Optimizing Floating Point Calculations, I

Michael A. Saum

msaum@math.utk.edu

Department of Mathematics

University of Tennessee, Knoxville

Overview

- Introduction
- Computer Basics
- Intel® Pentium® 4 Hardware Archictecture
- Memory Hierarchy
- Cache Basics
- Measuring Program Performance

Introduction

- Focus on writing computer programs in `C`, `FORTRAN`
- UNIX (Linux) platform
- Intel® Pentium® Hardware Platform
- While above seem very specific, concepts and techniques are applicable to most programming environments on most hardware platforms.

Computer Basics

- The Five Classic Components of a Computer:
  - Input
  - Output
  - Memory
  - Datapath
  - Control
- Datapath and Control are the domain of the *Central Processing Unit* (CPU) or Processor.
- The CPU runs at a specified *clock rate* which relates to how fast the hardware can perform basic functions. Pentium 4 speeds 3 - 4 GHz.
- Memory access speed typically ranges 400 - 800 MHz.

Computer Basics, contd.

- Associated with each CPU is an *instruction set* which define what actions the CPU can take.
- A CPU can be classified as either a *Reduced Instruction Set Computer (RISC)* or a *Complex Instruction Set Computer (CISC)*.
- A RISC computer can recognize and execute only a small number of instructions, usually < 128.
- A CISC computer has a much larger set of instructions it can recognize and execute.
- Modern Intel ® CPU's are CISC in principle but utilize RISC concepts in implementation.

Computer Basics, contd.

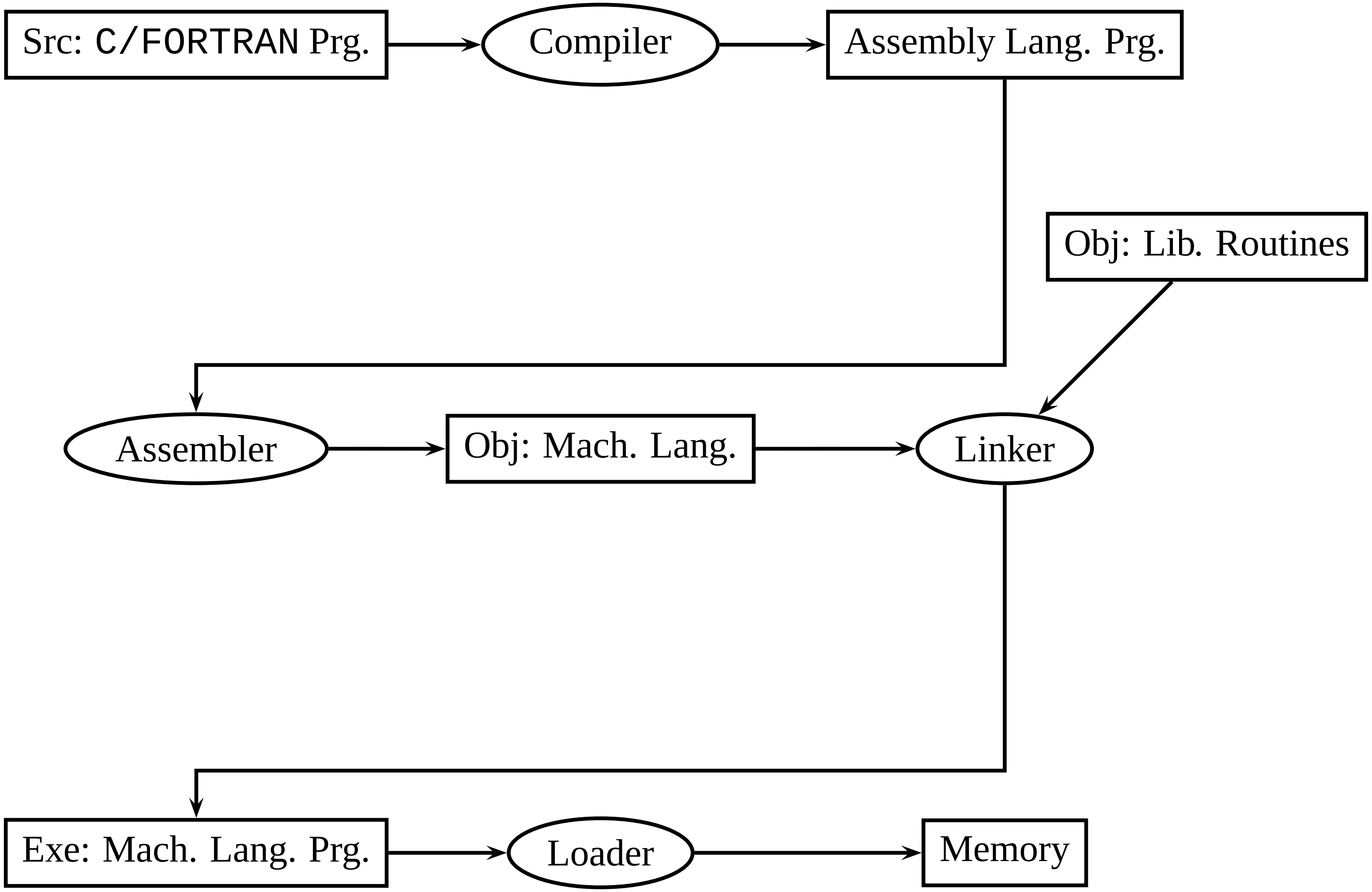
- Today's modern CPU's consist of a number of different components:
  - Fetch/Decode Unit - Fetches instructions to be executed and Decodes into instructions which can be processed.
  - Arithmetic Logical Unit (ALU) - consists of specialized processor functions for dealing with operations involving integer data, floating point data, etc.
  - Write Unit - Handles output to memory.
- *Pipelining* is a technique in which multiple instructions are overlapped in execution.

