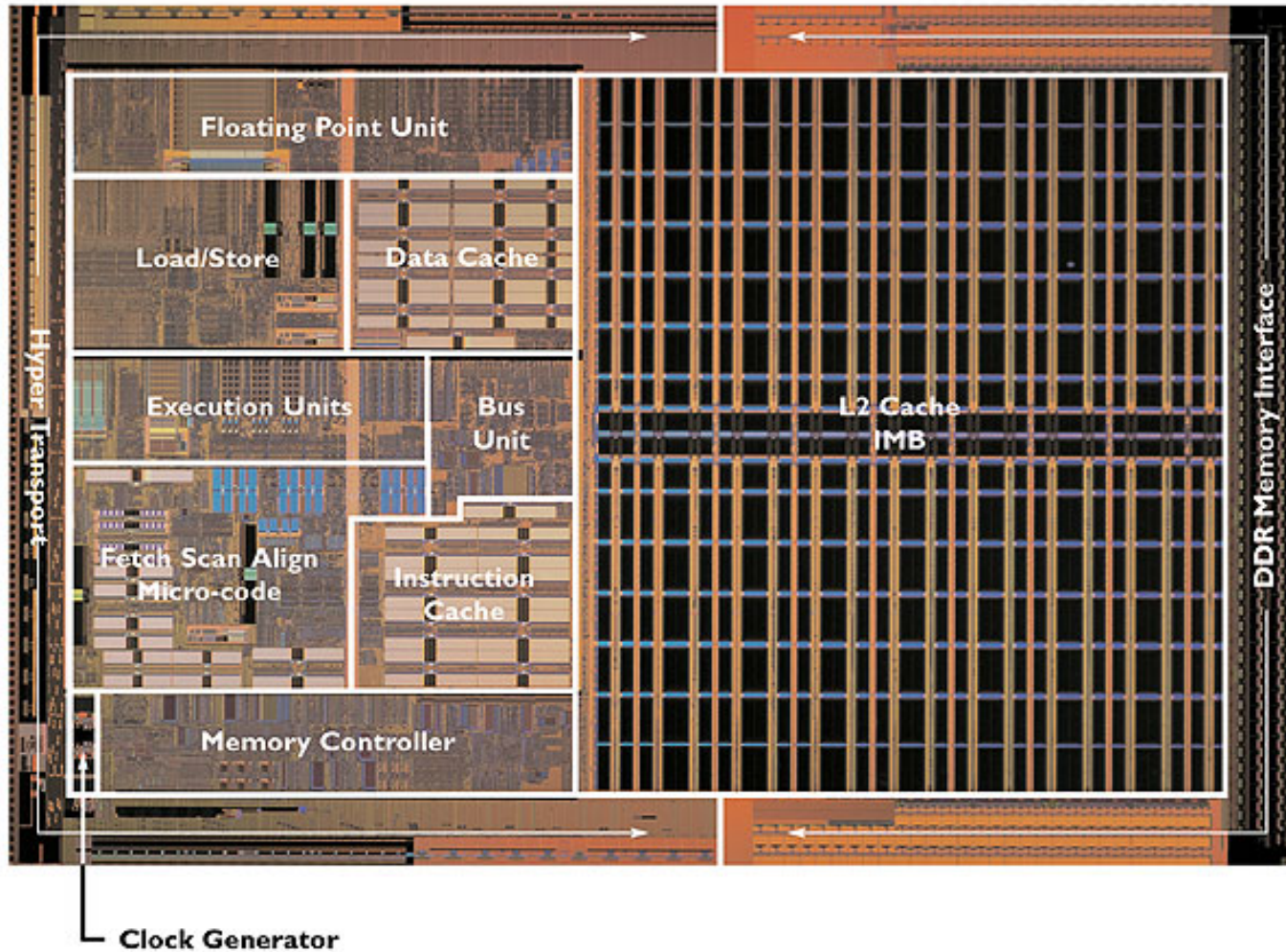


Advanced Computer Architecture

Fall 2007



Slides from ...

