LSI LOGIC

LR33020
GraphX Processor
Preliminary

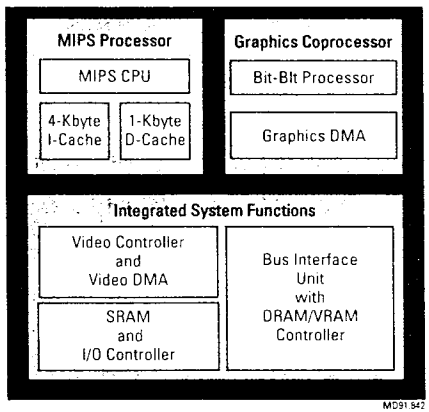
Introduction

The LR33020 GraphX Processor – a member of LSI Logic's LR33000 family of Self-Embedding™ processors – offers an integrated MIPS-compatible Processor, a Graphics Coprocessor and a sophisticated group of Integrated System Functions. The LR33020's MIPS Processor executes up to 33 MIPS (sustained) at 40 MHz, and its Graphics Coprocessor renders 180K filled rectangles per second (10 x 10-pixel, 8-bit color). With LSI Logic's new Metal Quad Flat Pack (MQUAD) packaging, the LR33020 provides high performance, superior integration and low cost. The LR33020's three main subsystems contain:

MIPS Processor – A 32-bit MIPS-compatible CPU with integrated instruction cache and data cache.

Graphics Coprocessor – A programmable Bit-Blt processor with source and destination DMA supporting bit-boundary block transfers (*Bit-Blts*), polygon draw and polygon fill operations. The Graphics Coprocessor's instruction set is very well tuned for high-level language use, in contrast to many other machines labeled RISC, which have user-level instructions or instruction mode combinations that are very difficult to reach from compiled languages.

Integrated System Functions – A unified block including a Bus Interface Unit (BIU) with an integrated DRAM/VRAM Controller, a programmable Video Controller, four internal DMA channels, external DMA support, one Serial Port and two Counter/Timers.



LR33020 High-Level Block Diagram

These tightly coupled subsystems provide a "system on a chip" tailored for low cost, high performance, graphics-oriented applications such as laser printers, X Window System terminals, graphics subsystems, digital scanners, video processors, X-ray and infra-red imaging systems, image enhancement and display systems, and video game consoles.

Features

- Thirty-two-bit RISC CPU that is compatible with the MIPS-I architecture
- 4-Kbyte direct-mapped instruction cache
- 1-Kbyte direct-mapped data cache
- LR2000/LR3000/LR33000 user software binary compatibility
- Advanced hardware debugging features, including address and data breakpoint registers, and instruction trace support
- Graphics Coprocessor supporting all X Window System graphics operations
- Bit-Blt Processor supports 16 logical operations, including AND, OR, NOR, XOR, invert and fill operations
- Two dedicated graphics DMA channels for burst-read and burst-write pixel access
- I/O Controller with four I/O select regions and programmable wait states
- One PS/2-compatible serial port configurable as two input and two output ports
- One 24-bit general purpose timer and one 12-bit DRAM refresh timer
- Programmable DRAM/VRAM Controller with support for single-cycle access to 64-bit interleaved DRAM
- Bus Interface Unit with a 4-word write buffer, byte steering logic for 8-bit and 32-bit boot PROM support, and a bus arbiter
- Programmable Video Controller, including a 32-word video FIFO and serializer, a direct RAM-DAC interface with endian conversion and two dedicated video DMA channels
- Dedicated video DMA channel to directly support DRAM-based frame buffers
- Hardware cursor support for VRAM-based frame buffers

Features (Continued)

- Internal 32-bit address and data buses; external 64-bit data bus
- Single 1x clock input
- 25/ 33/40 MHz versions available in 208-pin MQUAD or 224-pin CPGA

Block Diagram

Figure 1 shows the functional block diagram of the LR33020.

The LR33020's "system on a chip" approach integrates the following units onto a single device:

MIPS Processor
The 32-bit MIPS CPU is compatible with the MIPS-I architecture. The CPU maintains binary compatibility with the LR2000, LR3000 and LR33000 user software.

On-chip caches are integrated with the MIPS CPU. A 4-Kbyte direct-mapped instruction cache provides instructions to the CPU and on-chip Graphics Coprocessor. A 1-Kbyte direct-mapped local data cache provides data to the CPU. Graphics data is not cached to improve cache performance.

An Integrated System Control Coprocessor (CP0) provides efficient access to exception handling, configuration and debugging registers. Separate address and data breakpoint registers, as well as instruction trace support, are included in the LR33020.

Graphics Coprocessor
The Graphics Coprocessor is an integrated Bit-Blt Processor and Graphics DMA Controller that provides hardware acceleration and support for common 2-D graphics operations, including functions to support bit-mapped alphanumeric displays and the X Window System standard.

The Bit-Blt Processor can perform a wide range of Bit-Blt functions; including opaque, transparent and semi-transparent block transfer; line draw; and area fill operations. Instructions are snooped from the Embedded MIPS CPU cache and step data through the pipeline deterministically, ensuring correct recovery from interrupts and exceptions even while graphics operations are taking place.

The optimized opcode set includes single instructions that read source data and write destination data simultaneously. Optional burst read and block write facilities reduce memory bus traffic.

Two dedicated graphics DMA channels provide pixel access transparently to the CPU, with burst-read and burst-write support for improved bus bandwidth.

Integrated System Functions
The LR33020 includes an additional wide range of functions, which are summarized below.

- *Memory and I/O Controller* – A programmable DRAM/VRAM Controller provides glueless access to 32-bit DRAM or single-cycle glueless access to 64-bit interleaved DRAM. An I/O Controller provides four I/O select regions with programmable wait states.
- *Bus Interface Unit* – A four-word-deep write buffer improves bus access. Glueless 8-bit and 32-bit boot PROM support provides for low-cost and large-scale PROM requirements. A Data Bus Translator coordinates data movement across the two 32-bit data buses. A Bus Arbiter coordinates bus access by the DRAM refresh circuitry, the four DMA channels, the Core CPU and external bus masters.
- *Video Controller* – A programmable video timing controller provides composite blank, separate sync and composite sync signals. A 32-word video data FIFO and serializer supports a hardware cursor when used with VRAM or a direct interface to a RAMDAC when used with DRAM. Two dedicated video DMA channels perform VRAM refresh and support a hardware cursor.
- *Counter/Timers* – One 24-bit general purpose timer and one 12-bit DRAM Refresh timer are directly compatible with the memory-mapped counter/timers provided in the LR33000.
- *Serial Ports* – Four pins on the LR33020 can either be used as a single PS/2 serial interface or two standard input and two standard output ports. The PS/2 serial interface option provides for direct attachment of a keyboard. The standard input and output ports can be used to connect more devices to the LR33020.

The next sections in this document describe each of the above functional blocks in more detail.

Block Diagram
(Continued)

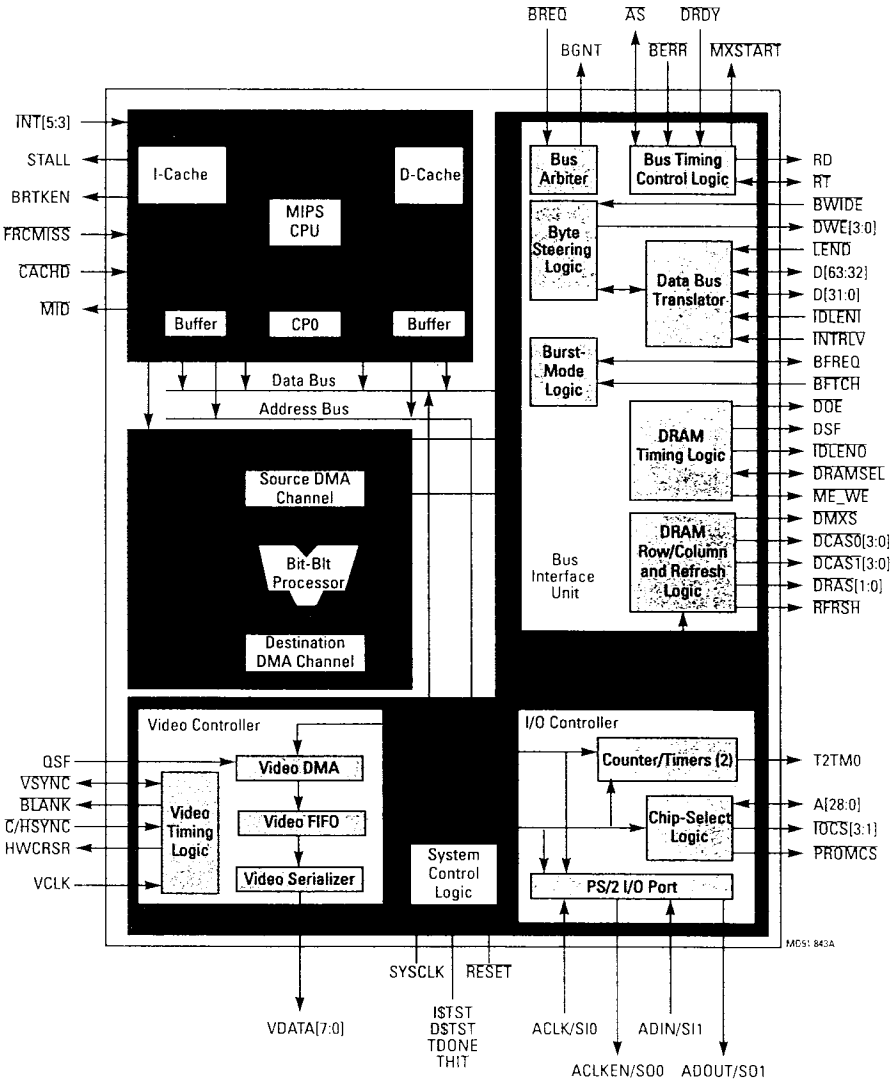


Figure 1. LR33020 Block Diagram

MIPS Processor

The LR33020 implements the high performance MIPS architecture.

- **Instruction Set Compatibility** – The LR33020 maintains full user binary compatibility with the R2000, R3000, R3000A and LR33000 microprocessors.
- **Efficient Pipelining** – The CPU's five-stage pipeline design assists in obtaining an execution rate approaching one instruction per cycle. Pipeline stalls and exceptional events are handled precisely and efficiently.
- **Full 32-bit Operation** – The LR33020 contains thirty-two 32-bit registers. All instructions and addresses are 32 bits wide to provide a 4-Gbyte address space.
- **Enhanced Debug Features** – The LR33020 incorporates program trace logic and hardware breakpoints on the program counter and data address registers to ease software debugging.
- **On-Chip Cache Memory** – The LR33020 contains a 4-Kbyte instruction cache and a 1-Kbyte data cache. The processor can access both caches during a single clock cycle, which allows the processor to execute one instruction per cycle when executing from cache memory. Both caches employ direct address mapping. The data cache uses a write-through technique to maintain main memory coherency.

The following subsections describe the essential features of the MIPS CPU, including its CPU registers, instruction set, system control processor (CP0), operating modes, pipeline architecture and on-chip caches.

CPU Registers

The LR33020 CPU provides 32 general purpose 32-bit registers, a 32-bit Program Counter and two 32-bit registers (HI and LO). The HI and LO registers hold the results of integer multiply and divide operations. The CPU registers are shown in Figure 2. Notice that the LR33020 does not contain a Program Status Word (PSW) register; the functions traditionally provided by a PSW register are instead provided by the Status and Cause Registers incorporated within the system control coprocessor (CP0).

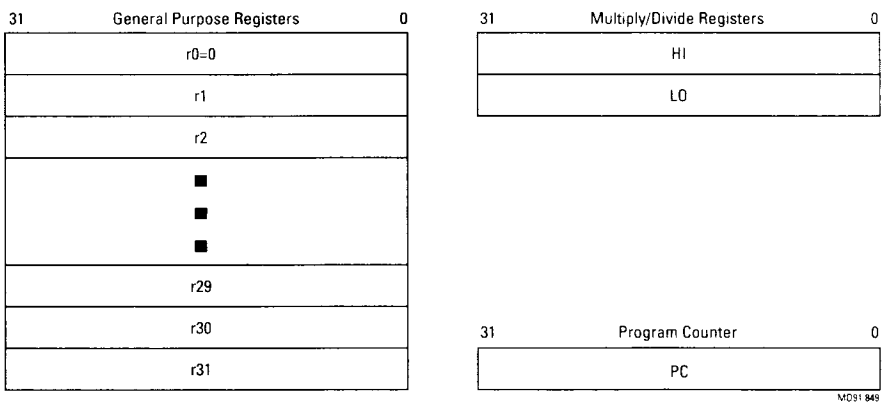


Figure 2. LR33020 CPU Registers

MIPS Processor
(Continued)

CPU Instruction Set Overview

All LR33020 instructions are 32 bits long. As shown in Figure 3, instructions have three basic formats: I-type (immediate), J-type (jump) and R-type (register).

