ARITHMETIC LOGIC UNIT - REGISTER SECTION

DATA FROM EXTERNAL REGISTERS

EMIT (PIPELINE) 16
BUFFER PRESELECT
A INPUT

(32 INPUTS)

DATA FROM EXTERNAL REGISTERS

(16 INPUTS)

BUFFER PRESELECT
B INPUT

TIMING/CLOCK LOGIC

EXTERNAL REGISTERS

PIPELINE OUTPUT SELECT

OUTPUT SELECT DECODER

CLOCK

STATS
LATCH

ROTATE AND SHIFT UNIT

ALU 2903
INTERNAL REGISTER FILE
R0-R15
Q-REGISTER

POST FORMAT SELECTORS

MULTIPLIER ACCUMULATOR

C - ADDRESS UNIT

D - ADDRESS UNIT

RANDOM NUMBER GENERATOR
MULTIPLIER - ACCUMULATOR

FROM ALU OUTPUT

MULTIPLIER MODE REGISTER

X - INPUT REGISTER

Y - INPUT REGISTER

PRODUCT REGISTER

CONTROL

CLOCK/TIMING LOGIC

ALU REGISTER DECODER

EXTENDED PRODUCT TO ALU A-INPUT Ø-2

MOST SIGNIFICANT PRODUCT TO ALU A-INPUT Ø-15

LEAST SIGNIFICANT PRODUCT REGISTER

DRIVER TO ALU A INPUT Ø-15
ANALOG TO DIGITAL CONVERTERS (HIGH SPEED - VIDEO)

COMPOSITE VIDEO INPUT (ANALOG IN)

1 VOLT P-P

BLACK LEVEL (OFFSET)

D.C. RESTORED VIDEO (ANALOG)

CLAMP PULSE

SYNC SEPARATOR

CLAMP PULSE LOGIC

VOLTAGE REFERENCE

REFERENCE ADJUST (GAIN)

ANALOG TO DIGITAL CONVERTER

CONVERT STROBE

CLOCK TIMING LOGIC

OUTPUT LATCH

OUTPUT CLOCK

* N IS THE NUMBER OF BITS CONVERTED BY THE A/D

N = 4
SEQUENCER - CONTROL UNIT

MICRO PROGRAM SEQUENCER
AMD - 2910

WRITEABLE CONTROL STORE
256 WORDS x 80 BITS

PIPELINE REGISTER
80 BITS

TIMING CLOCK LOGIC

TEST SELECTOR

PIPELINE CONTROL

TEST INPUT

INSTRUCTION IN

ADDRESS OUT

LSI-II CONTROL

MICRO ADDRESS SELECTOR

LSI-II DATA IN

MICRO ADDRESS REGISTER LSI-II

LSI-II DATA IN

LSI-II DATA IN

TO ALU ANALOG/DIGITAL CONTROL SECTION
Jeff Schier/Don McArthur Digital Image Processor
5/11/92 (F) Jeff Schier

The Schier/McArthur Digital Image Processor was constructed in 1976-1977 at Steina and Woody Vasulka's loft in Buffalo, N.Y. It began as a mathematic exploration by Don McArthur of the digital raster, and was built from digital modules locked to video time by the LSI-11 sixteen bit micro-computer. It was built in stages starting with the sync generator and computer interface, later adding a digital Selector, Arithmetic-Logic Unit, lookup/pattern RAM and a rectangular Window generator. The video outputs came from three 4 bit digital to analog converters, and was converted to color composite video by an external NTSC encoder. The video tape recorder was on a continuous standby, allowing documentation of the design process by Steina Vasulka through "pressing the record button".

The modules were "wire-wrapped" and connected to the computer control and timing bus at the rear of the modules. The digital video paths were patched together with multi-conductor ribbon cables, plugged into the front of each module. External audio could be patched in or out from the front panel, converting the video timing signals to sound. Emphasis was placed on internal square waveforms to form the first pictures, made from the horizontal and vertical bar patterns that subdivide the raster. A borrowed time base corrector was "hot-wired" to pull out 6 bits of live digital video from its A to D converter and color-mapped through the lookup/pattern RAM. A random "power-up" pattern was saved from the RAM and formed a favorite color test palette for adding colors to the image. The real time remapping of intensity to color formed a color precision (64 levels) unseen in analog colorizers. Dual four bit A to D converters were later constructed to digitally combine two image sources. Operations were performed at 4 bit resolution per red, green and blue channel, but were funneled down to 6 bits when running through the lookup/pattern RAM.

The digital combination of binary images formed unique geometric color patterns. These were unexpected and did not correspond to other analog processes. This became evident when the Arithmetic/Logic Units (ALU's) were installed. The ALU's performed arithmetic functions (addition/subtraction) and logical functions (And, Or, Exor Negation) and wacky mixed arithmetic and logical operations that were "thrown in" by the semiconductor vendor, such as (A OR B plus 1). The bitwise combination of image combined with overflow-wraparound conditions generated unusual patterns of color and box-like textures, without equivalence in analog video. The binary operations made sense, but the images were a digital surprise. "Official" test images were needed to test out the ALU process. This consisted of a white styrofoam sphere or cone and Woody's hand waved in front of the camera. These test images contain 16 discernible levels of grey, useful to disclose the arithmetic/logical binary
combinations and overflow conditions. The explorations of real time digital video at the Vasulka's predated later image processing and digital video effects units. The exploration of binary operations between images has largely been ignored in image processing and computer graphics, in its quest for photo-realistic imagery.

Time locked software marching to the video frame rate formed the real time control structure needed to operate the digital image processor. Various test and control table programs were written in Fortran and PDP-11 Assembly language to operate the processing modules. Walter Wright programmed "BARBAR" an assembly language control program with independent timing control stacks. BarBar's timing stacks control processing module functions, time delays, and the looping of the control sequence. The inclusion of random functions exercise the hardware, contributing to long sequences of digital permutations.

