Integrated Device Technology, Inc.

RISController™ CPU FOR HIGH-PERFORMANCE EMBEDDED SYSTEMS

IDT79R3001

FEATURES:

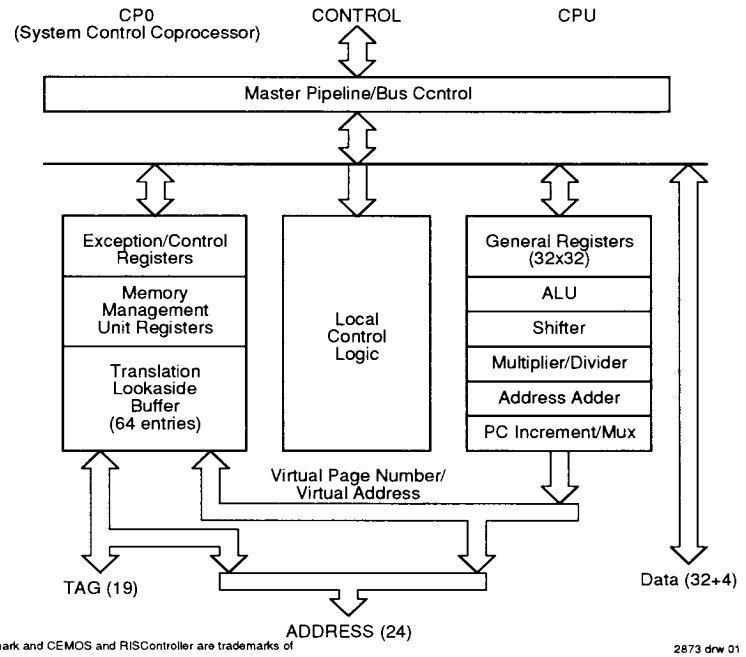
- Enhanced Instruction Set compatible version of IDT79R3000 RISC CPU
- Achieves high-performance with reduced parts count and lower overall system cost
- Flexible on-chip cache controller supports various cache, main memory sizes
- Supports optional data parity with parity error output signal
- Works with IDT79R3010A RISC Floating-Point Coprocessor
- DMA interface support
- Large synchronous memory space for real-time systems
- Full 32-bit operations — 32-bit registers, 32-bit address and data interface
- On-chip memory management unit with 64 fully-associative TLB entries maps 4GB virtual address space
- High-speed interrupt response (6 interrupt input pins) with precise exception capability
- High-speed CMOS technology results in speeds from 12.5 to 40MHz
- Supports caches from 8kB to 16MB
- Independent block refill sizes for the instruction and data caches
- Concurrent cache refill and execution
- Works on 8-, 16- and 32-bit data
- Supports unaligned 32-bit data
- Optimizing compilers for C, Ada, Pascal, Fortran, others
- RTOS support for C or Ada environments

DESCRIPTION:

The IDT79R3001 brings the high-performance inherent in the IDT79R3000 RISC Microprocessor to lower cost systems. It does this while maintaining full (both User and Kernel) software compatibility with both the IDT79R2000A and IDT79R3000 RISC Microprocessors.

The IDT79R3001 achieves lower system cost by reducing the number of components required to construct a synchronous memory (or cache) external to the processor and by simplifying the asynchronous memory interface. By removing the requirement for parity and allowing the system designer to select the cache organization which best suits the system,

FUNCTIONAL BLOCK DIAGRAM



2873 drw 01

overall parts count is dramatically reduced while maintaining high performance.

The IDT79R3001 RISC Microprocessor extends the ability of the IDT79R3000 family to support embedded and cost sensitive applications. Its level of integration and flexibility allows high-performance systems to be constructed at reasonable cost in a straightforward manner, without forcing the system designer to support features not required in his application.

The IDT79R3001 consists of two tightly coupled processors integrated on a single chip. The first processor is a full 32-bit CPU based on RISC principles to achieve a new standard of performance in microprocessor based systems. The second processor is a system control co-processor, called CP0, containing a fully associative 64-entry TLB (Translation Lookaside Buffer), MMU (Memory Management Unit), and control registers, supporting a 4GB virtual memory subsystem and a Harvard Architecture Synchronous Memory/Cache controller which achieves ultra-high bandwidth using industry standard SRAM devices.

This data sheet provides an overview of the features and architecture of the IDT79R3001 CPU. A more detailed description of the operation and timing of this device is incorporated in the *IDT79R3001 Hardware User's Guide*, and a detailed architectural overview is provided in the *MIPS RISC Architecture* book, both available from IDT. Further literature describing the hardware, software, and development tools for the IDT79R3001 is also available from IDT.

HARDWARE OVERVIEW

The IDT79R3001 is a high-performance RISC microprocessor incorporating a fast execution engine and sophisticated yet flexible memory interface designed to support the processor bandwidth requirements at minimal system cost.

Execution Engine

The IDT79R3001 contains the same basic execution engine as the ultra-high performance IDT79R3000 and thus achieves over 28 MIPS performance at 33MHz.

The key to the performance of the processor is the instruction pipeline, illustrated in Figure 2. The execution of a single IDT79R3001 instruction consists of five primary steps, some of which may be broken down further into smaller subsets.

The five primary stages of the pipeline, each of which require approximately one CPU cycle, are:

IF Instruction Fetch, when the processor fetches the instruction from the Instruction Synchronous Memory.

RD Read required operands from on-chip register file while decoding the instruction.

ALU Perform the required operation on instruction operands.

MEM Access data memory (load or store).

WB Write results back to register file.

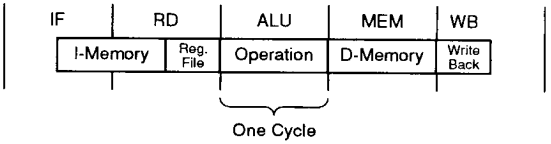


Figure 2. IDT79R3001 Five-Stage Pipeline

Thus, the CPU achieves an average execution rate approaching one instruction per CPU cycle, since the execution of five instructions at a time are overlapped within the processor (Figure 3). Optimizing compiler technology fully comprehends the interaction of software with the various pipeline resources, and serves to both eliminate any potential pipeline conflicts which might arise and to maximize instruction throughput.

The IDT79R3001 Memory Interfaces

The key to achieving the inherent performance of the IDT79R3001 is to design a memory subsystem capable of providing a new instruction to the processor on almost every clock cycle.

Like the IDT79R3000, the IDT79R3001 supports a hierarchical view of the memory subsystem. However, the IDT79R3001 allows the system designer to make more trade-offs in the partitioning and architecture of the various levels in order to more completely meet the needs of certain types of applications.

The IDT79R3001 supports two classifications of external memory: synchronous and asynchronous. The Harvard-Architecture (separate instruction and data memories) synchronous memory allows the processor to achieve the highest levels of performance. The processor is able to obtain both an instruction and data word from the synchronous memory on every clock cycle, resulting in high instruction and data throughput.

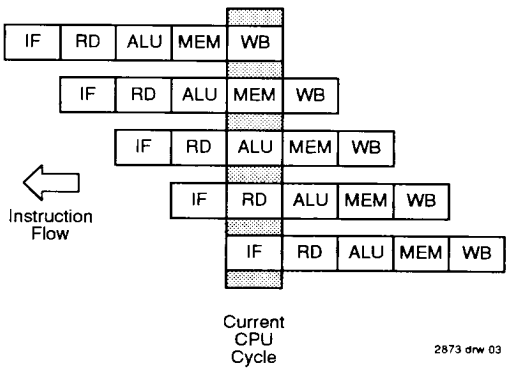


Figure 3. Instruction Execution in IDT79R3001 Pipeline

5

The asynchronous memory space contains larger, slower memory devices such as EPROM, main memory DRAMs, and peripheral devices. Multiple clock cycles are required for data movement in the asynchronous memory.

Many systems implement a memory hierarchy between these two memory spaces, whereby the synchronous memory space is used as processor caches and the asynchronous memory space is used for main memory. The IDT79R3001 integrates a flexible Direct-Mapped Cache Controller On-Chip, eliminating external cache control logic and minimizing cache management overhead. If the synchronous memory space is used for processor caches, then cache "misses" will cause the processor to automatically process an asynchronous memory transfer to refill the cache.

The key to achieving the system cost and performance goals of an IDT79R3001-based system is to partition the memory system to the needs of the application.

Synchronous Memory System

As with any high-performance processor, the IDT79R3001 requires high-bandwidth to achieve high-performance. Thus, it is important that the majority of its execution occur in the synchronous memory space. In applications which require substantial amounts of main memory, this memory space will be implemented as instruction and data caches.

The synchronous memory is designed to be able to supply both an instruction and data word to the processor on each clock cycle. When the synchronous memory spaces are used as caches, then they are used to hold instruction and data that is repetitively accessed by the CPU (for example, within a program loop). This reduces the number of slower asynchronous memory cycles and thus achieves higher performance.

Some microprocessors incorporate small amounts of cache on-chip, which has a very small and unpredictable effect on the execution of large programs. The IDT79R3001 supports

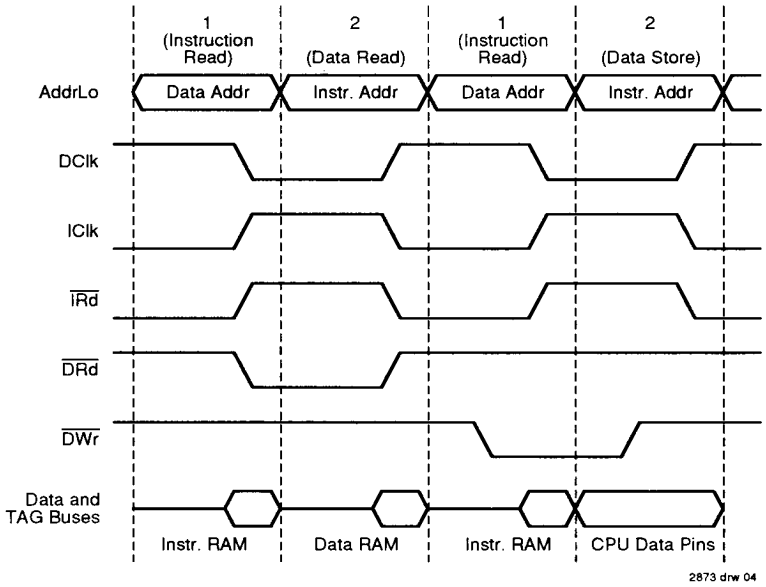


Figure 4. Synchronous Memory Control Timing

caches of from 8kB in size up through 16MB, thus bringing substantial performance improvements to very large programs and also allowing real-time system designers to design cache-based systems to support deterministic requirements.

The IDT79R3001 directly controls the synchronous memory interface (whether it is being used as caches or not) with a minimum of external components. The IDT79R3001 includes all control signals and cache TAG control logic (for a direct mapped cache) for the synchronous memory interfaces. Parity over the data portion of each synchronous memory can be optionally selected at RESET time for applications which desire to make this cost trade-off.

The synchronous interface works by dividing the basic CPU cycles into two phases. During one phase, a cache address is presented by the processor and captured by external latches (the latch control signals are directly generated by the CPU). During the next phase, the address for the other memory space is generated and captured while the data movement operation or the first cache is completed. The processor directly generates the SRAM Output Enable and Write Enable signals and the address latch enable signals, requiring no external decoding. This is illustrated in Figure 4.

Further, the IDT79R3001 supports the ability to refill multiple words into the cache from main memory when a cache-

miss occurs, further reducing system cost and increasing performance in cache-based systems. The IDT79R3001 can obtain 1, 4, 8, 16, or 32 words from main memory when processing a cache-miss, thus amortizing the cache-miss penalty over a large amount of data.

The IDT79R3001 also performs instruction streaming, which is the simultaneous execution of incoming instructions while the cache is being refilled.

The actual width of the tag bus, and whether or not parity over the data parts of each synchronous memory is included, is determined according to how the device is initialized. The IDT79R3001 can accommodate a TAG bus width of 0-19 bits, compatible with a variety of cache sizes and cacheable main memory choices. The IDT79R3001 allows the system designer to scale the synchronous memory system exactly according to the system needs, thus eliminating extra memory and logic devices and achieving substantial cost savings with no loss of performance.

Thus, the synchronous memory interface of the IDT79R3001 allows for high-bandwidth memory systems to be implemented with a minimum of control logic. This is desirable, since RISC performance tends to be a function of memory bandwidth. By simplifying the design of the synchronous memory system (illustrated in Figure 5), it is easier for the system designer to achieve high performance with minimum chip count and without requiring ultra-fast or specialty components.

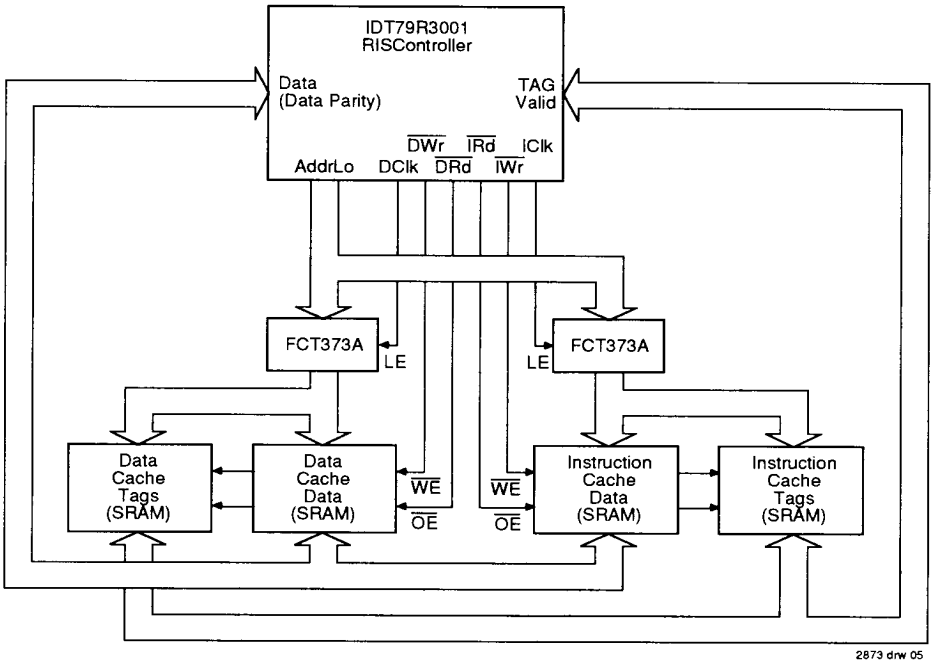


Figure 5. IDT79R3001 Synchronous Interface

The TAG Bus

The TAG bus of the IDT79R3001 has been designed to allow the system designer to implement the exact cache configuration that is right for the system. For larger caches, low-order TAG bits do not need to be supplied for the TAG comparison. Additionally, the number of high-order TAG bits supplied is determined by the system designer, according to the amount of cacheable main memory the system supports. Since most embedded systems would tend to implement caches of 16kB and greater, and cacheable memory spaces of 32MB or smaller, significant cost and area reductions are achieved by configuring a smaller TAG bus.

