VLSI Implementation of CORDIC-Based Geometry Rotation for High-Speed 3-D Computer Graphic Systems

Tze-Yun Sung[1] and Hsi-Chin Hsin[2]
[1]*Department of Microelectronics Engineering, Chung Hua University, Hsinchu, Taiwan 300-12, R. O. C.*
[2]*Department of Computer Science and Information Engineering, Formosa University, Hu-Wei, Yunlin, Taiwan 632-08, R. O. C*
E-mail:bobsung@chu.edu.tw

ABSTRACT

High performance architectures for the applications of intensive data with constraints on latency can be achieved by maximizing both parallelism and pipelining. In this paper, the hardware primitives of 3-D geometry rotation based on the redundant CORDIC arithmetic are presented. The proposed CORDIC based architecture has been implemented by VLSI for high-throughput power-aware 3-D computer graphic systems. The overall performance of 3-D vector interpolation and rotation can be improved significantly by using the proposed architecture, in which only one CORDIC computation time is required.

Keywords: Redundant CORDIC arithmetic, 3-D geometry rotation, vector interpolation, high-throughput, power-aware.

1: INTRODUCTION

Flexible hardware with precision control is much desirable for the power-aware 3-D graphics rendering applications [1], in which 3-D vector interpolation is one of the most important tasks. The 3-D CORDIC-based vector interpolator of Euh, et al. provides multiple precisions for the design of power-aware systems [2]. Lang and Antelo proposed the CORDIC-based geometry operations in 3-D computer graphics [3]. Arrigo and Chau proposed the CORDIC processing element for the computationally intensive operations of the quaternion-based kinematic altitude calculation [4]. Jiang, et al. proposed the compensated CORDIC algorithm for the real-time image rotation applications [5]. Sung, Shieh and Tsai proposed the power-aware vector interpolator based on the redundant CORDIC arithmetic [6]. The overview of the conventional 3-D rotation in Cartesian coordinate can be found in [7].

The well known CORDIC algorithm, which has been widely utilized to implement many signal processing tasks such as sinusoidal wave generation, vector rotation, coordinate transformation, and linear system solving, is suitable for the implementation of 3-D geometry rotation [8]-[9]. In CORDIC, only simple shifters and adders are needed, which can be efficiently realized by using reconfigurable hardware platforms, especially FPGA [10]. Thus, the CORDIC-based 3-D geometry rotation is suitable for the development of 3-D computer graphic systems. In this paper, the architecture of 3-D geometry rotation based on the CORDIC algorithm is proposed. It is suitable for VLSI implementation in terms of the computational complexity. The detailed operations, hardware requirements, and advantages of the proposed architecture are presented in the following sections.

The remainder of this paper proceeds as follows. In section 2, the conventional 2-D CORDIC algorithm is reviewed. The proposed 3-D CORDIC algorithm is given in section 3. The VLSI architecture of the proposed CORDIC algorithm and its implementation for 3-D geometry rotation are presented in section 4. The experimental results, numerical analysis and verification platform are presented in section 5. The advantages of the proposed architecture in comparison with the previous works are given in section 6. The conclusion can be found in section 7.

2: THE CONVENTIONAL 2-D CORDIC ALGORITHM

CORDIC (COordinate Rotation DIgital Computer) is an algorithm that performs a sequence of iteration computations for coordinate rotation [8]-[9]. CORDIC can be used to generate many important elementary functions by only simple adders and shifters. The basic CORDIC iteration equations are as follows:

$$x_{i+1} = x_i - m\sigma_i 2^{-s(m,i)} y_i \quad (1)$$

$$y_{i+1} = y_i + \sigma_i 2^{-s(m,i)} x_i \quad (2)$$

$$z_{i+1} = z_i - \sigma_i \alpha_{m,i} \quad (3)$$

where m denotes the circular (m=1), linear (m=0) or hyperbolic (m=-1) coordinate system, i=0, 1,2,…., n-1,