Computer Basics, contd.

- The bottom line is that a CPU acts on a stream of ones and zeros which it iterprets to perform specific operations.
- A *executable program* is simply this binary stream.
- In order to allow one to write a program of any complexity, we have developed a method to translate commands from a *high level language* to *assembly language* to *machine language* which is processed by the CPU.
- This process of translation is called the *Compilation Process*.

The Compilation Process

High level program must be translated into machine language and loaded into memory to run.



Binary Basics

- All CPU's have located on the chip a small number of storage locations called *registers*.
- These registers are described by how many bits they contain.
- A 32 bit processor implies that the CPU acts mostly on registers containing 32 bits.
- A byte is 8 bits.
- A 32 bit address space implies that there are 2^{32} uniquely addressable bytes.

Intel® CPU History

1978 8086 arch., extension of 8 bit microprocessor 8080. 16 bit arch., internal registers 16 bits wide.

1980 8087 Floating Point Coprocessor extends 8086 instruction set with $\sim$ 60 floating point instructions. Stack based instead of register based.

1982 80286 increases address space to 24 bits, elaborate memory mapping model.

1985 80386 extends address space to 32 bits, more flexible register usage.

1989 80486.

1992 Pentium.

1995 Pentium Pro.

1997 MMX extensions. 57 new instructions using floating point stack.

1999 Pentium III. SSE (Streaming Single Instruction Multiple Data [SIMD]) extensions. 70 new instructions. Added 8 separate registers, doubled width to 128 bits, SP packed data type implies that 4 32 bit FP Ops can be done in parallel. Also includes cache prefetch and streaming store instructions which bypass cache and write directly to memory.

Intel® CPU History, contd.

