P10®

Architecture Overview

DRAFT

PROPRIETARY AND CONFIDENTIAL INFORMATION

3Dlabs®

3Dlabs®

P10®

Architecture Overview

PROPRIETARY AND CONFIDENTIAL INFORMATION

Issue 1

Proprietary Notice

Email: info@3dlabs.com
Web: http://www.3dlabs.com

3Dlabs Ltd.
Meadlake Place
Thorpe Lea Road, Egham
Surrey, TW20 8HE
United Kingdom
Tel: +44 (0) 1784 470555
Fax: +44 (0) 1784 470699

3Dlabs K.K.
Shiroyama JT Mori Bldg 16F
40301 Toranomon
Minato-ku, Tokyo, 105, Japan
Tel: +81-3-5403-4653
Fax: +91-3-5403-4646

3Dlabs GmbH
Breckenheimer Weg 29
65205 Wiesbaden
Deutschland
Tel: +49 6122 916 778
Fax: +49 6122 919 646

3Dlabs Inc.
480 Potrero Avenue
Sunnyvale, CA 94086,
United States
Tel: +1 (408) 530-4700
Fax: +1 (408) 530-4701

Change History

Document	Issue	Date	Change
174.1.1 01	1	25/06/2001	Creation

Legend

@@@ indicates incomplete data

underlined coloured text indicates incomplete or unconfirmed data

Table of Contents

1
Introduction

1.1 Introduction

The Miranda P10 Graphics Processor is the first of a new series of 3D Processors with a highly scalable, multi-texture/multi-fragment per clock cycle architecture. This advanced design concept uses extensive parallelism and programmability to provide future-proof support for new, texture-intensive APIs such as Microsoft DX8.

Using ***programmable T&L*** and ***programmable pixel shaders*** in conjunction with highly optimised fixed-function units results in a simpler, faster and more flexible design.

Programmable registers also allow dynamic reconfiguration of the number of vertex shaders, the number of texture pipes and the number of rasterizers per chip to deliver the greatest possible throughput under changing task conditions.

Fixed-function registers for specialised tasks have been optimised for simplicity and speed with hand-polished main routines and the removal of legacy code. Memory bandwidth and DMA performance have been enhanced with support for high-density DDR memory configurations up to 8 x 8Mx32 devices and low overhead circular buffers to provide up to 17Gbytes/second peak throughput.

3Dlabs has achieved this without compromising its long-standing commitment to quality 3D rendering. P10 delivers accuracy, stability and full OpenGL compliance while providing a feature-rich device with unparalled real-world single-chip graphics performance.

1.2 Target Markets

Miranda P10's programmability and flexibility allow it to address an unusually wide range of market segments. Primary markets are:

- 3D Gaming
- High End Workstations

The following application areas are fully supported:

- CAD/CAM/CAE
- Animation
- Visualization/Simulation
- Custom embedding and IP options

1.3 Key Features and Platforms

The P10 ***basic feature set*** includes everything normally available on earlier devices plus:

- Up to 8 textures per fragment with any combination of trilinear, 3D, anisotropic filtering, bump mapping or cube mapping. True floating point coordinate generation.
- Programmable texture coordinate generation.
- Programmable shading unit (i.e. texture combiner).
- Programmable pixel unit.
- Accumulation buffering and convolution.
- T buffer full scene antialiasing.
- Integrated Geometry and Lighting.

I/O interfacing provision allows a full range of ***on- and off-board devices***:

- Analogue VGA
- Dual and Stereo heads
- DVI-I single link DFP and TV Encoder
- TV Out
- VIP2
- DVO
- I2C bus

Miranda P10 fully supports Intel's ***AGP 4X*** Accelerated Graphics Port standard, including:

- AGP4X
- AGP1X, AGP2X
- 33/66MHz PCI
- DMA and execute mode support
- Sideband addressing
- 3.3v/1.5v tolerant

P10 is compatible with most ***standard APIs*** including:

- OpenGL
- DX8/9 and DXVA
- XWindows
- NT4, Millenium, Win 9x and 2000 drivers are directly supported
- MacOS, Linux and other patforms are catered for

1.4 Design Performance

Performance estimates are based on design simulation rates pending availability of silicon-based test results. Primitive rates assume single tile coverage (reduced to 8x4 for z), Single directional light, Gouraud shaded, Depth buffered and .13 micron manufacturing. The feature set shown is in addition to features normally supported on earlier devices.

P10 Performance Overview[1]		
3Dmark (DX8)		**Bench-marks**
ProCDRS-03 (Workstation)	133	
Quake III Quincunx FSAA (OpenGL)		
Points, lines	75M lines/Sec.	**Prim-itives**
Triangles	75M lines/Sec.	
AA Lines	75M lines/Sec.	
Vertex rendering – no depth, texture or lighting	150M vertices/sec.	**Transform and Lighting**
Vertex rendering – with depth, not texture or lighting	132M vertices/sec.	
Vertex rendering – texture and fog, no lighting	106M vertices/sec.	
Scissor (core:memory)	19.2: - G/sec. (64 primitives/ cycle)	**Pixel Fill Rates**
32bpp Clear (core:memory)	4.8:4.25 G/sec.	
GID rejected (core:memory)	19.2:17 G/sec.	
Trilinear (core:memory, 32bpp, one texel/pixel read)	1.2 : 1.1G/s	
Peak Memory Bandwidth	17 GBytes/s	**Basic Features**
Max. memory	128Mbytes	
Operating Frequency (0.13 micron/.18 micron)	300MHz/200 MHz	
Up to 8 textures per primitive with any combination of trilinear, 3D, anisotropic filtering, bump mapping or cube mapping.	✔	
Programmable texture co-ordinate generation	✔	
Programmable shaders (i.e. texture combiners)	✔	
Programmable pixel unit	✔	
Accumulation buffering and convolution	✔	
Precomputed displacement maps and tesselation	✔	
T buffer full-scene antialiasing	✔	
Integrated geometry and lighting	✔	

