EECE416 Microcomputer Fundamentals

Microprocessor Architecture

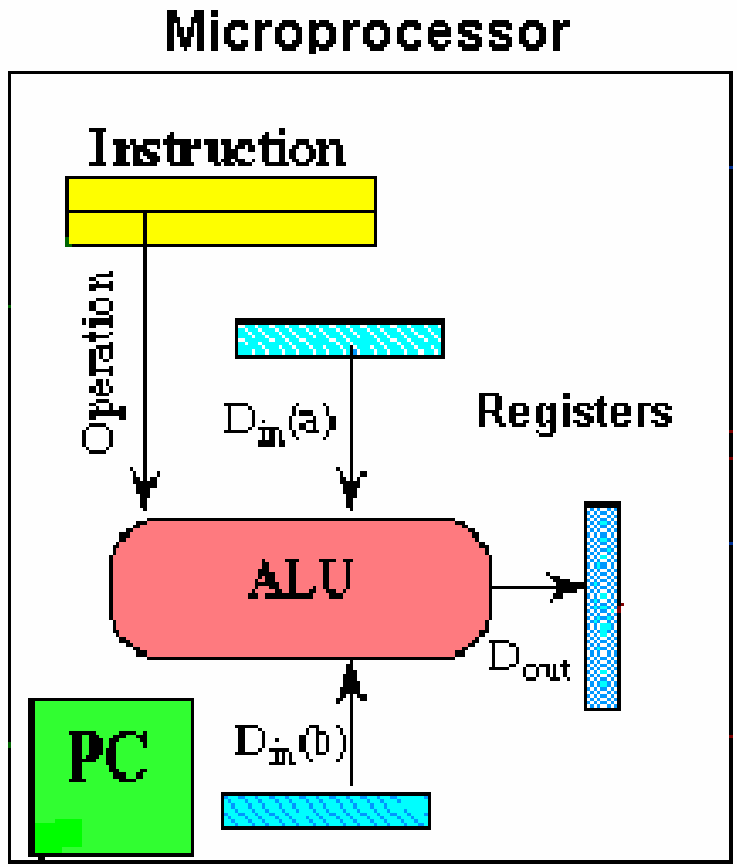
Dr. Charles Kim

Howard University

WWW.MWFTR.COM

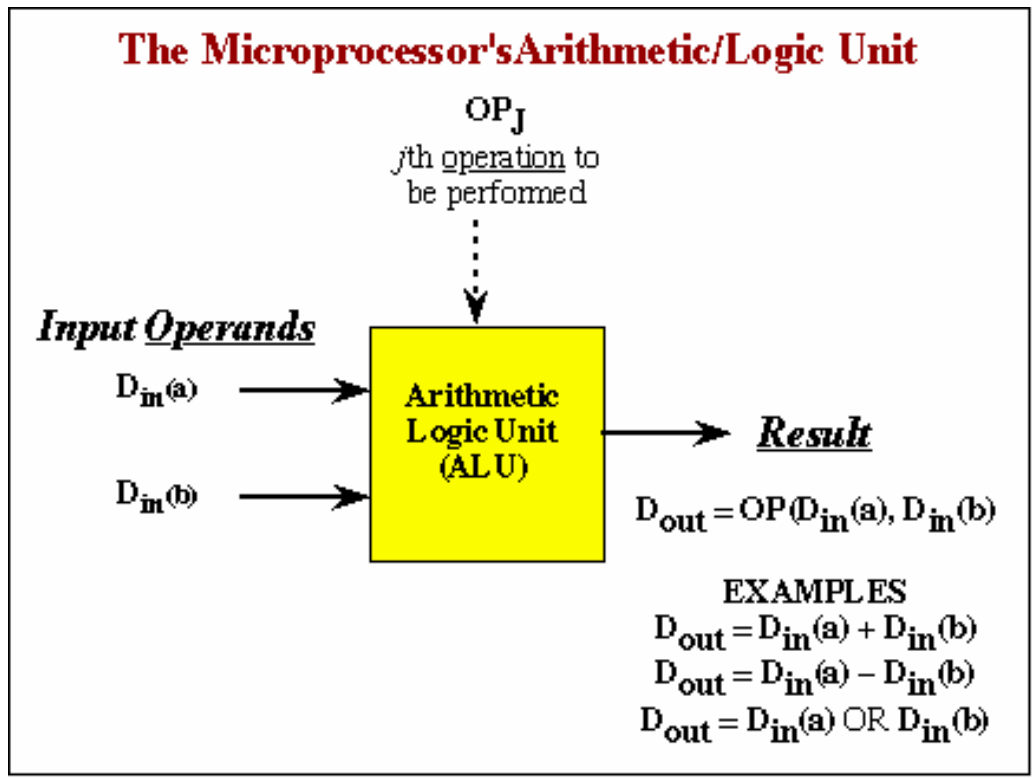
Computer Architecture

- Computer System
  - CPU (with PC, Register, SR) + Memory

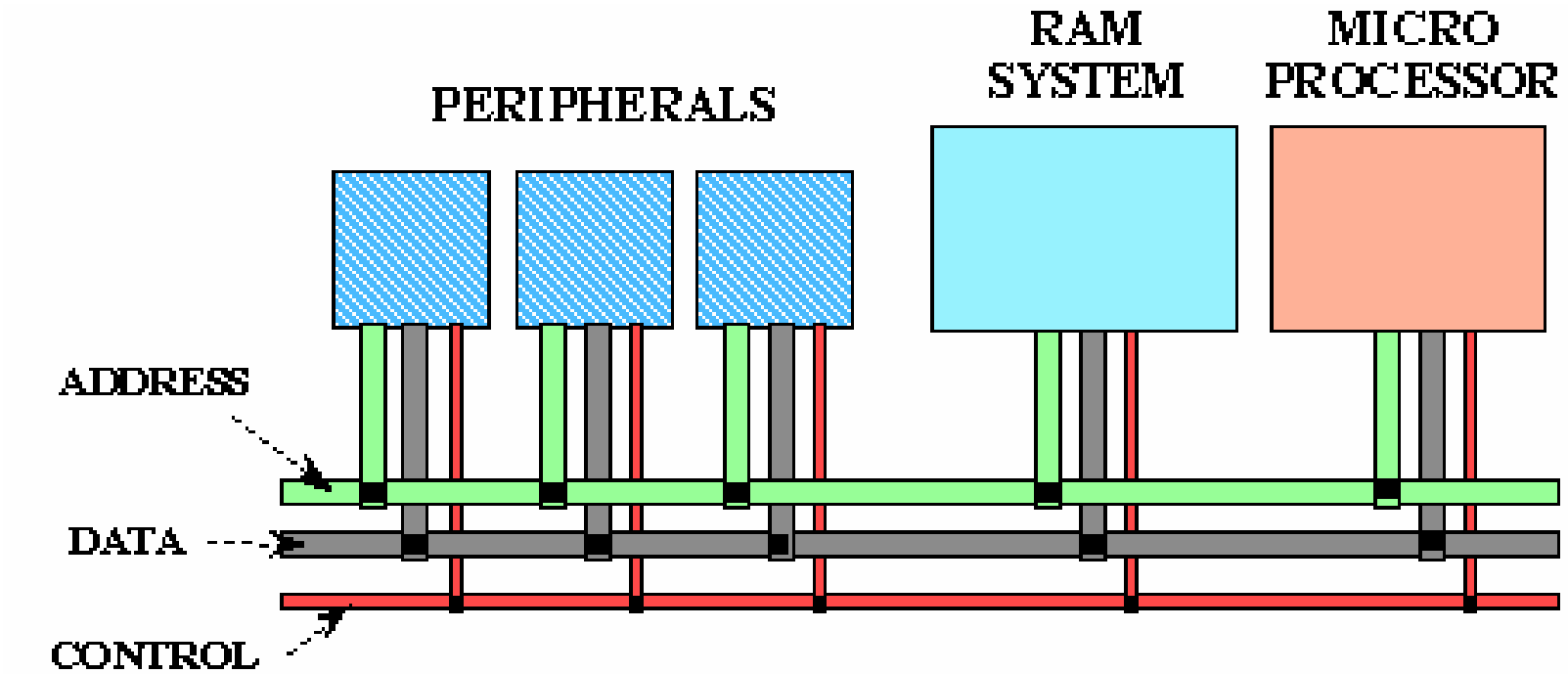


Computer Architecture