- We use course slides from :
 - “CS232: Advanced Computer Architecture II”
 - <http://www.cs.uiuc.edu/class/fa07/cs232/lectures/>
- ©2006 Craig Zilles (adapted from slides by Howard Huang)

department of
COMPUTER SCIENCE



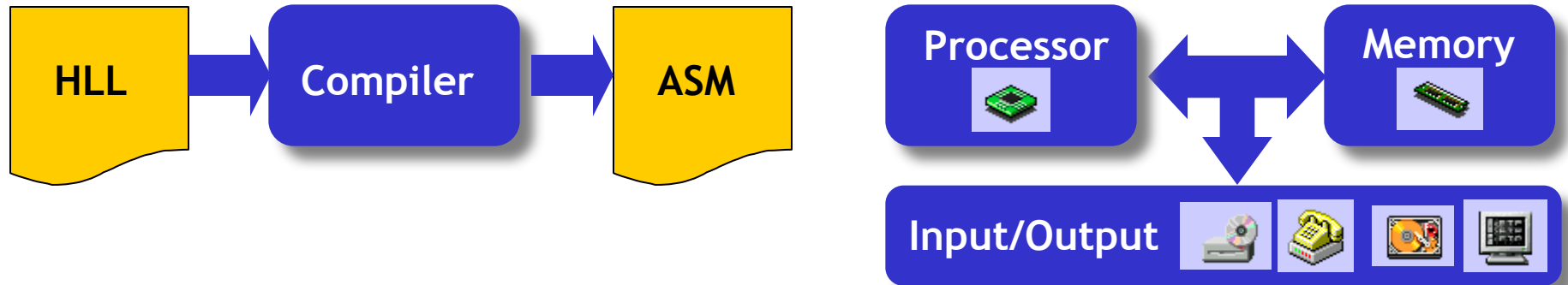
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

What is computer architecture about?

- **Computer architecture** is about building and analyzing computer systems.
- This course is roughly split into three parts.
 - The first third discusses **instruction set architectures**—the bridge between hardware and software.
 - Next, we introduce more advanced processor implementations. The focus is on **pipelining**, which is one of the most important ways to improve performance.
 - Finally, we talk about **memory** systems, **I/O**, and how to connect it all together.

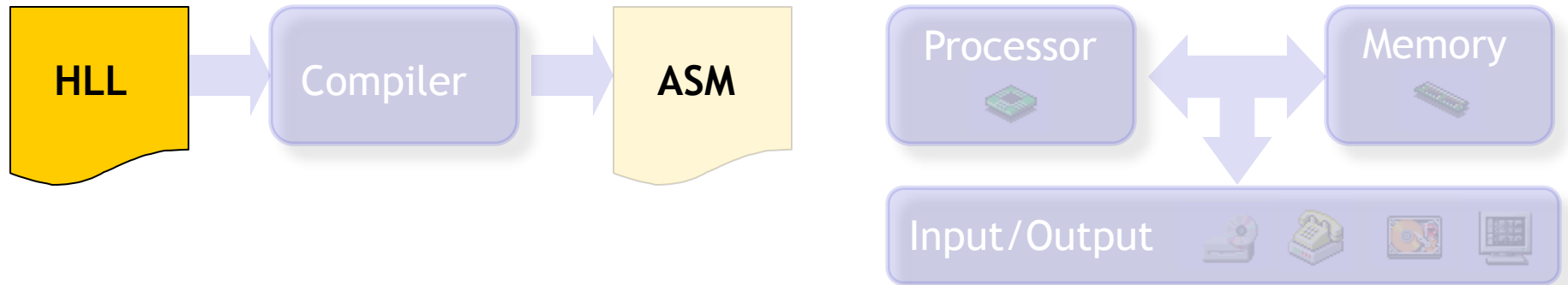
What is computer architecture about?

- **Computer architecture** is about building and analyzing computer systems.



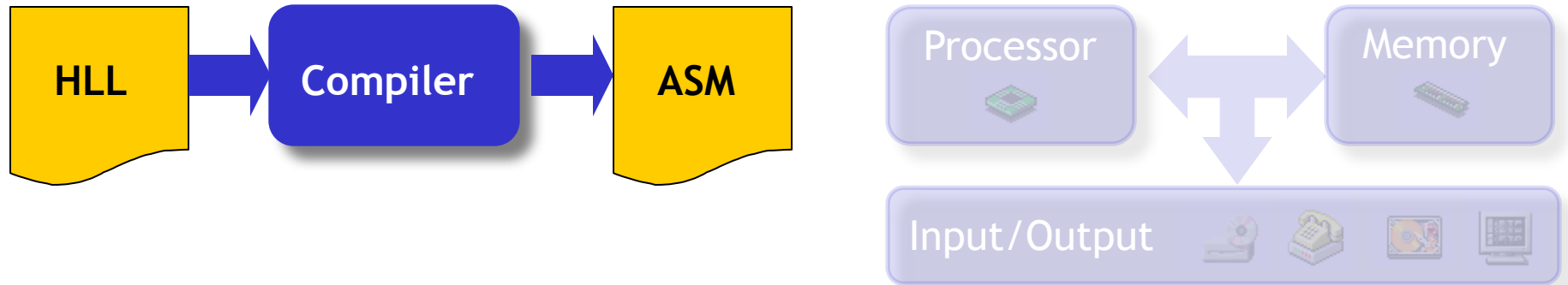
- We will take a tour of the whole machine.
- Specifically, we'll...