Table 1.1 P10 Performance Overview

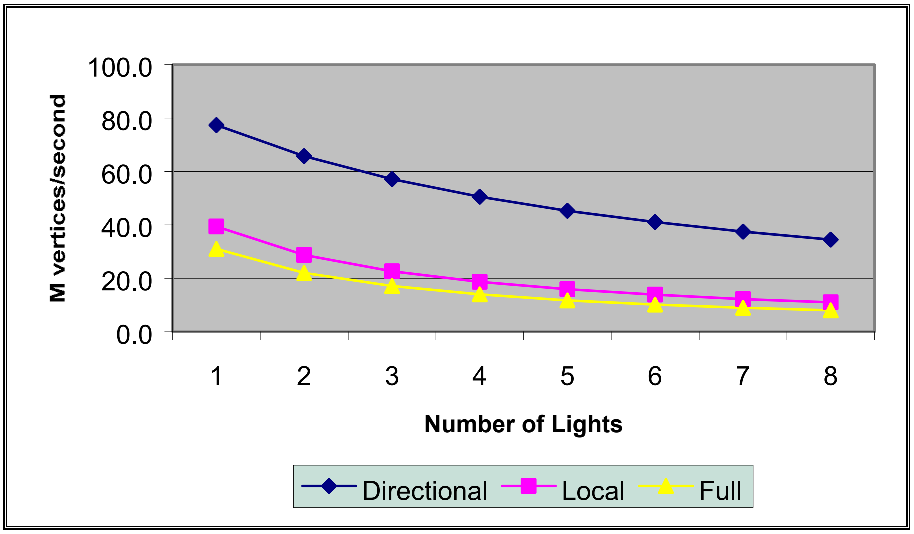


Table 1.3 Lighting Performance

1.5 Embedded Application Support Program

P10's highly flexible and compact design encourages embedded use for board, chip and IP solutions ranging from control and monitoring applications to real-time simulation, from medical imaging to test and training equipment, and more. The extensive programmability gives the ability to, for example, perform convolutions, radial gradient fills, even run the "Game of Life" on-chip.

To assist customers wishing to embed P10 in a proprietary environment, the 3Dlabs Embedded Support Program provides different IP and embedding options, full technical documentation, reference designs, technical support and, subject to the appropriate licensing, access to 3Dlabs driver source code and on-chip microcode.

On-chip programming is supported with assembler/disassembler pairs for each rogrammable unit. For high-level API development there are translators for DX8 and OpenGL to P10 source instructions, which include dead code removal, unused variable elimination, stall management, register coloring and other compiler techniques. Programs are assembled with a Dynamic Link Loader and downloaded to chip.

1.6 Changes from Earlier GLINT Devices

P10 introduces a radical new architecture, a host of new features and significant changes from earlier GLINT and Permedia devices. The following sections can provide only a brief summary of the new standard for fully integrated 3D on-chip functionality.

Legacy Rasterizer Chips	P10
Scanline Framebuffer	Tiled framebuffer
DDA based interpolators	Plane equations
Edge-walking rasterization	Tile-seeking rasterization
Multiple cycles per primitive	Multiple primitives per cycle
Fixed function units	Fixed/Programmable hybrid
FIFO-based memory	Cache-based memory
Asynchronous pipeline	Parallel pipes with pre-emption
Command and control data visits every unit	Command and control independent routing

Table 1.1 Miranda P10 Evolutionary Changes

1.6.1 Tile-based working

P10 adopts the tile as its sole unit of internal work. All operations are performed on 8x8 square screen-aligned ***planar byte pixel tiles*** similar to the 64x1 pixel spans used in earlier chips. All data types are stored the same way, so for example anything (e.g. the Depth buffer) can be a texture, and it is possible to render to a texture. Each memory access returns a planar byte tile.

Two or more accesses are used for pixel depths greater than 8 bits, which allows unusual formats such as ***24, 40 and 48 bpp***. All memory accesses are virtual and page faults are handled with a CPU-like page swap.

This uniformity results in tile scalability and substantial performance improvements, particularly in 3D and small 2D primitives (e.g. characters) where the improved scanline coherence and memory efficiencies are most noticeable. Performance is further enhanced by the use of 256-bit DDR memories running at 266MHz (peak bandwidth 17GB/s).

1.6.2 Multitasking

Architecture innovations include the Context unit, which implements ***pre-emptive multitasking*** to support time-critical operations such as render during frame blank. The Context unit caches context data and keeps a copy in local memory. A small cache handles frequently updated values such as mode registers.

When a context switch is needed the cache is flushed, the new context record is read from memory and the data converted into a message stream to update downstream units. Because only a small amount of cache data needs to be saved this process can be very fast – typically a ***context switch takes microseconds***. See "Isochronous Command Stream", section 1.7.1

1.6.3 Command Input

Unlike earlier graphics processors, P10 command and control data (register updates, mode changes etc.) does not generally take the same route as pixel data. This improves flexibility and bandwidth between units.

P10 uses two ***independent Command Units*** - one servicing the GP stream (for 3D and general 2D commands) the other servicing the Isochronous stream (for pre-emptive time-critical tasks). Both command units manage the Circular Buffers and Input DMA. The GP Command unit also manages Vertex Arrays.

1.6.3.1 Circular Buffers

Circular buffers, also new in P10, allow small packets of work to be transferred rapidly without the delays and overhead of setting up a DMA buffer and making an escape call to the O/S. Because DMA transfers take time to initiate they are normally optimized for large bursts of data to improve efficiency. This can result in the graphics system being idle while work accumulates in the DMA buffer, but not enough to trigger a burst.

Circular buffers are usually stored in local memory and mapped into the ICD. When commands and data are added to the circular buffers, chip-resident write pointer registers are updated accordingly (without any O/S intervention). When the current circular buffer goes empty the hardware automatically searches the pool of 16 circular buffers for more work and instigates a context switch if necessary.

Circular buffers process the command stream identically to input DMA and can call conventional DMA buffers.

1.6.3.2 Vertex Arrays and Vertex Caching for Indexed Arrays

Vertex arrays are supported for compactness and flexibility in data layout. An array element can hold up to 16 parameters, which can be stored consecutively in memory or held in arrays. Vertex elements can be accessed in sequence or using array indices. The most recent 16 array indices are cached to allow comparison with the current index to check for vertex meshing, which in turn allows ***substantial savings in memory reads*** and Shader processing.

