SUN MICROELECTRONICS

July 1997

# Multi-Cache Controller™

DATA SHEET — Integrated Cache Controller for SuperSPARC

## DESCRIPTION

The STP1091 is a high-performance external cache controller for the STP1020 (SuperSPARC) and STP1021 (SuperSPARC-II) microprocessors. It is used when a large secondary cache or an interface to a non-MBus system is required. The 2.2 million transistor STP1091 supports up to 2Mbytes of direct-mapped secondary cache (XBus™) so that effective memory latencies are reduced. The cache controller also integrates 33x8k synchronous tag RAM to reduce system cost. The STP1091 can be configured to interface with two multi-processor system buses: MBus, a circuit-switched MP bus, and XBus, a packet-switched bus. It isolates the SuperSPARC processor from these system buses, allowing faster processor clock operation with a slower system clock.

The STP1091 is a new version of the multi-cache controller STP1090. Like its predecessor this new part is fully SPARC version 8 compliant and is completely upward compatible with the earlier SPARC version 7 implementations running over 8500 SPARC applications and development tools.

The 50 MHz and 60 MHz versions of the STP1091 are for use with the 50 MHz and 60 MHz versions of the STP1020, respectively. The 75 MHz and 90 MHZ version of the STP1091 is usable for either the 75 MHz version of the STP1021 or the 75MHz and 90 MHz version of the STP1021A respectively.

All references to STP1021 in this document also apply to STP1021A.

### *Features*

- High performance cache controller with 75/90 MHz operating frequency
- Selectable system bus interface
  - SPARC standard MBus or other multiprocessing Buses
- Cache coherency support for multiprocessing
- Processor bus (VBus) and system bus (MBus or XBus) may be operated at different frequencies
- Integrated cache tags and cache controller with support for several external cache sizes
  - 1 Mbyte (MBus) / 512 KBytes, 1 Mbyte, or 2 MBytes (XBus)
- Integrated cache hit/miss monitoring registers
- 8-Bit Boot Bus for ROM and Peripherals (XBus only)
- Built-In Self Test (BIST) logic
- Full JTAG interface (IEEE1149.1)

### *Benefits*

- Delivers optimum STP1021/STP1021A (SuperSPARC) performance
- Increases reliability by reducing number of devices required in systems
- Allows a wide range of scalable systems to be built
- Decouples processor from rest of the system to permit ease of frequency scaleability
- Provides flexibility in external cache configurations for a variety of applications
- Convenient cache performance monitoring support
- Eliminates slow devices from high speed XBus
- Provides quick check of device integrity
- Provides better testability at the board/system level

The STP1091 is intended for use in a broad range of applications from uniprocessor desktop machines to large multiprocessor servers. The STP1091 external cache controller supports multiprocessor configurations using either MBus or XBus interfaces with up to 2 MBytes of secondary cache.

*Figure 1* shows an STP1021 based system using the STP1091 external cache controller in an MBus configuration. In this mode, it supports either no external cache or 1MByte external cache.

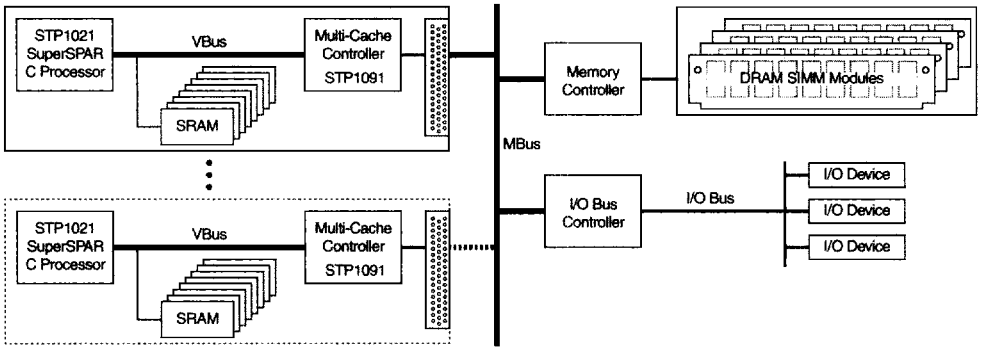



**Figure 1. Typical STP1091 Uniprocessor / Multiprocessor System with MBus Interface**

*Figure 2* shows an STP1021 based system using the STP1091 controller in an XBus configuration. In this mode, it supports cache sizes of 0.5 Mbyte, 1 Mbyte, or 2 MBytes. A maximum of four bus watchers can interface with the STP1091, and each of these can support different system buses.

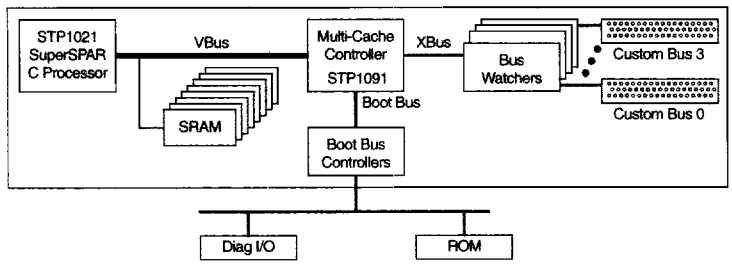



**Figure 2. STP1091 System with XBus Interface and External Bus Watchers**

## TECHNICAL OVERVIEW

### *Architecture*

*Figure 3* shows an overview of the STP1091 microarchitecture. The STP1091 can be partitioned into five large functional blocks:

**Cache Controller Core:** This includes the external cache tag memory, the processor command logic, and the bus command logic. The external cache tag memory keeps track of the usage of the secondary cache. The tag memory is organized as 33 x 8K-bit synchronous memory. In MBus configurations, the tag memory is used for both cache access and bus snooping, whereas in XBus mode, it is used only for cache access. The processor command logic is a group of finite state machines that handle incoming commands from SuperSPARC. They generate bus commands through a request queue. The bus command logic deals with the acknowledgment of these bus requests. The XBus and MBus interfaces place all requests in the input queue, and the bus command logic places corresponding replies in the reply queue.

**Processor Interface:** The STP1091 interfaces to the SuperSPARC processor through the VBus. The processor interface block shields the bus command logic from VBus arbitration by buffering all VBus accesses. It is also responsible for arbitrating the usage of VBus among the SuperSPARC processor, processor command logic, and the bus command logic.

**System Bus Interface:** This includes the MBus interface, the XBus interface, and the XBus arbitration logic. The STP1091 operates in the MBus or XBus mode, as selected by the MBSEL pin. When MBSEL is high, the MBus interface is selected.

**Queues and Synchronizers:** The input queue, request queue, and reply queue are first-in, first-out (FIFO) queues used to communicate between the two clock domains, namely the processor clock and the bus clock. They are implemented with dual-port register files. Control strobes are sent between the two domains through synchronizers, which can be disabled for synchronous operation where both clocks are the same.

**Boot Bus Interface:** The boot bus interface handles all accesses to the 8-bit boot bus. It implements the address and data multiplexing functions on the bus, as well as the automatic polling of interrupts.

The STP1091 also integrates BIST (Built-In-Self-Test) logic, JTAG interface, and has features that support system and software debugging.

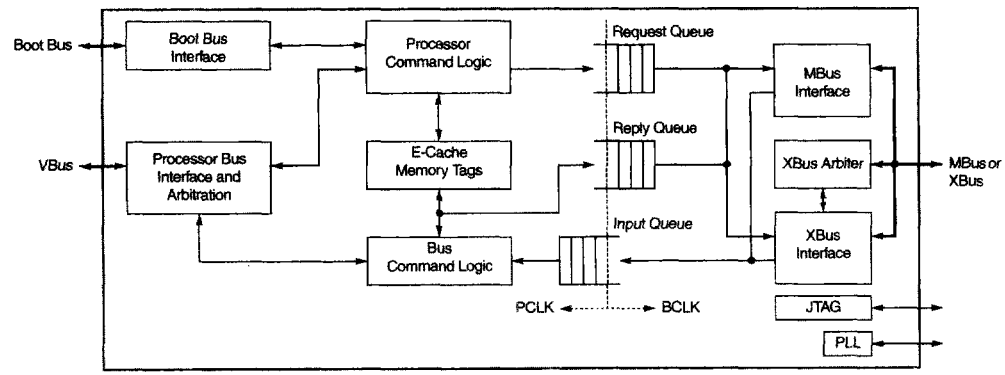



**Figure 3. STP1091 Block Diagram**

## *Modes of Operation*

The STP1091 allows the system designer to select one of the two modes of operation. This selection is designed-in by statically connecting the MBSEL pin to Ground or $V_{CC}$. When the MBSEL pin is tied to Ground, the STP1091 operates with XBus interface. When the MBSEL pin is pulled high, the part operates with a MBus interface. Irrespective of the system bus type, the STP1091 always interfaces to the processor through the VBus.

**TABLE 1: Configuration Changes with MBSEL Pin**

| | | |
|---|---|---|
| Sub-Block Size | 32 bytes | 64 bytes |
| Block Size | 128 bytes | 256 bytes |
| Minimum Cache | 1 MB | 512 KB |
| Maximum Cache | 1 MB | 2 MB |
| Boot-Bus | Not Available | Available |
| Interrupts | From Pins | From XBus Packets |

VBus is a non-multiplexed synchronous bus. It is especially tailored to provide an efficient connection between the STP1021, the STP1091 (the external cache controller), and the external cache memory made up of synchronous SRAMs. It has a 36-bit address bus, and a 64-bit data bus. All transactions on the VBus are synchronized with the STP1021 clock. The arbiter for the VBus transactions is integrated on the STP1091 chip.

In the VBus mode, the STP1021 provides an $\overline{\text{ADDR20}}$ signal besides the ADDR20 signal. By contrast, the STP1020N, and STP1020 do not drive this signal. The $\overline{\text{ADDR20}}$ is now driven out on a pin that used to be spare3. The $\overline{\text{ADDR20}}$ is useful in systems that incorporate 2 MBytes of external cache, and its integration onto the cache controller eliminates an external inverter.

MBus is a SPARC International standard bus designed to function as a processor-independent bus between one or more processors and memory. It is a 64-bit multiplexed high-performance bus. It is fully synchronous with all the transfers controlled by an MBus clock. It supports block transfers in sizes up to 128 bytes with a peak transfer rate of 320 MBytes/s. All transactions on the MBus are arbitrated by an external arbiter. The arbitration algorithm is not included in the MBus definition to allow flexibility in system design. MBus is defined for uniprocessor and multiprocessor systems. The uniprocessor form of MBus is termed "Level1", and the multiprocessor version is called "Level2".

XBus is an extension bus that allows the STP1091 to be connected to one or more system bus interfaces called "Bus Watchers". XBus uses an advanced, synchronous, packet-switched protocol to provide low latency and high bandwidth. It consists of 82 bussed signals, along with three point-to-point arbitration signals per bus watcher. The STP1091 contains a pipelined arbiter that controls accesses to the XBus. In the XBus mode, the STP1091 supports block sizes of 256 bytes.

## SIGNAL DESCRIPTIONS

### TABLE 2: Signal Descriptions - MBus Configuration (MBSEL = H)

| | | |
|---|---|---|
| ADDR[35:0] [1] | I/O | Processor physical address bus. |
| $\overline{\text{ADDR20}}$ | O | Inverted physical address ADDR20. Eliminates an external inverter for 2MB cache systems. |
| $\overline{\text{AERR}}$ | O | Indicates either an internal STP1091 error or $\overline{\text{ERROR}}$ is asserted by the processor.<br>H = No error.<br>L = An internal processor or STP1091 error. |
| BPLLRC | I | Capacitor for the phase filter of the bus clock PLL. This pin should be connected to an external capacitor to ground. With an internal resistor, this circuit provides the RC time constant for the phase filter of the bus clock domain PLL. |
| BURST [1] | I | Indicates whether a burst access is in progress. BURST is driven at the same time as ADDR[35:0], and it is asserted during both read bursts and write bursts. BURST is deasserted on the last address of a burst to allow the STP1091 to stop returning $\overline{\text{RRDY}}$ or $\overline{\text{WRDY}}$ with the last data of the burst.<br>H = A burst access is in progress.<br>L = A burst access is not in progress. |
| $\overline{\text{CCHBL}}$ [1] | I | This pin indicates the current processor transaction as one that may be cached in an external cache.<br>H = Noncacheable access.<br>L = Cacheable access. |
| $\overline{\text{CMDS}}$ [1] | I/O | Command strobe. Indicates the beginning of a bus cycle. The VBus master asserts this signal for one cycle to begin all of its accesses.<br>When the STP1091 is a bus master, as indicated by $\overline{\text{WGRT}}$ and $\overline{\text{RGRT}}$ being deasserted, it asserts $\overline{\text{CMDS}}$ to initiate invalidate and demap transactions.<br>H = Not a command word<br>L = VBus invalidate or demap command word on ADDR[35:0], $\overline{\text{DEMAP}}$, and $\overline{\text{WR}}$.<br>When the STP1091 is not a bus master, this signal indicates the first cycle of a VBus transaction.<br>H = Not a command word.<br>L = VBus command word on ADDR[35:0], $\overline{\text{CCHBL}}$, $\overline{\text{CSA}}$, $\overline{\text{DEMAP}}$, $\overline{\text{LDST}}$, SIZE[1:0], $\overline{\text{SU}}$, $\overline{\text{RD}}$, and $\overline{\text{WR}}$. |
| $\overline{\text{CSA}}$ [1] | I | Control-space access. The processor asserts this pin when performing a read or write to the internal tag RAM, E-cache, or registers of the STP1091.<br>H = Normal memory access.<br>L = Control-space access. |
| DATA[63:0] [1] | I/O | Processor data bus. |
| $\overline{\text{DEMAP}}$ [1] | I | Asserted with $\overline{\text{CMDS}}$ to indicate demap cycle. As an input indicates an external demap cycle.<br>H = Non-demap cycle.<br>L = Demap cycle from system. The TLB entries matching request will be removed. |

**TABLE 2: Signal Descriptions - MBus Configuration (MBSEL = H) (Continued)**

| | | |
|---|---|---|
| DPAR[7:0] [1] | I/O | Data bus parity. When parity is enabled, even parity is generated and checked. DPAR0 is parity for bits DATA[63:56]. When parity checking is disabled, odd parity is generated but not checked.<br>DPAR0: DATA[63:56]<br>DPAR1: DATA[55:48]<br>DPAR4: DATA[31:24]<br>DPAR5: DATA[23:16] |
| $\overline{\text{ERROR}}$ [1] | I | Processor error. The processor asserts this pin when it has entered an internal error state. The STP1091 initiates an internal reset when $\overline{\text{ERROR}}$ is asserted.<br>H = Normal operation.<br>L = Processor internal error. |
| IRL[3:0] | O | Interrupt request level. This field specifies to the processor the level of the highest priority interrupt request that is currently pending. If IRL[3:0] = 0000, no interrupts are pending.<br>Level 15 (IRL[3:0] = 1111): Nonmaskable interrupt.<br>Level 14: Highest maskable interrupt.<br>Level 1: Lowest maskable interrupt.<br>Level 0: No interrupts are pending. |
| $\overline{\text{LDST}}$ [1] | I | This pin indicates an atomic load/store (LDSTUB, LDSTUBA, SWAP, or SWAPA) operation. It is equivalent to the logical OR of $\overline{\text{RD}}$ and $\overline{\text{WR}}$ signals.<br>H = No LDST.<br>L = Atomic load/store (LDST) cycle. |
| MAD[63:0] | I/O | Multiplexed command / data bus. |
| $\overline{\text{MAS}}$ [2] | I/O | MBus address strobe. Asserted by current master when a valid address/command is present on MAD[63:0].<br>H = A valid address/command is not present on MAD[63:0]<br>L = A valid address/command is present on MAD[63:0] |
| $\overline{\text{MBB}}$ [2] | I/O | MBus busy. Asserted when there is any active transaction on MBus.<br>H = MBus free.<br>L = MBus busy. |
| $\overline{\text{MBG}}$ | I | MBus grant. This is a dedicated (not bussed) signal from the MBus arbiter to this bus master.<br>H = Not granted. The STP1021 may not initiate an MBus transaction.<br>L = Granted. The STP1021 may initiate an MBus transaction as soon as MBus is free. |
| $\overline{\text{MBR}}$ | O | MBus request. This is a dedicated (not bussed) signal from the STP1021 to the MBus arbiter.<br>H = No request.<br>L = Requesting to initiate a transaction on MBus. |
| MBSEL [2] | I | MBus select. This pin is used to select the system bus interface. This signal should not be changed during operation of this device.<br>H = MBus system interface<br>L = XBus system interface |
| MCLK | I | Bus clock. |

**TABLE 2: Signal Descriptions - MBus Configuration (MBSEL = H) (Continued)**

<table>
<tr><td>$\overline{MERR}$</td><td>I/O</td><td>MBus error. Encoded along with $\overline{MRDY}$ and $\overline{MRTY}$ to indicate acknowledgment type (the type of error response).
<table>
<tr><th>$\overline{MERR}$</th><th>$\overline{MRDY}$</th><th>$\overline{MRTY}$</th><th>Description</th></tr>
<tr><td>H</td><td>H</td><td>H</td><td>Idle cycle</td></tr>
<tr><td>H</td><td>H</td><td>L</td><td>Relinquish and retry</td></tr>
<tr><td>H</td><td>L</td><td>H</td><td>Valid data transfer</td></tr>
<tr><td>H</td><td>L</td><td>L</td><td>Reserved</td></tr>
<tr><td>L</td><td>H</td><td>H</td><td>Bus error (ERROR1)</td></tr>
<tr><td>L</td><td>H</td><td>L</td><td>Timeout error (ERROR2)</td></tr>
<tr><td>L</td><td>L</td><td>H</td><td>Uncorrectable error (ERROR3)</td></tr>
<tr><td>L</td><td>L</td><td>L</td><td>Retry</td></tr>
</table></td></tr>
<tr><td>$\overline{MEXC}$</td><td>O</td><td>VBus error. Encoded along with $\overline{RDY}$/$\overline{WRDY}$ and $\overline{RETRY}$ to indicate acknowledgment type (the type of error response).
<table>
<tr><th>$\overline{MEXC}$</th><th>$\overline{RRDY}$/$\overline{WRDY}$</th><th>$\overline{RETRY}$</th><th>Description</th></tr>
<tr><td>H</td><td>H</td><td>H</td><td>No reply</td></tr>
<tr><td>H</td><td>H</td><td>L</td><td>Retry</td></tr>
<tr><td>H</td><td>L</td><td>H</td><td>Data transfer complete</td></tr>
<tr><td>H</td><td>L</td><td>L</td><td>Undefined error (UD)</td></tr>
<tr><td>L</td><td>H</td><td>H</td><td>Bus error (BE)</td></tr>
<tr><td>L</td><td>H</td><td>L</td><td>Timeout error (TO)</td></tr>
<tr><td>L</td><td>L</td><td>H</td><td>Reserved</td></tr>
<tr><td>L</td><td>L</td><td>L</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>MID[3:0]</td><td>I</td><td>MBus module ID. The identifier of this MBus device and is usually hardwired by the system. MID3 is the most significant bit (MSB) and MID0 is the least significant bit (LSB).</td></tr>
<tr><td>$\overline{MIH}$ [2]</td><td>I/O</td><td>Memory inhibit. Asserted by a snooping cache when it notices a coherent read of cache block it owns. Memory responds to this signal by ignoring the request.<br>H = No memory inhibit.<br>L = Inhibit memory. The snooping cache which asserted $\overline{MIH}$ will respond with the data in place of memory.</td></tr>
<tr><td>MIRL[3:0]</td><td>I</td><td>Interrupt request level. This field specifies the level of the highest priority interrupt request that is currently pending. If MIRL[3:0] = 0000, no interrupts are pending.<br>Level 15: (MIRL[3:0] = 1111) NMI (disable all traps).<br>Level 14: Highest maskable interrupt.<br>Level 1: Lowest maskable interrupt.<br>Level 0: No interrupts are pending.</td></tr>
<tr><td>$\overline{MRDY}$</td><td>I/O</td><td>MBus ready. Encoded along with $\overline{MERR}$ and $\overline{MRTY}$ to indicate acknowledgment type (the type of error response). See table in $\overline{MERR}$ description.</td></tr>
<tr><td>$\overline{MRTY}$</td><td>I</td><td>MBus retry. Encoded along with $\overline{MERR}$ and $\overline{MRDY}$ to indicate acknowledgment type (the type of error response. See table in $\overline{MERR}$ description.</td></tr>
<tr><td>$\overline{MSH}$ [2] [3]</td><td>I/O</td><td>Memory shared. Asserted by a snooping cache when it notices a coherent read of a cache block it is caching. Both caches will mark the data as shared.<br>H = No sharing.<br>L = Shared data.</td></tr>
<tr><td>$\overline{OE}$ [1]</td><td>I/O</td><td>SRAM output enable. As an output this pin controls the pipelined output enable of external cache SRAM. It is used as an input to prevent bus collisions.<br>H = SRAM outputs disabled<br>L = SRAM outputs enabled</td></tr>
</table>

**TABLE 2: Signal Descriptions - MBus Configuration (MBSEL = H) (Continued)**

| | | |
|---|---|---|
| PCLK | I | Processor clock. Is the same clock as to the processor. |
| $\overline{\text{PEND}}$ | O | Pending. A store is pending in the STP1091 or on the MBus. This pin is asserted by the STP1091 when it has a store operation pending internally or on the system bus. This signal indicates that at least one outstanding write operation has not completed.<br>H = All write operations issued by processor are completed.<br>L = One or more write operations that were issued by processor are not yet complete. |
| $\overline{\text{PLLBYP}}$ [2] | I | PLL bypass. This pin is used to bypass both of the internal phase lock loops. When $\overline{\text{PLLBYP}}$ is asserted, PCLK directly supplies timing for the circuits in the STP1091's processor clock domain, and BCLK directly supplies timing for the circuits of the STP1091's bus clock domain. The normal delay compensation performed by the PLL is defeated.<br>H = PLLs are enabled. Normal operation.<br>L = PLLs are disabled. No clock delay compensation. |
| PPLLRC | | Capacitor for the phase filter of the processor clock PLL. This pin should be connected to an external capacitor to ground. With an internal resistor, this circuit provides the RC time constant for the phase filter of the processor clock domain PLL. |
| $\overline{\text{RD}}$ [1] | I | This pin is asserted when a read address is on ADDR35 - ADDR0. Also asserted with $\overline{\text{DEMAP}}$ to indicate completion of a bus demap operation by the processor.<br>H = No read.<br>L = With $\overline{\text{DEMAP}}$: demap operation requested by the STP1091 is complete. Without $\overline{\text{DEMAP}}$: a data read request. With $\overline{\text{LDST}}$ and $\overline{\text{WR}}$: an atomic load/store operation. |
| $\overline{\text{RESET}}$ | O | Reset. STP1091 output used to reset the processor when the system asserts $\overline{\text{RSTIN}}$.<br>H = Normal operation.<br>L = Reset to processor. |
| $\overline{\text{RETRY}}$ | O | Retry. This pin is encoded, along with $\overline{\text{RRDY}}$ or $\overline{\text{WRDY}}$, and $\overline{\text{MEXC}}$ to indicate the type of acknowledgment. See $\overline{\text{MEXC}}$ description for table. (If this pin is asserted before $\overline{\text{RRDY}}$ or $\overline{\text{WRDY}}$ is asserted for an access, the processor should terminate the current access and restart it once it reacquires the Vbus (if a processor read is pending, a processor write will not be retried until after the read has completed).) |
| $\overline{\text{RGRT}}$ | O | Read grant. This pin grants the processor read access on the VBus.<br>H = Processor not allowed read access.<br>L = Processor may make read accesses. |
| $\overline{\text{RRDY}}$ | O | Read ready. This pin indicates that read data is valid. When $\overline{\text{RRDY}}$ is asserted, the processor may reliably sample the incoming data on the same clock edge as $\overline{\text{RRDY}}$. This signal is used to qualify data specifically for a read access since a write may also be pending. This signal is encoded with $\overline{\text{MEXC}}$ and $\overline{\text{RETRY}}$. See $\overline{\text{MEXC}}$ description for table. |
| $\overline{\text{RSTIN}}$ [2] | I | Reset in. Reset from the system to the cache controller.<br>H = Normal operation.<br>L = Hardware reset (see reset section). |
| SIZE1 [1], SIZE0 [1] | O | These bits indicate the transfer size of the current transaction.<br>00 = Byte<br>01 = Half word<br>10 = Word<br>11 = Doubleword |

