



# A High Performance Colour Graphics Display System

Chong he Fang, B.E.

A thesis submitted for the degree of  
Master of Science  
in the Department of Computer Science of  
the University of Adelaide.

November 1987

Awarded: Jan 15th, 1988.

# Contents

<b>List of Figures</b>	<b>iii</b>
<b>Summary</b>	<b>vi</b>
<b>Declaration</b>	<b>vii</b>
<b>Acknowledgements</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 The display system of a workstation . . . . .	1
1.1.1 The significance of the graphics display system . . . . .	1
1.1.2 The raster graphics display system . . . . .	3
1.1.3 The advantages and disadvantages of the raster graphics display . .	5
1.1.4 The frame buffer . . . . .	5
1.1.5 Basic graphics capabilities . . . . .	8
1.2 This thesis . . . . .	12
1.2.1 Motivation . . . . .	12
1.2.2 Hypothesis . . . . .	13
1.2.3 The remainder of this thesis . . . . .	14
<b>2 Implementation</b>	<b>16</b>
2.1 Architectural features of workstation display systems . . . . .	16
2.1.1 RasterOp model . . . . .	16
2.1.2 Parallel architectures . . . . .	19
2.1.3 Peripheral and integral display system architectures . . . . .	22
2.2 The implemented display system . . . . .	24
2.2.1 General architecture of the host workstation . . . . .	25
2.2.2 The display subsystem . . . . .	26
2.3 The multi-mode frame buffer . . . . .	31
2.3.1 Frame buffer updating in a multi-window environment . . . . .	31
2.3.2 The screen format and frame buffer organization . . . . .	33
2.3.3 Multiple functionality modes and their data structures . . . . .	39
2.3.4 Other multi-mode frame buffers . . . . .	43
2.3.5 The design of the multi-mode frame buffer array . . . . .	44
2.3.6 The display controller . . . . .	56
2.3.7 The system interface . . . . .	61

<b>3</b>	<b>Virtual frame buffer</b>	<b>63</b>
3.1	The virtual frame buffer scheme . . . . .	63
3.1.1	The motivation for using a virtual frame buffer . . . . .	63
3.1.2	Ordinary paging virtual memory systems . . . . .	64
3.1.3	The difficulties of implementing a virtual colour frame buffer . . . . .	66
3.1.4	The solution – the page group concept . . . . .	68
3.1.5	Address translation for the multi-mode frame buffer . . . . .	70
3.2	A virtual frame buffer management simulator . . . . .	74
3.2.1	Introduction . . . . .	74
3.2.2	The address translator and its data structures . . . . .	76
3.2.3	Frame buffer resource management – the frame buffer configurator . . . . .	81
3.2.4	The paging handler and related data structures . . . . .	84
3.2.5	Sharing of the virtual frame buffer address space . . . . .	96
3.2.6	Experimentation . . . . .	105
<b>4</b>	<b>Using the display system</b>	<b>106</b>
4.1	The programming model of the display system . . . . .	106
4.1.1	Overview . . . . .	106
4.1.2	The co-ordinate system . . . . .	108
4.2	Programming the display subsystem . . . . .	112
4.2.1	Basic drawing procedures . . . . .	112
4.2.2	RasterOp mode operation . . . . .	113
4.2.3	Pixel mode operation . . . . .	123
4.2.4	Bit-plane mode operation . . . . .	132
4.2.5	Off-screen buffer management . . . . .	134
4.2.6	Multiprocessor environments . . . . .	138
4.3	Experimentation and performance issues . . . . .	140
4.3.1	Performance estimation . . . . .	140
4.3.2	Experimentation . . . . .	142
<b>5</b>	<b>Conclusions</b>	<b>151</b>
5.1	Concluding remarks . . . . .	151
5.2	Further work . . . . .	155
<b>A</b>	<b>Address mapping</b>	<b>158</b>
<b>B</b>	<b>Virtual frame buffer algorithms</b>	<b>160</b>
	<b>Bibliography</b>	<b>173</b>

