Case Study of Pentium 4

INTRODUCTION

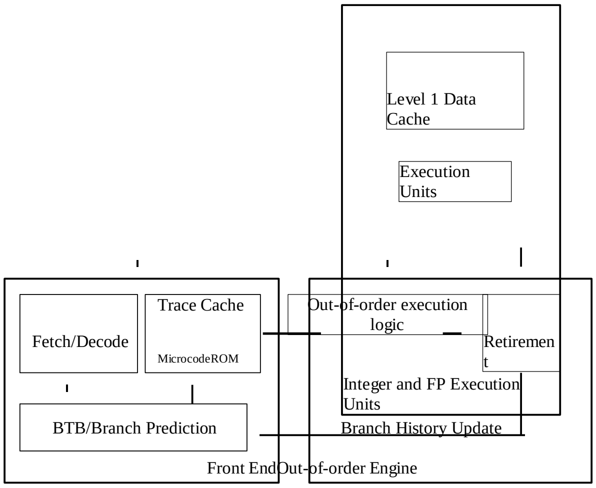
The Pentium 4 processor is Intel's new flagship microprocessor that was introduced at 1.5GHz in November of 2000. It implements the new Intel NetBurst microarchitecture that features significantly higher clock rates and world-class performance. It includes several important new features and innovations that will allow the Intel Pentium 4 processor to deliver industry-leading performance for the next several years. This paper provides an in-depth examination of the features and functions of the Intel NetBurst microarchitecture.

The Pentium 4 processor is designed to deliver performance across applications where end users can truly appreciate and experience its performance. For example, it allows a much better user experience in areas such as Internet audio and streaming video, image processing, video content creation, speech recognition, 3D applications and games, multi-media, and multi-tasking user environments. The Pentium 4 processor enables real- time MPEG2 video encoding and near real-time MPEG4 encoding, allowing efficient video editing and video conferencing. It delivers world-class performance on 3D applications and games, such as Quake 3*, enabling a new level of realism and visual quality to 3D applications.

The Pentium 4 processor has 42 million transistors implemented on Intel's 0.18u CMOS process, with six levels of aluminum interconnect. It has a die size of 217 mm^2 and it consumes 55 watts of power at 1.5GHz. Its 3.2 GB/second system bus helps provide the high data bandwidths needed to supply data to today's and tomorrow's demanding applications. It adds 144 new 128-bit Single Instruction Multiple Data (SIMD) instructions called SSE2 (Streaming SIMD Extension 2) that improve performance for multi-media, content creation, scientific, and engineering applications.

OVERVIEW OF THE NETBURST MICROARCHITECTURE

A fast processor requires balancing and tuning of many microarchitectural features that compete for processor die cost and for design and validation efforts. Figure 1 shows the basic Intel NetBurst microarchitecture of the Pentium 4 processor. As you can see, there are four main sections: the in-order front end, the out-of-order execution engine, the integer and floating-point execution units, and the memory subsystem.



Basic Block Diagram

In-Order Front End

The in-order front end is the part of the machine that fetches the instructions to be executed next in the program and prepares them to be used later in the machine pipeline. Its job is to supply a high-bandwidth stream of decoded instructions to the out-of-order execution core, which will do the actual completion of the instructions. The front end has highly accurate branch prediction logic that uses the past history of program execution to speculate where the program is going to execute next. The predicted instruction address, from this front-end branch prediction logic, is used to fetch instruction bytes from the Level 2 (L2) cache. These IA-32 instruction bytes are then decoded into basic operations called uops (micro-operations) that the execution core is able to execute.

The NetBurst microarchitecture has an advanced form of a Level 1 (L1) instruction cache called the Execution Trace Cache. Unlike conventional instruction caches, the Trace Cache sits between the instruction decode logic and the execution core as shown in Figure 1. In this location the Trace Cache is able to store the already decoded IA-32 instructions or uops. Storing already decoded instructions removes the IA-32 decoding from the main execution loop. Typically the instructions are decoded once and placed in the Trace Cache and then used repeatedly from there like a normal instruction cache on previous machines. The IA-32 instruction decoder is only used when the machine misses the Trace Cache and needs to go to the L2 cache to get and decode new IA-32 instruction bytes.

