TriTech Microelectronics

PRELIMINARY

TR25202

Pyramid3D™ 3D Accelerator

Features

- Full 10-component RGB model support
- Radiosity lighting support
- High integration, including video refresh, clock synthesizer and true-color DAC
- Unified buffer architecture for efficient memory utilization
- Perspective correction for both texture and true-color shading
- Complex shading effects including bump mapping and specular lighting
- Multiple simultaneous lighted textures with filtering
- Programmable pixel pipeline

Applications

- 3D GUI/game accelerators for MPC
- 3D media accelerators for MPC motherboard
- 3D accelerators for arcade
- 3D accelerators for home TV game-console
- Industrial virtual reality simulators
- Military war-game simulators

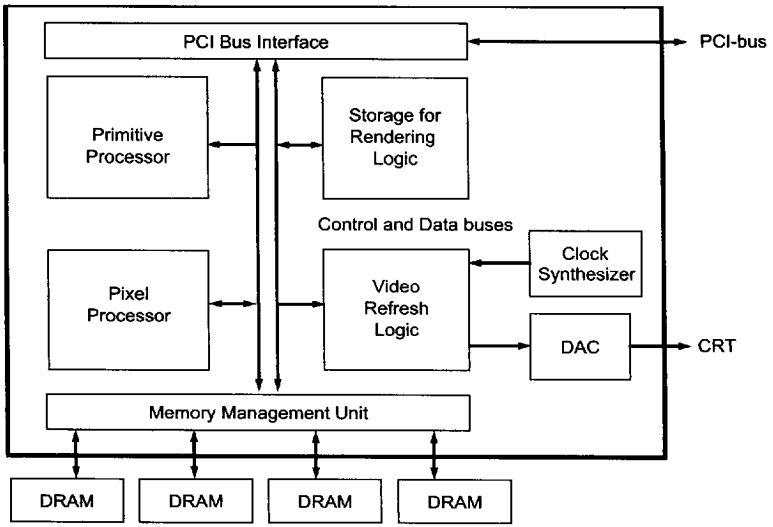
General Description

TR25202 is a member of TriTech's *Pyramid3D*™ family of highly integrated, programmable, and high performance 3D graphics accelerators. It is designed for the acceleration of games, 3D applications and user interfaces. It offers full compatibility with the emerging 3D standards including Direct3D™ for Windows® 95 and OpenGL™ for Windows NT.

TR25202 integrates on a single chip the primitive processor, pixel processor, PCI bus master interface, memory management unit, video refresh logic, clock synthesizer, and true-color DAC.

The features of TR25202 form a solid base on which support for different 3D APIs can be built easily. A full 3D graphics system requires only memory, in addition to TR25202. A low cost system is possible with two 256K x 32 SGRAMs or EDO DRAMs.

Block Diagram



9010734 0000065 913

Pin Configuration

TR25202 is packaged in a 304-pin Thermal Enhanced Ball Grid Array (TEBGA). The pinout top and bottom views are shown below, followed by the pin assignment table:

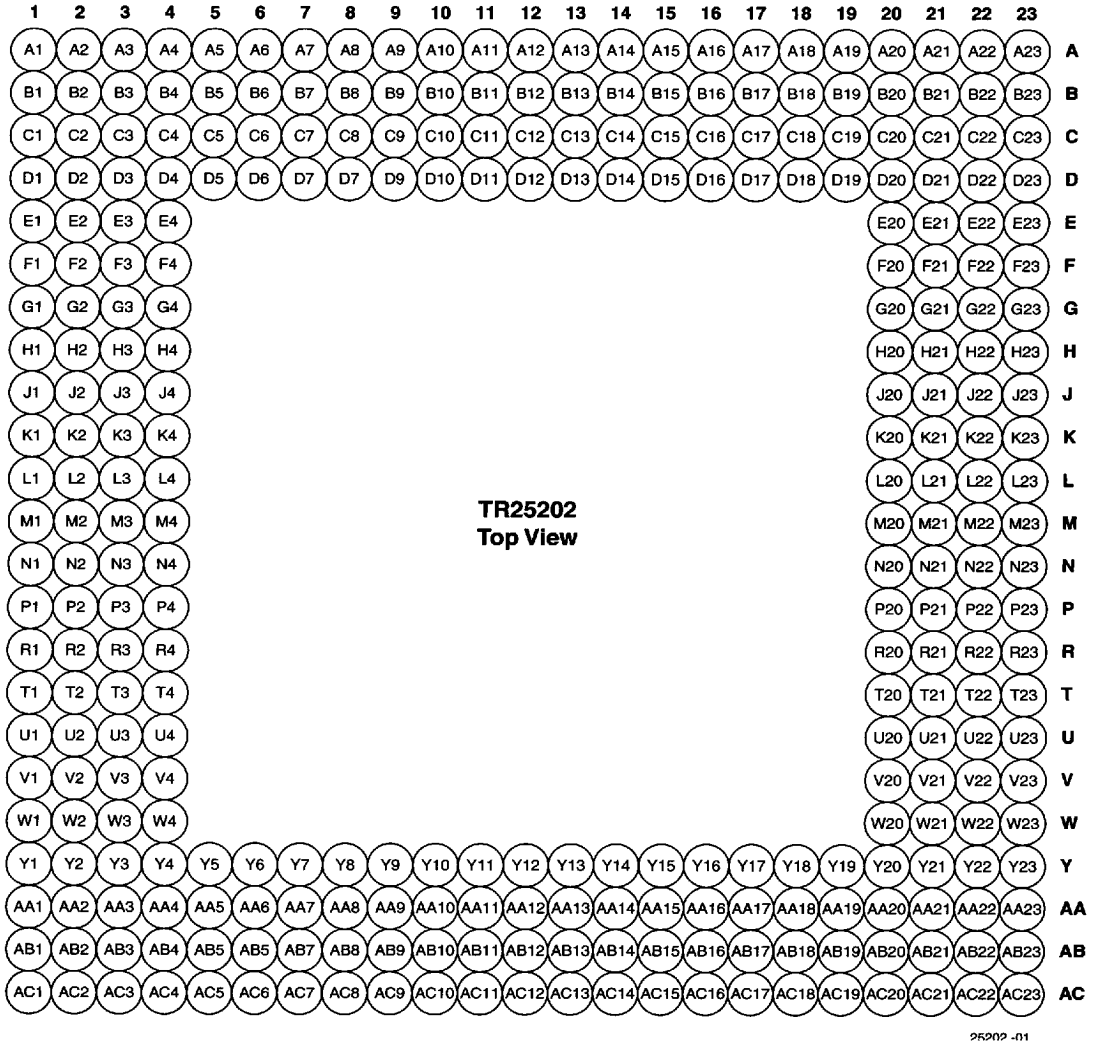


Figure 1 • *TR25202 Pinout (Top View)*

Pin Configuration (continued)

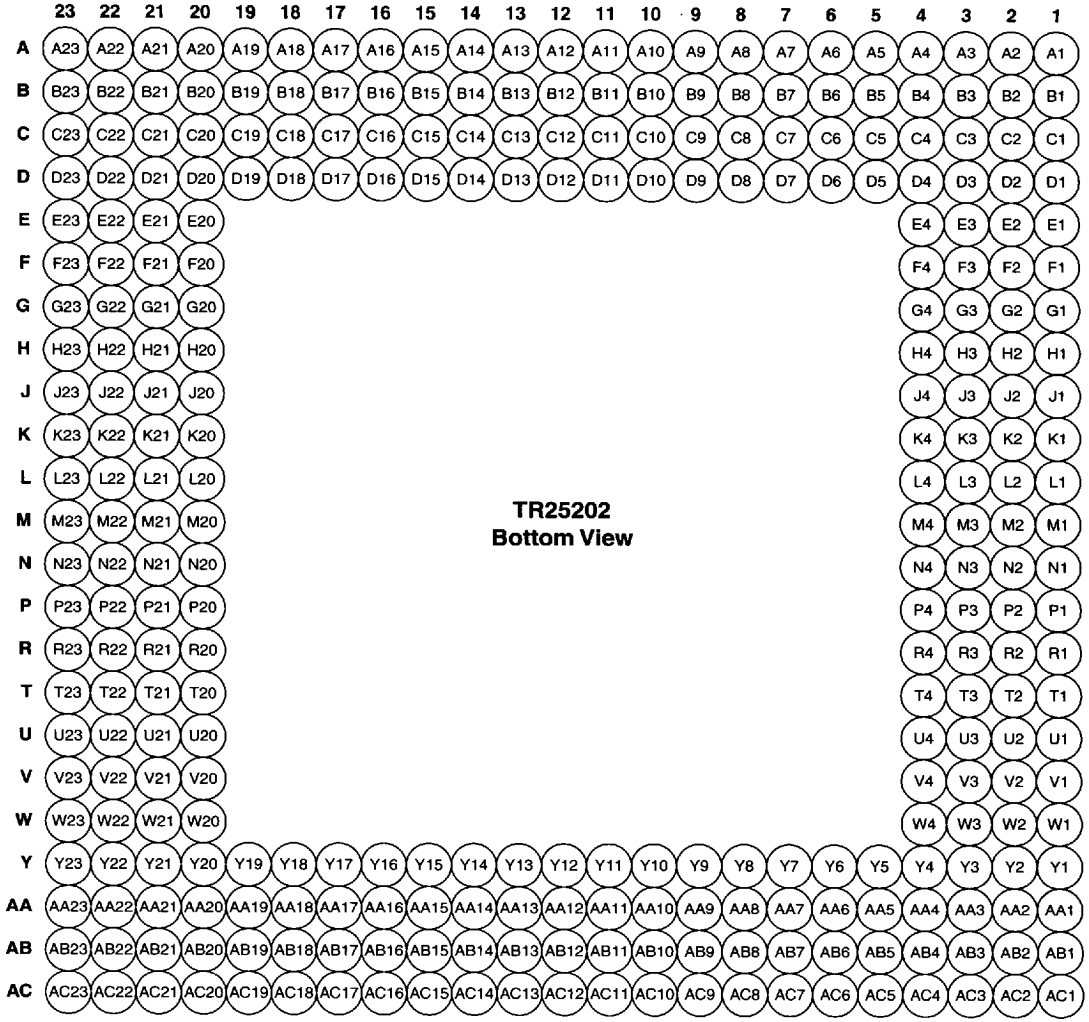


Figure 2 • *TR25202 Pinout (Bottom View)*

Pin Assignments

Pin	I/O Type	Category	Name
A1	out	B-memory	BCS[3]
A2	GND		
A3	I/O	video	VREF
A4	I	clock gen	XTAL_IN
A5	VDD	5V	
A6	GND		
A7	I	clock gen	CLK_TST[1]
A8	I	clock gen	CLK_TST[0]
A9	O	video	PCLK
A10	O	video	VSYNC
A11	I	video	VSYNC_IN
A12	I/O		USER_IO[2]
A13	I/O		USE_ROMB
A14	O	video	G[5]
A15	O	video	G[2]
A16	O	video	R[7]
A17	O	video	R[4]
A18	O	video	R[2]
A19	O	video	R[0]
A20	O	video	B[5]
A21	O	video	B[2]
A22	O	video	B[1]
A23	O	video	B[0]
B1	O	B-memory	BCS[2]
B2	O	video	IBLUE
B3	O	video	IGREEN
B4	GND		
B5		clock gen	FILTER_PROS
B6	GND		
B7	I/O	clock gen	VIDEO_CLK
B8	O	video	CSYNC
B9	O		HBLANK
B10	O		CBLANK
B11	I	video	HSYNC_IN
B12	I/O	video	USER_IO[3]
B13	I/O	video	USER_IO[0]
B14	O	video	G[6]