The system configures the on-chip TAG comparator at RESET Initialization time. If a TAG bit is not to be included in the synchronous memory TAG bit compare, a pull-down resistor of 4kΩ is connected to the appropriate IDT79R3001 TAG pin. If a TAG bit is to be included, no resistor is required (the IDT79R3001 pulls floating inputs to Vcc during RESET by a small pull-up, which is disabled when RESET is negated).

If a TAG bit is excluded from the cycle-by-cycle comparison, it is still driven out with the appropriate address value during write cycles or asynchronous memory reads. Thus, the system designer still has the full 4GB of address space available for address decoding, without requiring the synchronous memory to be able to cache all such addresses.

Figure 6 illustrates a reduced system, which implements 16kB of Instruction and 16kB of data cache, and 512MB of cacheable address space, using just 6 IDT71586 4kx16 Latched CacheRAM™ components and 4 pull-down resistors.

Note that in systems which do not implement the synchronous memory space as cache, then pull-down resistors would be added to all TAG pins. The Valid Pin still needs to be supplied on each cycle, thus allowing various memory schemes to be implemented (such as static column DRAM). However, the IDT79R3001 can be initialized to not assert the Valid pin as an output during Write cycles, simplifying the design of logic to drive the signal.

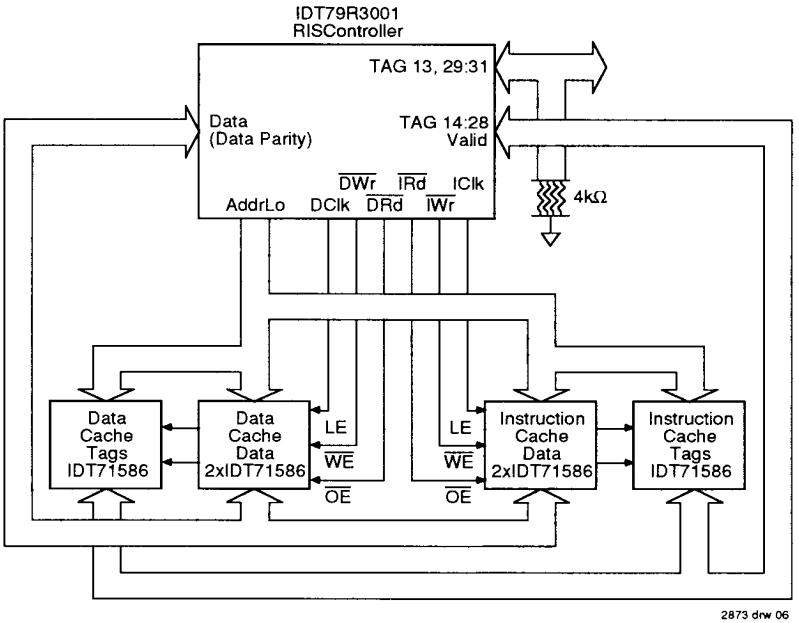


Figure 6. Small Footprint Cache for IDT79R3001

Cache Update

When the on-chip TAG comparator indicates that the item read from the cache was not the desired item, a cache-miss is processed. A main memory (asynchronous) transfer is automatically processed.

The IDT79R3001 desires to update the cache using a burst refill of multiple adjacent words from main memory. The processor is "stalled" until the first word of the block is available. The processor is then released, and the block of words is brought into the cache at the rate of one word per CPU clock cycle.

Note that if the cache-miss was in the instruction cache, the processor is capable of simultaneously executing the incoming instruction stream as the cache is updated, thus effectively making the cache update transparent to the system and increasing performance.

Write Cycles

The IDT79R3001 utilizes a write through cache. That is, data written by the processor is both written to the cache and main memory simultaneously. Thus, main memory always has a current copy of all data.

Typically, latching devices are used between the cache subsystem and the slower main memory. These Write Buffers capture the data simultaneous with the cache update, allowing the processor to continue to the next cycle without actually waiting for the main memory transfer to complete. The IDT79R3001 generates parity over the data field on write cycles, which can be propagated into both the synchronous and asynchronous memory spaces.

When the processor writes less than a 32-bit quantity (a "partial" word), the processor can perform a "read-modify-write" of the cache. That is, the processor will read the 32-bit word containing the partial address(es) to be updated from the cache. If a "hit" occurs, then the new data will be merged with the old and the new 32-bit value will be written both to the cache and to main memory. If a cache "miss" occurs, then only the partial data is written to main memory and the cache is unchanged. Partial word capability is selected as a RESET option.

THE ASYNCHRONOUS MEMORY INTERFACE

The IDT79R3001 also supports an asynchronous memory interface, which supports the use of slower memory devices such as slow DRAM or EPROM and also supports the use of peripherals and other "non-cacheable" devices.

In general, if a cache-miss (or parity error, if enabled) occurs, the processor will automatically use the asynchronous memory interface to retrieve the desired data, and will update the cache accordingly.

Additionally, software can force the use of the asynchronous memory space through the use of the on-chip MMU. When the processor seeks either instructions or data within a certain address range (kseg1), the processor knows that this data is uncacheable and will perform an asynchronous memory transfer. Additionally, within cacheable memory, TLB entries can be used to make certain pages as "uncacheable". When an address of an "uncacheable" page is used, the processor will automatically use the asynchronous memory space.

The asynchronous memory space uses the same data bus as the synchronous memory space. This facilitates the automatic updating of cache memory when the asynchronous memory is accessed due to cache-miss activity or memory writes. The asynchronous address bus is composed from the synchronous memory AddrLo bus, and the TAG bus. External logic devices (such as IDT74FCT374A registers) are used to capture AddrLo and TAG values for the asynchronous transfer address. Note that systems which exclude individual TAG bits from comparison (to reduce cache width) still have all TAGs available as outputs.

The data path between the processor and the asynchronous memory space is managed according to the needs of the application. Write Buffer FIFO devices, such as the IDT79R3020, are used to capture address and data during store cycles. These devices are used to capture the data in one-cycle, and allow the processor to continue to execute from the synchronous memory while the slower asynchronous memory actual retires the write.

The read path is also constructed according to the needs of the system. If block refill is used, then the read path is highly dependent on the design of the main memory system. Pipeline devices such as IDT74FCT540A, or simple latches such as IDT74FCT374, may be used.

A simple asynchronous memory interface is shown in Figure 7. In this system, main memory is assumed to be fast enough to support the block refill requirements of the system, thus simplifying the read path. In fact, both the read and write data paths are actually managed through a single set of IDT29FCT52A bidirectional latching transceivers.

During write cycles (which are typically captured by Write Buffers), the processor asserts $\overline{\text{MemWr}}$ to indicate that a write cycle is in progress. The memory system negates WrBusy to indicate that the processor is done with the write cycle.

5

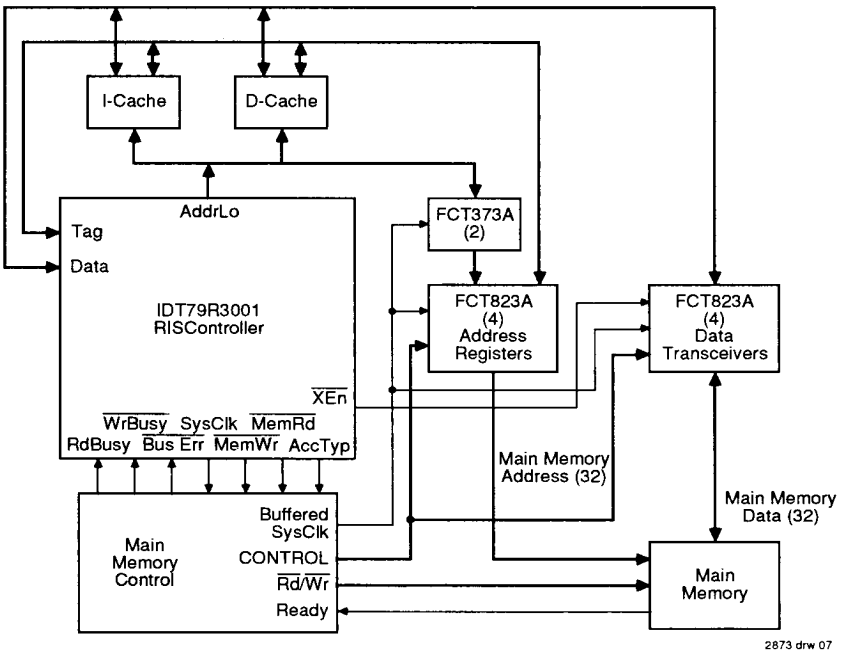


Figure 7. IDT79R3001 Asynchronous Interface

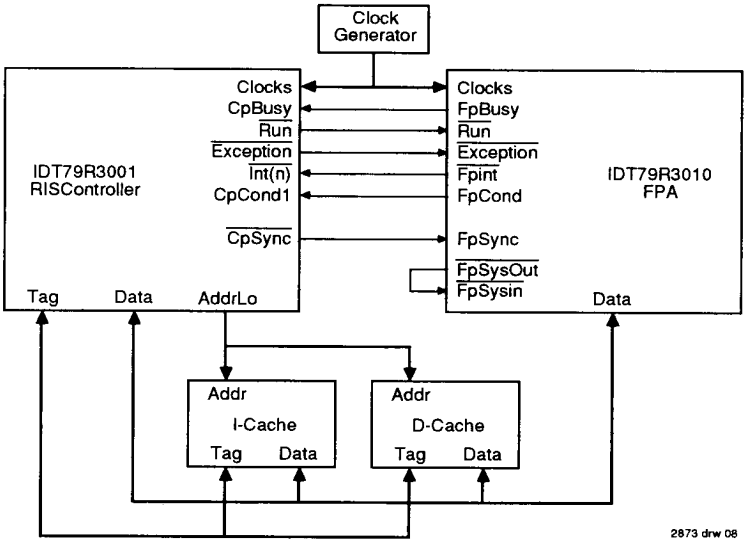


Figure 8. IDT79R3001 Interface to IDT79R3010 Floating Point Co-Processor

During read cycles, the processor will assert $\overline{\text{MemRd}}$ to indicate that a main memory read is in progress. The memory system will hold RdBusy active until the desired data is available. The processor will activate the $\overline{\text{XEn}}$ signal to allow data to be passed from the main memory to the processor databus. If the cache is to be updated with the new data, then the processor will assert the appropriate cache write signal to allow the cache RAMs to capture the incoming databus.

The AccTyp bus is used to indicate the size of the data transfer (8-, 16-, 24-, or 32-bits), and for main memory reads, whether or not the data is "cacheable". This simplifies the main memory address decoding, since the AccTyp indicates whether the main memory needs to perform a burst read of multiple words.

Co-Processor Interface

The IDT79R3001 implements a co-processor interface, which allows the use of the IDT79R3010 high-performance RISC Floating Point Accelerator without requiring the use of external interface components.

The co-processor interface has been designed to make system co-processors appear to the programmer as if they were on-chip extensions of the core execution engine. Thus, the IDT79R3010 FPA works as a true co-processor, rather than as a peripheral which must be programmed.

In the IDT79R3001 co-processor model, the CPU is responsible for controlling all data cycles. The co-processor keeps in synchronization with the CPU (including the pipeline stages), and uses a Phase-Locked Loop to keep synchronized with the processor bus traffic. The co-processor then "snoops" the data bus, watching for co-processor instructions. It also knows when data cycles on the bus are intended for it (either as a target in co-processor load operations, or as a source for co-processor restore operations), and performs the data portion of the operation when appropriate. Thus, co-processors effectively load and store directly with memory, without requiring operands to go through the CPU first. This achieves the highest levels of performance (note that the co-processor interface also supports move, whereby data can be moved directly between the CPU and any co-processor).

Figure 8 illustrates the use of the IDT79R3010 in a IDT79R3001 system. The co-processor interface manages synchronization between the parts, and is used to communicate status from the co-processor to the CPU. CpBusy, or Co-processor Busy, stalls the CPU until the busy co-processor resource (requested by a co-processor instruction) is free, and CpCond, or Co-processor Condition, is used to report status on co-processor test instructions. CpSync, is used to help the co-processor stay "locked" to the CPU, so that the co-processor knows when data is on the bus to be sampled on load operations, or when to place data on the bus for store operations.

Note that the co-processor sits on the same data bus as the CPU, but has no connection to the address bus. The CPU is responsible for performing all memory addressing, including the determination of "cache hit", write-buffer full cycles, and any processing that might be required for cache misses.

INTERRUPTS

The IDT79R3001 features 6 separate interrupt input pins. Interrupts are not vectored, but rather cause the general exception vector address to be the next execution address.