# List of Figures

1.1	A computer graphics system. . . . .	3
1.2	A raster image. . . . .	4
1.3	A raster graphics display system. . . . .	5
1.4	Models of the frame buffer. . . . .	6
1.5	The raster operation. . . . .	9
2.1	The PERQ workstation. . . . .	17
2.2	The Sun-2 workstation. . . . .	18
2.3	The Iris workstation. . . . .	20
2.4	A cellular array processor architecture. . . . .	22
2.5	A peripheral display system. . . . .	23
2.6	An integral display system. . . . .	24
2.7	The architecture of the host workstation. . . . .	25
2.8	The architecture of the implemented graphics display system. . . . .	29
2.9	Block diagram of a video RAM chip. . . . .	35
2.10	The organization of a frame buffer bit-plane. . . . .	36
2.11	Frame buffer address and data formats. . . . .	41
2.12	Basic arrangement of the frame buffer memory. . . . .	45
2.13	The multi-mode memory data path. . . . .	47
2.14	Procedure RasterOp. . . . .	49
2.15	Copying one primitive image rectangle. . . . .	51
2.16	RasterOp in a bit-plane organized memory. . . . .	52
2.17	The block diagram for the BLT chip. . . . .	53
2.18	Functional partition of the display controller. . . . .	57
2.19	The updating controller. . . . .	58
2.20	The refresh controller. . . . .	60
2.21	Block diagram of the system interface. . . . .	62
3.1	The composition of an address in a paging virtual memory system. . . . .	64
3.2	The address translation. . . . .	65
3.3	The address translation algorithm. . . . .	66
3.4	A stack of binary images forms a colour image. . . . .	69
3.5	The page group concept. . . . .	69
3.6	Fields of the frame buffer address. . . . .	71
3.7	Frame buffer address format. . . . .	72
3.8	The NS32000 address translation scheme. . . . .	77
3.9	The structure of PT1 and PT2 for the simulator. . . . .	79

3.10	Procedure Address_translator. . . . .	80
3.11	The page frame list data structure. . . . .	83
3.12	Partitioning the physical frame buffer. . . . .	83
3.13	Overview of the virtual frame buffer scheme. . . . .	86
3.14	Procedure Pager. . . . .	87
3.15	Procedure Get_plane_mode_PT1_entry. . . . .	87
3.16	Procedure Look_up_plane_mode_PT2. . . . .	88
3.17	Procedure Call_in_page_group. . . . .	90
3.18	The PFN database. . . . .	91
3.19	The working set list. . . . .	92
3.20	A global entry and its relation to the other data structures. . . . .	99
3.21	Procedure Check_global. . . . .	100
4.1	A programming model of the display system. . . . .	107
4.2	A pixel in a linearly addressed raster storage. . . . .	110
4.3	A “form” representation of image storage. . . . .	110
4.4	New “form” representations of raster storage. . . . .	111
4.5	The paste algorithm. . . . .	116
4.6	Copying a 16-pixel segment using RasterOp mode. . . . .	117
4.7	Copying a 16-pixel segment in pixel mode. . . . .	117
4.8	Copying a 16-pixel segment in bit-plane. . . . .	118
4.9	Copying an image in the middle of a horizontal line. . . . .	119
4.10	Performance comparison for copying a 16-pixel segment, with a pixel-depth of eight. . . . .	120
4.11	Procedure RasterOp_BLT. . . . .	121
4.12	RasterOp with transformation. . . . .	123
4.13	The pixel pointer type. . . . .	124
4.14	Procedure Line_relative. . . . .	126
4.15	Drawing an antialiasing line. . . . .	126
4.16	Procedure Smooth_line. . . . .	127
4.17	Procedure RasterOp_Pixel. . . . .	129
4.18	Procedure Maximum. . . . .	131
4.19	The data structure for frame buffer heap. . . . .	136
4.20	Procedure NewForm. . . . .	137
4.21	Calculating frame buffer base address for the three modes. . . . .	137
4.22	The sequence of frame buffer operation. . . . .	141
4.23	Estimation of the performance of the frame buffer. . . . .	143
4.24	The prototype hardware. . . . .	144
4.25	Examples of basic graphics operations. . . . .	145
4.26	Example of concurrent multi-mode operation. . . . .	146
4.27	The frame buffer data transfer cycle time. . . . .	147
4.28	The system bus utilization for various graphics operations. . . . .	148
4.29	Measurements of the image updating speed. . . . .	149
A.1	Address mapping for the display system. . . . .	158
A.2	Mapping for BLT register select bits. . . . .	159
A.3	Mapping for VSC register select bits. . . . .	159