Pin	I/O Type	Category	Name
B15	O	video	G[3]
B16	O	video	G[0]
B17	O	video	R[5]
B18	O	video	R[3]
B19	O	video	R[1]
B20	O	video	B[6]
B21	O	video	B[3]
B22	O	A-memory	ACS[1]
B23	O	A-memory	ACS[0]
C1	I/O	B-memory	BDQ[2]
C2	I/O	B-memory	BDQ[1]
C3	I/O	B-memory	BDQ[0]
C4	VDD	5V	
C5	O	video	IRED
C6	O	clock gen	XTAL_OUT
C7		clock gen	FILTER_VIDEO
C8	O	video	HSYNC
C9	VDD	5V	
C10	O		VBLANK
C11	GND		
C12	I/O		USER_IO[4]
C13	I/O		USER_IO[1]
C14	O	video	G[7]
C15	O	video	G[4]
C16	O	video	G[1]
C17	O	video	R[6]
C18	GND		
C19	VDD	5V	
C20	O	video	B[7]
C21	O	video	B[4]
C22	I/O	A-memory	ADQ[30]
C23	I/O	A-memory	ADQ[31]
D1	I/O	B-memory	BDQ[4]
D2	I/O	B-memory	BDQ[3]
D3	GND		
D4	VDD	5V	
D5	GND		

Pin Assignments (continued)

Pin	I/O Type	Category	Name
D6	I/O	video	RREF
D7	VDD	5V	
D8	I/O	clock gen	PROS_CLK
D9	GND		
D10	VDD	5V	
D11	VDD	5V	
D12	VDD	5V	
D13	VDD	5V	
D14	GND		
D15	VDD	5V	
D16	GND		
D17	VDD	5V	
D18	GND		
D19	VDD	5V	
D20	I/O	A-memory	ADQ[26]
D21	I/O	A-memory	ADQ[27]
D22	I/O	A-memory	ADQ[28]
D23	I/O	A-memory	ADQ[29]
E1	I/O	B-memory	BDQ[7]
E2	I/O	B-memory	BDQ[6]
E3	I/O	B-memory	BDQ[5]
E4	VDD	memory	
E20	GND		
E21	I/O	A-memory	ADQ[23]
E22	I/O	A-memory	ADQ[24]
E23	I/O	A-memory	ADQ[25]
F1	I/O	B-memory	BDQ[10]
F2	I/O	B-memory	BDQ[9]
F3	I/O	B-memory	BDQ[8]
F4	GND		
F20	VDD	memory	
F21	I/O	A-memory	ADQ[20]
F22	I/O	A-memory	ADQ[21]
F23	I/O	A-memory	ADQ[22]
G1	I/O	B-memory	BDQ[13]
G2	I/O	B_memory	BDQ[12]
G3	I/O	B-memory	BDQ[11]

Pin	I/O Type	Category	Name
G4	VDD	memory	
G21	I/O	A-memory	ADQ[17]
G22	I/O	A-memory	ADQ[18]
G23	I/O	A-memory	ADQ[19]
H1	O	B-memory	BDQM[3]
H2	I/O	B-memory	BDQ[15]
H3	I/O	B-memory	BDQ[14]
H4	VDD	memory	
H20	VDD	memory	
H21	I	A-memory	AMEMCLKIN
H22	O	A-memory	AMEMCLK
H23	I/O	A-memory	ADQ[16]
J1	O	B-memory	BA[11]
J2	O	B-memory	BDQM[2]
J3	GND		
J4	VDD	5V	
J20	GND		
J21	GND		
J22	O	A-memory	ARAS
J23	O	A-memory	AWE
K1	O	B-memory	BA[8]
K2	O	B-memory	BA[9]
K3	O	B-memory	BA[10]
K4	GND		
k20	VDD	5V	
K21	O	A-memory	ACAS
K22	O	A-memory	ADQM[1]
K23	O	A-memory	ADQM[0]
L1	O	B-memory	BA[6]
L2	O	B-memory	BA[7]
L3	VDD	memory	
L4	VDD	memory	
L20	VDD	memory	
L21	O	A-memory	AA[2]
L22	O	A-memory	AA[1]
L23	O	A-memory	AA[0]
M1	O	B-memory	BA[3]

Pin Assignments (continued)

Pin	I/O Type	Category	Name
M2	O	B-memory	BA[4]
M3	O	B-memory	BA[5]
M4	GND		
M20	GND		
M21	O	A-memory	AA[5]
M22	O	A-memory	AA[4]
M23	O	A-memory	AA[3]
N1	O	B-memory	BA[0]
N2	O	B-memory	BA[1]
N3	O	B-memory	BA[2]
N4	VDD	memory	
N20	VDD	memory	
N21	VDD	memory	
N22	O	A-memory	AA[7]
N23	O	A-memory	AA[6]
P1	O	B-memory	BDQM[0]
P2	O	B-memory	BDQM[1]
P3	O	B-memory	BCAS
P4	VDD	5V	
P20	GND		
P21	O	A-memory	AA[10]
P22	O	A-memory	AA[9]
P23	O	A-memory	AA[8]
R1	O	B-memory	BWE
R2	O	B-memory	BRAS
R3	GND		
R4	GND		
R20	VDD	5V	
R21	GND		
R22	O	A-memory	ADQM[2]
R23	O	A-memory	AA[11]
T1	I/O	B-memory	BDQ[16]
T2	O	B-memory	BMEMCLK
T3	I	B-memory	BMEMCLKIN
T4	VDD	memory	
T20	VDD	memory	
T21	I/O	A-memory	ADQ[14]

Pin	I/O Type	Category	Name
T22	I/O	A-memory	ADQ[15]
T23	O	A-memory	ADQM[3]
U1	I/O	B-memory	BDQ[19]
U2	I/O	B-memory	BDQ[18]
U3	I/O	B-memory	BDQ[17]
U4	GND		
U20	VDD	memory	
U21	I/O	A-memory	ADQ[11]
U22	I/O	A-memory	ADQ[12]
U23	I/O	A-memory	ADQ[13]
V1	I/O	B-memory	BDQ[22]
V2	I/O	B-memory	BDQ[21]
V3	I/O	B-memory	BDQ[20]
V4	VDD	memory	
V20	GND		
V21	I/O	A-memory	ADQ[8]
V22	I/O	A-memory	ADQ[9]
V23	I/O	A-memory	ADQ[10]
W1	I/O	B-memory	BDQ[25]
W2	I/O	B-memory	BDQ[24]
W3	I/O	B-memory	BDQ[23]
W4	GND		
W20	VDD	memory	
W21	I/O	A-memory	ADQ[5]
W22	I/O	A-memory	ADQ[6]
W23	I/O	A-memory	ADQ[7]
Y1	I/O	B-memory	BDQ[29]
Y2	I/O	B-memory	BDQ[28]
Y3	I/O	B-memory	BDQ[27]
Y4	I/O	B-memory	BDQ[26]
Y5	VDD	PCI	
Y6	GND		
Y7	VDD	PCI	
Y8	GND		
Y9	VDD	PCI	
Y10	GND		
Y11	VDD	PCI	

Pin Assignments (continued)

Pin	I/O Type	Category	Name
Y12	VDD	5V	
Y13	GND		
Y14	VDD	PCI	
Y15	GND		
Y16	GND		
Y17	VDD	PCI	
Y18	VDD	5V	
Y19	GND		
Y20	VDD	PCI	
Y21	GND		
Y22	I/O	A-memory	ADQ[3]
Y23	I/O	A-memory	ADQ[4]
AA1	I/O	B-memory	BDQ[31]
AA2	I/O	B-memory	BDQ[30]
AA3	I	PCI interface	PCI_CLK
AA4	I/O	PCI interface	PCI_AD[29]
AA5	GND		
AA6	GND		
AA7	VDD	pci	
AA8	VDD	5V	
AA9	I/O	PCI interface	PCI_AD[21]
AA10	I/O	PCI interface	PCI_AD[17]
AA11	VDD	PCI	
AA12	GND		
AA13	GND		
AA14	VDD	5V	
AA15	I/O	PCI interface	PCI_AD[15]
AA16	VDD	PCI	
AA17	I/O	PCI interface	PCI_AD[11]
AA18	I/O	PCI interface	PCI_C_BEB[0]
AA19	I/O	PCI interface	PCI_AD[6]
AA20	GND		
AA21	I/O	A-memory	ADQ[0]
AA22	I/O	A-memory	ADQ[1]
AA23	I/O	A-memory	ADQ[2]
AB1	O	B-memory	BCS[0]
AB2	O	B-memory	BCS[1]

Pin	I/O Type	Category	Name
AB3	I	PCI interface	PCI_GNTB
AB4	I/O	PCI interface	PCI_AD[30]
AB5	I/O	PCI interface	PCI_AD[27]
AB6	I/O	PCI interface	PCI_AD[25]
AB7	I/O	PCI interface	PCI_C_BEB[3]
AB8	I/O	PCI interface	PCI_AD[23]
AB9	I/O	PCI interface	PCI_AD[20]
AB10	I/O	PCI interface	PCI_AD[18]
AB11	I/O	PCI interface	PCI_C_BEB[2]
AB12	I/O	PCI interface	PCI_IRDYB
AB13	I/O	PCI interface	PCI_DEVSELB
AB14	I/O	PCI interface	PCI_PERRB
AB15	I/O	PCI interface	PCI_PAR
AB16	I/O	PCI interface	PCI_AD[14]
AB17	I/O	PCI interface	PCI_AD[12]
AB18	I/O	PCI interface	PCI_AD[9]
AB19	I/O	PCI interface	PCI_AD[7]
AB20	I/O	PCI interface	PCI_AD[5]
AB21	I/O	PCI interface	PCI_AD[2]
AB22	I/O	PCI interface	PCI_AD[0]
AB23	O	A-memory	ACS[2]
AC1	I	PCI interface	PCI_INTAB
AC2	I	PCI interface	PCI_RSTB
AC3	I	PCI interface	PCI_REQB
AC4	I/O	PCI interface	PCI_AD[31]
AC5	I/O	PCI interface	PCI_AD[28]
AC6	I/O	PCI interface	PCI_AD[26]
AC7	I/O	PCI interface	PCI_AD[24]
AC8	I	PCI interface	PCI_IDSEL
AC9	I/O	PCI interface	PCI_AD[22]
AC10	I/O	PCI interface	PCI_AD[19]
AC11	I/O	PCI interface	PCI_AD[16]
AC12	I/O	PCI interface	PCI_FRAMEB
AC13	I/O	PCI interface	PCI_TRDYB
AC14	I/O	PCI interface	PCI_STOPB
AC15	I/O	PCI interface	PCI_SERRB
AC16	I/O	PCI interface	PCI_C_BEB[1]

Pin Assignments (continued)

Pin	I/O Type	Category	Name
AC17	I/O	PCI interface	PCI_AD[13]
AC18	I/O	PCI interface	PCI_AD[10]
AC19	I/O	PCI interface	PCI_AD[8]
AC20	I/O	PCI interface	PCI_AD[4]

Pin	I/O Type	Category	Name
AC21	I	PCI interface	PCI_AD[3]
AC22	I/O	PCI interface	PCI_AD[1]
AC23	O	A-memory	ACS[3]