**TABLE 2: Signal Descriptions - MBus Configuration (MBSEL = H) (Continued)**

| | | |
|---|---|---|
| $\overline{SU}$ [1] | I | Supervisor access. This signal is asserted by the processor with $\overline{CMDS}$ when the access was initiated in supervisor mode.<br>H = User (unprivileged) transaction.<br>L = Supervisor (privileged) transaction. |
| $\overline{SYNC}$ [2] | I | Synchronous clocks. When this pin is asserted, the synchronizers are bypassed, eliminating their delay but requiring that BCLK and PCLK be identical.<br>H = Asynchronous. PCLK and BCLK may have different rates.<br>L = Synchronous. PCLK and BCLK must be identical. |
| TCK [2] | I | JTAG test clock. |
| TDI [2] | I | JTAG test data. |
| TDO | O | JTAG test data output or PLL output (see $\overline{TEST}$ below). |
| $\overline{TEST}$ | I | Three-state all output drivers and monitor PLL on TDO. |
| TMS | I | JTAG test mode select. |
| $\overline{TRST}$ | I | JTAG test reset. |
| $\overline{WE}$[7-0] [1] | O | SRAM write enables. These signals directly control the write enable signals of synchronous SRAM used as external cache. These pins are driven only when asserted, otherwise they are in the high-impedance state. $\overline{WE}$x bit ordering corresponds to the big-endian convention. That is:<br>$\overline{WE0}$: DATA[63:56] $\overline{WE4}$: DATA[31:24]<br>$\overline{WE1}$: DATA[55:48] $\overline{WE5}$: DATA[23:16]<br>$\overline{WE2}$: DATA[47:40] $\overline{WE6}$: DATA[15:8]<br>$\overline{WE3}$: DATA[39:32] $\overline{WE7}$: DATA[7:0]<br>H = SRAM read<br>L = SRAM write |
| $\overline{WEE}$ | O | E-cache write-enable enable. When asserted, the STP1021 may assert its write enables to write E-cache directly. This pin is used to control the assertion of processor's $\overline{WE}$[7:0] signals.<br>H = The processor may not drive $\overline{WE}$[7:0].<br>L = The processor may drive $\overline{WE}$[7:0]. |
| $\overline{WGRT}$ | O | Write grant. This pin grants the processor write access on the VBus.<br>H = The processor is not allowed write access.<br>L = The processor may make write accesses. |

**TABLE 2: Signal Descriptions - MBus Configuration (MBSEL = H) (Continued)**

| | | |
|---|---|---|
| $\overline{WR}$ [1] | I/O | As an input, this pin is asserted with a write address on ADDR[35:0] and write data on DATA[63:0]. It is also asserted by the processor with $\overline{DEMAP}$ to send a demap request to the system bus.<br>H = Not a write cycle.<br>L = Write (or load/store with $\overline{RD}$ and $\overline{LDST}$ low or demap) cycle.<br><br>As an output the STP1091 asserts this pin with an address on ADDR[35:0] to invalidate lines in the processor's internal cache(s) containing that address.<br>H = Normal<br>L = Demap |
| $\overline{WRDY}$ | O | Write ready. When $\overline{WRDY}$ is asserted, the STP1091 has sampled the processor's write data, and so the processor may generate the next access. In the case of burst writes, the processor switches address and data for the next write within the burst on the same clock edge as $\overline{WRDY}$ was asserted. This pin is used to qualify data specifically for a write access since a read may also be pending. This signal is encoded with $\overline{MEXC}$ and $\overline{RETRY}$. See $\overline{MEXC}$ description for table. |

1. These pins have internal holding drivers.
2. These pins have internal pull-up resistors.
3. These pins have an open drain.

**TABLE 3: Signal Descriptions - XBus Configuration (MBSEL = L)**

| | | |
|---|---|---|
| ADDR[35:0] [1] | I/O | Processor physical address bus. |
| ADDR20 | O | Inverted physical address ADDR20. Eliminates an external inverter for 2MB cache systems. |
| BCLK | I | Bus clock. |
| BPLLRC | I | Capacitor for the phase filter of the bus clock PLL. This pin should be connected to an external capacitor to ground. With an internal resistor, this circuit provides the RC time constant for the phase filter of the bus clock domain PLL. |
| BURST [1] | I | Indicates whether a burst access is in progress. BURST is driven at the same time as ADDR[35:0], and it is asserted during both read bursts and write bursts. BURST is deasserted on the last address of a burst to allow the STP1091 to stop returning RRDY or WRDY with the last data of the burst.<br>H = A burst access is in progress.<br>L = A burst access is not in progress. |
| CCERR | O | Indicates either an internal STP1091 error or ERROR is asserted by the processor<br>H = No error.<br>L = An internal processor or STP1091 error. |
| CCHBL [1] | I | This pin indicates the current processor transaction as one that may be cached in an external cache.<br>H = Noncacheable access.<br>L = Cacheable access. |
| CMDS [1] | I/O | Command strobe. Indicates the beginning of a bus cycle. The VBus master asserts this signal for one cycle to begin all of its accesses. When the STP1091 is a bus master, as indicated by WGRT and RGRT being deasserted, it asserts CMDS to initiate invalidate and demap transactions.<br>H = Not a command word.<br>L = VBus invalidate or demap command word on ADDR[35:0], DEMAP, and WR.<br><br>When the STP1091 is not a bus master, this pin indicates the first cycle of a VBus transaction.<br>H = Not a command word.<br>L = VBus command word on ADDR[35:0], CCHBL, CSA, DEMAP, LDST, SIZE[1:0],SU, RD, and WR. |
| CSA [1] | I | Control-space access. The processor asserts this pin when performing a read or write to the internal tag RAM, E-cache, or registers of the STP1091.<br>H = Normal memory access.<br>L = Control space access. |
| DATA[63:0] | I/O | Processor data bus. |
| DEMAP [1] | I/O | Asserted with CMDS to indicate demap cycle. As an input indicates an external demap cycle.<br>When output:<br>H = Normal command word.<br>L = demap cycle system (system should remove TLB entries matching request).<br>When input:<br>H = Non-demap cycle.<br>L = Demap cycle from system. The TLB entries matching request will be removed. |

**TABLE 3: Signal Descriptions - XBus Configuration (MBSEL = L) (Continued)**

| | | |
|---|---|---|
| DPAR[7:0] [1] | I/O | Data bus parity. When parity is enabled, even parity is generated and checked. DPAR0 is parity for bits DATA63-DATA56. When parity checking is disabled, odd parity is generated but not checked.<br>DPAR0: DATA[63:56]<br>DPAR1: DATA[55:48]<br>DPAR4: DATA[31:24]<br>DPAR5: DATA[23:16] |
| $\overline{\text{ERROR}}$ [1] | I | Processor error. The processor asserts this pin when it has entered an internal error state. The STP1091 initiates an internal reset when $\overline{\text{ERROR}}$ is asserted.<br>H = Normal operation.<br>L = Processor internal error. |
| GTLREF | I | XBus level reference for GTL and GTL/TTL selection. Should be connected to a voltage source of $V_{ref}$ for GTL operation of the XBus interface signals. Should be connected to $V_{CC}$ for TTL operation of the XBus interface signals. Since this pin (and GTLREF1) sets threshold levels, care should be taken to insure that $V_{ref}$ is free of noise. GTLREF and GTLREF1 are connected together internally. |
| GTLREF1 | I | XBus level reference for GTL and GTL/TTL selection. GTLREF and GTLREF1 are connected together internally. |
| IRL[3:0] | O | Interrupt request Level. This field specifies, to the processor, the level of the highest priority interrupt request that is currently pending. If IRL[3:0] = 0000, no interrupts are pending.<br>Level 15: (IRL[3:0] = 1111): Nonmaskable interrupt.<br>Level 14: Highest maskable interrupt.<br>Level 1: Lowest maskable interrupt.<br>Level 0: No interrupts are pending. |
| LCMD[2:0] | O | Boot-bus command bits. Commands are issued by the STP1091 and interpreted by one or more external Boot Bus controllers. |
| $\overline{\text{LCMDS}}$ | O | Boot-bus command strobe. When asserted, this pin indicates that command information on LCMD (and write data on LDATA for WRITE-VALID commands) is valid. Input data is latched on the rising edge.<br>H = Inactive.<br>L = Bus command valid. |
| LDATA[7:0] | I/O | Boot-bus address/data. |

| LCMD2 | LCMD1 | LCMD0 | Name | Description |
|---|---|---|---|---|
| H | H | H | ADR-HIGH | Address bits 23-16 on LDATA |
| H | H | L | Interrupt | Interrupt Status on LDATA |
| H | L | H | ADR-MED | Address bits 15-8 on LDATA |
| H | L | L | ADR-LOW | Address bits 7-0 on LDATA |
| L | H | H | IDLE-WR | Idle for write |
| L | H | L | READ-VALID | Device data on LDATA |
| L | L | H | WRITE-VALID | STP1091 data on LDATA |
| L | L | L | IDLE | Idle |

**TABLE 3: Signal Descriptions - XBus Configuration (MBSEL = L) (Continued)**

| | | |
|---|---|---|
| $\overline{LDST}$ [1] | I | This pin indicates an atomic load/store (LDSTUB, LDSTUBA, SWAP, or SWAPA) operation. It is equivalent to the logical OR of $\overline{RD}$ and $\overline{WR}$ signals. No other transactions may occur while $\overline{LDST}$ is asserted.<br>H = No LDST.<br>L = Atomic load/store (LDST) cycle. |
| MBSEL [2] | I | MBus select. This signal is used to select the system bus interface. This signal should not be changed during operation of this device.<br>H = MBus system interface.<br>L = XBus system interface. |
| $\overline{MEXC}$ | O | VBus error. Encoded along with $\overline{RDY}/\overline{WRDY}$ and $\overline{RETRY}$ to indicate acknowledgment type (the type of error response). |

| $\overline{MEXC}$ | $\overline{RDY}/\overline{WRDY}$ | $\overline{RETRY}$ | Description |
|---|---|---|---|
| H | H | H | No reply |
| H | H | L | Retry |
| H | L | H | Data transfer complete |
| H | L | L | Undefined error (UD) |
| L | H | H | Bus error (BE) |
| L | H | L | Timeout error (TO) |
| L | L | H | Reserved |
| L | L | L | Reserved |

| | | |
|---|---|---|
| $\overline{OE}$ [1] | I/O | SRAM output enable. As an output, this pin controls the pipelined output enable of external cache SRAM. It is used as an input to prevent bus collisions.<br>H = SRAM outputs disabled.<br>L = SRAM outputs enabled. |
| $\overline{PEND}$ | O | Pending. A store is pending in the STP1091 or in the system beyond the STP1091. This signal is asserted by the STP1091 when it has a store operation pending internally or on the system bus. This pin indicates that at least one outstanding write operation has not completed.<br>H = No incomplete write operations outstanding from this processor.<br>L = One or more write operations issued by this processor are not yet complete. |
| PCLK | I | Processor clock. Should be the same as clock to the processor. |
| $\overline{PLLBYP}$ [2] | I | PLL bypass. This pin is used to bypass both of the internal phase lock loop. When $\overline{PLLBYP}$ is asserted PCLK directly supplies timing for the circuits in the STP1091's processor clock domain, and BCLK directly supplies timing for the circuits of the STP1091's bus clock domain. The normal delay compensation performed by the PLL is defeated.<br>H = PLLs are enabled. Normal operation.<br>L = PLLs are disabled. No clock delay compensation. |
| PPLLRC | | Capacitor for the phase filter of the processor clock PLL. This pin should be connected to an external capacitor to ground. With an internal resistor, this circuit provides the RC time constant for the phase filter of the processor clock domain PLL. |
| $\overline{RD}$ [1] | I | This pin is asserted when a read address is on ADDR[35:0]. Also asserted with $\overline{DEMAP}$ to indicate completion of a bus demap operation by the processor.<br>H = No read.<br>L = With $\overline{DEMAP}$: demap operation requested by the STP1091 is complete. Without $\overline{DEMAP}$: a data read request. With $\overline{LDST}$ and $\overline{WR}$: an atomic load/store operation. |

**TABLE 3: Signal Descriptions - XBus Configuration (MBSEL = L) (Continued)**

| | | |
|---|---|---|
| $\overline{\text{RESET}}$ | O | Reset. This STP1091 output is used to reset the processor when the system asserts $\overline{\text{RSTIN}}$.<br>H = Normal operation.<br>L = Reset to processor. |
| $\overline{\text{RETRY}}$ | O | Retry. This pin is encoded, along with $\overline{\text{RRDY}}$ or $\overline{\text{WRDY}}$, and $\overline{\text{MEXC}}$ to indicate the type of acknowledgment. See $\overline{\text{MEXC}}$ description for table. If this signal is asserted before $\overline{\text{RRDY}}$ or $\overline{\text{WRDY}}$ is asserted for an access, the processor should terminate the current access and restart it once it reacquires the Vbus (if a processor read is pending, a processor write will not be retried until after the read has completed). |
| $\overline{\text{RGRT}}$ | O | Read grant. This pin grants the processor read access on the VBus.<br>H = Processor not allowed read access.<br>L = Processor may make read accesses. |
| $\overline{\text{RRDY}}$ | O | Read ready. This pin indicates that read data is valid. When $\overline{\text{RRDY}}$ is asserted, the processor may reliably sample the incoming data on the same clock edge as $\overline{\text{RRDY}}$. This signal is used to qualify data specifically for a read access since a write may also be pending. This signal is encoded with $\overline{\text{MEXC}}$ and $\overline{\text{RETRY}}$. See $\overline{\text{MEXC}}$ description for table. |
| $\overline{\text{RSTIN}}$ [2] | I | Reset in. Reset from the system to the cache controller.<br>H = Normal operation.<br>L = Hardware reset (see reset section). |
| SIZE1 [1], SIZE0 [1] | I | These bits indicate the transfer size of the current transaction.<br>00 = Byte<br>01 = Half word<br>10 = Word<br>11 = Doubleword |
| $\overline{\text{SU}}$ [1] | I | Supervisor access. This pin is asserted by the processor with $\overline{\text{CMDS}}$ when the access was initiated in supervisor mode.<br>H = User (unprivileged) transaction.<br>L = Supervisor (privileged) transaction. |
| $\overline{\text{SYNC}}$ [2] | I | Synchronous clocks. When this pin is asserted, the synchronizers are bypassed, eliminating their delay, but requiring that BCLK and PCLK be identical.<br>H = Asynchronous. PCLK and BCLK may have different rates.<br>L = Synchronous. PCLK and BCLK must be identical. |
| TCK [2] | I | JTAG test clock. |
| TDI [2] | I | JTAG test data. |
| TDO | O | JTAG test data output or PLL output (see $\overline{\text{TEST}}$ below). |
| $\overline{\text{TEST}}$ [2] | I | 3-state all output drivers and monitor PLL on TDO. |
| TMS [2] | I | JTAG test mode select. |
| TRST [2] | I | JTAG test reset. |

**TABLE 3: Signal Descriptions - XBus Configuration (MBSEL = L) (Continued)**

| | | |
|---|---|---|
| $\overline{WE}$[7:0] [1] | O | SRAM write enables. These pins directly control the write enable signals of synchronous SRAM used as external cache. These pin are driven only when asserted, otherwise they are in the high-impedance state. $\overline{WEx}$ bit ordering corresponds to the big-endian convention. That is:<br>$\overline{WE0}$: DATA[63:56] $\overline{WE4}$: DATA[31:24]<br>$\overline{WE1}$: DATA[55:48] $\overline{WE5}$: DATA[23:16]<br>$\overline{WE2}$: DATA[47:40] $\overline{WE6}$: DATA[15:8]<br>$\overline{WE3}$: DATA[39:32] $\overline{WE7}$: DATA[7:0]<br>H = SRAM read<br>L = SRAM write |
| $\overline{WGRT}$ | O | *Write grant. This pin grants the processor write access on the VBus.*<br>H = The processor not allowed write access.<br>L = The processor may make write accesses. |
| $\overline{WR}$ [1] | I/O | As an input, this pin is asserted with a write address on ADDR[35:0]and write data on DATA[63:0]. It is also asserted by the processor with DEMAP to send a demap request to the system bus.<br>H = Not a write cycle.<br>L = Write (or load/store with $\overline{RD}$ and $\overline{LDST}$ low or demap) cycle.<br>As an output, the STP1091 asserts this pin with an address on ADDR[35:0] to invalidate lines in the processors's internal cache(s) containing that address.<br>H = Normal.<br>L = Demap. |
| $\overline{WRDY}$ | O | Write ready. When $\overline{WRDY}$ is asserted, the STP1091 has sampled the processor's write data, and so the processor may generate the next access. In the case of burst writes, the processor switches address and data for the next write within the burst on the same clock edge as $\overline{WRDY}$ was asserted. This pin is used to qualify data specifically for a write access since a read may also be pending. This pin is encoded with $\overline{MEXC}$ and $\overline{RETRY}$. See $\overline{MEXC}$ description for table. |
| XDATA[63:0] [3] | I/O | XBus multiplexed command / data bus. |
| $\overline{XREQ0}$[1] [2]<br>$\overline{XREQ0}$[0] [2] | I | Request field from Bus Watcher 0 (BW0). |
| $\overline{XREQ1}$[1] [2]<br>$\overline{XREQ1}$[0] [2] | I | Request field from Bus Watcher 1 (BW1). |
| $\overline{XREQ2}$[1] [2]<br>$\overline{XREQ2}$[0] [2] | I | Request field from Bus Watcher 2 (BW2). |
| $\overline{XREQ3}$[1] [2]<br>$\overline{XREQ3}$[0] [2] | I | Request field from Bus Watcher 3 (BW3). |
| XPAR[3:0] [3] | I/O | Parity bits.<br>XPAR3 = Parity over XDATA[63:48]<br>XPAR2 = Parity over XDATA[47:32]<br>XPAR1 = Parity over XDATA[31:16]<br>XPAR0 = Parity over XDATA[15:0] |
| $\overline{XGNT0}$ [3] | O | XBus Grant to Bus Watcher 0 (BW0). |

**TABLE 3: Signal Descriptions - XBus Configuration (MBSEL = L) (Continued)**

| | | |
|---|---|---|
| $\overline{XGNT1}$ [3] | O | XBus Grant to Bus Watcher 1 (BW1). |
| $\overline{XGNT2}$ [3] | O | XBus Grant to Bus Watcher 2 (BW2). |
| $\overline{XGNT3}$ [3] | O | XBus Grant to Bus Watcher 3 (BW3). |

1. These pins have internal holding drivers.
2. These pins have internal pull-up resistors.
3. In GTL operation, the I/O buffer is open drain, while in TTL operation the I/O buffer is 3-state.

**TABLE 4: Signal Descriptions - Power Connections**

| | | |
|---|---|---|
| $V_{CCC}$ | I | Supply voltage ($V_{CC}$) for internal (core) logic. |
| $V_{CCCKB}$ | I | Supply voltage ($V_{CC}$) for bus clock and PLL. |
| $V_{CCCKP}$ | I | Supply voltage ($V_{CC}$) for processor clock and PLL. |
| $V_{CCPX}$ | I | Supply voltage ($V_{CC}$) for bus outputs. |
| $V_{CCI}$ | I | Supply voltage ($V_{CC}$) for inputs. |
| $V_{CCP}$ | I | Supply voltage ($V_{CC}$) for processor outputs. |
| $V_{SSC}$ | I | Ground for internal (core) logic. |
| $V_{SSCKB}$ | I | Ground for bus clock and PLL. |
| $V_{SSCKP}$ | I | Ground for processor clock and PLL. |
| $V_{SSI}$ | I | Ground for inputs. |
| $V_{SSP}$ | I | Ground for processor outputs. |
| $V_{SSPX}$ | I | Ground for bus outputs. |

# ELECTRICAL SPECIFICATIONS

## *Absolute Maximum Ratings [1]*

| | | | |
|---|---|---|---|
| $V_{CC}$ | Supply voltage range | 0 to 6.0 | V |
| $V_I$ | Input voltage range [2] | -0.5 to $V_{CC}$ + 0.5 | V |
| $V_O$ | Output voltage range | -0.5 to $V_{CC}$ + 0.5 | V |
| $I_{IK}$ | Input clamp current ($V_I$ < 0 or $V_I$ > $V_{CC}$) | 20 | mA |
| $I_{OK}$ | Output clamp current ($V_O$ < 0 or $V_O$ > $V_{CC}$) | 50 | mA |
| | Current into any output in the low state | 96 | mA |
| $T_{STG}$ | Storage temperature | -65 to 150 | C |

1. Operation of the device at values in excess of those listed above will result in degradation or destruction of the device. All voltages are defined with respect to ground. Functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
2. Unless otherwise noted, all voltages are with respect at $V_{SS}$.

## *Recommended Operating Conditions*

| | | | | | | |
|---|---|---|---|---|---|---|
| $V_{CC}$ | Supply Voltage | For SuperSPARC-II Systems | 4.95 | - | 5.05 | V |
| $V_{SS}$ | Ground | | – | 0 | – | V |
| $V_{IH}$ | TTL high-level input voltage | All except PCLK, BCLK/MCLK | 2.0 | – | $V_{CC}$ + 0.3 | V |
| | | PCLK, BCLK/MCLK | 2.2 | – | $V_{CC}$ + 0.3 | V |
| $V_{IL}$ | TTL low-level input voltage | | -0.3 | – | 0.8 | V |
| $I_{OH}$ | TTL high-level output current, all outputs | | – | – | -370 | A |
| $I_{OL}$ | TTL low-level output current | All outputs except $\overline{MSH}$ | – | – | 2.0 | mA |
| | | $\overline{MSH}$ | – | – | 8.0 | mA |
| $V_{IHG}$ | GTL high-level input voltage | | $V_{REF}$ + 0.15 | – | – | V |
| $V_{ILG}$ | GTL low-level input voltage | | – | – | $V_{REF}$ - 0.15 | V |
| $I_{OHG}$ | GTL high-level output current | | – | – | 10 | A |
| $I_{OLG}$ | GTL low-level output current | | – | – | 36 | mA |
| $V_{REF}$ | GTL reference voltage | | 0.7 | 0.8 | 0.9 | V |
| $T_A$ | Operating ambient temperature | | 0 | – | [1] | C |

1. Maximum ambient temperature is limited by air flow such that the maximum junction temperature does not exceed 80 C

## DC Characteristics

| | | | | | | |
|---|---|---|---|---|---|---|
| $V_{OH}$ | TTL high-level output voltage | $V_{CC}$ = Min, $I_{OH}$ = Max | 2.4 | – | – | V |
| $V_{OL}$ | TTL low-level output voltage | $V_{CC}$ = Max, $I_{OL}$ = Max | – | – | 0.4 | V |
| $V_{OLG}$ | GTL low-level output voltage | $V_{CC}$ = Max, $I_{OLG}$ = Max | – | – | 0.4 | V |
| $V_{OHG}$ | GTL high-level output voltage | $V_{CC}$ = Min, $I_{OHG}$ = Max | 1.2 | – | – | V |
| $I_{CC}$ | Supply current | $V_{CC}$ = Max | – | – | 1.2 | A |
| $I_{CCQ}$ | Quiescent supply current | $V_{CC}$ = Max, $V_I$ = $V_{SS}$ or $V_{CC}$ | – | – | 640 | mA |
| $I_{OZ}$ | High-impedance output current [1] | $V_{CC}$ = Max, $V_O$ = 2.4V | – | – | 20 | A |
| | | $V_{CC}$ = Max, $V_O$ = 0.4V | – | – | -20 | A |
| $I_{IH}$ | Input high current | Inputs with pullups<br>$V_{CC}$ = Max, $V_I$ = Vss to Vcc | – | – | -300 | A |
| | | Inputs with holding drivers [2]<br>$V_{CC}$ = Max, $V_I$ = Vcc | – | – | -250 | A |
| $I_{IL}$ | Input low current | Inputs with holding drivers [2]<br>$V_{CC}$ = Max, $V_I$ = Vss | – | – | 500 | A |
| $I_I$ | Input current | All other inputs<br>$V_{CC}$ = Max, $V_I$ = $V_{SS}$ to $V_{CC}$ | – | – | 50 | A |
| $C_I$ | Input capacitance [2] | | – | 5 | – | pF |
| $C_O$ | Output capacitance [2] | | – | 10 | – | pF |

1. Outputs without holding drivers.
2. This specification is provided as an aid to board design. This specification is not assured during manufacturing testing.