These pins are not encoded internally; external logic can choose to implement these interrupt lines as either 6 or 64 interrupt sources; software would then perform the appropriate decoding to get to the specific interrupt handler.

Interrupts are recognized in the ALU stage of the on-chip pipeline. Instructions less advanced in the pipeline are "flushed" and will be restarted when the return from exception occurs (an on-chip register contains the address of the instruction which was excepted). Instructions further advanced in the pipeline are allowed to continue. Unlike other RISC processors, the IDT79R3001 does not require the programmer to save and restore pipeline status to allow normal execution to be resumed. Depending on the application and exception, at most software would need to save/restore the on-chip data registers, status register, Exception PC and exception "cause" register.

Note that the co-processor model includes "precise exceptions." That is, an exception is signaled to the exact instruction which generated the exceptional condition. No further state commitments are made by the IDT79R3001 and, thus, the exact context at the time of the exception is known to the programmer. This is true even for multi-cycle operations, such as those of the FPA.

DMA INTERFACE

The IDT79R3001 features a simple DMA interface which allows an external master to gain control of the synchronous memory space. Note that it is not necessary to include logic on the CPU to arbitrate for the asynchronous memory space; the read/write buffer interface is where such arbitration logic belongs and it is left to the system designer to implement the type of asynchronous memory structure that best fits the application.

When an external master "owns" the synchronous bus, the CPU will tri-state the following pins and buses:

- **AddrLo:** The synchronous memory direct address bus.
- **Data & Tag:** The synchronous memory RAM data lines.
- **Cache Control:** $\overline{IRd}$, $\overline{IWr}$, ICIk, $\overline{DRd}$, $\overline{DWr}$ and DCIk. This allows the external master to use the existing control lines to control the synchronous memory.
- **XEn:** The read buffer transceiver enable, which will allow the external master to use the read/write buffer path for DMA.
- **Valid:** This enables the DMA interface to be used for multi-processing applications.

5

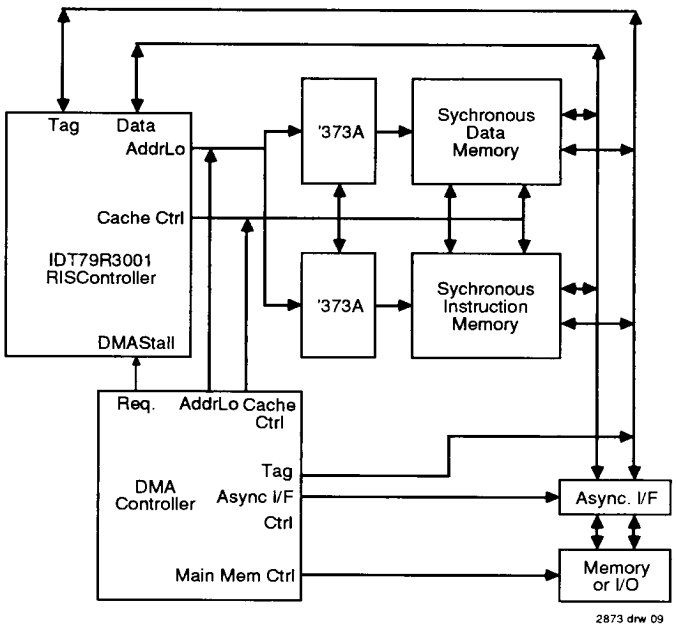


Figure 9. IDT79R3001 DMA Interface

MODE SELECTABLE FEATURES FOR THE IDT79R3001

Input	W Cycle	X Cycle	Y Cycle	Z Cycle
$\overline{Int0}$	Reserved	Reserved	Reserved	Reserved
$\overline{Int1}$	Reserved	Reserved	Reserved	Reserved
$\overline{Int2}$	$\overline{DBlkSize0}$	$\overline{DBlkSize1}$	$\overline{Parity\ On}$	Valid Output
$\overline{Int3}$	$\overline{IBlkSize0}$	$\overline{IBlkSize1}$	StorePartial	ControlLow
$\overline{Int4}$	$\overline{PllOn}$	$\overline{PllOn}$	$\overline{PllOn}$	$\overline{PllOn}$
$\overline{Int5}$	Reserved	BigEndian	$\overline{TriState}$	Reserved

NOTE:
1. Reserved signals must be "HIGH" during these cycles.

2873 tbl 01

The DMA interface consists of a single input signal, DMAStall, which causes the processor to stall and to tri-state the above named lines. The external master is guaranteed mastership of the bus within a very short number of cycles, depending on the exact external bus activity of the CPU when the DMA was requested. The DMA master negates the DMAStall signal when the DMA operation is completed to allow the CPU to resume processing. Consult the *IDT79R3001 Hardware User's Guide* for more details.

Figure 9 illustrates the system connection of an external DMA master to a IDT79R3001 system.

ADVANCED FEATURES

The IDT79R3001 contains special features which provide added flexibility across a number of applications, as well as allow for system diagnostic support.

In support of diagnostics, the IDT79R3001 allows for cache "swapping" (interchange of which memory bank is for instruction and which is for data), which is useful in system initialization, cache flushing, and diagnostics. Additionally, the caches can be "isolated" from main memory, which forces cache "hits" to occur regardless of the tag comparison, and which is useful in determining that the synchronous memory space RAMs are functional.

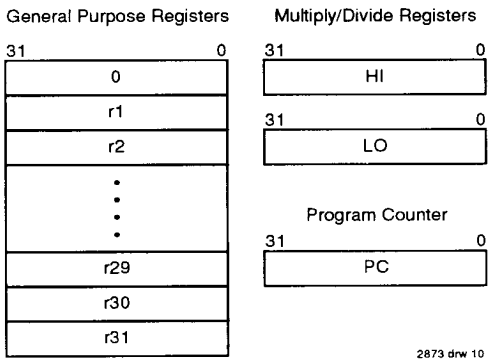


Figure 10. IDT79R3001 Registers

An additional feature is the ability to enable parity checking over the data field of each synchronous memory. If parity is enabled, the processor will check the parity when a synchronous access occurs; if a parity error is detected, it is signaled to the external world on the $\overline{Parity\ Error}$ signal and a cache-miss cycle is processed. the $\overline{Parity\ Error}$ signal will remain low until the parity error flag in the CP0 status register is cleared by software.

A number of other system selectable features are selected at reset time. The input reset "vectors" are sampled on the interrupt input lines during the last four cycles of the reset period. The input vectors are listed in Table 1. These selections include the ability to select the block refill sizes for each of the instruction and data memories, whether Big Endian or Little Endian order is to be used, whether to use data parity, and whether or not to accommodate a Phase-Locked Loop for a co-processor. The initialization of the CPU and meaning of each input vector is more fully explained in the *IDT79R3001 Hardware User's Guide.*

PROCESSOR ARCHITECTURE

The IDT79R3001 is a full implementation of the IDT79R2000A/IDT79R3000 Instruction Set Architecture (the MIPS-I ISA). This architecture is discussed in great detail in *MIPS RISC Architecture*, available from IDT.

IDT79R3001 CPU Registers

The IDT79R3001 CPU provides 32 general purpose (orthogonal) 32-bit registers, a 32-bit Program Counter and two 32-bit registers used to hold the results of the CPU integer multiply and divide operations.

Two of the 32 general registers have special purposes designed to increase processor performance: register r0 is hardwired to the value "0", a useful constant; and register r31 is used as the link register in jump-and-link instructions (the return address for subroutine calls). Otherwise, there is no requirement that a particular register be used as a stack or frame pointer, etc., although there is a register convention as part of the "mips ABI" (Applications Binary Interface standard) which the compiler suite uses.

The CPU registers are illustrated in Figure 10. Note that there is no Program Status Word register shown in this figure. The functions traditionally provided by a PSW register are instead provided in the Status and Cause Registers incorporated within the on-chip System Control Co-Processor (CP0). The instruction set does not use condition codes.

Instruction Set Overview

All IDT79R3001 instructions are 32 bits long and there are only three instruction formats (see Figure 11). This approach simplifies decoding, thus minimizing instruction execution time. The IDT79R3001 processor initiates a new instruction on every RUN cycle, and is able to complete an instruction on almost every clock cycle. The only exceptions are the LOAD instructions and BRANCH instructions, which each have a single cycle of latency associated with their execution (that is, the instruction immediately after the branch is always executed regardless of the branch condition; similarly, the data loaded by a LOAD instruction is not available to the subsequent instruction). However, in the majority of cases the compilers (and even the MIPS assembler) are able to reorder instructions to fill these latency cycles with useful instructions which do not require the results of the previous instruction (in the worst case, a NOP instruction is inserted). This effectively eliminates these latency effects and does not require the applications programmer to be aware of the pipeline structure.

The actual instruction set of the CPU was determined after extensive simulations to determine which instructions should be implemented in hardware and which operations are best synthesized in software from other basic operations. This methodology has resulted in the highest performance processor available.

The IDT79R3001 instruction set can be divided into the following groups:

- **Load/Store Instructions** move data between memory and the general registers. These are all "I-Type" instructions. The only addressing mode supported is base register plus signed, immediate 16-bit offset. This effectively allows three addressing modes: register plus offset, register (using zero offset), and immediate (using r0,the zero register).

  The Load instruction has a single cycle of latency, as described above. That is, the instruction immediately after the load instruction cannot rely on the new data; however, the assembler and compilers automatically handle this, reordering code to insure that no conflicts occur. Note that the store operation has no latency in its effect.

  Loads and stores can be performed on byte, half-word, word, or unaligned word data (32-bit data not aligned on a modulo-4 address).
- **Computational Instructions** perform arithmetic, logical, and shift operations on values in registers. They occur in both "R-Type" (both operands and the result are general registers), and "I-Type" (one operand is a 16-bit immediate value) formats.

  Note that computational instructions are three operand instructions: that is, the result register can be different from both source registers. This means that operands need not be overwritten by arithmetic operations. This results in a more efficient use of the register set, and further increases performance.

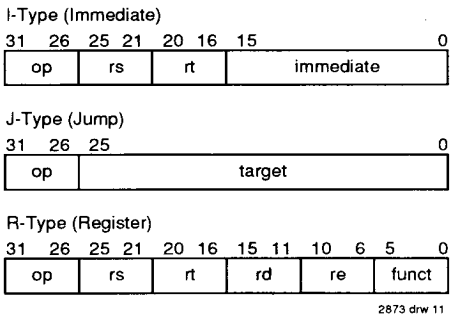


Figure 11. IDT79R3001 Instruction Formats

- **Jump and Branch Instructions** change the flow of control of a program. Jumps are always to a paged absolute address formed by combining a 26-bit target with four bits of the Program Counter ("J-Type" format for subroutine calls), or 32-bit register byte addresses ("R-Type," for Returns and dispatches). Branches have 16-bit offsets relative to the program counter ("I-Type").

  Jump and Link instructions save a return address in Register 31. The IDT79R3001 instruction set features numerous branch conditions. Included is the ability to branch based on a comparison of two registers, or on the comparison of a register to zero. Thus, net performance is increased since the processor does not have to precede the branch instruction with arithmetic operations.
- **Co-processor Instructions** perform operations in the co-processors (such as the IDT79R3010 FPA). Co-processor Loads and Stores are "I- Type;" computational instructions have co-processor dependent formats.
- **Co-processor 0 Instructions** perform operations on the System Control Co-processor (CP0) registers to manipulate the memory management and exception handling facilities of the on-chip co-processor.
- **Special Instructions** perform a variety of tasks, including movement of data between general and special registers, system calls, and breakpoint operations. These are always "R-Type."

IDT79R3001 System Control Co-processor (CP0)

The IDT79R3001 can operate with up to four tightly coupled co-processors, designated CP0-CP3. CP0 is included on-chip as co-processor 0, the System Control co-processor. CP0 is responsible for supporting both the virtual memory system and the exception handling functions of the IDT79R3001.

5

IDT79R3001 INSTRUCTION SUMMARY

OP	Description	OP	Description
	Load/Store Instructions		**Multiply/Divide Instructions**
LB	Load Byte	MULT	Multiply
LBU	Load Byte Unsigned	MULTU	Multiply Unsigned
LH	Load Halfword	DIV	Divide
LHU	Load Halfword Unsigned	DIVU	Divide Unsigned
LW	Load Word	MFHI	Move From HI
LWL	Load Word Left	MTHI	Move To HI
LWR	Load Word Right	MFLO	Move From LO
SB	Store Byte	MTLO	Move To LO
SH	Store Halfword		
SW	Store Word		**Jump and Branch Instructions**
SWL	Store Word Left	J	Jump
SWR	Store Word Right	JAL	Jump and Link
		JR	Jump to Register
	Arithmetic Instructions (ALU Immediate)	JALR	Jump and Link Register
		BEQ	Branch on Equal
ADDI	Add Immediate	BNE	Branch on Not Equal
ADDIU	Add Immediate Unsigned	BLEZ	Branch on Less than or Equal to Zero
SLTI	Set on Less Than Immediate	BGTZ	Branch on Greater Than Zero
SLTIU	Set on Less Than Immediate Unsigned	BLTZ	Branch on Less Than Zero
		BGEZ	Branch on Greater than or Equal to Zero
ANDI	AND Immediate		
ORI	OR Immediate	BLTZAL	Branch on Less Than Zero and Link
XORI	Exclusive OR Immediate	BGEZAL	Branch on Greater than or Equal to Zero and Link
LUI	Load Upper Immediate		
			Special Instructions
	Arithmetic Instructions (3-operand, register-type)	SYSCALL	System Call
		BREAK	Break
ADD	Add		
ADDU	Add Unsigned		**Coprocessor Instructions**
SUB	Subtract	LWCz	Load Word from Coprocessor
SUBU	Subtract Unsigned	SWCz	Store Word to Coprocessor
SLT	Set on Less Than	MTCz	Move To Coprocessor
		MFCz	Move From Coprocessor
SLTU	Set on Less Than Unsigned	CTCz	Move Control to Coprocessor
AND	AND	CFCz	Move Control From Coprocessor
OR	OR	COPz	Coprocessor Operation
XOR	Exclusive OR	BCzT	Branch on Coprocessor z True
NOR	NOR	BCzF	Branch on Coprocessor z False
	Shift Instructions		**System Control Coprocessor (CP0) Instructions**
SLL	Shift Left Logical		
SRL	Shift Right Logical	MTC0	Move To CP0
SRA	Shift Right Arithmetic	MFC0	Move From CP0
SLLV	Shift Left Logical Variable	TLBR	Read indexed TLB entry
SRLV	Shift Right Logical Variable	TLBWI	Write Indexed TLB entry
SRAV	Shift Right Arithmetic Variable	TLBWR	Write Random TLB entry
		TLBP	Probe TLB for matching entry
		RFE	Restore From Exception

2873 tbl 02

CP0 Registers

As a co-processor, CP0 has a number of registers which it uses to perform its control functions. These include 64 fully associative Translation Lookaside Buffers (TLBs), used to manage the virtual memory space; registers to manage the TLB set; and the exception handling registers. Figure 12 illustrates the register set of the System Control Co-processor. Table 3 provides a brief explanation of the function of each of these registers. A more detailed explanation of the use of each of these registers is included in the *MIPS RISC Architecture* manual.