Pin Definitions

Pin Name	I/O Type	Description
*RESET	I	System reset signal
Video, Analog and User I/O Pins These pins include the pins related to video signal generation (both digital and analog). It also contains the pins which are related to the internal or external clock generator. Also included are four general purpose pins for application specific I/O. The TR25202 contains the required circuitry for clock generation (PLL) and for direct analog video out (current mode DAC). However, it is possible to replace these with external components, and in order to use the internal clock generator the clock generator outputs must be connected on the board level to the corresponding outputs.		
PROS_CLK	I/O (NC)	Processor Clock. Normally not connected, can be used either as a clock input or as a clock output depending on the CLK_TST signals.
VIDEO_CLK	I/O (NC)	Video Clock. Normally not connected, can be used either as a clock input or as a clock output depending on the CLK_TST signals.
CHIP_LOOP_FILTER	Analog I/O	Chip Loop Filter Connection. The loop filter connection for the chip clock PLL. Typical loop filter component values are C=100nF, R=400Ω.
VIDEO_LOOP_FILTER	Analog I/O	Video Loop Filter Connection. The loop filter connection for the video clock PLL. Typical loop filter component values are C=100nF, R=400Ω.
XTAL_IN, XTAL_OUT	Analog I/O	Crystal In, Crystal Out. Connections for the external crystal for the internal clock generator. Typical crystal frequency is 14.3181818 MHz.
VREF	I/O	Voltage Reference (for the Video DAC). This is the output of the TR25202 internal voltage reference (1.23 V). The output has relatively high impedance (10kΩ), so it is possible to override it with an external voltage reference. It is recommended that a bypass capacitor be attached to this pin.
RREF		Resistor Reference. A resistor of 1.1kΩ should be connected between this pin and the ground. The formula relating the DAC msb current, VREF, and RREF is: $\frac{I_{msb}}{8} = \frac{V_{REF}}{R_{REF}}$ where I_{msb} is the current change corresponding to the most significant bit in the color data. Typical value for the ratio is: $\frac{V_{ref}}{R_{ref}} = 1.1mA$
IRED, IGREEN, IBLUE	Analog O	Analog (current mode) red, green, and blue video output signals.
CLK_TST[1:0]	I	CLK_TST mode. Configures the direction of the VIDEO_CLK and PROS_CLK.

Pin Definitions (continued)

Pin Name	I/O Type	Description
PCLK	O	Pixel Clock Out; for an external DAC. This clock is a delayed (bonding PAD delay) version of the internally used video clock.
USER_IO[4:0]	I/O	User I/O. General purpose I/O connection pins. These pins can be read and written, and their direction changed using internal registers.
USE_ROM#	O	Use ROM (active low). TR25202 can use a ROM which is connected to the digital RGB lines for boot configuration and as a BIOS ROM. The USE ROM# line is used to differentiate between normal digital video usage and ROM access usage. It should be connected to the ROM chip select and output enable lines. (Both signals should be active low, also the ROM used must set the data pins to high impedance state when it is not selected).
CBLANK	O	Composite Blank. The combined blank signal created from the horizontal and vertical blank signals.
HBLANK	O	Horizontal Blank.
VBLANK	O	Vertical Blank.
CSYNC	O	Composite Sync. The combined sync signal created from the horizontal and vertical sync signals.
HSYNC	O	Horizontal Sync.
VSYNC	O	Vertical Sync.
HSYNC_IN	I	Horizontal Sync In. TR25202 will detect the transition from non-active to active state on this line, and synchronize TR25202's internal operation to it.
VSYNC_IN	I	Vertical Sync In. TR25202 will detect the transition from non-active to active state on this line, and synchronize TR25202's internal operation to it.
B[7:0]	I/O	Blue (digital). This bus is also used as the data bus (input) when performing ROM accesses. It is also possible to utilize it as an extra digital input resource if the digital RGB output is not used.
R[7:0]	O	Red (digital). This bus is also used as the high order address bits when performing ROM accesses. It is also possible to utilize it as an extra digital output resource if the digital RGB output is not used.
G[7:0]	O	Green (digital). This bus is also used as the low order address bits when performing ROM accesses. It is also possible to utilize it as an extra digital output resource if the digital RGB output is not used.
Memory Pins The memory pins are in two groups and referred to as A and B memory pins respectively. These pin groups have similar operations. For specific description of the operation of the memory control pins please refer to the documentation of the memory interface.		
AA[11:0]	O	A-Memory Address. When used with SDRAM or SGRAM the memory address bus is also used to transfer configuration data and to perform bank select operations, so it is essential that the relevant address pins are connected to the corresponding address pins of the memories (it is not ok to swap the address pins).
ACAS#	O	A-Memory Column Address Select. Used for synchronous memory configurations. For "normal" DRAM configurations the DRAM CAS lines should be connected to ADQM# lines.

Pin Definitions (continued)

Pin Name	I/O Type	Description
ACS#	O	A-Memory Chip Select. These lines are needed for large memory configurations. The chip selects are decoded so that the first memory device should be connected to ACS#[0], the second to ACS#[1] etc.
ADQM#[3:0]	O	A-Memory Data Byte Enables. These are connected to the DQM lines for synchronous memory alternatives (SDRAM or SGRAM) and to the CAS lines for "normal" DRAMs.
ADQ[31:0]	I/O	A-Memory Data Bus. The normal configuration for the A data bus is 32 bits wide (+ 32 bits for the B data bus), but it is possible to create a system with 16-bit (+ 16) wide interface when using SDRAM as the basic element for the memory subsystem.
AMEMCLK	O	A-Memory Clock. The clock signal used by the synchronous memories. If a non-synchronous memory structure is used this signal is not connected to the memories.
AMEMCLKIN	I	A-Memory Clock In. Used for controlling the latching-in of the external data. This pin must be connected to the AMEMCLK pin. (The connection must be made even in configurations with non-synchronous memories).
ARAS#	O	A-Memory Row Address Select. Connected to the corresponding pin in both synchronous and non-synchronous memories.
AWE#	O	A-Memory Write Enable. Connected to the corresponding pin in both synchronous and non-synchronous memories.
BA[11:0]	O	B-Memory Address. When used with SDRAM or SGRAM, the memory address bus is also used to transfer configuration data and perform bank select operations, so it is essential that the relevant address pins are connected to the corresponding address pins on the memories (it is not ok to swap the address pins).
BCAS#	O	B-Memory Column Address Select. Used for synchronous memory configurations. For normal" "DRAM configurations the DRAM CAS lines should be connected to BDQM# lines.
BCS#	O	B-Memory Chip Select. These lines are needed on large memory configurations. The chip selects are decoded so that that the first memory device should be connected to BCS#[0] the second to BCS#[1] etc.
BDQM#[3:0]	O	B-Memory Data Byte Enables. These are connected to the DQM lines for the synchronous memory alternatives (SDRAM or SGRAM) and to the CAS lines for "normal" DRAMs.
BDQ[31:0]	I/O	B-Memory Data Bus. The normal configuration for the B data bus is 32 bits wide (+ 32 bits for the A data bus), but it is possible to create a system with 16-bit (+ 16) wide interface when using SDRAM as the basic element for the memory subsystem.
BMEMCLK	O	B-Memory Clock. The clock signal used by the synchronous memories. If a non-synchronous memory structure is used this signal is not connected to the memories.
BMEMCLKIN	I	B-Memory Clock In. Used for controlling the latching-in of the external data. This pin must be connected to the BMEMCLK pin. (The connection must be made even in configurations with non-synchronous memories).

Pin Definitions (continued)

Pin Name	I/O Type	Description
BRAS#	O	B-Memory Row Address Select. Connected to the corresponding pin in both synchronous and non-synchronous memories.
BWE#	O	B-Memory Write Enable. Connected to the corresponding pin in both synchronous and non-synchronous memories.
PCI Pins For a complete reference of the PCI related pins please refer to PCI Local Bus Specification Revision 2.1 , which can be obtained from PCI Special Interest Group.		
PCI_AD[31:0]	I/O	The multiplexed address and databus.
PCI_C/BE#[3:0]	I/O	Bus Command and Byte Enables. Used to transmit the command on the first cycle of the transaction and the byte enables on the following cycles.
PCI_CLK	I	PCI Clock. TR25202 supports PCI clock frequencies in the range 0-33 MHz.
PCI_DEVSEL#	I/O	Device Select. Used by the transaction target to indicate that it has decoded and recognized the address of the transaction.
PCI_FRAME#	I/O	Cycle Frame. Driven by the transaction initiator to indicate the beginning and duration of an access.
PCI_GNT#	I	Grant. Indicates the grant of the bus access when operating as a bus master.
PCI_IDSEL	I	Initialization Device Select. Used as a chip select during the configuration transactions. (Confguration transactions do not use the normal PCI address decoding).
PCI_INTA#	O	Interrupt A. Interrupt request. The wiring of this interrupt line to the CPU interrupts is motherboard and operating system dependent. The interrupt is reset by resetting the corresponding status register bit.
PCI_IRDY#	I/O	Initiator Ready , used on the PCI bus to indicate that the initiator is ready to transfer data on the current clock cycle. Pin direction depends whether TR25202 is participating to the transfer as a target or as an initiator.
PCI_PAR	I/O	Parity.
PCI_PERR#	I/O	Parity error.
PCI_REQ#	O	Request for bus. Used when operating as the initiator, for requesting the bus ownership.
PCI_RST#	I/O	Reset.
PCI_SERR#	I/O	System error.
PCI_STOP#	I/O	Stop transaction, used by the target when it needs to stop a transaction. Typical usage does not indicate any kind of error condition.
PCI_TRDY#	I/O	Target Ready , used on the PCI bus to indicate that the target is ready to transfer data on the current clock cycle. pin direction depends whether TR25202 is participating to the transfer as a target or as an initiator.

TR25202 Architecture Overview

TR25202 offers a single chip implementation of the rendering stage of a traditional 3-stage 3D graphics pipeline comprising tessellation, geometry, and rendering. Together with the host, the architecture performs all the required operations from the object oriented scene handling to per-pixel operations. The primitives are first initialized in the geometry stage by the host, then rasterized in TR25202's primitive processor. The resulting individual pixels are sent to TR25202's pixel processor, which finally writes them to the screen through TR25202's memory management unit. The device also contains a complete PCI bus master interface for communicating with the host, and a video interface comprising a clock synthesizer and a true-color DAC.

Pixel Processor

The pixel processor handles visibility checking using the Z-buffer, performs texture data fetching, and blends colors for transparency and other effects. It receives as input a list of pixels along with their properties from the primitive processor; and as output writes the resulting colors to the local frame buffer memory. The process is described with a shading program, which can be provided by the user. For high picture quality, all operations are performed with true-color accuracy. In the calculation of the final color, it is possible to use multiple textures and special effects including fog, environment mapping and bump mapping.

Primitive Processor

The primitive processor, also known as the rasterizer, generates the individual pixels which form each primitive, and forwards them to the pixel processor. Primitives can be triangles, lines or 2D regions, and are defined by their edges. The primitive processor first determines which pixels are inside the primitive and then calculates a set of eight associated pixel properties. They can be used to control the color, transparency, fog intensity, specular intensity, primary texture, and secondary texture of the pixels. All properties, including colors, are interpolated with full perspective correction, without any performance penalties. This guarantees that lighting and texture will match perfectly and any undesired warping is avoided.

PCI Interface

The Pyramid3D can be directly connected to a PCI bus without any extra logic. The PCI interface provides the host with linear access to the frame buffer and memory mapped registers. The frame buffer can be accessed in both RGB and YUV format, making real-time video over the PCI very simple. In addition, bus mastering is supported so that textures, geometry and individual triangles can be read from the main memory without host processor overhead.