1.6.4 Scalability

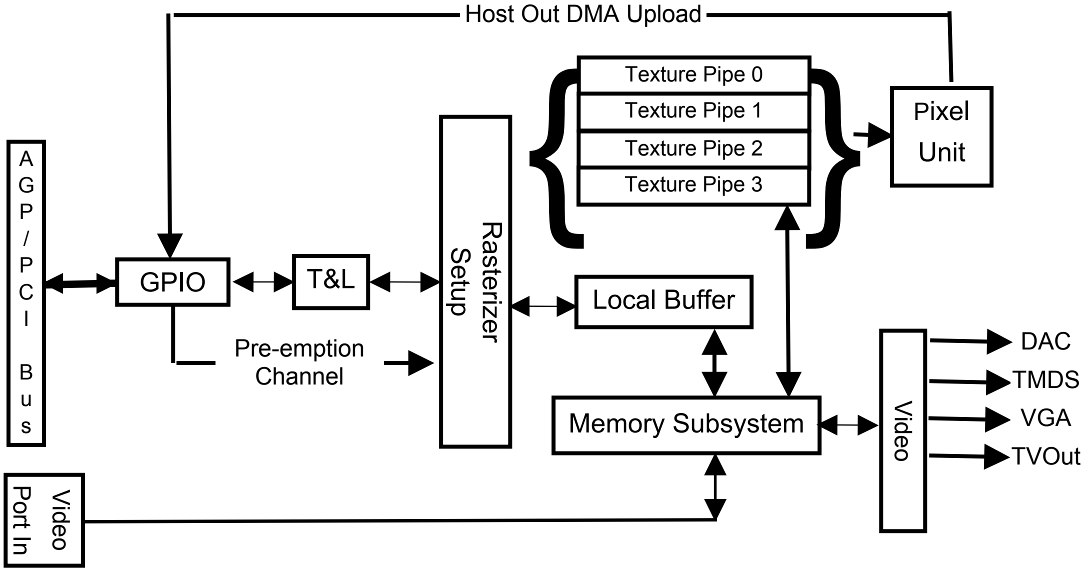
The design allows unusual flexibility in adapting performance to specific applications and to market targets as well as future proofing:

- Tile size can be varied
- the number of texture pipes and vertex shaders is configurable
- Changing the number of pipes and shaders does not affect the API
- Memory devices can be picked to suit market conditions (although 256bit DDRs are preferred).
- When a programmable register is idle it can be reprogrammed on the fly as an additional rasterizer to further improve fill and small primitive rates.

1.6.5 Legacy Support

Because of the design paradigm shift it has not been possible to continue support for many legacy items. This has incidentally removed up to 40% of the total code weight, which translates into a substantial reduction in gate count and chip complexity and a smaller, more flexible and faster design.

1.7 Chip Level Block Diagram



1.7.1 Isochronous Command Stream and Context Switching

Microsoft's 'hot button' for GDI+ establishes a new requirement for real-time processing slaved to the display state, to support tasks such as rendering during frame blank or non-tear bliting to a window.

P10 addresses this need by implementing a ***separate graphics core pre-emption*** channel which uses fast on-board context-switching (including switching during a primitive).

As context switchable state flows through into the rasterizer it goes through a Context Unit which snoops and caches the context data and keeps a local copy for context switches.

A second command queue handles real-time rendering commands using Video Timing Generator (VTG) and scanline timestamps. If the context switch is to allow isochronous rendering it invokes a small, dedicated isochronous stream rasterizer. A typical partial context switch to and from an isochronous context should take less than 700 cycles (3.5µs at 200MHz or ¼ scanline).

The Isochronous rasterizer only deals with rectangular primitives, which it can render in either direction. It is not a parallel blit engine – it is invoked only for Isochronous service requests using pre-empted processor capacity.

For more information, see the **Timestamp**, **Changeport** and **HoldPort** commands in the *Miranda P10 Reference Guide* volume III.

2

P10 Key Features

2.1 3D and 2D Graphics, MPEG and Other Features

P10 incorporates the following key functions in hardware to provide superior 3D, 2D and video resources:

2.2 3D Graphics

Supported Function	***Description***
Full primitive support	*Full primitive support: triangle lists, fans and strips. Line lists and strips. Point lists. All either aliased or anti-aliased.*
Efficient processing of small primitives	*Integrated set-up, backface cull calculation, low latency*
High fill rate	*Wide data paths, high performance memory*
Programable Shaders, programmable texture co-ordinate and pixel units.	
Textures	
Efficient texture storage	*Fully flexible formats, internal 256 entry LUT*
AGP textures	*Textures directly from AGP memory*
Dual/multi texture	*Single-pass multi-textures, up to 8 textures per primitive*
3D textures	*3D volumetric textures; trilinear, anisotropic filtered, bump, cube and displacement maps; tesselation*
High quality rendering	*Sub-pixel and sub-texel accurate*
High quality textures	*Accurate perspective correction and trilinear filtering with per pixel MIP-Mapping with true level of detail calculation.*
Lighting/Optical	
High quality lighting	*Interpolated diffuse and specular components*
Extremely realistic special effects	*Interpolated colored fog, fog table and depth-cueing*

Supported Function	***Description***
Translucent objects and sprites	*Blending/transparency on any primitive. Full dual texture blending. Interpolated alpha with direct support for all DirectX 6, 7 and OpenGL blend modes*
High quality texture cut-outs	*Color key with bilinear filter does not leave edge effects*
Anti-aliasing	*Edge anti-aliasing for zoomed sprites, full-scene T-buffer anti-aliasing*
Fast hidden surface elimination	*Depth (Z) buffering and non-linear Depth (Z) buffering. GID test for per pixel window clipping*
Fast shadow, fog and transparency effects	*Area stippling: vertex rendering with fog and texture at 106Mvertices/sec.*
Integrated Geometry and Lighting	*6 local lights at 20Mvertices/sec.*
High quality output at any color depth	*Dithering, programmable pixel formats*
Fast sprite handling	*Color key, scale, stretch, rotate, mirror*
Seamless integration of video and 3D	*Color key with depth test and perspective correction*
Minimize update area, target selection	*Hardware extent checking and picking*
Improved image quality at lower resolutions	*Full screen sort independent anti-aliasing*
Use of rendered images as textures	*Unified memory read and write to any buffer*
Full range of double buffer techniques	*Full screen flip, fast BLT, stereo buffers*
Virtual texture map management	*All memory is virtual/logical planar tiles, with cache-based page swapping.*