$$s(m,i) = \begin{cases} 0,1,2,3,4,5,...., m = 1 \\ 1,2,3,4,5,6,...., m = 0 \\ 1,2,3,4,4,5,...., m = -1 \end{cases},$$

$$\alpha_{m,i} = m^{-1/2} \tan^{-1}[\sqrt{m}\, 2^{-s(m,i)}] \quad (4)$$

$\sigma_i = sign(z_i)$ with $z_i \to 0$ in the rotation mode, and $\sigma_i = -sign(x_i) \cdot sign(y_i)$ with $y_i \to 0$ in the

vectoring mode. The scale factor $k_{m,i} = \sqrt{1 + m\sigma_i^2 2^{-2s(m,i)}}$ in the i-th iteration. After n iterations, the product of all the scale factors is as follows:

$$K_m = \prod_{i=0}^{n} k_{m,i} = \prod_{i=0}^{n} \sqrt{1 + m\sigma_i^2 2^{-2s(m,i)}} = \prod_{i=0}^{n} \sqrt{1 + m2^{-2s(m,i)}} \quad (5)$$

where the rotation direction is defined by $\sigma_i \in \{-1, +1\}$.

3: THE PROPOSED 3-D CORDIC ALGORITHM

Figure 1 shows a vector R in the 3-D space. Its Cartesian and spherical coordinates are denoted by (X_i, Y_i, Z_i) and (R_i, θ_i, ϕ_i), respectively. $\boldsymbol{R}$ is to be rotated yielding a new vector $\boldsymbol{S}$ with Cartesian and spherical coordinates $(X_{i+1}, Y_{i+1}, Z_{i+1})$ and $(R_i, \theta_i + \alpha_i, \phi_i + \beta_i)$, respectively. The relationship between the Cartesian and spherical coordinates of the original vector $\boldsymbol{R}$ and the rotated vector $\boldsymbol{S}$ are given by

$$X_i = R_i \cos\theta_i \sin\phi_i \quad (6)$$

$$Y_i = R_i \sin\theta_i \sin\phi_i \quad (7)$$

$$Z_i = R_i \cos\phi_i \quad (8)$$

$$X_{i+1} = R_i \cos(\theta_i + \alpha_i)\sin(\phi_i + \beta_i) \quad (9)$$

$$Y_{i+1} = R_i \sin(\theta_i + \alpha_i)\sin(\phi_i + \beta_i) \quad (10)$$

$$Z_{i+1} = R_i \cos(\phi_i + \beta_i) \quad (11)$$

Equations (9), (10) and (11) can be rewritten by

$$\begin{aligned} X_{i+1} &= R_i(\cos\theta_i \cos\alpha_i - \sin\theta_i \sin\alpha_i)(\sin\phi_i \cos\beta_i + \cos\phi_i \sin\beta_i) \\ &= R_i \cos\theta_i \sin\phi_i \cos\alpha_i \cos\beta_i + R_i \cos\theta_i \cos\phi_i \cos\alpha_i \sin\beta_i \\ &\quad - R_i \sin\theta_i \sin\phi_i \sin\alpha_i \cos\beta_i - R_i \sin\theta_i \cos\phi_i \sin\alpha_i \sin\beta_i \\ &= X_i \cos\alpha_i \cos\beta_i + U_i \cos\alpha_i \sin\beta_i - Y_i \sin\alpha_i \cos\beta_i - V_i \sin\alpha_i \sin\beta_i \end{aligned} \quad (12)$$

$$Y_{i+1} = Y_i \cos\alpha_i \cos\beta_i + V_i \cos\alpha_i \sin\beta_i + X_i \sin\alpha_i \cos\beta_i + U_i \sin\alpha_i \sin\beta_i \quad (13)$$