Texturing

In order to maximize image quality without maximizing memory usage, a wide variety of texturing methods are supported. The textures can range from 32 x 32 pixels with 4-bit indexed color, up to 1024 x 1024 pixels; they can also be of full 32-bit true-color quality. For indexed textures, the pixel processor has an internal 256-color 32-bit palette. Textures can contain transparency information. The texturing quality can be further increased with texture filtering. For all texture formats, it is possible to use MIP-mapping, bilinear and trilinear filtering.

Perspective Correction

Perspective correction is performed for both textures and shading without any performance penalty. This ensures that textures and lighting on all surfaces match perfectly. Full perspective correction also eliminates incorrect surface warping, which can otherwise be visible on nearby surfaces.

Complex Shading

Complex shading is performed in a single pass. Multiple simultaneous textures can be used, and are all properly filtered for the highest quality. The textures are combined with interpolated colors and settings using programmable shading operations. Complex shading effects such as textured surfaces with environment mapping can be efficiently generated.

Phong shading is efficiently simulated by using environment mapping. Multiple colored lights can be combined with reflective surfaces. The Pyramid3D architecture makes it possible to combine these highlights with textures and diffuse lighting, so that the result is correctly rendered. All this is performed in a single pass.

Bump Mapping

Bump mapping adds the final touch to surfaces. They can be made to look rough and detailed without any extra geometry. This also gives more control over the appearance of surfaces and makes

computation faster, since bump mapping is more efficient compared to adding the details with geometry.

Bump mapping adds surface detail without increasing geometric complexity. This operation requires multiple simultaneous texture channels provided by TR25202. The extra detail is specified by using a displacement map, defined like normal texture maps. It specifies the shape of the surface at each point.

Programmability

Programmability is central to the Pyramid3D architecture. The full programmability of the pixel processor makes it possible to realize many shading models and create new ones to suit the application.

Unified Buffer Architecture

Unified buffer architecture is used to store all the data, including frame buffer, textures, and geometry. This adds flexibility and makes memory usage more efficient, since no memory is ever left unused because of being assigned the wrong function. As a result, less memory is needed to support a wide range of 3D applications.

Functional Description[1]

Memory Interface

The preferred memory type for TR25202 is Synchronous DRAM (SDRAM) or Synchronous Graphics RAM (SGRAM). EDO DRAM and generic Fast Page Mode DRAM can also be used but with lower performance.

There are 3 commonly used memory configurations with TR25202:

- Type 1: Full performance can be achieved by connecting TR25202 with four 1Mx16 SDRAMs (a total of 8MBytes).
- Type 2: A version with lower performance can be created by using two 1Mx16 SDRAMs (a total of 4MBytes).
- Type 3: A further cost sensitive version can be created easily with two 256Kx32 SGRAMs or EDO DRAMs (a total of 2MBytes).

The performance drop between Type 1 and Type 2 is mostly significant when high resolution and/or true-color modes are used. Type 3 offers high polygon rendering rates comparable to Type 1 (SGRAM only), through a wide 64-bit memory interface, but with reduced local graphics memory depth for frame buffer, Z-buffer, and texture storage.

PCI Interface

TR25202 has a PCI bus master interface which conforms to PCI Local Bus Specification Revision 2.1. The PCI interface of TR25202 contains two base address registers. One register is used to map the internal registers and user controllable internal memories of TR25202 to the PCI bus. The other register is used for mapping the local graphics memory to the PCI bus.

PCI Bus Interface Feature Summary

Conforms to *PCI Local Bus Specification Revision 2.1.*

- Fast *DEVSEL#* assertion.
- When acting as a PCI target for write operations, it does not generate wait states.
- Memory on the graphics card is accessible using two independent apertures as suggested by *PCI Multimedia Design Guide rev 1.0.*
- Memory apertures can be configured to perform automatically the color space conversions needed, for example, for video streaming applications. TR25202 supports the conversions required for the pixel formats (RGB15+α, RGB24+α, and YUV 4:2:2) defined as the *Full Interoperability Level* in *PCI Multimedia Design Guide rev 1.0.*

1. Representation of numbers: decimal numbers are represented with no special suffix or prefix (e.g. 256); hexadecimal numbers are represented with suffix "h" (e.g. 7Fh); binary numbers are represented with suffix "b" (e.g. 00111010b).

PCI Configuration Space

TR25202 implements the PCI configuration space registers shown in Table 1.

Table 1. *PCI Configuration Space Registers*

Address	Register (bits 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0)
0	device_id | vendor_id
4	status | command
8	class_code | aa
12	- | ab | ac | ad
16	graphics_ram_BAR
20	control_registers_BAR
24	Not in use (reserved)
28	Not in use (reserved)
32	Not in use (reserved)
36	Not in use (reserved)
40	Not in use (reserved)
44	subsystem_ID | subsystem_vendor_ID
48	expansion_ROM_BAR | -
52	Not in use (reserved)
56	Not in use (reserved)
60	Max_Lat | Min_Gnt | ae | af
64	Core Clock Config a | - | ag | M_coef | N_coef
68	Memory Configuration - | ah | ai | b | c | -
72	Video Clock Config - | aj | M_coef | N_coef
76	Register access address register
80	Register access data register
84	Not in use (reserved)
88	Not in use (reserved)
92	Not in use (reserved)
96	Not in use (reserved)
100	Not in use (reserved)
104	Not in use (reserved)
108	Not in use (reserved)
112	Not in use (reserved)
116	Not in use (reserved)
120	Not in use (reserved)
124	Not in use (reserved)

Abbreviation	Name
a	non-overlap
aa	revision_id
ab	header_type
ac	latency_timer
ad	cache_line_size
ae	Interrupt Pin
af	Interrupt Line
ag	R_coef
ah	refresh_rate
ai	mm_1_2_4_depth
aj	R_coef
b	mm_16_32_width
c	mm_8_16_chips

- (register:0)
  - device_id (bits:31-16)
  - vendor_id (bits:15-0)
- (register:4)
  - status (bits:31-16)
  - command (bits:15-0)
- (register:8)
  - class_code (bits:31-8)
  - revision_id (bits:7-0)
- (register:12)
  - header_type (bits:23-16)
  - latency_timer (bits:15-8)
  - cache_line_size (bits:7-0)
- (register:16)
  - graphics_ram_BAR (bits:31-0)
- (register:20)
  - control_registers_BAR (bits:31-0)
- (register:44)
  - subsystem_ID (bits:31-16)
  - subsystem_vendor_ID (bits:15-0)
- (register:48)
  - expansion_ROM_BAR (bits:31-16)
- (register:60)
  - Max_Lat (bits:31-24)
  - Min_Gnt (bits:23-16)
  - Interrupt Pin (bits:15-8)
  - Interrupt Line (bits:7-0)

- Core Clock Config (register:64)

  The clock frequency can be calculated from the formula

$$F_{out} = \frac{M+2}{(N+2)\times 2^R} \times F_{Xtal}$$

  Caution: Unsuitable clock frequency parameters can make the board containing the circuit non-operational, and possibly damage the components.
  - non-overlap (bit:31)

    Control the internal clock buffer non-overlap time; 0 for short non-overlap time; 1 for longer non-overlap time.
  - R_coef (bits:15-14)
  - M_coef (bits:13-7)
  - N_coef (bits:6-0)
- Memory Configuration (register:68)

  Caution: Incorrect memory configuration can make the board containing the circuit non operational, and may damage the devices on the board
  - refresh_rate (bits:22-20)
  - mm_1_2_4_depth (bits:19-18)
  - mm_16_32_width (bit:17)
  - mm_8_16_chips (bit:16)
- Video Clock Config (register:72)

  The clock frequency can be calculated from the formula

$$F_{out} = \frac{M+2}{(N+2)\times 2^R} \times F_{Xtal}$$

  Caution: Unsuitable clock frequency parameters can make the board containing the circuit non operational, and possibly damage the components, or the video display
  - R_coef (bits:15-14)
  - M_coef (bits:13-7)
  - N_coef (bits:6-0)
- Register Access Address Register (register:76)
- Register Access Data Register (register:80)

  The register access register provide an alternative method for accessing TR25202 internal registers in situations where the normal memory mapped register access is not

available. Using this method is slow. For using the access registers, the target register address is written to the address register, and the value is written tothe data register. The actual register write happens when the most significant byte of the data access register is written. This can be done with 8, 16, or 32-bit configuration register write.

Accessing TR25202 Internal Registers

The following internal register ranges are available for access through the PCI interface. The registers are mapped to the PCI memory starting from the memory location specified by the *control_registers_BAR*. For example if the PCI BIOS has configured the TR25202 *control_registers_BAR* to the value 00010000h (65536) then the *cr_init* register is mapped to address 65536 + 64 × 4.

Range	Function
0-16	pixel processor
33-40	video refresh
42-55	system control
64-116	primitive processor
128-159	ppu_code
256-511	texture_palette

Memory Apertures

The PCI interface maps the graphics card memory to the PCI-bus. Different translations including native mode and linear frame buffer mode are available.

In order to implement interfaces to other PCI multimedia devices TR25202 provides simultaneous apertures to the memory (as suggested by the PCI Multimedia Design Guide rev 1.0). It is possible to configure the apertures to provide different views and color space translations for the apertures.

The graphics memory is accessible through a 32 Mbyte memory window which is located as specified by the *graphics_ram_BAR*. The uppermost bit of the address in the memory window selects the memory aperture which is used.

If the aperture which is selected is in the *raw* mode then the address is used to access the on-board graphics memory, in order to support memory configurations larger than 16 Mbytes the value of *aperture_start_addr* × 2048 is first added to it.

If the aperture which is selected is in the *linear frame buffer* mode then the address is first split to x-coordinate and y-coordinate values for the frame buffer (or texture map) memory. The splitting to coordinates is based on the *aperture_width* parameter.

The x-coordinate and y-coordinate are combined with the *aperture_start_addr* and *aperture_height* parameters to form the actual address which is used.

PCI Master Functions

The TR25202 can perform the following operations independently as a PCI master.

- Read sequences of triangle parameters for the rendering engine
- Upload data for textures and other images to the graphics memory
- Synchronize its operation to the operation of the rendering engine

System Control Registers

The system control registers contain registers which are used to control the PCI master functionality. Also some system debugging and state analysis registers are placed into this category.

PCI master control registers were originally placed to the PCI configuration space, but this placement offers a more portable high performance interface for accessing them.

This register set also contains the "extra io" registers which can be used to control the general purpose I/O pins of TR25202. These pins are used in a system dependent way.