- **ALU (Arithmetic Logic Unit)**
  - **Binary Full Adder**



Microprocessor Bus



Architecture by CPU+MEM organization

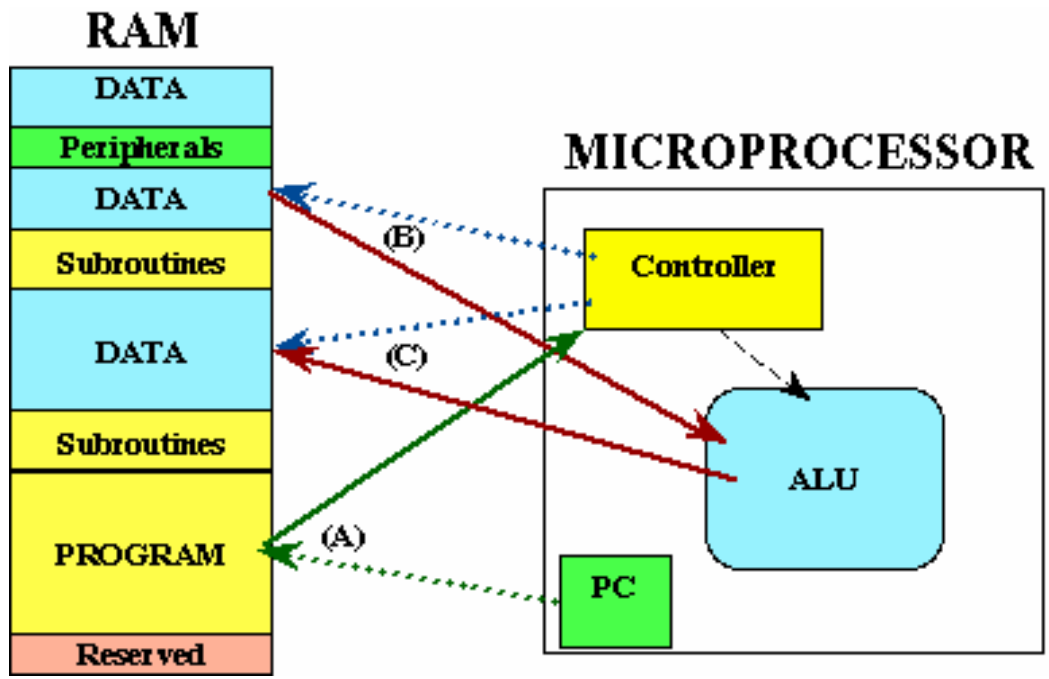
- Princeton (or von Neumann) Architecture
  - MEM contains both Instruction and Data
- Harvard Architecture
  - Data MEM and Instruction MEM
  - Higher Performance
  - Better for DSP
  - Higher MEM Bandwidth

