A RASTER DISPLAY SYSTEM FOR COMPUTER GRAPHICS AND IMAGE PROCESSING

M. Wein*, N. Burtnyk*, J. Norton†, and W.A. Davis‡
* *National Research Council*
† *Norpak Ltd.*
‡ *University of Alberta*

ABSTRACT

A raster display system has been developed for support of a wide range of applications in surface graphics and image processing. The system can display color images at an approximate resolution of either 500 lines or 1000 lines. The controller of the display is based on a bipolar bit-slice processor. The development of this system was carried out under a program of R/D contracted to Industry.

UN SYSTÈME D'AFFICHAGE A BALAYAGE TV POUR L'INFOGRAPHIE ET LE TRAITEMENT DES IMAGES

RÉSUMÉ

Un système d'affichage à balayage TV à été développé pour assister plusieurs applications dans les domaines d'infographie à surfaces et de traitement des images par ordinateur. Le système est capable d'afficher des images avec une résolution d'environ 500 lignes ou 1000 lignes. Le controlleur est basé sur un processeur bipolaire en tranches. Le développement de ce système était mis en exécution par un programme de recherche sur contrat en industrie.

Introduction

The Computer Graphics Section at the National Research Council of Canada has been concerned with research in two areas. The first activity has been in interactive graphics, mainly interactive techniques using vector graphics displays. The other activity has been in image processing and pattern recognition, using storage displays.

With the trend of price/performance improvements in frame buffer-based raster displays, two years ago the section took an active interest in these displays for use in both the interactive graphics and the image processing projects. However, attributes of the displays at that time presented certain impediments to their use. They were not suited to supporting dynamic interaction since the plotting speeds were considerably slower than those of vector displays (with the exception of the very expensive image generators used for flight simulation). We were not able to identify systems on the market that would fulfil our requirements. Some specialized fast systems were beginning to appear but the speed was achieved at the expense of flexibility and generality.

With the aid of NRC funds earmarked for bolstering industrial R & D, we were able to instigate the development of a Visual Display Processor (VDP) that would

a) satisfy the frame buffer/raster display needs for our work in both interactive graphics and picture processing, and

b) provide the manufacturer of the VDP with the opportunity for capitalizing on the technology by producing a marketable product.

The desired functions that were established as target capabilities, are based on typical applications in our laboratory environment, but are also suited to meeting a wider range of situations.

The objectives were as follows:

1. The system should provide a structure for further work in parallel processing, through the use of separate multiple processors.
2. A facility should exist for evaluating different resolutions, and refresh rates of both 30 and 60Hz.

3. The interface structure should permit a connection to a host computer operating under a real-time event-driven system control program. In our case the host is a PDP-11 under the RSX-11M system control program. The main feature characterizing such an environment is the need for a half-duplex channel with relatively infrequent reversals of the direction of data flow.

4. The system should greatly improve frame update times for interactive graphics displays.

It should be noted that in a laboratory there is a need for one system to meet as many of the objectives as possible in a single display processor. This contrasts with the aspirations of the manufacturer (Norpak Ltd.) to cover a wide range of applications by configuring different units in a family of products, each for a specific application. The architecture of the system attempts to satisfy both of the above.

Overview

The raster display system resulting from the above design constraints, as manufactured by NORPAK Ltd, is based on a large solid state memory (frame buffer) which is used to store picture information. The contents of the memory are read out sequentially and in synchronism with the sweep of the video monitor. The basic format of reading out video information is compatible with conventional television. In North America the format is established by EIA (Electronic Industries Association) standard RS-170. The displayed picture consists of 640 by 480 pixels within the 525-line raster. Optionallly, the displayed picture may be displayed on a high resolution display and consists of 1024x1024 pixels. Several display formats are possible and these are shown in Table 1.

Table 1

Display Formats

Raster	Interface	Frame Rate	No. of pixels
525	2:1	30	640 x 480
559 approx.	2:1	30	512^2
525	1:1	60	640 x 480
1023	2:1	30	1280 x 960
1103 approx.	2:1	30	512^2

The main components of the system are shown in a simple form in Fig. 1. These include the image memory, a controller, output pixel processor and a video monitor.

