

Pipeline Architecture for Clipping Straight Line Segments

Fakhrulddin H. Ali

Dena R. Ibrahim

College Of Engineering - University of Mosul-IRAQ
Email : fhali310@yahoo.com

Abstract

Sometimes, in computer graphics, there is a need to show a part or a piece of a specific picture to study it accurately. For this purpose a window can be used to select the interested part of the picture for display out of other parts. This window is called a clipping window and this process is called a clipping process .On the other hand, clipping is required when a part of a moving image becomes out of the visible zone for display. So the clipping process is one of the important issues in the field of computer graphic specially for real time applications. Therefore a study of the line clipping with respect to a rectangular window while looking to various methods to achieve clipping process in the image space is presented in this paper. In addition to that, performance measurements are conducted and compared. The paper develops a method to perform clipping in the object space in which the line segment is clipped against a viewing volume. Finally, pipelining is adopted to realize clipping algorithm in hardware.

Key Words : Regional Code, Clipping Window , Viewing Pyramid, Pipeline.

معمارية خط أنابيب لتقليم مقاطع خطوط مستقيمة

دينا رياض إبراهيم

فخر الدين حامد علي

كلية الهندسة - جامعة الموصل - العراق

الخلاصة

في تطبيقات الرسوم المولدة بالحاسوب هناك حاجة لعرض جزء من الصورة الكلية حيث يستخدم شبك القطع لاختيار الجزء المطلوب وإعادة عرضه على الشاشة. يسمى هذا الشبك بشبك التقليم وتسمى العملية بعملية التقليم. من ناحية أخرى التقليم أيضا ضروري ومطلوب عندما يخرج جزء من صورة متحركة خارج نطاق الرؤيا والعرض لذلك التقليم مهم في عملية توليد الصور المتحركة خصوصا عند الأداء في الزمن الحقيقي. بناء على ما تقدم في هذا البحث دراسة لكيفية تقليم قطعة الخط المستقيم بالنسبة لشبك مستطيل مع مقارنة الطرق المختلفة لانجاز ذلك في فضاء الصورة بالإضافة إلى تعزيز ذلك بقياس الأداء ومقارنته. كما تم تطوير عملية التقليم لأدائها في فضاء الشيء حيث يتم تقليم المستقيم بالنسبة إلى حجم الرؤيا. في النهاية تم تبني تقنية خط الأنابيب لتنفيذ التقليم في الكيان المادي .

Received: 14 – 8 - 2011

Accepted: 25 – 1 - 2011

1- Introduction

Computer graphic is one of the technical fields that is more public and more developing year after year. This is due to the fast development in computer itself and the increase its ability bearing in hand that the effective interaction between the users and computers depends on what they see on the monitors [1]. The picture is one of the fast and effective means in the field of communication where the computer graphic is a growth up field because it find a wide application area like education , science , medical , advertisement, simulation and training [2]. In computer graphics a picture could contains many objects that may concealment one another. To understand the complex spatial relationship that exists, users look for graphical tools that can efficiently manipulate the content of the views[3]. One of the important operations in computer graphic is modeling which is used to represent objects by a number of flat faces or polygons and the degree of complexity of an object depends on the number of faces used to represent it . Each face or polygon is represented by a number of line segments, as borders, so by increasing the faces representing the object the number of lines are increased too. However, clipping of each polygonal flat or planar face is accomplished by clipping each of its border straight line segments. This is why the clipping of the whole polygonal object is based on the clipping of a straight line segment which is the subject of this paper. There are several well-known clipping algorithms each having its strengths and weaknesses , the oldest one from (1974) is called the Sutherland Hodgman algorithm as presented by Newman and Spruoll , in its basic form it is relatively simple and very efficient in tow important cases one being when the line is completely inside the clipping boundaries and the other when it is completely outside [4]. The Weiler and Atherton (1977) algorithm is more complicated , it allows clipping of a subject by an arbitrarily shape of clip polygon so it is generally applicable only in 2D (two dimension) [5]. In (1978) an algorithm was independently adopted by Cyrus and Beck that did not develop the trivial reject cases that are so important for efficiency [6]. The Liang – Barsky algorithm in (1983) was good and more complicated but in certain cases fewer intersections need to be calculated than for Sutherland-Hodgman therefore it may be somewhat faster when many polygon lines intersect with clipping boundaries [7].In (1987) Sobkow , Pospilsil and Yang developed an algorithm which is used in the sense that they enumerate all possible relationship between a line and the clipping region and only compute intersection points where these are required as a part of the output [8].In (1989) Scholl and Voisard defined general polygons [9] and in (1992) they extended this general types for points and lines [10]. In (1990) Nicholl , Lee and Nicholl presented a very efficient two dimensional line clipping algorithm that is based on a case by case approach and which improved upon the efficiency of existing line clipping algorithm , however this algorithm is only applicable in two dimension [11]. According to the outcome of (AND) and (OR) of the codes of the two end points or vertices of the line segment it is determined to be inside of the window or not [12]. In (2007)Leonardo Guerreiro and Ralf Hartmut present a method for reconstructing of complex polygon after clipping [13].