$$Z_{i+1} = Z_i \cos\beta_i - W_i \sin\beta_i \quad (14)$$

where U_i, V_i and W_i are defined by

$$U_i = R_i \cos\theta_i \cos\phi_i \quad (15)$$

$$V_i = R_i \sin\theta_i \cos\phi_i \quad (16)$$

$$W_i = R_i \sin\phi_i \quad (17)$$

It is noted that U_{i+1}, V_{i+1} and W_{i+1} can be written by

$$U_{i+1} = U_i \cos\alpha_i \cos\beta_i - X_i \cos\alpha_i \sin\beta_i - V_i \sin\alpha_i \cos\beta_i + Y_i \sin\alpha_i \sin\beta_i \quad (18)$$

$$V_{i+1} = V_i \cos\alpha_i \cos\beta_i - Y_i \cos\alpha_i \sin\beta_i + U_i \sin\alpha_i \cos\beta_i - X_i \sin\alpha_i \sin\beta_i \quad (19)$$

$$W_{i+1} = W_i \cos\beta_i + Z_i \sin\beta_i \quad (20)$$

Based on equations (6), (7) and (8), the computations of equations (12), (13), (14), (18), (19) and (20) can be obtained by using the following set of CORDIC rotations:

$$U_{i+1} = \frac{1}{k_i^2}(U_i - X_i \rho_i 2^{-i} - V_i \delta_i 2^{-i} + Y_i \delta_i \rho_i 2^{-2i}) \quad (21)$$

$$V_{i+1} = \frac{1}{k_i^2}(V_i - Y_i \rho_i 2^{-i} + U_i \delta_i 2^{-i} - X_i \delta_i \rho_i 2^{-2i}) \quad (22)$$

$$W_{i+1} = \frac{1}{k_i}(W_i + Z_i \rho_i 2^{-i}) \quad (23)$$

$$X_{i+1} = \frac{1}{k_i^2}(X_i + U_i \rho_i 2^{-i} - Y_i \delta_i 2^{-i} - V_i \delta_i \rho_i 2^{-2i}) \quad (24)$$

$$Y_{i+1} = \frac{1}{k_i^2}(Y_i + V_i \rho_i 2^{-i} + X_i \delta_i 2^{-i} + U_i \delta_i \rho_i 2^{-2i}) \quad (25)$$

$$Z_{i+1} = \frac{1}{k_i}(Z_i - W_i \rho_i 2^{-i}) \quad (26)$$

where

$$\cos\alpha_i = \frac{1}{\sqrt{1 + 2^{-2i}}} \quad (27)$$

$$\sin\alpha_i = \frac{\delta_i 2^{-i}}{\sqrt{1 + 2^{-2i}}} \quad (28)$$

$$\cos\beta_i = \frac{1}{\sqrt{1 + 2^{-2i}}} \quad (29)$$

$$\sin\beta_i = \frac{\rho_i 2^{-i}}{\sqrt{1 + 2^{-2i}}} \quad (30)$$

$$k_i = \sqrt{1 + 2^{-2i}} \quad (31)$$

In addition, $\alpha_i = \delta_i \tan^{-1} 2^{-i}$, $\beta_i = \rho_i \tan^{-1} 2^{-i}$, and δ_i and $\rho_i \in \{-1, 1\}$.

Equations (21) and (22) can be expressed in the matrix form, which is given by

$$\begin{bmatrix} U_{i+1} \\ V_{i+1} \end{bmatrix} = \frac{1}{k_i^2}\left\{ \begin{bmatrix} 1 & -\delta_i 2^{-i} \\ \delta_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} U_i \\ V_i \end{bmatrix} - \rho_i 2^{-i} \cdot \begin{bmatrix} 1 & -\delta_i 2^{-i} \\ \delta_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \end{bmatrix} \right\} \quad (32)$$

Similarly, equations (24) and (25) can be written by

$$\begin{bmatrix} X_{i+1} \\ Y_{i+1} \end{bmatrix} = \frac{1}{k_i^2}\left\{ \begin{bmatrix} 1 & -\delta_i 2^{-i} \\ \delta_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \end{bmatrix} + \rho_i 2^{-i} \cdot \begin{bmatrix} 1 & -\delta_i 2^{-i} \\ \delta_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} U_i \\ V_i \end{bmatrix} \right\} \quad (33)$$