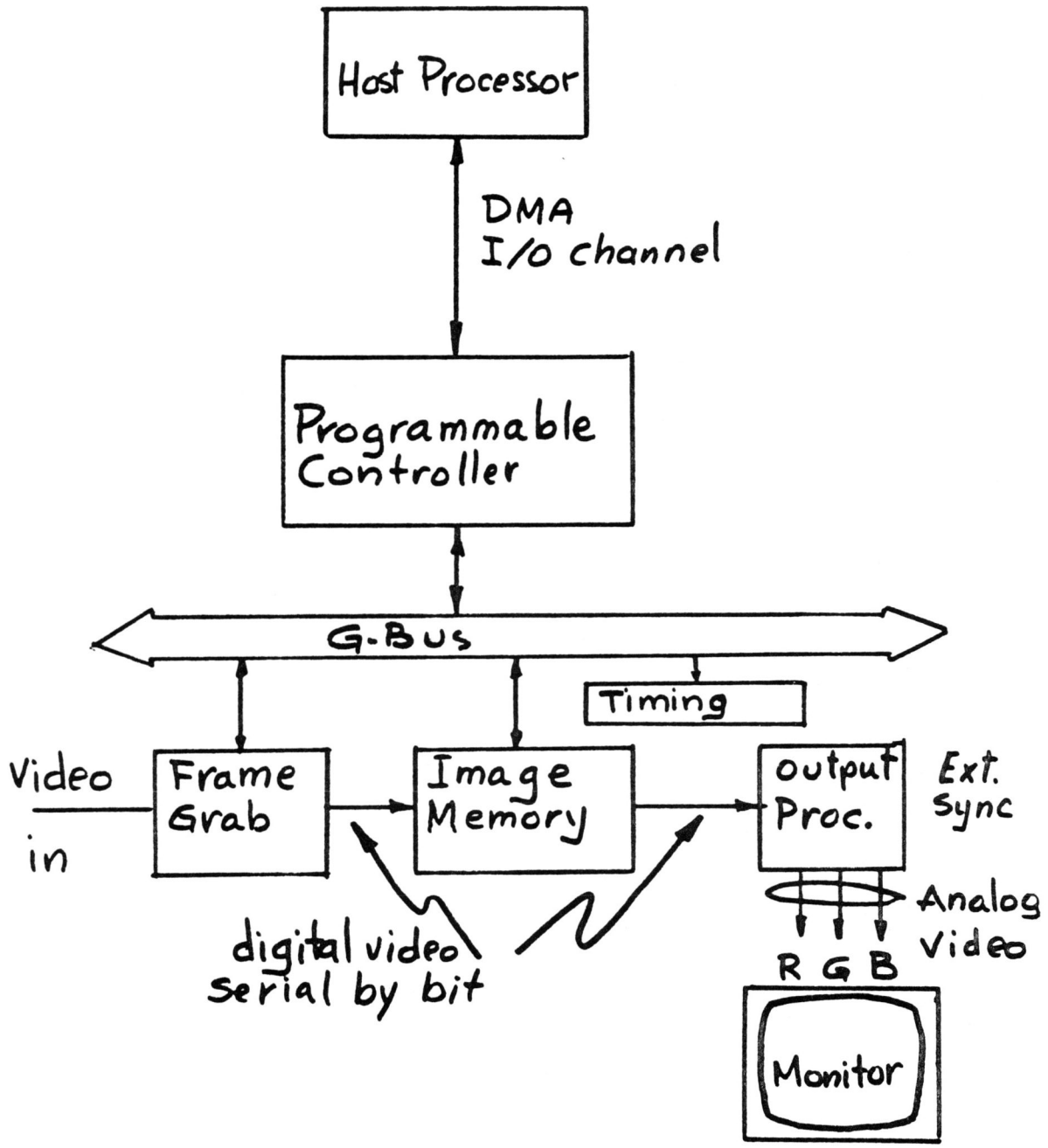

Fig. 1 System Block Diagram

The DMA input/output channel establishes a link between the controller and the host computer.