Princeton Architecture

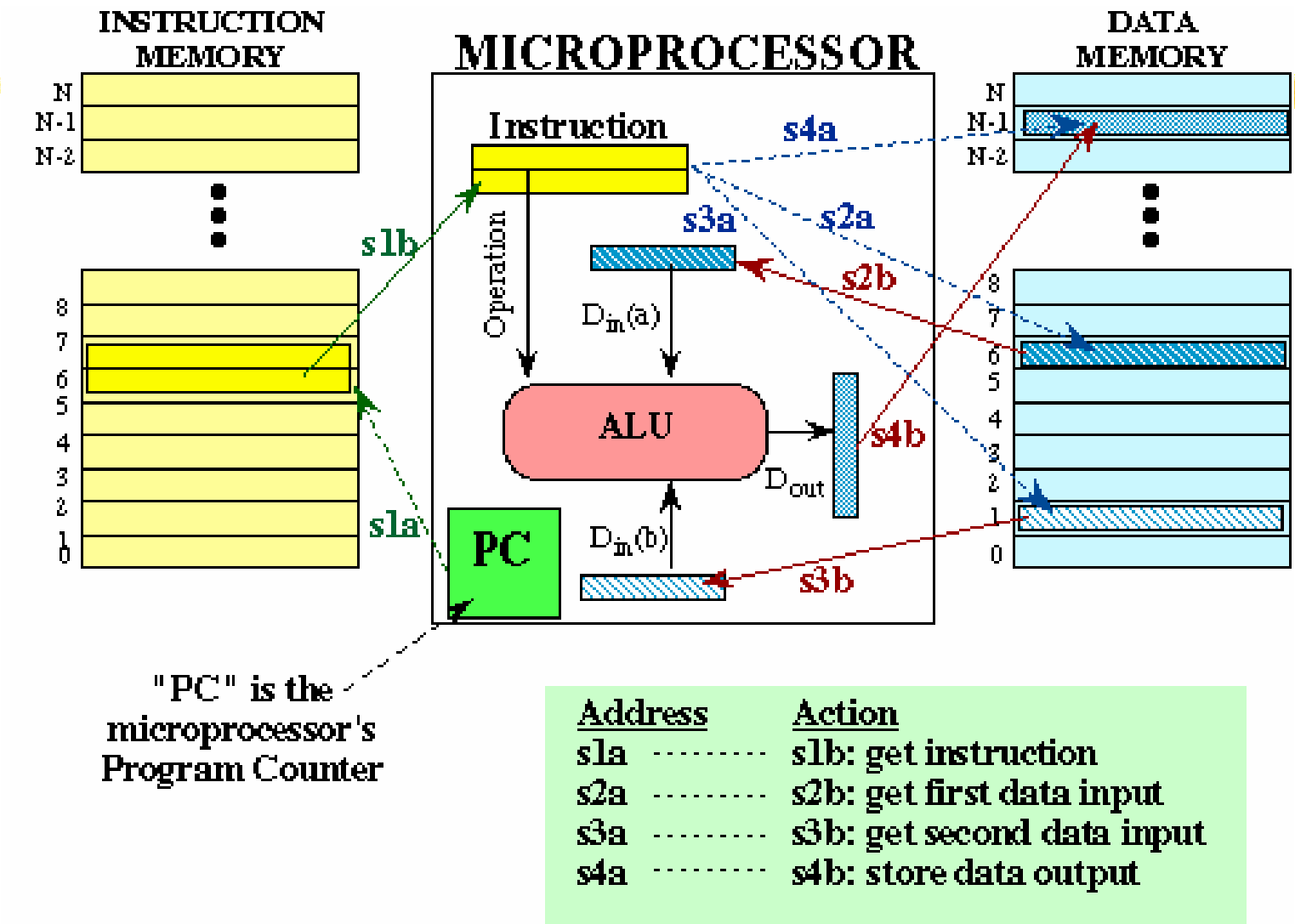
1.**Step (A):** The address for the instruction to be next executed is applied

(**Step (B):** The controller "decodes" the instruction

3.**Step (C)**: Following completion of the instruction, the controller provides the address, to the memory unit, at which the data result generated by the operation will be stored.