Hardware: Consists of a rack of digital processing modules, a gen-locked sync generator, a vertical interval control bus, and a microcomputer to orchestrate the field by field control. The digital video paths for the processing modules are "patched" through their front panels.

Signal Path: Input is received through camera sources, video tape sources, or the internal pattern source (H and V timed bar patterns). For camera/vtr sources, these route through the A/D converters first, are front panel patched to the processing modules, convert back to the D/A converters to R,G,B video and then go to an RGB to NTSC encoder for composite color output for recording and viewing.

1) Micro-computer - A 16 bit DEC LSI-11 microprocessor coordinates control words for the processing modules and handles user interface functions

2) A Vertical Interval Control Buffer and Transfer Bus - Control information is loaded into this control buffer by the microprocessor during the current active field. The data is shipped down to processing modules during the next vertical blanking interval.

3) Processing Modules A) Analog to Digital Converters (A/D) - two 4 bit converters.

B) Selectors - 3 groups of selectors - chooses between 8 horizontal, and 8 vertical frame locked patterns, and an external digital source. The selectors allow bit-wise selection of horizontal, vertical timing components and external video inputs.

C) Arithmetic Logic Units (ALU's) - Combines two digital input streams into a single output through combinations of arithmetic and Boolean logic functions (Output = function (A_in, B_in). The Boolean functions of 'AND', 'OR', 'EXOR', 'EXNOR', Ones Complement are present. The arithmetic 'A PLUS B PLUS CARRY', 'A MINUS B PLUS CARRY', 2's Complement are also available. Certain combination arithmetic with logical operations are possible, with a 'Constant' available on the 'B' input, useful for bit masking.
D) Lookup - an R,G,B lookup table with common digital 'Address input' is present, to perform intensity/pseudo color transformations. The memory could be loaded then scanned out as a small raster.

E) Window generator - three Window generators, form an adjustable frame for gating/routing the digital sources. The frames are independently programmable on a pixel/line basis. Wipe patterns and title boundaries are formed by these.

F) Digital to Analog Converters - one apiece for red, green and blue components at 4 bits per gun.

4) A Gen-Lockable Sync Generator - forms sync timing, and is subdivides the active screen into 512 H by 486 V coordinates. Both video sync and H and V timing information is available on the control bus, for pickoff by modules. A Phase Locked Loop locks the clock timing to an external sync source.

5) RGB to NTSC Color Encoder - the funnel for output, converts the RGB signals from the D/A converters to an NTSC color composite video signal for display and recording onto video tape.
BUFFER BUFFERERY

AND JUST LOOK AT THOSE FEATURES:

EXPANDING AND SHRINKING
EDGE EXTRACTION
GROWTH OF THE EDGES
IMAGE DECAY
    RANDOM DECAY
    GREYSCALE DECAY
    SPATIAL DECAY
IMAGE REGENERATION - BUILDING UP FROM FRACTURED PIECES
PICTURE PULVERIZATION
COLOR SHIFTING
COLOR EXCHANGE
COLOR MISREGISTRATION - MISPRINTING OF THE RED, GREEN AND BLUE COLOR COMPONENTS

DEGRADATION
DEGENERATION
COMPOST HEAP MODE
CHARACTER / TEXT GENERATOR
TEXTURAL NEBULIZER
DYNAMIC INTERACTION BETWEEN BUFFERS
MOVEMENT PHENOMENON
DISSOCIATION
TIME INTEGRATION
POSITION

FIELD MANIPULATION
MULTIPLE IMAGE INTERPENETRATION
DECEASE MODE
DIESEL GUN
COLOR BANDING
COLOR BLENDING
IMAGE WARPING
NON OPTICAL IMAGE AND SOUND EXTRACTION (NOISE)

AUDIO
COLOR CORRUPTION
RETIMING OF VIDEO
REEZE FRAMING
KEYING
MULTIDIMENSIONAL WIPES
FEEDBACK BETWEEN BUFFERS
HORIZONTAL AND VERTICAL DELAY LINES
PANCAKE AND WAFFLE MODE

BIG EXTRACTION
    SCRABBLING
    POACHING
    CODDING
    FRYING
    ONCE OVER LIGHTLY
    HARD BOILED
    OMELET
    WESTERN

FOOD PROCESSING
    JUICING
    DICING
    SLICING
    CRUSHING
    SLITTING
    PITTING

GREYSCALE COMPACTOR
IMAGE EMULSIFIER OVERVIEW

BY JEFFY SCHIER

THIS IS A SHORT OVERVIEW ON THE CHARACTER OF THE ELECTRONIC 'IMAGE EMULSIFIER.' THE EMULSIFIER WILL CAPTURE IMAGES AND ENCASE THEM WITHIN FINE MESHES OF STORED LIGHT (IMAGE BUFFERS). THESE MESHES ARE TREATED AS ELASTIC MEMBRANES OF COLOR AND TEXTURE UNDER THE SCRUTINY OF A DIGITAL DEMON (THE VIDEO PROCESSOR).