The data rate out of the image memory to the pixel processors is fixed and synchronous with the TV raster. On the other hand the bandwidth from the host processor through the controller to the image memory is variable and it determines to a large part the effective performance of the system. Three major factors have been identified to achieve the wide bandwidth:

(a) host DMA rate;
(b) the graphics bus:
(c) access to the image memory.

The Graphics Bus

The backbone of the display system, shown in Fig. 1 is a synchronous graphics bus (G-bus) which provides the communication path between different modules in the system. The bus is sufficiently wide (24 bits) that all information for one elementary transaction is moved in one bus cycle. This elementary transaction can then transfer three 8 bit pixel components to three image memory banks in on cycle. The bus is "position independent" so that there are no dedicated positions or slots, and it does not impose a protocol on the contents of the data. Thus, when a new module is defined, the protocol for communication between it and the controller is entirely private to the source and destination modules. Only the process of handshaking and priority computation are predetermined.

Image Memory

The display processor has a large memory partitioned into planes. Each plane contains one bit for each x,y coordinate point, or pixel, on the screen. Therefore a memory plane for a high resolution (2^{10} x 2^{10}) display contains 2^{20} bits. A memory "bank" is made up of 8 planes, and therefore contains 8 bits per pixel. A complete red-green-blue picture requires three memory banks, one for each color. The memory can be thought of as a cube with the X and Y axes corresponding to the X and Y axes on the display and the Z axis spanning the memory planes, as shown in Fig. 2.

The memory can be read from or written into in two logically different ways. The first accesses a single bit from each plane, from up to 24 planes in three banks. Therefore, one read cycle accesses the three color components at one coordinate location.

The second view of reading the memory is that of reading 16-bit wods within a plane of memory. Here the 16

