The IDT-C6 Processor *Architecture*

technical brief

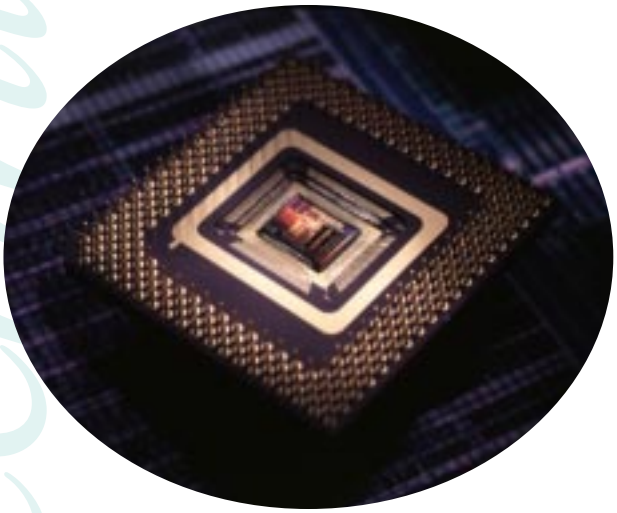
1. Introduction

Even though the IDT-C6™ processor is externally compatible with the Pentium processor, the internal architecture and design of the IDT-C6 processor is very different from that of the Pentium® processor and other contemporary x86 processors such as the AMD-K5™, AMD-K6™ and Cyrix 6x86™ processors. The IDT-C6 processor uses a unique design approach that provides significant benefits to the end-user.

This design approach provides high performance at low cost and low power using a relatively simple architecture that runs at high internal clock frequencies (MHz), includes large on-chip caches, and is extensively optimized for the target PC environment. The resulting IDT C6 processor is smaller (die size is only 88 mm^2 in 0.35 micron technology) and uses lower power than any contemporary x86 processor.

Philosophically, this is a return to the same basic concepts of RISC design that allowed microprocessor performance breakthroughs in the 1980's. Recently, however, contemporary x86 processors have followed a different path using very complex internal designs employing advanced architecture concepts such as superscalar execution, out-of-order instruction execution, reorder buffers, non-blocking caches, and so forth (these terms are all found in the data sheets of competitive products).

Unfortunately, while these advanced technical concepts make for good technical reading, the real bottom-line benefit that they provide to the end-user has been limited; especially when considering the resultant large chip sizes (resulting in high costs) and high power consumption. No such advanced technical hocus-pocus is to be found on an IDT-C6 processor — it merely offers compatibility with good performance, very low costs, and very low power consumption.



2. Key Concepts

The key concepts underlying the IDT-C6 processor design are:

- ***Simple instructions (load, store, branch, register-to-register) dominate instruction execution time.***
  This is the basic RISC design concept, which is also true in the x86 architecture: over 90% of instructions executed come from these basic categories. Of course, "simple" x86 instructions are much more complex than corresponding RISC architecture instructions.

  The IDT-C6 processor optimizes performance of these types of basic x86 instructions while minimizing the hardware provided for other little-used x86 functions. The little-used instructions are primarily implemented in microcode with minimal hardware support.

- ***Improving clock frequency has higher leverage than improving CPI.***
  The result of advanced computer design approaches over the last few years has been that the improvements in cycles-per-instruction (CPI) come at the expense of cycle-time improvements such that *total* performance (a linear function of frequency and CPI) of complex designs hasn't improved as fast as performance of simpler, yet high clock frequency, designs.

  The IDT-C6 processor optimizes *total system* performance by generally optimizing for highest clock frequency, even if this means reduced CPI for some functions versus more complex processor designs.

- ***Memory performance is the limiting CPI performance factor.***
  Due to the high ratio of internal clock speed versus the relatively limited PC processor-bus speed, off-chip memory-access performance is the primary factor in processor CPI performance (as opposed to internal instruction execution performance).

  The IDT-C6 processor addresses this phenomenon by providing very large on-chip caches that run at the high internal processor clock frequency. In addition, sophisticated TLB and cache management algorithms are included to further reduce bus activity.