ONCE AN IMAGE IS STORED, IT IS CODED MATTER, ELECTRICAL FUEL FOR FURTHER ORGANIC AND MECHANICAL ABSTRACTION. TO FULFILL THIS, THE DEMON IS INSTRUCTED (MICROPROGRAMMED) TO CREATE VISUAL PHENOMENA INCLUDING:

- MELTING PICTURES INTO FROSTY PUDDLES OF DRIPPING COLOR
- COAGULATING AND GRAFTING GREY-SCALES BETWEEN IMAGE PLANES
- GROW IRIDESCENT CRYSTALINE SPIRES, FROM EXTRACTED EDGES
- KNEAD, WARP, PULVERIZE AND SHATTER IMAGE BY SPATIAL AND SPECTRAL DECAY
- GIVE BIRTH ON SCREEN, TO NON OPTICAL IMAGES

FOR SOUND EXTRACTION (NOISE)

THIS PROJECT INVOLVES Merging THE DIVERSE METHODS OF VIDEO, DIGITAL ELECTRONICS, AND COMPUTER TECHNOLOGIES; TO EVOLVE A NEW GENERATION OF IMAGING TOOLS.
BUFFER BUFFOONERY

AND JUST LOOK AT THOSE FEATURES:

EXPANDING AND SHRINKING
EDGE EXTRACTION
GROWTH OF THE EDGES
IMAGE DECAY
  RANDOM DECAY
  GREY SCALE DECAY
  SPATIAL DECAY
IMAGE REGENERATION - BUILDING UP FROM FRACTURED PIECES
PICTURE PULVERIZATION
COLOR SHIFTING
COLOR EXCHANGE
COLOR MISREGISTRATION - MISPRINTING OF THE RED GREEN AND BLUE COLOR COMPONENTS
DEGRADATION
DESINTIGRATION
COMPOST HEAP MODE
CHARACTER / TEXT GENERATOR
TEXTURAL NEBULIZER
DYNAMIC INTERACTION BETWEEN BUFFERS
MOVEMENT PHENOMENON
DISLOCATION
  TIME
  INFORMATION
  POSITION
FIELD MANIPULATION
MULTIPLE IMAGE INTERPENETRATION
DECAY MODE
MOIRE GUN
COLOR BANDING
COLOR BENDING
IMAGE WARPING
NON OPTICAL IMAGE AND SOUND EXTRACTION (NOISE)
AUDIO
COLOR CORRUPTION
RETIMING OF VIDEO
FREEZE FRAMING
KEYING
MULTIUDIMENSIONAL WIPES
FEEDBACK BETWEEN BUFFERS
HORIZONTAL AND VERTICAL DELAY LINES
### INDEX

<table>
<thead>
<tr>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0 SYNOPSIS</td>
<td>1</td>
</tr>
<tr>
<td>0.1 ENVIRONMENT HISTORY</td>
<td>1</td>
</tr>
<tr>
<td>0.2 PHYSICAL LAYOUT</td>
<td>2</td>
</tr>
<tr>
<td>1.0 PROCESSOR INTERFACE</td>
<td>3</td>
</tr>
<tr>
<td>1.1 BUS CONTROL REGISTER</td>
<td>3</td>
</tr>
<tr>
<td>2.0 X Y COUNTER MODULE</td>
<td>4</td>
</tr>
<tr>
<td>2.1 ZOOM REGISTER</td>
<td>4</td>
</tr>
<tr>
<td>2.2 SHIFT REGISTER</td>
<td>5</td>
</tr>
<tr>
<td>2.3 PIN ASSIGNMENTS</td>
<td>5</td>
</tr>
<tr>
<td>3.0 DATA SELECTOR</td>
<td>6</td>
</tr>
<tr>
<td>3.1 SELECTION REGISTERS</td>
<td>6</td>
</tr>
<tr>
<td>3.2 INVERSION REGISTER</td>
<td>7</td>
</tr>
<tr>
<td>3.3 PIN ASSIGNMENTS</td>
<td>7</td>
</tr>
<tr>
<td>4.0 WINDOW MODULE</td>
<td>8</td>
</tr>
<tr>
<td>4.1 EDGE POSITION REGISTER</td>
<td>9</td>
</tr>
<tr>
<td>4.2 WINDOW FUNCTION ALU</td>
<td>10</td>
</tr>
<tr>
<td>4.3 PIN ASSIGNMENTS</td>
<td>11</td>
</tr>
<tr>
<td>5.0 ALU MODULE</td>
<td>12</td>
</tr>
<tr>
<td>5.1 ALU CONTROL REGISTERS</td>
<td>12</td>
</tr>
<tr>
<td>5.2 ALU NUMERIC REGISTERS</td>
<td>13</td>
</tr>
<tr>
<td>5.3 ALU PIN ASSIGNMENTS</td>
<td>14</td>
</tr>
<tr>
<td>6.0 MEMORY MODULE</td>
<td>15</td>
</tr>
<tr>
<td>6.1 MEMORY CONTROL REGISTER</td>
<td>15</td>
</tr>
<tr>
<td>6.2 MEMORY SHIFT REGISTER</td>
<td>16</td>
</tr>
<tr>
<td>6.3 WRITE THEN DISPLAY</td>
<td>16</td>
</tr>
<tr>
<td>6.4 PIN ASSIGNMENTS</td>
<td>17</td>
</tr>
<tr>
<td>7.0 EXPORT IMPORT MODULE</td>
<td>18</td>
</tr>
<tr>
<td>7.1 DATA REGISTERS</td>
<td>18</td>
</tr>
<tr>
<td>7.2 PIN ASSIGNMENTS</td>
<td>18</td>
</tr>
<tr>
<td>8.0 REGISTER LIST</td>
<td>19</td>
</tr>
</tbody>
</table>
This guide describes the functional and programming aspects of the Schier/MacArthur Imaging Bus (henceforth the SAM bus). This device is an interface from digital computer (LSI-11) to video. The system is physically modular in nature, and therefore facilitates a wide variety of operational modes.

The functions of these modules and the effects they manifest are similar to that of an all-analog processing environment; where some modules act as generators of video, like analog oscillators, while others act as filters or processors of an input video signal. The difference lies in that the SAM modules deal exclusively with video as a digitized stream of bits, which can be generated internally provided from an external source such as a digitized TV camera image, or a mixture of these. In either case the bit streams are converted to analog voltages to produce a conventional video signal.