Instruction decoding is simplified by low number of fixed register formats. More complicated (and less frequently used) operations and addressing modes can be synthesized by the compiler using sequences of simple RISC-type instructions. The assembler recognizes some CISC-type instructions for programming ease, but translates them into sequences of simple instructions.

The LR33020 instruction set is divided into the following groups:

- **Load/Store** instructions are the only instructions that move data between memory and general registers. These instructions are I-type, since the only addressing mode supported is base register plus 16-bit, signed immediate offset.
- **Computational** instructions perform arithmetic, logical and shift operations on values in registers. These instructions are both R-type (both operands and the result are registers) and I-type (one operand is a 16-bit immediate).
- **Jump** and **Branch** instructions change the control flow of a program. Jumps are always to a paged, absolute address formed by combining a 26-bit target with four bits of the program counter (J-type format, for subroutine calls), or 32-bit register byte addresses (R-type, for returns and dispatches). Branches have 16-bit offsets relative to the program counter (I-type). Jump and Link instructions save a return address in register *r31*.
- **Coprocessor** instructions perform operations in the coprocessors.
- **Coprocessor 0** instructions perform operations on the system control coprocessor (CP0) registers to manipulate the exception handling facilities of the processor.
- **Coprocessor 2** instructions perform operations on the Graphics Coprocessor (CP2) registers to set integrated system function parameters and control graphics operations.
- **Special** instructions perform system call and breakpoint operations. These instructions are always R-type.

Table 1 lists the instruction set of the LR33020 processor.

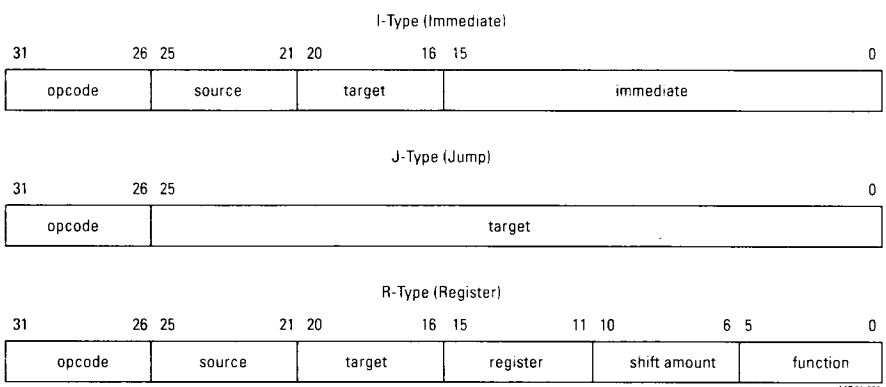


Figure 3. Instruction Types and Instruction Fields

MIPS Processor
(Continued)

Table 1. Instruction Summary

Op	Description
Load/Store Instructions	
LB	Load Byte
LBU	Load Byte Unsigned
LH	Load Halfword
LHU	Load Halfword Unsigned
LW	Load Word
LWL	Load Word Left
LWR	Load Word Right
SB	Store Byte
SH	Store Halfword
SW	Store Word
SWL	Store Word Left
SWR	Store Word Right
Arithmetic Instructions: ALU Immediate	
ADDI	Add Immediate
ADDIU	Add Immediate Unsigned
SLTI	Set on Less Than Immediate
SLTIU	Set on Less Than Immediate Unsigned
ANDI	AND Immediate
ORI	OR Immediate
XORI	Exclusive OR Immediate
LUI	Load Upper Immediate
Arithmetic Instructions: 3-Operand, Register-Type	
ADD	Add
ADDU	Add Unsigned
SUB	Subtract
SUBU	Subtract Unsigned
SLT	Set on Less Than
SLTU	Set on Less Than Unsigned
AND	AND
OR	OR
XOR	Exclusive OR
NOR	NOR
Shift Instructions	
SLL	Shift Left Logical
SRL	Shift Right Logical
SRA	Shift Right Arithmetic
SLLV	Shift Left Logical Variable
SRLV	Shift Right Logical Variable
SRAV	Shift Right Arithmetic Variable

Op	Description
Multiply/Divide Instructions	
MULT	Multiply
MULTU	Multiply Unsigned
DIV	Divide
DIVU	Divide Unsigned
MFHI	Move From HI
MTHI	Move To HI
MFLO	Move From LO
MTLO	Move To LO
Jump and Branch Instructions	
J	Jump
JAL	Jump And Link
JR	Jump Register
JALR	Jump And Link Register
BEQ	Branch on Equal
BNE	Branch on Not Equal
BLEZ	Branch on Less than or Equal to Zero
BGTZ	Branch on Greater Than Zero
BLTZ	Branch on Less Than Zero
BGEZ	Branch on Greater than or Equal to Zero
BLTZAL	Branch on Less Than Zero And Link
BGEZAL	Branch on Greater than or Equal to Zero And Link
Special Instructions	
SYSCALL	System Call
BREAK	Breakpoint
Coprocessor Instructions	
BCzT	Branch on Coprocessor z True
BCzF	Branch on Coprocessor z False
COP2	Coprocessor 2 Instruction
CFCz	Move from CPz Control Register
CTCz	Move to CPz Control Register
RFE	Restore From Exception
MTCz	Move To CPz Data Register
MFCz	Move From CPz Data Register
LWCz	Load word to CPz Data Register
SWCz	Store word from CP2 Data Register

System Control Coprocessor (CP0)
The LR33020's system control coprocessor (CP0) supports exception handling functions of the LR33020. Figure 4 summarizes the register set.

CPU Operating Modes
To facilitate the separation of user and supervisory software, the LR33020 has two operating modes: *user* and *kernel*. Normally, the processor operates in the user mode until an exception is detected, which forces it into the kernel mode. It remains in the kernel mode until a Restore From Exception (RFE) instruction is executed. Figure 5 shows the address space for the two operating modes.

User Mode – In user mode, a single, uniform address space (kuseg) of 2 Gbytes is available.

Kernel Mode – Four separate segments are defined in kernel mode:

- *kuseg* –References to this segment are treated just like user-mode addresses, thus streamlining kernel access to user data

MIPS Processor
(Continued)

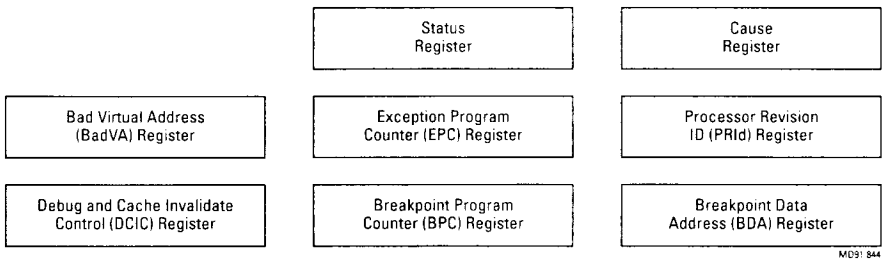


Figure 4. The CP0 Exception-Handling Registers

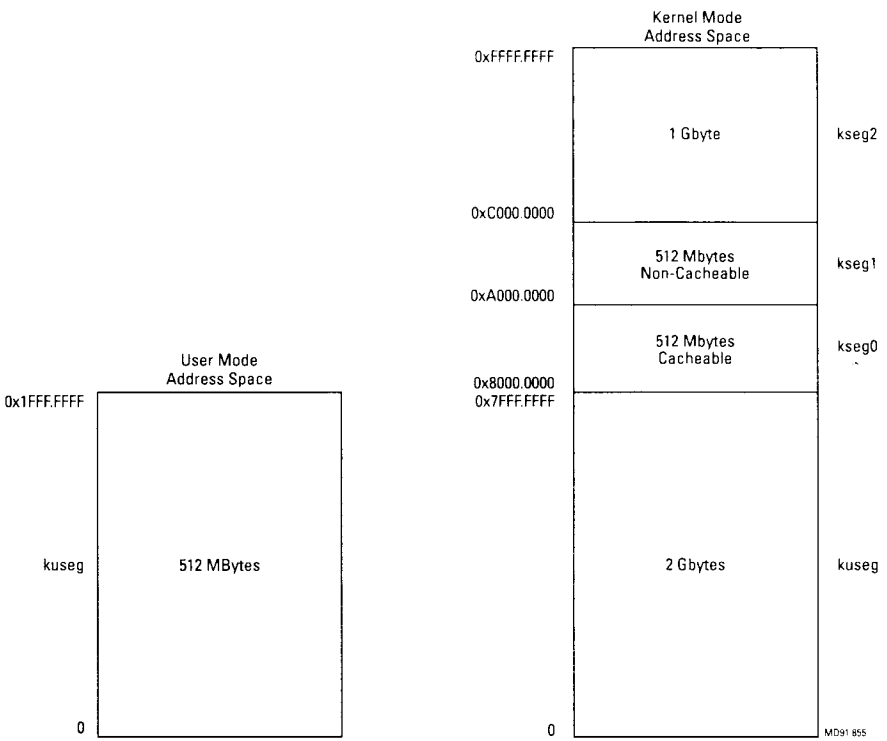


Figure 5. LR33020 Address Space

MIPS Processor (Continued)

- *kseg0* – References to this 512-Mbyte segment use cache memory and are hard-mapped to the first 512 Mbytes of memory.
- *kseg1* – References to this 512-Mbyte segment do not use the cache and are hard-mapped into the same 512-Mbyte segment of memory space as *kseg0*.
- *kseg2* – References to this 1-Gbyte segment are not mapped and the cacheability of data and instructions is controlled on a per-access basis by the $\overline{\text{CACHD}}$ signal. Note that the 29-bit external addressing space in the LR33020 only supports accesses to the lower 512 Mbytes. Accesses to higher addresses with the LR33020 are invalid and will cause the upper three bits to be ignored.

CPU Pipeline Architecture
The execution of a single LR33020 instruction consists of five primary steps:

1. **IF** – Fetch the instruction from the instruction cache (I-Cache).
2. **RD** – Read any required operands from CPU registers while decoding the instruction (register fetch, RF).
3. **ALU** – Perform the required arithmetic or logical operation on instruction operands (OP).
4. **MEM** – Access memory from the data cache (D-Cache).
5. **WB** – Write results back to register file (WB).

Each of these steps requires approximately one CPU cycle as shown in Figure 6 (parts of some operations overlap into another cycle while other operations require only 1/2 cycle).

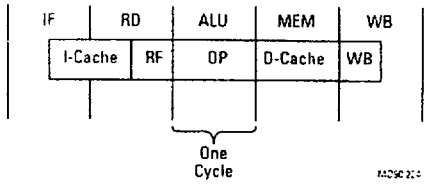


Figure 6. Instruction Execution Sequence

The LR33020 uses a 5-stage pipeline to achieve an instruction execution rate approaching one instruction per CPU cycle. Thus execution of five instructions at a time are overlapped as shown in Figure 7.

On-Chip Cache Memories
The LR33020 contains a 4-Kbyte instruction cache and a 1-Kbyte data cache. Both caches are directly mapped to the LR33020's address space. The cache memories hold instructions and data that are repetitively accessed by the CPU (for example, within a program loop) and thus reduce the number of references that must be made to the slower main memory.

To provide an additional performance advantage, the LR33020 has separate data and

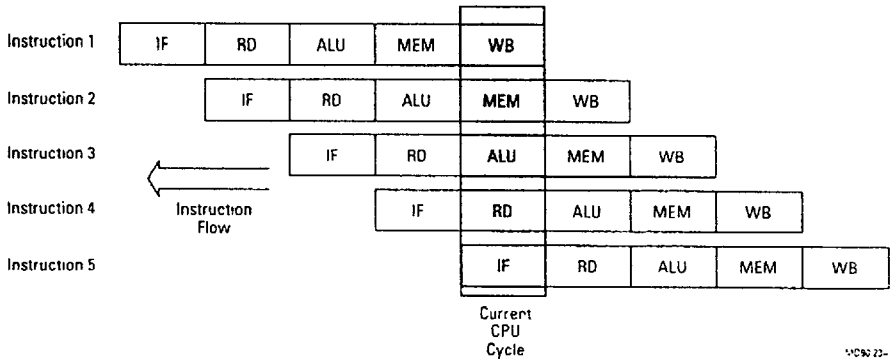


Figure 7. LR33020 Instruction Pipeline

MIPS Processor (Continued)

instruction buses to allow the processor to access both caches during a single clock cycle when executing from cache memory. Figure 1 (page 3) shows the buses. The dual buses allow the processor to execute one instruction per cycle, even during load and store operations.