Table 2.1 3D Hardware Function Descriptions

2.3 2D Graphics

Supported Function	***Description***
Full primitive support	*Points, lines, spans, rectangles, polygons*
Efficient processing of small primitives	*Integrated set-up calculation, low latency*
Window clip	*Hardware rectangle clipping*
High speed color brushes	*Internal pattern RAM*
High speed monochrome brushes	*Internal stipple table*
Raster operations	*Logic op unit*
Fast BLTS	*512 bit internal data path*
Fast upload and download	*Run-length encoded data*
High speed monochrome download	

Supported Function	***Description***
Flexible font caching support	*Byte aligned monochrome bitmaps in local memory*
Color translation	*Through internal LUT*
High speed stretch BLT	*Using texture operations*
Overlays	*Per-pixel main image/overlay selection with color key and alpha blending*
Statistic collection	*Via dedicated StatisticMode register*
Border color	*Standard*
Context save and restore	*Cache-based context switch typically 3.5µs*

Table 2.2 2D Hardware Function Descriptions

2.4 MPEG2

Supported Function	***Description***
MPEG motion compensation	*Motion compensation calculations performed in hardware: user-programmable DXVA*
Support for software decoders	*DMA from system or write directly to local memory*
High speed color space conversion	
Flexible YUV data formats	*4:4:4, 4:2:2, 4:1:1 as standard and and user-programmable additions.*
Fast arbitrary stretch/shrink with filter	*Bilinear filter at any zoom/shrink factor*
Full featured video effects	*Scale, shrink, stretch, rotate, mirror*

Table 2.3 MPEG2 Functions

2.5 Power Management

Supported Function	**Description**
Clocks can be individually stopped	Separate clocks for: geometry processor, graphics processor, memory sub-system, video sub-system, video output and AGP
Automatic frequency reduction	Reduces average power consumption when idle
Memory power down mode	Low power while maintaining refresh and screen update
DPMS	Power management for monitors

Table 2.4 Power Management Functions

3

MIRANDA P10 Architecture

Miranda P10 architecture consists of an integrated geometry and rasterization pipeline described below, together with various interface features.

3.1 Host Interfaces - AGP/PCI

The P10 Bus Interface design includes a PCI Target, PCI Master, AGP Master, PCI Configuration Space registers, local Control and Status registers, and a DMA Arbiter to handle bus master requests from the various controllers within the P10 device. The interface conforms to the *PCI Local Bus Specification* Revision 2.2. and AGP Interface Specification Revision 2.0.

3.1.1 Signalling voltage

- 1.5V (1X, 2X, 4X)
- 3.3V (1X, 2X)
- 5V tolerant

3.1.2 PCI Interface

3.1.2.1 PCI Target features

- PCI Config Space transactions
- PCI Memory Space transactions
- PCI Fast Writes (2X and 4X)
- PCI I/O Space transactions
- VGA palette write snooping
- 32-bit and 64-bit addressing (dual address cycles)
- PCI multi-function operation

3.1.2.2 PCI Master features

- PCI Memory Space transactions
- 32-bit read and write data transfers
- 32-bit and 64-bit addressing (dual address cycles)

3.1.3 AGPBus

AGP 4X is Intel’s high performance, component level interconnect targeted at 3D display applications, which uses a 66MHz PCI specification as an operation baseline and provides significant performance extensions to the PCI specification.

Implementing these features enables P10 to achieve better than 1 GByte per second bandwidth from the host for instructions, textures, video data (limited by the host system throughput).

The add-in slot defined for AGP uses a connector body which is not compatible with the PCI connector. Boards designed for use in an AGP slot are not mechanically interchangeable with PCI boards. P10 supports AGP2x, AGP4x and PCI at signal voltages from 1.5vdc to 3.3vdc only. Legacy 5vdc PCI logic may severely damage the chip.

3.1.3.1 AGP Master features

- AGP low-priority Read transactions
- AGP low-priority Write transactions
- AGP Fence and Flush transactions
- Operation at 1X, 2X, and 4X data rates
- Sideband and pipe operation
- 48-bit addressing using sideband
- 64-bit addressing using pipe and dual address cycles

3.2 Integrated 2D/3D Processor – T&L and Graphics Pipeline

Miranda P10 includes Transformation and Lighting, Graphics Core, Context Switching and I/O support for a wide range of hardware configurations, all of which are tightly integrated by discrete core command, isochronous command and pixel streams.

The pipeline uses a hybrid mixture of programmable and dedicated units which allow the chip to be used either for brute force highly-parallel fragment processing or for complex multi-pass texture algorithms or effects (precomputed convolutions, tesselation, special effects e.g. Game of Life).

In addition, the configuration of the chip can be changed dynamically thanks to a context state cache. Together with an isochronous rectangle rasterizer P10 can respond to VTG event pre-emption in real time, typically as little as 3us for isochronous events or 20us for a full context switch.

The microcode instruction set, sequencer commands etc. are described in the *Miranda P10 Reference Guide* and *Programmer's Guide*. Assemblers/disassemblers and other microprogramming aids are also available for developers.

For further information on the functionality of the graphics processor and T&L refer to the *Reference Guide* volume III.

3.3 Memory Interface

P10 memory is cache-based and all data types are stored as 8bpp planar tiles. All memory access is logical/virtual and page faults cause CPU-like page swaps.

Memory is preferably 256 bit wide DDR devices running at 266MHz. From 32MB to 256MB of x32 devices are supported, or alternatively up to 512MB of x16 devices.[2] SDR devices are not supported.

[2] The additional address lines will somewhat constrain performance with x16 memories.

There are two independent 128bit controllers which hold alternating groups of 8 tiles. Memory is divided into 4 regions corresponding to the 4 internal banks of a DDR device:

Bank	Controller 0	Controller 1
0	0-7	8-15
1	16-23	24-31
2	32-39	40-47
3	48-55	56-63
0	64-71	72-79

Local memory is used to store color, depth, stencil, and texture data. These are largely interchangeable depending on the microcode application context. For more information on data typing and usage refer to the *Miranda P10 Programmers Guide*.

For more information on Memory devices and layouts see “Memory Systems” in the *Miranda P10 Reference Guide*.