## *VBus Timing - Setup and Hold[1]*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $t_{su}$(VAI1) | VBus setup to PCLK | ADDR[35:20], ADDR[2:0] | 4.5 | – | 3.5 | – | ns |
| $t_{su}$(VAI2) | VBus setup to PCLK | ADDR[19:3] | 4.5 | – | 3.5 | – | ns |
| $t_{su}$(VD) | VBus setup to PCLK | DATA[63:0], DPAR[7:0] | 4.5 | – | 3.5 | – | ns |
| $t_{su}$(VCI1) | VBus setup to PCLK | $\overline{\text{ERROR}}$, $\overline{\text{LDST}}$, $\overline{\text{RD}}$, SIZE1-SIZE0, $\overline{\text{SU}}$ | 7.5 | – | 3.5 | – | ns |
| $t_{su}$(VCI2) | VBus setup to PCLK | $\overline{\text{WR}}$, $\overline{\text{DEMAP}}$ | 7.5 | – | 3.5 | – | ns |
| $t_{su}$(VCI3) | VBus setup to PCLK | BURST, $\overline{\text{CCHBL}}$, $\overline{\text{CSA}}$ | 7.5 | – | 3.5 | – | ns |
| $t_{su}$(CMDS) | VBus setup to PCLK | $\overline{\text{CMDS}}$ | 7.5 | – | 3.5 | – | ns |
| $t_{su}$(OE) | VBus setup to PCLK | $\overline{\text{OE}}$ | 4.5 | – | 3.5 | – | ns |
| $t_h$(VAI1) | VBus hold to PCLK | ADDR[35:20], ADDR[2:0] | 0.5 | – | 0.5 | – | ns |
| $t_h$(VAI2) | VBus hold to PCLK | ADDR[19:3] | 0.5 | – | 0.5 | – | ns |
| $t_h$(VD) | VBus hold to PCLK | DATA[63:0], DPAR[7:0] | 0.5 | – | 0.5 | – | ns |
| $t_h$(VCI1) | VBus hold to PCLK | $\overline{\text{ERROR}}$, $\overline{\text{LDST}}$, $\overline{\text{RD}}$, SIZE1-SIZE0, $\overline{\text{SU}}$ | 0.5 | – | 0.5 | – | ns |
| $t_h$(VCI2) | VBus hold to PCLK | $\overline{\text{WR}}$, $\overline{\text{DEMAP}}$ | 0.5 | – | 0.5 | – | ns |
| $t_h$(VCI3) | VBus hold to PCLK | BURST, $\overline{\text{CCHBL}}$, $\overline{\text{CSA}}$ | 0.5 | – | 0.5 | – | ns |
| $t_h$(CMDS) | VBus hold to PCLK | $\overline{\text{CMDS}}$ | 0.5 | – | 0.5 | – | ns |
| $t_h$(OE) | VBus hold to PCLK | $\overline{\text{OE}}$ | 0.5 | – | 0.5 | – | ns |

1. VBus timings are preliminary based on initial specifications and are subject to change.

## VBus Timing - Switching Characteristics [1] [2]

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $t_p$(VAO1) | Propagation delay, PCLK to ADDR[35:20], ADDR[2:0] | | – | 14.0 | – | 6.0 | ns |
| $t_p$(VAO2) | Propagation delay, PCLK to ADDR[19:3] | | – | 14.0 | – | 6.0 | ns |
| $t_p$(VDO) | Propagation delay, PCLK to DATA[63:0], DPAR[7:0] | | – | 14.0 | – | 5.5 | ns |
| $t_p$(VCO1) | Propagation delay, PCLK to VBus ($\overline{RGRT}$, $\overline{WRGT}$, $\overline{RRDY}$, $\overline{WRDY}$, $\overline{MEXC}$, $\overline{WEE}$, $\overline{RETRY}$, $\overline{PEND}$) | | – | 11.5 | – | 6.0 | ns |
| $t_p$(VCO2) | Propagation delay, PCLK to $\overline{DEMAP}$, $\overline{WR}$ | | – | 11.5 | – | 6.0 | ns |
| $t_p$(CMDS) | Propagation delay, PCLK to $\overline{CMDS}$ | | – | 11.5 | – | 6.0 | ns |
| $t_p$(OE) | Propagation delay, PCLK to $\overline{OE}$ | | – | 11.5 | – | 6.0 | ns |
| $t_p$(WE) | Propagation delay, PCLK to $\overline{WE}$[7:0] | | – | 14.0 | – | 6.0 | ns |
| $t_p$(RESET) | Propagation delay, PCLK to $\overline{RESET}$ | | – | 11.5 | – | 6.0 | ns |
| $t_p$(IRL) | Propagation delay (MBus mode), MIRL to IRL | $I_{OH}$ = Max<br>$I_{OL}$ = Max<br>$V_{load}$ = 2.25V<br>*(see Figure 29)* | – | 20.0 | – | 11.0 | ns |
| | Propagation delay (XBus mode), PCLK to IRL | | – | 15.0 | – | 11.0 | ns |
| $t_{oh}$(VAO1) | Output hold, PCLK to ADDR[35:20], ADDR[2:0] | | 1.0 | – | 0.5 | – | ns |
| $t_{oh}$(VAO2) | Output hold, PCLK to ADDR[19:3] | | 1.0 | – | 0.5 | – | ns |
| $t_{oh}$(VDO) | Output hold, PCLK to DATA[63:0], DPAR[7:0] | | 1.0 | – | 0.5 | – | ns |
| $t_{oh}$(VCO1) | Output hold, PCLK to VBus ($\overline{RGRT}$, $\overline{WRGT}$, $\overline{RRDY}$, $\overline{WRDY}$, $\overline{MEXC}$, $\overline{WEE}$, $\overline{RETRY}$, $\overline{PEND}$) | | 1.0 | – | 0.25 | – | ns |
| $t_{oh}$(VCO2) | Output hold, PCLK to $\overline{DEMAP}$, $\overline{WR}$ | | 1.0 | – | 0.25 | – | ns |
| $t_{oh}$(CMDS) | Output hold, PCLK to $\overline{CMDS}$ | | 1.0 | – | 0.25 | – | ns |
| $t_{oh}$(OE) | Output hold, PCLK to $\overline{OE}$ | | 1.0 | – | 0.25 | – | ns |
| $t_{oh}$(WE) | Output hold, PCLK to $\overline{WE}$[7:0] | | 1.0 | – | 0.25 | – | ns |
| $t_{oh}$(RESET) | Output hold, PCLK to $\overline{RESET}$ | | 1.0 | – | 0.25 | – | ns |
| $t_{oh}$(IRL) | Output hold (XBus mode), PCLK to IRL | | 1.0 | – | 0.25 | – | ns |

1. VBus timings are preliminary based on initial characterization and are subject to change.
2. Switching characteristics are given with maximum number of outputs simultaneously switching.

## *MBus Timing - Setup and Hold [1]*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $t_{su}$(MAD) | MADnn setup to CLK | MAD[63:0] | 5.0 | – | 5.0 | – | ns |
| $t_{su}$(MB) | Bused setup to CLK | $\overline{MAS}$, $\overline{MERR}$, $\overline{MRDY}$, $\overline{MRTY}$, $\overline{MIH}$, $\overline{MBB}$ | 6.0 | – | 5.0 | – | ns |
| $t_{su}$(MPP) | Point-to-point setup to CLK | $\overline{MBG}$ | 8.0 | – | 6.0 | – | ns |
| $t_{su}$(MSH) | $\overline{MSH}$ setup to CLK | $\overline{MSH}$ | 6.0 | – | 5.0 | – | ns |
| $t_h$(MAD) | MAD hold time from CLK | MAD[63:0] | 1.0 | – | 1.0 | – | ns |
| $t_h$(MB) | Bused hold time from CLK | $\overline{MAS}$, $\overline{MERR}$, $\overline{MRDY}$, $\overline{MRTY}$, $\overline{MIH}$, $\overline{MBB}$ | 1.0 | – | 1.0 | – | ns |
| $t_h$(MPP) | Point-to-point hold time from CLK | $\overline{MBG}$ | 1.0 | – | 1.0 | – | ns |
| $t_h$(MSH) | $\overline{MSH}$ hold time from CLK | $\overline{MSH}$ | 1.0 | – | 1.0 | – | ns |

1. The STP1091-50 works with 40 MHz MBUS clock. The STP1091-60 and STP1091-75 work with either 40 or 50 MHz MBUS clock.

## *MBus Timing - Switching Characteristics [1]*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $t_p$(MAD) | Propagation delay, BCLK to MBus MAD[63:0] | $I_{OL}$ = Max<br>$I_{OH}$ = Max<br>$V_{LOAD}$ = 2.25V<br>(See *Figure 29*) | – | 13.5 | – | 11.5 | ns |
| $t_p$(MC1) | Propagation delay, BCLK to MBus control ($\overline{MAS}$, $\overline{MRDY}$, $\overline{MERR}$, $\overline{MIH}$, $\overline{MBB}$) | | – | 13.5 | – | 10.0 | ns |
| $t_p$(MRR) | Propagation delay, BCLK to MBus point-to-point $\overline{MBR}$ | | – | 12.5 | – | 12.0 | ns |
| $t_p$(MSHHL) | Propagation delay, BCLK to $\overline{MSH}$ (high to low) | | – | 13.5 | – | 10.0 | ns |
| $t_p$(MSHLH) | Propagation delay, BCLK to $\overline{MSH}$ (low to high) [2] | | – | [2] | – | [2] | ns |
| $t_p$(AERRHL) | Propagation delay, BCLK to $\overline{AERR}$ (high to low) | | – | 13.5 | – | 10.0 | ns |
| $t_p$(AERRLH) | Propagation delay, BCLK to $\overline{AERR}$ (low to high) [2] | | – | [2] | – | [2] | ns |
| $t_{oh}$(MAD) | Output hold, BCLK to MBus MAD[63:0] | | 1.5 | – | 1.5 | – | ns |
| $t_{oh}$(MC1) | Output hold, BCLK to MBus control ($\overline{MAS}$, $\overline{MRDY}$, $\overline{MERR}$, $\overline{MIH}$, $\overline{MBB}$) | | 2.5 | – | 1.5 | – | ns |
| $t_{oh}$(MBR) | Output hold, BCLK to MBus point-to-point $\overline{MBR}$ | | 2.5 | – | 1.5 | – | ns |
| $t_{oh}$(MSHHL) | Output hold, BCLK to $\overline{MSH}$ (high to low) | | 2.5 | 17.5 | 1.5 | 17.5 | ns |
| $t_{oh}$(MSHLH) | Output hold, BCLK to $\overline{MSH}$ (low to high) | | | | | | |
| $t_{oh}$(AERR) | Output hold, BCLK to $\overline{AERR}$ | | 2.5 | – | 1.5 | – | ns |

1. The STP1091-50 works with 40 MHz MBUS clock. The STP1091-60 and STP1091-75 work with either 40 or 50 MHz MBUS clock.
2. Pins have open-drain implementation. Timing is dependent on external circuits.

## XBus Timing - Setup and Hold [1]

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $t_{su}$(XC) | XBus control (GTL) setup to BCLK | $\overline{XREQn}$[1], $\overline{XREQn}$[0], $\overline{XGNT3}$-$\overline{XGNT0}$ | 6.9 | – | 6.4 | – | ns |
| $t_{su}$(XD) | XBus XDATA (GTL) setup to BCLK | XD[63:0], XPAR[3:0] | 6.9 | – | 6.4 | – | ns |
| $t_h$(XC) | XBus control (GTL) | $\overline{XREQn}$[1], $\overline{XREQn}$[0], $\overline{XGNT3}$-$\overline{XGNT0}$ | 0 | – | 0 | – | ns |
| $t_h$(XD) | XBus XDATA (GTL) | XD[63:0], XPAR[3:0] | 0 | – | 0 | – | ns |

1. The STP1091-50 works with 40 MHz MBUS clock. The STP1091-60 and STP1091-75 work with either 40 or 50 MHz MBUS clock.

## XBus Timing (GTL Mode) - Switching Characteristics [1]

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $t_p$(XC) | BCLK to XBus control ($\overline{XREQn}$[1]-$\overline{XREQn}$[0], $\overline{XGNT3}$-$\overline{XGNT0}$) | $I_{OL}$ = Max<br>$I_{OH}$ = Max<br>$V_{LOAD}$ = 1.2V | – | 8.9 | – | 6.4 | ns |
| $t_p$(XD) | BCLK to XBus XDATA, XPARn | | – | 8.9 | – | 6.4 | ns |
| $t_p$(CCERR) | BCLK to $\overline{CCERR}$ (high to low) | | – | 8.9 | – | 6.4 | ns |
| $t_{oh}$(XC) | BCLK to XBus control ($\overline{XREQn}$[1]-$\overline{XREQn}$[0], $\overline{XGNT3}$-$\overline{XGNT0}$) | | 0.6 | – | 0.6 | – | ns |
| $t_{oh}$(XD) | BCLK to XBus XDATA, XPARn | | 0.6 | – | 0.6 | – | ns |
| $t_{oh}$(CCERR) | CCERR hold time from BCLK (high to low) | | 0.6 | – | 0.6 | – | ns |

1. The STP1091-50 works with 40 MHz MBUS clock. The STP1091-60 and STP1091-75 work with either 40 or 50 MHz MBUS clock.

## Clock Timing

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $t_w$(PCLK) | VBus clock cycle pulse duration | PCLK [1] | 25 | – | – | 11.1 | – | – | ns |
| | PCLK duty cycle | | 25 | 50 | 75 | 25 | 50 | 75 | % |
| $t_w$(BCLK) | XBus clock cycle pulse duration | BCLK | 25 | – | – | 20 | – | – | ns |
| | BCLK duty cycle | | 25 | 50 | 75 | 25 | 50 | 75 | % |
| $t_w$(MCLK) | MBus clock cycle pulse duration | MCLK | 25 | – | – | 20 | – | – | ns |
| | MCLK duty cycle | | 25 | 50 | 75 | 25 | 50 | 75 | % |
| $t_w$(TCK) | JTAG clock cycle pulse duration | TCK | 100 | – | – | 100 | – | – | ns |
| | TCK duty cycle | | 25 | 50 | 75 | 25 | 50 | 75 | % |
| $t_w$(RSTIN) | RSTIN pulse duration [2] [3] | $\overline{RSTIN}$ | 8 | – | – | 8 | – | – | PCLK |
| $t_w$(TRST) | JTAG RST pulse duration [4] | $\overline{TRST}$ | 50 | – | – | 50 | – | – | ns |

1. For asynchronous operation, PCLK must be at least 10% faster than BCLK, but must not exceed a ratio of 3: 1.
2. $\overline{RSTIN}$ must be held asserted for 100 ms on power-up.
3. Functional minimum parameter; not checked by manufacturing test.
4. $\overline{TRST}$ must be asserted for 50 ns after power is applied.

## JTAG and Miscellaneous Timing - Setup and Hold

| | | | | |
|---|---|---|---|---|
| $t_{su}$(RSTIN) | $\overline{RSTIN}$ setup to PCLK (synchronous) [1] | 15 | – | ns |
| $t_{su}$(TDI) | TDI to TCK rising edge | 10 | – | ns |
| $t_{su}$(TMS) | TMS to TCK rising edge | 10 | – | ns |
| $t_h$(RSTIN) | $\overline{RSTIN}$ hold time from PCLK (synchronous) [1] | 15 | – | ns |
| $t_h$(TDI) | TDI hold time from rising edge | 20 | – | ns |
| $t_h$(TMS) | TMS hold time from rising edge | 20 | – | ns |

1. $\overline{RSTIN}$ can occur anytime during a clock cycle. Time given is minimum to ensure synchronous operation.

## JTAG Timing - Switching Characteristics

| | | | | | |
|---|---|---|---|---|---|
| $t_p$(TDO) | TCK (falling edge) to TDO | $I_{OL}$ = Max, $I_{OH}$ = Max | – | 25 | ns |
| $t_{oh}$(TDO) | TCK (falling edge) to TDO | $V_{LOAD}$ = 2.25V (See *Figure 29*) | 5 | – | ns |

## Boot Bus Timing - Setup and Hold

| | | | | |
|---|---|---|---|---|
| $t_{su}$(B) | LDATA setup to CLK | 2 | – | ns |
| $t_h$(B) | LDATA hold from CLK [1] | 2 | – | ns |

1. This value is characterized but not tested. This value is the earliest that the STP1091 will drive data following a read cycle.

## Boot Bus Timing - Switching Characteristics

| | | | | | |
|---|---|---|---|---|---|
| $t_p$(LDATA) | BCLK to LDATA | | – | 15 | ns |
| $t_p$(LCMD) | BCLK to LCMD | | – | 15 | ns |
| $t_p$(LCMDS) | BCLK to LCMDS | | – | 15 | ns |
| $t_p$(LCMDLCMOS) | LCMD valid to $\overline{\text{LCMDS}}$ [1] | | 1 BCLK | – | |
| $t_p$(LDCML) | LDATA valid to $\overline{\text{LCMDS}}$ [1] | $I_{OL}$ = Max | 1 BCLK | – | |
| $t_{oh}$(LDATA) | LDATA hold from BCLK | $I_{OH}$ = Max | 0 | – | ns |
| $t_{oh}$(LCMD) | LCMD hold from BCLK | $V_{LOAD}$ = 2.25V | 0 | – | ns |
| $t_{oh}$(LCMDS) | $\overline{\text{LCMDS}}$ hold from BCLK | | 0 | – | ns |
| $t_{oh}$(LCMDLCMH) | LCMD hold from $\overline{\text{LCMDS}}$ high [1] | | 1 BCLK | – | |
| $t_{oh}$(LDLCMH) | LCMD hold from $\overline{\text{LCMDS}}$ high [1] | | 1 BCLK | – | |
| $t_w$(LCMDS) | $\overline{\text{LCMDS}}$ pulse width [1] | | 3 BCLK | – | ns |

1. Functional minimum parameter; not checked by manufacturing test.

## Holding Drivers and Pull-Ups

Some pins on the STP1091 have holding drivers. Holding drivers are high-impedance buffers that keep the bus at its previous level until a strong driver changes the level. Holding drivers prevent bus signals from drifting near the threshold at low transition rates as can happen on 3-state buses. Holding drivers will source up to 250 A (pull-up) or sink up to 500 A (pull-down).

**TABLE 5: Pins with Holding Drivers**

| | | | |
|---|---|---|---|
| ADDR[35:0] | $\overline{\text{WR}}$ | $\overline{\text{OE}}$ | $\overline{\text{CSA}}$ |
| DATA[63:0] | $\overline{\text{RD}}$ | $\overline{\text{WE}}$(7-0) | $\overline{\text{LDST}}$ |
| DPAR[7:0] | $\overline{\text{CMDS}}$ | SIZE1, SIZE0 | $\overline{\text{CCHBL}}$ |
| BURST | $\overline{\text{DEMAP}}$ | $\overline{\text{ERROR}}$ | $\overline{\text{SU}}$ |

Pins listed in *Table 6* have internal pull-up resistors. These pull-up resistors result in loads that need to be comprehended in system design. The worst-case loading for these inputs is 300 A at $V_{IL}$ of 0.40 volts.

**TABLE 6: Pins with Internal Pull-up Resistors**

| | | | | |
|---|---|---|---|---|
| PLLBYP | TCK | XREQ3 - XREQ1 | XREQ3 - XREQ0 | MSH |
| SYNC | TMS | XREQ2 - XREQ1 | XREQ2 - XREQ0 | MAS |
| RSTIN | TDI | XREQ1 - XREQ1 | XREQ1 - XREQ0 | MIH |
| LDATA[3:0] | TEST | XREQ0 - XREQ1 | XREQ0 - XREQ0 | MBB |
| MBSEL | TRST | | | |

## *External Passive Components*

There are power-supply decoupling capacitors mounted directly on the PGA package of the STP1091. These are eight 0.1 F capacitors, two each between $V_{CCI}$ and $V_{SSI}$, $V_{CCP}$ and $V_{SSP}$, $V_{CCC}$ and $V_{SSP}$, and between $V_{CCPX}$ and $V_{SSPX}$.

The PLLRC and BPLLRC pins require and external capacitor. This capacitor forms part of the RC filter for the phase-lock loop control signal. Each of these pins should have a 0.1 F capacitor to ground for proper operation.

Unused inputs should be pulled high or low. Configuration pins (such as MBSEL) also need to be pulled high or low. A 1 k to 5 k resistor to +5V is recommended for pulling signals high.

## TIMING CONSIDERATIONS

### *VBus Timing*

The VBus read, write and invalidate operations are explained in the following section.

#### *Cache Disabled/Non-Cacheable Single Read*

*Figure 4* shows a single read with the cache disabled. The external cache controller (STP1091) goes to the system bus to accomplish this operation. It deasserts $\overline{\text{RGRT}}$ to allow the STP1021 to complete pending write operations. When the data is available, the STP1090 negates grant, drives the data, and asserts $\overline{\text{RRDY}}$.

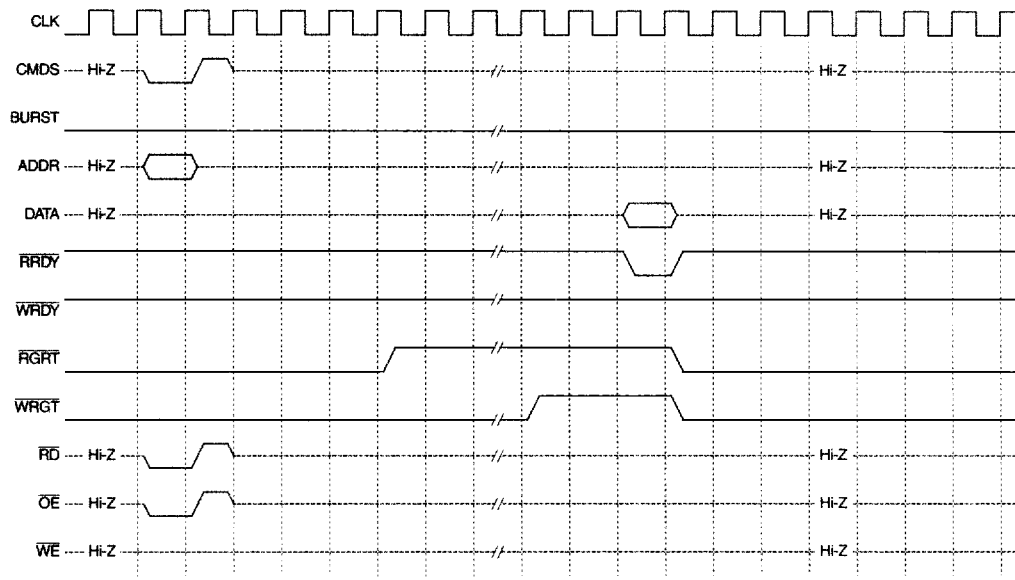



**Figure 4. VBus Cache Disabled/Non-Cacheable Single Read**

### *Cache Disabled Write (or Non-Cacheable) Write*

*Figure 5* shows a cache disabled write. The external cache controller (STP1091) terminates the VBus cycle by issuing a $\overline{\text{WRDY}}$ without asserting $\overline{\text{WEE}}$. A non-cacheable write would be identical.