The LR33020 has several features that ensure coherency between its caches and main memory. The LR33020 uses bus snooping when another external system device is performing DMA. During bus snooping, the LR33020 monitors the address bus, and when it detects an address that matches the address for a valid entry in the cache, it invalidates that entry to ensure that the data therein does not become stale. To ensure that the main memory is consistent with the data cache, the LR33020 uses a write-through technique. Whenever the LR33020 executes a write and there is a data cache hit, that data is also written through to main memory. In addition to these automatic cache coherency features, software can invalidate cache entries by executing store instructions with the cache invalidate enable bits in the DCR Register set to one.

The LR33020 supports block refill for both caches. Refill block sizes can be set to two, four, eight or 16 words, and the refill sizes can be set independently for each cache, assuring maximum performance for the target system.

Graphics Coprocessor

The LR33020 Graphics Coprocessor provides hardware-assisted pixel block-transfer and area-fill capabilities via a 32-bit datapath. In addition the Graphics Coprocessor provides source skew alignment, transparency mask capabilities, color expansion, source plane masking, 16 logical raster operations, and destination boundary and pixel masking. Supported raster operations include combinations of superimposing, inverting and overwriting operations.

The LR33020 design allows Bit-Blt functions to proceed as fast as the DRAM or VRAM system can support. To use the Graphics Coprocessor, fast-page-mode RAM is required. Higher performance can be obtained by using 64-bit interleaving, VRAM masking and block write capabilities. In optimal conditions, the LR33020 can render 180K filled rectangles per second (10 x 10-pixel, 8-bit-color).

When using the Graphics Coprocessor, low-level graphic routines are performed via a combination of CPU instructions and dedicated graphics hardware. In general, low-level graphic routines should be written in assembly language to take full advantage of the Graphics Coprocessor. LSI Logic provides a basic set of graphics routines; users are free to extend this basic set or write their own functions.

To program the Graphics Coprocessor, the user first loads the CP2 registers with the desired values. The LR33020 then uses special instructions, referred to as STEP instructions, to move data through its graphics pipeline. The STEP instructions themselves only fetch the next word of data or store the processed word of data, although flags associated with the STEP instructions can activate masks. The values in the Graphics Coprocessor registers define which operations are performed on the pixel data.

Figure 8 shows a high-level block diagram of the Graphics Coprocessor. The main functions are described below.

- The *Source Data Graphics DMA Channel* fills the four-word *Source Data Queue* directly from memory (bypassing the LR33020 data cache). The dedicated STEP instructions can fetch up to four words in a single block fetch. The DMA channel can calculate the source data address across lines and line wraps.
- The *Graphics Grouper* contains four blocks that each process 32 pixel bits every clock cycle when a STEP instruction is executed.
  - The *Pixel Extractor* takes individual 32-bit words from the Data Source Queue and combines data bits in the required fashion to build pixel data. If the source and destination pixel data are not aligned on the same word or inter-word boundary, the current and previous word of pixel data are combined. If the source and destination data do not fall on the same word boundaries, a bar-

Graphics Coprocessor
(Continued)

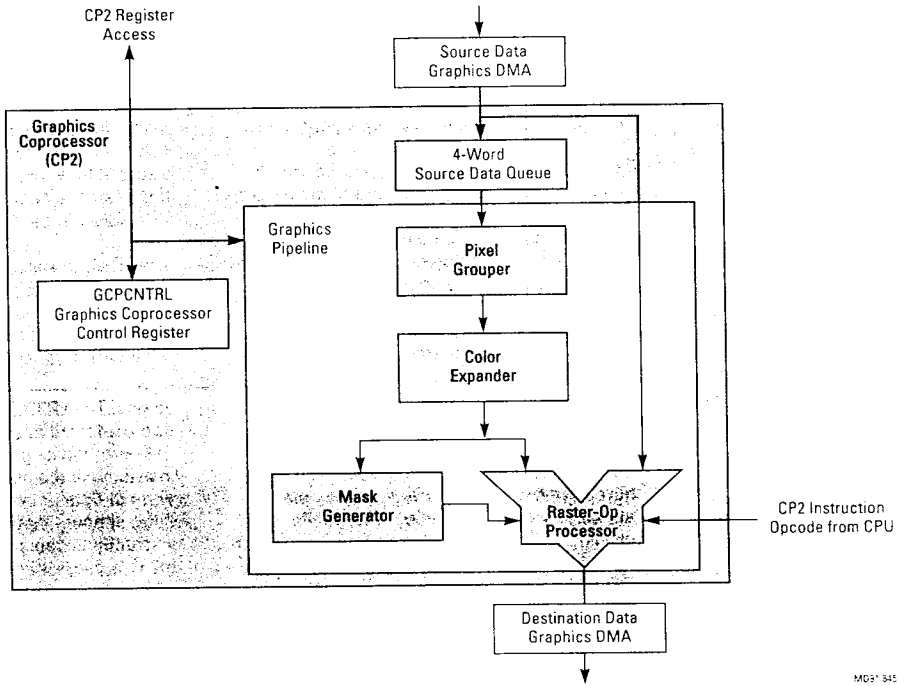


Figure 8. Graphics Coprocessor High-Level Block Diagram

rel shifter can rotate the word so that the pixels are aligned correctly.

- The *Color Expander* can add color data to single-bit pixel maps. For example, ASCII characters are often represented with bit maps. The color expander can expand any single bit into a 2-bit, 4-bit, 8-bit, 16-bit or 32-bit color.
- After pixel extraction and color expansion, there may be stray bits which should not be written to the RAM or which would interfere with the subsequent logical operation by the Raster-Op Processor. The *Mask Generator* can remove these unwanted bits from the pixel data. The Mask Generator also screens out extraneous bits in data words on the left and right edges of a line or block boundary.
- The *Raster-Op Processor* combines or replaces the destination pixel color with the new pixel color. The Raster-Op Processor supports all 16 X Window Raster Logical Operations, including a range of AND, OR and Exclusive-OR combinations. These logical operations provide the combinations of overwrite, superimposition and inversion supported by the Graphics Coprocessor.

■ The *Destination Graphics DMA Channel* stores data to main memory. Systems using VRAM block write can store up to four words of pixel data in a burst write. Also, in DRAM systems, a constant source can be written to four consecutive locations with a burst-write instruction, which accelerates fill operations.

Graphics Coprocessor (Continued)

CP2 Instructions
The Graphics Coprocessor receives instructions from the CPU instruction stream using the MIPS CoProcessor 2 (CP2) interface.

The coprocessor instructions supported include:

- Instructions for loading and reading coprocessor registers
- Instructions for sequencing of pixel data through the Bit-Blt datapath

Loading and Reading Coprocessor Registers – The MIPS instruction set provides six dedicated instructions for loading and reading coprocessor registers. Software uses these instructions to control and configure the Graphics Coprocessor

Sequencing Instructions – The LR33020 adds four dedicated COP2 instructions for controlling the Graphics Coprocessor. These instructions – STEP, SBSTEP, BSTEP and WSTEP – all step data through the Graphics Coprocessor datapath. The four sequencing instructions include five flags that set various conditions.

Table 2 shows the ten MIPS instructions that control the CP2 coprocessor and summarizes their function.

All these instructions are single-cycle instructions. Instructions that access memory may, however, require more cycles to complete if the memory is not cached.

A typical block move or block fill operation has four phases:

1. The software first uses the CP2 load and store instructions (described above) to load the Graphics Coprocessor control registers with descriptions of the pixel format, colors and so on.

2. The software initiates a DMA transfer by loading the Graphics DMA Controller with source address, destination addresses, pitch and block size for the block transfer or block fill operation. The Graphics DMA Controller then loads the Graphics Coprocessor source queue with up to four words of pixel data.

Table 2. LR33020 CP2 Instructions

Description	Type	Function
CFC2 *rt, rd*	CPU	Loads CPU general-purpose register *rt* from CP2 control register *rd*.
CTC2 *rt, rd*	CPU	Stores CPU general-purpose register *rt* to CP2 control register *rd*.
LWC2 *rt, offset (base)*	CPU	Loads CP2 data register *rt* from the memory location pointed to by ***base*** plus *offset*.
MFC2 *rt, rd*	CPU	Loads CPU general-purpose register *rt* from CP2 data register *rd*.
MTC2 *rt, rd*	CPU	Stores CPU general-purpose register *rt* to CP2 data register *rd*.
SWC2 *rt, offset (base)*	CPU	Stores CP2 data register *rt* to the memory location pointed to by *base* plus *offset*.
BSTEP, S(B), L, R	COP2	Steps to the beginning of the next destination line and writes the DEST register back to memory. If the S flag is set, an SSTEP instruction is also performed prior to the BSTEP operation. If the SB flag is set, an SBSTEP instruction is also performed prior to the BSTEP operation. If the L flag is set, the left mask is applied. If the R flag is set, the right mask is applied. If the RMW flag is set in the Graphics Configuration Register, then read/modify/write cycles are used for the destination data.
SBSTEP	COP2	Clears the Source Queue, steps to the beginning of the next source line and initiates another DMA fetch of pixel data.
SSTEP	COP2	Steps to the next word in the Source Queue. If the Source Queue is empty, initiates another DMA fetch of pixel data.
WSTEP, S(B), L, R, (B)FOUR	COP2	Writes the DEST register back to memory. If the S flag is set, an SSTEP instruction is also performed prior to the WSTEP operation. If the SB flag is set, an SBSTEP instruction is also performed. If the L flag is set, the left mask is applied. If the R flag is set, the right mask is applied. If the FOUR flag is set, the LR33020 writes the DEST register to four consecutive words in DRAM/VRAM via a burst-write operation. If the BFOUR flag is set, the LR33020 uses the VRAM block write feature to write the DEST register once to four consecutive words simultaneously. If the RMW flag is set in the Graphics Configuration Register, then read/modify/write cycles are used for the destination data.

Graphics Coprocessor (Continued)

3. Words of data are processed by the Graphics Coprocessor when the CPU issues a dedicated STEP instruction to CP2. The STEP instructions take pixel data from the source queue and pass the data through the Graphics Coprocessor datapath. Once a STEP instruction leaves the MEM stage of the CPU pipeline, it is executed by the graphics coprocessor in parallel with subsequent CPU instructions.

4. When the pixel data is processed, the Graphics DMA Controller stores the pixel data at the destination address.

As is clear from the above description, graphics data is moved through the Graphics Coprocessor by dedicated instructions. Source data is loaded and destination data is stored by dedicated instructions. Although this increases the number of instructions required for a graphics operation, this technique has the advantage of allowing deterministic handling on interrupts and exceptions. If the Graphics Coprocessor operated entirely via DMA channels that operated independently of processor control, it would not be possible to determine exactly when an interrupt or exception interceded on an ongoing graphics operation. In many system implementations, interrupts and exceptions do terminate ongoing graphics operations in an indeterminate manner, resulting in partly completed screen rendering and necessitating screen redraw. With the LR33020's dedicated graphics instructions, a graphics operation can recommence at exactly the right point after exception or interrupt servicing.

If an interrupt or exception alters the Graphics Controller control or data registers, then the original state should be saved and restored after servicing the interrupt or exception for undisturbed process execution. If the interrupts and exceptions do not affect the Graphics Coprocessor, then the CP2 data and control registers do not need to be saved and restored.

Integrated System Functions

As well as an embedded MIPS CPU and Graphics Coprocessor, the LR33020 uses a "system on a chip" strategy to incorporate the following system functions:

- SRAM and I/O Controller – I/O Select, Timers and Serial Ports
- DRAM/VRAM Controller
- Bus Interface Unit – Write Buffer, Data Translator, Byte Steering Logic and Bus Arbiter
- Video Controller

The following subsections describe the functionality of each of the above blocks.

SRAM and I/O Controller
The LR33020 SRAM and I/O Controller provides four I/O-select regions with separate wait-state-generation for each region, two timers and a serial port. These functions are described briefly below.

I/O Select Regions – The I/O select signals are called $\overline{\text{PROMCS}}$, $\overline{\text{IOCS1}}$, $\overline{\text{IOCS2}}$ and $\overline{\text{IOCS3}}$. These signals are activated when a memory request is generated for the associated address region. Each I/O select address space is 16 Mbytes. The access time used for each byte or word is programmable from 0 to 15 wait states. Figure 9 shows the SRAM and I/O Controller chip-select logic.

Timers – The LR33020 provides one general-purpose 24-bit timer/counter and a 12-bit counter for DRAM refresh. The registers for controlling and monitoring counter/timer operation are memory-mapped. Figure 5 shows the LR33020 Counter/Timer logic. Timer 2 generates the internal interrupt signal $\overline{\text{INT1}}$ and the Timer 2 Timeout signal (T2TMO).

Serial Port – The LR33020 has one PS/2-compatible serial port which can also function as two standard input ports and two standard output ports. Figure 11 shows the Serial Port Logic.