Memory Management System

The IDT79R3001 supports a virtual memory system, so that each task in a given application can be unaware of the addressing needs of other tasks. This is also useful in systems with limited physical memory; the IDT79R3001 provides for the logical expansion of memory by translating addresses composed in a large virtual space into available physical memory addresses.

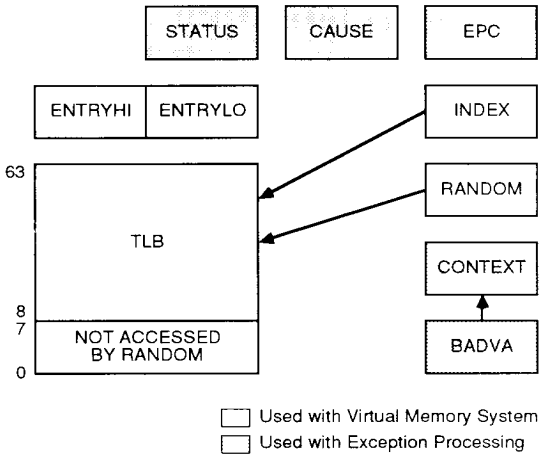


Figure 12. The System Control Co-processor (CP0) Registers

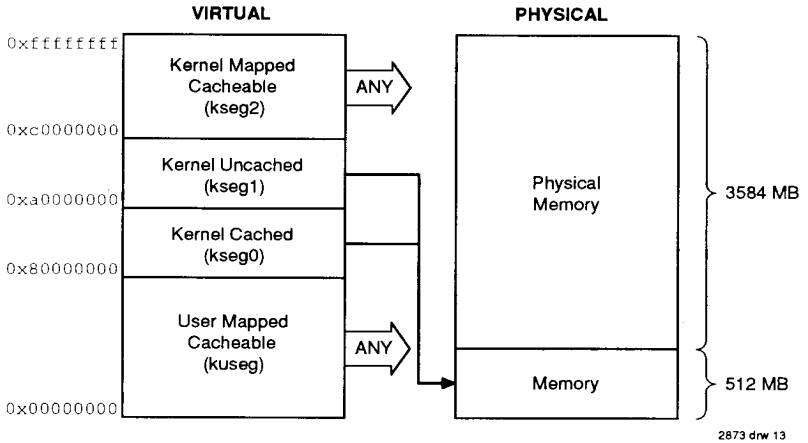
CP0 REGISTERS

Register	Description
EntryHi	High half of a TLB entry
EntryLo	Low half of a TLB entry
Index	Programmable pointer into TLB array
Random	Pseudo-random pointer into TLB array
Status	Mode, interrupt enables and diagnostic status information
Cause	Indicates nature of last exception
EPC	Exception Program Counter—contains address of instruction which detected the exception
Context	Pointer into kernel's virtual Page Table Entry array
BadVA	Most recent bad virtual address
PrID	Processor revision identification (Read only)

2873 tbl 03

5

MMU ADDRESS TRANSLATION



IDT79R3001 Operating Modes

The IDT79R3001 has two operating modes: User Mode and Kernel Mode. The IDT79R3001 normally operates in the User Mode until an exception is detected, forcing it into the Kernel Mode. The processor remains in Kernel Mode until the exceptions are handled and the processor executes an RFE (Return from Exception) instruction, which will restore it to User Mode. Kernel Mode allows software to alter machine state information such as that contained in the CP0 registers; that is, if in User Mode an access is attempted to Co-processor 0 and the Kernel has not enabled the User to access the co-processor, an exception will occur. Similarly, if a User task attempts to use a Kernel virtual address, an exception will occur. Thus, system resources are protected from User tasks.

The manner in which memory addresses are translated (mapped) depends on the operating mode of the IDT79R3001 and on the virtual address desired. Figure 13 illustrates the virtual address mapping performed by the IDT79R3001:

User Mode — in this mode, a single, uniform virtual address space (kuseg) of 2GB is available to each user task (tasks are further identified by a 6-bit process identifier field in order to form unique virtual addresses). All references to this segment are mapped using the TLB, which utilizes both the virtual address and the Process ID field to perform the virtual-to-physical mapping (note that this allows the cache to be shared by up to 64 User processes at a time without requiring time consuming Cache or TLB flushing).

Kernel Mode—Four separate segments are accessible through this mode:

- **kuseg**—When in the Kernel Mode, references to this segment are treated just like User Mode references, thus streamlining Kernel accesses to User memory.
- **kseg0**—References to this 512MB segment may use the cache memory, but are not translated by the TLB. Instead, these addresses map directly to the first 512MB of the physical address space. Note that many dedicated embedded applications will utilize this address space and kseg1 only, rather than any of the TLB mapped segments.
- **kseg1**—References to this 512MB segment are not mapped through the TLB. Additionally, this memory is viewed as uncacheable, which means that references through this segment will always use the asynchronous memory interface. As with kseg0, references through this segment are hard-mapped to the first 512MB of physical memory. When

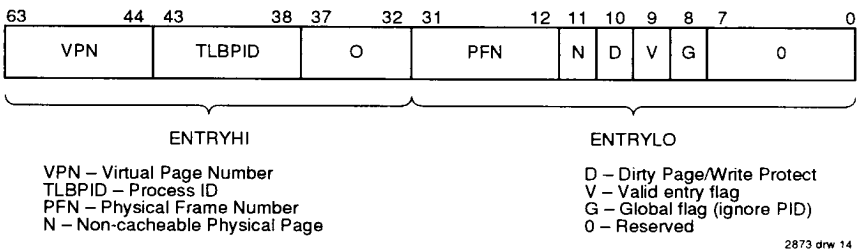


Figure 14. TLB Entry Format

the processor boots, the reset vector is contained in this segment, so that the processor does not require either the cache or the TLB to be valid at RESET time.

- **kseg2**—References to this 1GB segment are always mapped through the TLB. As with kuseg, the ability of memory pages to be cached is determined by a bit setting in the TLB entry for that page.

The Translation Lookaside Buffer (TLB)

The translation of virtual addresses in either kuseg or kseg2 (mapped segments) is performed by the on-chip Translation Lookaside Buffer array. This array consists of 64 fully-associative (content addressable) memory elements. Each entry maps a 4kB virtual page to a 4kB physical page. Each TLB entry contains other information about the virtual address it maps (such as which User process it maps) and also about the physical address (such as whether it is cacheable or writeable).

Figure 14 illustrates the format of each TLB entry. The translation operation is illustrated in Figure 15. The upper portion of the desired virtual address is compared against the VPN field of each TLB entry. Additionally, the current process ID (contained in the TLBHI register) is matched against the PID field of the TLB entry (if the TLB entry is marked as Global, the PID comparison is ignored). If a match occurs, and the TLB entry is marked as Valid, then the translation is completed by replacing the VPN of the virtual address with the corresponding PFN (Physical Frame Number).

Note that the use of the TLB does not incur an execution penalty, since the execution engine pipeline includes stages to cover for the time required to make the TLB search and translation.

TLB misses occur when no successful match occurs. These events are handled in software. The CP0 registers give the software enough information to obtain the appropriate TLB entry at speeds which exceed those achieved by many CPUs which use hardware TLB replacement (10-12 cycles under UNIX).

When a TLB miss occurs, the address of the instruction which was executing is stored in the EPC register, and the BadVA register contains the address which was being translated. The Context register uses the BadVA value to generate a direct pointer to the kernel Page Table Entry for the desired virtual address. The Random register suggests the TLB entry to be replaced by the new entry. Note that the lower eight TLB entries are not pointed to by Random; the kernel software can thus insure that it is constantly mapped, and deterministic response is guaranteed.

BACKWARD COMPATIBILITY WITH IDT79R2000A AND 79R3000 PROCESSORS

The IDT79R3001 can execute the same binary software (either kernel or user) that is executed by either the IDT79R2000A or IDT79R3000. At the system level, some hardware re-design is necessary to achieve the cost savings inherent in the IDT79R3001 hardware interface.

5

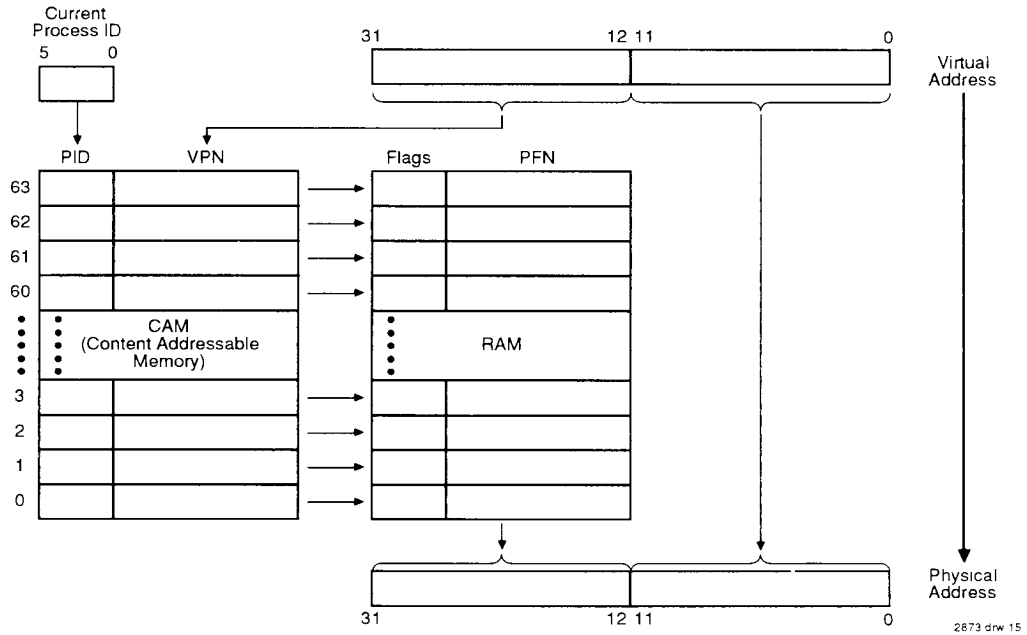


Figure 15. Virtual to Physical TLB Translation

PIN DESCRIPTIONS

Pin Name	I/O	Description
Memory Interface		
Data (0:31)	I/O	A 32-bit bus used for all instruction and data transmission among the processor, synchronous memory space, asynchronous memory space and co-processors.
DataP (0:3)	I/O	A 4-bit bus containing even parity over the data bus. If parity checking is enabled, a parity error will cause the $\overline{\text{PErr}}$ signal to be asserted and a cache-miss to occur. Regardless of whether parity checking is enabled, the processor will always generate parity on writes.
Tag (13:31)	I/O	A 19-bit bus used for transferring cache tags and high-order address bits between the processor, caches and asynchronous memory spaces.
AddrLo (0:23)	O	A 24-bit bus containing low-order byte addresses for both the synchronous (cache) and asynchronous memory spaces.
Synchronous Memory Control		
$\overline{\text{IRd}}$	O	The output enable for the instruction cache. The polarity of this signal is selectable.
$\overline{\text{IWr}}$	O	The write enable for the instruction cache. The polarity of this signal is selectable.
IClk	O	The instruction cache address latch clock. The clock runs continuously.
$\overline{\text{DRd}}$	O	The output enable for the data cache. The polarity of this signal is selectable.
$\overline{\text{DWr}}$	O	The write enable for the data cache. The polarity of this signal is selectable.
DClk	O	The data cache address latch clock. The clock runs continuously.
Valid	I/O	A high on this signal indicates that the Tags just read from the cache are valid. When a cache update occurs, the processor will generate the appropriate Valid bit.
$\overline{\text{PErr}}$	O	If parity checking is enabled, this signal is an active low output of the internal CP0 parity error status bit. It is driven low when a parity error is detected and remains low until software clears the parity error flag in the status register. This pin is physically the same pin as AccTyp2. Its function is selected during device reset.
Asynchronous Memory Interface		
$\overline{\text{XEn}}$	O	The transceiver enable for the read buffer.
AccTyp (0:2)	O	A 3-bit bus used to indicate the size of data being transferred on the asynchronous memory bus, whether or not a data transfer is occurring and the purpose of the transfer. If parity checking is enabled, AccTyp2 becomes the $\overline{\text{PErr}}$ signal.
$\overline{\text{MemWr}}$	O	Signals the occurrence of an asynchronous memory write cycle.
$\overline{\text{MemRd}}$	O	Signals the occurrence of an asynchronous memory read cycle.
$\overline{\text{BusError}}$	I	Signals the occurrence of a bus error during an asynchronous memory transfer cycle.
$\overline{\text{Run}}$	O	Indicates whether the processor is in a RUN or STALL state.
$\overline{\text{Exception}}$	O	Indicates the instruction about to commit processor state should be aborted and other exception related information.
$\overline{\text{SysOut}}$	O	A clock derived from the internal processor clock used to generate the system clock.
RdBusy	I	The asynchronous memory read stall termination signal. In most system designs, RdBusy is normally asserted and is deasserted only to indicate the successful completion of the memory read. RdBusy is sampled by the processor only during memory read stalls.
$\overline{\text{WrBusy}}$	I	The asynchronous memory write stall initiation/termination signal. $\overline{\text{WrBusy}}$ is only sampled during write operation.
Co-Processor Interface		
$\overline{\text{CpSync}}$	O	A clock which is identical to $\overline{\text{SysOut}}$ and used by co-processors for timing synchronization with the CPU.
CPBusy	I	The co-processor busy stall initiation/termination signal.
CpCond (0:3)	I	A 4-bit bus used to transfer conditional branch status from the co-processors to the CPU. CpCond(0) is used to control whether or not a cache burst refill occurs; the other signals are used as input port pins for co-processor branch instructions.
Processor Control Signals		
DMAStall	I	DMA Stall. Signals to the processor that it should stall accesses to the synchronous memories and tri-state the synchronous memory interface.
$\overline{\text{Int}}$ (0:5)	I	A 6-bit bus used to signal maskable interrupts to the CPU. A reset time, mode values are sampled from this bus to initialize the processor. During normal operation, these signals are not latched by the processor and must remain asserted until the processor acknowledges the interrupt (through software) to the interrupt source.
Clk2xSys	I	The master double frequency input clock, used to generate $\overline{\text{SysOut}}$.
Clk2xSmp/Rd	I	A double frequency clock input used to determine the sample point for data coming into the CPU and co-processors and used to determine the enable time of the synchronous memory RAMs.
Clk2xPhi	I	A double frequency clock input used to determine the position of the two internal phases.
$\overline{\text{Reset}}$	I	Initialization input used to force execution starting from the reset memory address. $\overline{\text{Reset}}$ should be asserted asynchronously but must be negated synchronously with the leading edge of $\overline{\text{SysOut}}$.