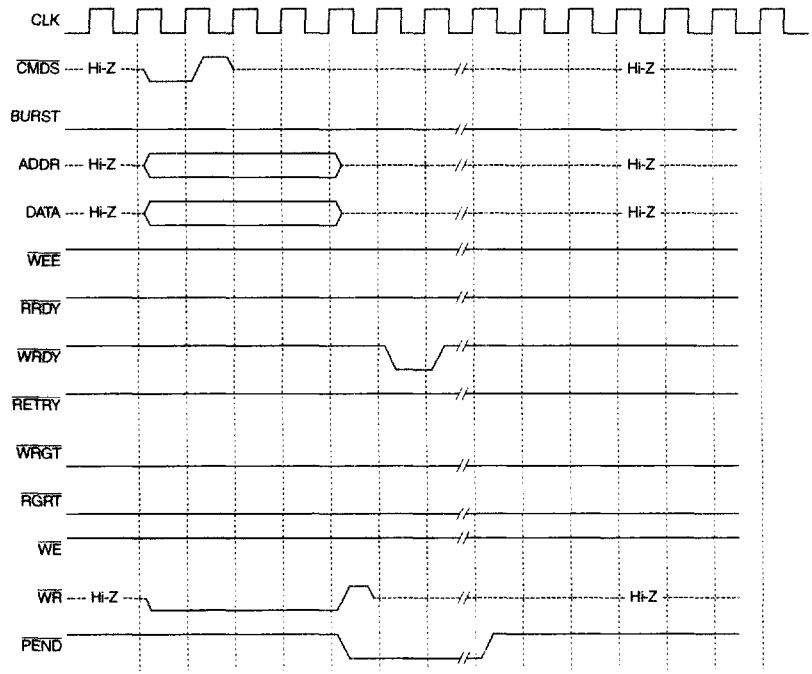



**Figure 5. VBus Cache Disabled/Non-Cacheable Single Write**

### *Cacheable Single Read Hit*

*Figure 6* shows a read by the STP1021 of a single cacheable word with an external cache hit. STP1021 asserts the address, cycle qualifiers, and the $\overline{OE}$ to SRAM. The STP1091 detects a tag match and issues a $\overline{RRDY}$ at the same time that the SRAMs drive data to STP1021. The $\overline{OE}$ from STP1021 is delayed in the registers internal to the synchronous SRAMs, and the data is enabled two cycles after the $\overline{OE}$ is issued to the chip. Note that the partially bussed (not driven by the STP1021 for the entire cycle) VBus control signals are actively deasserted for 1/2 cycle before being released to the bus keepers.

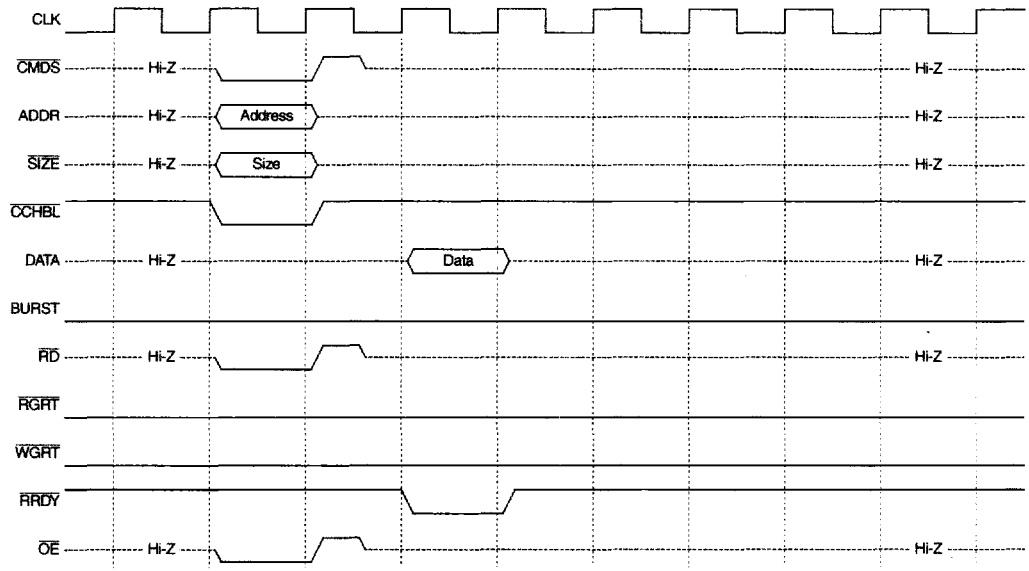



**Figure 6. VBus Cacheable Single Read Hit**

### *Cacheable Single Read Miss*

*Figure 7* shows a cacheable single-read miss. The STP1091 detects that a tag mismatch occurs and issues a cycle to the system bus to obtain data to fill the external cache. It removes $\overline{\text{RGRT}}$ to allow STP1021 to proceed with any write operation it may have had pending. When the system bus returns the requested data block, the STP1091 removes the bus grant to STP1021 (negates $\overline{\text{WGRT}}$) to obtain access to the SRAMs. The STP1091 writes the data into the SRAMs. The STP1091 issues a $\overline{\text{RRDY}}$ to STP1021, as the data word requested (by STP1021 read) is driven on the DATA lines (while the data is being written into SRAMs).

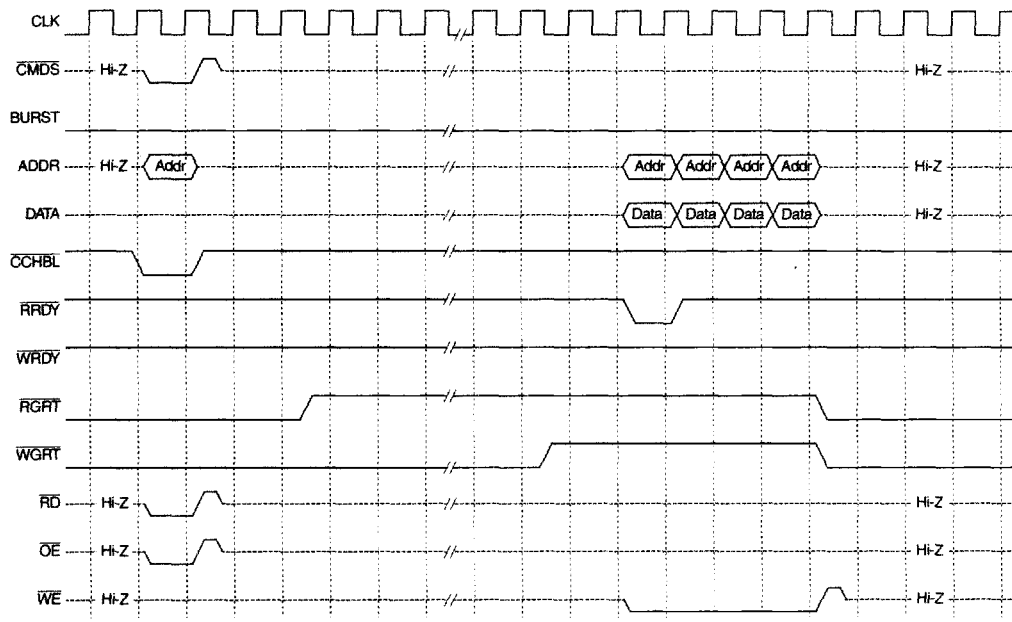



**Figure 7. VBus Cacheable Single Read Miss**

### *Burst Read Hit*

*Figure 8* shows a burst-read hit. As with a cacheable single-read hit, the STP1091 functions mainly to time the cycle by asserting $\overline{\text{RRDY}}$ as the SRAM provides the data.

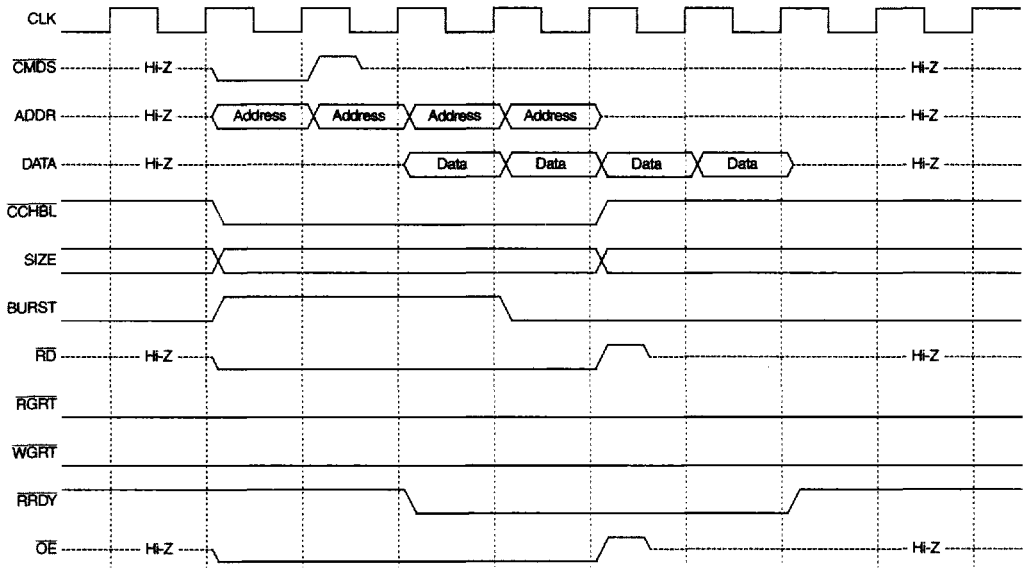



**Figure 8. VBus Burst Read Hit**

### *Burst Read Miss*

*Figure 9* shows a burst read miss. The external cache controller (STP1091) removes $\overline{\text{RRGT}}$ to indicate that the cycle is in progress and that STP1021 can proceed with an outstanding write if one is pending. When the data returns from the system bus, the STP1021 writes it into the SRAM and asserts $\overline{\text{RRDY}}$ when the requested data is on the VBus. Note that, in *Figure 9*, the STP1091 is in XBus configuration, and consequently the block size is 64 bytes. Only 32 bytes are sent to STP1021, while all 64 bytes are stored in SRAM. Also note that with critical word first ordering, the data returned starts from the index into the block for the requested doubleword, continues to the last index, and then wraps from index 0 to the starting index minus 1.

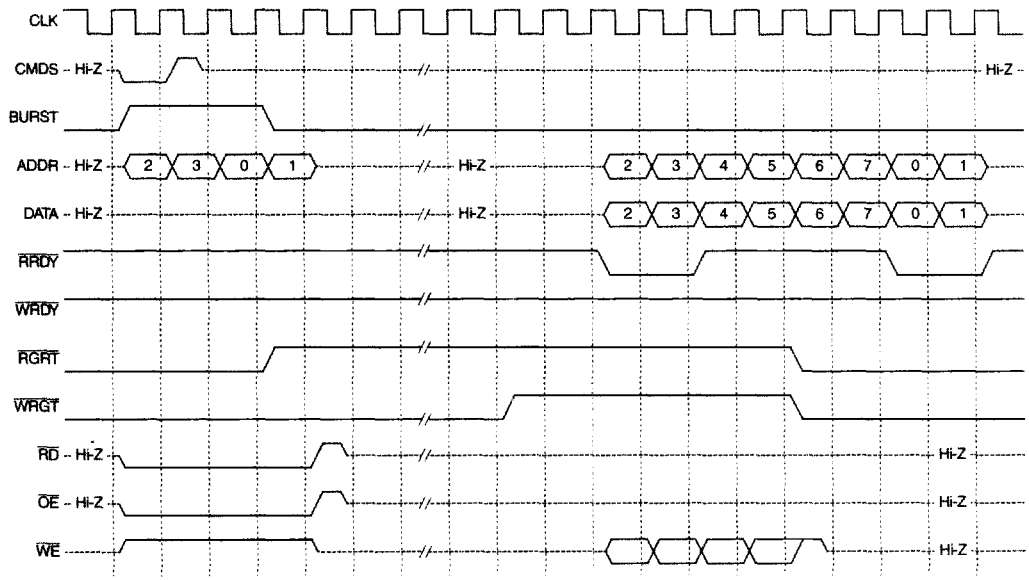



**Figure 9. VBus Burst Read Miss**

### *Cacheable Single Write Hit*

*Figure 10* shows a cacheable single-write hit. The STP1091 asserts $\overline{WEE}$ at the CMD + 2 cycle (i.e., two cycles after $\overline{CMDS}$) to allow the assertion of the write data (DATA, DPAR) and the write strobes ($\overline{WE7}$-$\overline{WE0}$). The STP1091 asserts the $\overline{WRDY}$ in the following cycle ($\overline{CMDS}$ + 3).

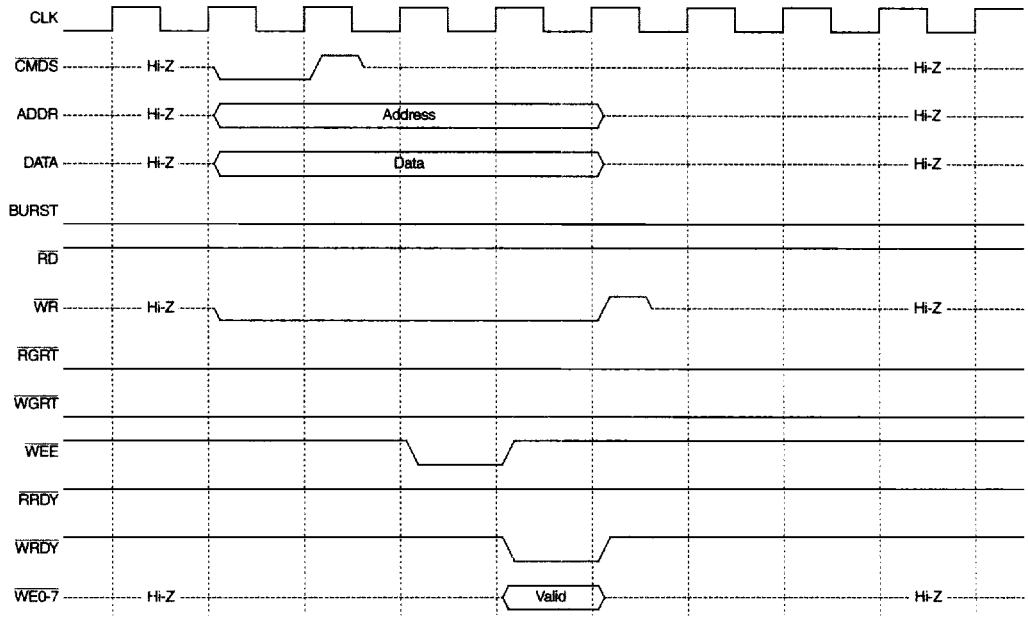



**Figure 10. VBus Cacheable Single Write Hit**

### *Cacheable Burst Write Hit*

*Figure 11* shows a burst write hit. It is basically the same except that $\overline{\text{WRDY}}$ is asserted for each data double-word written in the burst. The STP1021 deasserts BURST one cycle before the last write. Each of the individual writes in the burst from the STP1021 may be from one to eight bytes and may be at any address within the cache block. The number of consecutive writes may be of arbitrary length. If the external cache controller (STP1091) needs the VBus while a burst write cycle is occurring, it can deassert the $\overline{\text{WRGT}}$ signal to terminate the burst cycle prematurely. When the STP1021 reacquires the VBus, it continues the burst write from where it was interrupted.

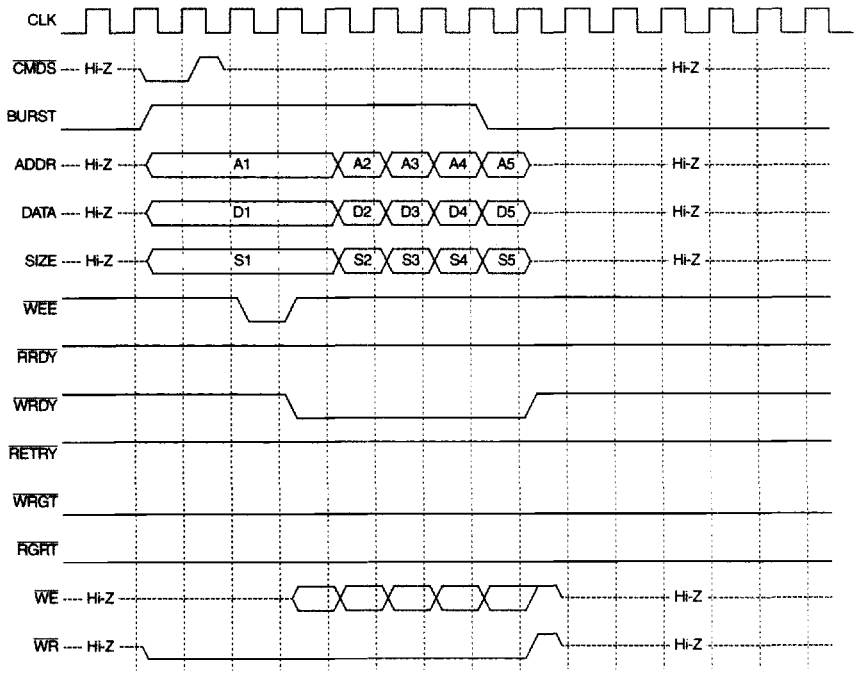



**Figure 11. VBus Cacheable Burst Write Hit**

### *Cache Invalidate*

*Figure 12* shows an invalidate. The external cache controller (STP1091) first removes the STP1021 from the VBus by revoking the $\overline{RGRT}$ and $\overline{WGRT}$ bus grants; it then asserts the address, $\overline{WR}$ and $\overline{CMDS}$. Multiple invalidates may occur consecutively. Invalidates may also occur when the STP1091 has obtained the VBus for SRAM reads or writes.

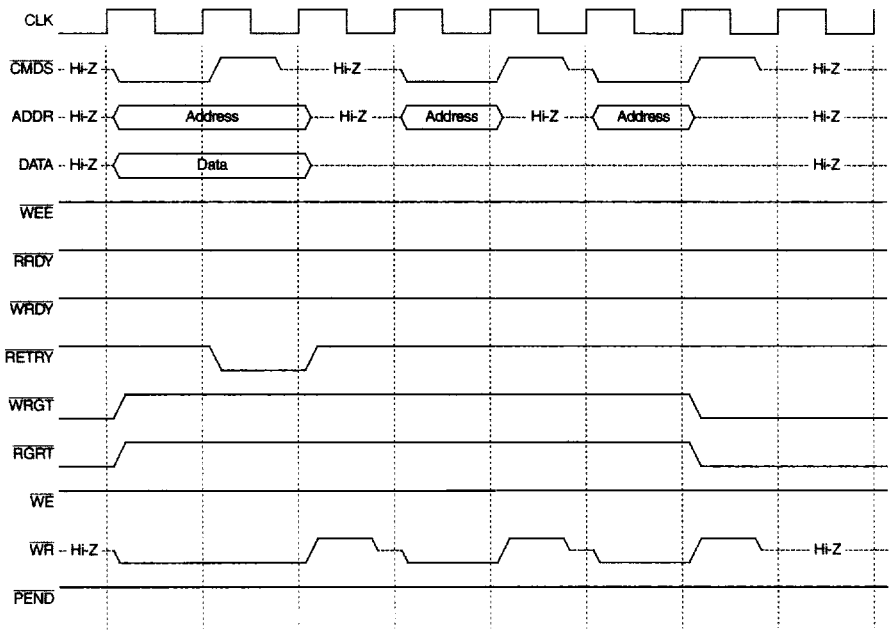


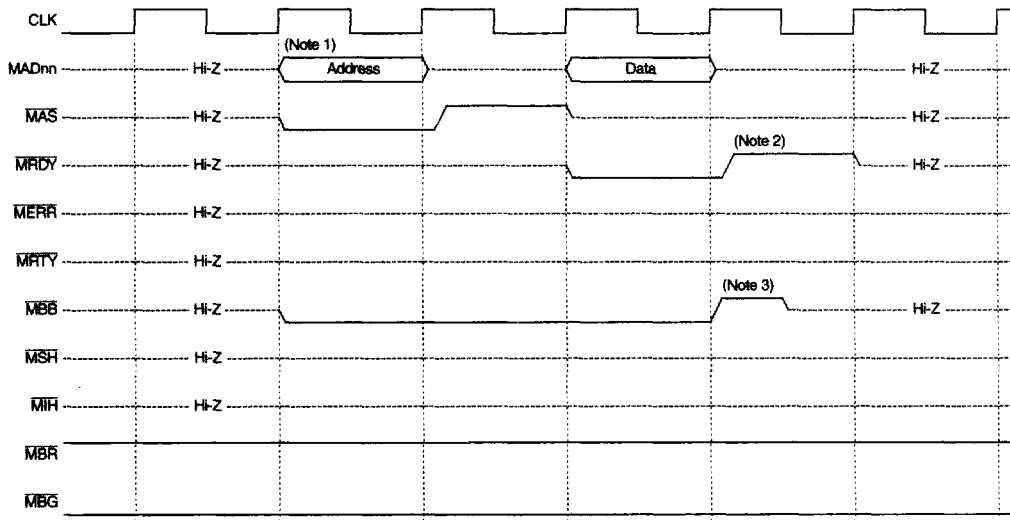

**Figure 12. VBus Invalidation**

## MBus Timing

The MBus read, write and invalidate operations are explained in the following section.

### *MBus Single Read*

The single read cycle transfers a byte, half-word, word, or a double-word. Big-endian word ordering is used (the least significant bytes in a word appear on the high bits of the bus according to SPARC standard). *Figure 13* shows an MBus single read operation.

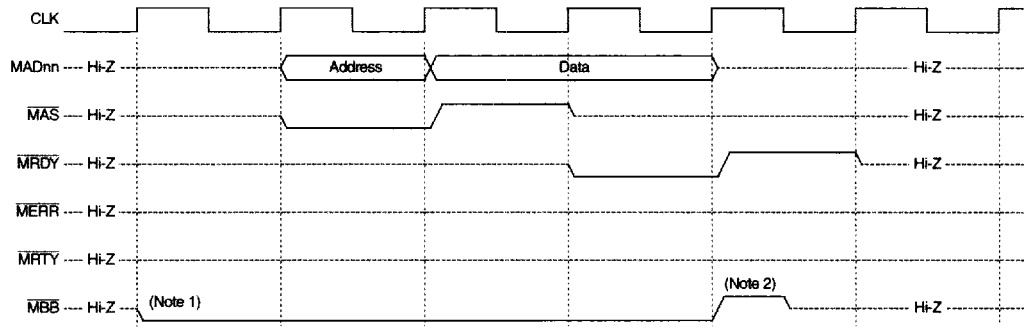



Notes: 1. MADnn lines are held to their previously driven state by system bus holders.
2. Control lines (MAS, MRDY, MERR, MRTY) are driven inactive for one clock before being released.
3. MBB is driven high for 1/2 clock cycle before being released.

**Figure 13. MBus Single Read**

### *MBus Single Write*

Single write operations are queued in the STP1021 store buffer. As soon as the STP1021 receives a bus grant, the transactions will be issued on the bus. The processor will not wait during this time, unless the buffer fills. Bytes, half-words, words, and double words may all be stored, with big-endian ordering. Any errors are reported as deferred data store errors. *Figure 14* shows an MBus single write operation.




Notes: 1. MBB is driven active one cycle before MAS during write and CI cycles.
2. MBB is driven inactive for 1/2 clock cycle before being released.

**Figure 14. MBus Single Write**

### *MBus Burst Read*

*Figure 15* shows a 32-byte burst read operation. A read operation can be performed on any size of data transfer that is specified by the SIZE bits. Read transactions support wrapping (critical word first ordering). Transactions involving fewer than eight bytes will have undefined data on the unused bytes.