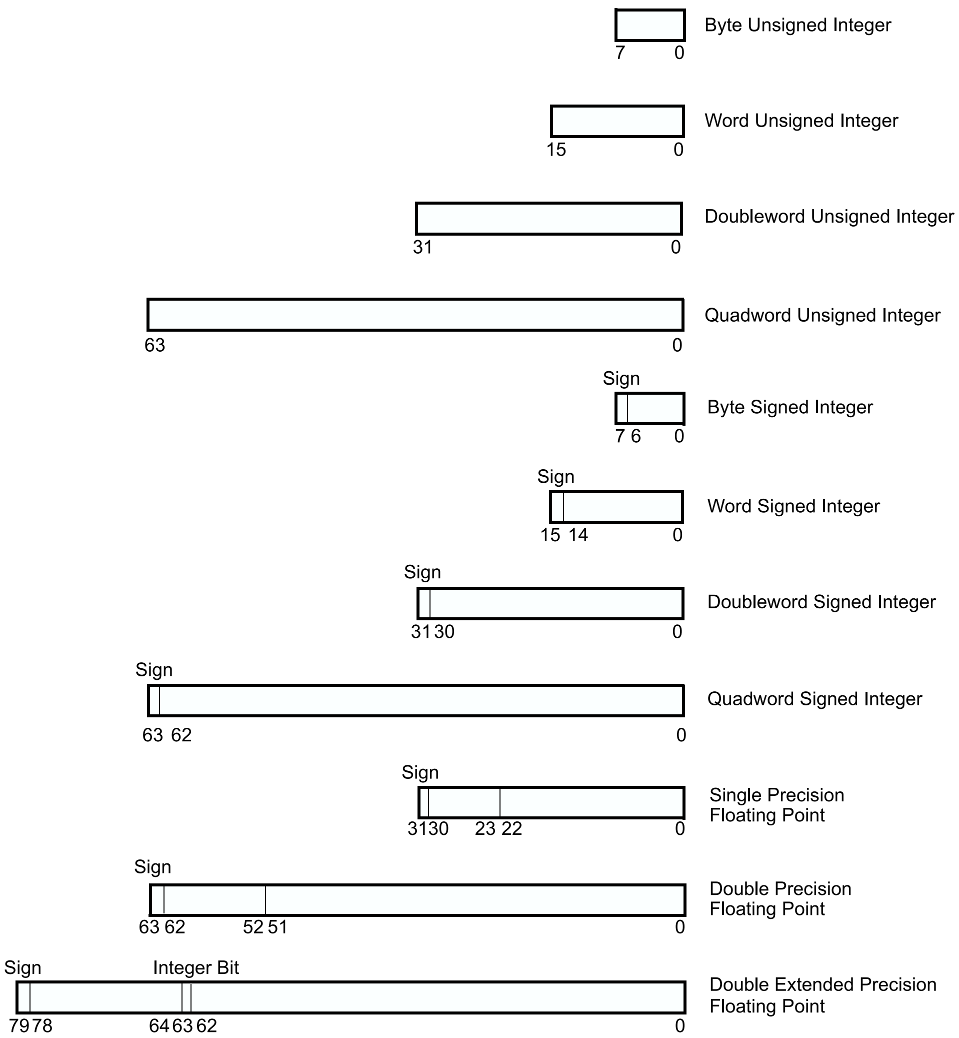
2001 Pentium 4 with SSE2 extensions. 144 new instructions. DP packed data type implies that 2 64 bit FP Ops can be done in parallel. Compilers can choose to use 8 SSE registers as FP registers.