Out-of-Order Execution Logic

The out-of-order execution engine is where the instructions are prepared for execution. The out-of-order execution logic has several buffers that it uses to smooth and re-order the flow of instructions to optimize performance as they go down the pipeline and get scheduled for execution. Instructions are aggressively re- ordered to allow them to execute as quickly as their input operands are ready. This out-of-order execution allows instructions in the program following delayed instructions to proceed around them as long as they do not depend on those delayed instructions. Out-of-order execution allows the execution resources such as the ALUs and the cache to be kept as busy as possible executing independent instructions that are ready to execute.

The retirement logic is what reorders the instructions, executed in an out-of-order manner, back to the original program order. This retirement logic receives the completion status of the executed instructions from the execution units and processes the results so that the proper architectural state is committed (or retired) according to the program order. The Pentium 4 processor can retire up to three uops per clock cycle. This retirement logic ensures that exceptions occur only if the operation causing the exception is the oldest, non-retired operation in the machine. This logic also reports branch history information to the branch predictors at the front end of the machine so they can train with the latest known-good branch-history information.

Integer and Floating-Point Execution Units

The execution units are where the instructions are actually executed. This section includes the register files that store the integer and floating-point data operand values that the instructions need to execute. The execution units include several types of integer and floating-point execution units that compute the results and also the L1 data cache that is used for most load and store operations.

Memory Subsystem

Figure 1 also shows the memory subsystem. This includes the L2 cache and the system bus. The L2 cache stores both instructions and data that cannot fit in the Execution Trace Cache and the L1 data cache. The external system bus is connected to the backside of the second-level cache and is used to access main memory when the L2 cache has a cache miss, and to access the system I/O resources.

CLOCK RATES

Processor microarchitectures can be pipelined to different degrees. The degree of pipelining is a microarchitectural decision. The final frequency of a specific processor pipeline on a given silicon process technology depends heavily on how deeply the processor is pipelined. When designing a new processor, a key design decision is the target design frequency of operation. The frequency target determines how many gates of logic can be included per pipeline stage in the design. This then helps determine how many pipeline stages there are in the machine.

There are tradeoffs when designing for higher clock rates. Higher clock rates need deeper pipelines so the efficiency at the same clock rate goes down. Deeper pipelines make many things take more clock cycles, such as mispredicted branches and cache misses, but usually more than make up for the lower per-clock efficiency by allowing the design to run at a much higher clock rate. For example, a 50% increase in frequency might buy only a 30% increase in net performance, but this frequency increase still provides a significant overall performance increase. High-frequency design also depends heavily on circuit design techniques, design methodology, design tools, silicon process technology, power and thermal constraints, etc. At higher frequencies, clock skew and jitter and latch delay become a much bigger percentage of the clock cycle, reducing the percentage of the clock cycle usable by actual logic. The deeper pipelines make the machine more complicated and require it to have deeper buffering to cover the longer pipelines.

CONCLUSION

The Pentium 4 processor is a new, state-of-the-art processor microarchitecture and design. It is the beginning of a new family of processors that utilize the new Intel NetBurst microarchitecture. Its deeply pipelined design delivers world-leading frequencies and performance. It uses many novel microarchitectural ideas including a Trace Cache, double-clocked ALU, new low- latency L1 data cache algorithms, and a new high bandwidth system bus. It delivers world-class performance in the areas where added performance makes a difference including media rich environments (video, sound, and speech), 3D applications, workstation applications, and content creation.

Intel Pentium 4

Made By:
Madhvi Mittal

Intel Pentium 4

- P4 General Introduction
- Chip Layout
- Micro-Architecture: NetBurst
- Memory Subsystem: Cache Hierarchy
- Branch Prediction
- Pipeline
- Hyper-Threading
- Conclusions

Intel Pentium 4: Introduction

- The Pentium 4 processor is Intel's new microprocessor that was introduced in November of 2000
- The Pentium 4 processor
  - Has 42 million transistors implemented on Intel's 0.18 CMOS process, with six levels of aluminum interconnect
  - Has a die size of 217 mm^2
  - Consumes 55 watts of power at 1.5 GHz
  - 3.2 GB/second system bus helps provide the high data bandwidths needed to supply data for demanding applications
  - Implements a new Intel NetBurst microarchitecture