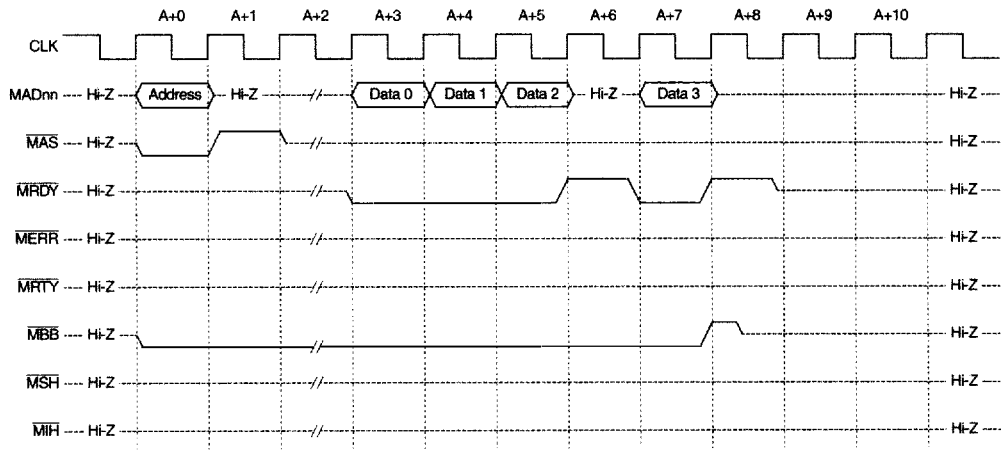
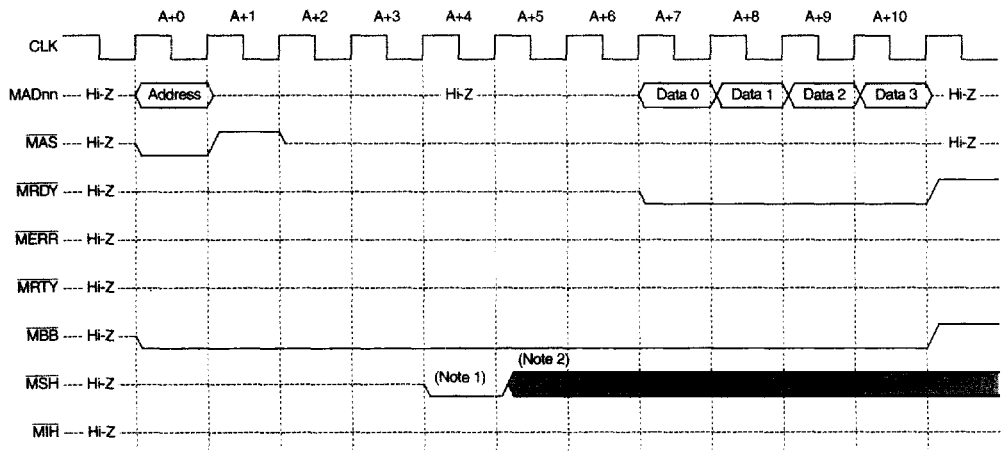



**Figure 15. MBus Burst Read**

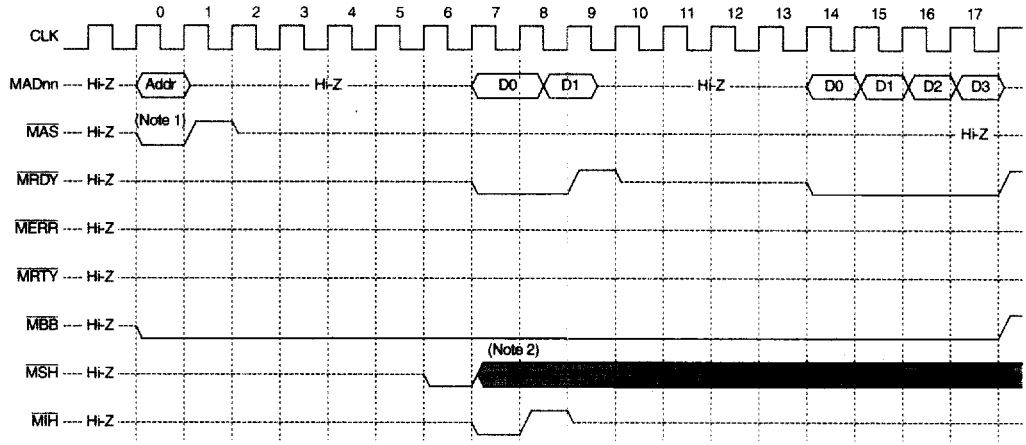
### *MBus Coherent Read*

Coherent Read (CR) transactions are used to read data from the current owner. The owner may be memory or another cache. CR will be used for all on-board data cache load misses and all on-board instruction cache misses. If another cache owns the data, it will respond by asserting the $\overline{MIH}$ signal and providing the data. All CR transactions use critical-word-first ordering. The double-word that is needed first will be the starting address of the transaction. Double-words from memory must be returned in modulo 32-byte address order. Once the needed data arrives, the processor will use it immediately. *Figure 16* shows an MBus coherent read of shared data. Any processor that has a valid cached copy of data referenced by CR transactions must assert the $\overline{MSH}$ signal to indicate that the information is shared. The STP1091 can accept the assertion of $\overline{MSH}$ at any time until receipt of the first data word. If the data is owned by another cache, the STP1091 will ignore any data ready responses until four cycles beyond the assertion of $\overline{MIH}$. This allows memory controllers to begin transmitting data sooner. Memory controllers must not respond with data until a time equal to the maximum $\overline{MIH}$ assertion delay for any cache in the system. *Figure 17* shows an MBus coherent read of owned data.




Notes: 1. $\overline{MSH}$ may occur from A+2 to A+7.
2. $\overline{MSH}$ is an open drain signal. It is not driven inactive. The system pull-up resistor returns it to an inactive level.

**Figure 16. MBUS Coherent Read of Shared Data**




Notes: 1. Device is not the Master.
2. MSH is an open drain signal. It is not driven inactive. The system pull-up resistor returns it to an inactive level.

**Figure 17. MBUS Coherent Read of Owned Data**

### *MBus Coherent Invalidate*

A Coherent Invalidate (CI) operation can only be performed on a block (32 bytes). All CI operations will be snooped by all snooping caches. If a Coherent Invalidate operation hits in a cache, that copy will be invalidated immediately, regardless of its state. Memory is responsible for the acknowledgment of the CI transaction. *Figure 18* shows a CI operation.

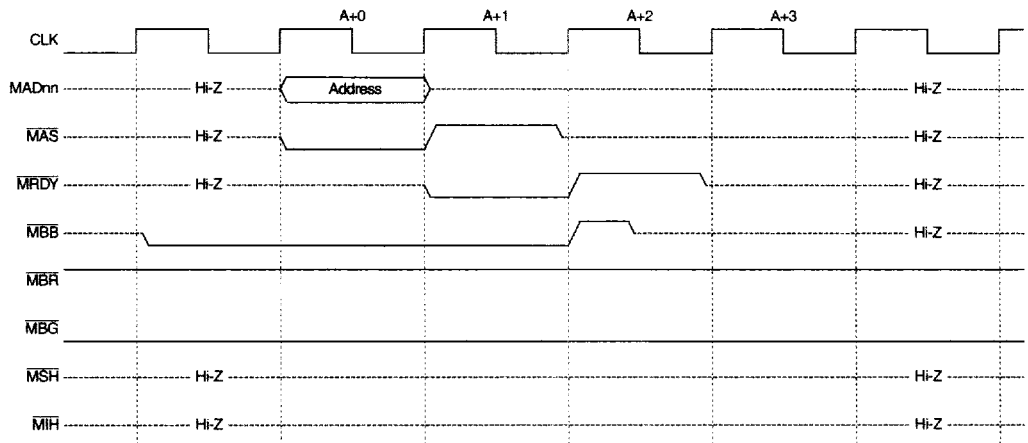



**Figure 18. MBus Coherent Invalidate**

### *Coherent Read and Invalidate*

Since the MBus supports a write-invalidate type of cache-consistency protocol, a special Coherent Read and Invalidate (CRI) transaction that combines a CR transaction with the CI transaction was included to reduce the number of MBus Coherent transactions,. Caches that are performing CR transactions with the knowledge that they intend to immediately modify the data can issue this transaction.

Each CRI transaction will be snooped by all system caches. If the address hits and the cache does not own the block, that cache immediately invalidate its copy of this block, no matter what state the data was in. If the address hits and the cache owns the block, the block will assert $\overline{\text{MIH}}$ and supply the data. When the data has been successfully supplied, the cache will then invalidate its copy of this block.

$\overline{\text{MSH}}$ is not driven during the CRI transaction.

### *Coherent Write and Invalidate*

A Coherent Write and Invalidate transaction combines a block write transaction with a CI transaction.

Each Coherent Write and Invalidate transaction will be snooped by all system caches. If the address hits, caches will invalidate their copies of this block, no matter what state the data was in. Neither $\overline{\text{MIH}}$ nor $\overline{\text{MSH}}$ is asserted for Coherent Write and Invalidate transactions. *Figure 19* shows a Coherent Write and Invalidate operation.

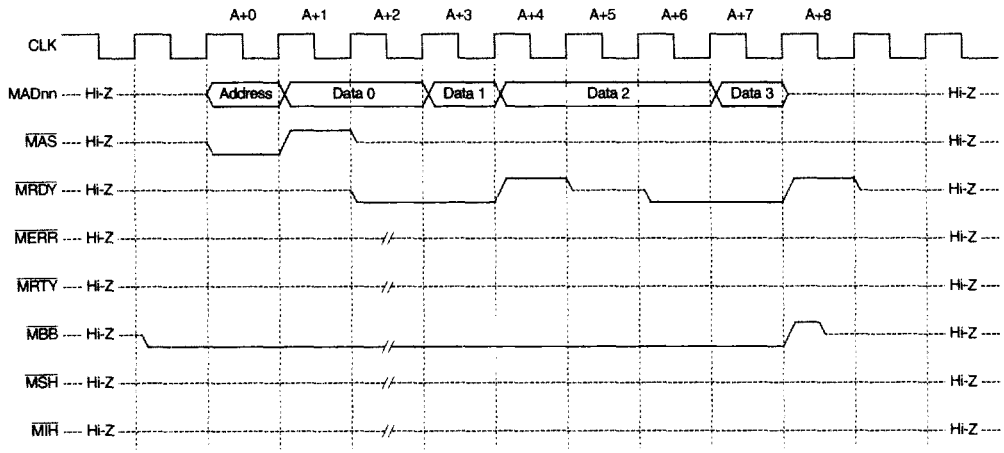
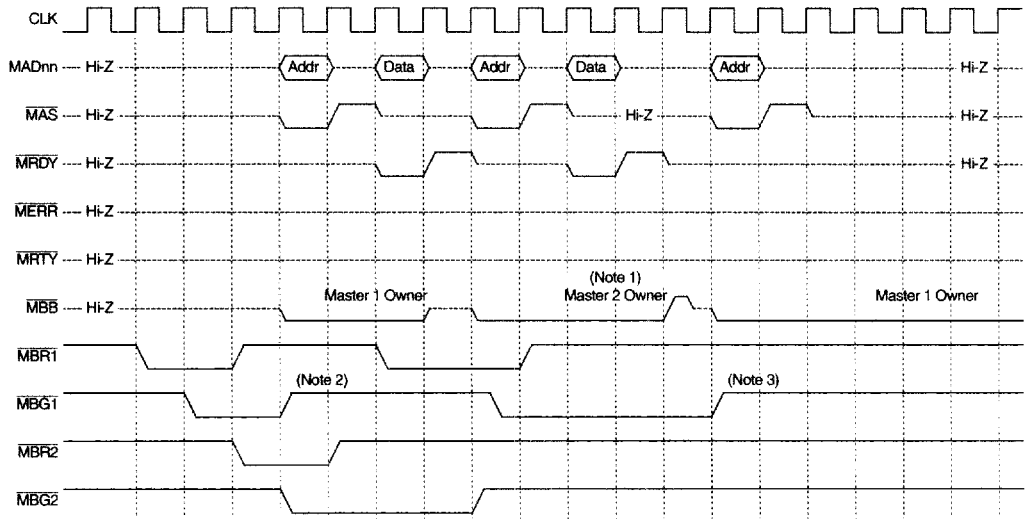



**Figure 19. MBus Coherent Write and Invalidate**

### *Arbitration*

MBus arbitration is accomplished by an external arbiter. The actual arbitration algorithm is implementation dependent. The STP1091 asserts $\overline{MBR}$ when it determines that it requires the MBus. It releases $\overline{MBR}$ immediately after receiving $\overline{MBG}$. $\overline{MBG}$ should remain asserted until $\overline{MBB}$ is negated. The STP1091 normally releases the $\overline{MBB}$ signal at the termination of the cycle (final acknowledgment); however, after an error acknowledgment, $\overline{BB}$ will remain asserted for a number of cycles while the finite state machines complete their error response transitions. A generic example of MBus arbitration is shown in *Figure 20*.




Notes: 1. Master 2 holds the bus by keeping $\overline{MBB}$ asserted.
2. The arbiter releases $\overline{MBG}$ at the earliest possible time. If Master 1 had not asserted $\overline{MBB}$, it would have lost the bus.
3. Arbiter releases $\overline{MBG1}$ when it detects $\overline{MBB}$ deasserted.

**Figure 20. MBUS Arbitration**

## XBus Operation

The XBus is a packet-switched (message) bus. Packet-switched busses differ from conventional circuit-switched busses in that the bus is not held busy for the total transaction. In a circuit-switched bus, a bus master (for example, a processor), that needs a resource (such as memory) arbitrates for the bus and obtains ownership. It supplies a slave address and waits for a response. The slave either accepts or supplies data and signals the master when it finishes. The master then releases the bus.

### *Bus Protocol*

*Cycles*

A bus cycle is one period of the bus clock; it forms the unit of time and one-way information transfer.

All cycles on the XBus fall into one of four categories: HEADER, DATA, MEMFAULT, and IDLE. A header cycle is always the first cycle of a packet; data cycles normally constitute the remaining cycles; MEMFAULT cycles are used to indicate an error in one of the data cycles of a packet; IDLE cycles are those during which no packet is being transmitted on the bus.

HEADER and MEMFAULT cycles are indicated by all-even encoding of parity. A given cycle with all-even encoding is a HEADER cycle if it is the first cycle of a packet: otherwise it is a MEMFAULT cycle.

DATA and IDLE cycles are indicated by the all odd encoding of parity. A given cycle with all-odd encoding is a DATA cycle if it is known to be inside some packet; otherwise, it is an IDLE cycle.

When the parity encoding is neither all-even nor all-odd an error is indicated.

*Packets*

A packet is a contiguous sequence of cycles that constitutes the next higher unit of transfer. The first cycle (header) of a packet carries address and control information, while subsequent cycles carry data. Packets are in two sizes: 2 cycles and 9 cycles.

An XBus device sends a packet after arbitrating for the XBus and getting grant. Packet transmission by a device is uninterruptible once the header cycle has been sent.

A 5-bit DCmd field in each packet encodes the packet type. One of these bits encodes whether the packet is a request or a reply; the other four encode the transmission type.

*Transactions*

A transaction consists of a pair of packets (request, reply) that together performs some logical function.

Packets usually come in pairs, but there are exceptions to this. For the FLUSH LINE transaction, several reply packets may be generated for one request. For a transaction that times out, no reply packet will be generated.

*Packet detection*

Header cycles are indicated by even parity encoding on each of the four parity bits XPAR3-XPAR0. The XBus device uses this information as well as its current XBus state information to recognize a header cycle. Once the header cycle has been recognized, the XBus device expects data. The number of data cycles is determined by the length bit in the message header. Data and idle cycles have the same parity encodings: therefore, the parity cannot be used to distinguish between them.

### *Bus Watchers*

Bus watchers interface XBus to application-specific system busses or devices. Their function is to translate XBus transactions to system-bus or device operations and translate system-bus or device requests to XBus transactions.

At the lowest operational level, bus watchers:

- Receive XBus packets, which request system-bus resources or system bus actions.
- Receive system-bus responses or replies to these requests and map them to XBus reply messages.
- Receive system-bus commands directed to the XBus and convert them to appropriate XBus command packets.
- Receive XBus replies and map them to corresponding system-bus replies.
- Snoop system0-bus operations for references to locally cached data and send messages to the STP1091 to perform coherency operations.

Bus watchers request use of XBus on dedicated lines to the STP1091, which contains the XBus arbiter. The meaning of the $\overline{\text{XREQn}}$[1]-$\overline{\text{XREQn}}$[0] signals depends on the sequence of values on the two lines. The sequences used and their meanings are described in *Table 7*.

**TABLE 7: Arbitration and Flow Control Encoding**

| | | |
|---|---|---|
| HH | – | Idle |
| HL | | Block STP1091 request packets for nine cycles. |
| HL | HL | Block STP1091 request and reply packets for nine cycles. |
| LH | HH | Request XBus for low-priority two-cycle packet. |
| LH | LH | Request XBus for low-priority nine-cycle packet. |
| LH | HL | Request XBus for low-priority two-cycle packet and block STP1091 packets for nine cycles. |
| LH | LL | Request XBus for low-priority nine-cycle packet and block STP1091 packets for nine cycles. |
| LL | HH | Not valid. |
| LL | LH | Request XBus for High-priority nine-cycle packet. |
| LL | HL | Not valid. |
| LL | LL | Request XBus for high-priority nine-cycle packet and block TMX390X55 packets for nine cycles. |

### *Arbitration Priorities*

The Xbus arbiter in STP1091 supports four priorities. Listed in descending priority order, they are:

- **BW HIGH:** XBus arbitration requests from bus watcher to send block read reply packets to STP1091.
- **CC HIGH:** XBus arbitration requests from STP1091 to send reply packets to a bus watcher (Highest priority).

- **BW LOW:** XBus arbitration requests from bus watcher to send system request packets and most system bus reply packets to STP1091.
- **CC LOW:** XBus arbitration requests from STP1091 to send requests packets to a bus watcher (lowest priority).

***XBus Timing Waveforms***

*Figure 21* shows a simple 2-cycle packet. The bus watcher request the use of the bus by asserting "01" binary followed by "11" binary on the $\overline{\text{XREQn1}}$ - $\overline{\text{XREQn0}}$ lines. The STP1091 grants the BW the bus for two cycles and the BW sends a 2-cycle packet.

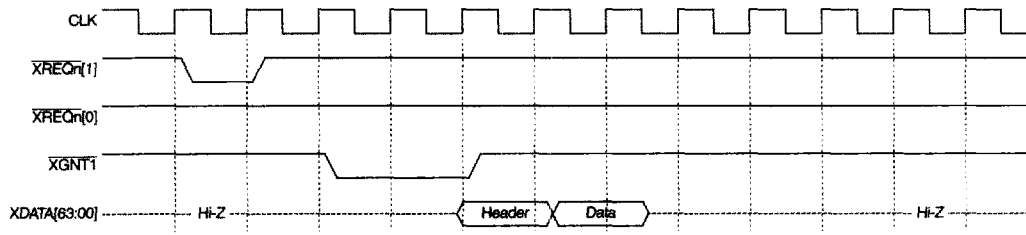



**Figure 21. XBus 2-Cycle Packet**

*Figure 22* shows a 9-cycle packet transmitted by a bus watcher. The $\overline{\text{XREQn}}$ lines are driven "01" binary, followed by another "01" binary. This is a request for a low-priority 9-cycle packet. The STP1091 grants the bus to the BW for nine cycles.

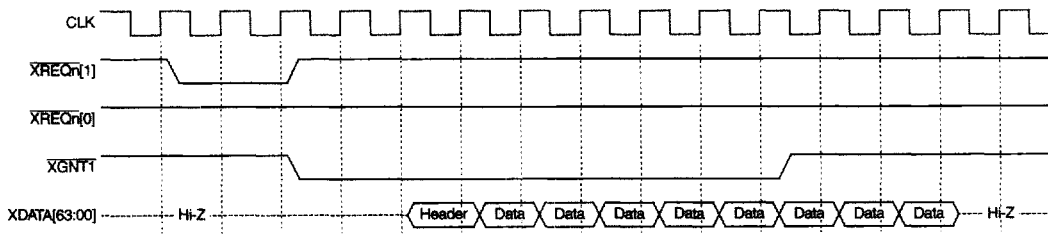



**Figure 22. XBus 9-Cycle Packet**

*Figure 23* shows a STP1091 2-cycle request packet followed by a reply packet from the bus watcher. Note the $\overline{\text{XREQn}}$ arbitration request is for a low-priority 2-cycle packet.

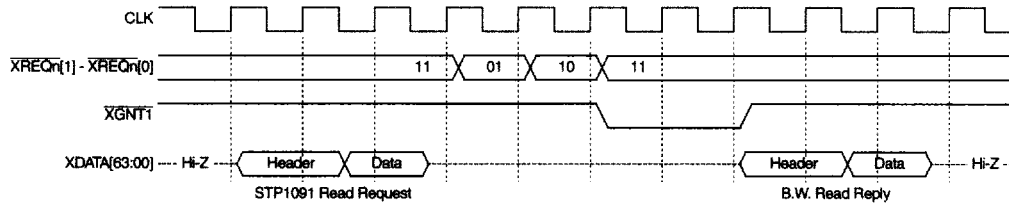



**Figure 23. 2-Cycle Request and 2-Cycle Reply**

*Figure 24* shows a STP1091 9-cycle packet (block write) and a corresponding 2-cycle reply packet.

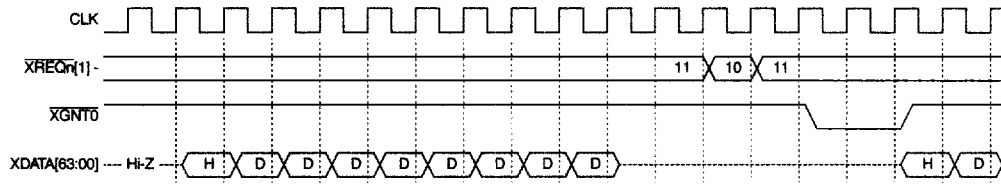



**Figure 24. 9-Cycle Request and 2-Cycle Reply**

## *Boot Bus*

The boot bus is a simple synchronous 12-pin interface provided by the STP1091 for accessing an EPROM for bootstrap loading and for accessing other low-speed peripherals. The boot bus supports an address space of 16M bytes. Provisions are made for reading or writing from 1 to 8 bytes from/to boot-bus devices and for polling the devices for interrupts. Boot bus is available only in the XBus configuration (when MBSEL is low). Boot bus is accessible from both the VBus and the XBus.

**TABLE 8: Boot-Bus Address Decoding**

| | |
|---|---|
| VBus | Noncacheable Space<br>ADDR35-ADDR28= 0xFF<br>ADDR27-ADDR24 = 0x0 or 0x1<br>ADDR23-ADDR00 = Boot-bus address |
| XBus [1] | PA35-PA28 = 0xFn<br>PA27-PA24 = 0x0<br>PA23-PA0 = Boot-bus address |

1. PA = Physical Address

**TABLE 9: Summary of the Boot Bus Physical Signals**

| | | |
|---|---|---|
| LDATA7-LDATA0 | ADDRESS/DATA Bus | BS |
| LCMD2-LCMD0 | Command Bus | BS |
| LCMD5 | Command Strobe | BS |

1. BS Signifies bi-state.

### *Write Valid*

The write valid command instructs the address decoder to write the selected device with the data onLDATA7-LDATA0. See *Figure 25*.

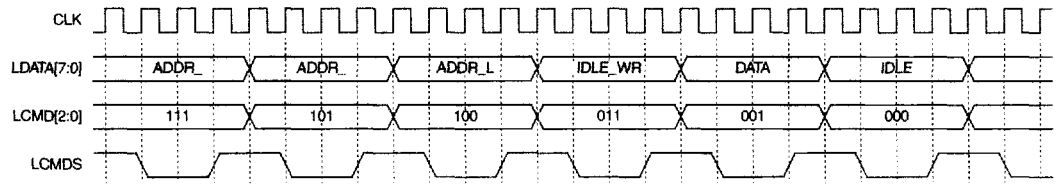



**Figure 25. Boot Bus Write**

### *Read Valid*

The read valid command instructs the address decoder to drive the selected device data ontoLDATA7-LDATA0. See *Figure 26*.