The PS/2 port may be used as a direct interface for a keyboard or graphical entry device that is compatible with the PS/2 standard.

The serial ports are standard unidirectional ports. When the Standard Serial Port Input Register is read, it returns the current value on the two standard input ports, SI0 and SI1. When the Standard Serial Port Output Port Register is written, it changes the values on the two standard output ports, SO0 and SO1.

Integrated System Functions
(Continued)

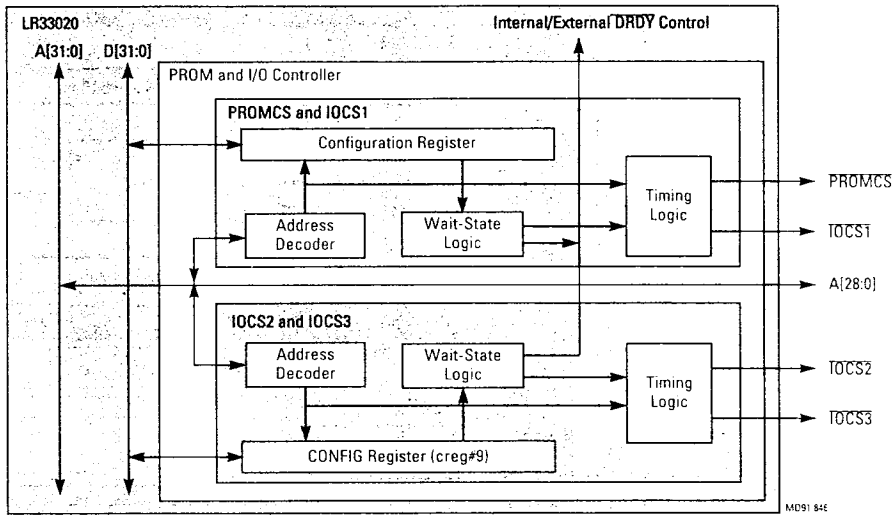


Figure 9. LR33020 SRAM and I/O Controller Chip-Select Logic

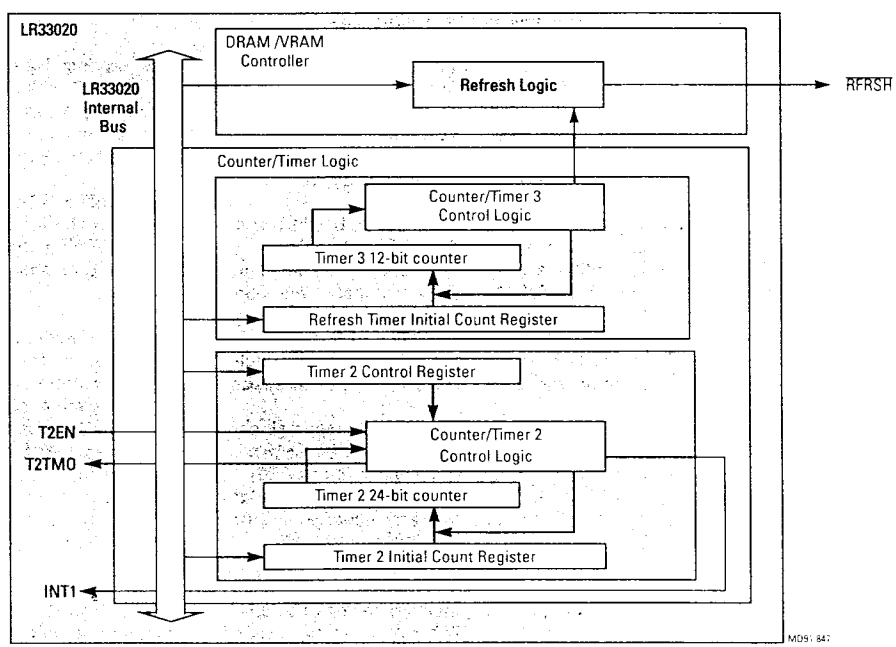


Figure 10. LR33020 Counter/Timer Logic

Integrated System Functions
(Continued)

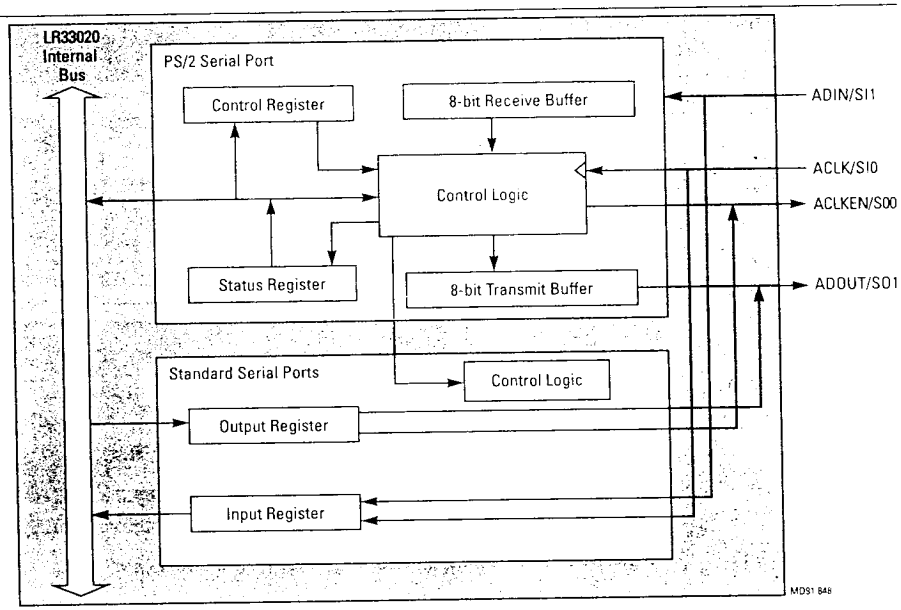


Figure 11. LR33020 Serial Port Logic

DRAM/VRAM Controller

The LR33020 includes virtually all the logic required for a complete interface for up to four banks of 32-bit-wide DRAM and/or VRAM.

The DRAM/VRAM Controller can support:

- One 32-bit-wide non-interleaved DRAM bank or one 64-bit-wide interleaved DRAM bank
- An additional 32-bit-wide non-interleaved bank of DRAM or VRAM, or an additional 64-bit-wide interleaved bank of DRAM or VRAM
- A fixed address range of 0x0000.0000 to 0x1FFF.FFFF is supported by the DRAM Controller to provide a 256-Mbyte memory array space

The DRAM/VRAM Controller can support single cycle access to DRAM via an interleaved two-bank memory array system using the $\overline{\text{DRAS0}}$, $\overline{\text{DRAS1}}$ and $\overline{\text{INTRLV}}$ signals. Writes to individual bytes in each of the banks are supported by four $\overline{\text{CAS}}$ signals to each bank ($\overline{\text{DCAS0}}$[3:0] and $\overline{\text{DCAS1}}$[3:0]).

The size of the lower memory bank can be any size from 1 Mbyte to 128 Mbytes. The addresses of the upper memory banks can be contiguous with the lower addresses.

Bus Interface Unit

The LR33020 includes a Bus Interface Unit (BIU) consisting of a write buffer, data translator, byte steering unit and a bus arbiter. A description of these functions follows.

Write Buffer – The BIU includes a four-word-deep write buffer to prevent the CPU from stalling on STORE instructions to memory. The write buffer is especially efficient during burst write operations.

If the write buffer is full and the CPU executes a write to external memory, the processor stalls until the buffer partly empties. If an entry is located in the cache, the cache is updated. Graphics Coprocessor write cycles do *not* use this write buffer. Because this write buffer does not provide for coherency checking with bus read cycles, it is flushed before a CPU data read cycle is allowed to run. No coherency checking is performed for memory reads from the DMA controllers.

The write buffer includes a programmable page comparator which retires consecutive accesses quickly, removing the need to insert

Integrated System Functions
(Continued)

extra cycles for $\overline{DRAS}$ deassertion during consecutive writes to the same DRAM page.

Data Translator – The BIU includes a data translator that allows the LR33020 to work with one or two 32-bit data buses. The data translator also handles distribution of even and odd addresses across interleaved memory banks.

Byte Steering Unit – The Byte Steering Unit assembles 32-bit words from 8-bit devices. This facility allows the LR33020 to fetch instructions directly from 8-bit PROMs.

Bus Arbiter – The BIU contains a bus arbiter that manages bus request and grant functions for the internal and external DMA channels. The LR33020 includes support for four internal DMA channels and one external DMA channel. The four internal DMA channels are:

- Two graphics DMA channels, one of which provides source data for the Bit-Blt Processor and the other of which stores the processed pixels at the programmed destination
- Two video DMA channels, one of which provides a hardware cursor when used with a VRAM-based system or a Video FIFO for DRAM-based frame buffers and the other of which provides VRAM refresh operations. VRAM split-read transfers are supported.

The external DMA channel is requested via the $\overline{BREQ}$ line. When the system bus is free, the LR33020 grants the bus to the external device by asserting BGNT. The external device must request bus masterhood again for each bus transaction it wishes to make.

During bus arbitration, DRAM/VRAM refresh cycles always take priority, as any other action might violate data integrity. The Video Controller takes second priority, as VRAM read transfers must take place during the screen flyback period. The external DMA channel takes third priority. The Graphics Coprocessor has fourth priority. The MIPS CPU has the lowest priority.

The LR33020 uses hierarchical prioritization for simultaneous bus requests. The device owning the bus relinquishes the bus to higher-priority devices at the end of each bus cycle.

Video Controller
The Video Controller contains three main elements: the video timing logic, the video FIFO and serializer and the Video DMA Controller. A summary of these three functions follows.

Video Timing Logic – The video timing logic provides the video synchronization and blanking circuitry for rasterized data. Up to three synchronization signals can be provided at any one time, with one signal functioning either as a horizontal synchronization ($\overline{HSYNC}$) or composite synchronization ($\overline{CSYNC}$) signal. The LR33020 supports three types of non-interlaced video timing signals:

- $\overline{VSYNC}$, $\overline{HSYNC}$ and composite $\overline{BLANK}$ outputs from a single VCLK input
- $\overline{HSYNC}$ and composite $\overline{BLANK}$ outputs from a VCLK and a $\overline{VSYNC}$ input
- $\overline{CSYNC}$ and composite $\overline{BLANK}$ outputs from a single VCLK input

The placement of the respective timing edges is fully programmable, with 16-bit registers controlling the relative edges of horizontal and vertical front and back porches, synchronization shelves and active display area.

Video FIFO and Serializer– The LR33020's 32-word video FIFO can serve different purposes depending on whether the system uses DRAM or VRAM for video information:

- In DRAM-based systems, the video FIFO can feed data from the DRAM to the monitor or RAMDAC, thus providing a channel for pixel data that does not consume CPU resources
- In VRAM-based systems, the video FIFO can supply a hardware cursor

The video serializer can serialize 1, 2, 4 or 8 bits of video data.

The video serializer includes bit-swapping logic to assist with big/little endian conversion.

Video DMA Controller – The Video DMA Controller can fetch frame-buffer data from a DRAM-based frame buffer and pipe the video data into the on-chip 32-word video FIFO independently of CPU control.

The Video DMA Controller can also coordinate access to VRAM by the RAMDAC and superimpose a 64x64x2 hardware cursor on the VRAM from DRAM.

The Video DMA Controller supports split read transfers from VRAM and supplies VRAM refresh if a VRAM frame buffer is being used.

LR33020 Signals

The LR33020 interface signals are described in the following four categories:

- BIU Signals
- Video Signals
- SRAM and I/O Controller Signals
- Miscellaneous Signals

BIU Signals

The BIU signals described in this section comprise the LR33020's interface to I/O devices and memory subsystems.

A[28:0]
Address Bus (Bidirectional) – This 29-bit bus provides instruction and data addresses to the memory systems and peripherals. The processor itself uses a 32-bit address, but the upper 3 bits of an address are not brought off chip. This bus is an output only when the LR33020 is the bus master. This bus is an input when the bus is owned by another bus master; the LR33020 can then generate the appropriate DRAM signals when another bus master requires the DRAM controller circuitry of the LR33020.

$\overline{AS}$
Address Strobe (Bidirectional) – This signal indicates the start and end of a bus cycle. The LR33020 asserts $\overline{AS}$ (LOW) when initiating a bus cycle that is not decoded by the LR33020's internal address decoder. $\overline{AS}$ is deasserted (HIGH) when the current transaction is completed. When the bus is granted to an external bus master, $\overline{AS}$ is sampled as an input to initiate a DRAM controller operation and/or snooping.

$\overline{BERR}$
Bus Error (Input) – This signal indicates that an exception condition has occurred. An external device should assert $\overline{BERR}$ (LOW) during a memory transaction. If the LR33020 is bus master at the time, the LR33020 then terminates the on-going memory transaction and takes an exception.