3.4 SVGA

The on-chip SVGA unit is register-level compatible with standard VGA devices and requires no software emulation. It natively supports all standard VGA modes including the obsolete SVGA modes 100 and 101 (640x400 and 640x480). The VESA VBE extended modes shown below are supported using the Graphics Processor:

Table 1- 2 VESA VBE Graphics Modes

Mode (hex)	Pixels	Colors	Window-ed	Lin-ear	Support-able in SVGA	Support-able in GP
0x103	800x600	256	✓	✓	✕	✓
0x105	1024x768	256	✓	✓	✕	✓
0x107	1280x1024	256	✓	✓	✕	✓
0x109	320x200	32K (5:5:5:1)	✓	✓	✕	✓
0x10D	320x200	64K (5:6:5)	✓	✓	✕	✓
0x10F	320x200	16.8M (8:8:8)	✓	✓	✕	✓
0x110	640x480	32K (5:5:5:1)	✓	✓	✕	✓
0x111	640x480	64K (5:6:5)	✓	✓	✕	✓
0x112	640x480	16.8M (8:8:8)	✓	✓	✕	✓
0x113	800x600	32K (5:5:5:1)	✓	✓	✕	✓
0x114	800x600	64K (5:6:5)	✓	✓	✕	✓
0x115	800x600	16.8M (8:8:8)	✓	✓	✕	✓
0x116	1024x768	32K (5:5:5:1)	✓	✓	✕	✓
0x117	1024x768	64K (5:6:5)	✓	✓	✕	✓
0x118	1024x768	16.8M (8:8:8)	✓	✓	✕	✓
0x119	1280x1024	32K (5:5:5:1)	✓	✓	✕	✓
0x11A	1280x1024	64K (5:6:5)	✓	✓	✕	✓
0x11B	1280x1024	16.8M (8:8:8)	✓	✓	✕	✓

The following VESA VBE text modes are supportable via SVGA:

Table 1- 3 VESA VBE Text Modes

Mode (hex)	Characters (col/row)
0x108	80x60
0x109	132x25
0x10A	132x43
0x10B	132x50
0x10C	132x60

P10 allows VESA bankswitching to be done through the bypass to enable additional VESA mode support. ModeX is also supported.

3.5 DMA

P10 supports a comprehensive set of DMA engines and uses Circular buffer input stream handling to reduce Command DMA setup overhead and latencies. Input streams can be from host or on-card memory with two levels of nesting. Output DMA returns data to host or local memory, performs image uploads and state return.

3.5.1 Graphics Core to Graphics I/O – Upload Controller

The GPIO Upload DMA Unit – GPIOUD – uploads message data from the graphics pipeline to the PCI and AGP bus masters.

The unit is controlled by PCI slave register writes and reads, which are resynchronised from P clock to K clock and back through the PCI slave write (PciGpWr) and PCI slave read (GpPciRd) FIFOs respectively.

The GP input half of the unit maintains 2 input message ports and 16+1 circular buffers. These generate outgoing message streams on the API and Isochronous output message FIFOs.

The GP output half of the unit maintains an output message port and a Sync interrupt signal. These are driven from the incoming message stream on the input message FIFO.

- Autonomous - set-up/fetch parallelism
- No wait state - maximum transfer rate
- Programmable block size - large DMA buffers
- Separate DMA controllers for upload and download can run concurrently

3.5.2 Graphics I/O to Geometry and Rasterizer – GPIO Command DMA

The GPIO Command DMA Unit issues DMA requests and processes the return data for GP command packets. These are inserted into the message stream. DMA packets are usually submitted via circular buffers which manage the GP core command interface.

3.5.3 Circular Buffers

Apart from the input message port, the circular buffer provides the only command interface to the GP core. They replace the GP Input FIFO and command DMA schemes of earlier chips.

The intention is that 16 user contexts (Api) and the GDI+ driver (Iso) each have their own private circular buffer backed by a DMA engine.[3] Wraparound is handled automatically by the GPIO Bus Interface.

[3] A "user context" here is considered to be the display driver, an OpenGL ICD process, or anything else wanting to make use of the GP core for 2D or 3D rendering.

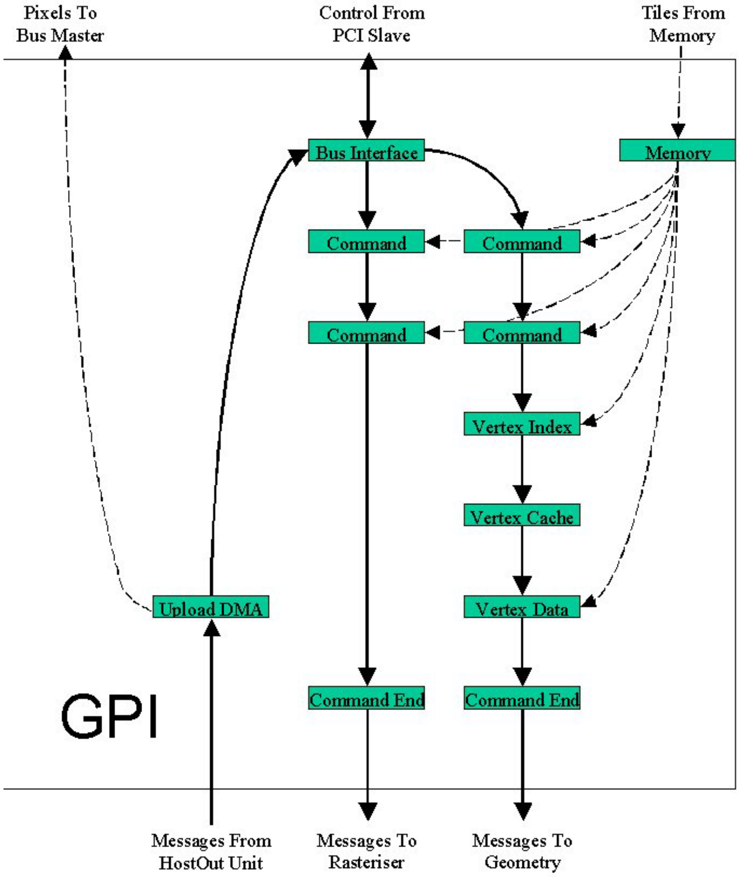


Figure 2.5 Graphics Processor I/O

3.5.4 Interrupt Controller

- End-of-DMA - allows DMA chaining
- VSYNC - efficient double buffering
- Scanline - special effects
- Texture invalid
- Bypass DMA interrupt
- I2C start condition - alert host to start of I2C transfer
- Sync - indicates graphics core is idle
- Error - e.g. writing to a full FIFO

4

Video Unit and RAMDAC

Miranda P10 uses high-speed 10-bit 350MHz DACs or the Digital Output port at 260MHz for Video Output. DVO can be single- or double-edged, 12 or 24 bits wide depending on how the two channels are deployed. RGB 888 and other formats are supported, RGBA requires 24bit double-edge (see below).