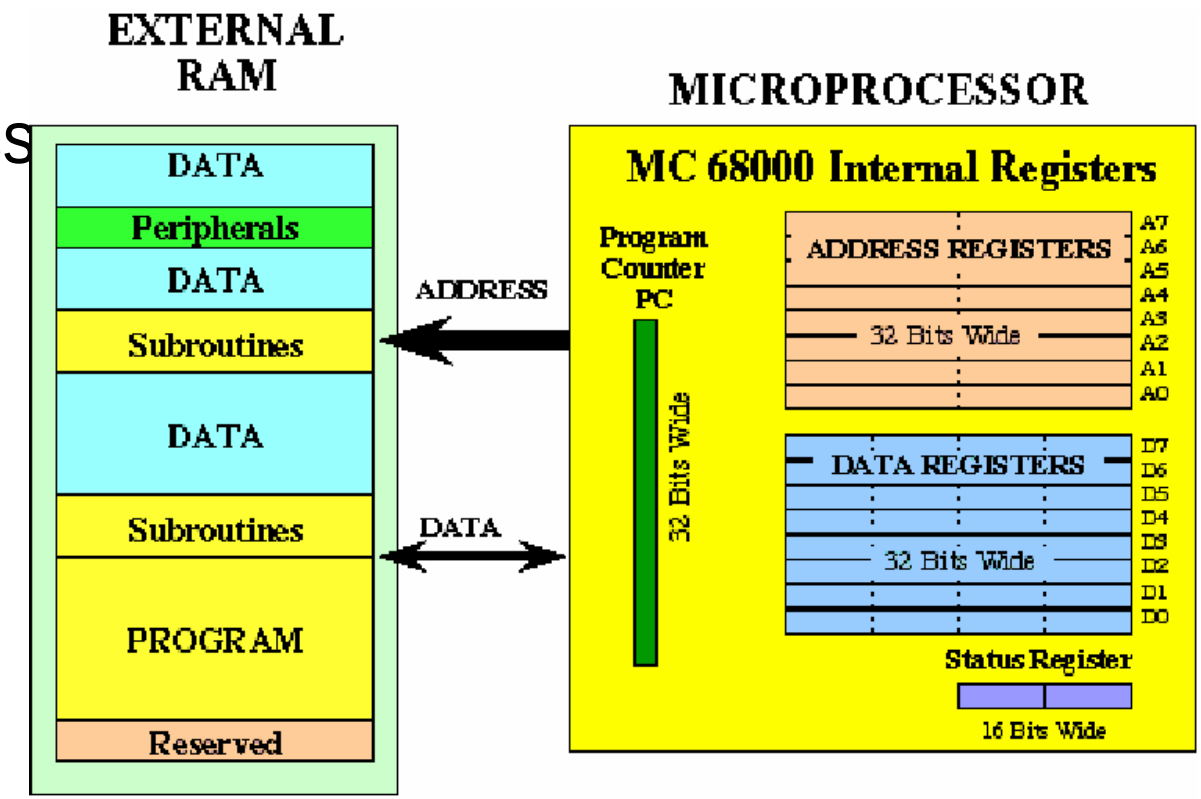


Harvard Architecture



Internal Memory ("register")

- External memory access is Very slow
- For quicker retrieval and storage
- Internal registers



Architecture by Instructions and their Executions

- CISC (Complex Instruction Set Computer)
  - Variety of instructions for complex tasks
  - Instructions of varying length
- RISC (Reduced Instruction Set Computer)
  - Fewer and simpler instructions
  - High performance microprocessors
  - Pipelined instruction execution (several instructions are executed in parallel)

CISC

- Architecture of prior to mid-1980's
  - IBM390, Motorola 680x0, Intel80x86
- Basic Fetch-Execute sequence to support a large number of complex instructions
- Complex decoding procedures
- Complex control unit
- One instruction achieves a complex task

RISC

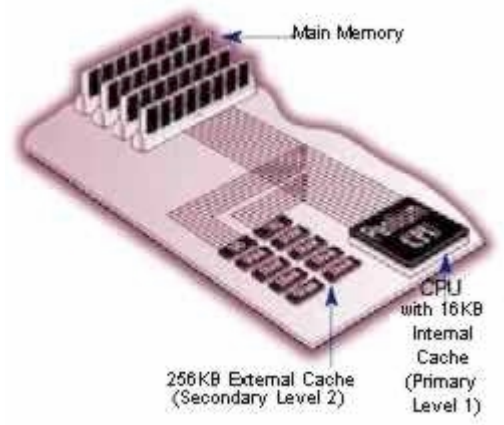
- Architecture after mid-1980's
  - IBM801, Sun Sparc, Mips, HP, IBM RS/6000
  - PowerPC
- Small set of simple instructions
- Pipelined architecture
  - Overlapping many concurrent operations