System Control Register Definition

Address	Register
42	PCI Master Command Address
43	PCI Master State
44	PCI Master Internal Address
45	PCI Master External Address
48	status_reg
49	ref_reg
50	debug_reg
51	io_reg
52	extra_io_reg
54	Memory Aperture-0 Configuration
55	Memory Aperture-1 Configuration

System Control Register Map

Address	Register (bits 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0)
42	PCI Master Command Address
43	PCI Master State
	aa | master_counter
44	PCI Master Internal Address
45	PCI Master External Address
46	Not in use (reserved)
47	Not in use (reserved)
48	status_reg
	- | video_y_coord | a | b | c | d | e | f | g | h
49	ref_reg
	- | i | video_y_ref
50	debug_reg
51	io_reg
52	extra_io_reg
53	Not in use (reserved)
54	Memory Aperture-0 Configuration
	j | k | - | ab | - | ac | - | ad
55	Memory Aperture-1 Configuration
	l | m | ae | - | af | - | ag | - | ah

Abbreviation	Name
a	pci_master_interrupt_active
aa	master_state
ab	aperture_width
ac	aperture_height
ad	aperture_start_addr
ae	aperture_data_swap
af	aperture_width
ag	aperture_height
ah	aperture_start_addr
b	pixel_visible
c	pci_video_interrupt_active
d	video_compare
e	rasterizer_idle
f	ppu_idle
g	rasterizer_init_ok
h	ppu_init_ok
i	video_irq
j	aperture_raw_linear
k	aperture_16_32_bits
l	aperture_raw_linear
m	aperture_16_32_bits

System Control Register Description

- PCI Master Command Address (register:42)
- PCI Master State (register:43)
  - master_state (bits:31-24)

    When read this register provides debugging information about the current state of the PCI master unit. If written with value 00000000 halts the pci master non zero value starts the PCI master (assuming master enable bit is 1 and other)
  - master_counter (bits:23-0)
- PCI Master Internal Address (register:44)
- PCI Master External Address (register:45)
- status_reg (register:48)
  - video_y_coord (bits:26-16)
  - pci_master_interrupt_active (bit:15)

    This bit is set to one when the circuit has interrupt request active, if the interrupt has originated from the pci_master block. Interrupt can be caused either by the PCI master command control command or by the *video_y_ref* register. The interrupt is active until the device driver resets the interrupt. The interrupt is reset by writing value 1 into this register bit.
  - pixel_visible (bit:14)

    This bit is set to one when a visible pixel has been detected by the pixel processor in the Z-read operation. The bit is reset by writing value 1 into this register bit.
  - pci_video_interrupt_active (bit:13)

    This bit is set to one when the device has interrupt request active, if the interrupt has originated from the video_y comparator The interrupt is active until the device driver resets the interrupt. The interrupt is reset by writing value 1 into this register bit.
  - video_compare (bit:4)
  - rasterizer_idle (bit:3)
  - ppu_idle (bit:2)
  - rasterizer_init_ok (bit:1)
  - ppu_init_ok (bit:0)
- ref_reg (register:49)
  - video_irq (bit:11)

    If this bit is set to one the device will generate an interrupt request when the video_y value reaches the *video_y_ref* value.
  - video_y_ref (bits:10-0)
- debug_reg (register:50)
- io_reg (register:51)
- extra_io_reg (register:52)
- Memory Aperture-0 Configuration (register:54)
  - aperture_raw_linear (bit:31)

    0: the aperture is in raw mode.

    1: the aperture is in linear frame buffer mode.

aperture_16_32_bits, *aperture_width* and *aperture_height* are only used when the aperture is in the linear frame buffer mode.

- aperture_16_32_bits (bits:30)
  0: 16-bit mode (or packed YUV)
  1: 32-bit mode (RGB24+α pixel mode)
- aperture_width (bits:26-24)
  Used in splitting the X and Y coordinates from the memory address in linear mode. The value isthe number of bits in X coordinate. Values are translated as:

value	texture/screen width (pixels)
0	32
1	64
2	128
3	256
4	512
5	1024
6	2048
7	reserved

- aperture_height (bits:21-16)
- aperture_start_addr (bits:13-0)

• Memory Aperture-1 Configuration (register:55)
- aperture_raw_linear (bit:31)
  0: the aperture is in raw mode
  1: the aperture is in linear frame buffer mode
  aperture_16_32_bits, *aperture_width* and *aperture_height* are only used when the aperture is in the linear frame buffer mode.
- aperture_16_32_bits (bits:30)
  0: 16-bit mode (or packed YUV)
  1: 32-bit mode (RGB24+α pixel mode)
- aperture_data_swap (bits:29-28)
  Bits for supporting access from a big endian host to 8 or 16-bit pixels.
- aperture_width (bits:26-24)
  Used in splitting the X and Y coordinates from the memory address in linear mode. The value is the number of bits in X coordinate. Values are translated as:

value	texture/screen width (pixels)
0	32
1	64
2	128
3	256
4	512
5	1024
6	2048
7	reserved

- aperture_height (bits:21-16)
- aperture_start_addr (bits:13-0)

Video Interface

The current video interface unit contains the following registers:

Address	Register
33	video_width_height
34	screen_width_height
35	video_vblank
36	video_hblank
37	video_vsync
38	video_hsync
39	video_base_conf
40	video_bit_config

Video Interface Register Map

Address	Register (bits 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0)
33	video_width_height
	- (bits 31-22) | video_height (bits 21-11) | video_width (bits 10-0)
34	screen_width_height
	- (bits 31-22) | screen_height (bits 21-11) | screen_width (bits 10-0)
35	video_vblank
	- (bits 31-22) | vblank_start (bits 21-11) | vblank_end (bits 10-0)
36	video_hblank
	- (bits 31-22) | hblank_start (bits 21-11) | hblank_end (bits 10-0)
37	video_vsync
	- (bits 31-22) | vsync_start (bits 21-11) | vsync_end (bits 10-0)
38	video_hsync
	- (bits 31-22) | hsync_start (bits 21-11) | hsync_end (bits 10-0)
39	video_base_conf
	- (bits 31-22) | aa (bits 21-11) | - (bit 10) | screen_base_addr (bits 9-0)
40	video_bit_config
	- (bits 31-6) | a | b | c | d | e | f

Abbreviation	Name
a	dac_test
aa	screen_memory_height
b	hblank_polarity
c	vblank_polarity
d	hsync_polarity
e	vsync_polarity
f	pixel_width

Video Interface Register Description

- `video_width_height (register:33)`

  The `video_width_height` register specifies size of the area scanned by the video-x and video-y counters. The visible screen occupies a portion of this memory, starting from the (0,0)-point.
  - `video_height` (bits:21-11)

    The `video_width` register specifies the total width covered by the video-x counter
  - `video_width` (bits:10-0)

    The `video_height` register specifies the total height covered by the video-y counter

- `screen_width_height (register:34)`

  The `screen_width_height` register specifies the size of the actually displayed screen. Pixels are sent to the display while the values of video-x counter is less than the `screen_width` and while the video-y counter is less than the `screen_height`.
  - `screen_height` (bits:21-11)

    Specifies the width of the displayed screen.
  - `screen_width` (bits:10-0)

    Specifies the height of the displayed screen.

- `video_vblank (register:35)`

  Controlling the signal to the display requires a so called blank signal. The `video_vblank` register specifies the timing of the vertical blank signal relative to the video-y counter.
  - `vblank_start` (bits:21-11)

    Specifies the video-y counter value which starts the vertical blank signal
  - `vblank_end` (bits:10-0)

    Specifies the video-y counter value which ends the vertical blank signal. The blank area can be made to overlap the screen area. Also it is possible to initialize the `vblank_end` register to a lower value than

vblank_start. This causes the blank area to wrap around the bottom of the video coordinates.

- video_hblank (register:36)

  Controlling the signal to the display requires a so called blank signal. The video_hblank register specifies the timing of the horizontal blank signal relative to the video-x counter.
  - hblank_start (bits:21-11)

    Specifies the video-x counter value which starts the horizontal blank signal
  - hblank_end (bits:10-0)

    Specifies the video-x counter value which ends the horizontal blank signal. The blank area can be made to overlap the screen area. Also it is possible to initialize the hblank_end register to a lower value than hblank_start. This causes the blank area to wrap around the right edge of the video coordinates.
- video_vsync (register:37)

  Video displays require a synchronization signal. The video_vblank register specifies the timing of the vertical sync signal relative to the video-y counter.
  - vsync_start (bits:21-11)

    Specifies the video-y counter value which starts the vertical sync signal
  - vsync_end (bits:10-0)

    Specifies the video-y counter value which ends the vertical sync signal. The sync area can be made to overlap the screen area. Also it is possible to initialize the vsyn_end register to a lower value than vsync_start. This causes the sync area to wrap around the bottom edge of the video coordinates.
- video_hsync (register:38)
  - hsync_start (bits:21-11)

    Specifies the video-x counter value which starts the horizontal sync signal
  - hsync_end (bits:10-0)

    Specifies the video-x counter value which ends the horizontal sync signal. The sync area can be made to overlap the screen area. Also it is possible to initialize the hsync_end register to a lower value than hsync_start. This causes the sync area to wrap around the right edge of the video coordinates.
- video_base_conf (register:39)

  video_base_conf contains information about how the screen data are stored in the memory.
  - screen_memory_height (bits:21-15)

    Internally the screen memory is stored as blocks of 64*16 = 1024 pixels. The screen_memory_height can be calculated with the following formula:

    1024 * *height_as_16_pixel_blocks* * *pixel_size*

    For 16-bit pixels the *pixel_size* = 1, and for 32-bit pixels the *pixel_size* = 2.
  - screen_base_addr (bits:13-0)

    The screen_base_addr specifies the start address of the screen memory as a multiple of 2048 bytes.
- video_bit_config (register:40)

  Video_bit_config contains a collection of bits used to configure the behavior of the video interface.
  - dac_test (bit:5)

    Reserved for manufacturing testing
  - hblank_polarity (bit:4)

    Specifies the active (blank) value of the hblank signal.
  - vblank_polarity (bit:3)

    Specifies the active (blank) value of the vblank signal.
  - hsync_polarity (bit:2)

    Specifies the active (blank) value of the hsync signal
  - vsync_polarity (bit:1)

    Specifies the active (blank) value of the vsync signal
  - pixel_width (bit:0)

    Specifies whether 16-bit (=0) or 32-bit (=1) pixels are used.

Primitive Processor

The TR25202 has a primitive processor unit which operates at the triangle level. This means that after initialization it can autonomously rasterize a complete triangle. The primitive processor operates in the screen coordinate space, it produces the screen coordinates, Z-depth and up to 8 perspective correct interpolated values for each pixel. Of the 8 interpolated values four have an accuracy of 8 bits and are thus suitable for color and transparency values. 4 of the values have 12 bits of accuracy and are thus suitable to be used as texture coordinates.

Primitive Processor Registers

All the primitive processor registers are double buffered. This means that it is possible to initialize new values to the primitive processor, while the previous triangle is processed.

All primitive processor registers preserve their values when the triangle is rasterized. It is not necessary to reload values which do not change between consecutive triangles.

The primitive processor is signalled that the initialization is ready by writing a value to the YE register.

Primitive Processor Register Map

Address	Register (31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0)
64	cr_init
65	cr_dy
66	cr_dx
67	cg_init
68	cg_dy
69	cg_dx
70	cb_init
71	cb_dy
72	cb_dx
73	ct_init
74	ct_dy
75	ct_dx
76	atu_init
77	atu_dy
78	atu_dx
79	atv_init
80	atv_dy
81	atv_dx
82	btu_init
83	btu_dy
84	btu_dx
85	btv_init
86	btv_dy
87	btv_dx
88	zorig_shr
89	zbase_init
90	zbase_dy
91	zbase_dx
92	edgeorder
	– | aa | ab | ac | ad
93	edge0_init
94	edge0_dy
95	edge0_dx
96	edge1_init
97	edge1_dy
98	edge1_dx
99	edge2_init
100	edge2_dy

Address	Register
	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
101	edge2_dx
102	grid_reg
	- | a | b | c | d | e | f | - | g | h | i | j | - | RHIG
103	p_init
104	p_dy
105	p_dx
106	X_init
107	Y_init
108	YE
109	RASTER_ext
	- | k
110	Not in use (reserved)
111	Not in use (reserved)
112	Not in use (reserved)
113	Not in use (reserved)
114	Not in use (reserved)
115	Not in use (reserved)
116	Not in use (reserved)

Abbreviation	Name
a	B_qloop
aa	right_2
ab	right_1
ac	left_2
ad	left_1
b	B_not(Vmsb)
c	B_not(Umsb)
d	A_qloop
e	A_not(Vmsb)
f	A_not(Umsb)
g	grid11
h	grid10
i	grid01
j	grid00
k	Soft_reset

Primitive Processor Register Descriptions

- `cr_init (register:64)`
  Specifies the initial value of the RED value integrator in the primitive processor.
- `cr_dy (register:65)`
  Specifies the increment which is added the RED value integrator, when the primitive processor steps one pixel in Y-direction.
- `cr_dx (register:66)`
  Specifies the increment which is added the RED value integrator, when the primitive processor steps one pixel in X-direction.
- `cg_init (register:67)`
  Specifies the initial value of the GREEN value integrator in the primitive processor.
- `cg_dy (register:68)`
  Specifies the increment which is added the GREEN value integrator, when the primitive processor steps one pixel in Y-direction.
- `cg_dx (register:69)`
  Specifies the increment which is added the GREEN value integrator, when the primitive processor steps one pixel in X-direction.