However, this article summaries the subject and introduced the current work. The rest of this paper is organized as follows: article number (2) introduces the graphics pipeline, article number (3) presents the two dimensional (2D) clipping. In article (4) the three dimensional (3D) clipping is presented. The hardware implementation is in article(5). The test and results are in article (6). The conclusions are stated in article (7).

2- The Graphics Pipeline [14]

There is four major tasks in graphical pipeline as shown in figure (1) every one of these tasks may consists of subtasks where modeling represents creation of the graphical data base. Geometric processing includes transformation and clipping as shown in figure (2) for 2D. In addition to that, hidden surface removal and projection are included for 3D as illustrated in figure(3). Rasterization generates the pixels which are stored in the frame buffer memory for display on the screen.

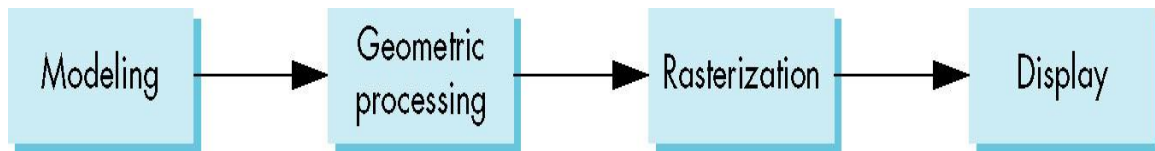


Figure (1) The graphic pipeline

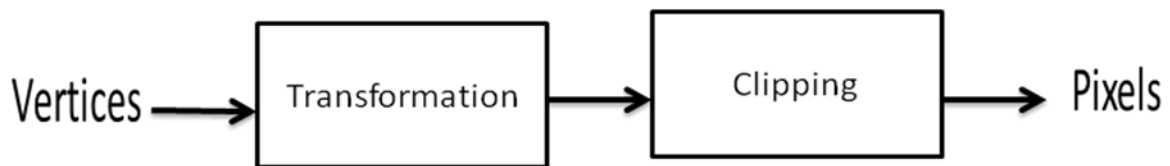


Figure (2) Geometric processing units in 2D

3- 2D Clipping

The two dimensional clipping is used when the image data base is two dimensional or when the application is three dimensional but after projection on the projection plane and transformation from the object space to the image space. Many methods can be followed to accomplish clipping as summarized in the following sub-articles.

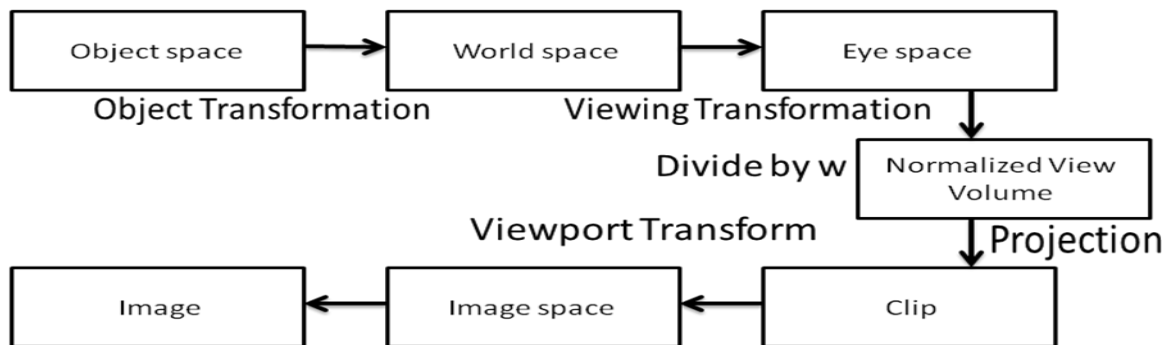


Figure (3) Geometric processing units in 3D

3-1 Direct method

This method is a modern way of Cohen Sutherland algorithm where the similarity between them is on making the initial tests that decrease the number of intersection points that must be calculated and the coding process which includes finding the position of each end point of

the line segment according to the boundaries of the clipping window by four bits code binary number called region code as shown in figure (4).

By recognizing this code we can determine quickly which line is completely inside the clipping window and which is completely outside. Any line contains '1' in the same bit position of both endpoints codes then it is completely outside the clipping window and it should be eliminated. The way of testing the line according to the boundaries of the clipping window is by making logical

AND process on the code of two endpoints or vertices of the line segment so if the result is not zero it is completely outside the clipping window and it will be canceled. If the previous tests are failed then the intersection point with the boundaries of the clipping window should be calculated [12]. This is done by a direct calculation of the intersection point, where each boundary of the clipping window is defined as a line using equation (1).