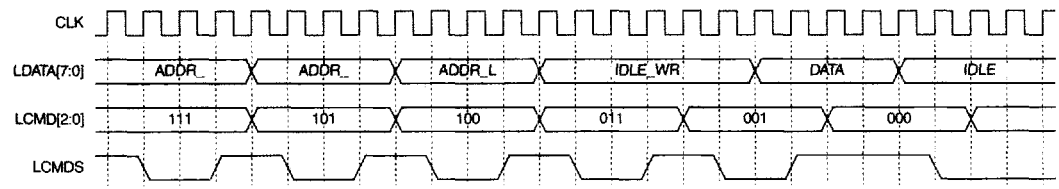



**Figure 26. Boot Bus Read**

## Clocks

In order to reduce system clock skew, a phase-lock loop (PLL) is implemented for each of the clock inputs. For testing and other purposes, a PLL bypass mechanism is provided. When $\overline{\text{PLLBYP}}$ is active, the PLL circuitry of both PCLK and BCLK will be bypassed completely.

### *Phase-Lock Loop Operation*

The PLL operates by constantly measuring internal clock routing and receiver delay and internally generating a clock that is effectively *ahead* of the external clock by an amount equal to the internal routing delay. This ensures that all internal logic sees a clock signal nearly equivalent to that at the external clock pin. All system logic using either PCLK or BCLK is expected to provide acceptable setup and hold times relative to the processor clock input pin.

Prior to normal operation, the PLL must be allowed time to stabilize (i.e., after power up or when PLL has been disabled). During this time, the $\overline{\text{RESET}}$ pin must be asserted. The time required for stabilization is 100 milliseconds.

The input clocks to the STP1091 must never be stopped or changed from normal periodic operation while the PLL is enabled. Doing so will cause PLL instability and unpredictable operation.

To ensure proper operation of the PLL, $V_{CCCKB}$, $V_{SSCKB}$, $V_{CCCKP}$, and $V_{SSCKP}$ should be filtered of system noise. *Figure 27* shows a recommended circuit.

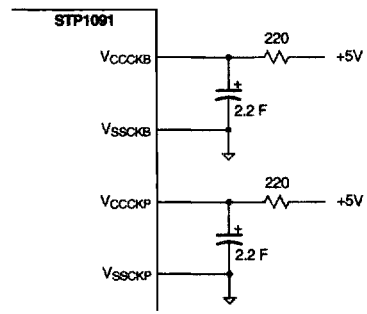



**Figure 27. Typical Phase Lock Loop (PLL) Filter Circuit**

*Note: It is essential that the JTAG TAP controller be reset prior to, or at the same time as $\overline{\text{RESET}}$ in order for the PLL to begin initialization. The TAP controller may be initialized either by asserting the $\overline{\text{TRST}}$ pin or by asserting the TMS pin for five consecutive cycles of TCK (test clock). If this reset does not occur, the PLL clock feedback loop may not be established, and unpredictable operation may result. Whenever the JTAG interface is not in use by a particular system, asserting the $\overline{\text{TRST}}$ signal statically is strongly recommended.*

### *Input Clock Requirements*

The STP1091 can tolerate most clean, stable clock sources when the PLL is enabled. With the PLL enabled, the STP1091 uses only the *rising edge* of the input clocks. Internally, the STP1091 doubles the frequency of the input clocks and then halves them to produce stable clocks with 50% duty cycles. The high time of the input clocks must be between 25% and 75%.

When the PLLs are bypassed, care must be taken to provide a 50% duty cycle on each of the clock inputs. Pin timings for operation with the PLLs bypassed are not fully defined.

IMPORTANT NOTE: Operation in a system with the PLLs bypassed is not recommended or fully specified. Use in this manner will generally require reduced operating frequencies and careful system design.

### *Relationship of PCLK and BCLK*

Due to the design of the internal synchronizers, PCLK must be at least 10% faster than BCLK and the ratio of PCLK to BCLK must not exceed 2.8 to 1. These restrictions are true for asynchronous operation (i.e. the $\overline{\text{SYNC}}$ pin is not asserted).

When the $\overline{\text{SYNC}}$ pin is asserted, BCLK and PCLK must be connected to the same clock with a maximum of 150 ps of skew between them.

## Reset

Reset can come from the system $\overline{\text{RSTIN}}$ (system reset) or from the STP1021. The STP1021 can initiate two different resets; one is watch-dog reset (WD), and the other is software internal reset (SI). Remote processors on the system bus can initiate only software internal resets. The reset register is used to determine the type of reset.

On system reset, the STP1091 will do the following:

- Asynchronously 3-state all DATA/ADDR output drivers on the VBus.
- Asynchronously 3-state all bidirectional output drivers on the MBus/XBus.
- Drive all control strobes on the VBus to high.
- Reset the STP1021 by asserting $\overline{\text{RESET}}$.
- Disable E-cache.
- Reset all finite state machines.
- Reset all internal queues.
- Reset the STP1091 control register, status register, interrupt pending register, and reset register.
- Set Interrupt mask register to 1s.

After system reset, the STP1091 will do the following:

- Continue to reset the STP1021 for eight cycles.
- Configure E-cache tag column redundancy for *150 cycles*. During this period of time, bidirectional control strobes are 3-stated, and unidirectional output control strobes are deasserted. After configuring E-cache column tag redundancy, $\overline{\text{RGRT}}$ and $\overline{\text{WGRT}}$ are asserted.

On software internal reset, the STP1091 deasserts $\overline{RGRT}$ and $\overline{WGRT}$, waits for pending operations to complete (E-cache updates will not be completed), then clears store exception pending (SXP) in the status register and the WD bit in the reset register, and resets the STP1021 for eight cycles. On a software internal reset, the parity enable (PE) bit in the STP1091 and the STP1021 may be different. The system software must ensure that both PE bits are identical before issuing the first write after software internal reset.

On watch-dog reset, the STP1091 will do the following:

- In MBus configuration, assert $\overline{AERR}$.
- Set the WD bit in the reset register.

***Reset Requirements***

To ensure the proper operation of the STP1091, the following requirements must be met by the system for reset:

- At power on of the system, system reset $\overline{RSTIN}$ should be asserted for a minimum of 100 ms after the voltage is within the operating tolerance of the chip. If $\overline{RSTIN}$ is asserted at any other time, it must stay asserted for a minimum of eight BCLK cycles. $\overline{RSTIN}$ can be asynchronous to either or both of BCLK and PCLK.
- JTAG reset ($\overline{TRST}$) must be asserted at power on for a minimum of 50 ns. $\overline{TRST}$ can be asynchronous to any or all of BCLK, PCLK and TCK. Two TCLKs elapse after $\overline{TRST}$ is deasserted before TMS can be asserted.
- After $\overline{RSTIN}$ is deasserted, there should be no requests from XBus or MBus for a minimum of 150 PCLK cycles in order to allow the E-cache tag memory column redundancy programming to complete. Also, there should be no JTAG operations during this time.
- All the 3-state outputs on the MBus or the XBus (as selected by MBSEL) will be placed in their high-impedance state. It is the responsibility of the system logic to assure that these signals remain in their appropriate states with pullups as necessary.
- $\overline{RESET}$ is asserted to the STP1021 asynchronously as soon as $\overline{RSTIN}$ is asserted. The STP1091 keeps asserting $\overline{RESET}$ for eight cycles after $\overline{RSTIN}$ is deasserted. The STP1021 3-states all bidirectional signals on VBus asynchronously when $\overline{RESET}$ is asserted. During $\overline{RSTIN}$ the STP1091 drives the bidirectional VBus signals with weak drivers toward $V_{CC}$. After $\overline{RSTIN}$ is deasserted, the STP1091 drives all the bidirectional control signals to logic high and then releases them before $\overline{RESET}$ is deasserted.
- After a boundary/internal scan test, the $\overline{TRST}$ and $\overline{RESET}$ should be asserted in the same way as during power on reset for the chip to enter normal operation mode.
- $\overline{RSTIN}$ should be held deasserted during internal scan.

***Error Handling***

Errors are handled in four different ways in the STP1091.

- Errors logged to STP1091's error register and reported to the STP1021 through encoding of control strobes $\overline{MEXC}$, $\overline{RRDY}$ (or $\overline{WRDY}$), and $\overline{RETRY}$ are:
  - Errors on a read or a LDST operation.
  - Store exception pending condition of a write miss.
  - Data parity errors on VBus when the STP1091 processor is the master.
  - Errors on a demap initiated by the STP1091 processor.

- Errors are reported to the STP1021 through a level-15 interrupt (for the XBus configuration only). Errors reported in this way are:
  - Asynchronous errors, which include errors of operations that have been acknowledged by STP1091 to the STP1021. These include, for example, stream operations for block copy/zero, shared writes in the XBus configuration, or noncacheable writes, in which errors occur later in the operation.
  - Data parity errors on the VBus when the STP1091 accesses external cache for an incoming bus request.

All these errors are logged into the error register of STP1091. For the MBus configuration, these types of errors are reported to the system by asserting $\overline{\text{AERR}}$.

- Errors are reported to system by asserting $\overline{\text{CCERR}}$. Errors reported in this way are:
  - XBus errors.
  - Cache consistency errors.
  - VBus parity errors on a flush operation.

These errors are considered catastrophic. They are logged into the error register of the STP1091 before $\overline{\text{CCERR}}$ is asserted.

- Errors neither reported or logged. For example, errors on the STP1091 prefetch operation are ignored.

In the MBus configuration, an error on an outgoing request is reported back to the STP1091 with the MBus acknowledgment type by encoding $\overline{\text{MRDY}}$, $\overline{\text{MERR}}$, and $\overline{\text{MRTY}}$.

In the XBus configuration, an error on an outgoing request is reported to the STP1091 in two different ways; the error bit in the header cycle of the reply packet is set, or odd parity is used on a data cycle to indicate a memory fault. In this case, the three least-significant bits of the memory fault data cycle contains the error code.

For MBus and XBus configurations, if a parity error occurs on the VBus when the STP1091 accesses external cache in response to an incoming bus request, a VBus parity error will be reported to the requestor as an uncorrectable error.

Any illegal access from VBus will be reported as a time-out error to STP1091. Illegal accesses from the system bus side are ignored. Atomic load-store to boot bus or the STP1091 registers, out-of-range control space access, and read of interrupt generation register are examples of illegal accesses from VBus.

A parity error on VBus when the STP1021 is the bus master is reported to the STP1021 as an undefined error.

# DEBUG SUPPORT

## TABLE 10: Boundary Scan Bit Order in MBus Mode

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | I | MBSEL | 53 | I | MAD10 | 105 | E | oe-merr | 157 | O | MAD41 | 209 | I | PLLBYP | 261 | O | DATA54 |
| 2 | I | MIRL1 | 54 | O | MAD10 | 106 | I | $\overline{MRDY}$ | 158 | I | MAD42 | 210 | O | IRL0 | 262 | I | DATA53 |
| 3 | O | MIRL1 | 55 | I | MAD11 | 107 | O | $\overline{MRDY}$ | 159 | O | MAD42 | 211 | O | IRL1 | 263 | O | DATA53 |
| 4 | I | MIRL0 | 56 | O | MAD11 | 108 | E | oe-mrdy | 160 | I | MAD43 | 212 | O | IRL2 | 264 | I | DATA52 |
| 5 | O | MIRL0 | 57 | I | MAD12 | 109 | I | $\overline{MRTY}$ | 161 | O | MAD43 | 213 | O | IRL3 | 265 | O | DATA52 |
| 6 | I | MIRL3 | 58 | O | MAD12 | 110 | O | $\overline{MRTY}$ | 162 | I | MAD44 | 214 | I | ADDR24 | 266 | I | DATA51 |
| 7 | O | MIRL3 | 59 | I | MAD13 | 111 | I | $\overline{MBG}$ | 163 | O | MAD44 | 215 | O | ADDR24 | 267 | O | DATA51 |
| 8 | I | MIRL2 | 60 | O | MAD13 | 112 | O | $\overline{MBG}$ | 164 | I | MAD45 | 216 | I | ADDR25 | 268 | I | DATA50 |
| 9 | O | MIRL2 | 61 | I | MAD14 | 113 | E | oe-xpar | 165 | O | MAD45 | 217 | O | ADDR25 | 269 | O | DATA50 |
| 10 | I | — | 62 | O | MAD14 | 114 | I | $\overline{MBB}$ | 166 | I | MAD46 | 218 | I | ADDR26 | 270 | I | DATA49 |
| 11 | O | — | 63 | I | MAD15 | 115 | O | $\overline{MBB}$ | 167 | O | MAD46 | 219 | O | ADDR26 | 271 | O | DATA49 |
| 12 | I | — | 64 | O | MAD15 | 116 | E | oe-mbb | 168 | I | MAD47 | 220 | I | ADDR27 | 272 | I | DATA48 |
| 13 | O | — | 65 | E | oe-mxd2 | 117 | I | $\overline{MAS}$ | 169 | O | MAD47 | 221 | O | ADDR27 | 273 | O | DATA48 |
| 14 | I | — | 66 | E | oe-mxd3 | 118 | O | $\overline{MAS}$ | 170 | E | oemxd10 | 222 | I | ADDR28 | 274 | I | DATA47 |
| 15 | O | — | 67 | I | MAD16 | 119 | E | oe-mas | 171 | E | oemxd11 | 223 | O | ADDR28 | 275 | O | DATA47 |
| 16 | I | — | 68 | O | MAD16 | 120 | I | $\overline{MIH}$ | 172 | I | MAD48 | 224 | I | ADDR29 | 276 | I | DATA46 |
| 17 | O | — | 69 | I | MAD17 | 121 | O | $\overline{MIH}$ | 173 | O | MAD48 | 225 | O | ADDR29 | 277 | O | DATA46 |
| 18 | E | oe-bb-dt | 70 | O | MAD17 | 122 | E | oe-mih | 174 | I | MAD49 | 226 | I | ADDR30 | 278 | I | DATA45 |
| 19 | O | — | 71 | I | MAD18 | 123 | I | $\overline{MSH}$ | 175 | O | MAD49 | 227 | O | ADDR30 | 279 | O | DATA45 |
| 20 | O | — | 72 | O | MAD18 | 124 | O | $\overline{MSH}$ | 176 | I | MAD50 | 228 | E | oeaddr3 | 280 | I | DATA44 |
| 21 | I | MID1 | 73 | I | MAD19 | 125 | E | oe-msh | 177 | O | MAD50 | 229 | I | ADDR31 | 281 | O | DATA44 |
| 22 | O | MID1 | 74 | O | MAD19 | 126 | I | — | 178 | I | MAD51 | 230 | O | ADDR31 | 282 | I | DATA43 |
| 23 | I | MID2 | 75 | I | MAD20 | 127 | I | — | 179 | O | MAD51 | 231 | I | ADDR32 | 283 | O | DATA43 |
| 24 | O | MID2 | 76 | O | MAD20 | 128 | I | — | 180 | I | MAD52 | 232 | O | ADDR32 | 284 | I | DATA42 |
| 25 | O | $\overline{AERR}$ | 77 | I | MAD21 | 129 | I | — | 181 | O | MAD52 | 233 | I | ADDR33 | 285 | O | DATA42 |
| 26 | E | oe-aerr | 78 | O | MAD21 | 130 | O | $\overline{MBR}$ | 182 | I | MAD53 | 234 | O | ADDR33 | 286 | I | DATA41 |
| 27 | I | spare-in | 79 | I | MAD22 | 131 | I | MID3 | 183 | O | MAD53 | 235 | I | ADDR34 | 287 | O | DATA41 |
| 28 | O | spare-out | 80 | O | MAD22 | 132 | O | MID3 | 184 | I | MAD54 | 236 | O | ADDR34 | 288 | I | DATA40 |
| 29 | I | $\overline{RSTIN}$ | 81 | I | MAD23 | 133 | I | MID0 | 185 | O | MAD54 | 237 | I | ADDR35 | 289 | O | DATA40 |
| 30 | I | MAD00 | 82 | O | MAD23 | 134 | O | MID0 | 186 | I | MAD55 | 238 | O | ADDR35 | 290 | E | oe-data5 |
| 31 | O | MAD00 | 83 | E | oe-mxd4 | 135 | O | — | 187 | O | MAD55 | 239 | I | PCLK | 291 | E | oe-data4 |
| 32 | I | MAD01 | 84 | E | oe-mxd5 | 136 | I | MAD32 | 188 | E | oemxd12 | 240 | I | DATA63 | 292 | I | DATA39 |
| 33 | O | MAD01 | 85 | I | MAD24 | 137 | O | MAD32 | 189 | E | oemxd13 | 241 | O | DATA63 | 293 | O | DATA39 |
| 34 | I | MAD02 | 86 | O | MAD24 | 138 | I | MAD33 | 190 | I | MAD56 | 242 | I | DATA62 | 294 | I | DATA38 |
| 35 | O | MAD02 | 87 | I | MAD25 | 139 | O | MAD33 | 191 | O | MAD56 | 243 | O | DATA62 | 295 | O | DATA38 |
| 36 | I | MAD03 | 88 | O | MAD25 | 140 | I | MAD34 | 192 | I | MAD57 | 244 | I | DATA61 | 296 | I | DATA37 |
| 37 | O | MAD03 | 89 | I | MAD26 | 141 | O | MAD34 | 193 | O | MAD57 | 245 | O | DATA61 | 297 | O | DATA37 |
| 38 | I | MAD04 | 90 | O | MAD26 | 142 | I | MAD35 | 194 | I | MAD58 | 246 | I | DATA60 | 298 | I | DATA36 |
| 39 | O | MAD04 | 91 | I | MAD27 | 143 | O | MAD35 | 195 | O | MAD58 | 247 | O | DATA60 | 299 | O | DATA36 |
| 40 | I | MAD05 | 92 | O | MAD27 | 144 | I | MAD36 | 196 | I | MAD59 | 248 | I | DATA59 | 300 | I | DATA35 |
| 41 | O | MAD05 | 93 | I | MAD28 | 145 | O | MAD36 | 197 | O | MAD59 | 249 | O | DATA59 | 301 | O | DATA35 |
| 42 | I | MAD06 | 94 | O | MAD28 | 146 | I | MAD37 | 198 | I | MAD60 | 250 | I | DATA58 | 302 | I | DATA34 |
| 43 | O | MAD06 | 95 | I | MAD29 | 147 | O | MAD37 | 199 | O | MAD60 | 251 | O | DATA58 | 303 | O | DATA34 |
| 44 | I | MAD07 | 96 | O | MAD29 | 148 | I | MAD38 | 200 | I | MAD61 | 252 | I | DATA57 | 304 | I | DATA33 |
| 45 | O | MAD07 | 97 | I | MAD30 | 149 | O | MAD38 | 201 | O | MAD61 | 253 | O | DATA57 | 305 | O | DATA33 |
| 46 | E | oe-mxd0 | 98 | O | MAD30 | 150 | I | MAD39 | 202 | I | MAD62 | 254 | I | DATA56 | 306 | I | DATA32 |
| 47 | E | oe-mxd1 | 99 | I | MAD31 | 151 | O | MAD39 | 203 | O | MAD62 | 255 | O | DATA56 | 307 | O | DATA32 |
| 48 | I | MAD08 | 100 | O | MAD31 | 152 | E | oe-mxd8 | 204 | I | MAD63 | 256 | E | oe-data7 | 308 | O | $\overline{WE3}$ |
| 49 | O | MAD08 | 101 | E | oe-mxd6 | 153 | E | oe-mxd9 | 205 | O | MAD63 | 257 | E | oe-data6 | 309 | O | $\overline{WE2}$ |
| 50 | I | MAD09 | 102 | E | oe-mxd7 | 154 | I | MAD40 | 206 | E | oemxd14 | 258 | I | DATA55 | 310 | O | $\overline{WE1}$ |
| 51 | O | MAD09 | 103 | I | MERR | 155 | O | MAD40 | 207 | E | oemxd15 | 259 | O | DATA55 | 311 | O | $\overline{WE0}$ |
| 52 | I | GTLREF1 | 104 | O | MERR | 156 | I | MAD41 | 208 | I | BCLK | 260 | I | DATA54 | 312 | E | oe-we0 |
| 313 | E | oe-dpar1 | 343 | O | ADDR18 | 373 | E | oe-addr0 | 403 | I | $\overline{CSA}$ | 433 | O | DATA06 | 462 | O | DATA20 |
| 314 | I | DPAR3 | 344 | I | ADDR17 | 374 | I | ADDR03 | 404 | I | $\overline{LDST}$ | 434 | I | DATA07 | 464 | I | DATA21 |

## TABLE 10: Boundary Scan Bit Order In MBus Mode (Continued)

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 315 | O | DPAR3 | 345 | O | ADDR17 | 375 | O | ADDR03 | 405 | I | CCHBL | 435 | O | DATA07 | 465 | O | DATA21 |
| 316 | I | DPAR2 | 346 | I | ADDR16 | 376 | I | ADDR02 | 406 | E | oe-dpar0 | 436 | E | oe-data0 | 466 | I | DATA22 |
| 317 | O | DPAR2 | 347 | O | ADDR16 | 377 | O | ADDR02 | 407 | I | DPAR4 | 437 | E | oe-data1 | 467 | O | DATA22 |
| 318 | I | DPAR1 | 348 | I | ADDR15 | 378 | I | ADDR01 | 408 | O | DPAR4 | 438 | I | DATA08 | 468 | I | DATA23 |
| 319 | O | DPAR1 | 349 | O | ADDR15 | 379 | O | ADDR01 | 409 | I | DPAR5 | 439 | O | DATA08 | 469 | O | DATA23 |
| 320 | I | DPAR0 | 350 | I | ADDR14 | 380 | I | ADDR00 | 410 | O | DPAR5 | 440 | I | DATA09 | 470 | E | oe-data2 |
| 321 | O | DPAR0 | 351 | O | ADDR14 | 381 | O | ADDR00 | 411 | I | DPAR6 | 441 | O | DATA09 | 471 | E | oe-data3 |
| 322 | O | $\overline{\text{RESET}}$ | 352 | I | ADDR13 | 382 | I | $\overline{\text{OE}}$ | 412 | O | DPAR6 | 442 | I | DATA10 | 472 | I | DATA24 |
| 323 | O | $\overline{\text{WEE}}$ | 353 | O | ADDR13 | 383 | O | $\overline{\text{OE}}$ | 413 | I | DPAR7 | 443 | O | DATA10 | 473 | O | DATA24 |
| 324 | I | SIZE1 | 354 | I | ADDR12 | 384 | E | oe-oe | 414 | O | DPAR7 | 444 | I | DATA11 | 474 | I | DATA25 |
| 325 | I | SIZE0 | 355 | O | ADDR12 | 385 | I | $\overline{\text{WR}}$ | 415 | O | $\overline{\text{WE4}}$ | 445 | O | DATA11 | 475 | O | DATA25 |
| 326 | I | $\overline{\text{ERROR}}$ | 356 | E | oe-addr1 | 386 | O | $\overline{\text{WR}}$ | 416 | O | $\overline{\text{WE5}}$ | 446 | I | DATA12 | 476 | I | DATA26 |
| 327 | I | $\overline{\text{SU}}$ | 357 | I | ADDR11 | 387 | E | oe-wr | 417 | O | $\overline{\text{WE6}}$ | 447 | O | DATA12 | 477 | O | DATA26 |
| 328 | I | SYNC | 358 | O | ADDR11 | 388 | I | $\overline{\text{RD}}$ | 418 | O | $\overline{\text{WE7}}$ | 448 | I | DATA13 | 478 | I | DATA27 |
| 329 | I | $\overline{\text{ADDR20}}$ | 359 | I | ADDR10 | 389 | I | BURST | 419 | E | oe-we1 | 449 | O | DATA13 | 479 | O | DATA27 |
| 330 | O | $\overline{\text{ADDR20}}$ | 360 | O | ADDR10 | 390 | O | RETRY | 420 | I | DATA00 | 450 | I | DATA14 | 480 | I | DATA28 |
| 331 | I | ADDR23 | 361 | I | ADDR09 | 391 | O | PEND | 421 | O | DATA00 | 451 | O | DATA14 | 481 | O | DATA28 |
| 332 | O | ADDR23 | 362 | O | ADDR09 | 392 | O | MEXC | 422 | I | DATA01 | 452 | I | DATA15 | 482 | I | DATA29 |
| 333 | I | ADDR22 | 363 | I | ADDR08 | 393 | O | $\overline{\text{WRDY}}$ | 423 | O | DATA01 | 453 | O | DATA15 | 483 | O | DATA29 |
| 334 | O | ADDR22 | 364 | O | ADDR08 | 394 | O | $\overline{\text{RRDY}}$ | 424 | I | DATA02 | 454 | I | DATA16 | 484 | I | DATA30 |
| 335 | I | ADDR21 | 365 | I | ADDR07 | 395 | O | $\overline{\text{WGRT}}$ | 425 | O | DATA02 | 455 | O | DATA16 | 485 | O | DATA30 |
| 336 | O | ADDR21 | 366 | O | ADDR07 | 396 | O | $\overline{\text{RGRT}}$ | 426 | I | DATA03 | 456 | I | DATA17 | 486 | I | DATA31 |
| 337 | I | ADDR20 | 367 | I | ADDR06 | 397 | I | $\overline{\text{CMDS}}$ | 427 | O | DATA03 | 457 | O | DATA17 | 487 | O | DATA31 |
| 338 | O | ADDR20 | 368 | O | ADDR06 | 398 | O | $\overline{\text{CMDS}}$ | 428 | I | DATA04 | 458 | I | DATA18 | | | |
| 339 | E | oe-addr2 | 369 | I | ADDR05 | 399 | E | oe-cmds | 429 | O | DATA04 | 459 | O | DATA18 | | | |
| 340 | I | ADDR19 | 370 | O | ADDR05 | 400 | I | $\overline{\text{DEMAP}}$ | 430 | I | DATA05 | 460 | I | DATA19 | | | |
| 341 | O | ADDR19 | 371 | I | ADDR04 | 401 | O | $\overline{\text{DEMAP}}$ | 431 | O | DATA05 | 461 | O | DATA19 | | | |
| 342 | I | ADDR18 | 372 | O | ADDR04 | 402 | E | oe-dmap | 432 | I | DATA06 | | I | DATA20 | | | |