Although the modular nature of the SAM bus doesn't offer much in the way of friendliness to the user unaccustomed to the ways of digital, the modularity provides a most fertile structure for the exploration of a vast variety of control structures. Approximately 100 device registers facilitate computer program control over real-time processing and synthesis of video. Any or all of these control registers may be modified as often as field rate (1/60th of a second).

0.1 ENVIRONMENT HISTORY

The SAM bus was designed for the Vasulkas by Don Mac Arthur in 1976. Since then, 5 modules have been designed and implemented on the bus, most by Jeffy Scheir. The host processor is a Digital Equipment Corporation LSI-11 with 48 kilobytes of memory and dual floppy-disk drive. Control of the SAM bus by the processor is implemented via approximately 100 device registers which are mapped into LSI-11 address space in the range of 171000 to 171776 octal.
0.2 PHYSICAL LAYOUT

The S&H bus is a card rack containing 8 modules; the sync generator, 2 for the processor interface, and 5 programmable modules. The sync generator provides timing for the bus and is gen-lockable to an external video source. The processor interface modules are connected by ribbon cable to an interface card residing in the LSI-11 backplane.

The inputs and/or outputs of each module are present on 16 pin DIP connectors on the front panel. Pinout conventions are such that any single output can safely drive any reasonable number of inputs. Two outputs may not be tied together. The DIP connectors are connected to one another by ribbon cable patch cords. The particular pins and pinouts will be discussed with each module.
1.0 PROCESSOR INTERFACE

The S&M bus shares 256 words of buffer memory with the LSI-11. These locations behave to the LSI-11 just like read-write memory locations. At regular intervals, usually video field rate, the contents of this buffer memory is transferred to device registers on the S&M bus. This event can be enabled/disabled in software via the S&M bus status and control register.

1.1 BUS CONTROL REGISTER - 171776

Bit 0 of this register is currently the only one implemented. On LSI-11 bus reset and power-up, this bit is cleared to 0. Any modifications made to the buffer memory by software will have no effect on the device registers since the transfer is not enabled. When this bit is set by software, the buffer memory to device register transfer is enabled. During the actual transfer, buffer memory cannot be accessed by the LSI-11, and a bus timeout trap will occur if an attempt is made to do so. Setting this bit also has other implications; at the end of buffer transfer, the S&M interface issues an interrupt to the LSI-11, causing the processor to fetch a new PC and PS from address 170 octal. Software must deal with this; an example of how to set up and service the S&M device is depicted thusly

```
START: MOV #SBSERV,#170
       MOV #0,#172
       BIS #1,#171776
LOOP:  BR LOOP

; FOLLOWING ROUTINE IS CALLED BY S&M HARDWARE INTERRUPT
; AT THE END OF A BUFFER TRANSFER

SBSERV: BIC #1,#171776
        JSR PC,BUSBSERV
        BIS #1,#171776
        RTI
```

; PUT ADDRESS OF SERVICE ROUTINE
; AND PROC. STATUS WORD
; IN VECTOR LOCATION
; ENABLE TRANSFER-INTERRUPT
; WAIT FOR INTERRUPT OR GO ON
; ABOUT BUSINESS.

;DISABLE INTERRUPT-TRANSFER
;CALL SOME ROUTINE THAT UPDATES-ACCESS
;S&M BUFFER MEMORY
;REENABLE INTERRUPT
; AND RETURN
2.0 X Y COUNTER MODULE

This module is based around two independent 8 bit up counters. The X counter uses a pixel rate clock as input. The Y counter uses a horizontal line rate clock. Both these clocks may be divided by a value from 1 to 16 before being input to the counters.

2.1 ZOOM REGISTER - 171052

The lower nybble of each byte of the zoom register control the divisor by which the respective clock is divided; 0 = divide by 1, 15 = divide by 16. It is termed the zoom register because of the visual effect it produces.
2.2 SHIFT REGISTER - 171050

At the start of an active field, both the X and Y counters are reset to the values contained in this register. The counters will then proceed to count up from that value, wrapping around to 0 when reaching 255.

2.3 X Y COUNTER PIN ASSIGNMENTS

<table>
<thead>
<tr>
<th>PIN</th>
<th>SIGNAL</th>
<th>PIN</th>
<th>SIGNAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X0</td>
<td>9</td>
<td>Y0</td>
</tr>
<tr>
<td>2</td>
<td>X1</td>
<td>10</td>
<td>Y1</td>
</tr>
<tr>
<td>3</td>
<td>X2</td>
<td>11</td>
<td>Y2</td>
</tr>
<tr>
<td>4</td>
<td>X3</td>
<td>12</td>
<td>Y3</td>
</tr>
<tr>
<td>5</td>
<td>X4</td>
<td>13</td>
<td>Y5</td>
</tr>
<tr>
<td>6</td>
<td>X5</td>
<td>14</td>
<td>Y5</td>
</tr>
<tr>
<td>7</td>
<td>X6</td>
<td>15</td>
<td>Y6</td>
</tr>
<tr>
<td>8</td>
<td>X7</td>
<td>16</td>
<td>Y7</td>
</tr>
</tbody>
</table>
3.0 DATA SELECTOR MODULE

This module consists of twelve 16 to 1 data selectors. It permits the channeling of any one of 16 inputs to any of twelve outputs. Also, any of these outputs may be selectively inverted. The data selector is housed in the same module and above the X Y counter module. A conceptual block diagram of the data selector follows.

3.1 SELECTION REGISTERS - 171040 RED
171042 GREEN
171046 BLUE

There is one selection register for every 4 bits of output. These 3 groups of 4 bits are arbitrarily referred to as red, green, and blue. This convention is purely a mental one. Any one of the 16 inputs may be directed to a particular output by placing the input channel number in the appropriate output bit field of a selection register. For example; placing the pattern 1010101010101010 (125252 octal) in the red selection register would channel input number 10 to all 4 bits of the red output group.
The twelve outputs from the data selector section can each be complemented by placing a 1 in the appropriate bit of the inversion register. A 0 will pass the respective channel unchanged.