1001	1000	1010
0001	0000	0010
0101	0100	0110

Figure (4) : Region code

$$\frac{Y_2 - Y_1}{X_2 - X_1} = \frac{Y - Y_1}{X - X_1} \quad (1)$$

Where (X_1, Y_1) , (X_2, Y_2) represent the vertices of each line. The coordinate value of the first intersection point is defined by one of the following:

Left: $X_c = X_{min}$, Right: $X_c = X_{max}$, Up: $Y_c = Y_{min}$, Down: $Y_c = Y_{max}$

To find the second coordinate value one of the equations (2-5) is used [15].

$$Y_c = \frac{(Y_2 - Y_1)(X_{min} - X_1)}{(X_2 - X_1)} + Y_1 \quad (2)$$

$$Y_c = \frac{(Y_2 - Y_1)(X_{max} - X_1)}{(X_2 - X_1)} + Y_1 \quad (3)$$

$$X_c = \frac{(X_2 - X_1)(Y_{max} - Y_1)}{(Y_2 - Y_1)} + X_1 \quad (4)$$

$$X_c = \frac{(X_2 - X_1)(Y_{min} - Y_1)}{(Y_2 - Y_1)} + X_1 \quad (5)$$

3-2 Midpoint division method

This method is designed to achieve the end points of the clipped line and also reject the lines that are completely outside which makes it a very good method to implement the clipping process on the picture that is larger than the screen size. This algorithm is of two parts, the first one determines if the line lies inside the window boundaries to save it, otherwise the line must be tested if it is outside the clipping boundaries to cancel it. If these two tests are not productive the line must be divided in its midpoint and the previous tests are repeated for each half separately and so on until achieving the intersection point [16].

3-3 Trigonometric method

This algorithm depends on the slope of the line that must be clipped, so if the value of slope is zero or infinity then the cosine law will be used while if the slope has a value between zero and infinity then the triangular ratio can be used. To find the slope of the line there will be no direct equation of the slope to avoid dividing by zero that makes the slope value unknown [17].

4- 3D Clipping

All computer graphic images are two dimensional where the images that a computer can produce via a monitor, a plotter or printer are displayed or recorded on two dimensional screen or sheet of paper. What is meant by the term three dimensional is that attempts are made to create the illusion of the third dimension, the depth. In computer graphics where images are being created synthetically, variety of techniques have had to be specially developed in order to produce the illusion of realistic three dimensional objects and this makes it much more complicated process like, perspective projection, shadows and shading effect [18]. For this purpose a volume is used for restricting the vision range instead of the window to account for the third dimension (z). Additional clipping is done for the third dimension where objects that lie behind the observer cannot be seen so the clipping process becomes very important and for this reason the clipping is called three dimensional clipping [19]. The projection process to produce a perspective image has two undesirable effects: objects behind the view point may appear on the screen, and objects may exceed the prescribed limits of the view port. These effects can be eliminated by testing each point, in eye coordinates, against a viewing pyramid, which defines the portion of eye coordinate space which the viewer can actually see. Two conditions must be met for an individual point to be visible within the pyramid:

$$-Z_e \leq X_e \leq Z_e \quad (6)$$

$$-Z_e \leq Y_e \leq Z_e \quad (7)$$

Note that these conditions exclude points behind the viewing reference point which should not be displayed.

The front vertices will be transformed from eye coordinate to screen coordinate, this means applying projection process to determine the location of each point on the projection plane or the screen. Line cannot be processed as easily as points and must be clipped against the limits of the viewing pyramid. An external line segment may be rejected as invisible if no part of it intersects the pyramid, otherwise the endpoints of the visible portion part of the line are calculated in the three dimensional coordinate system then transform of the end points of the line, after clipping, into the screen coordinate is performed to display it. The clipping process must operate on three dimensional descriptions. There are several assumptions considered to decrease the complexity and make the clipping process easier [19]. One of them is that the

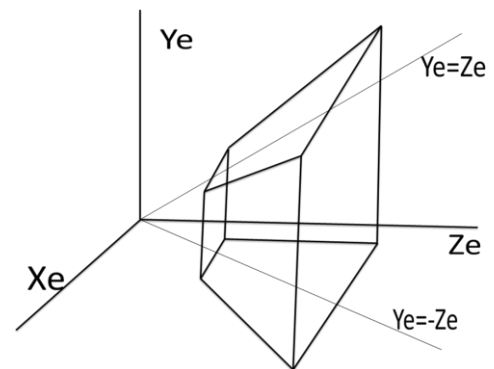


Figure (5): Viewing pyramid

center of the projection is set at the origin of the eye coordinates and the viewing direction is the positive z-axis. The second is that the pyramid is considered having (45) degree angle between the z axis with each of its four sides planes. For the convenience of the clipping task there is a redefine of a new coordinate system:

$$\begin{matrix} X_c = X_e & (8) \\ Y_c = Y_e & (9) \end{matrix}$$

$$Z_c = Z_e \quad (10)$$

To test the coordinates of clipping points equations (11) and (12) are applied.

$$-Z_c \leq X_c \leq Z_c \quad (11)$$

$$-Z_c \leq Y_c \leq Z_c \quad (12)$$

A three dimensional clipping algorithm can be derived by extending the two dimensional scheme. The algorithm determines whether an end point of the line lies outside the limits by computing a four bit code for each endpoint:

First bit : X_c is to the left of pyramid, that is, $X_c < -Z_c$

Second bit : X_c is to the right of pyramid, that is, $X_c > Z_c$

Third bit : Y_c is below the pyramid, that is, $Y_c < -Z_c$

Fourth bit : Y_c is above the pyramid, that is, $Y_c > Z_c$

If the codes for both endpoints are zero then both endpoints lie within the pyramid and the line is trivially accepted for display. If the logical AND of the codes is not zero, both endpoints lie on the invisible side of one of the planes and the entire line is trivially rejected. Otherwise the line segment crosses one or more of the pyramid planes. The algorithm computes the point of intersection of the line and uses this point to replace the end point that was on the invisible side of the plane [20].