Thus, in the proposed 3-D CORDIC algorithm, four 2-D CORDIC rotations are to be performed for the rotation of a vector in the 3-D space. The scale factors of Z_{i+1} and W_{i+1} are different from that of $U_{i+1}, V_{i+1}, X_{i+1}$ and Y_{i+1}. They can be compensated via the pre-scale of inputs or the post-scale of outputs with their respective constants K and K^2 given by

$$K = \prod_{i=0}^{n-1} k_i \quad (34)$$

$$K^2 = \prod_{i=0}^{n-1} k_i^2 \quad (35)$$

4: VLSI ARCHITECTURE AND IMPLEMENTATION

In the computer graphic system, vector interpolation can be performed on the basis of spherical interpolation, linear interpolation and CORDIC interpolation [1]. In comparison with the algorithms based on either spherical or linear interpolation, the proposed CORDIC based algorithm is more efficient and flexible in terms of computation complexity and hardware implementation. Figure 2 shows the proposed

architecture using redundant CORDIC arithmetic for 3-D geometry rotation. In which, the generation of (U_{i+1},V_{i+1}) and (X_{i+1},Y_{i+1}) is obtained by using two 2-D CORDIC processors, two hardwire shifters and two adders/sub-tractors; the generators W_{i+1} and Z_{i+1} consist of half 2-D CORDIC processor.

Figure 3 shows the auxiliary generator for the initial coordinates (U_0 , V_0 , W_0). In order to improve the computation speed and save hardware, the doubly pipelined CORDIC array [11] is adopted. Together with Figure 2, the proposed architecture is therefore composed of the auxiliary generator (U_0 , V_0 , W_0), the redundant CORDIC arithmetic (for the computation of 3-D geometry rotation), and dual-memory banks (for storing coordinates (X_i,Y_i,Z_i) and (U_i,V_i,W_i)).The proposed system diagram for 3-D geometry rotation is shown in Figure 4. The hardware code is written in Verilog®-hardware description Language (HDL) [12]. The control unit is designed by the finite state machine (FSM), which is shown in Figure 5. The hardware code of FSM is given in Appendix.

The chip of the proposed 3-D geometry rotation is synthesized by TSMC 0.18 μm 1P6M CMOS cell libraries [13]-[14]. The gate count is reported by the Synopsys® design analyzer [13]. The power consumption is reported by PrimPower®[13]. The layout view of the 32-bit 3-D geometry rotation is shown in Figure 6. The core size is $5320\mu m \times 5320\mu m$, and the power dissipation is 49.35mW with clock rate of 20 MHz at 1.8V. The critical path is 14.27 ns . All control signals are generated internally on-chip. This chip is of high throughput with low gate counts and parallel-pipelined structure.

5: EXPERIMENTAL RESULTS

The platform for architecture development and verification has been designed and implemented. The architecture has been implemented on the Xilinx XC2V6000 FPGA emulation board [15], and the chip has been integrated with the 8051 microcontroller. The 8051 microcontroller reads the original image from PC via USB 2.0 bus, and writes the rotated image back to PC. The 3-D geometry rotation function is performed on the Xilinx XC2V6000 FPGA chip for graphic rendering. Figure 7 shows the architecture development and verification board.

6: ADVANTAGES OF THE PROPOSED ARCHITECTURE ANDALGORITHM

In the Euler angle method, a sequence of three rotations is performed, each of which performs the rotation with respective to their respective axes [2], [4]. In [2], the 3-D rotation is implemented by cascading two 2-D CORDIC processors. Lang and Antelo developed a method to replace the two 2-D CORDIC processors by one 3-D CORDIC processor [3]. In which, the sequence of rotations is thus composed of one 2-D CORDIC rotation followed by one 3-D CORDIC rotation. In comparison with both of the aforementioned methods requiring more than two 2-D CORDIC computations, the proposed CORDIC based 3-D rotation algorithm requires only one 2-D CORDIC computation in parallel. The auxiliary generator for coordinates (U_0 , V_0 , W_0) and the redundant arithmetic CORDIC for 3-D rotation can be implemented in parallel. Table 1 shows the comparison of this work with that of Eberly [7], Euh, Chittamuru and Burson [2] and Lang and Antelo [3].