Do low-level programming in a high-level language



- We'll look at bit-wise logical and shifting operations in C (Stage HLL: *High Level Language*).

Study Instruction Set Architectures

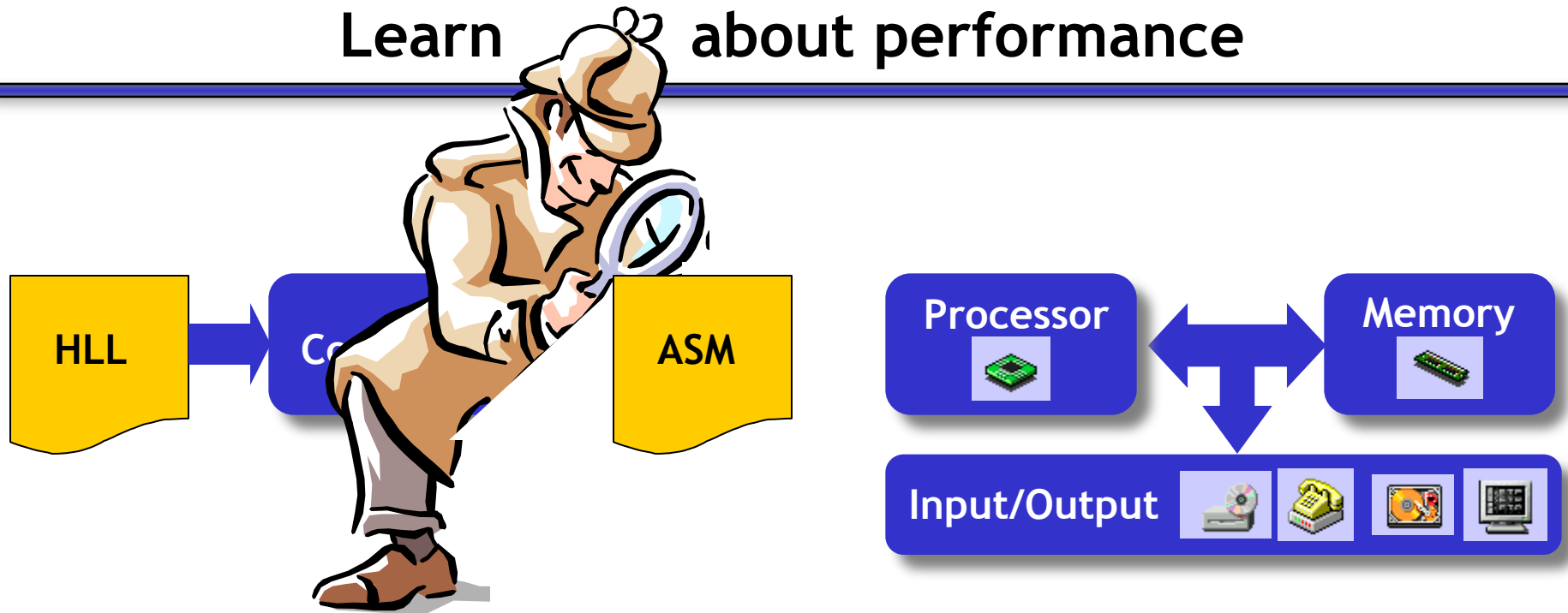


- The Instruction Set Architecture (ISA) is the bridge between the hardware and the software.
 - We'll learn the MIPS ISA in detail
 - We'll get a brief introduction to the x86 ISA
 - We'll learn how HLL program constructs are represented to the machine
 - We won't learn how compilers work, but we'll learn what they do