P10 supports typical screen resolutions up to 1600x1200 with refresh rates of 96Hz or 1920x1080 with refresh rates of 90Hz, or 2048x1536 at 60Hz. It supports packed pixel formats, with color depths of 8, 16, 24, 32 and 40 bits per pixel. It has dot-clock phase locked loops (PLLs) and triple 8-bit D/A converters. The RAMDAC contains a 64x64x2 bit cursor array to support a 2, 4, or 16 color hardware cursor with cursor shapes cache.

Stereo is supported on the main and overlay channels (left and right buffers). Dual head capability is built-in with two discrete video channels and Genlock to an external sync source (Hsync or Vsync). An external clock can be used directly or as an external reference source for the PLLs.

4.1.1 Pixel Formats

P10's planar tile structure and video bus support up to 64bpp in a wide variety of formats. Each 8x8 pixel screen-aligned tile is handled in one-byte increments up to 8 bytes per tile. Each memory access returns one tile, with multiple reads for 16, 32, 40 etc. bit depths. Each tile can be defined as a color, texture, depth or alpha as required, so an unusually wide range of pixel formats can be supported. 32 bit colour and 565 colour formats are handled directly, other formats such 555, 4444, etc. are configured In the Pixel Unit.

The table shows the bit positions in the input data used to represent different color components.

Format	Name	RGB	Bits/pixel	R	G	B	A	Index
0	CI8	-	8	-	-	-	-	0-7
1	3:3:2	0	8	0-2	3-5	6-7	-	-
1	3:3:2	1	8	5-7	2-4	0-1	-	-
2	5:5:5:1	0	16	0-4	5-9	10-14	15	-
2	5:5:5:1	1	16	10-14	5-9	0-4	15	-
3	5:6:5	0	16	0-4	5-10	11-15	-	-
3	5:6:5	1	16	11-15	5-10	0-4	-	-
4	8:8:8:8	0	32	0-7	8-15	16-23	24-31	-
4	8:8:8:8	1	32	16-23	8-15	0-7	24-31	-
5	10:10:10:2	0	32	0-9	10-19	20-29	30-31	-
5	10:10:10:2	1	32	20-29	10-19	0-9	30-31	-
6	CI4	-	4	-	-	-	-	0-3, 4-7

Table 3.1.1 Pixel formats

The pixel size is independent of the color format, so it is possible to have an 8 bit pixel with a 32 bit stride. The bitmask format is different because it uses 4 bits per pixel regardless of pixel size; this format must be used with a one byte pixel size. The pipeline maintains 16 bits per component, but various operations use different numbers of bits. Color key uses 8 bits, blends use 8 bits, LUTs use 8 bits for input but output 10 bits.

4.1.1.1 Pixel Channel Key

Each pixel to be displayed may have contributions from any of the four channels. The pixel color is determined by working through the channels in the order underlay, main, overlay, cursor:

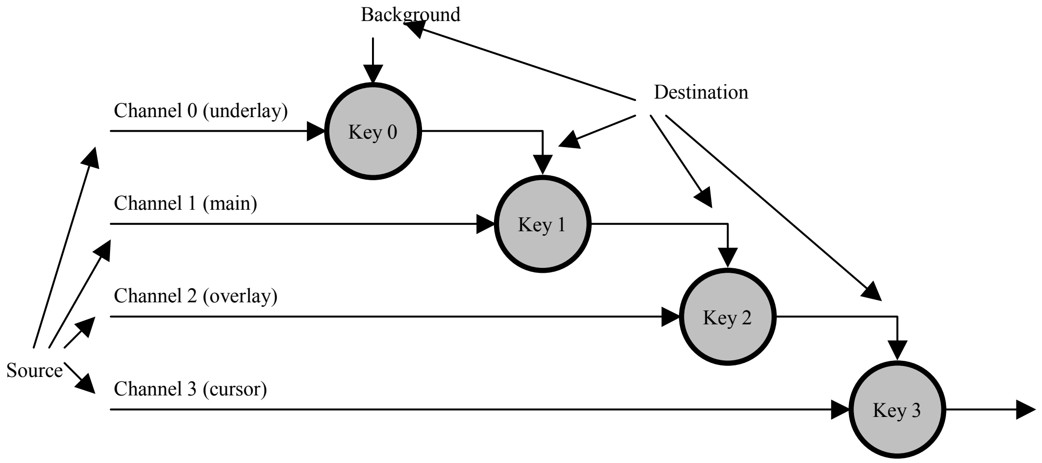


Figure 3.1 Pixel Channel Keys

4.1.2 Implementing Cursors

P10 implements standard Windows cursors as well as X Windows and Macintosh desktop.

4.1.2.1 LUTs

Two lookup tables are used to remap the pixel color. Typical applications include using one table to dereference index data while another gamma-corrects RGB data, or to support two different gammas (perhaps one for video, the other for 3D).

4.1.3 Scaling

P10 handles general video overlay scaling (where the data needs to be up- or down-converted with high quality scaling) through the graphics processor. The video sub-system is also able to upscale in X and Y by a limited amount which is suitable for displaying small framebuffers on fixed resolution displays.

For example, in a two-head system, one head may be used to drive a projector with a fixed resolution of 800x600, while the other head displays the same data on a flat panel display at 1024x768. To get good quality projection, the framebuffer is set to 800x600, but this will not fill the flat panel display so the hardware scaling can be used to increase the effective size of the framebuffer.

4.1.4 Synchronization

There are two lock bits which may be used to synchronize different channels within a head, or different heads. The lock registers hold a mask of which channels take part in the lock, and there are two lock registers per head.

All heads have access to all lock pins so they can be used to synchronize two heads in the same chip; the pins can also be shared by separate chips.

4.1.5 Clocks and PLLs

Clock/PLL configurations are highly flexible and support external clock reference for e.g. genlock. There is one clock for the graphics processor (KClk), one for the memory clock (MClk) and one for each display head (DClk0..DClkn).