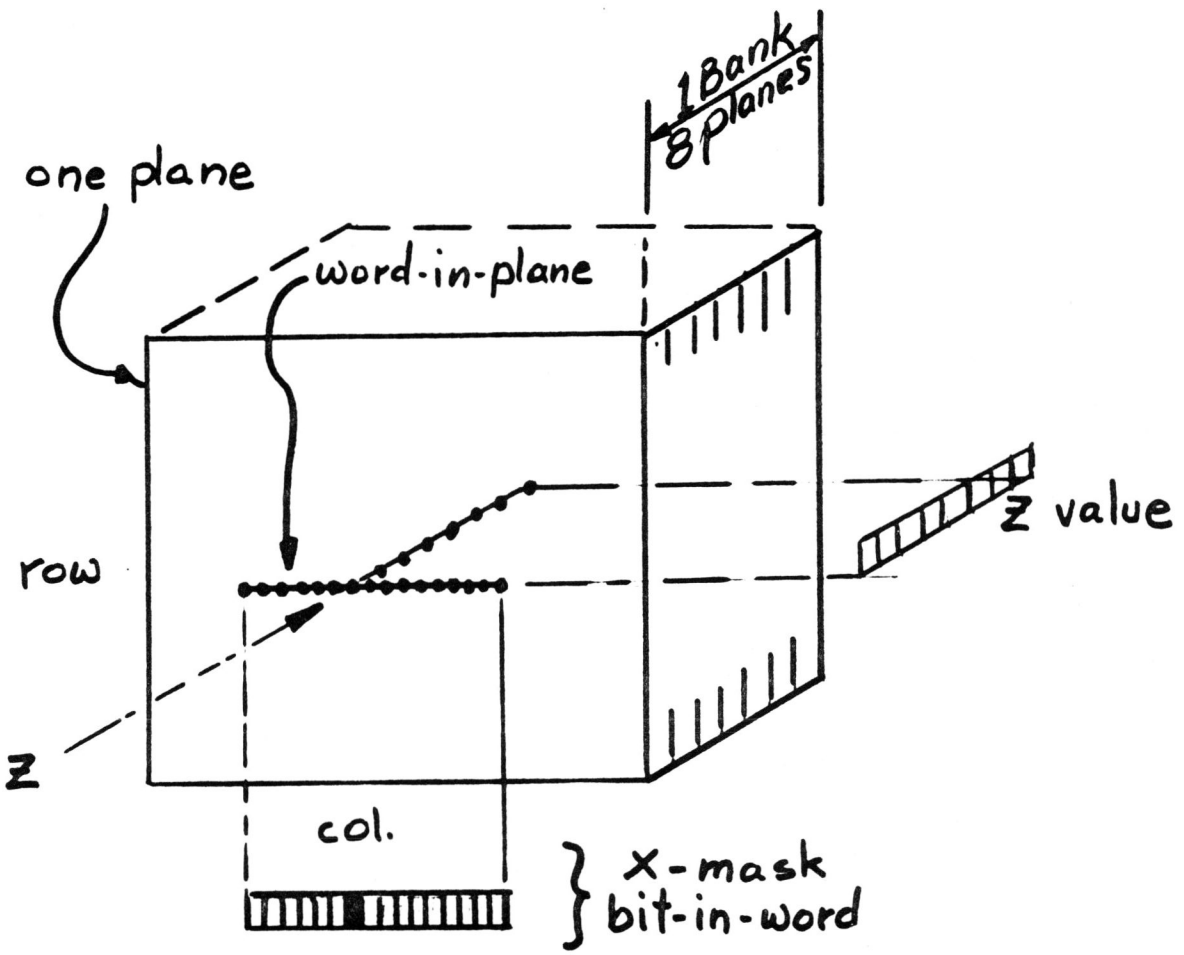

Fig. 2 Memory Organization

bits represent 16 horizontally adjacent pixels, and each of the 16 bits in one plane represents one bit of the pixel value at the corresponding XY location.

The first view represents the way the image memory appears logically from the G-bus. The second view represents the physical organization of the memory plane. Each memory plane is organized into an array of 16-bit words in order to support the high data rate required for refreshing the display at video rates. Video output is generated by reading out sequentially the memory words in each plane and shifting the words to generate serial output.

The two styles of memory access can be combined in a way, that allows a pixel value to be written from the G-bus simultaneously to some, or all horizontally adjacent pixels, that have the same hardware word in each plane. This allows rapid filling of uniformly shaded images.

Programmable Controller

The requirements imposed by the functional objectives, primarily the objective that the display be flexible enough for use in both interactive graphics and picture processing, poses a severe demand on the controller. Bit-slice microprocessors of the AMD 2901 series provides a good compromise of speed, flexibility and price and were thus chosen as the principal components of the controller.

A simplified block diagram is shown in Fig. 3. Four 4-bit slice chips are organized to make up the 16-bit data word. As configured the processor bears some resemblance to a 16-bit computer minus the interrupt processor and DMA channels. The processor, similar to many current computers, has two levels of programming: the machine code using single word 16 bit instructions, and a micro code program which executes the instructions. Whereas machine instructions in a mini-computer are executed typically by 4 to 6 micro instructions, the machine instructions in the display controller are more special-purpose, complex, and need longer micro-code "programs" to execute.

The width of the micro-code control word is 64 bits and is considerably greater than those encountered in a typical mini (24 bits in HP, 44 bits in PDP-11). The wide word maintains flexibility in permitting future implementation of a wide range of special microcode. The structure is not directed at a narrowly defined architecture and its machine code.

The processor is organized around its local 16-bit data bus (D-bus). All the components can communicate over the D-bus, although there are additional data paths which may be used in order to avoid delaying bus conflicts.