- ***Optimize the design for the target user.***
  The IDT-C6 processor implements very specific and detailed design tradeoffs to provide high performance with low cost. Minimal hardware is provided for functions that are not heavily used or that are not critical to performance in the target environments (low-end desktop and mobile systems). These design optimizations are based on extensive detailed analysis of actual behavior of modern PC operating systems and applications.

- ***Small is beautiful.***
  The IDT-C6 processor is highly optimized for small physical size and few logic transistors. In addition to the obvious cost benefits, this small size provides secondary benefits of low power consumption and improved reliability.

3. Component Summary

3.1 General Architecture

Figure 1 illustrates the basic components of the IDT-C6 processor.

Fundamentally, the IDT-C6 processor's internal design is a relatively simple five-stage pipeline execution core with an additional instruction translation stage to translate x86 instructions coming from the fetch stage into the internal micro-instruction format.

Fetching and translating x86 instructions is asynchronous to the internal execution pipeline. Instructions are issued one at a time in program order. Instructions are executed and retired in order. Cache misses stall the pipeline until the data is available for the requesting instruction.

In spite of this very basic micro-architecture, the IDT-C6 processor achieves very high performance through several mechanisms:

- High internal clock frequency. The design is heavily optimized to provide high frequency.
- Good CPI on highly used instructions. The IDT-C6 processor implements specific design features to reduce the number of clocks for heavily used instructions — including complex functions such as protect-mode segment-register loads and string instructions.
- Large and fast on-chip caches and TLBs.
- Lots of fine-tuning and low-level optimizations. This includes such items as fast unaligned data access and fewer pipeline interlocks than the Pentium processor.