There are 4 PLLs which can be individually programmed to different frequencies; PLL0 has 4 sets of registers to allow switching between different frequencies (required for VGA). The default settings for the 4 registers for PLL0 are:

Register set	Frequency
0	25.057MHz
1	28.278MHz
2	undefined
3	undefined

The PLLs can use the internal 14MHz oscillator as a refence clock, or an external source for genlocking. Each clock specifies its source which can be the PCI clock, an external clock froma pin, or one of the PLLs; any PLL can drive any clock.

One of the standard sources (PClk or the PLLs) can be output to a pin; the frequency of this clock can be divided by 1, 2, or 4, and optionally inverted.

For detailed configuration instructions see the *P10 Programmer's Guide*.

4.1.6 Digital Port Control

Both display heads share a single digital port which can be used to output or input digital video. Input video is only used when 2 P10s share the same display (other types of video input should use the video input port). Output video may be used to drive a flat panel controller or a TV encoder.

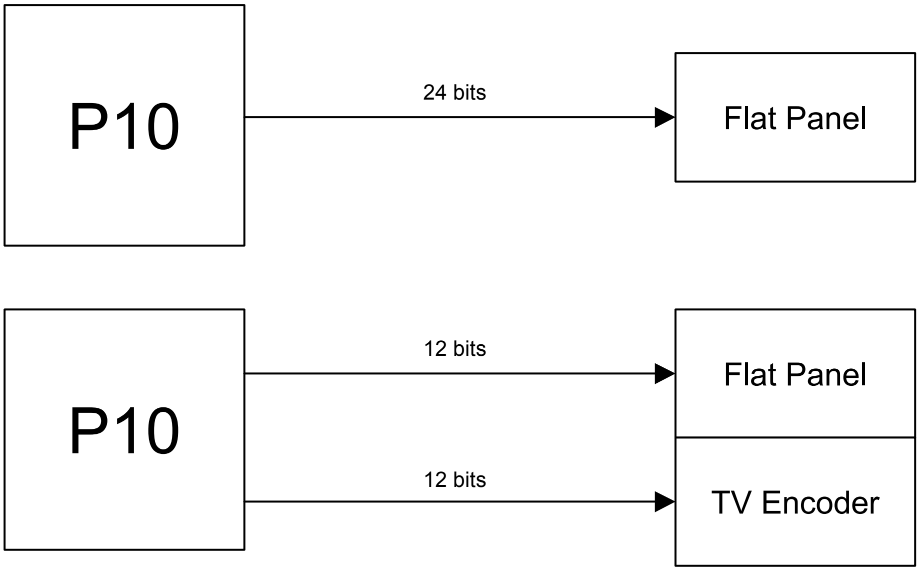


Figure 3.2 Digital Port Configuration

There are 24 data pins to which devices may be attached. The way the digital port pins are configured depends on how external devices have been connected to P10. Some examples are:

Usage	Mode[4]	C 0[5]	C 1[6]	DE[7]	M 0[8]	M 1[9]	Notes
Single flat panel	Out0	X	X	No	SinglePixel	Off	Single edge 24 bit data
Fast flat panel	Out 0	X	X	Yes	DoublePixel	Off	Dual edge 24 bit data.
Dual flat panel	Shared	Out	Out	Yes	SinglePixel	Single Pixel	Dual edge 12 bit data (x2)
Video editing	Out0	X	X	Yes	AlphaPixel	Off	Dual edge 48 bit data.

4.2 Digital Video Merge Bus

P10 is intended to work in dual-rasterizer environments. Two P10s can be configured to work together and split workload. The screen is divided into 64x64 pixel supertiles which are allocated to each P10 in a chequer board pattern. The P10s are joined by the digital port and configured such that one outputs data to the other which combines the data with its own and drives the display.

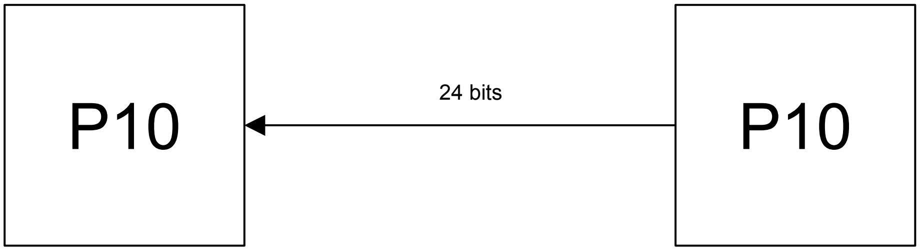


Figure 3.3 Digital Video Merge Bus

The digital port can be run 12 bits wide (double clocked) to allow flat panel output as well as interleaving.

[4] *Mode = VideoDigitalPortControl.Mode*

[5] *C0 = VideoDigitalPortControl.Channel0*

[6] *C1 = VideoDigitalPortControl.Channel1*

[7] *DE = VideoDigitalPortControl.DoubleEdge*

[8] *M0 = VideoDPMode.Mode (head 0)*

[9] *M1 = VideoDPMode.Mode (head 1)*

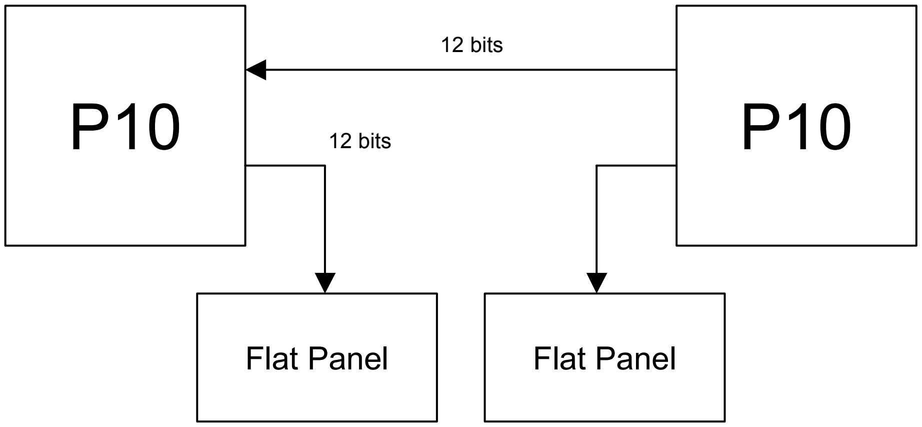


Figure 3.4 Dual-head Digital Port Configuration

For further information on PLL/clock configuration for dual head or mixed digital and analog setup see the P10 Programmer’s Guide.