- `cb_init (register:70)`
  Specifies the initial value of the BLUE value integrator in the primitive processor.
- `cb_dy (register:71)`
  Specifies the increment which is added the BLUE value integrator, when the primitive processor steps one pixel in Y-direction.
- `cb_dx (register:72)`
  Specifies the increment which is added the BLUE value integrator, when the primitive processor steps one pixel in X-direction.
- `ct_init (register:73)`
  Specifies the initial value of the TRANSPARENCY value integrator in the primitive processor.
- `ct_dy (register:74)`
  Specifies the increment which is added the TRANSPARENCY value integrator, when the primitive processor steps one pixel in Y-direction.
- `ct_dx (register:75)`
  Specifies the increment which is added the TRANSPARENCY value integrator, when the primitive processor steps one pixel in Y-direction.
- `atu_init (register:76)`
  Specifies the initial value of the A-TEXTURE U-COORDINATE value integrator in the primitive processor.
- `atu_dy (register:77)`
  Specifies the increment which is added the A-TEXTURE U-COORDINATE value integrator, when the primitive processor steps one pixel in Y-direction.
- `atu_dx (register:78)`
  Specifies the increment which is added the A-TEXTURE U-COORDINATE value integrator, when the primitive processor steps one pixel in X-direction.
- `atv_init (register:79)`
  Specifies the initial value of the A-TEXTURE V-COORDINATE value integrator in the primitive processor.
- `atv_dy (register:80)`
  Specifies the increment which is added the A-TEXTURE V-COORDINATE value integrator, when the primitive processor steps one pixel in Y-direction.
- `atv_dx (register:81)`
  Specifies the increment which is added the A-TEXTURE V-COORDINATE value integrator, when the primitive processor steps one pixel in X-direction.
- `btu_init (register:82)`
  Specifies the initial value of the B-TEXTURE U-COORDINATE value integrator in the primitive processor.
- `btu_dy (register:83)`
  Specifies the increment which is added the B-TEXTURE U-COORDINATE value integrator, when the primitive processor steps one pixel in Y-direction.
- `btu_dx (register:84)`
  Specifies the increment which is added the B-TEXTURE U-COORDINATE value integrator, when the primitive processor steps one pixel in X-direction.
- `btv_init (register:85)`
  Specifies the initial value of the A-TEXTURE V-COORDINATE value integrator in the primitive processor.
- `btv_dy (register:86)`
  Specifies the increment which is added the B-TEXTURE V-COORDINATE value integrator, when the primitive processor steps one pixel in Y-direction.
- `btv_dx (register:87)`
  Specifies the increment which is added the B-TEXTURE V-COORDINATE value integrator, when the primitive processor steps one pixel in X-direction.
- `zorig_shr (register:88)`
  Scaling factor for the Z-depth calculations
- `zbase_init (register:89)`
  The initial value for the Z-depth register

- zbase_dy (register:90)
  The value which is added to the Z-depth value, when the primitive processor steps one pixel in the Y-direction
- zbase_dx (register:91)
  The value which is added to the Z-depth value, when the primitive processor steps one pixel in the X-direction
- edgeorder (register:92)
  - right_2 (bits:7-6)
    Specifies which of the edges is used as the second right edge of the triangle
  - right_1 (bits:5-4)
    Specifies which of the edges is used as the first right edge of the triangle
  - left_2 (bits:3-2)
    Specifies which of the edges is used as the second left edge of the triangle
  - left_1 (bits:1-0)
    Specifies which of the edges is used as the first left edge of the triangle
- edge0_init (register:93)
  The initial value of edge detector 0
- edge0_dy (register:94)
  The value which is added to the edge detector 0 value, when the primitive processor steps one pixel in the Y-direction
- edge0_dx (register:95)
  The value which is added to the edge detector 0 value, when the primitive processor steps one pixel in the X-direction
- edge1_init (register:96)
  The initial value of edge detector 1
- edge1_dy (register:97)
  The value which is added to the edge detector 1 value, when the primitive processor steps one pixel in the Y-direction
- edge1_dx (register:98)
  The value which is added to the edge detector 1 value, when the primitive processor steps one pixel in the X-direction
- edge2_init (register:99)
  The initial value of edge detector 2
- edge2_dy (register:100)
  The value which is added to the edge detector 2 value, when the primitive processor steps one pixel in the Y-direction
- edge2_dx (register:101)
  The value which is added to the edge detector 2 value, when the primitive processor steps one pixel in the X-direction
- grid_reg (register:102)
  - B_qloop (bit:21)
  - B_not(Vmsb) (bit:20)
    Inverts the most significant bit (msb) of B texture interpolator V coordinate.
  - B_not(Umsb) (bit:19)
    Inverts the most significant bit (msb) of B texture interpolator U coordinate.
  - A_qloop (bit:18)
  - A_not(Vmsb) (bit:17)
    Inverts the most significant bit (msb) of A texture interpolator V coordinate.
  - A_not(Umsb) (bit:16)
    Inverts the most significant bit (msb) of A texture interpolator U coordinate.
  - grid11 (bit:11)
  - grid10 (bit:10)
  - grid01 (bit:9)
  - grid00 (bit:8)
  - RHIG (bits:5-0)
- p_init (register:103)
  The initial value for the perspective register
- p_dy (register:104)
  The value which is added to the perspective value, when the primitive processor steps one pixel in the Y-direction
- p_dx (register:105)
  The value which is added to the perspective value, when the primitive processor steps one pixel in the X-direction
- X_init (register:103)
  The initial X-coordinate where the triangle rasterization starts.

- `Y_init (register:104)`
  The initial Y-coordinate where the triangle rasterization starts. The actual area covered by the triangle depends on the values of the `X_init` and `Y_init` registers and on the edge parameters.
- `YE (register:105)`
  The Y-coordinate which ends the rasterization process when it is reached. `YE` has an additional function where writing to it signals the primitive processor should start rasterizing a new triangle. This register should be written last during the initialization of the primitive processor.
- `RASTER_ext (register:106)`
  - `Soft_reset` (bit:0)
    If written with value 1 performs a soft reset for the primitive processor, aborts the current triangle and starts the rasterization of the next triangle (if any).

Pixel Processor

The Pixel processing consists of pixel pipelines which process the pixels provided by the primitive processor. The TR25202 pixel pipelines perform Z-buffering and handle point-sampled, bilinear and trilinear textures with mipmapping.

The texture data and the Gouraud color data can be combined flexibly using the pixel pipeline.

Pixel Processor Registers

Registers	Address
0	not used
1	coef_reg0
2	coef_reg1
3	coef_reg2
4	coef_reg3
5	Atex_conf1
6	Atex_conf2
7	Btex_conf1
8	Btex_conf2
9	base_addr
10	dither
11	modulation
12	ppu_mode
13	frame_mode
14	ppu_code_start_addr
15	palette_base

Pixel Processor Register Map

Address	Register (31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0)
0	not used
1	coef_reg0
	alpha | red | green | blue
2	coef_reg1
	alpha | red | green | blue
3	coef_reg2
	alpha | red | green | blue
4	coef_reg3
	alpha | red | green | blue
5	Atex_conf1
	- | amhig | - | abaseb
6	Atex_conf2
	aa | - | a | b | c | - | d | e | ab | - | ac | - | ad
7	Btex_conf1
	- | bmhig | - | bbaseb
8	Btex_conf2
	ae | - | f | g | h | - | i | j | af | - | ag | - | ah
9	base_addr
	- | zbaseb | - | cbaseb
10	dither
	- | ai | aj | ak | al | am | an | ao | ap
11	modulation
	modvy | modvx | modhy | modhx
12	ppu_mode
	- | aq | k | l | m | n | o | -
13	frame_mode
	- | p | ar | q | r | s | t | u
14	ppu_code_start_addr
	- | as
15	palette_base
	at | au | av | aw
16	Not in use (reserved)

Abbreviation	Name
a	ayloop
aa	asubs
ab	amode
ac	aphig
ad	apwid
ae	bsubs

Abbreviation	Name
af	bmode
ag	bphig
ah	bpwid
ai	ditha7
aj	ditha6
ak	ditha5
al	ditha4

Abbreviation	Name
am	ditha3
an	ditha2
ao	ditha1
ap	ditha0
aq	stenoper
ar	zmemm
as	start_addr
at	B_pal_mask
au	B_pal_base
av	A_pal_mask
aw	A_pal_base
b	axloop
c	amip
d	acent
e	adouble
f	byloop
g	bxloop
h	bmip
i	bcent
j	bdouble
k	stenokd
l	stencil
m	transkp
n	dithsrg
o	nodith
p	cmemm
q	fastccv
r	fastcen
s	zequal
t	ocheck
u	rasttr

Pixel Processor Register Descriptions

- not used (register:0)
- coef_reg0 (register:1)
  - alpha (bits:31-24)
  - red (bits:23-16)
  - green (bits:15-8)
  - blue (bits:7-0)
- coef_reg1 (register:2)
  - alpha (bits:31-24)
  - red (bits:23-16)
  - green (bits:15-8)
  - blue (bits:7-0)
- coef_reg2 (register:3)
  - alpha (bits:31-24)
  - red (bits:23-16)
  - green (bits:15-8)
  - blue (bits:7-0)
- coef_reg3 (register:4)
  - alpha (bits:31-24)
  - red (bits:23-16)
  - green (bits:15-8)
  - blue (bits:7-0)
- Atex_conf1 (register:5)
  - amhig (bits:21-16)
  - abaseb (bits:13-0)

    Base address of A-texture, measured in units of 2048 bytes. Notice that the A/B texture is selected by the texture selection bit in the ppu instruction, not by the primitive processor interpolator used as the texture address source.
- Atex_conf2 (register:6)

  Parameters for A-texture
  - asubs (bits:31-29)
  - ayloop (bit:26)
  - axloop (bit:25)
  - amip (bit:24)
  - acent (bit:21)
  - adouble (bit:20)
  - amode (bits:19-16)
  - aphig (bits:10-8)
  - apwid (bits:2-0)
- Btex_conf1 (register:7)
  - bmhig (bits:21-16)

    Notice that the A/B texture is selected by the texture selection bit in the ppu instruction, not by the primitive processor interpolator used as the texture address source.
  - bbaseb (bits:13-0)