7: CONCLUSION

A high-throughput CORDIC-based architecture for 3-D geometry rotation in the computer graphic system is presented. In which, only one CORDIC computation time is needed. The proposed architecture by the use of CORDIC processor is simple, regular and thus suitable for VLSI implementation. In the power-aware 3-D graphic system, the performances of vector interpolation and geometry rotation can be improved significantly by using the proposed algorithm and architecture. As the latency, throughput and area are important issues in hardware implementation, the proposed algorithm is suitable for the development of high-speed applications from that point of view. The proposed 3-D geometry rotation is a reusable IP, which provides various performance, area and power consumption trade-offs in combination with the efficient hardware resources available in the target systems.

REFERENCES

1. Phong, B., "Illumination for Computer Generated Pictures," *Communications of the ACM,* (1975), 311-317.
2. Euh, J., Chittamuru, J., Burson, W., "Power-Aware 3D Computer Graphics Rendering," *Journal of VLSI Signal Processing Systems*, vol. 39, no. 1-2, (2005), 15-33.
3. Lang, T., Antelo, E., "High-Throughput CORDIC-Based Geometry Operations for 3D Computer Graphics," *IEEE Transactions on Computers*, vol. 54. no. 3, (2005), 347-361.
4. Arrigo, J. F., Chau, P. M., "Power Attitude Computation During Rapid Rotation Motion," *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 1, (2006), 63-69.
5. Jiang, X. G., Zhou, J. Y., Shi, J. H., Chen, H. H., " FPGA implementation of image rotation using modified compensated CORDIC," *6th International Conference on ASIC* (*ASICON-2005*), (2005), 699-702.
6. Sung, T. Y., Yu, C. W., Shieh, Y. S., "An Efficient CORDIC-Based Vector Interpolator in Power-Aware 3-D Graphics Rendering," *9th International Conference on Computer Science and Informatics (CSI-2006)*, Kaohsiung, Taiwan, R.O.C., (2006).

7. Eberly, D. H., 3-D Game Engine Design-A Practical Approach to Real-Time Computer Graphics, Morgan Kaufmann Pub., (2001).
8. Volder, J. E., "The CORDIC Trigonometric Computing Technique," *IRE Transactions on Electronic Computers*, vol. EC-8, (1959), 330-334.
9. Walther, J. S., "A Unified Algorithm for Elementary Functions," *Spring Joint Computer Conference Proceedings*, vol.38, (1971), 379-385.
10. Mencer, O., Semeria, L., Morf, M., Delosme, J., "Application of Reconfigurable CORDIC Architecture," *The Journal of VLSI Signal Processing*, *Special Issue on Reconfigurable Computing*, (2000), 211-221.
11. Sung, T. Y., Hu, Y. H., Yu, H. J., "Doubly Pipelined CORDIC Array for Digital Signal Processing," *Int'l Conf. on Acoustic, Speech and Signal Processing*, Tokyo, Japan, (1986), 1169-1172.
12. Thomas, D. E., Moorby, P. H., The Verilog Hardware Description Language, Fifth Edition, Kluwer Academic Pub., (2002).
13. Synopsys, http://www. synopsys. com/products.
14. "TSMC 0.18 μm CMOS Design Libraries and Technical Data, v.3.2," Taiwan Semiconductor Manufacturing Company (TSMC), Hsinchu, Taiwan, R.O.C. and National Chip Implementation Center (CIC), National Science Council (NSC), Hsinchu, Taiwan, R.O.C., (2006).
15. Xilinx FPGA products, http://www. xilinx.com/products.