### 3.3 DATA SELECTOR PIN ASSIGNMENTS

**INPUT CONNECTOR ASSIGNMENTS**

<table>
<thead>
<tr>
<th>PIN</th>
<th>SIGNAL</th>
<th>PIN</th>
<th>SIGNAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CHAN0 INPUT</td>
<td>9</td>
<td>CHAN8 INPUT</td>
</tr>
<tr>
<td>2</td>
<td>CHAN1 &quot;</td>
<td>10</td>
<td>CHAN9 &quot;</td>
</tr>
<tr>
<td>3</td>
<td>CHAN2 &quot;</td>
<td>11</td>
<td>CHAN10 &quot;</td>
</tr>
<tr>
<td>4</td>
<td>CHAN3 &quot;</td>
<td>12</td>
<td>CHAN11 &quot;</td>
</tr>
<tr>
<td>5</td>
<td>CHAN4 &quot;</td>
<td>13</td>
<td>CHAN12 &quot;</td>
</tr>
<tr>
<td>6</td>
<td>CHAN5 &quot;</td>
<td>14</td>
<td>CHAN13 &quot;</td>
</tr>
<tr>
<td>7</td>
<td>CHAN6 &quot;</td>
<td>15</td>
<td>CHAN14 &quot;</td>
</tr>
<tr>
<td>8</td>
<td>CHAN7 &quot;</td>
<td>16</td>
<td>CHAN15 &quot;</td>
</tr>
</tbody>
</table>

**OUTPUT CONNECTOR ASSIGNMENTS**

<table>
<thead>
<tr>
<th>PIN</th>
<th>SIGNAL</th>
<th>PIN</th>
<th>SIGNAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RED0 OUTPUT</td>
<td>9</td>
<td>BLUEC OUTPUT</td>
</tr>
<tr>
<td>2</td>
<td>RED1 &quot;</td>
<td>10</td>
<td>BLUE1 &quot;</td>
</tr>
<tr>
<td>3</td>
<td>RED2 &quot;</td>
<td>11</td>
<td>BLUE2 &quot;</td>
</tr>
<tr>
<td>4</td>
<td>RED3 &quot;</td>
<td>12</td>
<td>BLUE3 &quot;</td>
</tr>
<tr>
<td>5</td>
<td>GREEN0 OUTPUT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>GREEN1 &quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>GREEN2 &quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>GREEN3 &quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.0 WINDOW MODULE

The window module generates 3 so-called windows: a set of 4 edges, two vertical (X), two horizontal (Y) (with respect to the frame). The position of each edge and how the inside and outside are defined are under software control.

There are 3 duplicate window generating channels, Red, Green, and Blue. Each channel has 4 position control registers designated X1 Y1 X2 Y2. Each X and Y pair generate a pulse whose starting and ending times correspond to the value in the respective registers. Thus

WINDOW ALU:

RED - 171030
GREEN - 52
BLUE - 34
4.1 EDGE POSITION REGISTERS

The lower 9 bits of each edge position register control the position of each edge of the window in the field. The $X_1$ and $Y_1$ registers control the leading edge, the $X_2$ and $Y_2$ registers control the trailing edge. The way these values map into the field is depicted as follows:

**Window Edges:**
- RED $X_1$: 171000
- RED $Y_1$: 1
- RED $X_2$: 4
- RED $Y_2$: 6

- GREEN 171010
- GREEN $Y$: 2
- GREEN $Y$: 4
- GREEN $Y$: 6

- BLUE 171020
- BLUE $Y$: 2
- BLUE $Y$: 4
- BLUE $Y$: 6
4.2 WINDOW FUNCTION ALU

The x and y pulses of each of the three channels are input as operands to three ALUs. These sections control the way the inside and outside are derived from the X Y pulse pair. The output of the ALUs are used as a key which switches between two inputs, or one input and black.
### 4.3 WINDOW MODULE PIN ASSIGNMENTS

#### A AND B INPUT CONNECTOR PIN ASSIGNMENTS

<table>
<thead>
<tr>
<th>PIN</th>
<th>SIGNAL</th>
<th>PIN</th>
<th>SIGNAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RED0 INPUT</td>
<td>9</td>
<td>BLUE0 INPUT</td>
</tr>
<tr>
<td>2</td>
<td>RED1</td>
<td>10</td>
<td>BLUE1</td>
</tr>
<tr>
<td>3</td>
<td>RED2</td>
<td>11</td>
<td>BLUE2</td>
</tr>
<tr>
<td>4</td>
<td>RED3</td>
<td>12</td>
<td>BLUE3</td>
</tr>
<tr>
<td>5</td>
<td>GREEN0 INPUT</td>
<td>6</td>
<td>GREEN1</td>
</tr>
<tr>
<td>7</td>
<td>GREEN2</td>
<td>8</td>
<td>GREEN3</td>
</tr>
</tbody>
</table>

#### OUTPUT CONNECTOR PIN ASSIGNMENTS

<table>
<thead>
<tr>
<th>PIN</th>
<th>SIGNAL</th>
<th>PIN</th>
<th>SIGNAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RED0 OUTPUT</td>
<td>9</td>
<td>BLUE0 OUTPUT</td>
</tr>
<tr>
<td>2</td>
<td>RED1</td>
<td>10</td>
<td>BLUE1</td>
</tr>
<tr>
<td>3</td>
<td>RED2</td>
<td>11</td>
<td>BLUE2</td>
</tr>
<tr>
<td>4</td>
<td>RED3</td>
<td>12</td>
<td>BLUE3</td>
</tr>
<tr>
<td>5</td>
<td>GREEN0 OUTPUT</td>
<td>6</td>
<td>GREEN1</td>
</tr>
<tr>
<td>7</td>
<td>GREEN2</td>
<td>8</td>
<td>GREEN3</td>
</tr>
</tbody>
</table>
5.0 ALU MODULE

This module is based around three 4-bit wide ALUs. They are arranged as 3 independent channels and are designated Red, Green, and Blue. The Arithmetic Logic Unit performs mathematical operations on 2 operands: A and B. Software can select any one of 64 operations for each ALU, as well as selecting one of two sources as the A operand, and one of four sources as the B operand.