2003 AMD64 extends address space from 32 to 64 bits. Increases number of registers to 16 and 128 bit registers to 16.

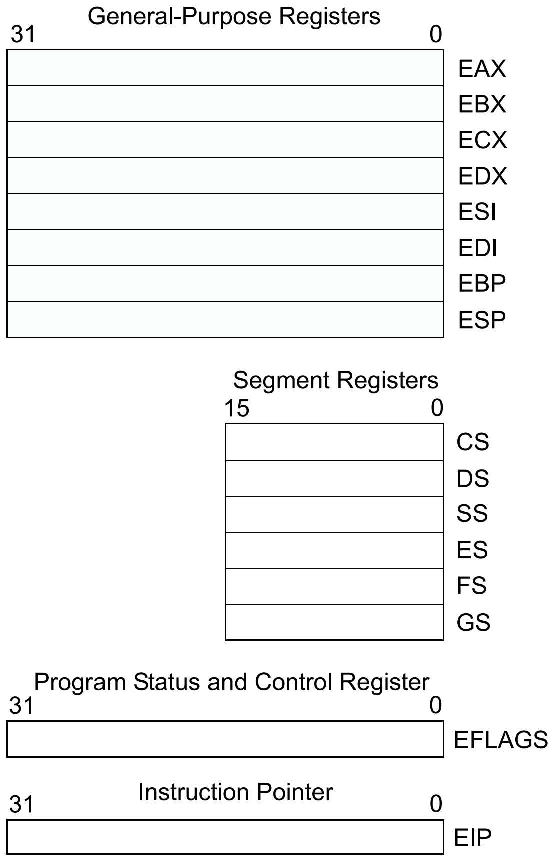
2004 Intel embraces AMD64 extensions producing EM64T. SSE3 adds 13 instructions to support complex arithmetic.

IA-32 is commonly used to describe Intel Pentium 4 CPU's. These processors have 8 general purpose registers (GPR) each 32 bits wide. Note that this is the same since 80386 introduced in 1985.