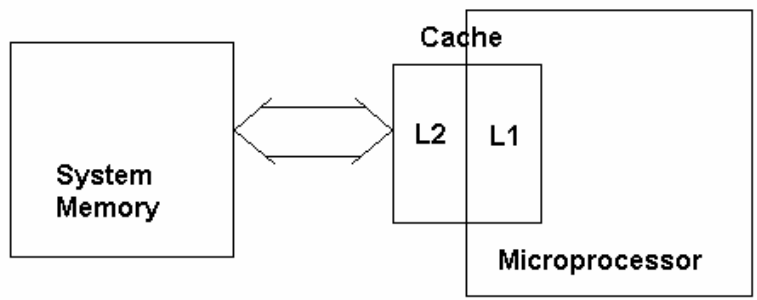
Characteristics of RISC

- Favorable changes for RISC
  - **Caches** to speed instruction fetches
  - Dramatic memory size increases/cost decreases
  - Better ***pipelining***
  - Advanced optimizing compilers
- Characteristics of RISC
  - Instructions are of a uniform length
  - Increased number of registers to hold frequently used variables (16 - 64 Registers)
  - Central to High Performance Computing

Side Bar: Cache

- A small amount of fast memory (RAM)
  - holds recently accessed data or instructions
  - supplies them faster.
- Cache memory Organization
  - lines
  - Sets
    - separate group of memory addresses
  - 2 and 64 lines per set.
- Cache Memory Levels
  - level 1 cache
    - Small and Faster
  - level 2 cache
    - Large and Slower
  - Level 3 caches
    - in some cases.





Characteristics of RISC

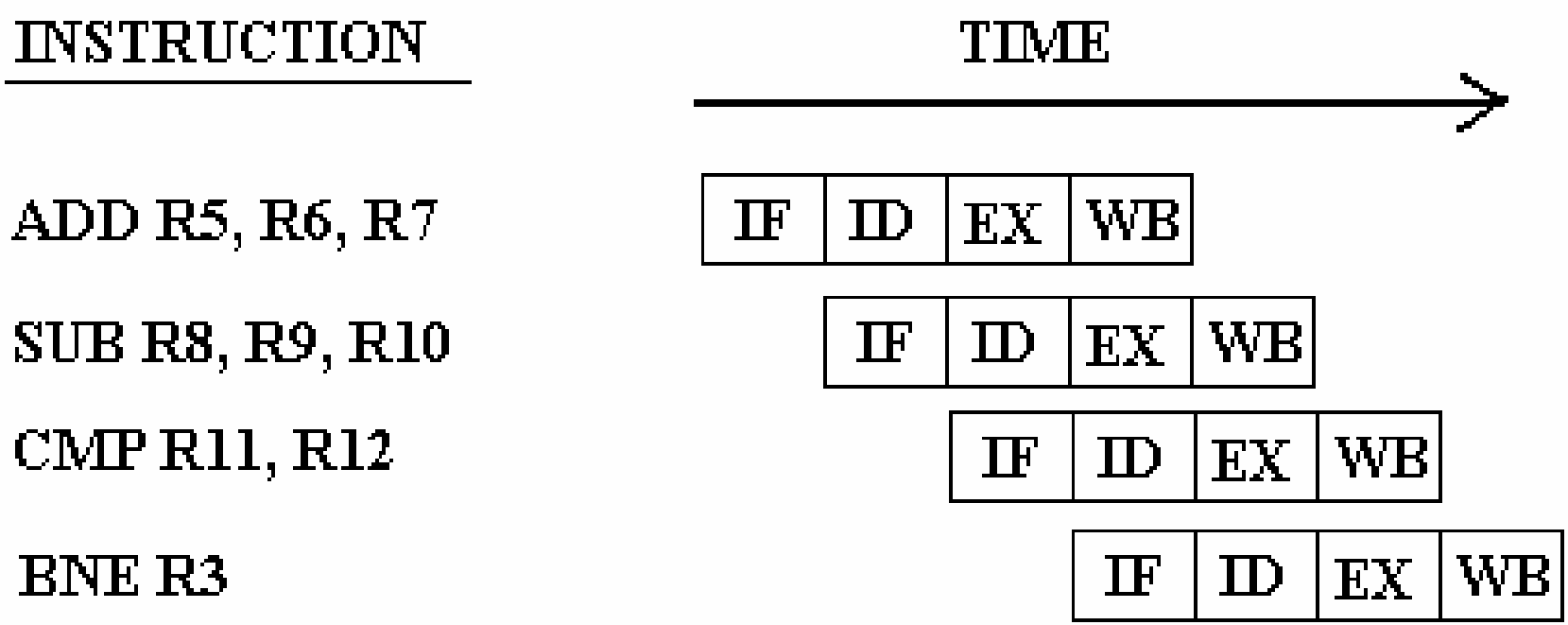
- **Memory Issues (Speed and Size)**
  - RISC processors get faster and faster, they eat memory faster
  - Memory costs money
  - Finding memory locations - a slow process
  - **Speed**
    - CPU speeds increase steadily, while memory speeds a fraction of the speed of CPU's
    - Use of memory caches.
  - **Size**
    - A method that increases the maximum size for a program is virtual memory
    - Actual location is translated into a different physical address
    - By separated into pages, programs are easier to fit together in memory
- **Pipelining**