BFREQ
Block Fetch Request (Bidirectional) – This signal indicates that a burst access to memory is being requested. When an external bus master is requesting use of the DRAM/VRAM controller, this signal is used as an input to indicate whether a block fetch is to be executed. If the memory system being addressed does not support block fetch, then $\overline{BFTCH}$ should not be asserted. If the on-chip DRAM/VRAM controller is enabled, this signal can be ignored for DRAM cycles.

$\overline{BFTCH}$
Block Fetch Acknowledge (Input) – In externally controlled memory systems, this signal indicates that the memory system is ready for a block fetch. This signal should be asserted in response to BFREQ for externally controlled memory systems. Assert $\overline{BFTCH}$ (LOW) only if the addressed device supports block fetch. Leaving the signal deasserted (HIGH) forces the LR33020 to accept a single-word read operation. $\overline{BFTCH}$ should not be asserted (LOW) for DRAM cycles.

BGNT
Bus Grant (Output) – This signal informs the bus master requesting the bus that the LR33020 has relinquished the bus and has 3-stated A[28:0], D[31:0], D[63:32], $\overline{DWE}$[3:0], $\overline{RT}$ and $\overline{AS}$. On the activation of $\overline{AS}$, the DRAM controller examines the address bus and $\overline{RT}$ signal. If appropriate, the DRAM controller initiates a DRAM cycle. The address bus and the $\overline{RT}$ signal must be stable for proper operation. The external device is also responsible for the proper generation of the $\overline{DWE}$[3:0] signals. When $\overline{INTRLV}$ is asserted, the LR33020's DRAM controller gates odd-word write accesses to D[63:32] and odd-word read accesses to D[31:0], because all external masters must reside on D[31:0] (and not D[63:32]). External masters using the LR33020's DRAM controller can only perform single-cycle read and write accesses.

$\overline{BREQ}$
Bus Request (Input) – External bus masters can use this signal (active low) to request the memory bus from the LR33020. Bus masters requesting the bus may or may not use the LR33020 DRAM Controller.

BWIDE
Byte-Wide ROM Space (Input) – This signal allows execution from an 8-bit-wide memory system. This signal informs the LR33020 that the address space being accessed is 1 byte wide. When the LR33020 detects this signal, it fetches four bytes from memory beginning at the first generated word-aligned byte address until the complete addressed word has been collected.

LR33020 Signals
(Continued)

$\overline{\text{CACHD}}$
Cacheable Datum (Input) – This signal informs the LR33020 whether the current data is cacheable. Assert $\overline{\text{CACHD}}$ (LOW) to indicate that the data being fetched is cacheable. For external DMA bus writes, assert $\overline{\text{CACHD}}$ (LOW) to indicate that snooping is desired. If the cache is to operate at all times, tie $\overline{\text{CACHD}}$ low.

D[31:0]
Data Bus (Bidirectional) – This 32-bit bus provides data access to the memory and peripherals in the system. The LR33020 drives the bus on memory write cycles and reads the bus on memory read cycles. When the bus is owned by another master using the LR33020 DRAM controller, a read cycle to the odd bank in an interleaved system memory causes the data on DATA[63:32] to be driven onto this bus.

D[63:32]
Data Bus (Bidirectional) – This 32-bit bus provides the data to and from the DRAM memory on two-way interleaved systems only. Together with the LR33020 DRAM/VRAM controller circuitry, D[63:32] allows for the direct connection of 64-bit wide interleaved memory to the LR33020. 64-bit wide interleaved memory can support a zero-wait-state DRAM memory system, allowing single-cycle cache refill and video data refreshing out of DRAM. D[63:32] provides the odd-word data path to the LR33020. When the bus is owned by another master in an interleaved memory system, a write cycle to the odd bank causes the data on DATA[31:0] to be driven onto this bus.

$\overline{\text{DCAS0}}$[3:0] and $\overline{\text{DCAS1}}$[3:0]
DRAM Even and Odd Column Address Strobes (Output) – These signals are used to strobe the column address into DRAM. During single cycle or burst (fast-page-mode) accesses in interleaved systems, the LR33020 asserts the Column Address Strobe signals (LOW) when a valid column address is being presented to the DRAM bank. To support interleaved systems, $\overline{\text{DCAS0}}$[3:0] are activated during accesses to even addresses, and $\overline{\text{DCAS1}}$[3:0] are activated during accesses to odd addresses.

All four signals in the bus are activated during read operations. For write operations, each signal selects the appropriate byte write action. The LR33020 asserts these signals two clocks after the $\overline{\text{DRAS}}$ signal (one clock after $\overline{\text{DMXS}}$ is asserted). Between words in a burst-read operation, the LR33020 holds this signal active for one or two clocks and then inactive for one clock. The timing presented by the Column Address Strobe signals allow the direct interface to a DRAM memory system that supports both single-cycle and fast-page-mode accesses.

$\overline{\text{DMXS}}$
DRAM Address Multiplexer Select (Output) – This signal should drive an external multiplexer to select either the row or the column address for the DRAM array. The timing presented by this signal conforms with the timing of the $\overline{\text{DRAS}}$ and $\overline{\text{DCAS}}$ signals generated by the LR33020. This signal is deasserted when the DRAM sequence ends.

$\overline{\text{DOE}}$
DRAM Output Enable (Output) – This signal is asserted (LOW) to indicate a DRAM/VRAM read operation or a VRAM Serial Access Mode (SAM) transfer. $\overline{\text{DOE}}$ should be tied to the VRAM $\overline{\text{OE}}$ pins.

$\overline{\text{DRAMSEL}}$
DRAM Select (Bidirectional) – When asserted (LOW), this signal indicates the start of a DRAM cycle that has been initiated by the LR33020 DRAM Controller. When another master owns the bus, this signal is an input which causes the LR33020 DRAM Controller to initiate a DRAM cycle.

$\overline{\text{DRAS0}}$
$\overline{\text{DRAS1}}$
Low and High DRAM Row Address Strobes (Output) – These signals are used to strobe the row address into DRAM. The LR33020 asserts $\overline{\text{DRAS0}}$ or $\overline{\text{DRAS1}}$ LOW one clock after the $\overline{\text{DRAMSEL}}$ and data bus (D[31:0] or D[63:32]) are valid. The timing presented by these signals allow direct interface to a DRAM memory system.

$\overline{\text{DRDY}}$
Data Ready (Input) – External logic asserts this signal to indicate readiness for the current bus transaction data. When DRAM, ROM, or I/O space is selected, the on-chip memory controller may assert an internal version of this signal to complete a transaction. When any other address space is selected, the external device must assert $\overline{\text{DRDY}}$ at the appropriate time.

DSF
DRAM Special Function Select (Output) – This signal selects DRAM and VRAM special functions, such as serial register loads. This signal should be tied to the DRAM/VRAM DSF input

LR33020 Signals (Continued)

pins. It is driven appropriately to select different operations in the DRAM/VRAM, such as serial register loads. DSF may be left unconnected in DRAM-only systems.

DWE [3:0]
Byte Write Enable Strobes (Output) – These signals indicate valid data on their respective byte of data on D[31:0] or D[63:32]. The LR33020 DRAM controller drives DWE[3:0] when it has ownership of the bus. When another bus master owns the bus, the LR33020 3-states these signals; the external bus master must then drive DWE[3:0] to generate DRAM write cycles. As the on-chip DMA Controller only performs full-word writes, the DMA Controller does not use DWE[3:0]. When DRAM/VRAM arrays are controlled by the LR33020's on-chip DRAM/VRAM controller, the arrays use DCAS-controlled writes instead of DWE[3:0].

IDLENI
Input Data Bus Latch Enable In (Input) – This signal latches the data coming on chip into the data buffer latches. The LR33020 IDLENI input should be driven by the LR33020 IDLENO output. If timing in the DRAM system is tight, some small delay may need to be added to the signal.

IDLENO
Input Data Bus Latch Enable Out (Output) – This signal comes off chip from the DRAM controller to control the timing for the input data buffer latch enables on chip. IDLENO should drive the IDLENI input pin. This provides appropriate delay for latching DRAM read data.

INTRLV
Interleaved Memory Bank Selection (Input) – This signal indicates an interleaved memory access to the LR33020 DRAM/VRAM controller. The interleaved memory may be DRAM, VRAM, SRAM or other similar devices. For each memory reference, the signal must be asserted (LOW) for interleaved memory banks or deasserted (HIGH) for non-interleaved memory banks. For instance, if the system is configured with interleaved DRAM only, the output signal DRAMSEL could be tied directly to this input.

ME_WE
Mask Enable/Write Enable Select (Output) – This signal selects DRAM write operations and VRAM special functions. ME_WE should be tied to the ME/WE input of the DRAMs/VRAMs. ME_WE is also driven HIGH during DRAM Refresh.

MXSTART
Memory Transaction Start (Output) – This signal indicates that the CPU is initiating a memory cycle. The LR33020 asserts MXSTART (LOW) for one cycle at the beginning of each memory cycle initiated by the CPU. MXSTART is not driven LOW during DMA bus cycles.

RFRSH
Refresh Cycle (Output) – When the LR33020 DRAM controller is enabled, this signal indicates a DRAM refresh cycle. When the LR33020 is disabled, RFRSH provides a general-purpose timer strobe. This signal should be used by dual-bank memory systems to enable DCAS and DRAS simultaneously to both banks. When the LR33020 DRAM controller is disabled, this signal becomes a general-purpose timer output indicating that the timer has counted down from its preset value to zero.

RD
Read Strobe (Output) – The bus master asserts RD (LOW) to indicate that the current transaction is a read transaction and the memory may drive data onto the data bus. When deasserted (HIGH), only the bus master may drive the data bus.

RT
Read Transaction (Bidirectional) – This signal indicates whether the current memory transaction is a read or a write operation. This signal should be driven valid when an external bus master owns the bus and is performing a read or write operation.

Video Signals
The video signals described in this section can interface directly to an external RAMDAC.

BLANK
Video Composite Blank Signal (Output) – This signal (active LOW) is asserted by the LR33020 when the current video dot position is in the front porch or back porch blanking or sync period. The timing and duration of this signal is user programmable.

C/HSYNC
Video Composite or Horizontal Synchronization (Output) – When using separate horizontal and vertical video timing signals, this signal (active LOW) is asserted by the LR33020 when horizontal synchronization is taking place. When using composite video timing, this signal provides the composite synchronization signal. For both

LR33020 Signals
(Continued)

modes, the timing and duration of this signal is user programmable via the Coprocessor 2 register space. For composite synchronization, serration pulse timing is also user programmable. In addition, an overscan mode is provided.

HWCRSR
Hardware Cursor Active (Output) – When hardware cursor mode is enabled, HWCRSR indicates that the hardware cursor data is being shifted out on the VDATA port.

QSF
VRAM Shift Indicator (Input) – When using VRAM-based frame buffers that support split serial access mode (split SAM), this LR33020 input connects directly to the VRAMs' QSF signal output. The LR33020 uses QSF to determine when the next split SAM transfer should occur.

VCLK
Video Clock (Input) – This signal is the pixel scanning dot clock. This signal can also divide the pixel clock by 2, 4 or 8. All the video timings generated by the LR33020 are based on counters and comparators clocked by this signal. When the LR33020 is serializing video data, the divide rate must be consistent with the number of bits shifted per VCLK cycle. This clock can have a maximum frequency of 80 MHz.

VDATA[7:0]
Video Data Bus (Output) – When enabled, this bus serializes the data from the 32 x 32 Video FIFO in the LR33020's video controller. VDATA[7:0] shifts data out on both edges of the video clock (VCLK). The serializer bus can be programmed to shift data by 1, 2, 4 or 8 bits per clock edge. The video blanking ($\overline{\text{BLANK}}$) suspends the data output.

$\overline{\text{VSYNC}}$
Video Vertical Synchronization (Bidirectional) – When asserted (LOW), this signal indicates that vertical synchronization is to take place. The timing and duration of this signal is programmable and is based on the VCLK input. When programmed as an input, this signal allows an external device to reset the vertical counter. In printer applications, for example, the VCLK input can indicate the start of a new page.

SRAM and I/O Controller Signals
This signal group includes the timer, serial port and chip select signals. The Serial Port and Timer 2 can both communicate with external devices via the signals described below.

ACLK/SI0
PS/2 Port Clock/Serial Port 0 Input (Input) – When the Serial Port Logic is configured as a PS/2 Compatible port, this signal is the LR33020 clock input for the PS/2 Serial Port. When receiving a serial stream on the PS/2 Serial Port, ACLK shifts incoming data into the LR33020. When transmitting a serial stream on the PS/2 Serial Port, ACLK strobes data out of the LR33020. When the Serial Port Logic is instead configured as separate standard serial ports, this signal is the input port for Serial Port 0.