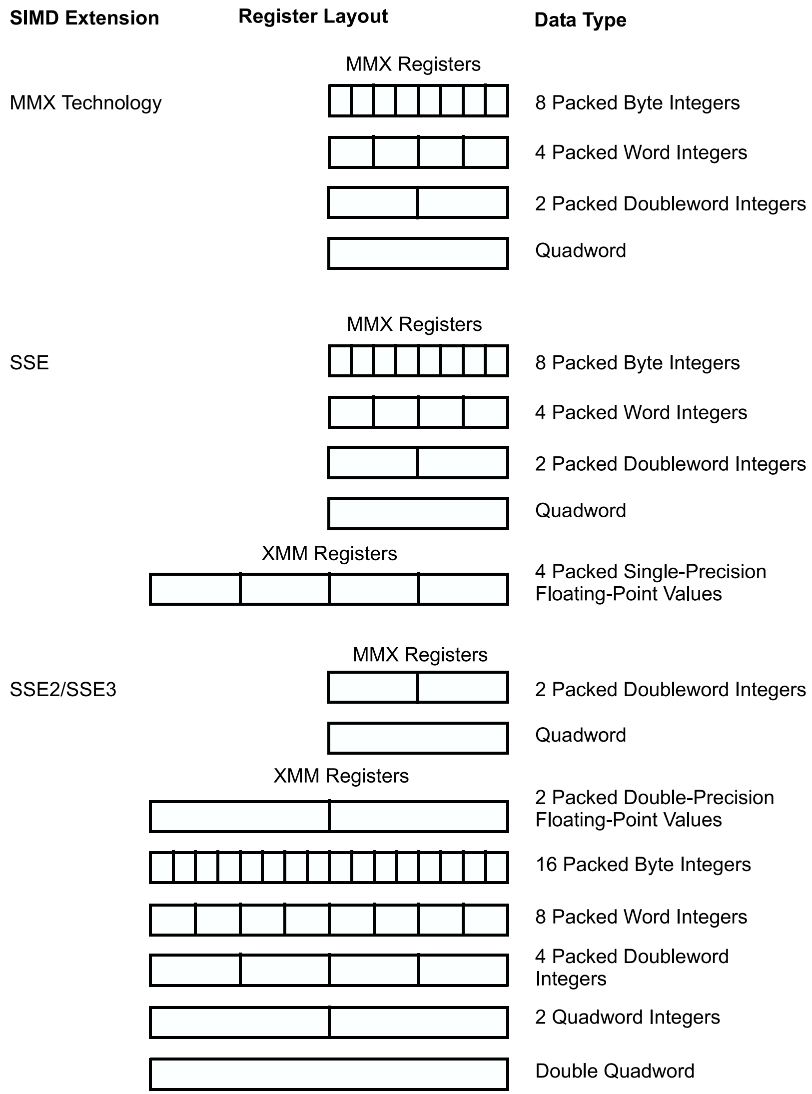
Numeric Datatypes



General Purpose Registers



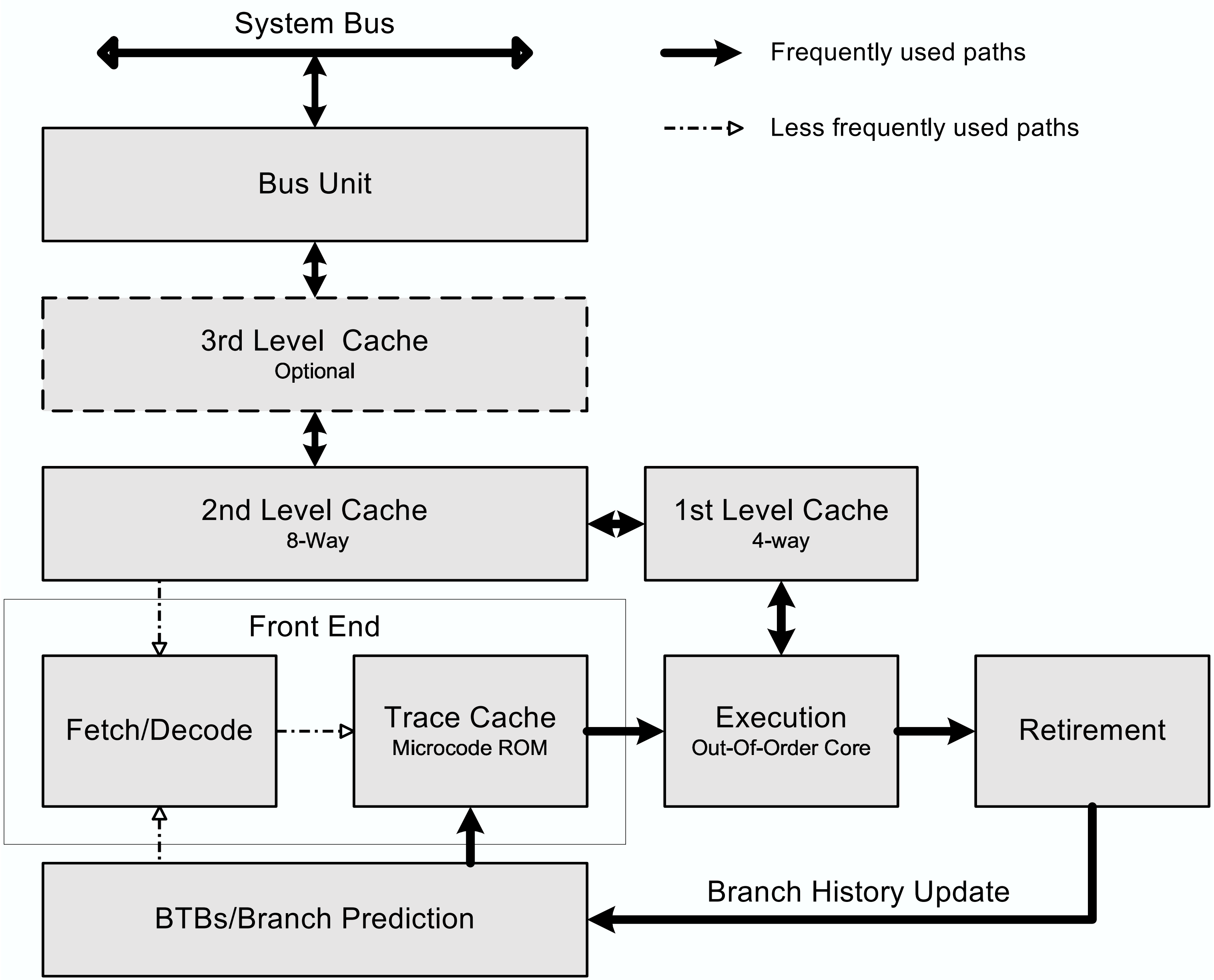
SIMD Registers



IA-32 Register Summary

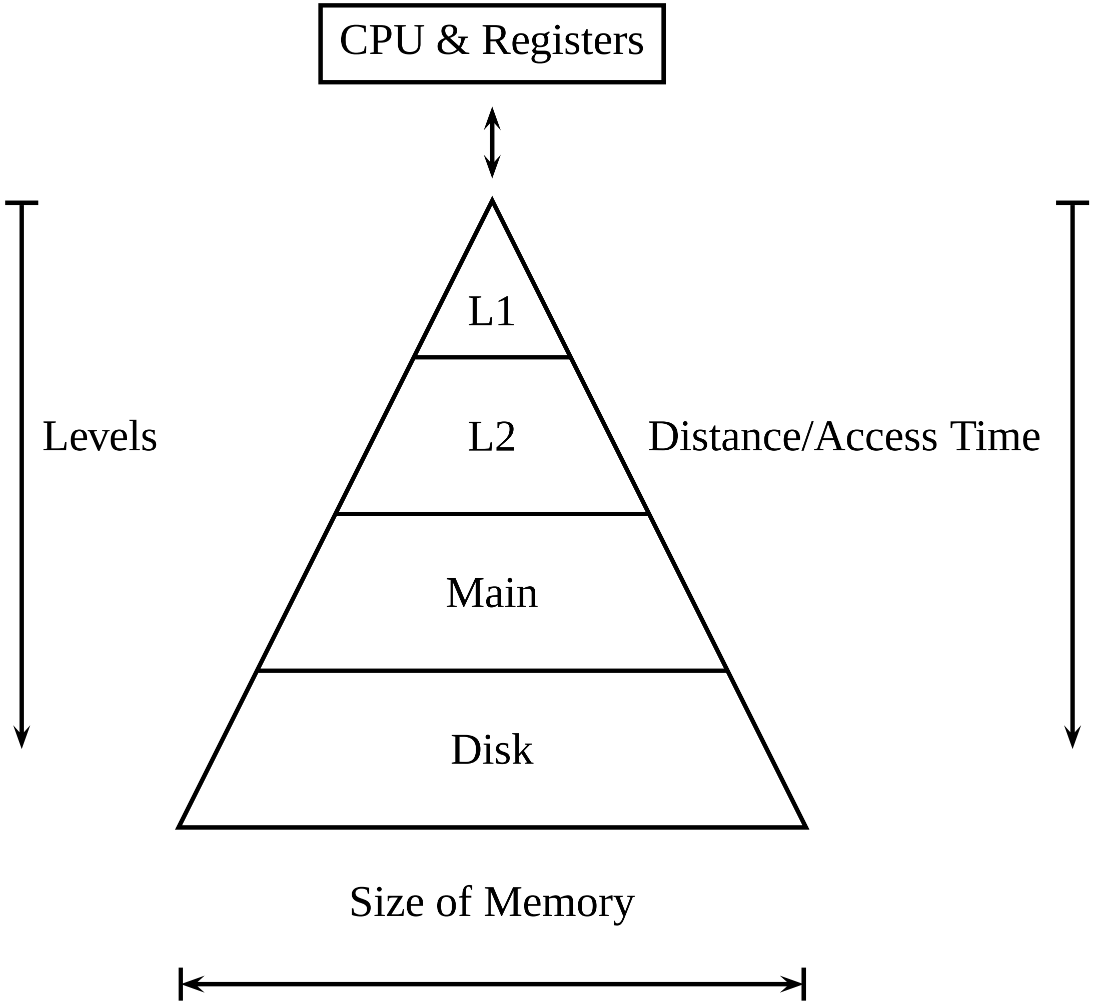
- Linear address space 32 bit (4 GB), physical address space 36 bit (64 GB)
- 8 GPR, 6 segment registers (addressing), EFLAGS register (control), EIP register (instruction pointer)
- 8 x87 FPU data registers, misc FPU control registers
- 8 MMX registers for SIMD operations on 64-bit packed byte, word, doubleword integers
- 8 XMM data registers for SIMD operations on 128-bit packed SP and DP FP data and on 128-bit packed byte, word, doubleword, and quadword integers.