B.1	Procedure Derive . . . . .	160
B.2	Procedure Call_in_block. . . . .	161
B.3	Procedure Put_into_working_set. . . . .	161
B.4	Procedure Evict_page_group. . . . .	162
B.5	Procedure Evict_3_mode. . . . .	163
B.6	Procedure Get_page_frame_back. . . . .	164
B.7	Additional actions for procedure Call_in_block. . . . .	165
B.8	Procedure Create_shared_area. . . . .	166
B.9	Procedure Create_shared_table. . . . .	167
B.10	Procedure Create_shared_entry. . . . .	168
B.11	Procedure Map_to_global. . . . .	169
B.12	Procedure Derive_shared_entry. . . . .	169
B.13	Procedure Delete_old_region. . . . .	170
B.14	Procedure Delete_entry. . . . .	171
B.15	Procedure Delete_different_group. . . . .	172

# Summary

A high performance colour graphics display system plays an important role in the man-machine interface of a computer workstation. With rapid progress in the technology of TV monitors and the reducing cost of frame buffer memory, the raster graphics display is becoming predominant in the graphics display field. The advantage of the raster display is that because the brightness and colour of each picture element can be specified independently, any picture can be conveniently displayed with comparatively low cost. The main difficulty of the high performance raster graphics display is that a great many bits in the frame buffer must be modified to make major changes to the picture. Therefore, the capability of rapidly updating the frame buffer is one of the most important properties of a raster graphics display system.

This thesis describes the design of a high resolution colour graphics display system for a shared-memory 32-bit multiprocessor workstation. This display system makes picture creation and rearrangement simple and rapid by introducing a specially structured multiple functionality mode frame buffer. This multi-mode frame buffer supports fast raster operations, flexible picture element manipulation, a virtual frame buffer architecture and multiprocessor parallel picture updating in the frame buffer. This system has been designed as a hardware testbed for experimentation with various graphics applications and for the display of multiple overlapped active windows.

A virtual frame buffer simulator is presented to show a scheme which enables the multi-mode colour frame buffer to be a demand-paged virtual frame buffer. This not only enlarges the frame buffer space, which is essential for the display of active multiple overlapped windows and the panning of very large images, but also facilitates the management of image storage and reinforces security.

An experimental hardware display system has been built, and basic graphics operations have been tested on the prototype. An analysis of the resulting performance is presented to show the appropriateness of this display system architecture and to indicate suitable directions for further improvement.

# Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma in any University, and, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

I hereby consent to the thesis being made available for photocopying and loan if it is accepted for the award of the degree.

Chong he Fang

25th November 1987



# Acknowledgements

I have been extremely fortunate in having the help of many people during the course of this project. I would first like to thank Prof. Christopher Barter and Dr. Christopher Marlin for their constant encouragement, inspiration and direction throughout the whole project.

I would also like to thank Peter Hawryskiewicz, Francis Vaughan, Peter Ashenden, and Peter Fife for numerous stimulating discussions. Werner Dorfl, Tony Romano, and Peter Daly have contributed a great deal to the building of the prototype hardware; without their efforts, I would never have obtained a running display system. Quentron Optics have generously given me the use of their QDS-1000 processor board and software environment for my experiments.

Finally, I would like to thank my wife Ching-hua who has acted out of love and grace, and not a sense of justice.

This thesis was completed with the assistance of a scholarship under the University Research Grant scheme of the University of Adelaide.