Pipelining

- A method of increasing the number of concurrent operations
- Two types
  - Arithmetic pipeline
    - Ex: floating point multiplication
  - Instruction pipeline
    - Partitions the fetch-execution into several stages

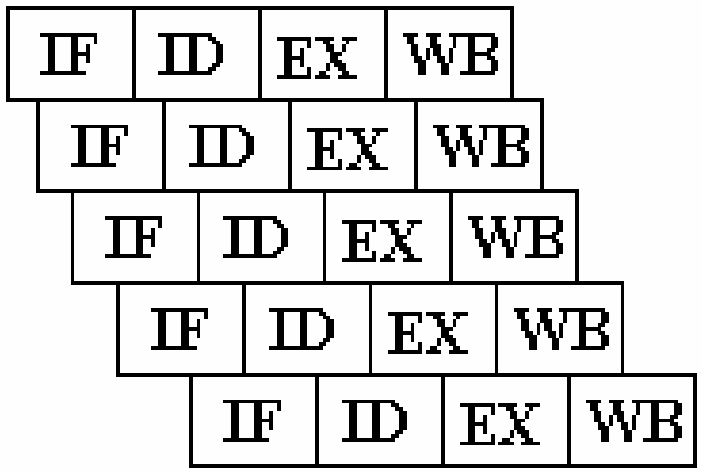
Pipeline-based Architectural techniques

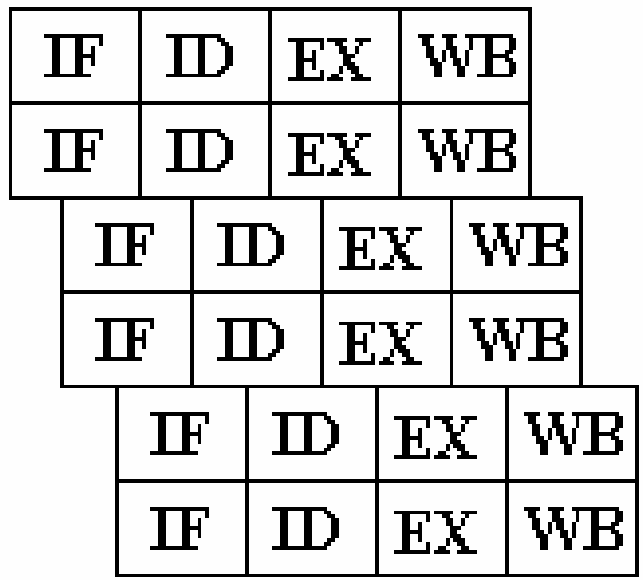
- **Basic Model: 4 stages**
  - **IF (Instruction Fetch)**
  - **ID(Instruction Decode)**
  - **EX(Execute)**
  - **WB(Write Back)**



Pipelining Techniques

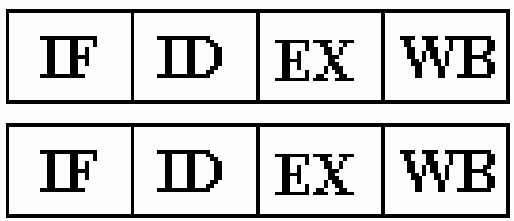
- Superpipeline architecture :
  - **Fine grained partitioning of the stages**

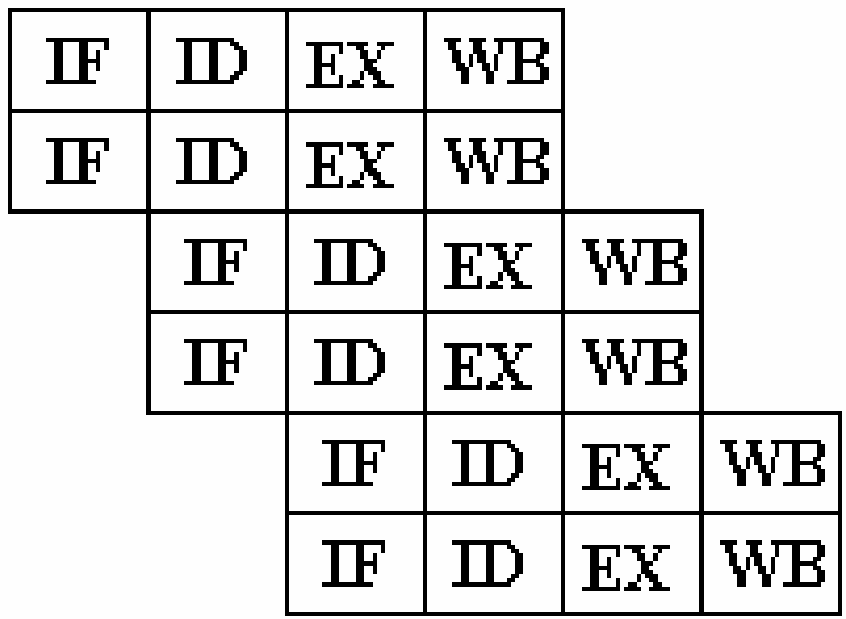




Pipelining Techniques

- Superscalar pipeline architecture:
  - **More than one instruction in one clock cycle**