5- System with multiprocessing units

Systems with multiprocessing units are those that emphasize parallel processing and can be divided in to two architecture configurations [21]:

- Pipeline processing units: performs overlapped computation to exploit temporal parallelism.
- Array processing units: uses multiple synchronized processing units to achieve spatial parallelism.

Due to the high computing required to perform clipping process in real time a single processing element is too slow to satisfy such requirement. This necessitates the development of a system with multiprocessing units by splitting the load required for parallel processing between the various individual processing units. One way to do this load partitioning is by splitting it according to the function required where each processing unit deals with the whole object but performs the associated function of the overall required computation on the object exploiting pipelining. So the system is designed with three stages pipe, the first one performs coding, the second stage decides the required clipping through generating the clipping control or activating signals, and the third stage performs the required clipping. Even distribution of processing unit is essential to ensure a balanced operation of such system where in this method the information is passed from on stage to another forcing them to wait for each other. Since all the stages have to complete their operation before the next time unit so the system speed is inversely proportional to the slowest processing stage. This disadvantage can be only overcome by minimizing the waiting interval and reducing the

interaction between processing units or stages to minimum [2]. Another way of partitioning is distributing the input space where the object (line to be clipped) is partitioned in to two parts (two vertices) each one assigned to a separate coding unit and forcing the two units to work in parallel. Their outputs are combined to perform the regional code of the endpoints of the line. To achieve more balanced system operation, the same procedure is also applied to partitioning the clipping operation over eight parallel units. During processing, some of these eight units are active (up to four) and the rest are un activated depending on the position of the two vertices. So the designed system exploits pipelining where after the pipe is full, it produces an output for each time unit or clock. The over all architecture of the designed pipeline system is shown in figure (6) where same system implements both 2D and 3D clipping algorithms using FPGA kit (Spartan 3E- XCS500E) with Very high speed integrated circuit Hardware Description Language (VHDL) as a program language [22][23].

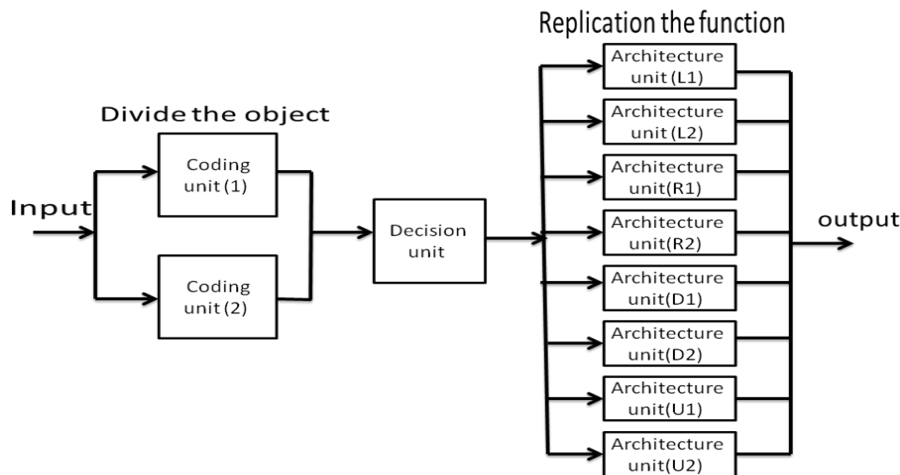


Figure (6) : The designed architecture of line clipping hardware

The two input vertices of the line segment are first positional coded by the coding units of (V1&V2) . The line is trivially accepted or rejected or not by the decision unit. The next section in the pipeline consists of many parallel units for intersecting each vertex with each side of the clipping zone. Although this architecture is for both 2D clipping and 3D clipping, there is a difference between the two concerning the calculation which is in the image space (x and y only for 2D) or in the object space (x , y , and z for 3D).

6- Tests and results

This article explains the test results of implementing the algorithms software using a laptop with (2.2 GHz) frequency, where each algorithm is tested to clip (9) line segments with different location and orientation. The time measurement is conducted for 10000 number of repetition, the measured results of executing each method by the software are shown in table (1).

Table (1) : Software execution time

Algorithm	Total time (sec)	Line per sec
Direct method	0.22	$(10000*9)/0.22 = 409090$
Midpoint Division	0.275	$(10000*9)/0.275 = 327272$
Trigonometric	1.98	$(10000*9)/1.98 = 45454$
3D	8.35	$(10000*9)/8.35 = 10778$

The direct method, being the fastest 2D method, is adopted to design the 2D clipping hardware system. The same method is also developed to design the 3D clipping architecture. The execution time, maximum frequency and the number of lines per second for each system are stated in table number (2).

Table (2) : Processor execution time

System	Min. period	No. of Lines per sec.
2D	5.291 ns	189000100
3D	5.685 ns	175901400

The utilization summary of the device for each processor are presented in tables (3) and (4).