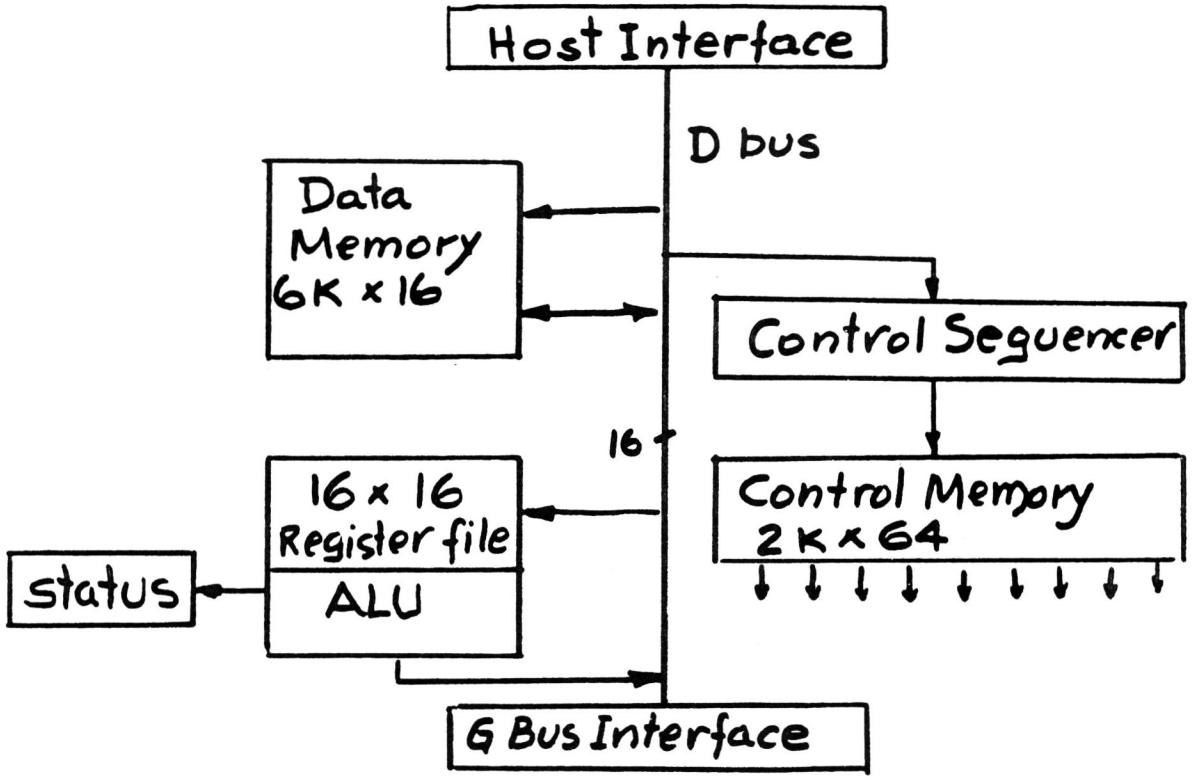

Fig. 3 Programamble Controller

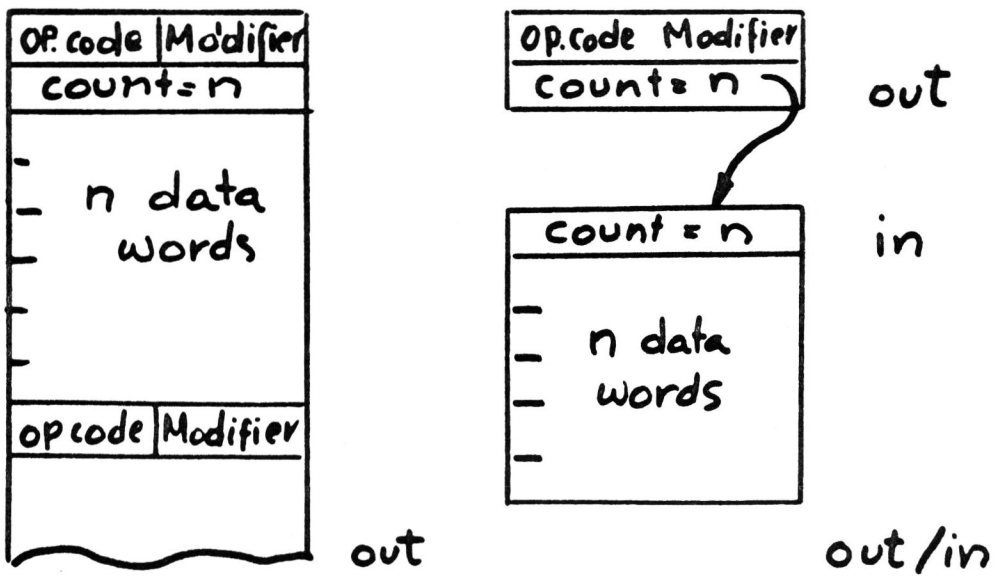

Fig. 4 Command Format

The controller consists of the following components:

(a) Macro memory of 6K 16-bit words, used to store command sequences and temporary data.

(b) Arithmetic logic unit with a file of 16 registers. This is a classical ALU.

(c) A microprogram control memory of 2K 64-bit words.

(d) Microprogram sequencer.

(e) Condition and status multiplexer.

The control memory can be either read/write or read-only or a combinati on of both, partitioned on a 1-K boundary. It should be pointed out that user microcoding requires considerable detailed knowledge of the structure, a good micro code assembler and a logic diagnostic tool.

Display Command Set

The set of commands consists of two groups supporting the two fields in which we are interested: interactive graphics and image processing. Interactive graphics is supported with commands that output lines or areas to the frame buffer. These lines and areas may have uniform, or uniformly tapered intensity and color. Some commands are suited to building the image object by object. Others are scan-line oriented. The choice is dictated by the algorithms used to generate the image.

Image processing is supported by a large repertoire of pixel operations which permit the transmission of values to or from the frame buffer. The commands operate on meaningful entities - row or column of pixels, square neighbourhood or arbitrarily defined regions.

A typical characteristic of pixel operations is the bidirectional nature of the flow of information. For example, in the case of a pixel read operation, row/column addressing is flowing to the display, while pixel values are being returned to host. Typically, the reversal of the direction and the resultant short message length lead to inordinate overhead in many operating systems. The commands have been formulated to minimize the reversal by providing buffering in the programmable controller.

In the image processing support, pixel data are transmitted in one of four formats. In all forms the element is an 8-bit pixel value.

Format 1: Two values, packed, directed to two banks at one row/column address.

Format 2: Two values, packed, directed to one bank, at two adjacent pixel locations.

Format 3: Full color (RGB), packed, three values directed to three banks at one row column address.

Format 4: Full color (RGB), similar to Format 3, but unpacked.

Each command attempts to execute directly the functions that one would define in call sequences in a well designed support package. Consequently, the link between the call from an application program and the processor instruction is simple and direct.

The form of the command is shown in Fig. 4. The command itself consists of one or two 16 bit words. The first contains the operation code and a modifier. The second word, when required contains the data count for the block of data words that follows. We have avoided the common practice of including command bits and flags in data words.

Display commands may be passed from the host computer to the display controller in one of two modes. In the basic mode each command is executed as it is received and is not retained. The controller acts as if it were hard-wired.

The second mode permits storing sequences of commands in the macro memory in the form of subroutines. Subsequent commands can invoke each of the subroutines. Since most commands can use either relative or absolute row/column addressing, the stored subroutine permits repeated operations on pixel neighbourhoods. Such operations are encountered frequently in image processing.

A base for future development

This paper is concentrating on the role of the display system in our laboratory program. A second aspect, that of the future development as a product and the intentions of the manufacturer as represented by one co-author (Norton), are not addressed in this paper).

In the laboratory we intend to use the system for experimental work in image synthesis with a stress on dynamic graphics. The system was designed as a general tool and deliberately excludes special hardware for specific functions. Consequently, several enhancements will be required to support the work in dynamic graphics. We intend to develop dedicated special-purpose processors for rapid generation of surface images. The bus structure of the display lends itself to connecting special purpose

processors. Furthermore, the present command structure allows communication between the host and these special processors.

A second activity will be the development of specialized micro-code. We will be experimenting with implementing low level algorithms that are needed in surface graphics. This work will require that part of the control store be read/write memory rather than the more conventional ROM. It remains to be seen how practical it is to develop and down load user micro-code.