Table 1. Comparisons of this work with that of Eberly, Euh, Lang and Antelo

3-D Graphics System	Eberly [7]	Euh [2]	Lang [3]	This work
Coordinate system	Cartesian	Polar	Cartesian	Polar
No. of 2-D CORDIC processor	1	2	3	5
No. of memory bank	1	1	1	2
CORDIC computation(s)	3	2	2	1
Auxiliary coordinate generator	No.	No.	No.	Yes

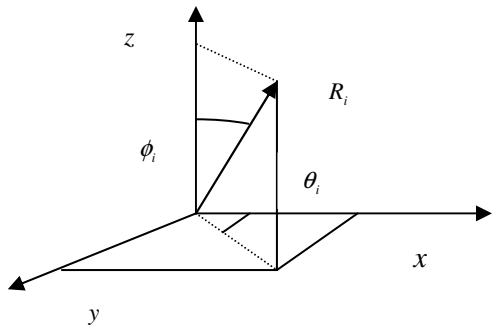

Figure1. The vector ***R*** to be rotated in the 3-D space.

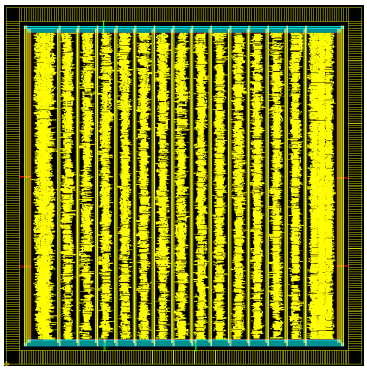

Figure 6. The layout view of the implemented 32-bit 3-D geometry rotation.

