PIXEL-PLANES: A VLSI-ORIENTED DESIGN FOR 3-D RASTER GRAPHICS

H. Fuchs and J. Poulton

University of North Carolina at Chapel Hill

ABSTRACT

We present here the design for a machine which can rapidly generate realistic continuous-tone color video images of three dimensional objects from a polygonal data base. The design consists of an array of identical custom LSI chips which form a pixel-mapped image buffer enhanced with certain processing capabilities. Visibility calculations are performed concurrently for all pixels using a depth buffer algorithm, and for a p-sided polygon, O(p) processing steps are required. The same circuitry which performs visibility calculations also carries out polygon edge definition and smooth shading, and requires only a small part of each pixel's memory area on the chip. Processing time for a single polygon is independent of its size and orientation; convex polygons with any number of vertices can be processed. The expected speed of the system to generate a smooth-shaded solid modeled image should equal that of currently-available caligraphic systems drawing stick figures; to generate images at this rate, our system must complete a one-bit addition in 250 nsec, a speed easily obtainable with current nMOS processes.

RÉSUMÉ

Ce rapport présente le plan d'un appareil qui peut rapidement produire des images vidéo-couleur à tonalités continues réalistes d'objets à trois dimensions à partir d'un base de données polygonale. Le plan comprend un ensemble de boîtiers LSI (intégration à grande échelle) identiques qui forment un étage tampon de production d'images par projection par pixel avec certaines capacités de traitement. Les calculs de visibilité sont effectués concurremment pour tous les pixels à l'aide d'un algorithme tampon de profondeur et, pour le polygone à p-côtés, les étapes de traitement O(p) sont nécessaires. Le même circuit qui effectue les calculs de visibilité effectue également la définition de bord du polygone avec un faible ombragement et ne nécessite qu'une petite partie de la zone de mémoire de chaque pixel sur le boîtier. Le temps de traitement pour un seul polygone est indépendant de sa dimension et de son orientation; les polygones convexes avec n'importe quel nombre de crêtes peuvent être traités. On estime que la vitesse du système à produire une image solide faiblement ombrée devrait être égale à la vitesse des systèmes caligraphiques qui produisent des figures en bâtonnets; afin de produire des images à cette vitesse, notre système doit effectuer une addition à un bit en 250 ns, vitesse facilement atteignable avec les traitements courants nMos.

We present here the design for a machine which can rapidly generate realistic continuous-tone color video images of three-dimensional objects from a polygonal data base. The important features of this design are:

1) The expected polygon processing time during image generation is easily as fast as current real-time line-drawing systems.

2) Processing time for a polygon is independent of size and orientation of that polygon, and increases only linearly with the number of vertices. Convex polygons with any number of vertices can be processed.

3) The design is implemented in a number of identical chips, is modular, and is easily expandable to accommodate increased screen resolution.

4) The design consists almost entirely of a rectangular grid of cells, each of which corresponds to a single pixel; a cell consists of a set of storage registers augmented with a minimal amount of processing hardware. Since most of the chip area in a full-scale implementation of the current layout would be occupied by the pixel storage registers which make up the "frame buffers" in current raster graphic systems, we hope that a commercial version of our system can be implemented in not substantially more silicon area than simple current frame buffers.

We have designed and carried to layout a prototype chip and are currently performing timing simulations on it. Fabrication is expected to be complete by May, 1981.

Since the geometric transformations, clipping and perspective computations needed to get the screen coordinates of a polygon from its original object space (data base) representation can easily be done by current real-time line drawing systems such as the Evans and Sutherland Picture System II, our design concentrates on the traditionally burdensome tasks after this -- the polygon drawing, visibility, and pixel painting calculations.

The design consists essentially of a pixel-mapped image buffer which has been enhanced with certain processing capabilities for performing visibility calculations using a depth buffer; these calculations are performed concurrently for all pixels within a p-sided polygon in O(p) steps. The main components are two special "multiplier trees", (Fig. 1) and an array of pixel cells (Fig. 2). Each multiplier tree consists of 2**N one-bit adders and one-bit registers, where 2**N is the picture resolution (N=9, typical). Each pixel cell in the array consists of buffered image and depth registers, a one-bit adder, a one-bit comparator, and enabling control circuitry. The system's fundamental operation is calculating, concurrently in each pixel, the value of a function $F(x,y)= Ax + By + C$, (for that pixel's x,y location value) as the coefficients A,B,C are input to the system. For each polygon, a sequence of these calculations is used to a) define the edges of the polygon, b) determine the depth value at each pixel, and c) determine the smooth shaded color at each visible pixel.