Pipelining Techniques

- **VLIW(Very Long Instruction Word) architecture:**
  - Several instructions are packed into one long instruction word

Memory Ref. Instr. 1	Memory Ref. Instr. 2	Float Pt. Istr 1	Float Pt. Inst 2	Integer Inst
159 ... 128	127 ... 96	95 ... 64	63 . . . 32	31 . . . 0

- **Vector pipelines:**
  - Vector operations are implemented with single instruction

Differences	CISC	RISC
Instruction Type	Many	Few
Addressing Mode	Many	Few
Instruction Format	Variable	Fixed
Pipelining	Difficult	Efficient
Control Method	Micro-programmed control	Hardwired control
Instruction Access	Most access memory	Mostly load-store architecture
Optimization Dependency for high performance	Less on optimizing compiler	Highly on optimizing compiler

History of RISC

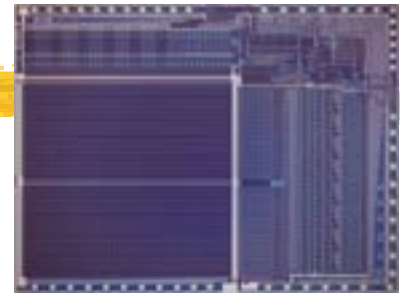
- ***RISC Roots:* CDC 6600 (1965) . .**
  - Control Data Corporation CDC 6600 'Supercomputer'
  - Designed by Seymour Cray
  - Emphasized a small (74 op codes) load/store and register-register instruction as a means to greater performance.
  - The CDC 6600 itself has roots in the UNIVAC 1100, which many CDC 6600 engineers worked on.

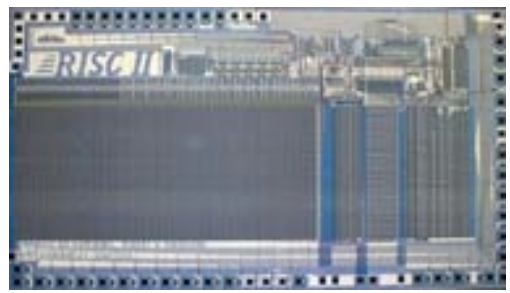
History of RISC

- ***RISC Formalized:*** **IBM 801 (1975). . .**
  - The first system to formalize these principles was the IBM 801 project (1975)
  - The design goal was to speed up frequently used instructions while discarding complex instructions that slowed the overall implementation.
  - Like the CDC 6600, memory access was limited to load/store operations
  - Execution was pipelined, allowing 1 instruction per cycle.

History of RISC

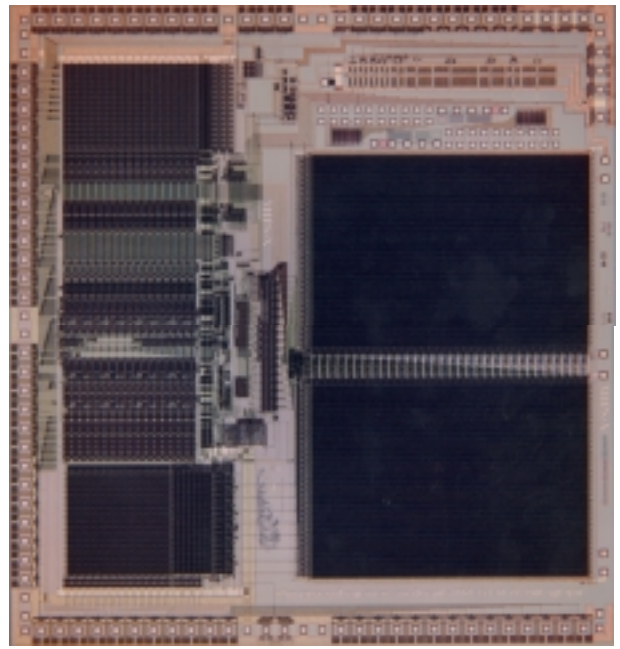
- ***RISC Refined:*** **Berkeley RISC**
  - Some time after the 801, around 1981, projects at Berkeley (RISC I and II) further developed these concepts.
  - The term RISC came from Berkeley's project, which was the basis for the fast Pyramid minicomputers and SPARC processor.
  - Execute in a 3 stage pipeline.





RISC Refined: Stanford MIPS . .