    Base address of B-texture, measured in units of 2048 bytes

- Btex_conf2 (register:8)
  Parameters for B-texture
  - bsubs (bits:31-29)
  - byloop (bit:26)
  - bxloop (bit:25)
  - bmip (bit:24)
  - bcent (bit:21)
  - bdouble (bit:20)
  - bmode (bits:19-16)
  - bphig (bits:10-8)
  - bpwid (bits:2-0)
- base_addr (register:9)
  - zbaseb (bits:29-16)
    Base address for z-buffer measured in units of 2048 bytes
  - cbaseb (bits:13-0)
    Base address for graphics memory measured in units of 2048 bytes
- dither (register:10)
  - ditha7 (bits:23-21)
  - ditha6 (bits:20-18)
  - ditha5 (bits:17-15)
  - ditha4 (bits:14-12)
  - ditha3 (bits:11-9)
  - ditha2 (bits:8-6)
  - ditha1 (bits:5-3)
  - ditha0 (bits:2-0)
- modulation (register:11)
  - modvy (bits:31-24)
  - modvx (bits:23-16)
  - modhy (bits:15-8)
  - modhx (bits:7-0)
- ppu_mode (register:12)
  - stenoper (bits:10-9)
    00: no operation
    01: stencil set
    10: stencil clear
    11: stencil invert
  - stenokd (bit:8)
  - stencil (bit:7)
  - transkp (bit:6)
  - dithsrg (bit:5)
  - nodith (bit:4)
- frame_mode (register:13)
  - cmemm (bit:8)
  - zmemm (bits:7-5)
  - fastccv (bit:4)
  - fastcen (bit:3)
  - zequal (bit:2)
  - ocheck (bit:1)
  - rasttr (bit:0)
- ppu_code_start_addr (register:14)
  - start_addr (bits:4-0)
    The location of the first instruction in the microcode of the pixel processing unit
- palette_base (register:15)
  Register contains information used in indexing the internal texture palette memory
  - B_pal_mask (bits:31-24)
    Bit mask for B texture palette fetches
  - B_pal_base (bits:23-16)
    The base index for B texture palette fetches
  - A_pal_mask (bits:15-8)
    Bit mask for A texture palette fetches
  - A_pal_base (bits:7-0)
    The base index for A texture palette fetches

Pixel Processor Memory Blocks

Pixel processing unit contains a program memory containing 32 32-bit words. This memory is mapped to the address range 128-159.

Pixel processing unit also contains memory for storing the color palette used in some of the texture map modes. This memory is mapped to the address range 256-511.

Summary of Registers

Address	Register
0	not used
1	coef_reg0
2	coef_reg1
3	coef_reg2
4	coef_reg3
5	Atex_conf1
6	Atex_conf2
7	Btex_conf1
8	Btex_conf2
9	base_addr
10	dither
11	modulation
12	ppu_mode

Address	Register
13	frame_mode
14	ppu_code_start_addr
15	palette_base
33	video_width_height
34	screen_width_height
35	video_vblank
36	video_hblank
37	video_vsync
38	video_hsync
39	video_base_conf
40	video_bit_config
42	PCI Master Command Address
43	PCI Master State
44	PCI Master Internal Address
45	PCI Master External Address
48	status_reg
49	ref_reg
50	debug_reg
51	io_reg
52	extra_io_reg
54	Memory Aperture-0 Configuration
55	Memory Aperture-1 Configuration
64	cr_init
65	cr_dy
66	cr_dx
67	cg_init
68	cg_dy
69	cg_dx
70	cb_init
71	cb_dy
72	cb_dx
73	ct_init
74	ct_dy
75	ct_dx

Address	Register
76	atu_init
77	atu_dy
78	atu_dx
79	atv_init
80	atv_dy
81	atv_dx
82	btu_init
83	btu_dy
84	btu_dx
85	btv_init
86	btv_dy
87	btv_dx
88	zorig_shr
89	zbase_init
90	zbase_dy
91	zbase_dx
92	edgeorder
93	edge0_init
94	edge0_dy
95	edge0_dx
96	edge1_init
97	edge1_dy
98	edge1_dx
99	edge2_init
100	edge2_dy
101	edge2_dx
102	grid_reg
103	p_init
104	p_dy
105	p_dx
106	X_init
107	Y_init
108	YE
109	RASTER_ext

TR25202 Features

Rendering

- Full 10-component RGB model support (diffuse RGB, specular RGB, transparency, fog, texture u1/v1)
- Radiosity hardware support
- Fully programmable pixel processor
- Perspective correct true-color shading
- Perspective correct transparency
- Perspective correct texture mapping
- Multiple simultaneous textures
- Environment mapping
- Bump mapping
- Stencil operations
- Logic operations
- Specular highlights
- Properly handled lighted textures
- Rasterized screen door transparency
- Destination blending for transparency effects
- Fog and depth cue

Textures

- Texture magnification filtering with point sampling or filtering
- Texture minification filtering with point sampling or MIP mapping
- Bilinear and Trilinear filtering supported
- Texture sizes from 32x32 pixels to 1024x1024 pixels
- Amount of texture maps limited only by available memory
- Textures can be looped or clamped
- RGB map formats: 32-bit RGBA and 16-bit RGB and 16-bit RGBA
- Indexed map formats: 8-bit and 4-bit
- Indexed maps have an internal 256 color 32-bit palette (RGBA)
- Full blending and filtering possible with indexed maps
- Real time texture paging and animation
- Rendering to texture maps supported

Memory

- 2-32 MBytes of SDRAM, SGRAM or EDO DRAM supported
- Memory bus width 64-bits or 32-bits
- Memory bandwidth up to 800 MBytes/sec with 64-bit bus
- All geometry and texture data can be stored in local memory
- Rendering resolutions up to 2048 x 2048 pixels
- 24-bit or 16-bit color (dithering supported)
- 24-bit or 16-bit depth buffer
- 1 bit stencil mask
- Frame buffer can be accessed in RGB or YUV format
- Support for double and triple buffering and stereo imaging

Video Refresh

- Display resolutions from 320 x 200 to 1600 x 1200 pixels
- Internal video refresh logic
- Internal programmable clock generator (up to 200MHz)
- Internal true-color DAC (up to 200MHz pixel clock)

Physical Characteristics

- 32-bit PCI version 2.1, bus master and target without glue logic
- 304-pin thermally enhanced BGA packaging
- 100 MHz operation
- I/O interface at 3.3V / 5V

Compatibility

- Drivers for Microsoft Windows 95 and Windows NT
- Drivers for DirectDraw and Direct3D
- Drivers for OpenGL for Windows NT
- Drivers for Kinetix 3D Studio MAX (Heidi) and AutoCAD

Peak Performance

- 1,300,000 shaded, 16bpp textured triangles/sec
- 1,000,000 shaded, 16bpp textured, Z-buffered triangles/sec
- Triangles are given as visible hicolor 25 pixel triangles
- Pixel fill rate 50,000,000 pixels/sec

Absolute maximum Ratings

Beyond these limits damage may occur to the device.

Symbol	Parameter	Min	Max	Units
V	Supply Voltage	-0.25	6.5	V
Ts	Storage Temperature	-40	125	°C

Recommended DC Operating Conditions

Valid for 25°C ambient temperature and 5V supply unless otherwise stated.

Symbol	Parameter	Min	Typ	Max	Units
Vdd1	Supply Voltage	4.75	5.0	5.25	V
Vdd2	Supply Voltage	3.0	3.3	3.6	V
Avd	Analog Supply Voltage	4.75	5.0	5.25	V
CLK	Crystal Frequency		14.3181818		MHz

General Specifications

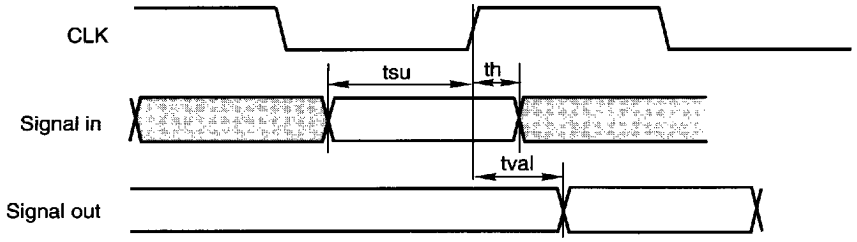
Symbol	Parameter	Condition	Min	Max	Units
V_{il}	TTL input LO	Vdd = 5V		0.8	V
V_{ih}	TTL input HI	Vdd = 5V	2.0		V
I_{il}	Input leakage	V_i = Vdd or Vss	-10	10	µA
V_{ol}	Low Level Output	I_{ol} = 6mA, Vdd = 5V		0.55	V
V_{oh}	High Level Output	I_{oh} = -2mA, Vdd = 5V	2.4		V
V_{ol}	Low Level Output	I_{ol} = 1500uA, Vdd = 3.3V		0.1Vdd	
V_{oh}	High Level Output	I_{oh} = -500uA, Vdd = 3.3V	0.9Vdd		
I_{oz}	High Z leakage	V_o = Vdd or Vss	-10	10	µA

Electrical/Timing Specifications

Symbol	Parameter	Condition	Min	Typ	Max	Units
I_{VDD}	Digital Supply Current (using CMOS -level clock)	Power up RESET=Logic **0**		300	TBA	mA
I_{AVD}	Analog Supply Current	Power up RESET=Logic **0**		40	TBA	mA
I_{VDDPD}	Digital Supply Current	Power down RESET=Logic **1**		1		µA
I_{AVDPD}	Analog Supply Current	Power down RESET=Logic **1**		1		µA

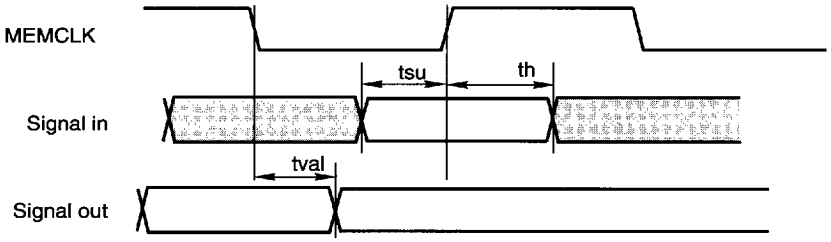
PCI Interface

The PCI interface is designed to be compatible to PCI Local Bus Specification revision 2.1.



Symbol	Parameter	Min	Max	Units
tsu	Input set up time to CLK			
	bused signals	7		ns
	point to point	10		ns
th	Input hold time from CLK	0		ns
tval	CLK to output valid delay			
	bused signals	2	11	ns
	point to point	2	12	ns

Memory Interface

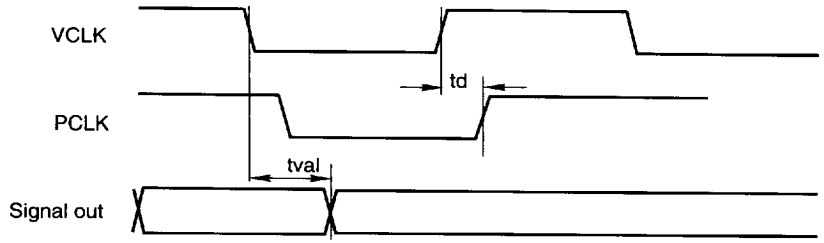


All timings are relative to the `memclk` created by TR25202.

Symbol	Parameter	Min	Max	Units
tsu	Input set up time to CLK	1		ns
th	Input hold time from CLK	1		ns
tval	CLK fall to output valid delay	0	1	ns

The memory interface is designed to be compatible with SGRAM and SDRAM devices with clock frequencies up to 100MHz.

