## RISC Architectures

COMP375 Computer Architecture and Organization

## RISC Design Principles

- Large number of registers
  - Needed to support register-to-register operations
  - Minimize the procedure call and return overhead
- Fixed-length instructions
  - Facilitates efficient instruction execution
- Simple instruction format
  - Fixed boundaries for various fields

## RISC Design Principles

- Simple operations
  - Simple instructions that can execute in one cycle
- Register-to-register operations
  - Only load and store operations access memory
  - Rest of the operations on a register-to-register basis
- Simple addressing modes
  - A few addressing modes (1 or 2)

## RISC Design Principle

- Start an instruction every cycle
- Simple, fixed length instructions are easy to pipeline.
- Only two instruction have memory operands all other operands are in registers.
- Delayed branches

## Example Differences

	CISC		RISC
	VAX 11/780	Intel 486	MIPS R4000
# instructions	303	235	94
Addr. modes	22	11	1
Inst. size (bytes)	2-57	1-12	4
GP registers	16	8	32

chart @ Dandamud

#### **RISC Traits**

- Pipelined
- Simple instructions
- Few instructions
- No microcode
- Few addressing modes
- Load/Store architecture
- Sliding register stack
- Delayed branches
- Fast

## CICS advantages include:

- 1. Sliding register stack
- 2. Instructions designed to match high level language features
- 3. Load/Store architecture
- 4. Large microcode memory

## Current RISC Systems

- PowerPC The processor in the Apple Power Mac. Produced by IBM and Apple.
- Sparc The processor in Sun workstations and servers. Produced by Sun Microsystems. First commercial RISC.
- **MIPS** Frequently used in embedded devices.
- Itanium In new servers replacing the Intel Pentium. Produced by Intel.

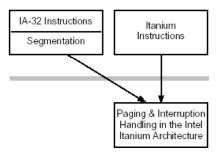
#### Intel Itanium®

- Intel's latest RISC system.
- The current processor is the Itanium 2.
- Intel seems to indicate that this is the replacement for the Pentium chip.

## Support of Pentium Instructions

- The Itanium can execute both Itanium instructions and Pentium (IA-32) instructions
- There are jump to IA-32/Itanium instructions

Intel® Itanium® System Environment



#### Parallelism

Parallel activities can be done at many different levels.

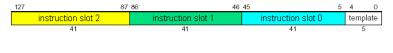
- Parallel, independent programs
  - Multi-core processors
- Instruction Level Parallelism (ILP)
  - Itanium
  - It is easier to detect ILP in long sequences of instructions that are not broken by jumps.

### **Discovering Parallelism**

- Most programs are written as a sequential stream of instructions.
- The CPU has to discover any parallelism to support pipelining and superscalar execution.
- The compiler has a much bigger picture of the program and can easily recognize opportunities for parallelism.
- It can be difficult for the compiler to pass parallelism information to the CPU.

#### Instruction Bundles

- Explicitly Parallel /instruction Computing (EPIC)
- Three 41 bit instructions are grouped into a bundle with a 5 bit template.
- There must be no dependencies within the instructions of a bundle.



# A 2.0 GHz Itanium will run faster than a 3.0 GHz Pentium because

- Itanium Hz are faster than Pentium Hz
- 2. The Itanium can run three instructions at once.
- 3. The Itanium cost more
- 4. The Itanium is Hyper-threaded.

## Compiler to Processor Hints

- Every memory load and store in the Itanium architecture has a 2-bit cache hint field
- The compiler can provide a hint to indicate if a branch is likely to be taken.
- Templates define which execution units will be used and if dependencies exist.