- Some time after the 801, around 1981, Stanford University (MIPS) further developed these concepts.
- The Stanford MIPS project was the basis for the **MIPS**.
- MIPS stood for Microprocessor without Interlocked Pipeline Stages, using the compiler to eliminate register conflicts.

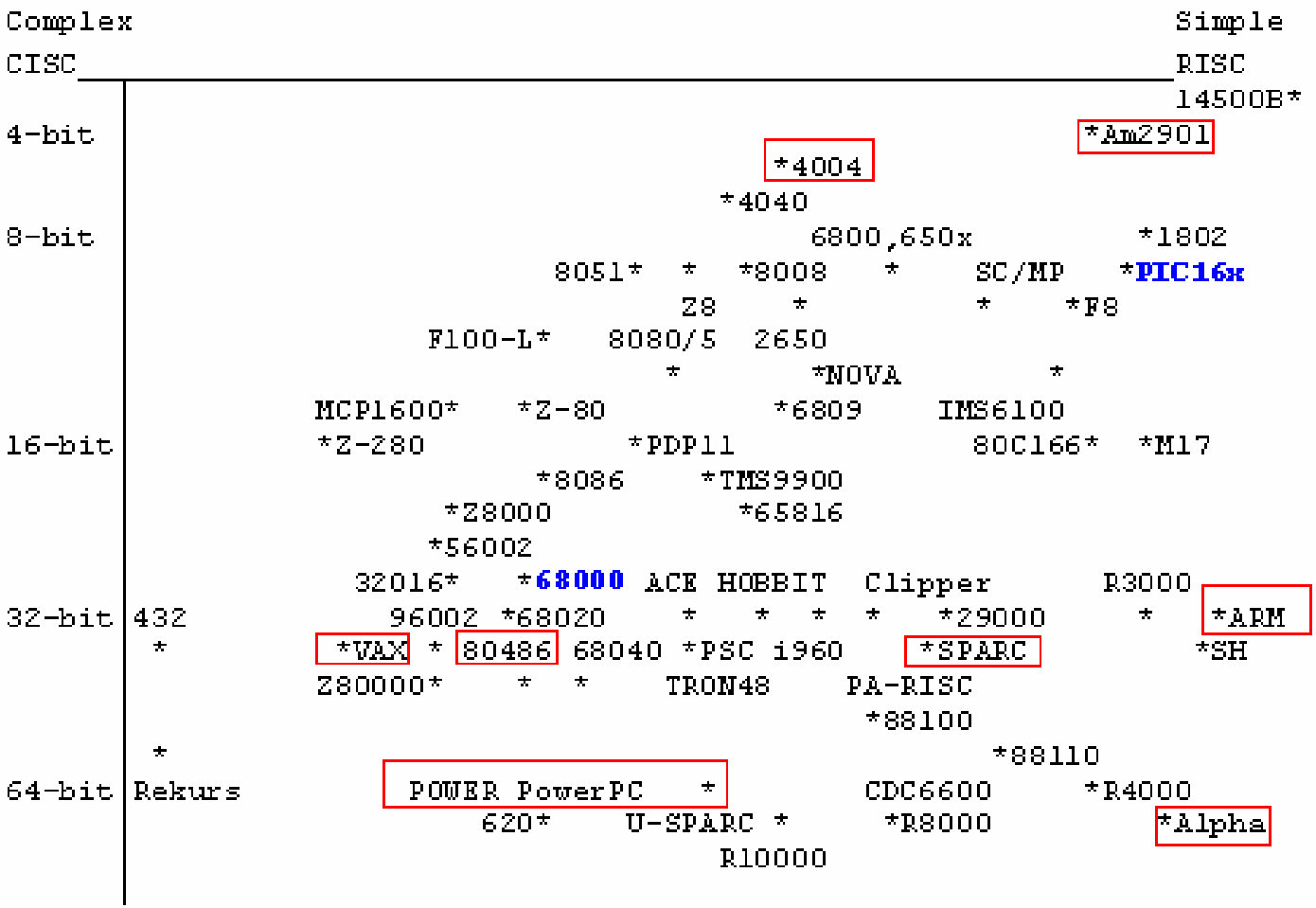


SPARC & ARM

- SPARC
  - Scalable (originally Sun) Processor ARChitecture
  - Designed by Sun Microsystems for their own.
  - Standard 68000-based CPUs and a standard operating system, Unix.

- ARM
  - Advanced RISC Machine, originally Acorn RISC Machine
  - One of the most elegant modern processors in existence.
  - Simple, with a short 3-stage pipeline, and it can operate in big- or little-endian mode.

Processor Classification



Assignment 1

- Pick a topic from below and write a short report (4 – 5 page long) and Presentation file
- Your audience
  - CEO, President, and CTO of your company
  - Weekly Technical Education Briefing
- Topics to Choose
  - ARM processor and its varieties – A specific ARM professor which is best for Cell phones
  - Pipelining – Intel Pentium 4 Pipelining: Strength and Drawback
  - Current Practice of Cache in PC and Its Layout in Motherboard
- Report Due: R September 20
- Presentation File Due: M September 24
- Presentation: T September 25