Intel® Netburst® Architecture



Cache Basics

Memory on a computer is organized in a hierarchial manner:



Cache Basics, contd.

- Memory Characteristics

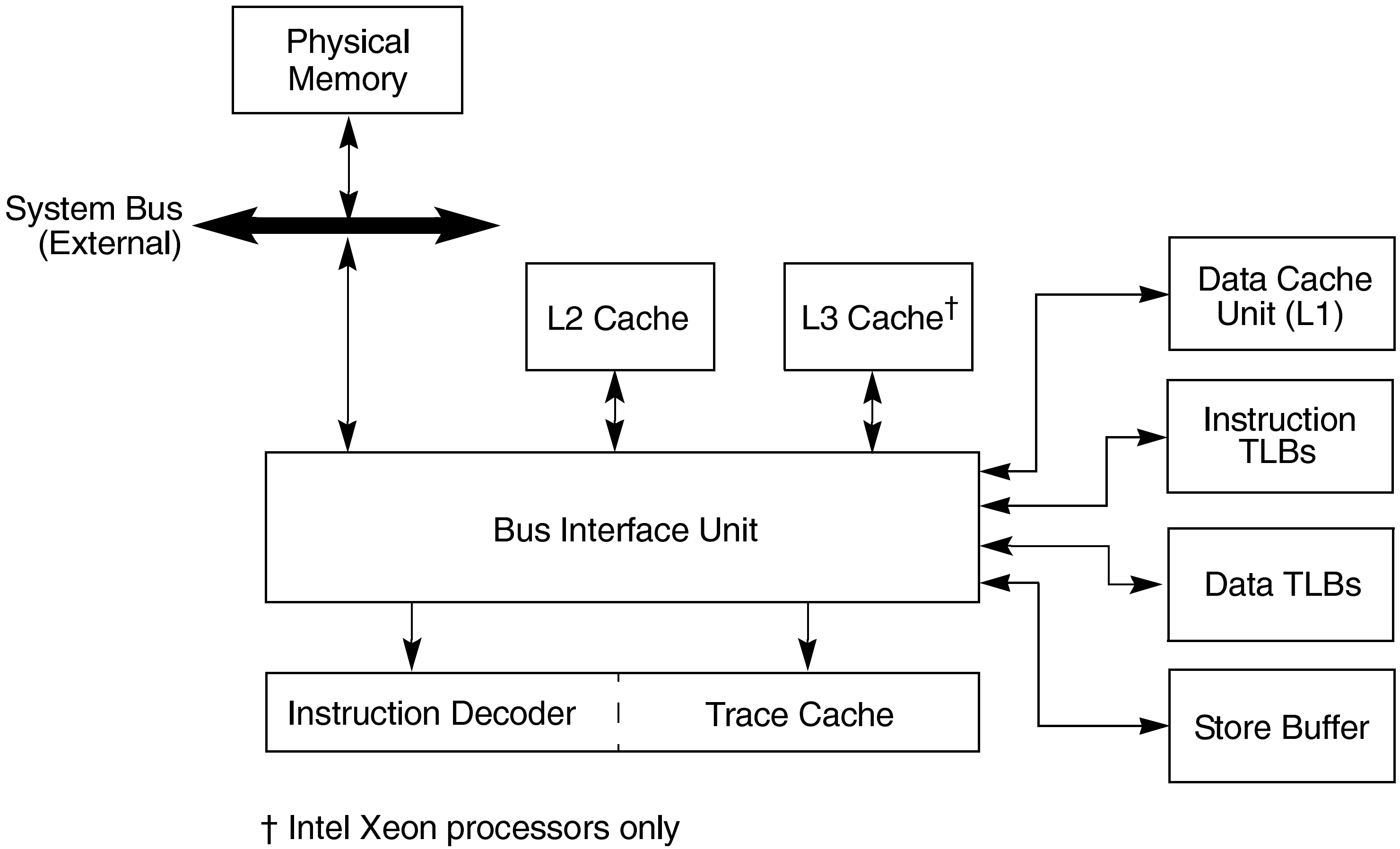
2004 $/GB	Type	Access Time (ns)
$4K - $10K	SRAM - Static (On Chip Caches)	0.5 - 5
$100 - $200	DRAM - Dynamic (Main Memory)	50 - 70
$0.5 - $2	Disk	5e6 - 20e6

