Technical Note QD16
8*24 GC QuickDraw and Deaccelerated CopyBits

CONTENTS

This Technical Note discusses conditions that may cause `_CopyBits` to slow down when QuickDraw acceleration is on via the Apple 8*24 GC Display Card.

[Feb 01 1991]

Introduction

When a drawing call is issued, GC IPC (Interprocess Communication) takes control of the call and passes it to GC QuickDraw. After the normal port set up (which involves caching the port parameters if this is the first drawing call after the port was set), GC QuickDraw returns control to the application through the IPC and performs, in parallel, the drawing to its own monitor as well as any other monitors that may be affected by the operation. The application then continues its execution, probably issuing more drawing calls that get executed in the same asynchronous manner.

The result of this mode of operation is improved performance, since the application gets control back immediately after issuing the call and the GC QuickDraw moves video data to its own video buffer as well as that of other cards in a more rapid manner by using block transfers and without requiring any action by the main processor.

Back to top

_CopyBits Conforms To The Same Scheme, Except...

`_CopyBits` conforms to the same operational scheme, but there are some instances in which GC QuickDraw cannot perform the call in parallel; in this cases it is even possible to suffer a performance loss, since the whole call may have to be completed before control is given back to the application and GC QuickDraw has to make calls and access data across the NuBus.

The_situations that compromise GC QuickDraw parallel operation are as follows:

- When the destination device has a `SearchProc` installed and the source color environment is different from the destination environment.

QuickDraw calls a `SearchProc` whenever the source and destination have different depths and when two indexed pixel maps have different color tables, even though their depths may be identical. When GC acceleration is enabled, these conditions cause the following two types of behavior, dependent upon the source pixel map:

- If the source is an indexed pixel map, then GC QuickDraw executes the part of the setup that involves calling the `SearchProc`, returns control to the main processor, then completes the call in parallel. The act of calling the `SearchProc` before returning control makes the call slower than when no `SearchProc` is involved, since parallel operation does not occur throughout the whole call.
- If the source is a direct RGB pixel map, then GC QuickDraw has to call the `SearchProc` for every pixel that is drawn, and the application does not regain control until after the call to `_CopyBits` has been completed.

- When the source or destination is offscreen and not created using a `GWorld`.

GC QuickDraw has no way to detect when an application is going to manipulate a pixel map it has created in memory, so if it has to draw to or copy from such a `PixMap`, GC QuickDraw has to complete the operation before returning control to the application.

This behavior is contrary to the case when using a `GWorld` for offscreen environments, since in the case of a `GWorld`, GC QuickDraw is alerted by the call to `_GetPixBaseAddr` that the application is getting ready to directly change the pixels. This is the reason why it is so important that applications call `_GetPixBaseAddr` **every** time they are about to manipulate a `GWorld` pixel map directly.

- When the source `PixMap` has a color table that uses indexes that refer to a palette.

QuickDraw now allows a color table to have indexes that point to entries in the palette associated with the destination window; when bit 14 in the `ctFlags` field is set, the value fields in the color table are treated as palette entries. When such a `PixMap` is the source for `_CopyBits`, then GC QuickDraw has to make a number of calls to the Palette Manager as part of the setup before returning control and completing the call.

This case is similar to that of a indexed pixel map when a `SearchProc` is involved; therefore, it only implies a partial loss of parallelism; it is good to keep in mind that this case can only occur when the source `PixMap` is indexed.

Back to top

References

Inside Macintosh , Volumes V & VI, Color QuickDraw

develop , "Macintosh Display Card 8*24 GC: The Naked Truth," July 1990

Technical Note #275, 32-Bit QuickDraw: Version 1.2 Features

Developer Notes for the Macintosh Display Cards 4*8, 8*24 and 8*24 GC

NuBus is a trademark of Texas Instruments.

Back to top

Downloadables

Acrobat version of this Note (K).   Download