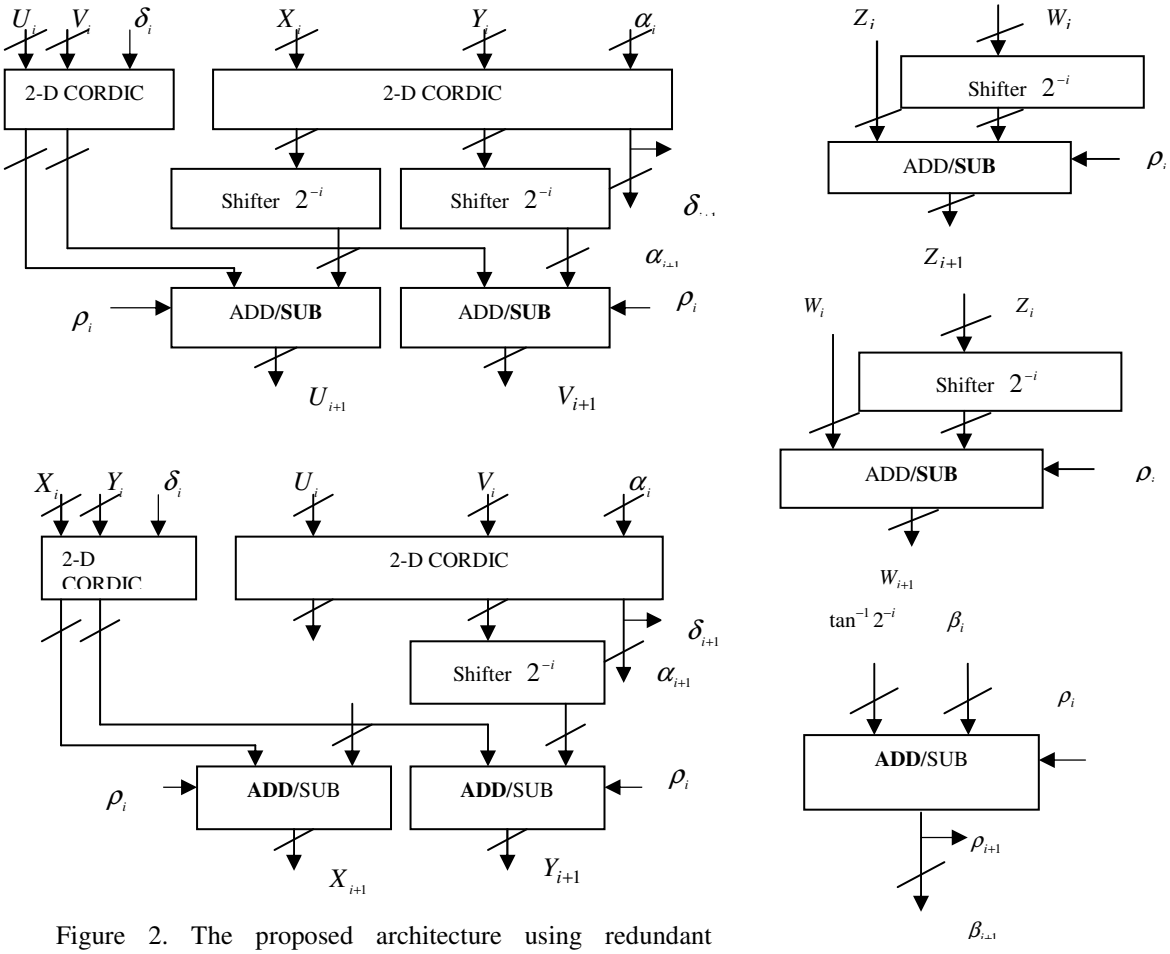

Figure 2. The proposed architecture using redundant CORDIC arithmetic for 3-D rotation.

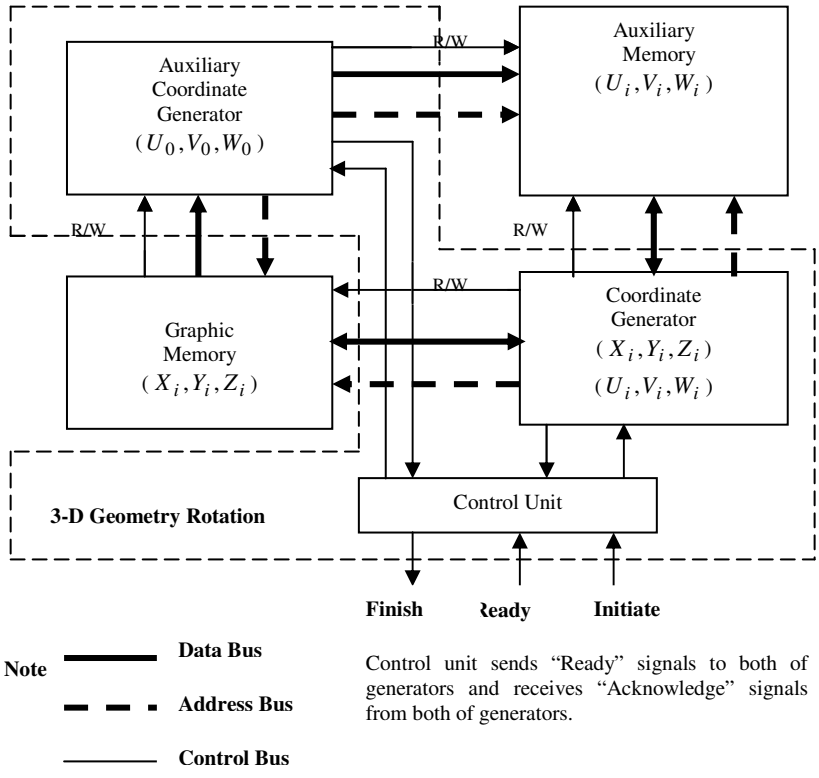

Figure 4. The proposed system diagram for 3-D geometry rotation.