Pentium 4 Chip Layout

- 400 MHz System Bus
- Advanced Transfer Cache
- Hyper Pipelined Technology
- Enhanced Floating Point/Multi-Media
- Execution Trace Cache
- Rapid Execution Engine
- Advanced Dynamic Execution

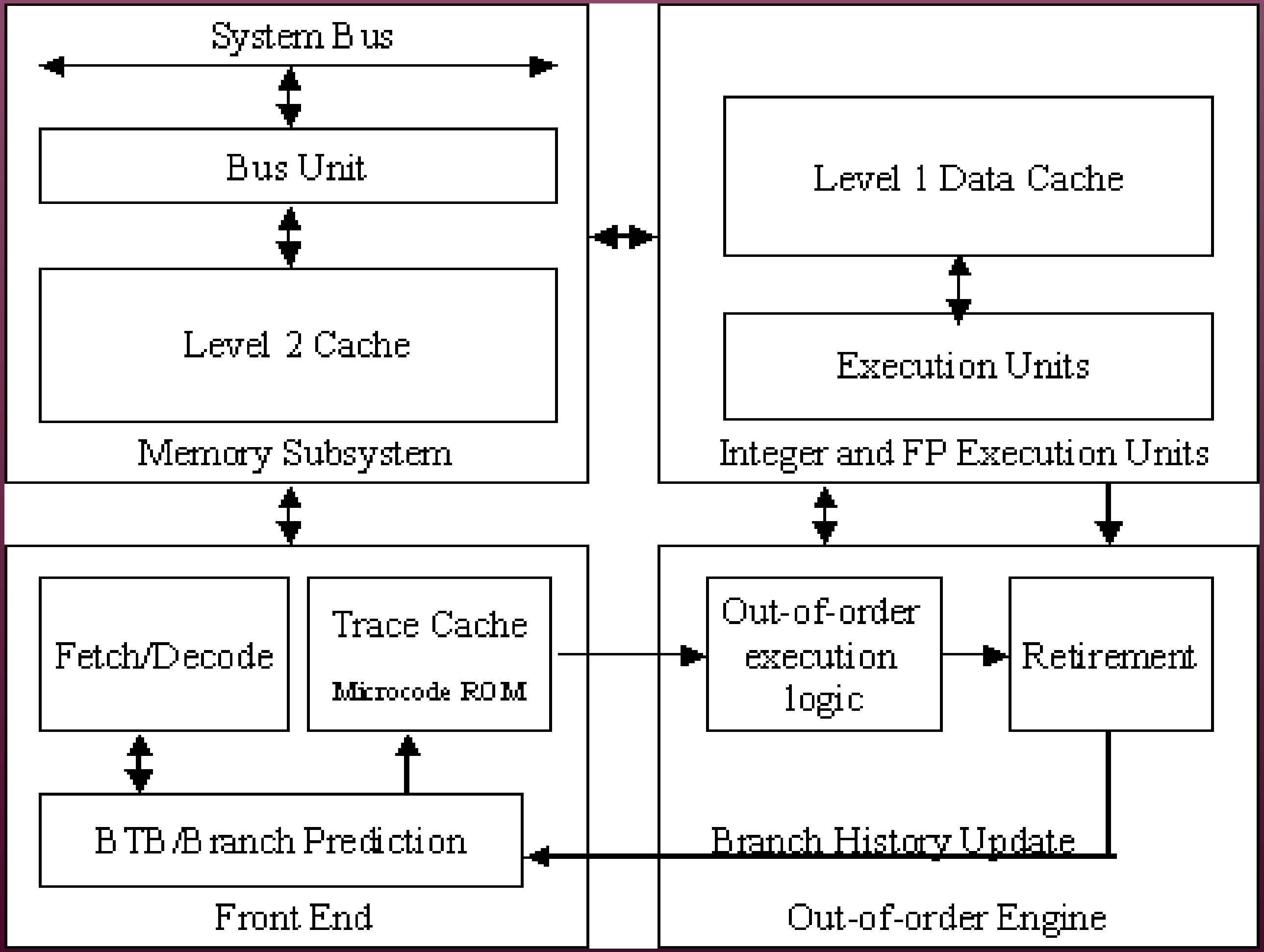
Intel NetBurst Microarchitecture Overview

