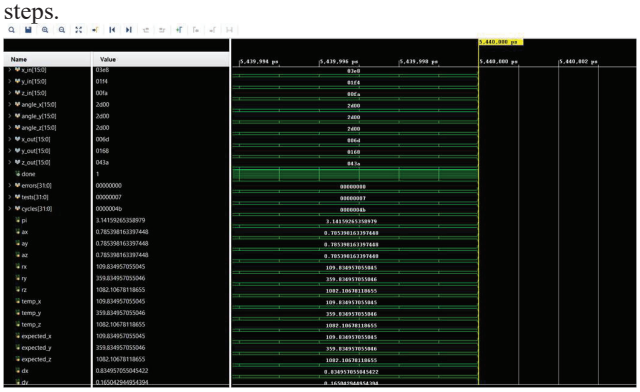

Fig. 3. Vivado simulation results of the 3D CORDIC rotation module at 5,440 ns (tests = 7, errors = 0, done = 1). For input coordinates (1000,500,250) rotated by 45° (0x2d00, 0.7854 rad) about all three axes, the RTL outputs (109,360,1082) match the double-precision reference (109.83,359.83,1082.11) within 1 LSB per axis.

FPGA Resources Utilization, Timing, and Power Estimation

After functional verification, the design was synthesized, placed, and routed to Xilinx Artix-7 FPGA (XC7A100T-1CSG324C) using Xilinx Vivado 2018.3 tool. Table I summarizes post-implementation resources utilization, timing and power estimates taken from post route Vivado reports.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

TABLE I. POST-IMPLEMENTATION RESOURCE UTILIZATION, TIMING, AND POWER ON ARTIX-7 FPGA (XC7A100T)

Resource / Metric	**Used**	**Available**	**Utilization (%)**
Slice LUTs	2,797	63,400	4.41%
Slice Registers (FFs)	1,839	126,800	1.45%
Slices	885	15,850	5.58%
Block RAM Tiles (RAMB36E1)	20	135	14.81%
DSP48 Slices	0	240	0.00%
Bonded IOBs	19	210	9.05%
Target Clock Frequency	100.00 MHz	—	—
Worst Negative Slack (WNS)	+0.311 ns	—	—
Achieved Frequency (Fmax)	103.21 MHz	—	—
Total On-Chip Power	154 mW	—	—
Dynamic Power	56 mW	—	—
Device Static Power	98 mW	—	—

As seen in Table I, the complete geometric and rasterization pipeline works without using any DSP48 slices, which ensures that all 3D rotations, gain compensations, and line drawing are executed only using shift-add operations. Using one iterative two-phase datapath for each axis keeps the logic utilization at 4.41% of LUTs and 1.45% of Flip-Flops, and thus allows to leave more than 95% of Artix-7 resources for extra embedded tasks. The design works at 100 MHz system clock with zero failing endpoints (+0.311 ns setup slack and +0.024 ns hold slack), and therefore the maximum possible operating frequency is 103.21 MHz. On-chip power dissipation is 154 mW (56 mW dynamic and 98 mW static) at 25.7 °C junction temperature; the three-axis rotation core generates 27 mW of dynamic power while Block RAM dissipates 5 mW.

Live Hardware Demonstration

In order to verify the real-time work of the engine, the bitstream was programmed into an Artix-7 evaluation board connected to a common 640 × 480 @ 60 Hz VGA screen. Under the full control of the central finite-state machine without any host or soft-core processor, the engine reads the vertices of the cube from the on-chip ROM, performs real-time 3-axis CORDIC rotations of the vertices, projects them into 2D screen plane, and finally, does the rasterization of the wireframe edges.

Comparative Analysis

TABLE II. BENCHMARK COMPARISON WITH EXISTING FPGA ARCHITECTURES

Parameter	**[8]**	**[9]**	**[5]**	**Proposed**
FPGA	Spartan-3E	Spartan-3E	Kintex-7	**Artix-7**
Scope	3D-Bres. + VGA	Transf. Unit	2D PR-CORDIC	**Full 3D + VGA**
LUTs	266	69	390	**2,797**
FFs	541	82	56	**1,839**
BRAM	19	1	0	**20**
DSP / Mult	2	2	0	**0**
Fmax (MHz)	68.33	50.00	N/R	**103.21**

Sources: [8] Younis & Sheet (XC3S500E), [9] Younis & Sheet (XC3S1600E), [5] Li et al. (16-bit fixed-point iterative).