ACLKEN/SO0
PS/2 Port Clock Enable/Serial Port Output (Output) – When the Serial Port Logic is configured as a PS/2 Compatible port, this signal controls the PS/2 Serial Port clock signal. When ACLKEN is asserted (HIGH), ACLK can be asserted (HIGH) by an auxiliary device. When ACLKEN is deasserted (LOW), external hardware should deassert ACLK (LOW). When the Serial Port Logic is instead configured as separate standard serial ports, this signal is the output port for Serial Port 0.

ADIN/SI1
PS/2 Port 0 Data In/Serial Port 1 Input (Input) – When the Serial Port Logic is configured as a PS/2 Compatible port, this signal is the Serial Port 0 data input. If receive is enabled for this port, data is shifted in this pin on the falling edge of the ACLK input. When the Serial Port Logic is instead configured as separate standard serial ports, this signal is the input port for Serial Port 1.

ADOUT/SO1
PS/2 Port 0 Data Out/Serial Port 1 Output (Output) – When the Serial Port Logic is configured as a PS/2 Compatible port, this signal is the Serial Port 0 data output. When transmitting data, the LR33020 transmits data out this pin on the falling edge of ACLK. This signal is asserted (HIGH) when the LR33020 Serial Port 0 is not transmitting data. When the Serial Port Logic is instead configured as separate standard serial ports, this signal is the output port for Serial Port 1.

$\overline{\text{IOCS1}}$, $\overline{\text{IOCS2}}$ and $\overline{\text{IOCS3}}$
I/O Space 1, 2, and 3 Selects (Output) – The LR33020 asserts one of these signals (LOW) when the address bus accesses the corresponding space. These signals can be used as chip-select signals and/or output-enable signals for 3-stated buffers. The address bus and the $\overline{\text{RT}}$ signals are stable when these signals are

LR33020 Signals
(Continued)

generated. The length of an $\overline{\text{IOCS}}$ signal is a programmable number of SYSCLK's, although the signal length can be shortened by an external device asserting $\overline{\text{DRDY}}$. The number of SYSCLK's for $\overline{\text{IOCS1}}$ is set in the LR33020 Configuration register; the $\overline{\text{IOCS2}}$ and $\overline{\text{IOCS3}}$ wait states are programmed in the Coprocessor 2 register space. The I/O address space always uses the D[31:0] data path.

$\overline{\text{PROMCS}}$
ROM Memory Space Select (Output) – This signal indicates that the LR33020 is accessing the ROM address space. $\overline{\text{PROMCS}}$ can be used as a chip-select signal and/or an output-enable signal for 3-state buffers. The address bus and the $\overline{\text{RT}}$ signals are stable when this signal is generated. The length of this signal is a programmable number of SYSCLK's, although the signal length can be also shortened by an external device asserting $\overline{\text{DRDY}}$. Wait-state duration is programmed in the LR33020 Configuration Register. The ROM address space always uses the D[31:0] data path.

T2TMO
Timer 2 Timeout (Output) – The LR33020 toggles this signal when the LR33020 Timer 2 Counter reaches zero.

Miscellaneous Signals
Configuration, interrupt, status, test and general signals are described in this section.

BRTKEN
Branch Taken (Output) – When asserted (active HIGH), this signal indicates that the CPU is taking a branch in the instruction execution. This signal is useful for tracing instruction flow.

$\overline{\text{FRCMISS}}$
Force Instruction Cache Miss (Input) – This signal is used for chip testing. Emulation hardware can also force the next instruction to be a cache miss via this signal. In most situations, this signal should be tied high.

$\overline{\text{INT}}$[5:3]
Interrupts (Input) – The LR33020 provides three of the six hardware interrupts specified by the MIPS architecture specification. The other three interrupts are used internally by the LR33020. Note that the polarity of the interrupt signals has reversed from the interrupt implementation in the LR33000.

LEND
Endian-ness Configuration (Input) – When asserted (LOW), this signal indicates that the processor is using little-endian byte ordering. When deasserted (HIGH), this signal indicates that the processor is using big-endian byte ordering.

$\overline{\text{MID}}$
Memory Transaction Type (Output) – When deasserted (HIGH), this signal indicates that the bus transaction is an instruction fetch. When asserted (LOW), this signal indicates that the bus transaction is a data read or write transaction.

$\overline{\text{RESET}}$
Reset Schmitt Trigger (Input) – When asserted (LOW), this signal resets the processor state. $\overline{\text{RESET}}$ is a Schmitt trigger input with a pull-up resister; a reset circuit can thus be implemented with minimal external logic. When this signal is asserted (LOW), all chip output and bidirectional pins are 3-stated.

STALL
Stall (Output) – When asserted (HIGH), this signal indicates that the processor is in a stall state.

SYSCLK
System Clock (Input) – This is the primary clock input to the LR33020. This clock is the CPU cycle clock, the DRAM controller synchronous clock and the general purpose timer clock. All memory bus signals are synchronous to this clock.

TDONE
Cache Test Done (Output) – This signal is used for test-mode access to the caches.

THIT
Cache Test Hit (Output) – This signal is used for test-mode access to the caches.

Specifications

This section provides the electrical specifications for the LR33020. Table 3 lists the absolute maximum rating for the LR33020. Operation beyond the limits set forth in this table may impair the useful life of the device.

Table 3. Absolute Maximum Ratings

Parameter	Symbol	Min	Max	Units
Supply Voltage	VDD	-0.3	+7.0	V
Input Voltage	VIN	-0.3[1]	+7.3	V
Storage Temperature	TST	-40	+125	°C

Note:
1. VIN Min. = -3.0 V for pulse width less than 15 ns.

Table 4 lists the LR33020's recommended operating conditions, Table 5 lists the capacitance of its inputs and outputs, and Table 6 lists the DC electrical specifications for the LR33020. Table 7 lists the AC electrical specifications for the LR33020, and Figures 9 through 11 are timing waveforms which illustrate the AC timing values. In the AC specifications, all timing is referenced to 1.5 V, and all output timing assumes 25 pF of capacitive load.

Table 4. Recommended Operating Conditions

Symbol	Parameter	Limits	Unit
VDD	DC Supply	+4.75 to +5.25	V
TA	Ambient Temperature	-0 to +70	°C

Note: Not more than one output should be shorted at a time; duration of the short should not exceed 30 seconds.

Table 5. Capacitance

Symbol	Parameter	Condition	Min	Typ	Max	Units
CIN	Input Capacitance	VIN = 5.0 V, TA = 25° C, f = 1 MHz	–	–	5	pF
COUT	Output Capacitance	VIN = 5.0 V, TA = 25° C, f = 1 MHz	–	–	10	pF
CIO	I/O Bus Capacitance	VIN = 5.0 V, TA = 25° C, f = 1 MHz	–	–	15	pF

Table 6. DC Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Units
VIL	Voltage Input LOW		–	–	0.8	V
VIH	Voltage Input HIGH		2.0	–	–	V
VOH	Voltage Output HIGH	IOH = -4.0 mA	2.4	4.5	–	V
		IOH = -8.0 mA	2.4	4.5	–	V
VOL	Voltage Output LOW	IOL = 4.0 mA	–	0.2	0.4	V
		IOL = 8.0 mA	–	0.2	0.4	V
IIH	Current Input HIGH	VIN = VDD or VSS	–	–	10	µA
IIL	Current Input LOW[2]	VIN = VDD or VSS	–	–	-10	µA
IOZ	Current 3-State Output Leakage[2]	VOH = 2.4 V, VOL = 0.4 V	-10	±1	10	µA
IDD	Quiescent Supply Current	VIN = VDD or VSS	–	–	4	mA
ICC	Supply Current	VIN = Max, f = 25 MHz	–	–	400	W
		VIN = Max, f = 33 MHz	–	–	500	W
		VIN = Max, f = 40 MHz	–	–	600	

Notes:
1. Specified at VDD equals 5 V ± 5% at ambient temperature over the specified range.
2. For inputs and outputs with pull-up resistors, low voltage level current is 500 µA max.

Specifications
(Continued)

Table 7. AC Timing Values

Parameter		Description	40		33		25		Units
			Min	Max	Min	Max	Min	Max	
1.	SHAV	SYSCLK+ to Address Valid	–	17	–	20	–	26	ns
2.	SHBRV	SYSCLK+ to BFREQ Valid	–	15	–	16	–	20	ns
3.	SHBRI	SYSCLK+ to BFREQ Invalid	–	15	–	16	–	20	ns
4.	SHML	SYSCLK+ to $\overline{\text{MXSTART}}$ Low	–	15	–	16	–	20	ns
5.	SHRTL	SYSCLK+ to $\overline{\text{RT}}$ Low	–	15	–	16	–	20	ns
6.	SHRTH	SYSCLK+ to $\overline{\text{RT}}$ High	–	14	–	15	–	18	ns
7.	SHDTV	SYSCLK+ to $\overline{\text{MID}}$ Valid	–	15	–	16	–	20	ns
8.	SHDTI	SYSCLK+ to $\overline{\text{MID}}$ Invalid	–	15	–	16	–	20	ns
9.	SLAL	SYSCLK- to $\overline{\text{AS}}$, $\overline{\text{RD}}$ Low	–	17	–	20	–	23	ns
10.	SLSL	SYSCLK- to $\overline{\text{PROMCS}}$ or $\overline{\text{IOCS}}$[3:1] Low	–	14	–	15	–	18	ns
11.	SLSH	SYSCLK- to $\overline{\text{PROMCS}}$ or $\overline{\text{IOCS}}$[3:1] High	–	14	–	15	–	18	ns
12.	AVASL	Address Valid to $\overline{\text{AS}}$, $\overline{\text{RD}}$, $\overline{\text{PROMCS}}$, and $\overline{\text{IOCS}}$[3:1] Low	5	–	5	–	5	–	ns
13.	CSSL	Control Setup to SYSCLK-	4	–	5	–	6	–	ns
14.	CHSL	Control Hold from SYSCLK-	3	–	4	–	5	–	ns
15.	BEHSL	$\overline{\text{BERR}}$ Hold from SYSCLK-	5	–	6	–	7	–	ns
16.	DBSU	Data Setup before SYSCLK+	2	–	2	–	3	–	ns1
17.	DBH	Data Hold after SYSCLK+	5	–	5	–	6	–	ns
18.	SHASH	SYSCLK+ to $\overline{\text{AS}}$, $\overline{\text{RD}}$ High	–	12	–	13	–	15	ns
19.	AHSH	Address Hold from $\overline{\text{AS}}$, $\overline{\text{RD}}$, $\overline{\text{PROMCS}}$, and $\overline{\text{IOCS}}$[3:1] High	0	–	0	–	0	–	ns
20.	SHWRV	SYSCLK+ to $\overline{\text{DWE}}$[3:0] Valid	–	15	–	16	–	20	ns
21.	SHDV	SYSCLK+ to Data Valid	–	19	–	21	–	26	ns
22.	SHWRH	SYSCLK+ to $\overline{\text{DWE}}$[3:0] High	–	13	–	14	–	16	ns
23.	WRHDI	$\overline{\text{DWE}}$[3:0] High to Data Invalid	10	–	10	–	10	–	ns
24.	SLRL	SYSCLK- to $\overline{\text{DRAS}}$[1:0] Low	–	13	–	14	–	17	ns
25.	SLRH	SYSCLK- to $\overline{\text{DRAS}}$[1:0] High	–	13	–	14	–	17	ns
26.	ASRL	Address Setup before $\overline{\text{DRAS}}$[1:0] Low	10	–	10	–	12	–	ns
27.	SLMSL	SYSCLK- to $\overline{\text{DMXS}}$ Low	–	13	–	14	–	16	ns
28.	SLMSH	SYSCLK- to $\overline{\text{DMXS}}$ High	–	13	–	14	–	16	ns
29.	SLCAL	SYSCLK- to $\overline{\text{DCAS0}}$[3:0] and $\overline{\text{DCAS1}}$[3:0] Low	–	14	–	15	–	18	ns
30.	SLOEL	SYSCLK- to $\overline{\text{DOE}}$ Low	–	13	–	14	–	17	ns
31.	SHCAH	SYSCLK+ to $\overline{\text{DCAS0}}$[3:0] and $\overline{\text{DCAS1}}$[3:0] High	–	14	–	15	–	18	ns
32.	SHOEH	SYSCLK+ to $\overline{\text{DOE}}$ High	–	13	–	14	–	17	ns
T1.	SHAX	SYSCLK+ to ACLKEN and ADOUT Valid		14		15		18	ns
V1.	VHSV	VCLK+ to Video Signal Valid	–	11	–	12	–	15	ns
V2.	VPWH	VCLK Pulse Width High	–	6	–	6	–	6	ns
V3.	VPWL	VCLK Pulse Width Low	–	6	–	6	–	6	ns
V4.	VP	VCLK Period	–	12.5	–	12.5	–	12.5	ns