- Designed to achieve high performance for integer and floating point computations at high clock rates
- Features:
  - hyper-pipelined technology that enables high clock rates and frequency headroom (up to 10 GHz)
  - a high-performance, quad-pumped bus interface to the Intel NetBurst microarchitecture system bus
  - a rapid execution engine to reduce the latency of basic integer instructions
  - out-of-order speculative execution to enable parallelism
  - superscalar issue to enable parallelism

Intel NetBurst Microarchitecture Overview

- Features:
  - Hardware register renaming to avoid register name space limitations
  - Cache line sizes of 64 bytes
  - Hardware pre-fetch
  - A pipeline that optimizes for the common case of frequently executed instructions
  - Employment of techniques to hide stall penalties such as parallel execution, buffering, and speculation

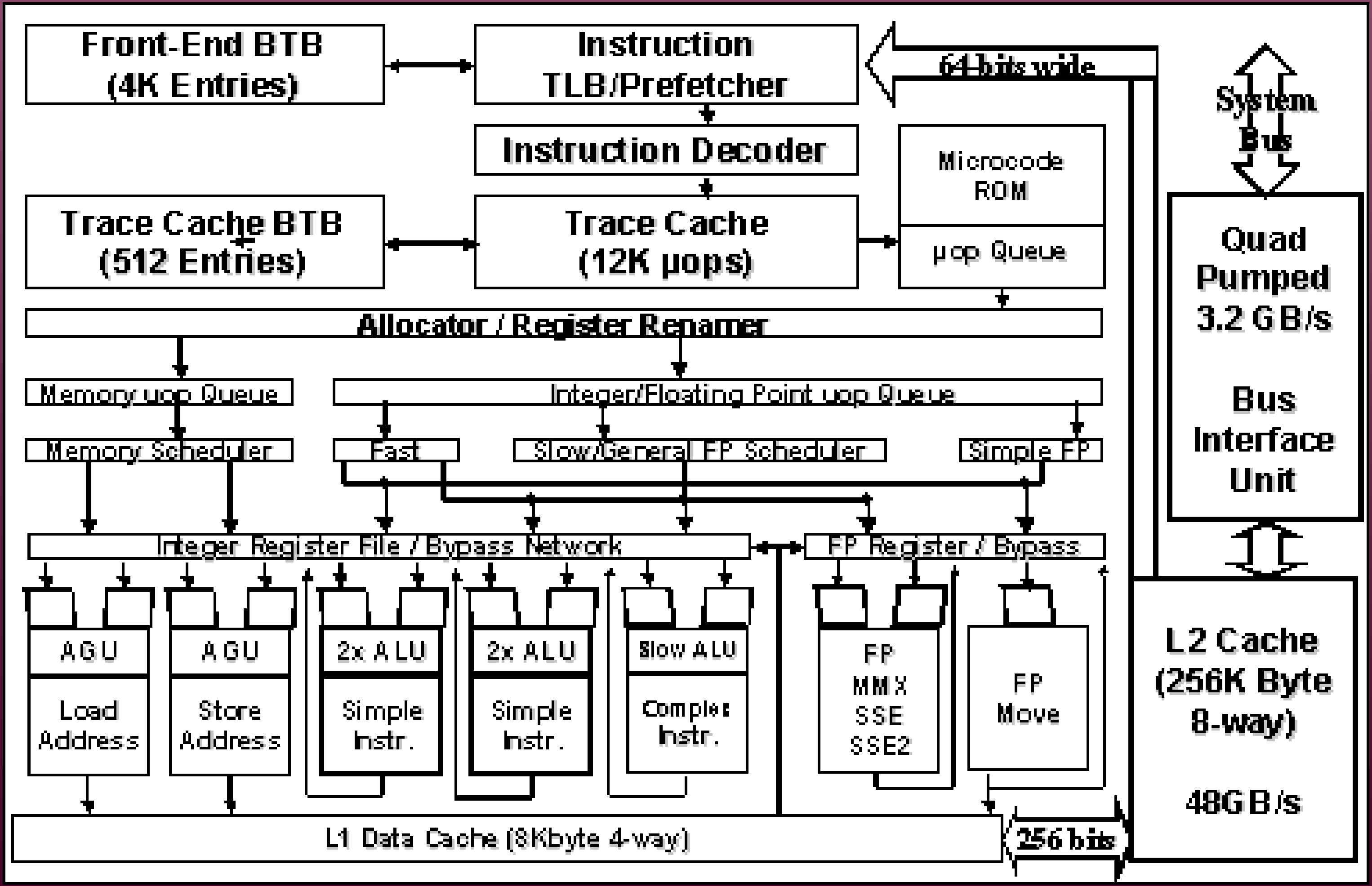
Pentium 4 Basic Block Diagram:



Pentium 4 Basic Block Diagram Description

- Four main sections:
  - The In-Order Front End
  - The Out-Of-Order Execution Engine
  - The Integer and Floating-Point Execution Units
  - The Memory Subsystem

Intel NetBurst Microarchitecture in Detail:



Front-End BTB (4K Entries)

Instruction TLB/Prefetcher

64 bits wide

System Bus

Instruction Decoder

Microcode ROM

Trace Cache BTB (512 Entries)

Trace Cache (12K µops)

µop Queue

Quad Pumped 3.2 GB/s

Bus Interface Unit

Allocator / Register Renamer

Memory uop Queue

Integer/Floating Point uop Queue

Memory Scheduler

Fast

Slow/General FP Scheduler

Simple FP

Integer Register File / Bypass Network

FP Register / Bypass

AGU Load Address

AGU Store Address

2x ALU Simple Instr.

2x ALU Simple Instr.

Slow ALU Complex Instr.

FP MMX SSE SSE2

FP Move

L2 Cache (256K Byte 8-way)

48GB/s

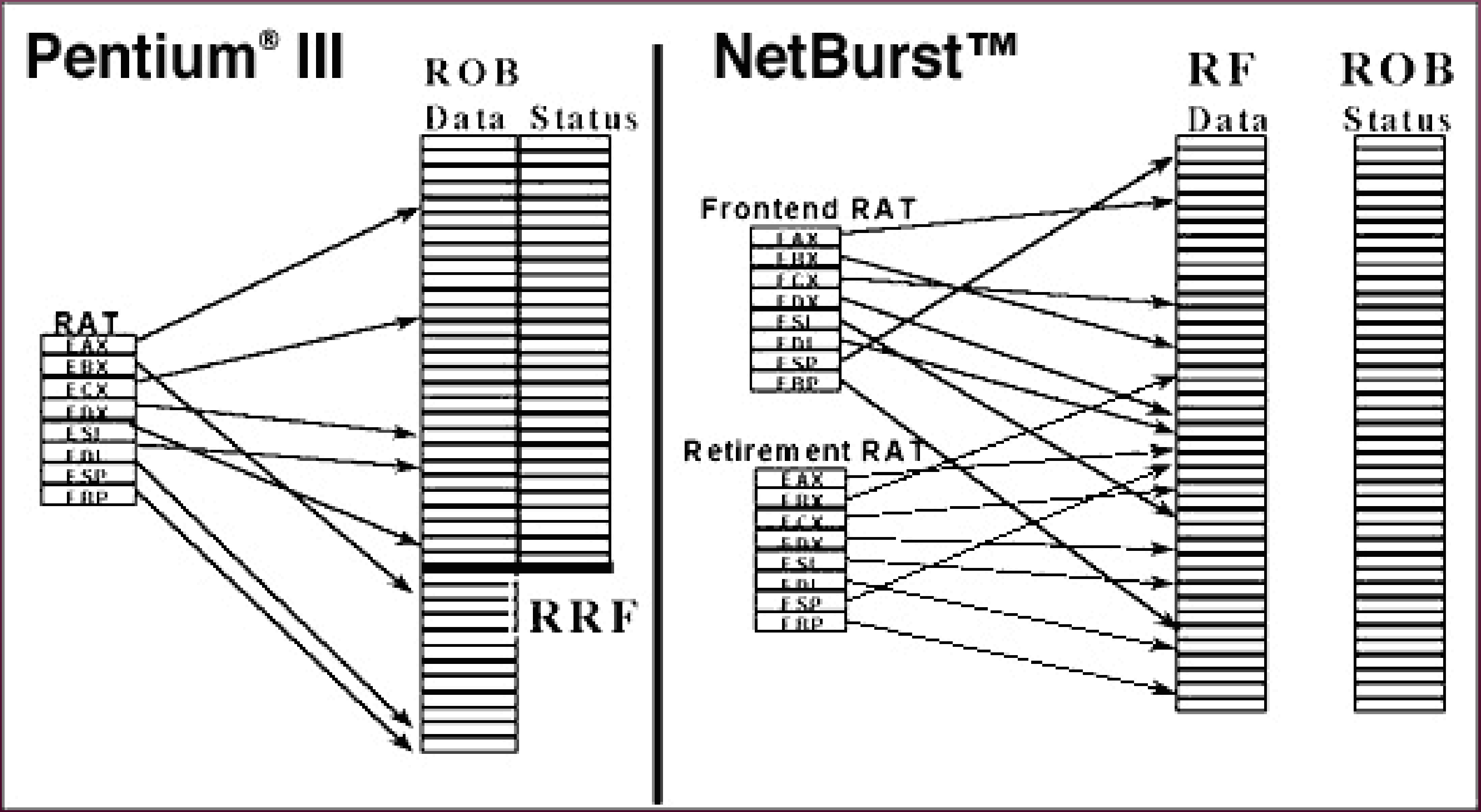
L1 Data Cache (8Kbyte 4-way)