We note several important points concerning this scheme: Transformation units in current systems (such as the aforementioned E&S Picture System II) deal in vertex, not edge data; however, well-know formulae can be applied to convert vertex data into the linear and planar equations needed by our system. This conversion can either be accomplished by the transformation system, by a separate module, or built into a later version of our IC implementation. A second feature of the design which might at first appear undesirable is the distribution of the large number of product terms from the multipliers to the pixel cells. The multipliers, however, exhibit a tree structure; by including copies of the

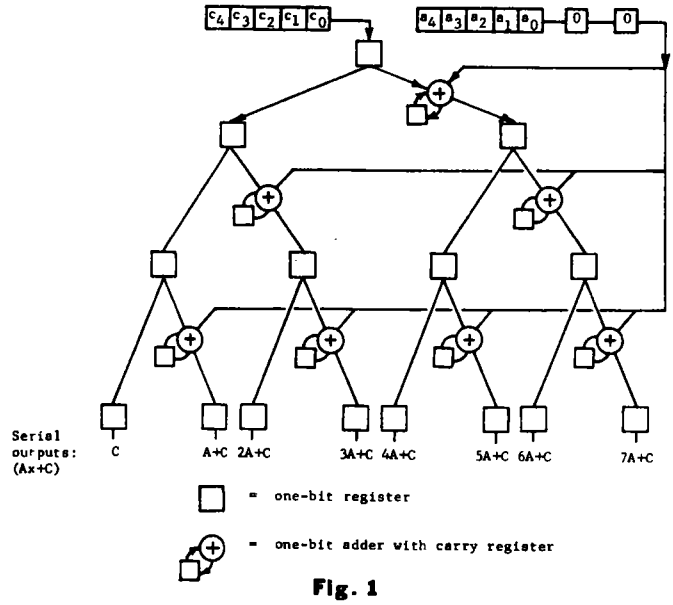

Fig. 1
Multiplier Tree

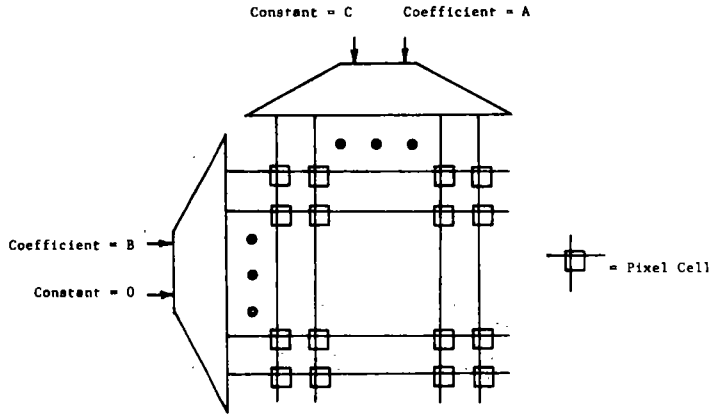

Fig. 2: Multiplier Trees with Pixel Cell Array

lowest branches of this structure in each memory chip in the design, it is necessary to pass only two serial signal lines each for x- and y-data to each memory chip, regardless of the array size within the chip. These copies of the multiplier hardware occupy a negligible area on each chip.

The chip layout for our first implementation of this scheme is shown in figure 3. In order to provide reliability and testability, standard PLA's have been used for combinational logic, and extra test pads and circuitry have been built in. Without these features, the layout could be made much more compact, and we expect future implementations to have much greater density than the present one. The present 16 bits/pixel will be expanded to about 70 bits/pixel; more efficiently designed control circuitry will take up a very small portion of each pixel cell's area.