MIPS (in Turkish)

- Microprocessor without Interlocked Pipeline Stages, MIPS Technologies, 1985
 - http://tr.wikipedia.org/wiki/MIPS_Mimarisi
- İndirgenmiş komut kümesi türü ilk mikroişlemci mimarisidir.
 - Her komut aynı boyuttadır ve komut bilgisayar donanımı tarafından kolayca çözülebilir.
- Intel x86 ise karmaşık komut kümeli bilgisayar sayılır.
 - Komutların boyutları farklıdır ve komutları çözebilmek için bilgisayar donanımına gömülmüş programlar (microcode) gereklidir.
- RISC yapısından ötürü tasarımı çok temiz ve basittir.
 - Sistem karmaşık işlemleri destekleyen yapılar yaratmaktansa sık yapılan basit işlemleri iyileştirme üzerine kuruludur.
 - Bu tasarım avantajından dolayı üniversitelerdeki bilgisayar mimarisi derslerinde genellikle MIPS mimarisi okutulur.
 - Yine basit ve sağlam tasarımından ötürü çoğu modern mikroişlemci mimarisi (IBM/Motorola PowerPC, DEC, ARM) MIPS mimarisinden esinlenerek geliştirilmiştir.
- 1990 itibariyle üretilen her üç RISC işlemciden birinin MIPS mimarisinde olduğu tahmin edilmektedir. İlk MIPS tasarımları 32 bit, daha yeni tasarımlar ise 64 bittir.
- MIPS mimarisi SGI bilgisayarlarından gömülü sistemlere kadar geniş bir yelpazede kullanılmaktadır.
 - Örneğin; Nintendo 64, Sony PlayStation ve Sony PSP MIPS mimarisi ile çalışan işlemcilere sahiptirler.

Learn about performance



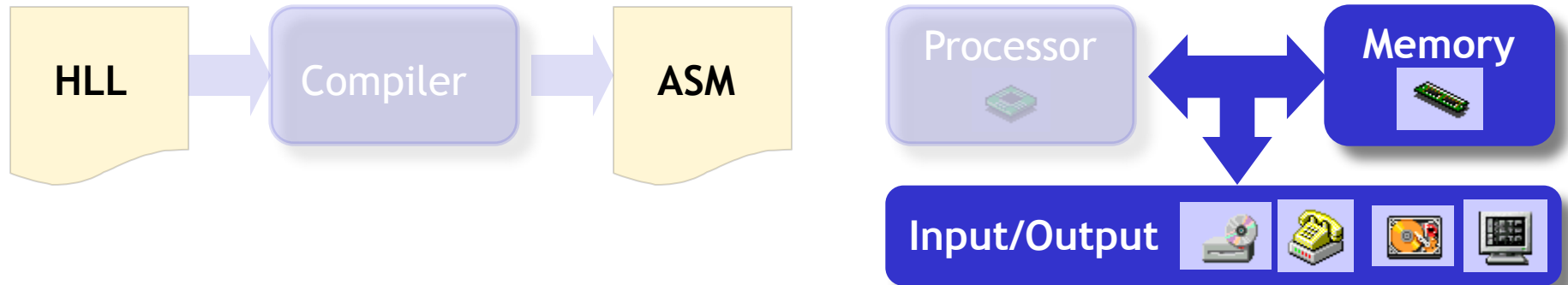
- We'll learn how to performance tune programs.
- We'll exploit explicit parallelism to make programs run faster
 - We'll optimize a program using SSE instructions

Learn about Modern Processor Organization



- The key technique we'll focus on is: Pipelining
 - Pipelining allows processors to work ~~on multiple~~ instructions at the same time.

Learn about Memory and I/O systems



- We'll learn how virtual memory makes programming easy
- We'll learn how caches make memory fast
- We'll learn about buses and disks

Why should you care?

- It is **interesting**.
 - How do you make a processor that runs at 3Ghz?
- It will help you be a **better programmer**.
 - Understanding how your program is translated to assembly code lets you reason about correctness and performance
 - Demystify the seemingly arbitrary (*e.g.*, bus errors, segmentation faults)
- Many cool jobs require an understanding of computer architecture.
 - The cutting edge is often pushing computers to their limits.
 - Supercomputing, games, portable devices, etc.
- Computer architecture illustrates many **fundamental ideas** in computer science
 - Abstraction, caching, and indirection are CS staples