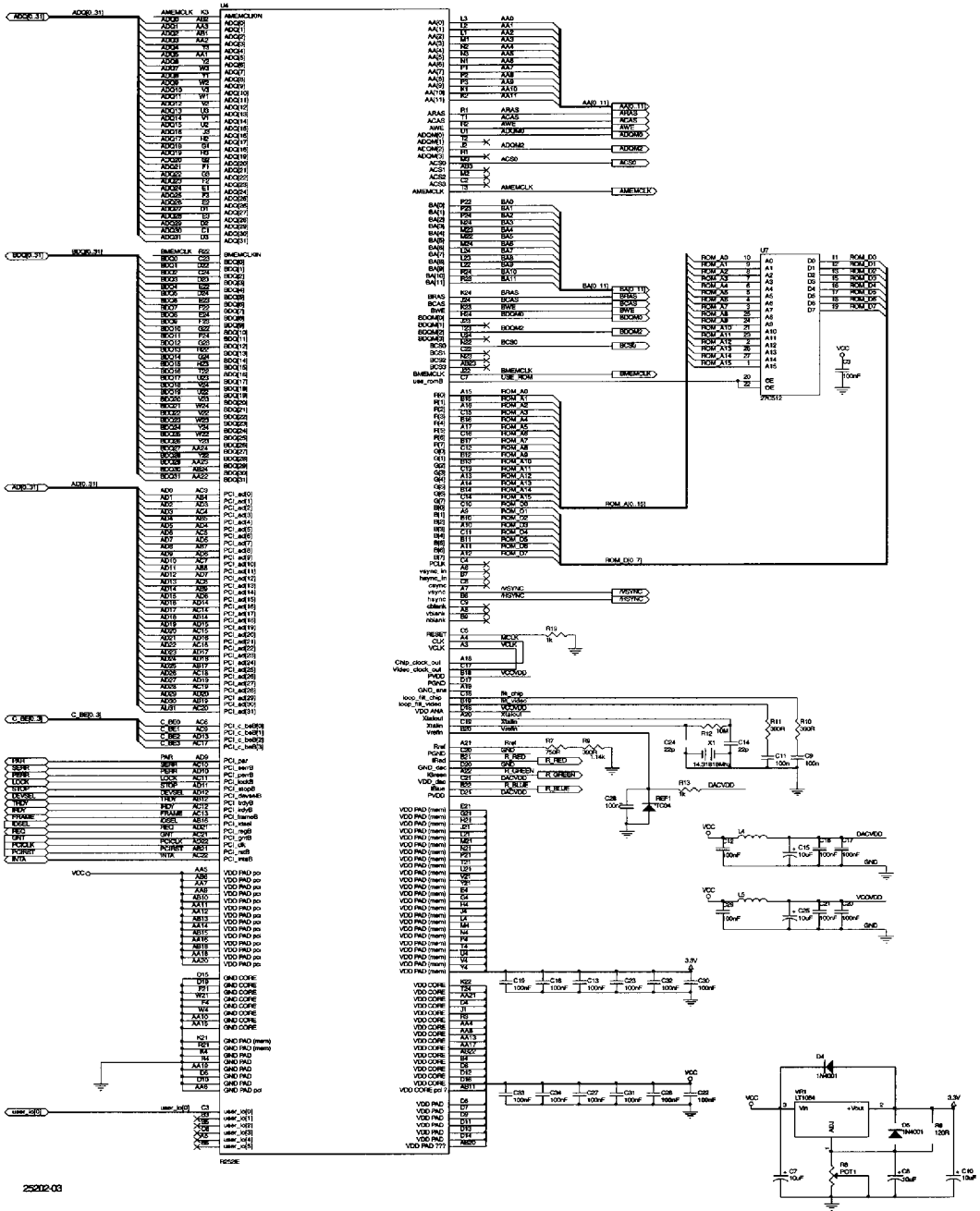
Video Interface



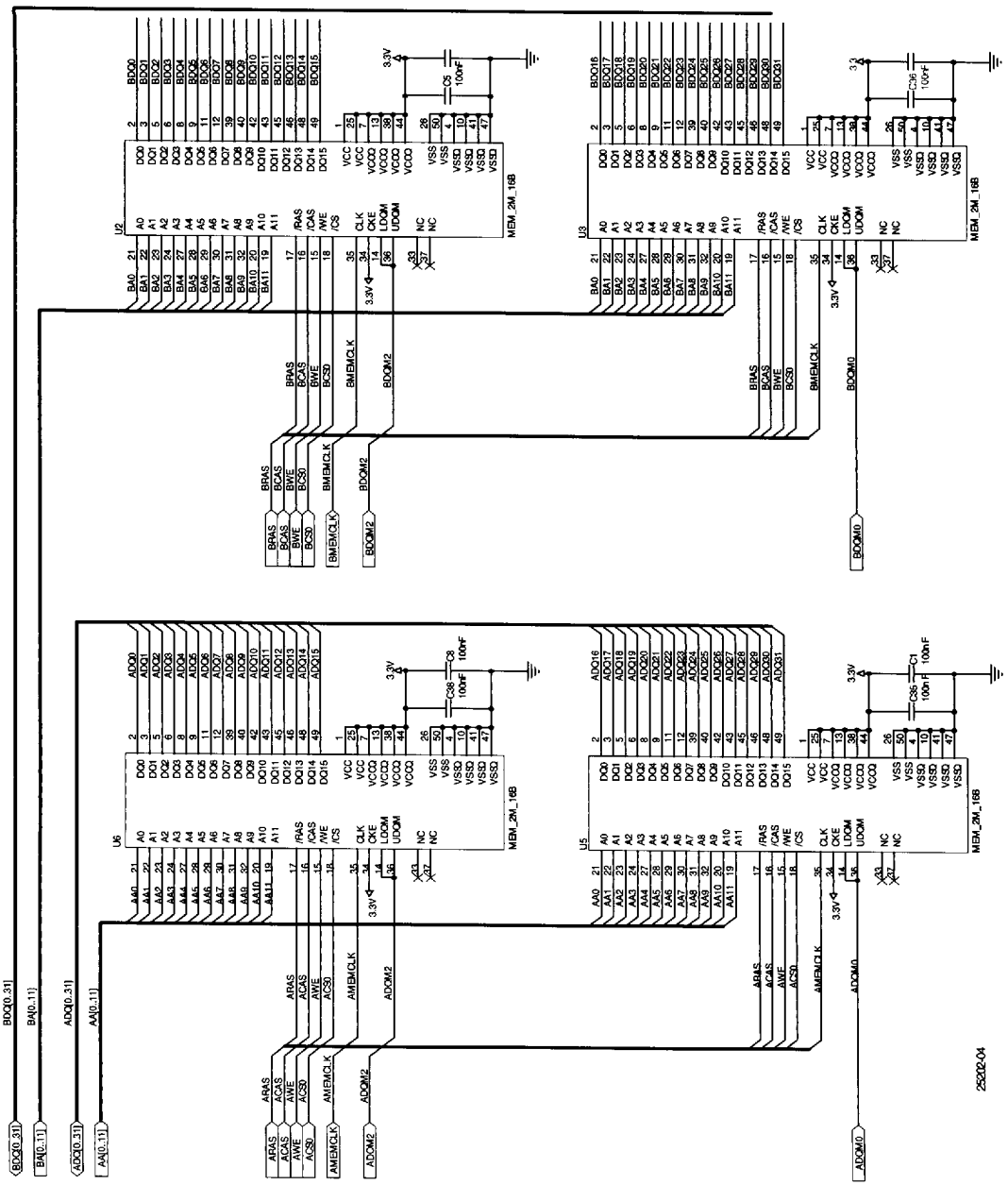
All timing are relative to the `memclk` created by TR25202.

Symbol	Parameter	Min	Max	Units
td	PCLK delay from VCLK	0	3	ns
tval	CLK fall to output valid delay	0	1	ns

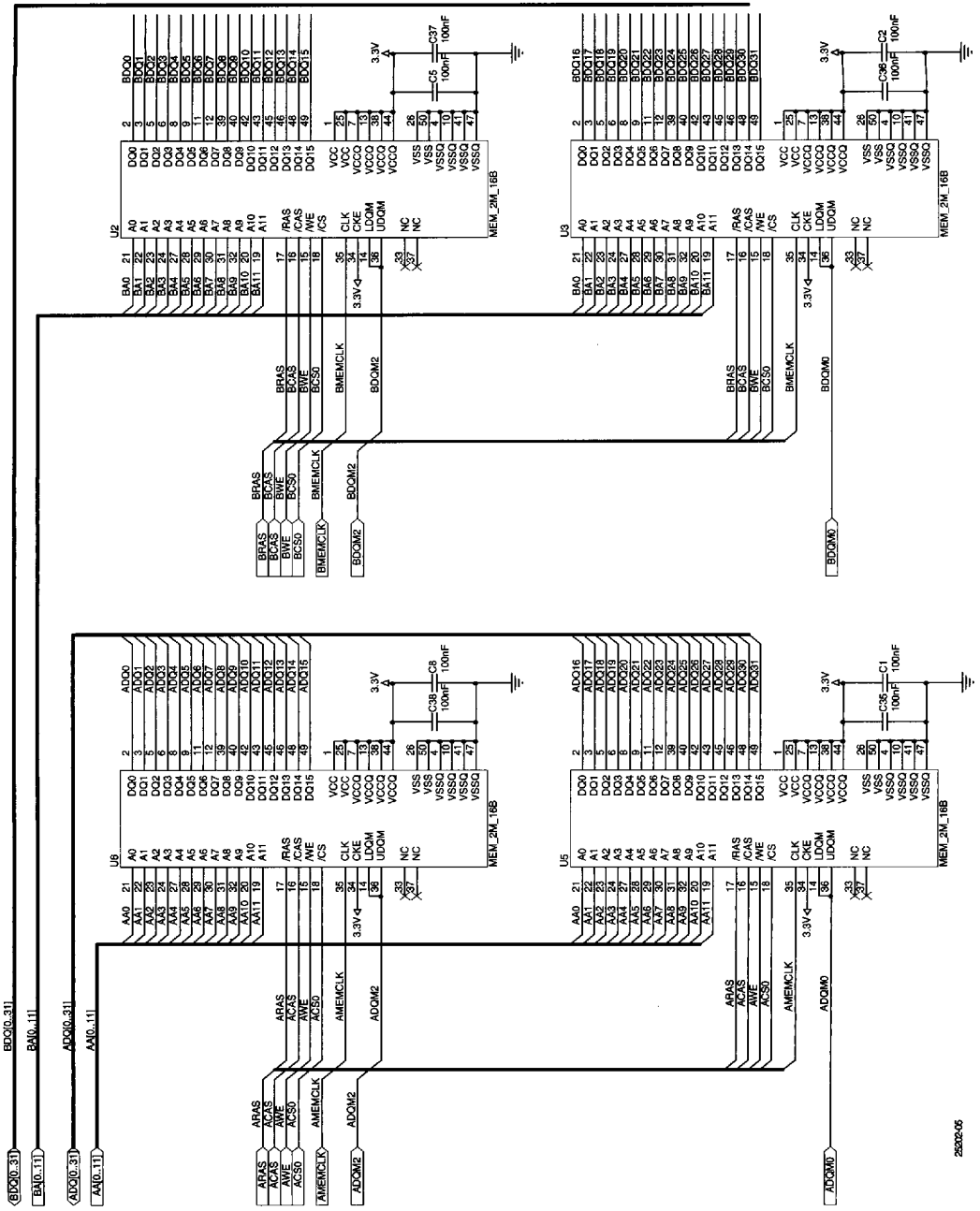
Schematics



Schematics (continued)



Schematics (continued)



9010734 0000102 610

30101D00-01