Specifications
(Continued)

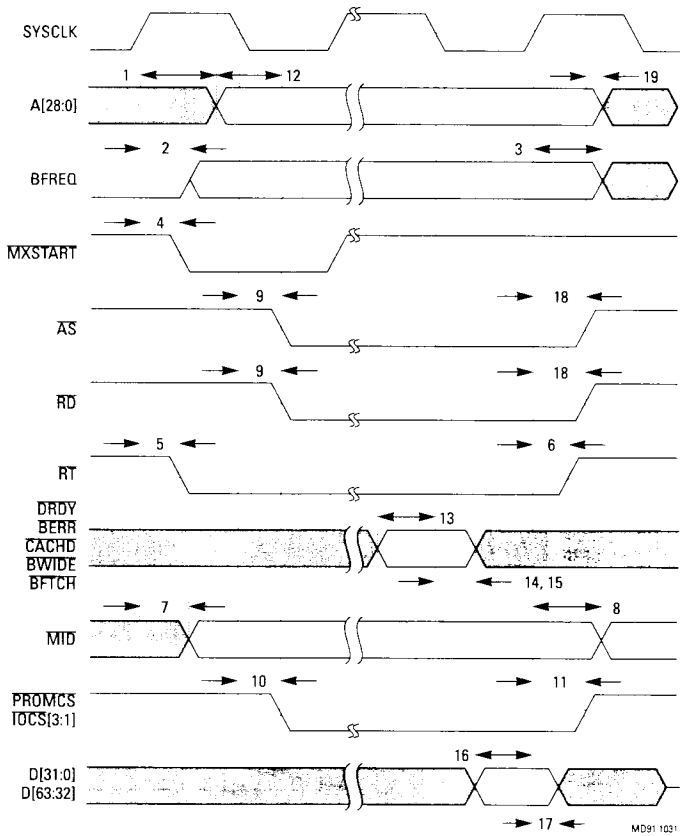


Figure 12. Read Transaction Timing

Specifications
(Continued)

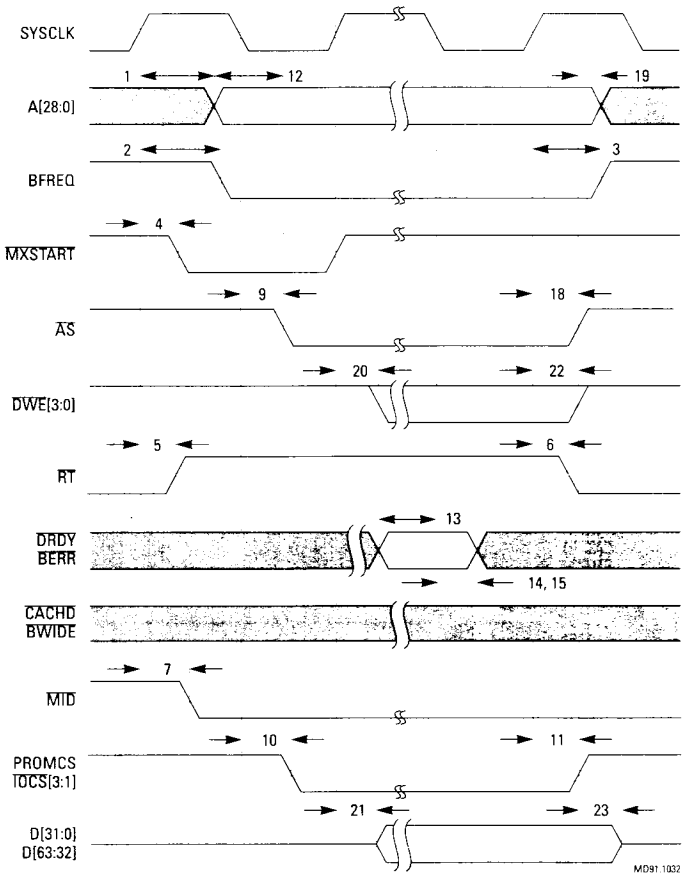
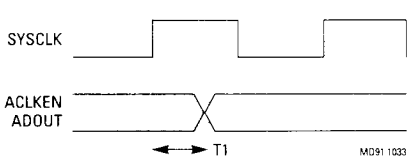


Figure 13. Write Transaction Timing



Note:
1. Input pins ACLK and ADIN are asynchronous inputs that are internally synchronized by SYSCLK.

Figure 14. Serial Port Timing

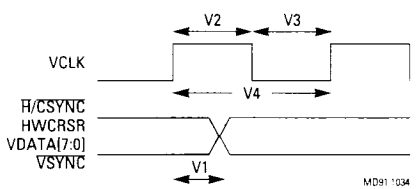
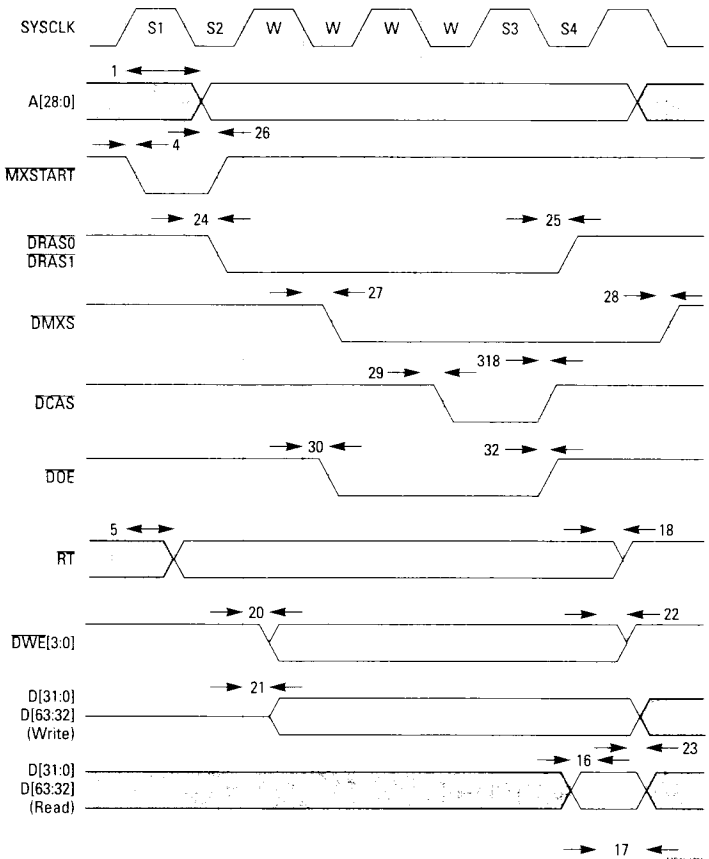


Figure 15. Video Timing

Specifications
(Continued)



Notes:
1. $\overline{DOE}$ is not asserted during write transactions.
2. 27 and 28 apply when $\overline{IDLENT}$ is slow with respect to SYSCLK.

Figure 16. DRAM Access Timing

Pinout, Package and Ordering Information

The LR33020 is available in two different packages: a 224-pin Ceramic Pin Grid Array (CPGA) and a 208-pin Metal Quad Flat Package (MQUAD). Designers will choose the package that meets the cost and performance requirements of their application.

Table 8 lists and describes the two packages by order number.

This section contains two types of information for each package type: a pinout and a mechanical drawing. For the 224-pin CPGA, Figure 17 and Figure 18 contain the two types of information. For the 208-pin MQUAD, Figure 19 and Figure 20 contain the two types of information.

Table 8. LR33000 Ordering Information

Order Number	Clock Frequency	Package Type	Operating Range
LR33020MC-25	25	208 MQUAD	Commercial
LR33020MC-33	33	208 MQUAD	Commercial
LR33020MC-40	40	208 MQUAD	Commercial
LR33020GC-25	25	224 CPGA	Commercial
LR33020GC-33	33	224 CPGA	Commercial
LR33020GC-40	40	224 CPGA	Commercial

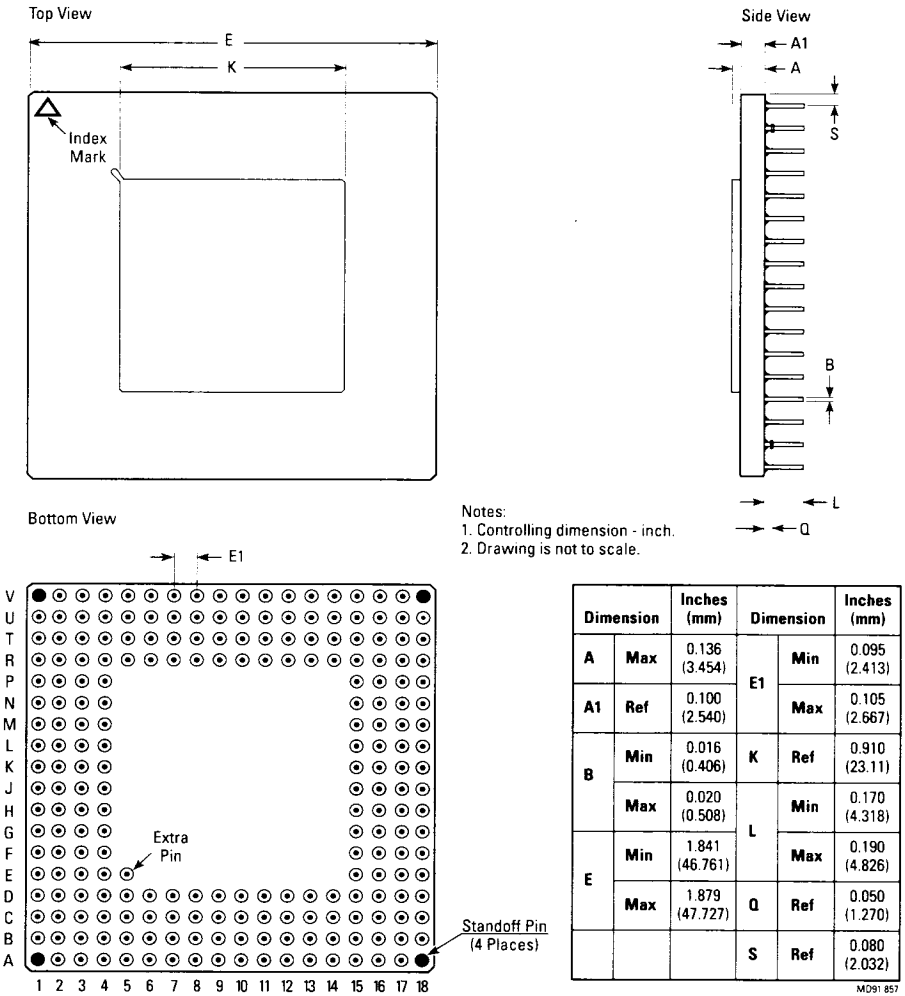
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	ACLKEN	D62	VSS	D28	D27	D25	D56	D23	D21	VDD	VSS	D50	D48	D47	VSS	D13	D44	D42
B	VDD	*NC*	D30	D61	D59	D57	VSS	D55	D53	D52	D51	D49	D16	VSS	D46	D12	VDD	D41
C	VSS	ADIN	D31	VDD	D60	D26	VDD	D24	D22	D20	D19	D17	VDD	D14	D11	D10	D9	D39
D	THIT	ISTST	ACLK	D63	VDD	D29	D58	VSS	D54	VDD	D18	D15	D45	D43	VSS	D8	D7	D37
E	BRTKEN	RESET	DSTST	ADOUT											D40	D6	D5	VDD
F	INT3	VDD	BERR	VSS											D38	VSS	D36	D4
G	TDONE	INT5	INT4	STALL											VSS	D35	D34	D2
H	VSS	VDD	VDD	RFRSH											D3	D33	D1	D32
J	BLANK	HWCRSR	CRSYNC	VSYNC											D0	VSS	VDD	VDD
K	DSF	*NC*	*NC*	VDD						Top View					BREQ	VCLK	SYSCLK	BGNT
L	VSS	VDATA0	VDATA1	VDD											VDD	RD	ME_WE	RT
M	VDATA2	VDATA3	VSS	VDATA7											AS	VSS	VDD	MID
N	VDATA4	VDATA5	VDATA6	A1											DRDY	MXSTART	T2TMO	VSS
P	VDD	VSS	A0	A5	*NC*										INTRLV	CACHD	BFTCH	DSF
R	VSS	A2	A4	A7	VSS	A13	A17	A22	VSS	VDD	DCAS13	DCAS03	DWE0	PROMCS	VDD	LEND	PROMISS	BFREQ
T	A3	A6	VDD	VDD	A14	A16	A20	VDD	A26	DRAMSEL	DCAS10	DRAS0	DCAS02	VDD	DWE3	IOCS3	BWIDE	IDLENT
U	VDD	A8	A10	A15	VDD	A18	A21	A24	A27	DOE	DRAS1	DCAS12	DCAS01	VSS	DWE1	IOCS1	VDD	IDLENO
V	A9	A11	A12	VDD	VSS	A19	A23	A25	VSS	A28	VSS	DCAS11	DCAS00	DMXS	VSS	DWE2	IOCS2	VSS

MD91 1036

Note:
1. *NC* pins are not connected.