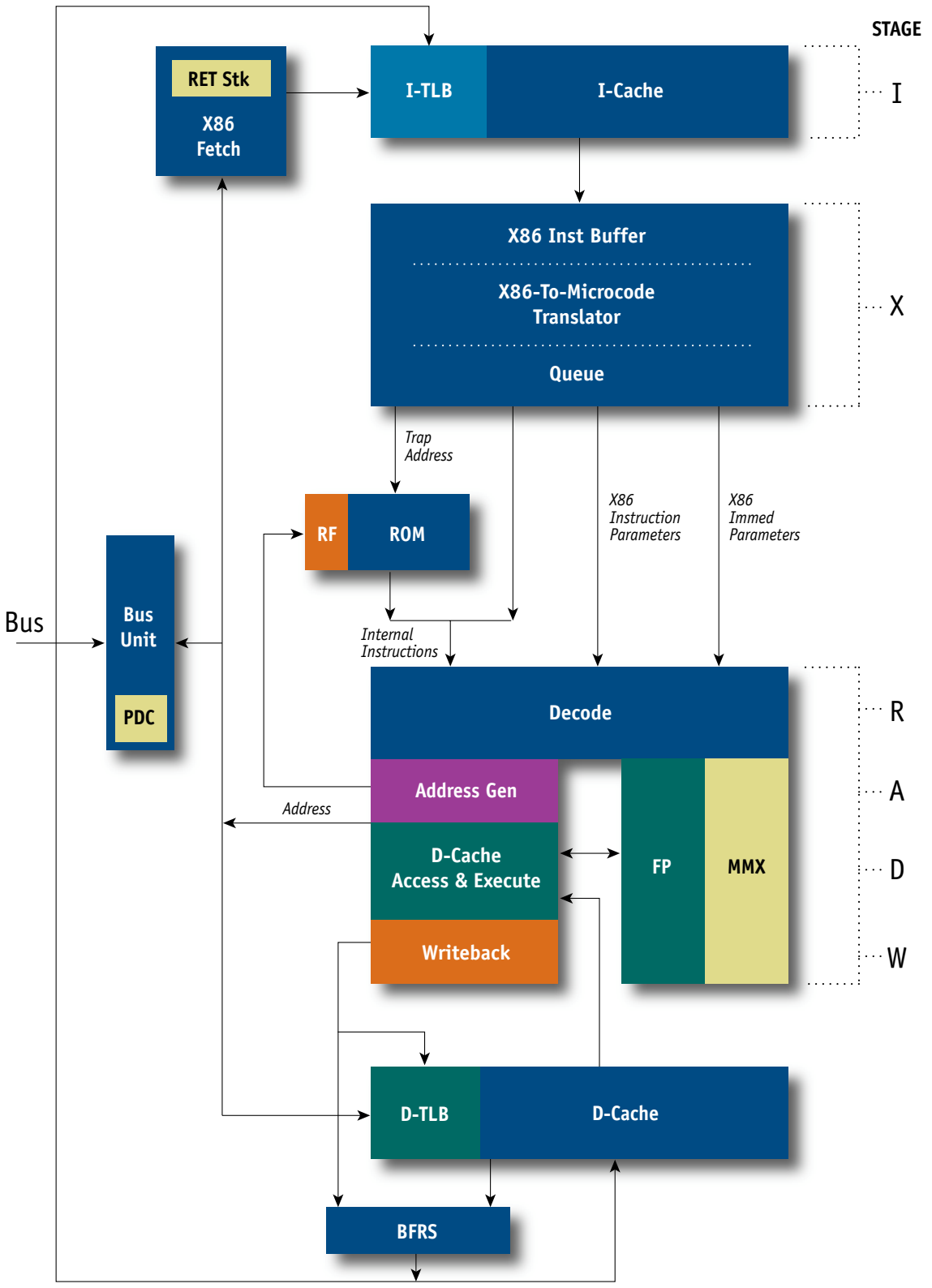


Figure 1. IDT-C6 Processor Data Flow

3.2 I-Cache

The I-cache contains 32 KB organized as two-way set associative with 32-byte lines. An LRU replacement algorithm is used. The associated I-TLB contains 64 entries organized as a 4-way set associative with a pseudo-LRU replacement algorithm. This large cache has a one-clock access time and operates at the high clock frequencies of the IDT-C6 processor.

The I-TLB utilizes an eight-entry unified page directory cache that significantly reduces the TLB miss penalty. In addition, the I-cache control logic includes several innovative features that minimize cache invalidates and unnecessary bus fetches.

As opposed to many other contemporary x86 processors, the data in the I-cache is exactly what came from the bus; that is, there are no "hidden" pre-decode bits. This facilitates the provision of large cache capacity in a small physical size.

In the mobile IDT-C6 processor, the I-cache is dynamically turned off when not used to reduce power requirements.

3.3 Translator Unit

The I-cache or bus unit delivers 16 or 8 bytes per clock to an x86 instruction buffer in the translator unit. The translator converts x86 instructions to internal instruction and data forms. Assuming that the instruction is in the x86 instruction buffer at the start of the cycle, the translator translates an entire x86 instruction in one clock. Instruction prefixes require an additional translator cycle for each prefix. However, due to the asynchronous fetch and "lookahead" capability of the translator, these extra cycles for prefixes rarely result in a bubble in the execution pipeline.

The output of the translator is: (1) the internal micro-instruction stream to perform the x86 instruction function, (2) the immediate data fields from the x86 instruction, and (3) various x86 state information used to control execution (for example, operand size). The internal instruction stream for an x86 instruction can consist of micro-instructions directly generated by the translator, or micro-instructions from the on-chip ROM, or both. For performance-sensitive instructions, there is no delay due to access of micro-code from ROM.

The microcode ROM capacity is larger than most x86 microcode ROMs to allow more unimportant (relative to performance) functions to be performed in microcode (versus in hardware), to allow extensive self-test microcode, and to allow extensive built-in debugging aids (for processor design debug).

Instruction fetch and translator operation is made asynchronous from micro-instruction execution via a three-entry translated-instruction queue between the translator and the execution unit. This queue allows the translator to "look-ahead" and continue translating x86 instructions even though the execution unit is stalled. The translator can also overlap generation of multiple internal instructions with translating prefixes on the subsequent instruction.