- An additional area of memory is called the *Translation Lookaside Buffer (TLB)* which contains virtual – physical address translations.
- TLB's keep track of large *pages* of memory that are in use, either 4KB, or 2MB/4MB starting addresses.
- When data is needed, best to retrieve from lowest level cache if available.

Cache Basics, contd.

- *Temporal locality* - Principle that if a data location is referenced it will be referenced again soon.
- *Spatial locality* - Principle that if a data location is referenced, data locations with nearby addresses tend to be referenced soon.
- A cache *line* is usually either 64 or 128 bytes of contiguous storage.
- When CPU requires a data item not currently in cache (L1,L2) it will read an entire line of data into the appropiate cache (L1, L2, or all).
- Levels in cache hierarchy are not inclusive. The fact that a line is in Level i does NOT imply that it is also in Level $i + 1$.

Cache Structure



Cache Characteristics

- Cache Parameters

Characteristic	Intel Pentium 4	AMD Opteron
L1 Cache Size	8 KB data, 12K instr trace	64 KB data, 64 KB instr
L1 Assoc.	4-way set assoc.	2-way set assoc.
L1 Line Size	64 bytes	64 bytes
L1 Policy	LRU, Write-through	LRU, Write-back
L2 Cache Size	512 KB (data and instr)	1 MB (data and instr)
L2 Assoc.	8-way set assoc.	16-way set assoc.
L2 Line Size	128 bytes	64 bytes
L2 Policy	LRU, Write-back	LRU, Write-back

- Associativity refers to how many locations in cache a line can be mapped to.
- Policy refers to Least Recently Used (LRU), i.e., what gets replaced.
- Write-through and Write-back refer to ensuring data integrity when writing to cache.

Program Performance

- The only complete and reliable measure of computer performance is time.
-

$$\text{Time} = \frac{\text{sec}}{\text{Prog}} = \frac{\text{Instr}}{\text{Prog}} \times \frac{\text{Clk cycles}}{\text{Instr}} \times \frac{\text{sec}}{\text{Clk cycle}}$$

- Basic components

Component	Unit
CPU execution time	seconds
Instr Count	# Instr retired
Clk cycles/Instr (CPI)	Avg. # Clk cycles/Instr
Clk cycle time	seconds/Clk cycle

Program Performance, contd.

- Algorithm affects Instr Count, possibly CPI
- Programming Language affects Instr Count, CPI
- Compiler affects Instr Count, CPI
- Instr Set Arch. affects Instr Count, clock rate, CPI
- Significant program performance degradation can occur when cache misses (including TLB misses) occur on a frequent basis.

Conclusions

- Intel® Pentium® 4 has much better computational capabilities than earlier Intel processors.
- Pipelined execution on Pentium 4 allows up to 3 IA-32 instr to execute in a single clock cycle.
- Pentium 4 L1 Cache is small.
- Pentium 4 MMX/SSE/SSE2/SSE3 extensions provide enhanced computational capabilities and allow operating on multiple data items at one time (SIMD).
- To achieve optimum floating point calculation performance, need to be able to use these capabilities.

Next Week, Part II

- Algorithms and techniques to utilize cache better.
- Compiler choices and options to enable utilization of enhanced features.
- High performance libraries designed to run fast, whats available and how to use.
- Performance Monitoring of applications and interpretation of results.
- Post-processing and visualization of data and results.