5.1 ALU CONTROL REGISTERS - 171000 RED
171004 GREEN
171010 BLUE
Bit 10 selects the source of the A operand. Bits 8 and 9 select the source of the P operand, while bits 0 through 5 control the function performed on the A and P operands by the ALU.

The A operand comes from one of 2 sources; the color group input comes from the A input connector. Each color is associated with the ALU of that color. The other possible source for the A operand is from the Numeric Data Register. There is one register for each of the 3 ALUs. This way software provides the value for the A operand to the ALU.

The P operand can come from one of 4 sources. 3 are any one color group from the A input connector. The fourth possible input is from the external connector.
5.3 ALU PIN ASSIGNMENTS

A OPERAND INPUT CONNECTOR

<table>
<thead>
<tr>
<th>PIN</th>
<th>SIGNAL</th>
<th>PIN</th>
<th>SIGNAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RED0 INPUT</td>
<td>9</td>
<td>BLUE0 INPUT</td>
</tr>
<tr>
<td>2</td>
<td>RED1 &quot;</td>
<td>10</td>
<td>BLUE1 &quot;</td>
</tr>
<tr>
<td>3</td>
<td>RED2 &quot;</td>
<td>11</td>
<td>BLUE2 &quot;</td>
</tr>
<tr>
<td>4</td>
<td>RED3 &quot;</td>
<td>12</td>
<td>BLUE3 &quot;</td>
</tr>
<tr>
<td>5</td>
<td>GREEN0 INPUT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>GREEN1 &quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>GREEN2 &quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>GREEN3 &quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

EXTERNAL OPERAND INPUT CONNECTOR

<table>
<thead>
<tr>
<th>PIN</th>
<th>SIGNAL</th>
<th>PIN</th>
<th>SIGNAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>REXT0 TO RED ALU</td>
<td>9</td>
<td>REXT0 TO BLUE ALU</td>
</tr>
<tr>
<td>2</td>
<td>REXT1 &quot;</td>
<td>10</td>
<td>REXT1 &quot;</td>
</tr>
<tr>
<td>3</td>
<td>REXT2 2</td>
<td>11</td>
<td>REXT2 &quot;</td>
</tr>
<tr>
<td>4</td>
<td>REXT3 &quot;</td>
<td>12</td>
<td>REXT3 &quot;</td>
</tr>
<tr>
<td>5</td>
<td>GEXT0 TO GREEN ALU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>GEXT1 &quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>GEXT2 &quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>GEXT3 &quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
This module is based around a 64 by 16 bit array of high-speed RAM (read-write memory). Software can control the source of data and address for the RAM, including loading the data from memory data registers which are accessible to the LSI-11. In addition, a 'write then display' mode can be selected whereby the contents of the high-speed RAM are updated automatically during vertical interval, and then switched to display during the active field.

6.1 MEMORY CONTROL REGISTER - 171170

One of 2 sources may be selected as the data source for the high-speed RAM. If bit 3 in the Memory Control Register is cleared to 0, the RAM will receive data from the S&H/LSI-11 buffer memory, that is the Memory Data Registers. These registers appear to the LSI-11 as write only; the contents of the RAM cannot be read back by software. If bit 3 of the Memory Control Register is set to 1, the RAM will receive data from the external data connector.
If bits 1 and 2 of the Memory Control Register are set to 0 (00 binary), the address information comes from the S&M/LSI-11 buffer memory, which is active during vertical interval. Therefore, if this source is selected, (and the write-then-display bit is 0), the output of the RAM during the active field will be the contents of the highest address of RAM. This is because when the S&M processor interface performs transfer during the vertical interval, it sweeps up through all the addresses from lowest to highest. And the last address presented to the high-speed RAM is the highest one; that which corresponds to the memory data register addressed at 171370.

When the address source select bits are set to 2 (10 binary), the address source is from external connector 0, and when the address source select bits are set to 3 (11 binary), from external connector 1.

6.2 MEMORY SHIFT REGISTER - 171172

If the address source select bits are set to 1 (01 binary), the address source will be an X Y counter, similar to the one in the X Y counter module. When its source is selected, the memory will be scanned out sequentially during the active field as an array of 8 horizontal by 8 vertical elements, each 16 bits deep. The starting address of the memory X Y counter may be set by software via the Memory Shift Register. The memory will be read out starting at the address specified in this register, wrapping around to address 0 when reaching address 63.