3.4 Execution Unit

Internal micro-instructions are executed in a tightly coupled four-stage pipeline that is very similar in structure to a basic RISC pipeline:

- ***Decode stage (R):*** Micro-instructions are decoded, register files are accessed, resource dependencies evaluated, and so forth.
- ***Addressing stage (A):*** Memory addresses are calculated and sent to the cache units. The IDT-C6 processor is capable of calculating most x86 instruction address forms in one clock; forms containing two registers require two clocks.
- ***Execute stage (D):*** ALU operations or load accesses to the D-cache are performed. All basic ALU functions take one clock except multiply and divide.
- ***Write-back stage (W):*** The results of operations are committed to the registers and store data is written to the D-cache or external write buffers.

Although the pipeline structure is similar to non-x86 processors, the micro-instructions and associated execution units are highly tuned to the x86 architecture. The micro-instructions closely resemble the corresponding x86 instructions Examples of specialized hardware features supporting the x86 architecture are: hardware handling of the x86 condition code, segment descriptor decode and manipulation instructions, hardware to automatically save the x86 floating-point environment, and so forth.

3.5 D-Cache

The D-cache is very similar to the I-cache: 32 KB organized as two-way set associative with 32-byte lines. An LRU replacement algorithm is used. The associated D-TLB contains 64 entries organized as 4-way set associative with a pseudo-LRU replacement algorithm. This large cache has a one-clock access time and is designed to operate at the high clock frequencies of the IDT-C6 processor. The D-TLB utilizes a large unified page directory cache (shared with the I-cache) which reduces the TLB miss penalty. In the mobile IDT-C6 processor, the D-cache is dynamically turned off when not used to reduce power requirements.

3.6 x86 Fetch Unit

One of the most visible frequency and size-versus-CPI tradeoffs made by the IDT-C6 processor is that there is limited branch prediction: only a special 8-entry Call/Return stack. This computer-science heresy helps support the high clock frequency and small die size but means that, *on average,* simple branches normally take more clocks on the IDT-C6 processor than on other contemporary x86 processors.

However, the clock difference is not as high as might be expected due to: (1) Return instructions are accurately predicted with the special Call/Return stack mechanism, (2) the total performance of complex branches (such as far calls) includes more than the actual branch time (which is overlapped with the additional operations on the IDT-C6 processor), and (3) the short IDT-C6 processor pipeline. Also, a clock on the IDT-C6 processor is a shorter period than on many of the processors that require fewer clocks per branch.

The most important consideration relative to the IDT-C6 processor's lack of branch prediction is that the gain from branch prediction is relatively small compared to the total time waiting on bus activity in high-frequency x86 processors. The IDT-C6 processor design focuses on using its transistors to minimize this bus wait time: large caches, large TLBs, smart cache management algorithms, and so forth, as opposed to a large branch prediction array.

3.7 Floating Point (FP) Unit

In addition to the basic integer execution unit, the IDT-C6 processor has a separate 80-bit floating-point execution unit that can execute floating-point instructions in parallel with integer instructions.

The floating-point unit is designed to maximize clock frequency and to minimize chip size while providing adequate levels of floating-point performance for typical desktop use. Some floating-point instructions are pipe-lined, but some are only partially pipe-lined.

The IDT-C6 processor issues only one instruction per clock but most integer instructions and most floating-point instructions can execute in parallel.

In the mobile IDT-C6 processor, the floating-point unit is dynamically turned off when not used to reduce power requirements.

3.8 MMX™ Instruction Unit

The IDT-C6 processor contains a separate execution unit for the new MMX-compatible instructions. The MMX architecture registers are the same as the floating-point registers, otherwise the MMX execution unit has its own adder, multiplier and shifter separate from the floating-point unit.

In the mobile IDT-C6 processor, the MMX unit is dynamically turned off when not used to reduce power requirements.

3.9 Bus Unit

The IDT-C6 processor bus unit provides an external bus interface compatible with the Pentium processor. In addition to the expected bus control functions, the bus unit implements a large page-directory cache to reduce the impact of TLB misses as well as several special optimizations intended to reduce cache misses.

Four 64-bit write buffers allow internal execution to proceed overlapped with waiting for external stores to complete.

Integrated Device Technology
2975 Stender Way
P.O. Box 58015
Santa Clara, CA 95054-3090

Tel: (408)727-6116
Tel: (408)492-8674

Web Site: www.centtech.com

Fax-On-Demand: 800-9-IDT-Fax (US)
408-492-8391 (Outside US)