256 bits

Instruction TLB/Prefetcher

- The Instruction TLB/Pre-fetcher translates the linear instruction pointer addresses given to it into physical addresses needed to access the L2 cache, and performs page-level protection checking
- Intel NetBurst microarchitecture supports three pre-fetching mechanisms:
  - A hardware instruction fetcher that automatically pre-fetches instructions
  - A hardware mechanism that automatically fetches data and instructions into the unified L2 cache
  - A mechanism fetches data only and includes two components:
    - A hardware mechanism to fetch the adjacent cache line within an 128-byte sector that contains the data needed due to a cache line miss
    - A software controlled mechanism that fetches data into the caches using the pre-fetch instructions

Out-of-Order Execution Engine Register Renaming Logic

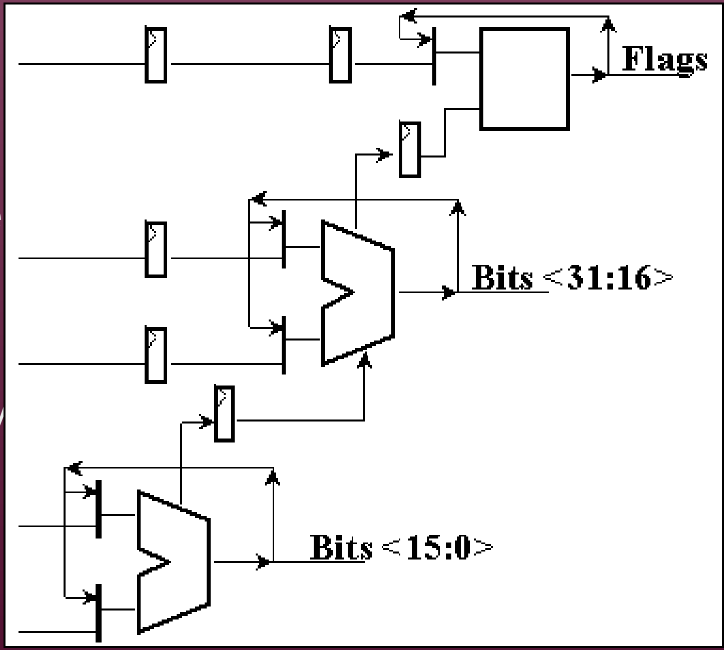


Integer and Floating-Point Execution Units

- Consists of:
  - Execution units
  - Level 1 (L1) data cache
- Execution units are where the instructions are executed
  - Units used to execute integer operations:
    - Low-latency integer ALU
    - Complex integer instruction unit
    - Load and store address generation units
  - Floating-point/SSE execution units
    - FP Adder
    - FP Multiplier
    - FP Divide
    - Shuffle/Unpack
- L1 data cache is used for most load and store operations