Table (3) : Device utilization summary for the 2D system

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	63	9,312	1%	
Number of 4 input LUTs	1,650	9,312	17%	
Logic Distribution				
Number of occupied Slices	887	4,656	19%	
Number of Slices containing only related logic	887	887	100%	
Number of Slices containing unrelated logic	0	887	0%	
Total Number of 4 input LUTs	1,702	9,312	18%	
Number used as logic	1,650			
Number used as a route-thru	52			
Number of bonded IOBs	13	232	5%	
Number of GCLKs	1	24	4%	
Number of MULT18x18SIOs	7	20	35%	
Total equivalent gate count for design	16,176			
Additional JTAG gate count for IOBs	624			

Table (4) : Device utilization summary for 3D system

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	79	9,312	1%	
Number of 4 input LUTs	2,617	9,312	28%	
Logic Distribution				
Number of occupied Slices	1,351	4,656	29%	
Number of Slices containing only related logic	1,351	1,351	100%	
Number of Slices containing unrelated logic	0	1,351	0%	
Total Number of 4 input LUTs	2,626	9,312	28%	
Number used as logic	2,617			
Number used as a route-thru	9			
Number of bonded IOBs	13	232	5%	
Number of GCLKs	1	24	4%	
Number of MULT18x18SIOs	8	20	40%	
Total equivalent gate count for design	24,819			
Additional JTAG gate count for IOBs	624			

6-1 2D Clipping test

The first unit used to implement this algorithm is responsible for coding. The inputs to this units are the coordinates of the clipping window (Xmin, Ymin, Xmax, Ymax) as well as the coordinate values of the vertices of the line segment that will be clipped V1(X1,Y1), V2(X2,Y2). The outputs are (Cx1 , Cx2) which are the positional codes of the line segment , and (fclip , fclip1) which are an indication that the coding process is completed as shown by an example waveforms in figures (7) and (8).

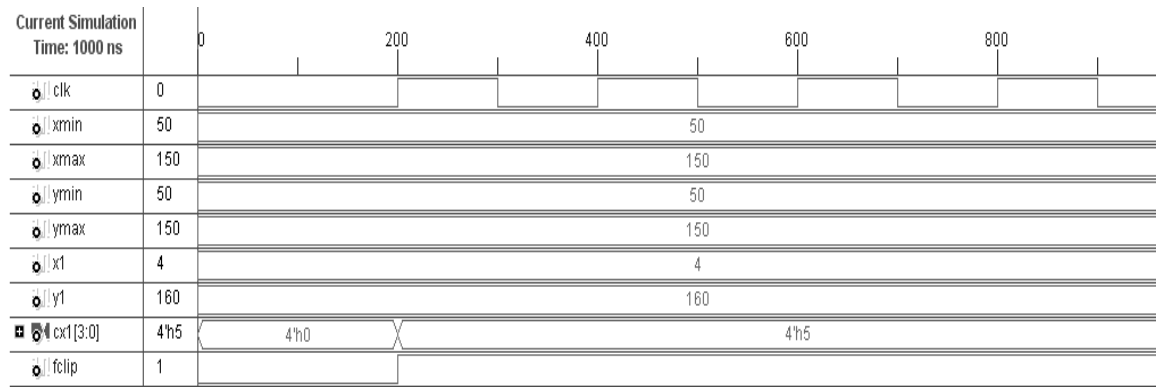


Figure (7): (V1) Coding simulation result of 2D processor

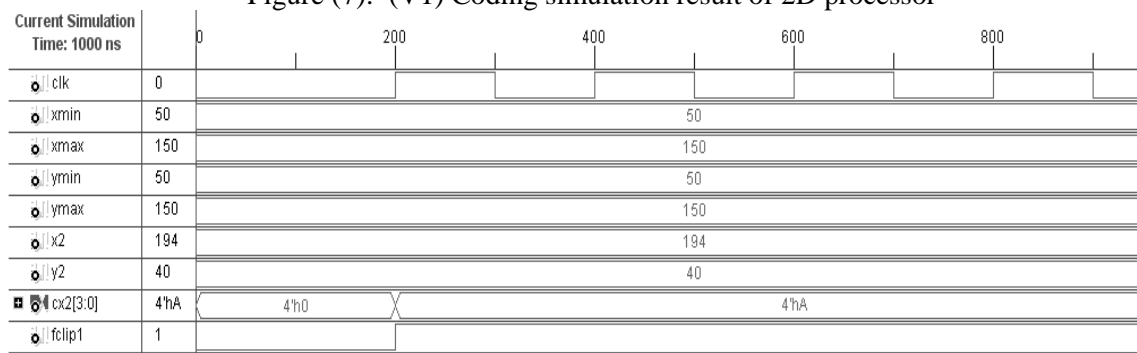


Figure (8): (V2) Coding simulation result of 2D processor

The second unit (Decision) checks if the line is completely inside the clipping window or completely outside or not so it gives an enable signal called (flag) in addition to a number of indicators or activating signals that, according to the location of each vertex of the line segment, specify which of the following unit will be activated and which will not be activated. This is illustrated by the specific example as shown in figure (9).