To calculate the expected processing speed of our design, assume that the polygon edges and Z values are represented to 16-bit accuracy, and the RGB intensity information to 24 bits. Under these conditions, roughly 180 machine cycles would be required to process a four-sided polygon. Current real-time (line-drawing) systems can process about one vertex every 12 microseconds (E&S Picture System II). Thus a four-sided polygon would take 48 microseconds, even when assuming the polygon clipping can be done in the same time as line clipping. In order to generate the full-color smooth-shaded, solid object images at the same rate, our machine would need a cycle time for a one-bit add of about 250 nsec, which is easily obtainable with current nMOS processes.

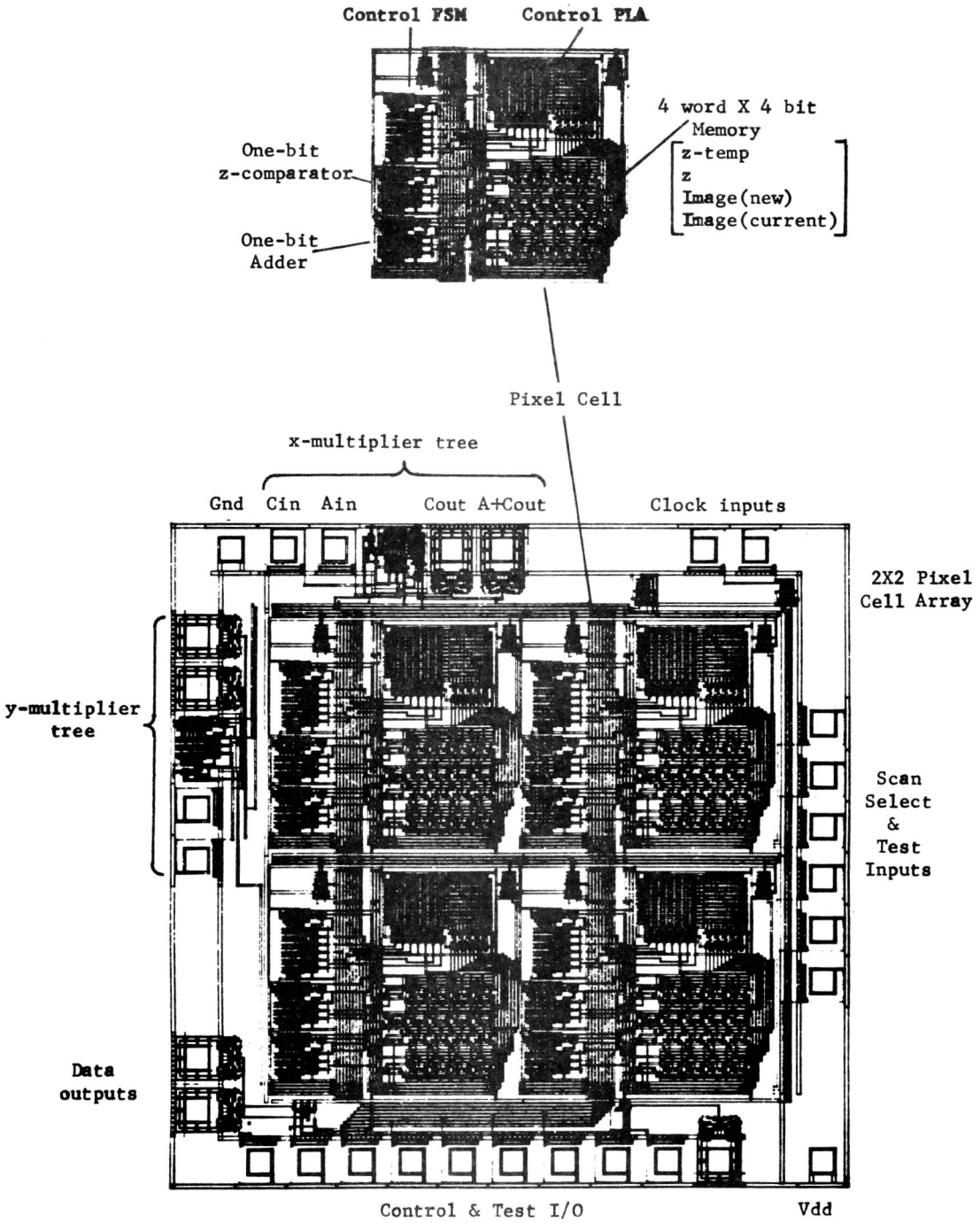

Fig.3: Chip Layout