Low Latency Integer ALU Staggered Add

- ALU operations are performed in a sequence of three fast clock cycles (the fast clock runs at 2x the main clock rate)
  - **First fast clock cycle** - The low order 16-bits are computed and are immediately available to feed the low 16-bits of a dependent operation the very next fast clock cycle
  - **Second fast clock cycle** - The high-order 16 bits are processed, using the carry out just generated by the low 16-bit operation
  - **Third fast clock cycle** - The ALU flags are processed
- Staggered add means that only a 16-bit adder and its input muxes need to be completed in a fast clock cycle



Integer and Floating-Point Execution Units: Complex Integer Operations

- Integer operations that are more complex go to separate hardware for completion
- Integer shift or rotate operations go to the complex integer dispatch port
- Shift operations have a latency of four clocks
- Integer multiply and divide operations have a latency of about 14 and 60 clocks, respectively.

Pentium 4: Memory Subsystem

- The Pentium 4 processor has a highly capable memory subsystem to enable the high-bandwidth stream-oriented applications such as 3D, video, and content creation
- This subsystem consists of:
  - Level 2 (L2) Unified Cache
  - 400 MHz System Bus
- L2 cache stores instructions and data that cannot fit in the Trace Cache and L1 data cache
- System bus is used to access main memory when L2 cache has a cache miss, and to access the system I/O resources
  - System bus bandwidth is 3.2 GB per second
  - Uses a source-synchronous protocol that quad-pumps the 100 MHz bus to give 400 million data transfers per second
  - Has a split-transaction, deeply pipelined protocol to provide high memory bandwidths in a real system
  - Bus protocol has a 64-byte access length

Cache Hierarchy: Trace Cache

- Level 1 Execution Trace Cache is the primary or L1 instruction cache
- Most frequently executed instructions in a program come from the Trace Cache
- Only when there is a Trace Cache miss fetching and decoding instructions are performed from L2 cache
- Trace Cache has a capacity to hold up to 12K μops in the order of program execution
- Performance is increased by removing the decoder from the main execution loop
- Usage of the cache storage space is more efficient since instructions that are branched around are not stored

Branch Prediction

- 2 Branch Prediction Units present on the Pentium 4
  - Front End Unit – 4KB Entries
  - Trace Cache – 512 Entries
- Allows the processor to begin execution of instructions before the actual outcome of the branch is known
- The Pentium 4 has an advanced branch predictor. It is comprised of three different components:
  - Static Predictor
  - Branch Target Buffer
  - Return Stack
- Branch delay penalty for a correctly predicted branch can be as few as zero clock cycles.
- However, the penalty can be as many as the pipeline depth.
- Also, the predictor allows a branch and its target to coexist in a signal trace cache line. Thus maximizing instruction delivery from the front end

Pentium 4 Pipeline Overview

- The Pentium 4 has a 20 stage pipeline
- This deep pipeline increases
  - Performance of the processor
  - Frequency of the clock
  - Scalability of the processor
- Also, it provides
  - High Clock Rates
  - Frequency headroom to above 1GHz