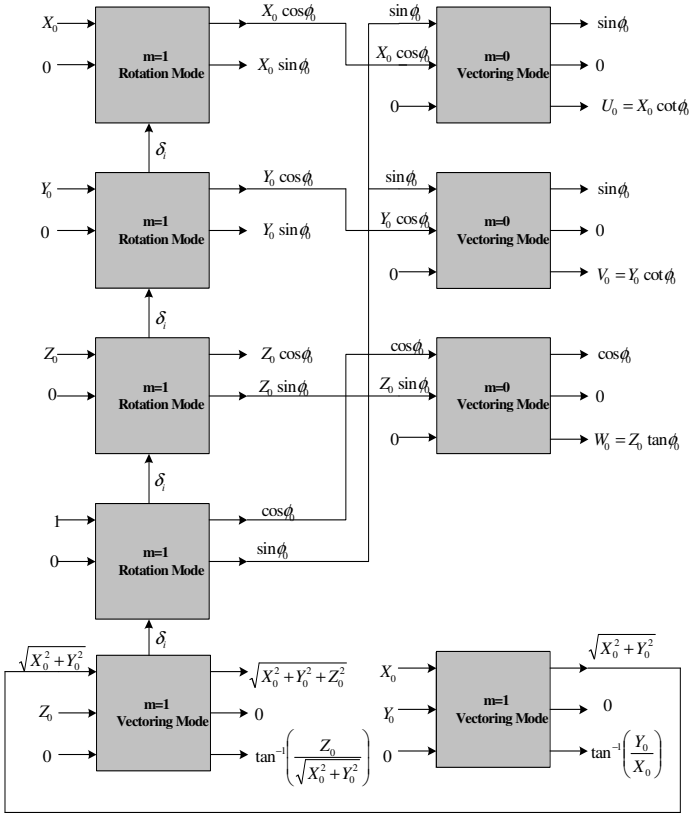

Figure 3. The auxiliary generator for coordinate (U_0, V_0, W_0).

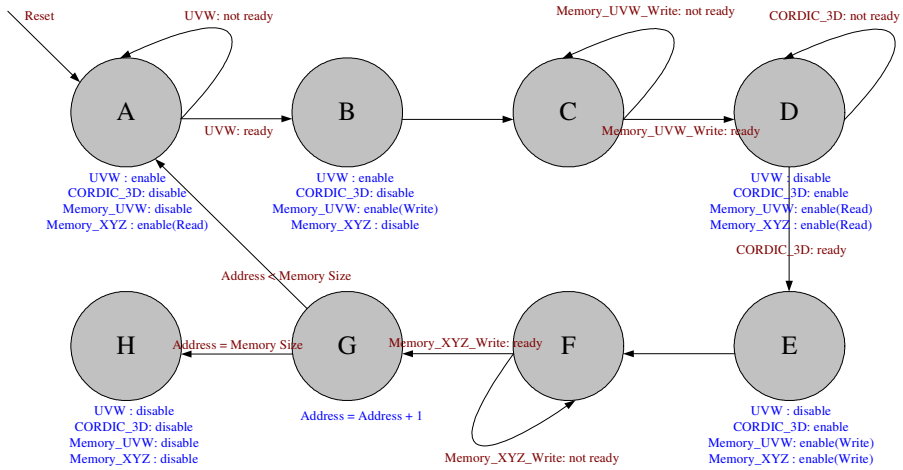

Figure 5. The state diagram of the FSM of control unit for 3-D geometry rotation.

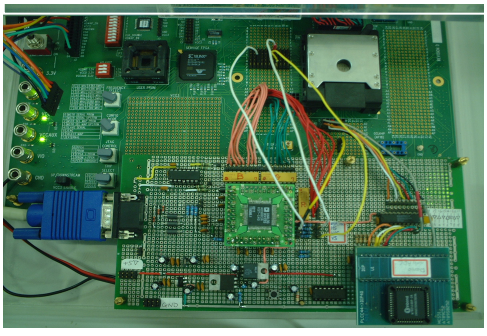

Figure 7. The architecture development and verification board.