ABSOLUTE MAXIMUM RATINGS[1, 3]

Symbol	Rating	Commercial	Unit
VTERM	Terminal Voltage with Respect to GND	–0.5 to +7.0	V
TA, TC	Operating Temperature	0 to +70[4] (Ambient) 0 to +90[5] (Case)	°C
TBIAS	Case Temperature Under Bias	–55 to +125[4] 0 to +90[5]	°C
TSTG	Storage Temperature	–55 to +125	°C
IIN	Input Voltage	–0.5 to +7.0	V

2873 tbl 05

NOTE:

1. Stresses greater than those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.
2. V_{IN} minimum = –3.0V for pulse width less than 15ns. V_{IN} should not exceed V_{CC} +0.5 Volts.
3. Not more than one output should be shorted at a time. Duration of the short should not exceed 30 seconds.
4. 16-25 MHz only.
5. 40 MHz only.

AC TEST CONDITIONS

Symbol	Parameter	Min.	Max.	Unit
VIH	Input HIGH Voltage	3.0	—	V
VIL	Input LOW Voltage	—	0.4	V
VIHS	Input HIGH Voltage	3.5	—	V
VILS	Input LOW Voltage	—	0.4	V

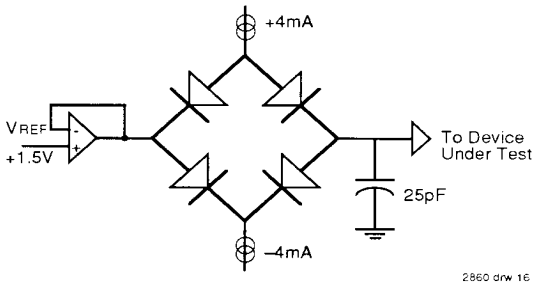
2873 tbl 06

RECOMMENDED OPERATING TEMPERATURE AND SUPPLY VOLTAGE

Grade	Temperature	GND	Vcc
Commercial 16-33 MHz	0°C to +70°C (Ambient)	0V	5.0 ±5%
Commercial 40 MHz	0°C to +90°C (Case)	0V	5.0 ±5%

2873 tbl 07

OUTPUT LOADING FOR AC TESTING



2860 drw 16

Signal	CL
IRd, IWr, DRd, DWr	50pf
All Others	25pf

5

DC ELECTRICAL CHARACTERISTICS
COMMERCIAL TEMPERATURE RANGE (T_A = 0°C to +70°C, V_{CC} = +5.0V ±5%)

Symbol	Parameter	Test Conditions	16.67MHz		20.0MHz		25.0MHz		33.33MHz		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
V_{OH}	Output HIGH Voltage	V_{CC} = Min., I_{OH} = –4mA	3.5	—	3.5	—	3.5	—	3.5	—	V
V_{OL}	Output LOW Voltage	V_{CC} = Min., I_{OL} = 4mA	—	0.4	—	0.4	—	0.4	—	0.4	V
V_{OHT}	Output HIGH Voltage(4,7)	V_{CC} = Min., I_{OH} = –8mA	2.4	—	2.4	—	2.4	—	2.4	—	V
V_{OHC}	Output HIGH Voltage(8)	V_{CC} = Min., I_{OH} =–4mA	4.0	—	4.0	—	4.0	—	4.0	—	V
V_{OLT}	Output LOW Voltage(4,7)	V_{CC} = Min., I_{OL} = 8mA	—	0.8	—	0.8	—	0.8	—	0.8	V
V_{IH}	Input HIGH Voltage(5)		2.0	—	2.0	—	2.0	—	2.0	—	V
V_{IL}	Input LOW Voltage		—	0.8	—	0.8	—	0.8	—	0.8	V
V_{IHS}	Input HIGH Voltage(2,5)		3.0	—	3.0	—	3.0	—	3.0	—	V
V_{ILS}	Input LOW Voltage(1,2)		—	0.4	—	0.4	—	0.4	—	0.4	V
I_{RESET}	Input HIGH Current(6)		10	100	10	100	10	100	10	100	µA
C_{IN}	Input Capacitance(7)		—	10	—	10	—	10	—	10	pF
C_{OUT}	Output Capacitance(7)		—	10	—	10	—	10	—	10	pF
I_{CC}	Operating Current	V_{CC} = Max.	—	575	—	650	—	750	—	800	mA
I_{IH}	Input HIGH Leakage(3)	V_{IH} = V_{CC}	—	100	—	100	—	100	—	100	µA
I_{IL}	Input LOW Leakage(3)	V_{IL} = GND	–100	—	–100	—	–100	—	–100	—	µA
I_{OZ}	Output Tri-state Leakage	V_{OH} = 2.4V, V_{OL} = 0.5V	–100	100	–100	100	–100	100	–100	100	µA

NOTES:
2873 tbl 08

1. V_{IL} Min. = –3.0V for pulse width less than 15ns. VIL should not fall below –0.5V for larger periods.
2. V_{IHS} and V_{ILS} apply to Clk2xSys, Clk2xSmp/Rd, Clk2xPhi, CpBusy, and Reset.
3. These parameters do not apply to the clock inputs.
4. V_{OHT} and V_{OLT} apply to the bidirectional data and tag buses only. Note that V_{IH} and V_{IL} also apply to these signals. V_{OHT} and V_{OLT} are supplies as additional information to help the system designer understand the relationship between current drive and output voltage on these pins..
5. V_{IH} should not be held above V_{CC} + 0.5 volts.
6. The IDT79R3001 contains an internal pull-up/current source on the TAG pins to facilitate initialization. This current source is disconnected when Reset is inactive.
7. Guaranteed by design.
8. V_{OHC} applies to RUN and Exception.

DC ELECTRICAL CHARACTERISTICS
COMMERCIAL TEMPERATURE RANGE (T_C = 0°C to +90°C, V_{CC} = +5.0V ±5%)

Symbol	Parameter	Test Conditions	40.0MHz		Unit
			Min.	Max.	
V_{OH}	Output HIGH Voltage	V_{CC} = Min., I_{OH} = -4mA	3.5	—	V
V_{OL}	Output LOW Voltage	V_{CC} = Min., I_{OL} = 4mA	—	0.4	V
V_{OHC}	Output HIGH Voltage(7)	V_{CC} = Min., I_{OH} = -4mA	4.0	—	V
V_{OHT}	Output HIGH Voltage(4,6)	V_{CC} = Min., I_{OH} = -8mA	2.4	—	V
V_{OLT}	Output LOW Voltage(4,6)	V_{CC} = Min., I_{OL} = 8mA	—	0.8	V
V_{IH}	Input HIGH Voltage(5)		2.0	—	V
V_{IL}	Input LOW Voltage(1)		—	0.8	V
V_{IHS}	Input HIGH Voltage(2,5)		3.0	—	V
V_{ILS}	Input LOW Voltage(1,2)		—	–0.4	V
I_{RESET}	Input HIGH Current(6)		10	100	µA
C_{IN}	Input Capacitance(6)		—	10	pF
C_{OUT}	Output Capacitance(6)		—	10	pF
I_{CC}	Operating Current	V_{CC} = 5V, T_A = 70°C	—	850	mA
I_{IH}	Input HIGH Leakage(3)	V_{IH} = VCC	—	100	µA
I_{IL}	Input LOW Leakage(3)	V_{IL} = GND	–100	—	µA
I_{OZ}	Output Tri-state Leakage	V_{OH} = VCC, V_{OL} = GND	–100	100	µA

2873 tbl 09

NOTES:

1. V_{IL} Min. = –3.0V for pulse width less than 15ns. V_{IL} should not fall below –0.5 Volts for larger periods.
2. V_{IHS} and V_{ILS} apply to Clk2xSys, Clk2xSmp, Clk2xRd, Clk2xPhi, CpBusy, and $\overline{\text{Reset}}$.
3. These parameters do not apply to the clock inputs.
4. V_{OHT} and V_{OLT} apply to the bidirectional data and tag buses only. Note that V_{IH} and V_{IL} also apply to these signals. V_{OHT} and V_{OLT} are provided to give the designer further information about these specific signals.
5. V_{IH} should not be held above V_{CC} + 0.5 volts.
6. Guaranteed by design.
7. V_{OHC} applies to $\overline{\text{RUN}}$ and $\overline{\text{Exception}}$.

5

AC ELECTRICAL CHARACTERISTICS[1,4]
COMMERCIAL TEMPERATURE RANGE (T_A = 0°C to +70°C, V_{CC} = +5.0V ±5%)

Symbol	Parameter	Test Conditions	16.67MHz		20.0MHz		25.0MHz		33.33MHz		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
Clock											
TCkHigh	Input Clock HIGH[2]	Note 7	12.5	—	10	—	8.0	—	6.0	—	ns
TCkLow	Input Clock LOW[2]	Note 7	12.5	—	10	—	8.0	—	6.0	—	ns
TCkP	Input Clock Period[2]		30	500	25	500	20	500	15	500	ns
	Clk2xSys to Clk2xSmp/Rd[5]		0	Tcyc/4	0	Tcyc/4	0.0	Tcyc/4	0	Tcyc/4	ns
	Clk2xSmp/Rd to Clk2xPhi[5]		9.0	Tcyc/4	7.0	Tcyc/4	5.0	Tcyc/4	3.5	Tcyc/4	ns
Run Operation											
TDEn	Data Enable[3]		—	–2.0	—	–2.0	—	–1.5	—	–1.5	ns
TDDis	Data Disable[3]		—	–1.0	—	–1.0	—	–0.5	—	–0.5	ns
TDVal	Data Valid	Load= 25pF	—	3.0	—	3.0	—	2.0	—	2.0	ns
TWrDly	Write Delay	Load= 25pF	—	5.0	—	4.0	—	3.0	—	2.0	ns
TDS	Data Set-up		9.0	—	8.0	—	6.0	—	4.5	—	ns
TDH	Data Hold		–2.5	—	–2.5	—	–2.5	—	–2.5	—	ns
TCBS	CpBusy Set-up		13	—	11	—	9.0	—	7.0	—	ns
TCBH	CpBusy Hold		–2.5	—	–2.5	—	–2.5	—	–2.5	—	ns
TAcTy	Access Type (1:0)	Load= 25pF	—	7.0	—	6.0	—	5.0	—	3.5	ns
TAT2	Access Type2	Load= 25pF	17	—	14	—	12	—	—	8.5	ns
TMWr	Memory Write	Load= 25pF	1.0	27	1.0	23	1.0	18	—	9.5	ns
TExc	Exception	Load= 25pF	—	7.0	—	7.0	—	5.0	—	3.5	ns
TAval	Address Valid	Load= 25pF	—	2.0	—	2.0	—	1.5	—	1.0	ns
TInts	$\overline{\text{Int(n)}}$ Set-up		9.0	—	8.0	—	6.0	—	4.5	—	ns
TWTH	$\overline{\text{Int(n)}}$ Hold		–2.5	—	–2.5	—	–2.5	—	–2.5	—	ns
Stall Operation											
TSAVal	Address Valid	Load= 25pF	—	30	—	23	—	20	—	15	ns
TSAcTy	Access Type	Load= 25pF	—	27	—	23	—	18	—	13.5	ns
TMRdi	Memory Read Initiate	Load= 25pF	1.0	27	1.0	23	1.0	18	1.0	13.5	ns
TMRdT	Memory Read Terminate	Load= 25pF	1.0	2.0	1.0	23	1.0	5.0	1.0	13.5	ns
TStl	Run Terminate	Load= 25pF	3.0	17	3.0	15	3.0	10	2.0	7.5	ns
TRun	Run Initiate	Load= 25pF	—	7.0	—	6.0	—	4.0	—	3.0	ns
TSMWr	Memory Write	Load= 25pF	3.0	27	3.0	23	3.0	18	2.0	9.5	ns
TSEc	Exception Valid	Load= 25pF	—	15	—	13	—	10	—	7.5	ns
TDMADis	DMA Drive On	Load= 25pF	3.0	15	3.0	15	3.0	15	3.0	15	ns
TDMAEn	DMA Drive Off	Load= 25pF	—	10	—	10	—	10	—	10	ns
Reset Initialization											
TRST	Reset Pulse Width		6.0	—	6.0	—	6.0	—	6.0	—	Tcyc
TRSTTAG	Reset Pulse Width, Pull-downs on Tag		140	—	140	—	140	—	140	—	µs
Capacitive Load Deration											
CLD	Load Derate[6]		0.5	1.0	0.5	1.0	0.5	1.0	0.5	1.0	ns/25pF

2873 tbl 11

NOTES:
1. All timings are referenced to 1.5V.
2. The clock parameters apply to all three 2xClocks: Clk2xSys, Clk2xSmp/Rd, and Clk2xPhi.
3. This parameter is guaranteed by design.
4. These parameters are illustrated in detail in the *IDT79R3001 Hardware Interface Guide.*
5. T_{cyc} is one CPU clock cycle (2 cycles of a 2x clock).
6. With the exception of $\overline{\text{Run}}$, no two signals on a given device will derate for a given load by a difference greater than 15%.
7. Transition time <2.5ns for 33MHz; <5ns for lower speeds.