6.3 WRITE THEN DISPLAY

Bit 4 of the memory control register, when set to 1, selects the write-then-display feature. Conceptually, this bit ORs vertical blanking with the lower bit of the address source select (bit 1). This then causes the memory to switch its address from one source during vertical blanking (write), to another source during the active field (then display). So when this feature is enabled, the memory will switch between either S&M bus and X Y, or between external 1 and external 2. If the former pair are selected (address bits = 00, W.D. Bit = 1) then the RAM will be updated from the Memory Data Registers during vertical blanking, and then displayed in the 3 x 9 format during the active field.
6. MEMORY MODULE PIN ASSIGNMENTS

EXTERNAL 0 & 1 CONNECTORS

<table>
<thead>
<tr>
<th>PIN</th>
<th>SIGNAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ADDR0 LSB</td>
</tr>
<tr>
<td>2</td>
<td>ADDR1</td>
</tr>
<tr>
<td>3</td>
<td>ADDR2</td>
</tr>
<tr>
<td>4</td>
<td>ADDR3</td>
</tr>
<tr>
<td>5</td>
<td>ADDR4</td>
</tr>
<tr>
<td>6</td>
<td>ADDR5 MSB</td>
</tr>
</tbody>
</table>

EXTERNAL DATA INPUT CONNECTOR

<table>
<thead>
<tr>
<th>PIN</th>
<th>SIGNAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DINO LSB</td>
</tr>
<tr>
<td>2</td>
<td>DIN1</td>
</tr>
<tr>
<td>3</td>
<td>DIN2</td>
</tr>
<tr>
<td>4</td>
<td>DIN3</td>
</tr>
<tr>
<td>5</td>
<td>DIN4</td>
</tr>
<tr>
<td>6</td>
<td>DIN5</td>
</tr>
<tr>
<td>7</td>
<td>DIN6</td>
</tr>
<tr>
<td>8</td>
<td>DIN7</td>
</tr>
</tbody>
</table>

DATA OUTPUT CONNECTOR

<table>
<thead>
<tr>
<th>PIN</th>
<th>SIGNAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DOUT0 LSB</td>
</tr>
<tr>
<td>2</td>
<td>DOUT1</td>
</tr>
<tr>
<td>3</td>
<td>DOUT2</td>
</tr>
<tr>
<td>4</td>
<td>DOUT3</td>
</tr>
<tr>
<td>5</td>
<td>DOUT4</td>
</tr>
<tr>
<td>6</td>
<td>DOUT5</td>
</tr>
<tr>
<td>7</td>
<td>DOUT6</td>
</tr>
<tr>
<td>8</td>
<td>DOUT7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PIN</th>
<th>SIGNAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>DOUT8</td>
</tr>
<tr>
<td>10</td>
<td>DOUT9</td>
</tr>
<tr>
<td>11</td>
<td>DOUT10</td>
</tr>
<tr>
<td>12</td>
<td>DOUT11</td>
</tr>
<tr>
<td>13</td>
<td>DOUT12</td>
</tr>
<tr>
<td>14</td>
<td>DOUT13</td>
</tr>
<tr>
<td>15</td>
<td>DOUT14</td>
</tr>
<tr>
<td>16</td>
<td>DOUT15 MSB</td>
</tr>
</tbody>
</table>
7.0 EXPORT/IMPORT MODULE

This module facilitates transfer of parallel data to and from external devices and/or other modules. The module supports 4 16-bit output ports and 2 16-bit input ports.

7.1 EXPORT IMPORT DATA REGISTERS - 171070 EXPORT0
171072 EXPORT1
171074 EXPORT2
171076 EXPORT3

171600 IMPORT0
171602 IMPORT1

Each port is associated with a data register. As with all the modules, the inputs and outputs are Low Power Schottky TTL, and all rules for this type of signal apply. The outputs will drive any reasonable number of L S TTL inputs; but no two outputs should be tied together.

7.2 EXPORT/IMPORT PIN ASSIGNMENTS

EXPORT CONNECTORS 0 - 3

<table>
<thead>
<tr>
<th>PIN</th>
<th>SIGNAL</th>
<th>PIN</th>
<th>SIGNAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>OUT0 LSB</td>
<td>9</td>
<td>OUT8</td>
</tr>
<tr>
<td>2</td>
<td>OUT1</td>
<td>10</td>
<td>OUT9</td>
</tr>
<tr>
<td>3</td>
<td>OUT2</td>
<td>11</td>
<td>OUT10</td>
</tr>
<tr>
<td>4</td>
<td>OUT3</td>
<td>12</td>
<td>OUT11</td>
</tr>
<tr>
<td>5</td>
<td>OUT4</td>
<td>13</td>
<td>OUT12</td>
</tr>
<tr>
<td>6</td>
<td>OUT5</td>
<td>14</td>
<td>OUT13</td>
</tr>
<tr>
<td>7</td>
<td>OUT6</td>
<td>15</td>
<td>OUT14</td>
</tr>
<tr>
<td>2</td>
<td>OUT7</td>
<td>16</td>
<td>OUT15 MSB</td>
</tr>
</tbody>
</table>

IMPORT CONNECTORS

<table>
<thead>
<tr>
<th>PIN</th>
<th>SIGNAL</th>
<th>PIN</th>
<th>SIGNAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IN0 LSB</td>
<td>9</td>
<td>IN8</td>
</tr>
<tr>
<td>2</td>
<td>IN1</td>
<td>10</td>
<td>IN9</td>
</tr>
<tr>
<td>3</td>
<td>IN2</td>
<td>11</td>
<td>IN10</td>
</tr>
<tr>
<td>4</td>
<td>IN3</td>
<td>12</td>
<td>IN11</td>
</tr>
<tr>
<td>5</td>
<td>IN4</td>
<td>13</td>
<td>IN12</td>
</tr>
<tr>
<td>6</td>
<td>IN5</td>
<td>14</td>
<td>IN13</td>
</tr>
<tr>
<td>7</td>
<td>IN6</td>
<td>15</td>
<td>IN14</td>
</tr>
<tr>
<td>2</td>
<td>IN7</td>
<td>16</td>
<td>IN15 MSB</td>
</tr>
</tbody>
</table>
### 8.0 S&H BUS REGISTERS AND MNEMONICS