The two main benefits of architecture are:
1) Full Elimination of Hardware Multipliers: Both [8] and [9] use FPGA implementations that require two 18x18 multipliers (MULT18X18SIO). Moreover, in [9] the transformation unit also uses dual-port BRAM to hold sine/cosine lookup table. On the contrary, our design achieves 3D rotations and line rasterization without any DSP slices and without using trigonometric BRAM tables at the same time while operating at the clock frequency of 103.21 MHz.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

2) Full-Pipeline Implementation at Low Power Consumption: While Li et al. [5] provide DSP-free fixed-point iterative CORDIC implementation that requires 390 LUTs for a single two-dimensional arctangent unit, the current design extends DSP-free shift-and-add rotations to three spatial axes and implements it together with orthographic projection, Bresenham rasterization and 640 x 480 VGA interface into one zero-processor RTL pipeline consuming only 56 mW of dynamic power.

V. CONCLUSION

This paper analyzes the architecture, fixed-point design, and FPGA implementation of a zero-processor, pure-RTL 3D wireframe graphics engine, which has been developed based on a Xilinx Artix-7 platform (XC7A100T). The system utilizes a cascade 3-axis iterative CORDIC rotator, canonical shift-and-add gain compensation technique, orthographic projection, and an integer-only Bresenham line rasterizer that is controlled by the central hardware finite state machine. Thus, the system avoids overhead associated with soft core processor utilization as well as usage of dedicated hardware multipliers. After the implementation, the system operates at a clock frequency of 103.21 MHz (worst negative slack 0.311 ns at 100 MHz) consuming only 2,797 LUTs (4.41%), 1,839 Flip-Flops (1.45%), 20 BRAM tiles (14.81%), and 0 DSP48 slices, and the overall on-chip power consumption amounts to 154 mW (dynamic 56 mW). Hardware evaluation conducted using the Artix-7 board proved that the proposed pure-RTL, multiplier-less architecture works successfully for 3D wireframe rendering in real-time on a standard VGA monitor. Future research will focus on framebuffer hardware-level optimization, perspective projection, and hidden-surface removal.

REFERENCES

[1] J. E. Volder, "The CORDIC Trigonometric Computing Technique," IRE Trans. Electron. Comput., vol. EC-8, pp. 330–334, Sep. 1959.
[2] J. E. Bresenham, "Algorithm for Computer Control of a Digital Plotter," IBM Systems Journal, vol. 4, no. 1, pp. 25–30, 1965.
[3] H. Li and Y. Xin, "Modified CORDIC Algorithm and Its Implementation Based on FPGA," in Proc. 3rd Int. Conf. Intelligent Networks and Intelligent Systems (ICINIS), 2010, pp. 618–621.
[4] M. Garrido, P. Källström, M. Kumm, and O. Gustafsson, "CORDIC II: A New Improved CORDIC Algorithm," IEEE Trans. Circuits Syst. II, Express Briefs, vol. 63, no. 2, pp. 186–190, 2016.
[5] K. Li, H. Fang, Z. Ma, F. Yu, B. Zhang, and Q. Xing, "A Low-Latency CORDIC Algorithm Based on Pre-Rotation and Its Application on Computation of Arctangent Function," Electronics, vol. 13, no. 12, p. 2338, Jun. 2024.
[6] B.-G. Nam, M.-W. Lee, and H.-J. Yoo, "Development of a 3-D Graphics Rendering Engine with Lighting Acceleration for Handheld Multimedia Systems," IEEE Trans. Consumer Electron., vol. 51, no. 3, pp. 1020–1027, Aug. 2005.
[7] B.-G. Nam, J. Lee, K. Kim, S. Lee, and H.-J. Yoo, "Cost-Effective Low-Power Graphics Processing Unit for Handheld Devices," IEEE Commun. Mag., pp. 152–159, Apr. 2008.
[8] B. M. K. Younis and N. S. M. Sheet, "Hardware Implementation of 3D-Bresenham's Algorithm Using FPGA," Tikrit Journal of Engineering Sciences, vol. 20, no. 2, pp. 37–47, Mar. 2013.
[9] B. M. K. Younis and N. S. M. Sheet, "Real Time Dynamic 3D Graphics Processor Using FPGA," IJRDE, 2013.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)