AC ELECTRICAL CHARACTERISTICS(1,4)

COMMERCIAL TEMPERATURE RANGE (Tc = 0°C to +90°C, Vcc = +5.0V ±5%)

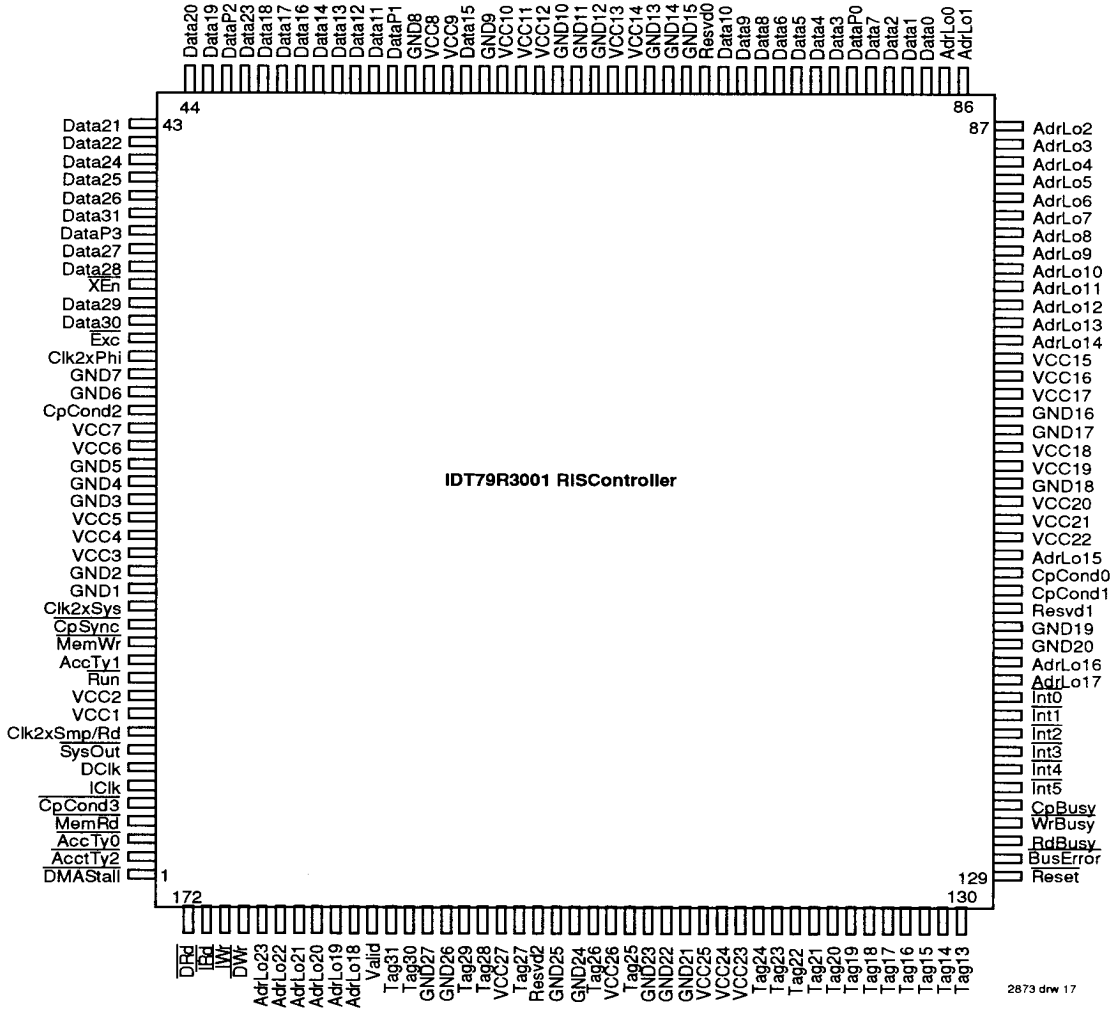
Symbol	Parameter	Test Conditions	40.0MHz		Unit
			Min.	Max.	
Clock					
TCkHigh	Input Clock HIGH(2)	Transition < 2.5ns	5.0	—	ns
TCkLow	Input Clock LOW(2)	Transition < 2.5ns	5.0	—	ns
TCkP	Input Clock Period(2)		12.5	500	ns
	Clk2xSys to Clk2xSmp/Rd(5)		0	Tcyc/4	ns
	Clk2xSmp/Rd to Clk2xPhi(5)		3.0	Tcyc/4	ns
Run Operation					
TDEn	Data Enable(3)		—	–1.5	ns
TDDis	Data Disable(3)		—	–0.5	ns
TDVal	Data Valid	Load= 25pF	—	1.5	ns
TWrDly	Write Delay	Load= 25pF	—	2.0	ns
TDS	Data Set-up		4.0	—	ns
TDH	Data Hold		–2.5	—	ns
TCBS	CpBusy Set-up		6.0	—	ns
TCBH	CpBusy Hold		–2.5	—	ns
TAcTy	Access Type (1:0)	Load= 25pF	—	3.0	ns
TAT2	Access Type2	Load= 25pF	—	7.5	ns
TMWr	Memory Write	Load= 25pF	—	9.0	ns
TExc	Exception	Load= 25pF	—	3.0	ns
Stall Operation					
TSAVal	Address Valid	Load= 25pF	—	12.5	ns
TSAcTy	Access Type	Load= 25pF	—	9.0	ns
TMRdi	Memory Read Initiate	Load= 25pF	—	9.0	ns
TMRdT	Memory Read Terminate	Load= 25pF	—	9.0	ns
TStl	Run Terminate	Load= 25pF	2.0	6.0	ns
TRun	Run Initiate	Load= 25pF	—	3.0	ns
TSMWr	Memory Write	Load= 25pF	2.0	9.0	ns
TSExc	Exception Valid	Load= 25pF	—	6.0	ns
TDMADis	DMA Drive On	Load= 25pF	3.0	15	ns
TDMAEn	DMA Drive Off	Load= 25pF	—	10	ns
Reset Initialization					
TRST	Reset Pulse Width		—	—	Tcyc
TRSTTAG	Reset Pulse Width, Pull-downs on Tag		—	—	µs
Capacitive Load Deration					
CLD	Load Derate(6)		—	—	ns/25pF

2873 tbl 12

NOTES:
1. All timings are referenced to 1.5V.
2. The clock parameters apply to all three 2xClocks: Clk2xSys, Clk2xSmp/Rd, and Clk2xPhi.
3. This parameter is guaranteed by design.
4. These parameters are illustrated in detail in the *IDT79R3001 Hardware Interface Guide.*
5. Tcyc is one CPU clock cycle (2 cycles of a 2x clock).
6. With the exception of $\overline{\text{Run}}$, no two signals on a given device will derate for a given load by a difference greater than 15%.

PIN CONFIGURATIONS

172-Pin Ceramic Flatpack (Cavity Side View)



NOTE:
1. AccTyp2 is redefined to be Parity Error if the parity enable option is selected at device initialization.

PIN CONFIGURATIONS (Continued)
144-Pin PGA (Top View)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	VCC14	AdrLo 6	AdrLo 10	AdrLo 11	VCC12	AdrLo 14	AdrLo 15	CpCond 0	AdrLo 16	AdrLo 17	Int(2)	Int(5)	Wr Busy	Reset	VCC10
B	AdrLo 3	Mem Wr	AdrLo 7	AdrLo 9	AdrLo 12	Cp Sync	AdrLo 13	CpCond 1	Int(1)	Int(3)	Cp Busy	Bus Error	Run	Tag13	Tag16
C	AdrLo 0	AdrLo 4	VCC13	AdrLo 5	AdrLo 8	GND13	GND12	VCC11	Int(0)	Int(4)	Rd Busy	GND	Tag14	Tag17	Tag20
D	Data 1	AdrLo 2	GND0										Tag15	Tag19	Tag21
E	DataP 0	Data 0	AdrLo 1										Tag18	Tag22	VCC9
F	VCC0	Data 7	Data 2										GND10	Tag23	Tag25
G	Data 4	Data 3	GND1										GND9	Tag24	Tag26
H	Data 6	Data 5	Data 8			IDT79R3001 RISController							VCC8	Tag28	Tag27
J	Data 10	DataP 1	Data 9										Tag31	Valid	Tag29
K	Data 15	Data 11	GND2										GND8	AdrLo 19	Tag30
L	VCC1	Data 12	Data 17										AdrLo 22	AdrLo 20	AdrLo 18
M	Data 13	Data 16	DataP 2										GND7	AdrLo 23	VCC7
N	Data 14	Data 18	Data 19	GND3	Data 24	DataP 3	VCC3	VCC4	GND5	GND6	Mem Rd	DMA Stall	DRd	IWr	AdrLo 21
P	Data 23	Data 20	AccTy1	Data 22	Data 26	Data 27	XEn	Data 30	Clk2x Sys	Clk2x Smp/Rd	DClk	Cp Cond3	AccTy0	IRd	DWr
Q	VCC2	Data 21	Data 25	Data 31	Data 28	GND4	Data 29	Excep-tion	Clk2x Phi	Cp Cond2	SysOut	VCC5	IClk	AccTy2	VCC6

2873 drw 18

NOTE:
1. AccTyp2 is redefined to be Parity Error if the parity enable option is selected at device initialization.