## TABLE 11: Boundary Scan Bit Order in XBus Mode

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | I | MBSEL | 53 | I | XDATA10 | 105 | E | oe-merr | 157 | O | XDATA41 | 209 | I | PLLBYP | 261 | O | DATA54 |
| 2 | I | LDATA7 | 54 | O | XDATA10 | 106 | I | XPAR1 | 158 | I | XDATA42 | 210 | O | IRL0 | 262 | I | DATA53 |
| 3 | O | LDATA7 | 55 | I | XDATA11 | 107 | O | XPAR1 | 159 | O | XDATA42 | 211 | O | IRL1 | 263 | O | DATA53 |
| 4 | I | LDATA6 | 56 | O | XDATA11 | 108 | E | oe-mrdy | 160 | I | XDATA43 | 212 | O | IRL2 | 264 | I | DATA52 |
| 5 | O | LDATA6 | 57 | I | XDATA12 | 109 | I | XPAR2 | 161 | O | XDATA43 | 213 | O | IRL3 | 265 | O | DATA52 |
| 6 | I | LDATA5 | 58 | O | XDATA12 | 110 | O | XPAR2 | 162 | I | XDATA44 | 214 | I | ADDR24 | 266 | I | DATA51 |
| 7 | O | LDATA5 | 59 | I | XDATA13 | 111 | I | XPAR3 | 163 | O | XDATA44 | 215 | O | ADDR24 | 267 | O | DATA51 |
| 8 | I | LDATA4 | 60 | O | XDATA13 | 112 | O | XPAR3 | 164 | I | XDATA45 | 216 | I | ADDR25 | 268 | I | DATA50 |
| 9 | O | LDATA4 | 61 | I | XDATA14 | 113 | E | oe-xpar | 165 | O | XDATA45 | 217 | O | ADDR25 | 269 | O | DATA50 |
| 10 | I | LDATA3 | 62 | O | XDATA14 | 114 | I | $\overline{\text{XREQ0}}$[0] | 166 | I | XDATA46 | 218 | I | ADDR26 | 270 | I | DATA49 |
| 11 | O | LDATA3 | 63 | I | XDATA15 | 115 | O | $\overline{\text{XREQ0}}$[0] | 167 | O | XDATA46 | 219 | O | ADDR26 | 271 | O | DATA49 |
| 12 | I | LDATA2 | 64 | O | XDATA15 | 116 | E | oe-mbb | 168 | I | XDATA47 | 220 | I | ADDR27 | 272 | I | DATA48 |
| 13 | O | LDATA2 | 65 | E | oe-mxd2 | 117 | I | $\overline{\text{XREQ0}}$[1] | 169 | O | XDATA47 | 221 | O | ADDR27 | 273 | O | DATA48 |
| 14 | I | LDATA1 | 66 | E | oe-mxd3 | 118 | O | $\overline{\text{XREQ0}}$[1] | 170 | E | oemxd10 | 222 | I | ADDR28 | 274 | I | DATA47 |
| 15 | O | LDATA1 | 67 | I | XDATA16 | 119 | E | oe-mas | 171 | E | oemxd11 | 223 | O | ADDR28 | 275 | O | DATA47 |
| 16 | I | LDATA0 | 68 | O | XDATA16 | 120 | I | $\overline{\text{XREQ1}}$[0] | 172 | I | XDATA48 | 224 | I | ADDR29 | 276 | I | DATA46 |
| 17 | O | LDATA0 | 69 | I | XDATA17 | 121 | O | $\overline{\text{XREQ1}}$[0] | 173 | O | XDATA48 | 225 | O | ADDR29 | 277 | O | DATA46 |
| 18 | E | oe-bb-dt | 70 | O | XDATA17 | 122 | E | oe-mih | 174 | I | XDATA49 | 226 | I | ADDR30 | 278 | I | DATA45 |
| 19 | O | LCMDS | 71 | I | XDATA18 | 123 | I | $\overline{\text{XREQ1}}$[1] | 175 | O | XDATA49 | 227 | O | ADDR30 | 279 | O | DATA45 |
| 20 | O | LCMD2 | 72 | O | XDATA18 | 124 | O | $\overline{\text{XREQ1}}$[1] | 176 | I | XDATA50 | 228 | E | oeaddr3 | 280 | I | DATA44 |
| 21 | I | LCMD1 | 73 | I | XDATA19 | 125 | E | oe-msh | 177 | O | XDATA50 | 229 | I | ADDR31 | 281 | O | DATA44 |
| 22 | O | LCMD1 | 74 | O | XDATA19 | 126 | I | $\overline{\text{XREQ2}}$[0] | 178 | I | XDATA51 | 230 | O | ADDR31 | 282 | I | DATA43 |
| 23 | I | LCMD0 | 75 | I | XDATA20 | 127 | I | $\overline{\text{XREQ2}}$[1] | 179 | O | XDATA51 | 231 | I | ADDR32 | 283 | O | DATA43 |
| 24 | O | LCMD0 | 76 | O | XDATA20 | 128 | I | $\overline{\text{XREQ3}}$[0] | 180 | I | XDATA52 | 232 | O | ADDR32 | 284 | I | DATA42 |
| 25 | O | $\overline{\text{CCERR}}$ | 77 | I | XDATA21 | 129 | I | $\overline{\text{XREQ3}}$[1] | 181 | O | XDATA52 | 233 | I | ADDR33 | 285 | O | DATA42 |
| 26 | E | oe-aerr | 78 | O | XDATA21 | 130 | O | $\overline{\text{XGNT0}}$ | 182 | I | XDATA53 | 234 | O | ADDR33 | 286 | I | DATA41 |
| 27 | I | spare-in | 79 | I | XDATA22 | 131 | I | $\overline{\text{XGNT1}}$ | 183 | O | XDATA53 | 235 | I | ADDR34 | 287 | O | DATA41 |
| 28 | O | spare-out | 80 | O | XDATA22 | 132 | O | $\overline{\text{XGNT1}}$ | 184 | I | XDATA54 | 236 | O | ADDR34 | 288 | I | DATA40 |
| 29 | I | $\overline{\text{RSTIN}}$ | 81 | I | XDATA23 | 133 | I | $\overline{\text{XGNT2}}$ | 185 | O | XDATA54 | 237 | I | ADDR35 | 289 | O | DATA40 |
| 30 | I | XDATA00 | 82 | O | XDATA23 | 134 | O | $\overline{\text{XGNT2}}$ | 186 | I | XDATA55 | 238 | O | ADDR35 | 290 | E | oe-data5 |
| 31 | O | XDATA00 | 83 | E | oe-mxd4 | 135 | O | $\overline{\text{XGNT3}}$ | 187 | O | XDATA55 | 239 | I | PCLK | 291 | E | oe-data4 |
| 32 | I | XDATA01 | 84 | E | oe-mxd5 | 136 | I | XDATA32 | 188 | E | oemxd12 | 240 | I | DATA63 | 292 | I | DATA39 |
| 33 | O | XDATA01 | 85 | I | XDATA24 | 137 | O | XDATA32 | 189 | E | oemxd13 | 241 | O | DATA63 | 293 | O | DATA39 |
| 34 | I | XDATA02 | 86 | O | XDATA24 | 138 | I | XDATA33 | 190 | I | XDATA56 | 242 | I | DATA62 | 294 | I | DATA38 |
| 35 | O | XDATA02 | 87 | I | XDATA25 | 139 | O | XDATA33 | 191 | O | XDATA56 | 243 | O | DATA62 | 295 | O | DATA38 |
| 36 | I | XDATA03 | 88 | O | XDATA25 | 140 | I | XDATA34 | 192 | I | XDATA57 | 244 | I | DATA61 | 296 | I | DATA37 |
| 37 | O | XDATA03 | 89 | I | XDATA26 | 141 | O | XDATA34 | 193 | O | XDATA57 | 245 | O | DATA61 | 297 | O | DATA37 |
| 38 | I | XDATA04 | 90 | O | XDATA26 | 142 | I | XDATA35 | 194 | I | XDATA58 | 246 | I | DATA60 | 298 | I | DATA36 |
| 39 | O | XDATA04 | 91 | I | XDATA27 | 143 | O | XDATA35 | 195 | O | XDATA58 | 247 | O | DATA60 | 299 | O | DATA36 |
| 40 | I | XDATA05 | 92 | O | XDATA27 | 144 | I | XDATA36 | 196 | I | XDATA59 | 248 | I | DATA59 | 300 | I | DATA35 |
| 41 | O | XDATA05 | 93 | I | XDATA28 | 145 | O | XDATA36 | 197 | O | XDATA59 | 249 | O | DATA59 | 301 | O | DATA35 |
| 42 | I | XDATA06 | 94 | O | XDATA28 | 146 | I | XDATA37 | 198 | I | XDATA60 | 250 | I | DATA58 | 302 | I | DATA34 |
| 43 | O | XDATA06 | 95 | I | XDATA29 | 147 | O | XDATA37 | 199 | O | XDATA60 | 251 | O | DATA58 | 303 | O | DATA34 |
| 44 | I | XDATA07 | 96 | O | XDATA29 | 148 | I | XDATA38 | 200 | I | XDATA61 | 252 | I | DATA57 | 304 | I | DATA33 |
| 45 | O | XDATA07 | 97 | I | XDATA30 | 149 | O | XDATA38 | 201 | O | XDATA61 | 253 | O | DATA57 | 305 | O | DATA33 |
| 46 | E | oe-mxd0 | 98 | O | XDATA30 | 150 | I | XDATA39 | 202 | I | XDATA62 | 254 | I | DATA56 | 306 | I | DATA32 |
| 47 | E | oe-mxd1 | 99 | I | XDATA31 | 151 | O | XDATA39 | 203 | O | XDATA62 | 255 | O | DATA56 | 307 | O | DATA32 |
| 48 | I | XDATA08 | 100 | O | XDATA31 | 152 | E | oe-mxd8 | 204 | I | XDATA63 | 256 | E | oe-data7 | 308 | O | $\overline{\text{WE}}$3 |
| 49 | O | XDATA08 | 101 | E | oe-mxd6 | 153 | E | oe-mxd9 | 205 | O | XDATA63 | 257 | E | oe-data6 | 309 | O | $\overline{\text{WE}}$2 |
| 50 | I | XDATA09 | 102 | E | oe-mxd7 | 154 | I | XDATA40 | 206 | E | oemxd14 | 258 | I | DATA55 | 310 | O | $\overline{\text{WE}}$1 |
| 51 | O | XDATA09 | 103 | I | XPAR0 | 155 | O | XDATA40 | 207 | E | oemxd15 | 259 | O | DATA55 | 311 | O | $\overline{\text{WE}}$0 |
| 52 | I | GTLREF1 | 104 | O | XPAR0 | 156 | I | XDATA41 | 208 | I | BCLK | 260 | I | DATA54 | 312 | E | oe-we0 |
| 313 | E | oe-dpar1 | 343 | O | ADDR18 | 373 | E | oe-addr0 | 403 | I | $\overline{\text{CSA}}$ | 433 | O | DATA06 | 462 | O | DATA20 |
| 314 | I | DPAR3 | 344 | I | ADDR17 | 374 | I | ADDR03 | 404 | I | $\overline{\text{LDST}}$ | 434 | I | DATA07 | 464 | I | DATA21 |
| 315 | O | DPAR3 | 345 | O | ADDR17 | 375 | O | ADDR03 | 405 | I | $\overline{\text{CCHBL}}$ | 435 | O | DATA07 | 465 | O | DATA21 |
| 316 | I | DPAR2 | 346 | I | ADDR16 | 376 | I | ADDR02 | 406 | E | oe-dpar0 | 436 | E | oe-data0 | 466 | I | DATA22 |

## TABLE 11: Boundary Scan Bit Order in XBus Mode (Continued)

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 317 | O | DPAR2 | 347 | O | ADDR16 | 377 | O | ADDR02 | 407 | I | DPAR4 | 437 | E | oe-data1 | 467 | O | DATA22 |
| 318 | I | DPAR1 | 348 | I | ADDR15 | 378 | I | ADDR01 | 408 | O | DPAR4 | 438 | I | DATA08 | 468 | I | DATA23 |
| 319 | O | DPAR1 | 349 | O | ADDR15 | 379 | O | ADDR01 | 409 | I | DPAR5 | 439 | O | DATA08 | 469 | O | DATA23 |
| 320 | I | DPAR0 | 350 | I | ADDR14 | 380 | I | ADDR00 | 410 | O | DPAR5 | 440 | I | DATA09 | 470 | E | oe-data2 |
| 321 | O | DPAR0 | 351 | O | ADDR14 | 381 | O | ADDR00 | 411 | I | DPAR6 | 441 | O | DATA09 | 471 | E | oe-data3 |
| 322 | O | $\overline{\text{RESET}}$ | 352 | I | ADDR13 | 382 | I | $\overline{\text{OE}}$ | 412 | O | DPAR6 | 442 | I | DATA10 | 472 | I | DATA24 |
| 323 | O | $\overline{\text{WEE}}$ | 353 | O | ADDR13 | 383 | O | $\overline{\text{OE}}$ | 413 | I | DPAR7 | 443 | O | DATA10 | 473 | O | DATA24 |
| 324 | I | SIZE1 | 354 | I | ADDR12 | 384 | E | oe-oe | 414 | O | DPAR7 | 444 | I | DATA11 | 474 | I | DATA25 |
| 325 | I | SIZE0 | 355 | O | ADDR12 | 385 | I | $\overline{\text{WR}}$ | 415 | O | $\overline{\text{WE4}}$ | 445 | O | DATA11 | 475 | O | DATA25 |
| 326 | I | $\overline{\text{ERROR}}$ | 356 | E | oe-addr1 | 386 | O | $\overline{\text{WR}}$ | 416 | O | $\overline{\text{WE5}}$ | 446 | I | DATA12 | 476 | I | DATA26 |
| 327 | I | $\overline{\text{SU}}$ | 357 | I | ADDR11 | 387 | E | oe-wr | 417 | O | $\overline{\text{WE6}}$ | 447 | O | DATA12 | 477 | O | DATA26 |
| 328 | I | SYNC | 358 | O | ADDR11 | 388 | I | $\overline{\text{RD}}$ | 418 | O | $\overline{\text{WE7}}$ | 448 | I | DATA13 | 478 | I | DATA27 |
| 329 | I | $\overline{\text{ADDR20}}$ | 359 | I | ADDR10 | 389 | I | BURST | 419 | E | oe-we1 | 449 | O | DATA13 | 479 | O | DATA27 |
| 330 | O | $\overline{\text{ADDR20}}$ | 360 | O | ADDR10 | 390 | O | RETRY | 420 | I | DATA00 | 450 | I | DATA14 | 480 | I | DATA28 |
| 331 | I | ADDR23 | 361 | I | ADDR09 | 391 | O | PEND | 421 | O | DATA00 | 451 | O | DATA14 | 481 | O | DATA28 |
| 332 | O | ADDR23 | 362 | O | ADDR09 | 392 | O | MEXC | 422 | I | DATA01 | 452 | I | DATA15 | 482 | I | DATA29 |
| 333 | I | ADDR22 | 363 | I | ADDR08 | 393 | O | $\overline{\text{WRDY}}$ | 423 | O | DATA01 | 453 | O | DATA15 | 483 | O | DATA29 |
| 334 | O | ADDR22 | 364 | O | ADDR08 | 394 | O | $\overline{\text{RRDY}}$ | 424 | I | DATA02 | 454 | I | DATA16 | 484 | I | DATA30 |
| 335 | I | ADDR21 | 365 | I | ADDR07 | 395 | O | $\overline{\text{WGRT}}$ | 425 | O | DATA02 | 455 | O | DATA16 | 485 | O | DATA30 |
| 336 | O | ADDR21 | 366 | O | ADDR07 | 396 | O | $\overline{\text{RGRT}}$ | 426 | I | DATA03 | 456 | I | DATA17 | 486 | I | DATA31 |
| 337 | I | ADDR20 | 367 | I | ADDR06 | 397 | I | $\overline{\text{CMDS}}$ | 427 | O | DATA03 | 457 | O | DATA17 | 487 | O | DATA31 |
| 338 | O | ADDR20 | 368 | O | ADDR06 | 398 | O | $\overline{\text{CMDS}}$ | 428 | I | DATA04 | 458 | I | DATA18 | | | |
| 339 | E | oe-addr2 | 369 | I | ADDR05 | 399 | E | oe-cmds | 429 | O | DATA04 | 459 | O | DATA18 | | | |
| 340 | I | ADDR19 | 370 | O | ADDR05 | 400 | I | $\overline{\text{DEMAP}}$ | 430 | I | DATA05 | 460 | I | DATA19 | | | |
| 341 | O | ADDR19 | 371 | I | ADDR04 | 401 | O | $\overline{\text{DEMAP}}$ | 431 | O | DATA05 | 461 | O | DATA19 | | | |
| 342 | I | ADDR18 | 372 | O | ADDR04 | 402 | E | oe-dmap | 432 | I | DATA06 | | I | DATA20 | | | |

# PARAMETER MEASUREMENT

## TTL Parameters

### Load Circuit Parameters

| | | | | | |
|---|---|---|---|---|---|
| $t_{en}$ | $t_{PZH}$ | 35 | 2.0 | -370 | 2.25 |
| | $t_{PZL}$ | | | | |
| $t_{dis}$ | $t_{PHZ}$ | 35 | 2.0 | -370 | 2.25 |
| | $t_{PLZ}$ | | | | |
| $t_{PD}$ | | 35 | 2.2 | -2.0 | 2.25 |
| $t_{PD(MSH)}$ | | 35 | 8.0 | -2.0 | 2.25 |

1. $C_{LOAD}$ includes probes and test fixture capacitance.

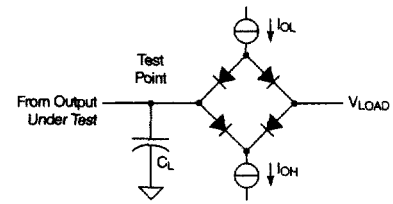
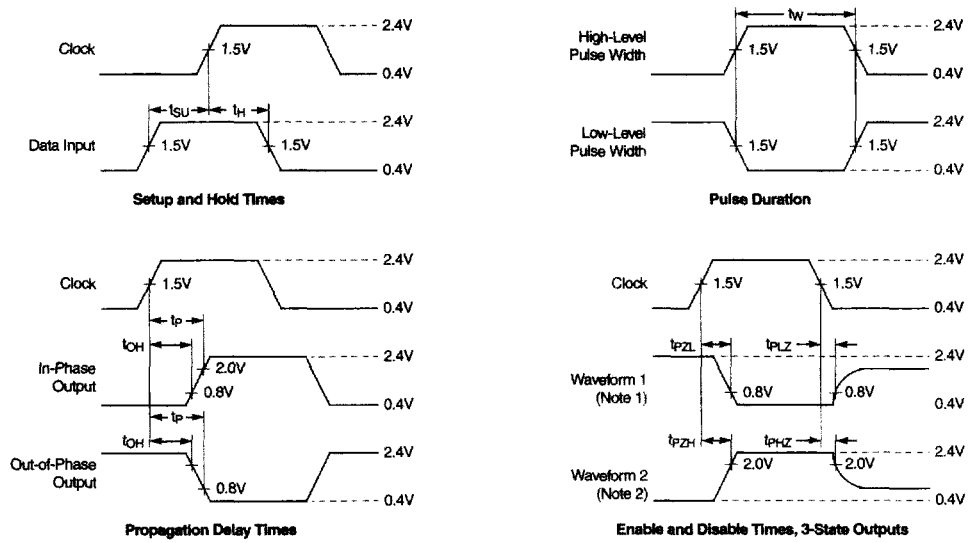



**Figure 28. TTL Load Circuit and Parameters**




1. Waveform 1 is for an output with internal conditions such that the output is low except when disabled by the output control.
2. Waveform 2 is for an output with internal conditions such that the output is high except when disabled by the output control. For $t_{PLZ}$ and $t_{PHZ}$, $V_{OL}$ and $V_{OH}$ are specified values.

**Figure 29. TTL Voltage Waveforms**

## GTL Parameters

**Load Circuit Parameters**

| | | | | | |
|---|---|---|---|---|---|
| $t_{en}$ | $t_{PZH}$ | 35 | 36 | 10 | 1.2 |
| | $t_{PZL}$ | | | | |
| $t_{dis}$ | $t_{PHZ}$ | 35 | 36 | 10 | 1.2 |
| | $t_{PLZ}$ | | | | |
| $t_{PD}$ | | – | 36 | 10 | 1.2 |

1. $C_{LOAD}$ includes probes and test fixture capacitance.

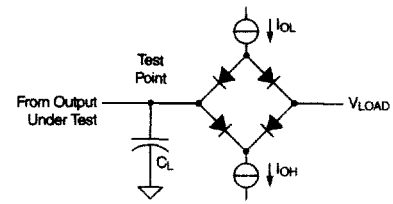
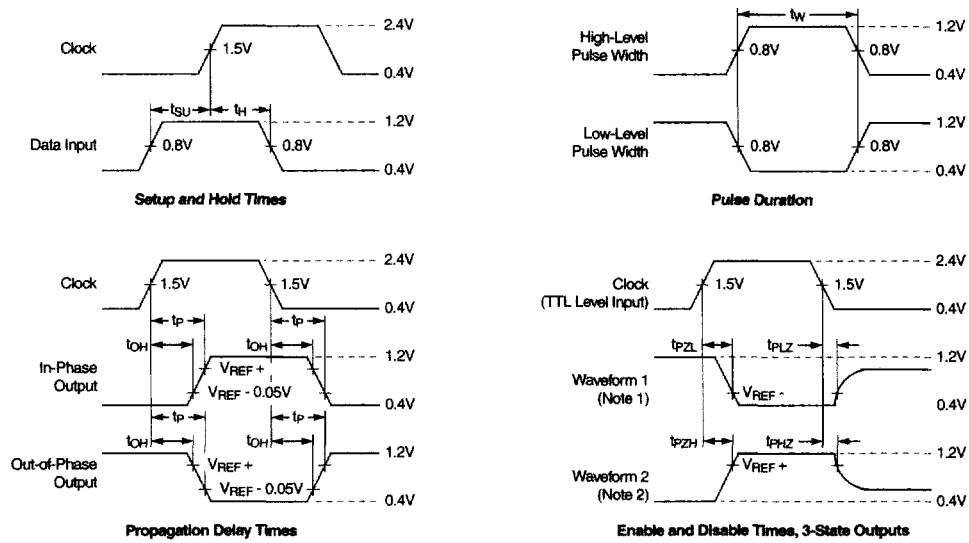



**Figure 30. GTL Load Circuit and Parameters**




1. Waveform 1 is for an output with internal conditions such that the output is low except when disabled by the output control.
2. Waveform 2 is for an output with internal conditions such that the output is high except when disabled by the output control. For $t_{PLZ}$ and $t_{PHZ}$, $V_{OL}$ and $V_{OH}$ are specified values.