Figure 17. 224-Pin CPGA Pinout

Pinout, Package and Ordering Information
(Continued)



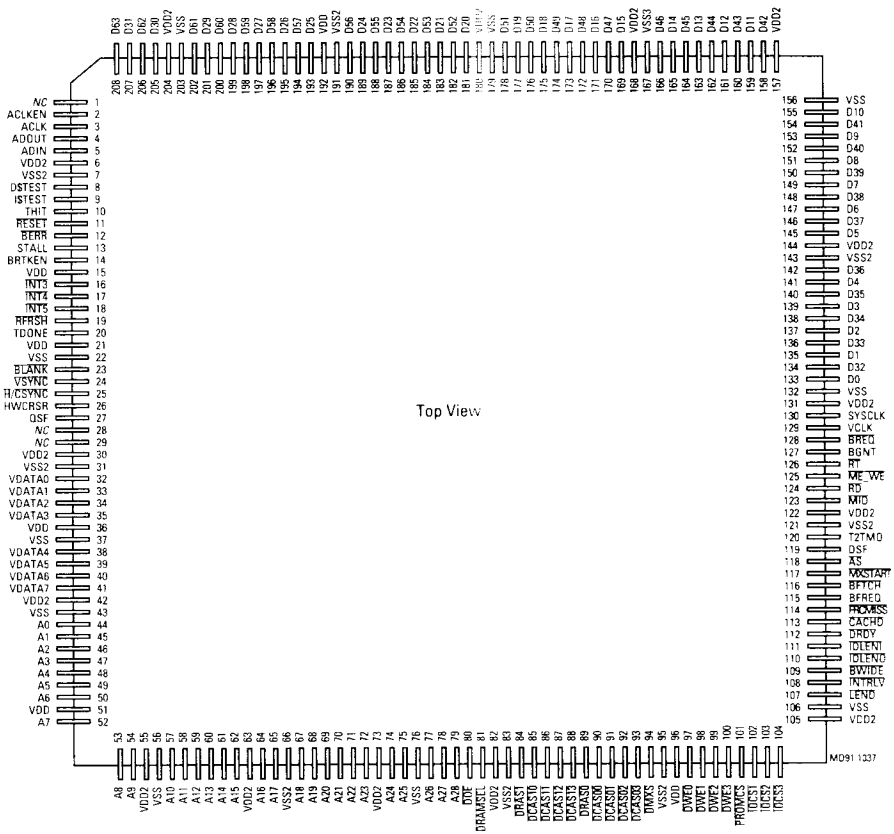
Notes:
1. Controlling dimension - inch.
2. Drawing is not to scale.

Dimension		Inches (mm)	Dimension		Inches (mm)
A	Max	0.136 (3.454)	E1	Min	0.095 (2.413)
A1	Ref	0.100 (2.540)		Max	0.105 (2.667)
B	Min	0.016 (0.406)	K	Ref	0.910 (23.11)
	Max	0.020 (0.508)	L	Min	0.170 (4.318)
E	Min	1.841 (46.761)		Max	0.190 (4.826)
	Max	1.879 (47.727)	Q	Ref	0.050 (1.270)
			S	Ref	0.080 (2.032)

MD91 857

Figure 18. 224-Pin CPGA Mechanical Drawing

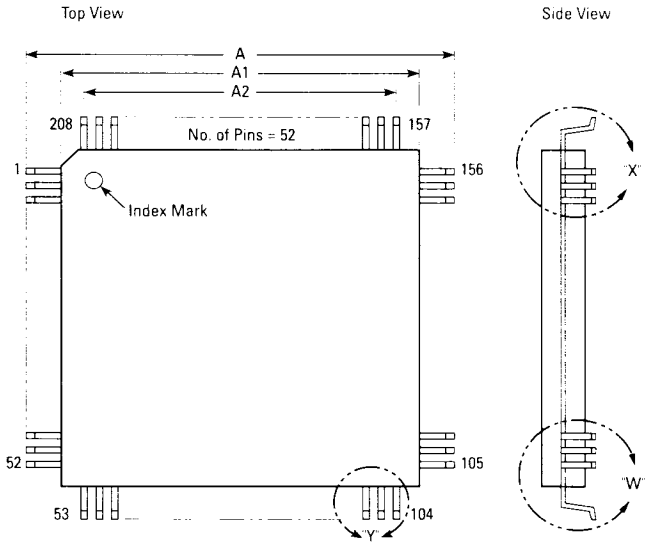
Pinout, Package and Ordering Information (Continued)

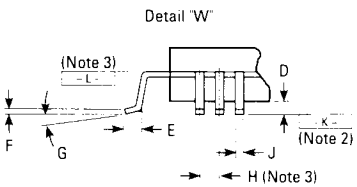


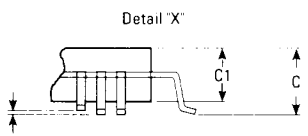
Note:
1. *NC* pins are not connected.

Figure 19. 208-Pin MQUAD Pinout

Pinout, Package and Ordering Information
(Continued)







Dimension		Millimeters (Inches)
A	Min	30.40 Sq (1.197)
	Max	30.80 Sq (1.213)
A1	Min	27.56 Sq (1.085)
	Max	27.72 Sq (1.091)
A2	Ref	25.50 Sq (1.004)
C	Max	3.86 (0.152)
C1	Max	3.43 (0.135)
D	Max	0.41 (0.020)
E	Min	0.40 (0.016)
	Max	0.60 (0.024)
F	Min	0.10 (0.004)
	Max	0.25 (0.010)
G	Min	0 degrees
	Max	7 degrees
H	Nom	0.50 (0.020)
J	Min	0.18 (0.007)
	Max	0.28 (0.011)
M	Max	0.10 (0.004)
P	Max	0.05 (0.002)

Notes:
1. Controlling dimension – mm.
2. Coplanarity of all leads shall be within 0.1 mm (difference between the highest and lowest lead with seating plane –K– as reference).
3. Lead pitch determined at –L–.
4. Leadframe to package offset tolerance is ± 0.10 MAX/± .004 MAX.
5. Drawing is not to scale.

MD81 856

Figure 20. 208-Pin MQUAD Mechanical Drawing

LSI LOGIC

LR33020
GraphX Processor
Preliminary

Sales Offices and Design Resource Centers

LSI Logic Corporation
Headquarters
Milpitas CA
■ **Tel: 408.433.8000**
FAX: 408.434.6457

Alabama
■ Tel: 205.883.3527
FAX: 205.883.3529

Arizona
Tel: 602.951.4560
FAX: 602.951.4580

California
San Jose
■ Tel: 408.954.1561
FAX: 408.954.1565

Irvine
■ Tel: 714.553.5600
FAX: 714.474.8101

San Diego
Tel: 619.689.7140
FAX: 619.689.7145

Encino
■ Tel: 818.379.2400
FAX: 818.783.5548

Colorado
Tel: 303.399.1112
FAX: 303.399.6558

Florida
Melbourne
Tel: 407.728.9481
FAX: 407.728.9587

Boca Raton
■ Tel: 407.395.6200
FAX: 407.394.2865

Illinois
■ Tel: 708.995.1600
FAX: 708.995.1622

Maryland
Bethesda
■ Tel: 301.897.5800
FAX: 301.897.8389

Columbia
Tel: 301.740.5664
FAX: 301.740.5603

Massachusetts
■ Tel: 617.890.0180
FAX: 617.890.6158

Minnesota
■ Tel: 612.921.8300
FAX: 612.921.8399

New Jersey
■ Tel: 908.549.4500
FAX: 908.549.4802

New York
Hopewell Junction
Tel: 914.226.1620
FAX: 914.226.1315

Victor
Tel: 716.223.8820
FAX: 716.223.8822

Camillus
Tel: 315.468.1646
FAX: 315.488.2947

North Carolina
■ Tel: 919.783.8833
FAX: 919.783.8909

Ohio
Tel: 513.438.2821
FAX: 513.438.2317

Oregon
Tel: 503.645.9882
FAX: 503.645.6612

Pennsylvania
Tel: 215.830.1404
FAX: 215.657.4920

Texas
Austin
Tel: 512.892.7276
FAX: 512.892.6564

Dallas
■ Tel: 214.788.2966
FAX: 214.233.9234

Washington
■ Tel: 206.822.4384
FAX: 206.827.2884

LSI Logic Corporation
of Canada Inc.
Headquarters
Calgary
■ Tel: 403.262.9292
FAX: 403.262.9494

Burnaby
■ Tel: 604.294.8444
FAX: 604.294.8443

Edmonton
■ Tel: 403.450.4400
FAX: 403.450.4411

Kanata
■ Tel: 613.592.1263
FAX: 613.592.3253

Montreal
■ Tel: 514.694.2417
FAX: 514.694.2699

Toronto
■ Tel: 416.620.7400
FAX: 416.620.5005

Denmark
EV Johannssen Electronic AS
■ Tel: 45.31.839022
FAX: 45.31.839222

Finland
Oy Fintronic AB
Tel: 358.0.6926022
FAX: 358.0.6821251

France
LSI Logic S.A.
■ Tel: 33.146.206600
FAX: 33.146.203138

Microel S.A.
Tel: 33.1.69070824
FAX: 33.1.69071723

Germany
LSI Logic GmbH
European Headquarters
Munich
ßTel: 49.89.99313100
FAX: 49.89.936806

LSI Logic GmbH
Headquarters
Munich
■ ßTel: 49.89.9269030
FAX: 49.89.917096

Dusseldorf
■ Tel: 49.211.5961066
FAX: 49.211.592130

Stuttgart
■ Tel: 49.711.881118
FAX: 49.711.8661359

AE Advanced Electronics
Bad Camberg
Tel: 49.6434.5041
FAX: 49.6434.4277

AE Advanced Electronics
Munich
■ Tel: 49.89.93009855
FAX: 49.89.93009866

Israel
LSI Logic Limited
■ Tel: 972.3.5403741
FAX: 972.3.5403747

Italy
LSI Logic SPA
■ Tel: 39.39.6056881
FAX: 39.39.6057867

Japan
LSI Logic K.K.
Headquarters
Tokyo
■ Tel: 81.33.589.2711
FAX: 81.33.589.2740

Tokyo
Tel: 81.33.5275.1731
FAX: 81.33.5275.1739

Tsukuba-Shi
■ Tel: 81.298.52.8371
FAX: 81.298.52.8376

Osaka
■ Tel: 81.6.947.5281
FAX: 81.6.947.5287

Yokohama
■ Tel: 81.45.902.4111
FAX: 81.45.902.4533

LSI Logic Corporation
of Korea Limited
■ Tel: 82.2.561.2921
FAX: 82.2.554.9327

Netherlands
LSI Logic/Arcobel
■ Tel: 31.4120.30335
FAX: 31.4120.30635

Norway
Art Chip AS
Tel: 47.2.720740
FAX: 47.2.656960

Scotland
LSI Logic Limited
■ Tel: 44.506.416767
FAX: 44.506.414836

Spain
LSI Logic S.A.
■ Tel: 34.1.575600
FAX: 34.1.5702807

Sweden
LSI Logic Limited
■ Tel: 46.8.703.4680
FAX: 46.8.7506647

Switzerland
LSI Logic Sulzer AG
■ ßTel: 41.32.536363
FAX: 41.32.536367

Taiwan
LSI Logic Corporation
■ Tel: 886.2.755.3433
FAX: 886.2.755.5176

United Kingdom
LSI Logic Limited
Headquarters
Bracknell
■ Tel: 44.344.426544
FAX: 44.344.481039

Manhattan Skyline Limited
Maidenhead, Berkshire
Tel: 44.628.75851
FAX: 44.628.782812

■ Sales Offices with Design Resource Centers

This document is preliminary. As such, it contains data derived from functional simulations and performance estimates. LSI Logic has not verified the functional descriptions or electrical and mechanical specifications using production parts.

LSI Logic Corporation reserves the right to make changes to any products and services herein at any time without notice. LSI Logic does not assume any responsibility or liability arising out of the application or use of any product or service described herein, except as expressly agreed to in writing by LSI Logic; nor does the purchase, lease, or use of a product or service from LSI Logic convey a license under any patent rights, copyrights, trademark rights, or any other of the intellectual property rights of LSI Logic or of third parties. All rights reserved.

Printed on Recycled Paper

Printed in USA
101.7111.553800.088.6K.TP.WCP