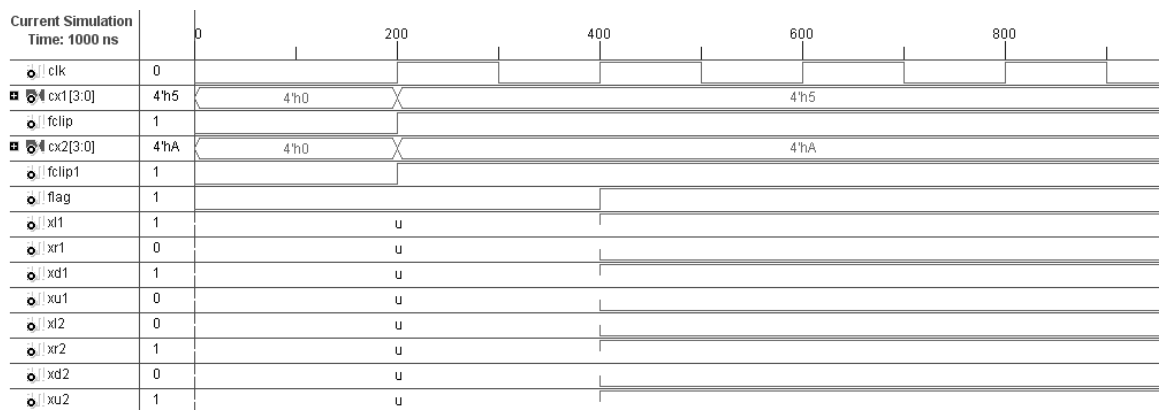


Figure (9): Decision simulation result of 2D processor

The signals (x1,xd1,xr2,xu2) when activated mean that the first vertex of the line segment is out of the window from both the left and down sides and the second vertex is out from both the right and up sides. This leads to that the unit (V1left) will compute the clipping intersection point for V1 for the example as shown in figure (10).

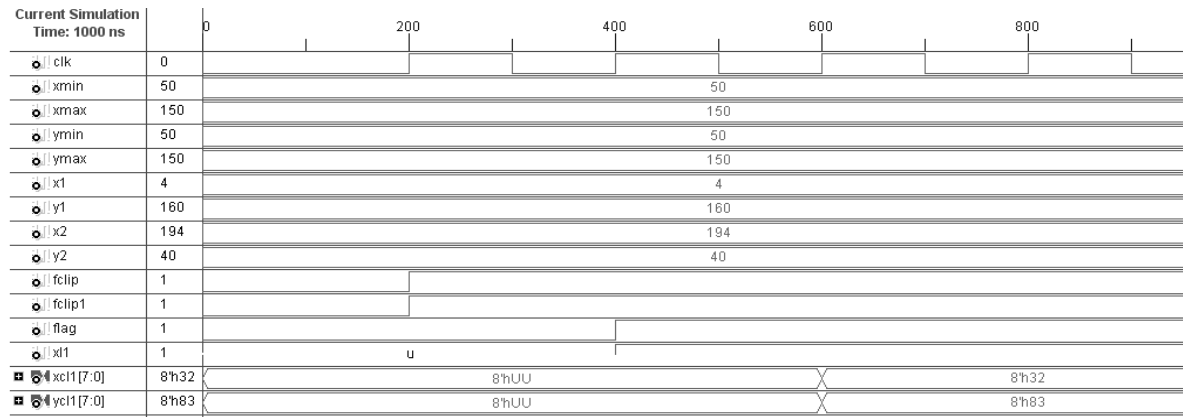


Figure (10): (V1 left) unit simulation result of 2D processor

For the other vertex the signals (xr2 for right) will be activated for clipping V2 from the right side of the clipping window in the same way used to clip V1 from the left side.

6-2 3D Clipping test

The inputs to the coding units are the coordinates values of the vertices of the line segment that will be clipped V1(X1,Y1,Z1) and V2(X2,Y2,Z2). The outputs are (Cx1,Cx2) which are the positional codes of the line segment, and (fclip, fclip1) which are an indication that the coding processes are completed as shown by an example waveforms in figures (11) and (12).