**Figure 31. GTL Voltage Waveforms**

# PIN ASSIGNMENTS

## *MBus Pinouts*

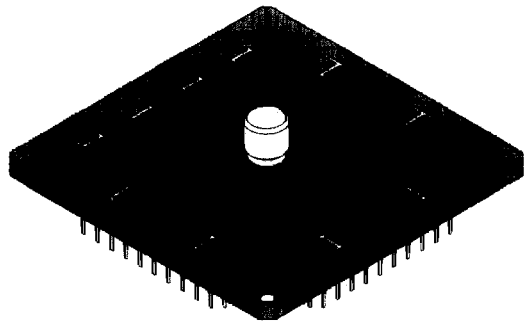
| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A9 | RETRY | E5 | WE7 | H6 | VCCI | N1 | DATA18 | V34 | ADDR28 | AD32 | ADDR29 | AJ11 | MAD19 | AM12 | MAD28 |
| A11 | VCCP | E7 | WE5 | H8 | WE | N3 | DATA24 | W1 | LDATA1 | AD34 | VCCPX | AJ13 | MAD24 | AM14 | VSSC |
| A13 | OE | E9 | DPAR7 | H10 | VSSP | N5 | DATA14 | W3 | DATA29 | AE1 | VSSPX | AJ15 | MAD30 | AM16 | MAS |
| A15 | VSSP | E11 | DEMAP | H12 | CCHBL | N7 | DATA11 | W5 | DATA31 | AE3 | SPARE | AJ17 | MIH | AM18 | VSSPX |
| A17 | ADDR05 | E13 | WRDY | H14 | VSSC | N29 | DATA49 | W7 | MX | AE5 | TCK | AJ19 | N.C. | AM20 | MID0 |
| A19 | ADDR09 | E15 | WR | H16 | ADDR00 | N31 | DATA47 | W29 | VSSCKP | AE7 | MAD01 | AJ21 | GTLREF1 | AM22 | VSSC |
| A21 | VSSP | E17 | ADDR03 | H18 | VCCP | N33 | DATA48 | W31 | VCCI | AE29 | VCCCKB | AJ23 | MAD42 | AM24 | MAD40 |
| A23 | ADDR15 | E19 | ADDR11 | H20 | ADDR18 | N35 | DATA | W33 | ADDR27 | AE31 | IRL0 | AJ25 | MAD47 | AM26 | VCCC |
| A25 | VCCP | E21 | ADDR16 | H22 | VSSC | P2 | DATA22 | W35 | VCCP | AE33 | ADDR26 | AJ27 | MAD52 | AM28 | MAD51 |
| A27 | SYNC | E23 | ADDR22 | H24 | DPAR01 | P4 | VSSC | Y2 | VSSPX | AE35 | ADDR30 | AJ29 | MAD56 | AM30 | VSSPX |
| B8 | VSSC | E25 | SIZE1 | H26 | VSSP | P6 | DATA16 | Y4 | MIRL0 | AF2 | TMS | AJ31 | MAD60 | AM32 | IRL1 |
| B10 | CMDS | E27 | DPAR2 | H28 | DATA33 | P8 | VSSP | Y6 | VCCC | AF4 | VCCC | AJ33 | MAD61 | AN5 | TDIODE1 |
| B12 | VSSP | E29 | WE0 | H30 | VCCI | P28 | VSSP | Y8 | N.C. | AF6 | MAD02 | AK4 | VSSC | AN7 | MAD14 |
| B14 | RD | E31 | DATA32 | H32 | DATA36 | P30 | DATA50 | Y28 | VCCCKP | AF8 | VSSC | AK6 | MAD06 | AN9 | MAD21 |
| B16 | VCCP | F4 | VSSC | H34 | VSSP | P32 | VSSC | Y30 | VCCP | AF28 | MAD58 | AK8 | VSSI | AN11 | MAD26 |
| B18 | ADDR08 | F6 | VCCP | J1 | DATA08 | P34 | DATA52 | Y32 | PLLBYP | AF30 | MAD59 | AK10 | MAD20 | AN13 | MERR |
| B20 | VCCP | F8 | VSSI | J3 | DATA15 | R1 | VCCP | Y34 | VCCC | AF32 | VCCC | AK12 | VCCPX | AN15 | MBB |
| B22 | ADDR17 | F10 | DPAR06 | J5 | DATA03 | R3 | DATA30 | AA1 | VCCPX | AF34 | ADDR25 | AK14 | MAD29 | AN17 | N.C. |
| B24 | VSSP | F12 | VSSP | J7 | DATA01 | R5 | DATA20 | AA3 | MIRL2 | AG1 | TDI | AK16 | VCCC | AN19 | MID3 |
| B26 | ADDR20 | F14 | MEXC | J29 | DATA35 | R7 | DATA17 | AA5 | LDATA2 | AG3 | RSTIN | AK18 | N.C. | AN21 | N.C. |
| B28 | VSSC | F16 | VCCC | J31 | DATA37 | R29 | DATA54 | AA7 | LCMD2 | AG5 | MAD00 | AK20 | VCCC | AN23 | MAD36 |
| C7 | DPAR4 | F18 | ADDR07 | J33 | DATA38 | R31 | DATA53 | AA29 | BPLLRC | AG7 | MAD05 | AK22 | MAD37 | AN25 | MAD41 |
| C9 | CSA | F20 | VCCC | J35 | DATA42 | R33 | DATA56 | AA31 | ADDR34 | AG29 | VSSC | AK24 | VCCPX | AN27 | MAD45 |
| C11 | WGRT | F22 | ADDR21 | K2 | DATA12 | R35 | VCCP | AA33 | DATA63 | AG31 | MAD63 | AK26 | MAD49 | AN29 | MAD53 |
| C13 | PEND | F24 | VSSP | K4 | VCCC | T2 | VSSP | AA35 | DATA61 | AG33 | IRL3 | AK28 | VSSI | AP8 | PMC2 |
| C15 | ADDR01 | F26 | RESET | K6 | DATA7 | T4 | DATA23 | AB2 | N.C. | AG35 | ADDR24 | AK30 | MAD55 | AP10 | MAD25 |
| C17 | ADDR06 | F28 | VSSI | K8 | VSSC | T6 | VCCC | AB4 | VSSC | AH2 | VSSPX | AK32 | VSSC | AP12 | VSSPX |
| C19 | ADDR10 | F30 | VCCP | K28 | VSSC | T8 | DATA21 | AB6 | MID1 | AH4 | MAD03 | AL3 | TEST | AP14 | MAD23 |
| C21 | ADDR14 | F32 | VSSC | K30 | DATA39 | T28 | DATA55 | AB8 | VSSPX | AH6 | VCCI | AL5 | MAD10 | AP16 | VCCPX |
| C23 | ADDR19 | G3 | DATA02 | K32 | VCCC | T30 | VCCP | AB28 | VCCC | AH8 | MAD09 | AL7 | MAD12 | AP18 | PMC3 |
| C25 | SU | G5 | DATA05 | K34 | DATA43 | T32 | DATA57 | AB30 | ADDR31 | AH10 | VSSPX | AL9 | MAD17 | AP20 | VCCPX |
| C27 | ERROR | G7 | DATA00 | L1 | VSSP | T34 | VSSP | AB32 | VSSC | AH12 | MAD18 | AL11 | MAD11 | AP22 | MAD34 |
| C29 | DPAR3 | G9 | WE4 | L3 | DATA19 | U1 | MIRL1 | AB34 | ADDR35 | AH14 | VSSC | AL13 | MAD13 | AP24 | VSSPX |
| D4 | N.C. | G11 | DPAR5 | L5 | DATA10 | U3 | DATA25 | AC1 | AERR | AH16 | MRDY | AL15 | MRTY | AP26 | MAD44 |
| D6 | VSSP | G13 | RGRT | L7 | DATA04 | U5 | DATA27 | AC3 | N.C. | AH18 | VCCPX | AL17 | PMC0 | AP28 | VSSC |
| D8 | LDST | G15 | BURST | L29 | DATA40 | U7 | DATA26 | AC5 | MID2 | AH20 | MAD32 | AL19 | MBR | AP30 | PMC1 |
| D10 | VCCC | G17 | ADDR04 | L31 | DATA44 | U29 | PCLK | AC7 | TRST | AH22 | VSSC | AL21 | MAD33 | AR9 | MAD22 |
| D12 | RRDY | G19 | ADDR13 | L33 | DATA45 | U31 | DATA60 | AC29 | MCLK | AH24 | MAD46 | AL23 | MAD38 | AR11 | VCCPX |
| D14 | VSSC | G21 | ADDR20 | L35 | VSSP | U33 | DATA59 | AC31 | ADDR32 | AH26 | VSSPX | AL25 | MAD39 | AR13 | MAD15 |
| D16 | ADDR02 | G23 | SIZE0 | M2 | VCCP | U35 | DATA58 | AC33 | ADDR33 | AH28 | MAD54 | AL27 | MAD50 | AR15 | MAD27 |
| D18 | VSSP | G25 | DPAR0 | M4 | DATA13 | V2 | MIRL3 | AC35 | VSSPX | AH30 | VCCI | AL29 | MAD48 | AR17 | MBG |
| D20 | ADDR12 | G27 | WE1 | M6 | VSSP | V4 | VSSI | AD2 | VCCPX | AH32 | MAD62 | AL31 | MAD57 | AR19 | MSH |
| D22 | VSSC | G29 | WE2 | M8 | DATA06 | V6 | DATA28 | AD4 | TDO | AH34 | VSSPX | AL33 | IRL2 | AR21 | N.C. |
| D24 | ADDR23 | G31 | WE3 | M28 | DATA41 | V8 | VCCI | AD6 | VSSPX | AJ3 | MAD07 | AM4 | TDIODE0 | AR23 | MAD35 |
| D26 | VCCC | G33 | DATA34 | M30 | VSSP | V28 | PPLLRC | AD8 | GTLREF | AJ5 | MAD04 | AM6 | VSSPX | AR25 | VCCPX |
| D28 | WEE | H2 | VSSP | M32 | DATA46 | V30 | DATA62 | AD28 | VSSCKB | AJ7 | MAD08 | AM8 | MAD16 | AR27 | MAD43 |
| D30 | VSSP | H4 | DATA09 | M34 | VCCP | V32 | VSSI | AD30 | VSSPX | AJ9 | MAD13 | AM10 | VCCC | | |

## XBus Pinouts

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A9 | RETRY | E5 | WE7 | H6 | VCCI | N1 | DATA18 | V34 | ADDR28 | AD32 | ADDR29 | AJ11 | XD19 | AM12 | XD28 |
| A11 | VCCP | E7 | WE5 | H8 | WE | N3 | DATA24 | W1 | LDATA1 | AD34 | VCCPX | AJ13 | XD24 | AM14 | VSSC |
| A13 | OE | E9 | DPAR7 | H10 | VSSP | N5 | DATA14 | W3 | DATA29 | AE1 | VSSPX | AJ15 | XD30 | AM16 | XREQ01 |
| A15 | VSSP | E11 | DEMAP | H12 | CCHBL | N7 | DATA11 | W5 | DATA31 | AE3 | SPARE | AJ17 | XREQ10 | AM18 | VSSPX |
| A17 | ADDR05 | E13 | WRDY | H14 | VSSC | N29 | DATA49 | W7 | MX | AE5 | TCK | AJ19 | XREQ30 | AM20 | XGNT2 |
| A19 | ADDR09 | E15 | WR | H16 | ADDR00 | N31 | DATA47 | W29 | VSSCKP | AE7 | XD01 | AJ21 | GTLREF1 | AM22 | VSSC |
| A21 | VSSP | E17 | ADDR03 | H18 | VCCP | N33 | DATA48 | W31 | VCCI | AE29 | VCCCKB | AJ23 | XD42 | AM24 | XD40 |
| A23 | ADDR15 | E19 | ADDR11 | H20 | ADDR18 | N35 | DATA | W33 | ADDR27 | AE31 | IRL0 | AJ25 | XD47 | AM26 | VCCC |
| A25 | VCCP | E21 | ADDR16 | H22 | VSSC | P2 | DATA22 | W35 | VCCP | AE33 | ADDR26 | AJ27 | XD52 | AM28 | XD51 |
| A27 | SYNC | E23 | ADDR22 | H24 | DPAR01 | P4 | VSSC | Y2 | VSSPX | AE35 | ADDR30 | AJ29 | XD56 | AM30 | VSSPX |
| B8 | VSSC | E25 | SIZE1 | H26 | VSSP | P6 | DATA16 | Y4 | LDATA6 | AF2 | TMS | AJ31 | XD60 | AM32 | IRL1 |
| B10 | CMDS | E27 | DPAR2 | H28 | DATA33 | P8 | VSSP | Y6 | VCCC | AF4 | VCCC | AJ33 | XD61 | AN5 | TDIODE1 |
| B12 | VSSP | E29 | WE0 | H30 | VCCI | P28 | VSSP | Y8 | LDATA3 | AF6 | XD02 | AK4 | VSSC | AN7 | XD14 |
| B14 | RD | E31 | DATA32 | H32 | DATA36 | P30 | DATA50 | Y28 | VCCCKP | AF8 | VSSC | AK6 | XD06 | AN9 | XD21 |
| B16 | VCCP | F4 | VSSC | H34 | VSSP | P32 | VSSC | Y30 | VCCP | AF28 | XD58 | AK8 | VSSI | AN11 | XD26 |
| B18 | ADDR08 | F6 | VCCP | J1 | DATA08 | P34 | DATA52 | Y32 | PLLBYP | AF30 | XD59 | AK10 | XD20 | AN13 | XPAR0 |
| B20 | VCCP | F8 | VSSI | J3 | DATA15 | R1 | VCCP | Y34 | VCCC | AF32 | VCCC | AK12 | VCCPX | AN15 | XREQ00 |
| B22 | ADDR17 | F10 | DPAR06 | J5 | DATA03 | R3 | DATA30 | AA1 | VCCPX | AF34 | ADDR25 | AK14 | XD29 | AN17 | XREQ20 |
| B24 | VSSP | F12 | VSSP | J7 | DATA01 | R5 | DATA20 | AA3 | LDATA4 | AG1 | TDI | AK16 | VCCC | AN19 | XGNT1 |
| B26 | ADDR20 | F14 | MEXC | J29 | DATA35 | R7 | DATA17 | AA5 | LDATA2 | AG3 | RSTIN | AK18 | XREQ21 | AN21 | XGNT3 |
| B28 | VSSC | F16 | VCCC | J31 | DATA37 | R29 | DATA54 | AA7 | LCMD2 | AG5 | XD00 | AK20 | VCCC | AN23 | XD36 |
| C7 | DPAR4 | F18 | ADDR07 | J33 | DATA38 | R31 | DATA53 | AA29 | BPLLRC | AG7 | XD05 | AK22 | XD37 | AN25 | XD41 |
| C9 | CSA | F20 | VCCC | J35 | DATA42 | R33 | DATA56 | AA31 | ADDR34 | AG29 | VSSC | AK24 | VCCPX | AN27 | XD45 |
| C11 | WGRT | F22 | ADDR21 | K2 | DATA12 | R35 | VCCP | AA33 | DATA63 | AG31 | XD63 | AK26 | XD49 | AN29 | XD53 |
| C13 | PEND | F24 | VSSP | K4 | VCCC | T2 | VSSP | AA35 | DATA61 | AG33 | IRL3 | AK28 | VSSI | AP8 | PMC2 |
| C15 | ADDR01 | F26 | RESET | K6 | DATA7 | T4 | DATA23 | AB2 | LCMDS | AG35 | ADDR24 | AK30 | XD55 | AP10 | XD25 |
| C17 | ADDR06 | F28 | VSSI | K8 | VSSC | T6 | VCCC | AB4 | VSSC | AH2 | VSSPX | AK32 | VSSC | AP12 | VSSPX |
| C19 | ADDR10 | F30 | VCCP | K28 | VSSC | T8 | DATA21 | AB6 | LCMD1 | AH4 | XD03 | AL3 | TEST | AP14 | XD23 |
| C21 | ADDR14 | F32 | VSSC | K30 | DATA39 | T28 | DATA55 | AB8 | VSSPX | AH6 | VCCI | AL5 | XD10 | AP16 | VCCPX |
| C23 | ADDR19 | G3 | DATA02 | K32 | VCCC | T30 | VCCP | AB28 | VCCC | AH8 | XD09 | AL7 | XD12 | AP18 | PMC3 |
| C25 | SU | G5 | DATA05 | K34 | DATA43 | T32 | DATA57 | AB30 | ADDR31 | AH10 | VSSPX | AL9 | XD17 | AP20 | VCCPX |
| C27 | ERROR | G7 | DATA00 | L1 | VSSP | T34 | VSSP | AB32 | VSSC | AH12 | XD18 | AL11 | XD11 | AP22 | XD34 |
| C29 | DPAR3 | G9 | WE4 | L3 | DATA19 | U1 | LDATA7 | AB34 | ADDR35 | AH14 | VSSC | AL13 | XD13 | AP24 | VSSPX |
| D4 | N.C. | G11 | DPAR5 | L5 | DATA10 | U3 | DATA25 | AC1 | CCERR | AH16 | XPAR1 | AL15 | XPAR2 | AP26 | XD44 |
| D6 | VSSP | G13 | RGRT | L7 | DATA04 | U5 | DATA27 | AC3 | LDATA0 | AH18 | VCCPX | AL17 | PMC0 | AP28 | VSSC |
| D8 | LDST | G15 | BURST | L29 | DATA40 | U7 | DATA26 | AC5 | LCMD0 | AH20 | XD32 | AL19 | XGNT0 | AP30 | PMC1 |
| D10 | VCCC | G17 | ADDR04 | L31 | DATA44 | U29 | PCLK | AC7 | TRST | AH22 | VSSC | AL21 | XD33 | AR9 | XD22 |
| D12 | RRDY | G19 | ADDR13 | L33 | DATA45 | U31 | DATA60 | AC29 | BCLK | AH24 | XD46 | AL23 | XD38 | AR11 | VCCPX |
| D14 | VSSC | G21 | ADDR20 | L35 | VSSP | U33 | DATA59 | AC31 | ADDR32 | AH26 | VSSPX | AL25 | XD39 | AR13 | XD15 |
| D16 | ADDR02 | G23 | SIZE0 | M2 | VCCP | U35 | DATA58 | AC33 | ADDR33 | AH28 | XD54 | AL27 | XD50 | AR15 | XD27 |
| D18 | VSSP | G25 | DPAR0 | M4 | DATA13 | V2 | LDATA5 | AC35 | VSSPX | AH30 | VCCI | AL29 | XD48 | AR17 | XPAR3 |
| D20 | ADDR12 | G27 | WE1 | M6 | VSSP | V4 | VSSI | AD2 | VCCPX | AH32 | XD62 | AL31 | XD57 | AR19 | XREQ11 |
| D22 | VSSC | G29 | WE2 | M8 | DATA06 | V6 | DATA28 | AD4 | TDO | AH34 | VSSPX | AL33 | IRL2 | AR21 | XREQ31 |
| D24 | ADDR23 | G31 | WE3 | M28 | DATA41 | V8 | VCCI | AD6 | VSSPX | AJ3 | XD07 | AM4 | TDIODE0 | AR23 | XD35 |
| D26 | VCCC | G33 | DATA34 | M30 | VSSP | V28 | PPLLRC | AD8 | GTLREF | AJ5 | XD04 | AM6 | VSSPX | AR25 | VCCPX |
| D28 | WEE | H2 | VSSP | M32 | DATA46 | V30 | DATA62 | AD28 | VSSCKB | AJ7 | XD08 | AM8 | XD16 | AR27 | XD43 |
| D30 | VSSP | H4 | DATA09 | M34 | VCCP | V32 | VSSI | AD30 | VSSPX | AJ9 | XD13 | AM10 | VCCC | | |

# PACKAGE DIMENSIONS

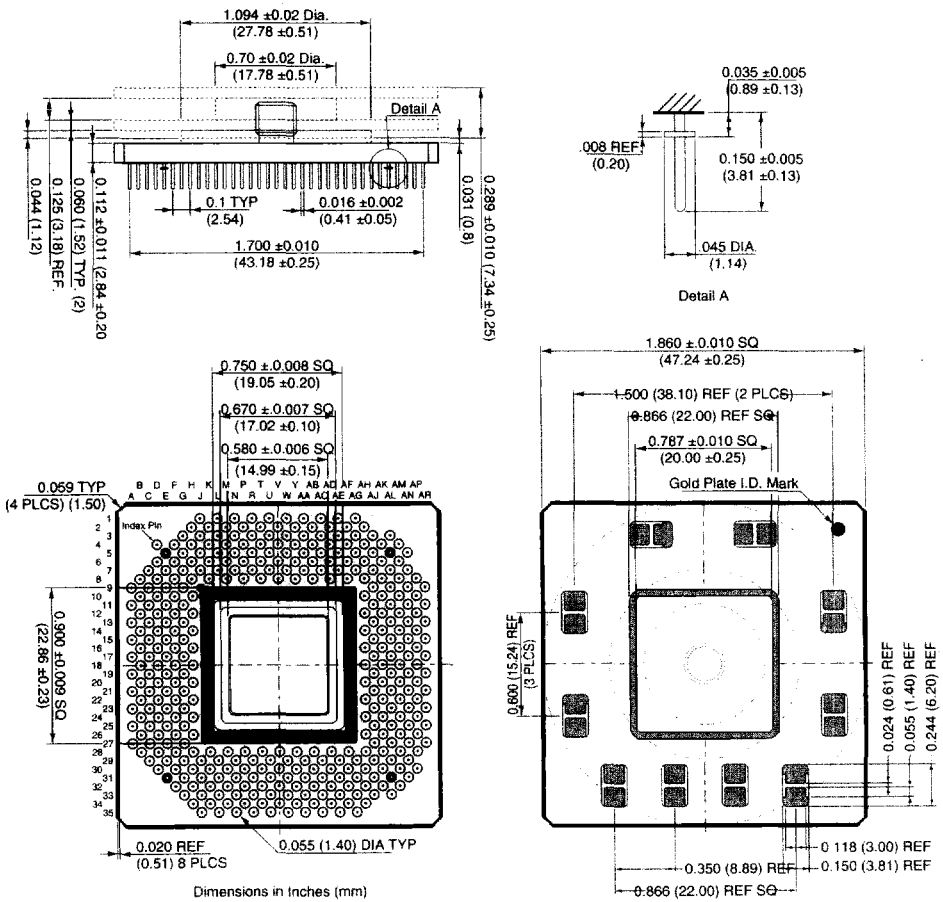
## 376-Pin PGA Package



**Thermal Resistance vs. Air Flow [1] [2]**

| | | | | |
|---|---|---|---|---|
| $\Theta_{JA}$ | 6.8 | 4.7 | 3.7 | 2.5 |

1. $T_J$ can be calculated by: $T_J = T_A + P_d \times \Theta_{JA}$
2. Thermal resistance measured using the disk-type fin supplied by Texas instruments.




Dimensions in Inches (mm)

## ORDERING INFORMATION [1]

| | | |
|---|---|---|
| STP1091PGA-75 | 75 MHz | Production Parts (for use with SuperSPARC) |
| STP1091PGA-90 | 90 MHz | Production Parts (for use with SuperSPARC) |
| STP1020HS | – | Disk-Fin Type Heat Sink. |

1. Standard parts do not have heat sinks. Heat sinks should be ordered separately.

**Document Part Number: STP1091**