5

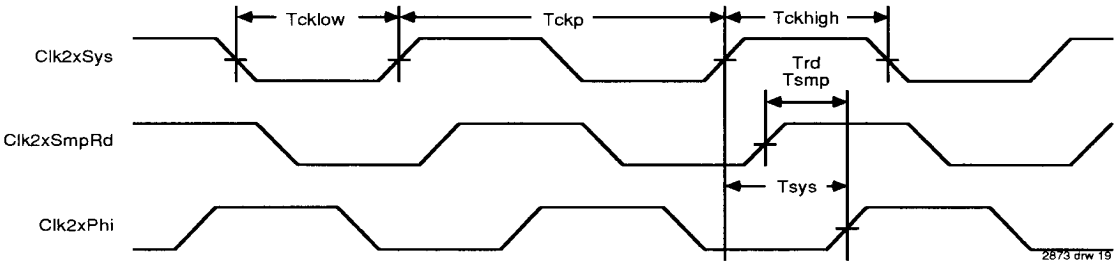


Figure 16. Input Clock Timing

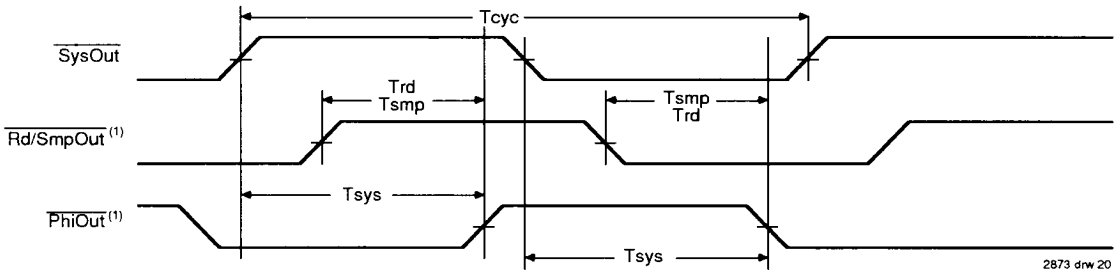


Figure 17. Processor Reference Clock Timing

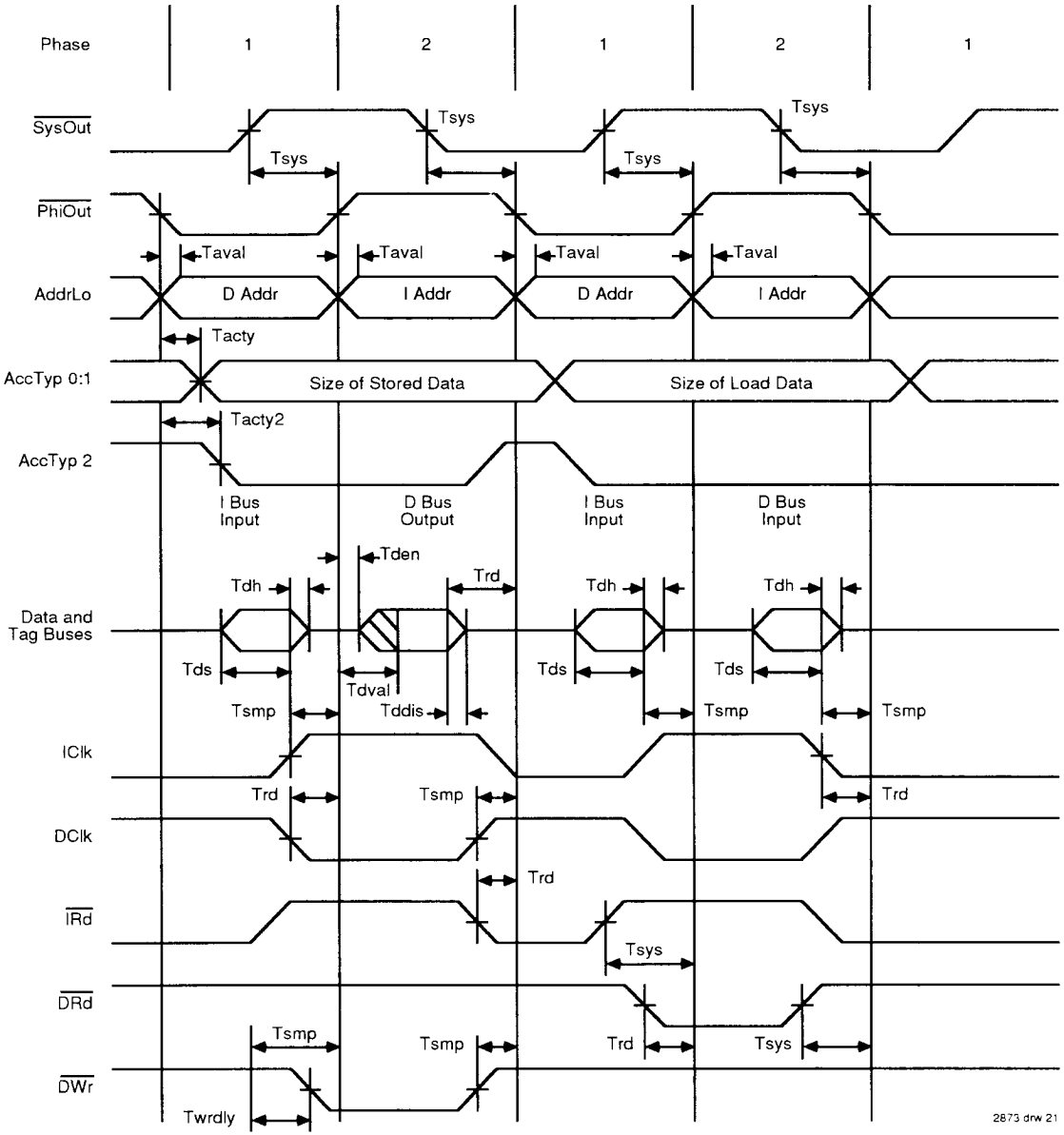


Figure 18. Synchronous Memory (Cache) Timing

5

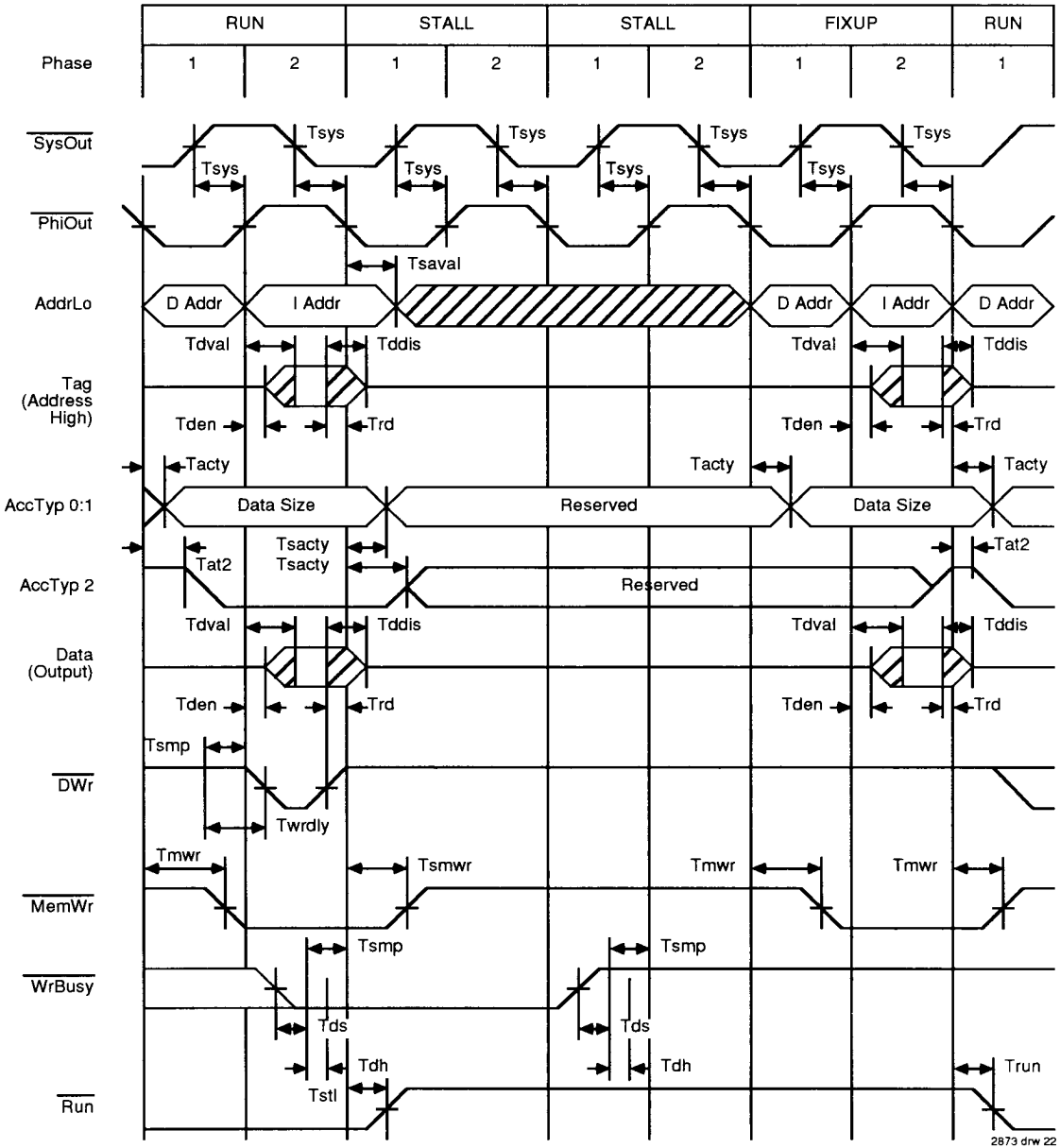


Figure 19. Memory Write Timing

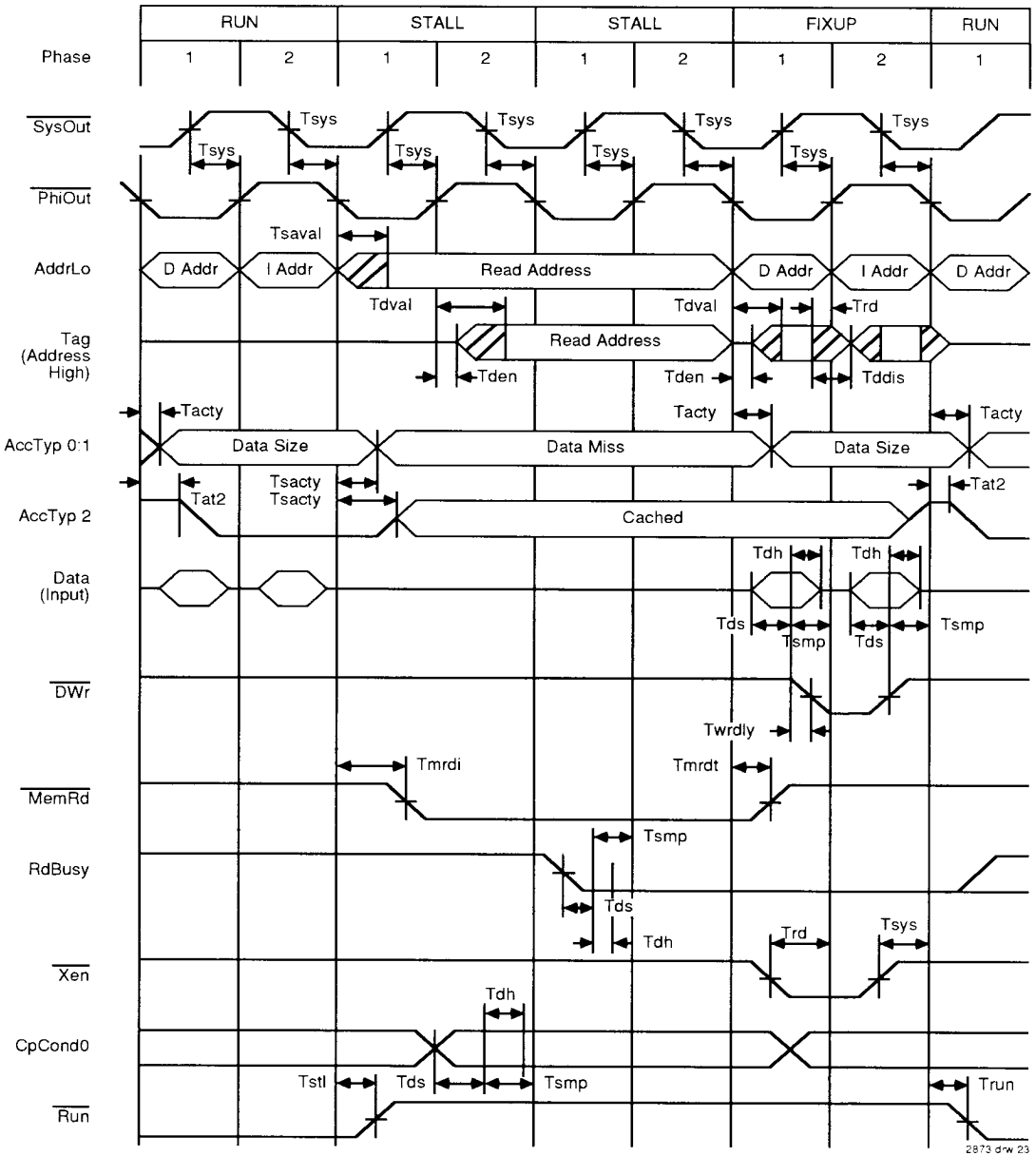


Figure 20. Memory Read Timing

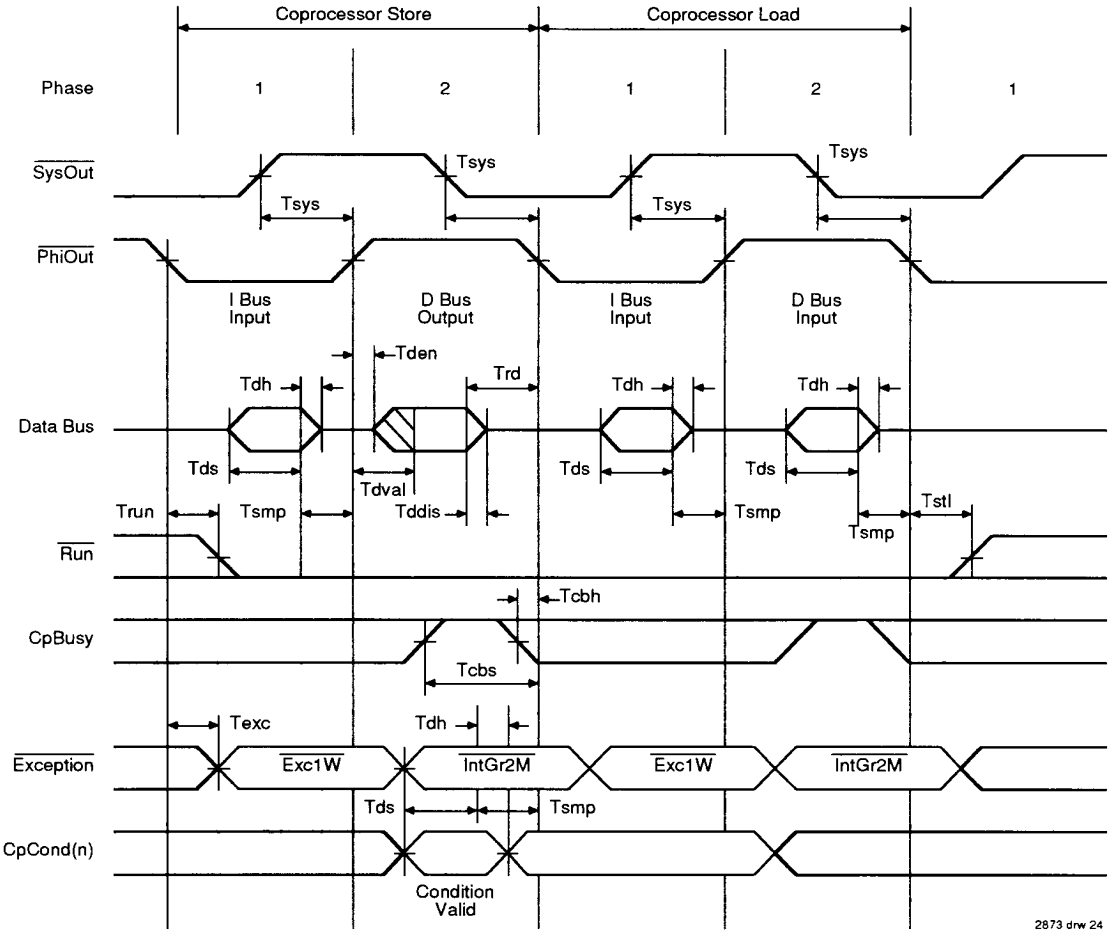


Figure 21. Co-Processor Load/Store Timing

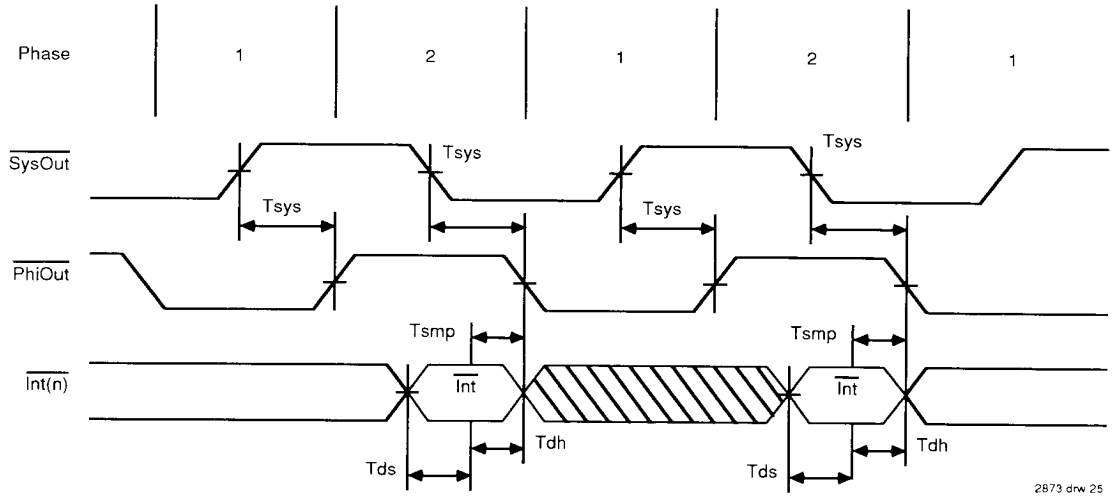


Figure 22. Interrupt Timing

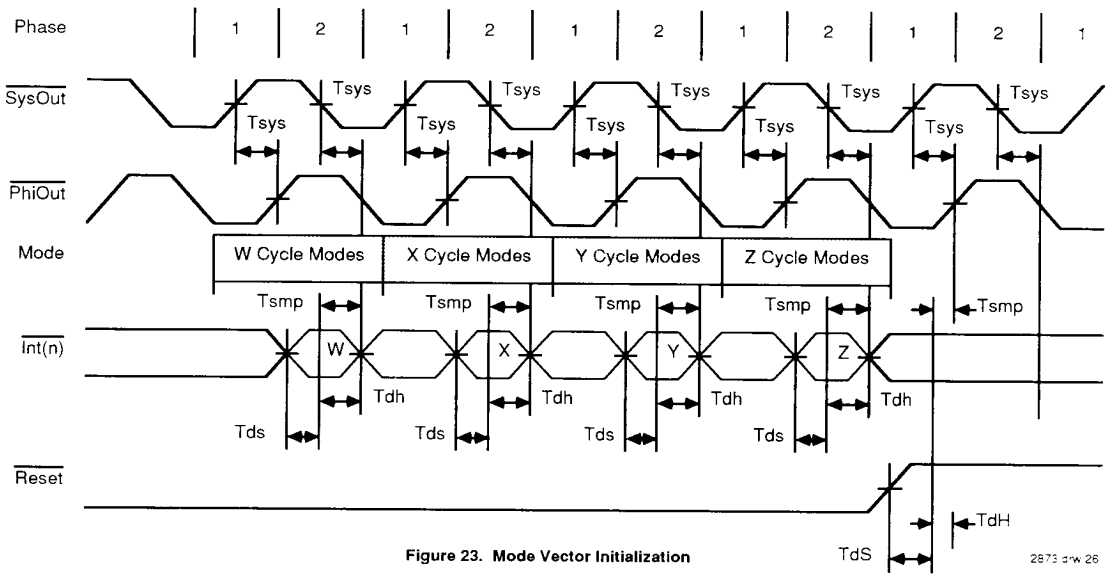


Figure 23. Mode Vector Initialization

NOTES:
1. Reset must be negated synchronously; however, it can be asserted asynchronously. Designs must not rely on the proper functioning of SysOut prior to the assertion of Reset.
2. If Phase-Lock On or is asserted as mode select options, they should be asserted throughout the Reset period, to insure that the slowest coprocessor in the system has sufficient time to lock to the CPU clocks.
3. Reset is actually sampled in both Phase 1 and Phase 2. To insure proper initialization, it must be negated relative to the end of Phase 1.

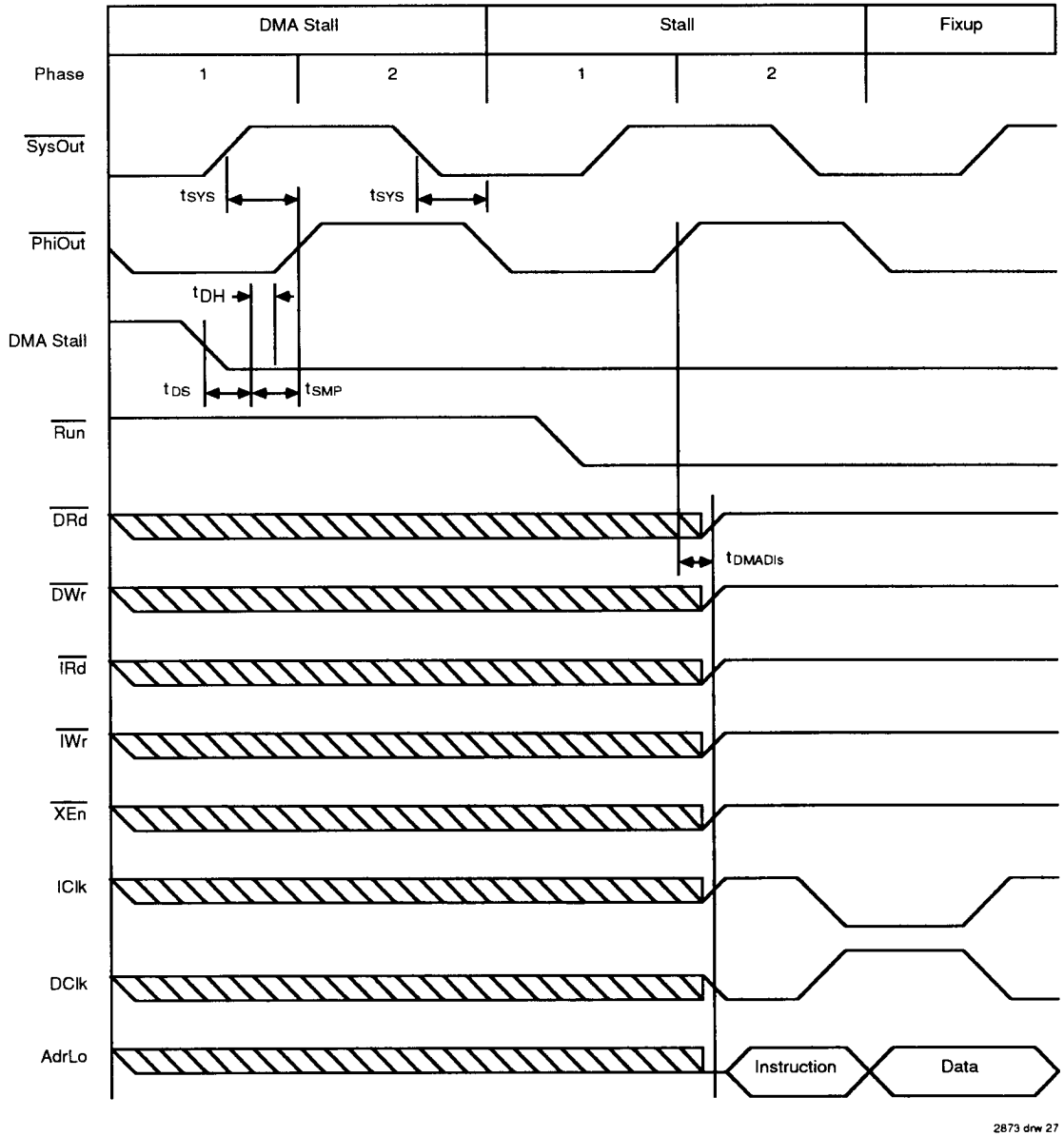


Figure 24. Entering DMA Stall

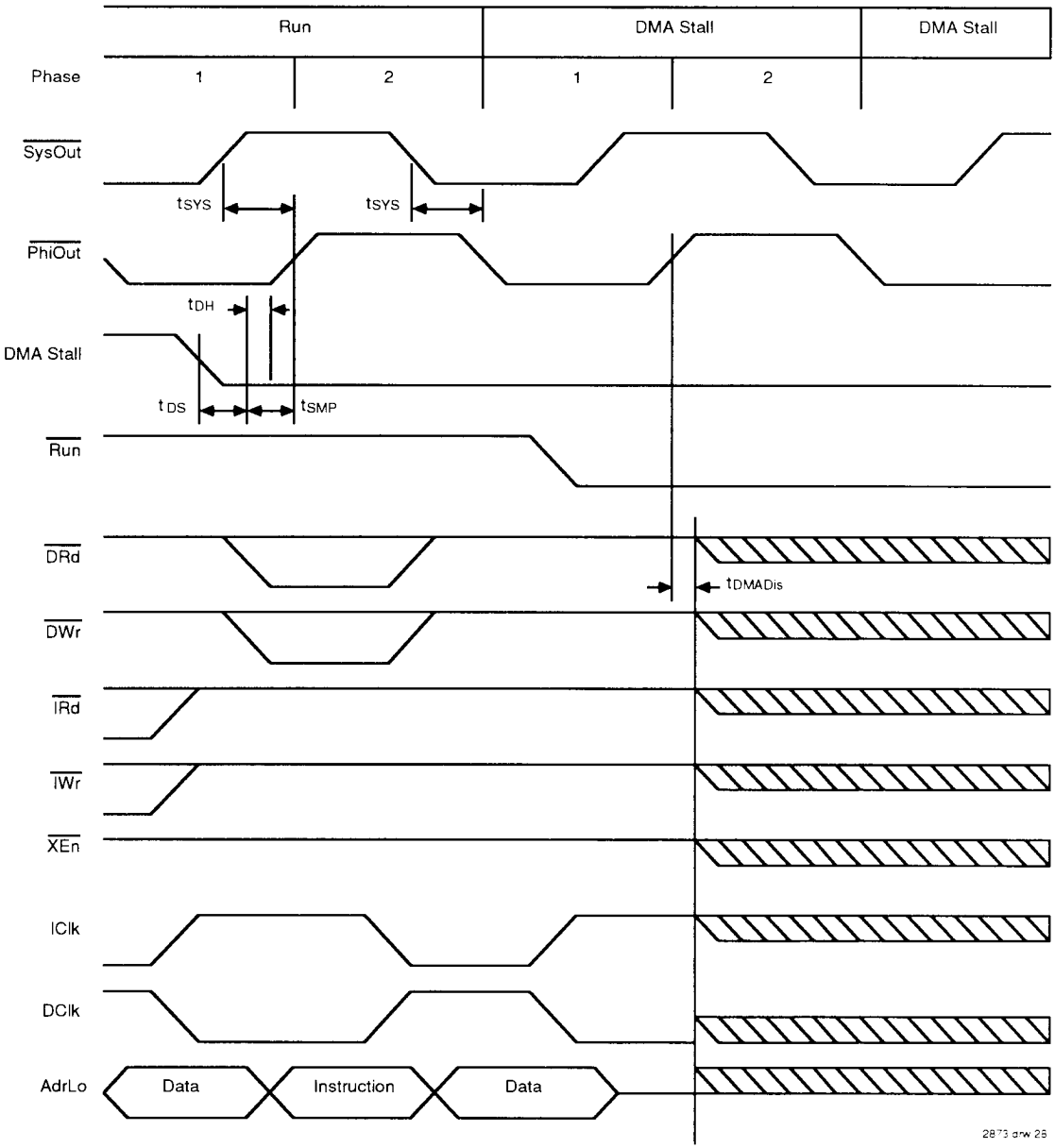
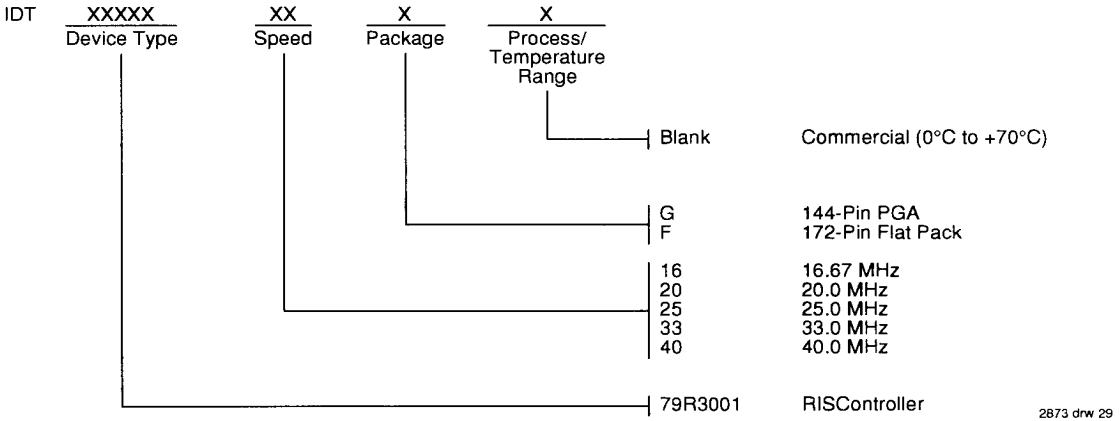


Figure 25. Completing DMA Stall

5

ORDERING INFORMATION



VALID COMBINATIONS

IDT	79R3001- 16,20,25,33	All Packages
	79R3001- 40	G