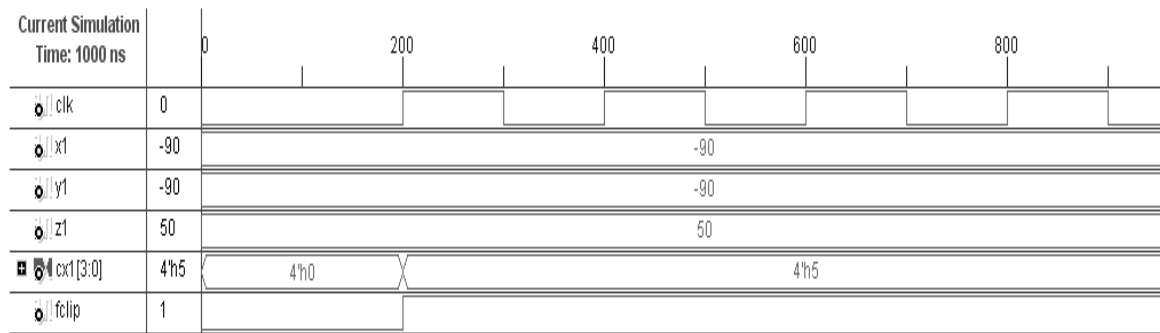


Figure (11): V1 Coding simulation result of 3D processor

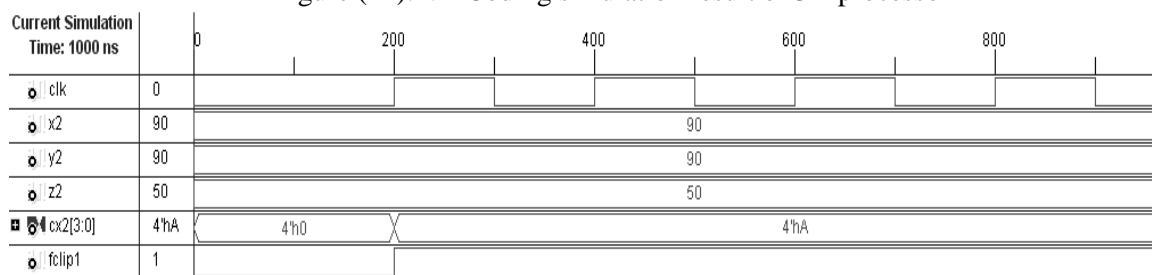


Figure (12): V2 Coding simulation result of 3D processor

The second unit (Decision) checks if the line is completely inside the view volume or completely outside or not so it gives an enable signal called (flag) in addition to a number of indicators or activation signals that, according to the location of each vertex of the line segment, specify which of the following units will be activated and which will be not as in the specific example which is shown in figure (13).

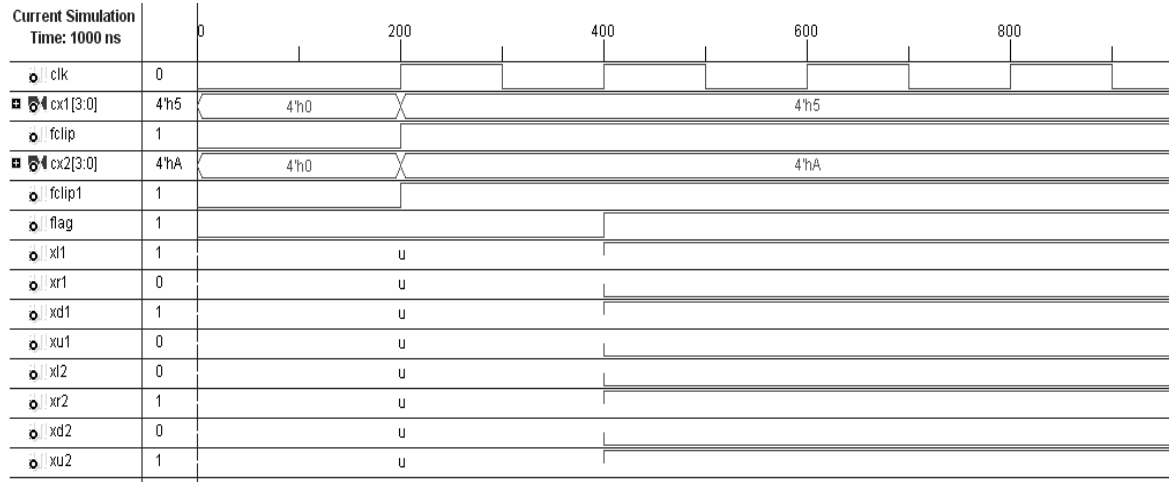


Figure (13): Decision simulation result of 3D processor

The signals or indicators which are activated are (xl1) and (xd1), this mean that the first vertex of the line segment is out from the left and down sides while the indicators (xr2) and (xu2) mean that the second vertex is out from the right and up sides. So when (fclip = 1 , fclip1 = 1 , flag = 1 , xl1 = 1 and xr2 = 1) clipping on (V1) from the left is performed and on (V2) from the right as in figure (14). Because the values of (Yc1) is not less than (-Zc1) and (Yc2) is not greater than (Zc2), no further clipping is required from down for (V1) and from up for (V2).

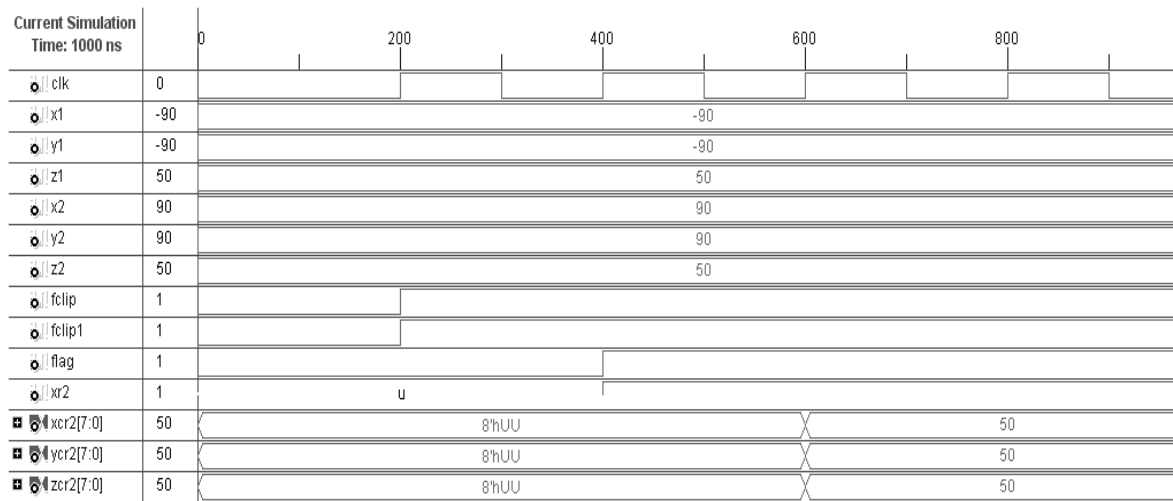


Figure (14): (V2 Right) simulation result of 3D processor

7- Conclusions

The clipping process of a straight line segment is considered essential for clipping a picture because the polygon is bounded by a number of line segments as borders and from a number of polygons (or faces) the whole picture is formed and stored in the graphical data base. In