5

Software Drivers

3Dlabs have extensive experience and a proven track record in delivering high performance, high quality, ready-to-ship WHQL certified software drivers that extract the maximum performance from both the Miranda P10 3D processor and the entire system.

5.1 2D Windows NT version 4/Windows 2000 with DirectX 7 and 8, Windows ME

Other software drivers may be made available depending on current market requirements.

5.2 3D Drivers

P10 has been designed to accelerate the key consumer focused 3D APIs and drivers. 3Dlabs' processors have historically been the reference port for many 3D drivers including Microsoft's OpenGL DDK.

P10 high performance 3D drivers support:

- Direct3D 7 and 8
- OpenGL 1.1 (OpenGL 1.2 when this is supported by Microsoft)
- Autodesk's Heidi for 3D Studio MAX support, including all D3D and OpenGL Depth and Stencil modes.

5.3 SVGA BIOS

- SVGA BIOS based on the proven, industry-standard Phoenix Technologies BIOS core

6

OEM and Embedded Solutions

6.1 PC 2001 and PC 99 compliance

P10 meets full PC 2001 and PC 99 compliance parameters detailed below:

Description	Consumer		Office		Entertainment	
	PC 2001	PC 99	PC 2001	PC 99	PC 2001	PC 99
System Requirements for Graphics Adapters						
Graphics adapter uses PCI, AGP or high-speed bus	✓	✓	✓	✓	✓	✓ AGP
System provides hardware-accelerated 3D graphics	✓	✓	✓	✓ [1]	✓	✓
System uses WC with higher-performance processors	✓	✓	✓	✓	✓	✓
Primary graphics adapter works normally with default VGA mode driver	✓	✓	✓	✓	✓	✓
Adapter and driver support multiple adapters and multiple monitors	✓	✓	✓	✓	✓	✓
Adapter supports television output if system does not include large-screen monitor	✓ [1]	✓ [1]	✓ [1]	✓ [1]	✓	✓ [1]
Hardware Acceleration for Video Playback						
Adapter supports video overlay surface with scaling	✓	✓	✓	✓	✓	✓
Hardware supports VGA destination color keying for video rectangle	✓	✓ [3]	✓ [3]	✓ [3]	✓	✓
Adapter supports MPEG-2 motion compensation acceleration	✓ [1]	✓ [1]	✓ [1]	✓ [1]	✓ [1]	✓ [1]
Adapter provides the ability to scan at the same frequency as the incoming video		✓ [1]		✓ [1]		✓ [1]
Multiple-Adapter and Multiple-Monitor Support						
Extended resources can be dynamically relocated after system boot	✓	✓	✓	✓	✓	✓
VGA resources can be disabled by software	✓	✓	✓	✓	✓	✓
System includes DTV support	✓	✓	✓	✓	✓	✓
Video input, capture, and broadcast device support is based on DirectX foundation class and WDM Stream class		n/a [2]		n/a [2]		n/a [2]
Hardware MPEG-2 decoder uses Digital data output port for video data		n/a [2]		n/a [2]		n/a [2]
PCI-based tuners and decoders support bus mastering with scatter/gather DMA		n/a [2]		n/a [2]		n/a [2]
Background tasks do not interfere with MPEG-2 playback	✓	✓	✓	✓	✓	✓

Description	Consumer		Office		Entertainment	
	PC 2001	PC 99	PC 2001	PC 99	PC 2001	PC 99
All components meet PC 2001 / PC 99 general device requirements		✓		✓		✓

Table 6.3 - PC 2001 and PC 99 compliance

[1] Recommended

[2] Optional

3 Required for Video

6.2 Data sheet

Texture Mapping

- True perspective correction
- Multiple texture engine (8+)
- Trilinear filtering with per-pixel MIP-mapping
- Palletized and RGB textures
- Bump Mapping, Convolutions, Displacement Mapping
- Transparency Maps
- Local texture buffer
- Specular, diffuse, ambient multiple lights
- Fast texture paging/loading
- AGP execute mode for remote texturing
- Color keying

3D Rendering

- Points, lines, triangles & bitmaps
- Gouraud and flat shading
- 8-, 16- 24-, 32- and 40--bit RGB/A
- Depth (z), GID buffering
- Fogging & depth-cueing
- Alpha blending (flat and Gouraud)
- H/W full screen anti-aliasing (FSAA)
- Dithering
- Area stippling
- Stencil test and stencil buffer
- Scissors test and logic operations

Display Features

- 8-, 16-, 24-, 32- and 40-bit RGB/A
- 8-bit color index
- Double and triple-buffering
- Hardware dithering
- Hardware pan
- Overlays

Fast Video Playback

- MPEG2 playback acceleration
- YUV color space conversion
- Scaling and shrink (bilinear filtered)
- Dithering
- Color keying (blue-screen)
- Alpha overlay blending

GUI Acceleration

- BitBlt with ROPs
- Points, lines, polygons
- Fills and text primitives
- Fast linear framebuffer
- On chip SVGA
- Windows

PCI/AGP Interface

- 32-bit glueless PCI V2.1
- MHz PCI / 266MHz AGP 4X
- Dual 2.5/3.3VDC 4X and 2X compatible
- Target and master support
- DMA mastering
- 256 entry command FIFO
- Big-endian apertures on bus
- Interrupts

Memory Architecture

- 128-bit DDRAM interface
- Single multi-function memory
- Optimal memory usage
- 8 to 256 Mbytes

Display Resolutions

- 320x200 to ???
- Ergonomic refresh rates

TV/Video Output

- 350 MHz RAMDAC interface
- LCD flat panel support
- 240MHz Digital Video output

Power Management

- VESA DPMS
- VESA DDC support
- Separate clocks for all sub-systems
- Automatic frequency reduction when idle
- RAM power down mode

HPBGA Package

- ???-pin BGA
- 2.5/3.3 V (5V Tolerant PCI/AGP)

Driver Support

- Direct3D, DirectX and OpenGL
- Windows 95/98, Windows NT/Windows 2000, Windows ME.
- Heidi for 3D Studio MAX

INDEX