#### 8.1 WINDOW MODULE -
- 171000 REDX1 (WINDOW EDGE POSITIONING)
- 171002 REDY1
- 171004 REDX2
- 171006 REDY2
- 171010 GRENX1
- 171012 GRENY1
- 171014 GREX2
- 171016 GREY2
- 171020 BLUEX1
- 171022 BLUEY1
- 171024 BLUEX2
- 171026 BLUEY2
- 171030 WALUR (WINDOW FUNCTION ALU'S)
- 171032 WALUG
- 171034 WALUB

#### 8.2 DATA SELECTOR MODULE -
- 171040 REDSEL (DATA SELECTOR SOURCE SELECT)
- 171042 GRSEL
- 171044 BLUSEL
- 171046 SELINV (INVERSION REGISTER)

#### 8.3 X Y COUNTER -
- 171050 SHIFT
- 171052 ZOOM (X Y COUNTER INITIAL VALUE)

#### 8.4 ALU MODULE -
- 171100 ALURED (ALU FUNCTION-INPUT SELECT)
- 171102 NUMRED
- 171104 ALUGRN
- 171106 NUMGRN
- 171110 ALUBLU
- 171102 NUMELU

#### 8.5 HIGH SPEED MEMORY MODULE -
- 171170 MEMSTA (MEMORY STATUS)
- 171172 MEMSHF (MEMORY DISPLAY SHIFT)
- 171200 MEMORY
- 171376 MEMORY

#### 8.6 EXPORT/IMPORT MODULE -
- 171070 EXPRTO (PARALLEL OUTPUT PORT)
- 171072 EXPR1
- 171074 EXPR2
- 171076 EXPR3
- 171600 IMPRT0 (PARALLEL INPUT PORT)
- 171602 IMPRT1
- 171606 INDEX (FIELD ODD-EVEN INDEX)

#### 8.7 S&H BUS CONTROL -
- 171776 STATUS (S&H BUS STATUS AND CONTROL)
The RGB outputs (from one) go to the processor. The H_in is the horizontal drive from the color proc.

H_s is the horizontal sample, should be the same freq as H_s - (Not used normally)

X_in is an input for the Dot Clock - Not normal.

X_out is the Xclk - dot clock that goes to the McArtur.

RGB in goes to the McArtur.
Boots on power up - BS @ 173000
Left drive is dyo (ordko)
Right is dy1 (ordk1)
emulates on RX02

RT-11 V5.00 has a help function.
Just type help for instructions.

From ODT reboot with 1730006

**Hardware Config**

<table>
<thead>
<tr>
<th></th>
<th>1923cpu M8156</th>
<th>D5D440 interface 804932</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Terminal card SLU 7940</td>
<td>memory M8044DF</td>
</tr>
<tr>
<td>3</td>
<td>Custom McArtur card</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Terminal cable has
Disk cable has top written on it and its keyed.
McArtur cable has top on it also
Jeffrey Schier  
3928 Shafter  
Oakland, Ca. 94609  
(415)-653-5825

Education:  BSEE 1978, State University of N.Y. Buffalo

Work Experience:
1987-1989: Pinnacle Systems
  Project Manager - developed PriZm digital video effects system (DVE). Defined architecture and coordinated implementation from prototype through production. The DVE performs real-time rotation, perspective transformations, and curved/warping of the image; with the aid of a color menu interface. The DVE was integrated within a Video Workstation line merging still-storage, digital video effects, and paint/3D animation. System processing was based upon an 80286 user interface, with an 80386/80387 coordinate processor. The (4:2:2:4) pixel pipeline includes adjustable FIR filters, random access transformation memory, and table driven address generation. Designed and implemented the transformation memory. Logic design and PCB layout centered around 80386 computers using PCAD, ORCAD, CUPL, ABEL, Microsoft C and Assembler under MS-DOS. ASIC development using Xilinx and Actel chips. Application is in industrial/broadcast video.

1985-1987: Aurora Systems
  Senior R+D Engineer - Development of multi-port image memory, with bit-slice controller for a 32 bit, full color "AU280 Paint System". The frame buffers operate at video rates, for real-time image processing on multiple data streams. The image subsystem lives on a VME bus, driven by a SUN 3/68020 computer. Hardware development was centered around a Valid Scalr workstation, with PAL/PROM logic programming/simulation using ABEL. Application is in broadcast television and commercial computer graphics.

1983-1985: Cubicomp Corp.
  Senior Research Engineer - Hardware/software development of 3-D Solid Modeling system running on IBM PC/PC-AT. New product definition, hardware/diagnostics, Gen-Lock/Video Compatibility, Color Encoder/Keyer, Matrix OCR Support, Inkjet Printer software drivers.

1980-1983 Grinnell Systems
  Senior Research Engineer: Lead design of GMR2800 Computer Graphics Image Processing system. System is bit-slice (AMD2900) based, with memory feedback pipeline for image processing. Microcode compiler/assembler developed in-house. Implementation of sync timing, gen-lock, Flash A/D Converter, D/A Converter, EIA Sync Drivers, Multi-mode computer interfaces (DEC Unibus and Q-bus).

1979-1980 Nicolet-Cad Corporation
  Project Leader for Color graphics CAD Workstation applied to PC board design. Design of overlay card, sync/timing generation, Image memory, Bit-slice graphics processor enhancements.

1976-1979 Vasulka Corporation

Programming Languages: C, Fortran, Assembler, Microcode  
Operating Systems: UNIX, MS-DOS, DEC RT11/RSX11M

Video Workshops: S.U.N.Y. Buffalo, Media Study Buffalo

Papers: "Merging of 2D and 3D effects", SMPTE 1988 conference N.Y.  
  "Video Architectures - Approaching Real Time", Wescon/Northcon 1985  
  IEEE C.G+A, Feb. 1986 - "Cherries in Needlepoint Texture"

Member of SMPTE and ACM/SIGGRAPH

References furnished upon request.