this paper, a new parallel processing modular architecture for both 2D and 3D clipping is presented. Pipelining is adopted in the design which, when the pipe is filled, can produce an output for each unit time clock. Examining the results in table (1) the time of the direct method is less among other algorithms this mean that its speed is the highest therefore this method is implemented directly in hardware and developed to work as three dimensional clipping algorithm. The speed up factor of implementing the clipping in hardware is (426) for the 2D processor and (16320) for the 3D processor. To confirm 3D performance in real time the speed is approximated by 35M polygon per second assuming polygons or faces of an average of five sides. For a modern monitor with an assumed vertical frequency of 100 Hz or frames per second, a complexity of 350000 polygonal faces can be processed by the 3D clipping processor in real time (i.e. in one frame). In the future, a single adaptive algorithm which assigns 2D clipper after projection or 3D clipper before is to be built and implemented according to a predefined strategy as an extension of this work.

References

- 1- Guodong Lu , Xuanhui Wu , Qunsheng Peng "An Efficient Line Clipping Algorithm Based on Adaptive Line Rejection " , Zhejiang University , China 2002.
- 2- Fakhraldeen H.Ali " A Concurrent Processing System for The generation of Real-Time Three Dimension Graphic " ,Ph.D thesis, Bradford University , UK ,1989.
- 3- William E.Lorensen " Geometric Clipping Using Boolean Textures", General Electric Company , Corporate Research and Department , Schenectay New York , IEEE, 1993.
- 4-Sutherland I.E. and Hodgman G.W. "Reentrant polygon clipping " , Comm. Of the ACM Vol.17, No.1, pp.32-42,(1974).
- 5-Weiler K. and Atherton "Hidden surface removal using polygon area sorting" In: Proceeding of the 4th annual conference on computer graphics and interactive techniques, San Jose, California , pp.214-222,(1977).
- 6-Mike Cyrus and Jay Back "Generalized Two- and Three Dimensional Clipping" computer graphics Vol.3, No.1, pp.23-28(1978).
- 7- Liang, Y. and Barsky, B. "An analysis and algorithm for polygon clipping", Commun. ACM 26, 11 (Nov), 868-877,(1983).
- 8-M.S. Sobkow , P.Posipilsil and Y.-H. "A Fast Two Dimensional Line Clipping Algorithm Via Line Encoding" computer and graphics Vol.11, No.4, pp.459-467, (1987).
- 9- Scholl, M. and Voisard, A. "Thematic map modeling", In: Proceedings of the First International Symposium on Large Spatial Databases, Santa Barbara, CA, (1989).
- 10- Voisard, A. "Bases de données géographiques: du module de données à l'interface utilisateur". Ph.D. Thesis, University of Paris-Sud (Centre d'Orsay), (1992) .
- 11- Tina M.Nicholl and Robin A.Nicholl , "performance geometric transformation by program transformation" , ACM transaction on graphics , vol.9 ,pp28-40, no.1 January (1990).
- 12- Donald Hearn and M.Pauline Baker " Computer Graphic C Version " ,Prentice Hall , Inc. New Jersey ,chapters(6-10-11-12),USA (2002).

- 13- Leonardo Guerreiro and Ralf Hartmut "Polygon Clipping and Polygon Reconstruction", Rio de Janeiro, Brazil, 2007.
- 14- Frank Pfenning "computer graphic – clipping", Carnegie Mellon University, March 11, (2003) <http://www.cs.cmu.edu/~fp/courses/graphics/>
- 15- Howard Anton, Irl Bivens and Stephen Davis "Calculus" seventh edition, (chapter 12) USA (2002).
- 16- Malay K. Pakhlea "Computer Graphics, Multimedia and Animation", 2nd edition, New Delhi, June (2010).
- 17- Thomas "calculus and analytic geometry", fifth edition, personal education Inc. (2005)
- 18- Malcolm Richardson "Practical Computer Graphics 2nd edition", department of innovation studied, University of East London, Cambridge, UK, (1999).
- 19- Cornet K. Pokorny, Curtis F. Gerald "Computer Graphics the Principle Behind the Art and Science", California Polytechnic State University, United State of America, (1989).
- 20- William M. Newman, Robert F. Sproull, "Principles of Interactive Computer Graphics", McGraw-Hill Book Company, (1975).
- 21- Kai Hwang, Faye A. Briggs "computer architecture and parallel processing", McGraw-Hill Inc (1984).
- 22- www.xilinx.com "ISE Design Suite 9.2i Release note and installation Guide", (2002-2008).
- 23- IEEE standard VHDL language reference manual (IEEE Std 1076-2001), Institute of Electrical and Electronics Engineers, (2001).

The work was carried out at the college of Engineering. University of Mosul