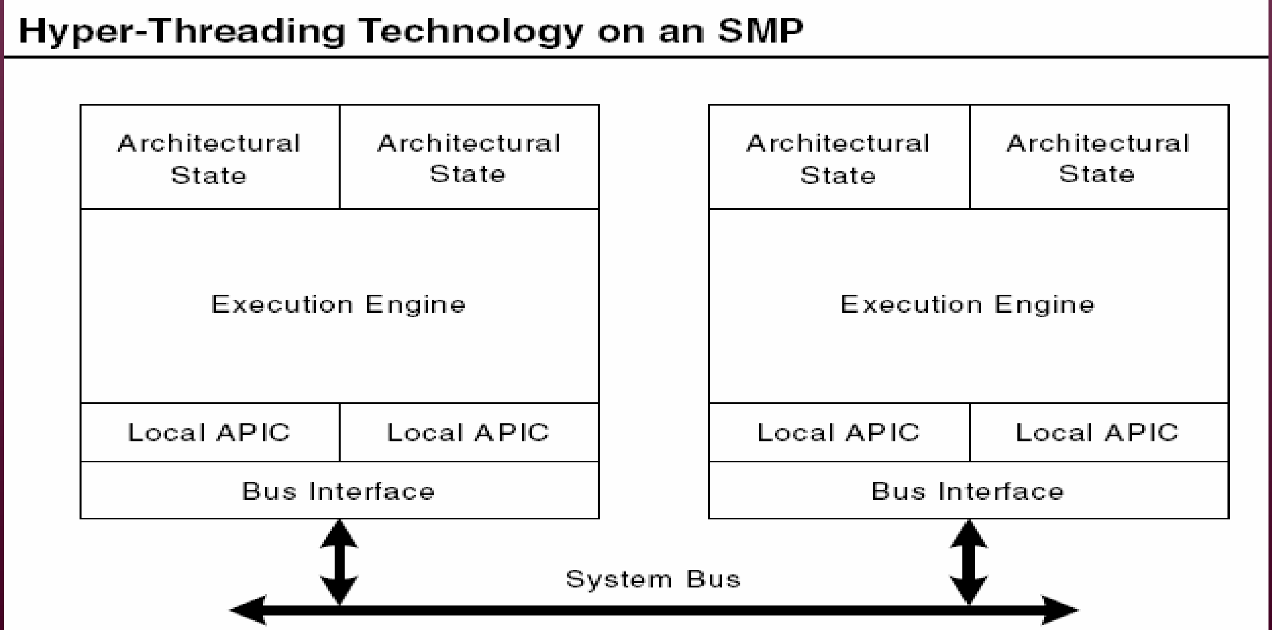
Pipeline Stage Names

- TC Nxt IP
- TC Fetch
- Drive
- Allocate
- Rename
- Que
- Schedule

- Dispatch
- Retire
- Execution
- Flags
- Branch Check

Pentium 4 Hyper-Threading Technology

- Enables software to take advantage of both task-level and thread-level parallelism by providing multiple logical processors within a physical processor package.



Hyper-Threading Basics

- Two logical units in one processor
  - Each one contains a full set of architectural registers
  - But, they both share one physical processor’s resources
- Appears to software (including operating systems and application code) as having two processors.
- Provides a boost in throughput in actual multiprocessor machines.
- Each of the two logical processors can execute one software thread.
  - Allows for two threads (max) to be executed simultaneously on one physical processor

Hyper-Threading Resources

- Replicated Resources
  - Architectural State is replicated for each logical processor. The state registers control program behavior as well as store data.
    - General Purpose Registers (8)
    - Control Registers
    - Machine State Registers
    - Debug Registers
  - Instruction pointers and register renaming tables are replicated to track execution and state changes.
  - Return Stack is replicated to improve branch prediction of return instructions
  - Finally, Buffers were replicated to reduce complexity

Instructions Set

- Pentium 4 instructions divided into the following groups:
  - General-purpose instructions
  - x87 Floating Point Unit (FPU) instructions
  - x87 FPU and SIMD state management instructions
  - Intel (MMX) technology instructions
  - Streaming SIMD Extensions (SSE) extensions instructions
  - SSE2 extensions instructions
  - SSE3 extensions instructions
  - System instructions

Instruction Set: MMX Instructions

- MMX is a Pentium microprocessor that is designed to run faster when playing multimedia applications
- The MMX technology consists of three improvements over the non-MMX Pentium microprocessor:
  - 57 new microprocessor instructions have been added to handle video, audio, and graphical data more efficiently
  - Single Instruction Multiple Data (SIMD), makes it possible for one instruction to perform the same operation on multiple data items
  - The memory cache on the microprocessor has increased to 32KB, meaning fewer accesses to memory that is off chip
- MMX instructions operate on packet byte, word, double-word, or quad-word integer operands contained in either memory, MMX registers, and/or in general-purpose registers