RISC Vs CISC

- CISC (complex instruction set computer)
  - VAX, Intel X86, IBM 360/370, etc.
- RISC (reduced instruction set computer)
  - MIPS, DEC Alpha, SUN Sparc, IBM 801

RISC vs. CISC

- Characteristics of ISAs

CISC	RISC
Variable length instruction	Single word instruction
Variable format	Fixed-field decoding
Memory operands	Load/store architecture
Complex operations	Simple operations

RISC vs. CISC Instruction Set Design

- The historical background:
  - In first 25 years (1945-70) performance came from both technology and design.
  - Design considerations:
    - small and slow memories: compact programs are fast.
    - small no. of registers: memory operands.
    - attempts to bridge the semantic gap: model high level language features in instructions.
    - no need for portability: same vendor application, OS and hardware.
    - backward compatibility: every new ISA must carry the good and bad of all past ones.

Result: powerful and complex instructions that are rarely used.

Top 10 80x86 Instructions

° Rank	instruction	Integer Average Percent total executed
1	load	22%
2	conditional branch	20%
3	compare	16%
4	store	12%
5	add	8%
6	and	6%
7	sub	5%
8	move register-register	4%
9	call	1%
10	return	1%
	Total	96%

- Simple instructions dominate instruction frequency

Complex Instruction Set Computer (CISC)

Memory in those days was expensive

- bigger program->more storage->more money

Hence needed to *reduce* the number of instructions per program

Number of instructions are reduced by having *multiple operations* within a single instruction

Multiple operations lead to many different kinds of instructions that access memory

- In turn making instruction length variable and fetch-decode-execute time unpredictable – making it more complex
- Thus hardware handles the complexity

Example: x86 ISA

Reduced Instruction Set Computer (RISC)

Original idea to reduce the ISA

- Provide *minimal set of instructions* that could carry out all essential operations

Instruction complexity is reduced by

1. Having *few simple* instructions that are the *same length*

2. Allowed memory access only with *explicit* load and store instructions

Hence each instruction performs less work but instruction execution time among different instructions is consistent

The complexity that is removed from ISA is moved into the domain of the assembly programmer/compiler

Examples: LC3, MIPS, PowerPC (IBM), SPARC (Sun)

RISC vs. CISC

The difference between CISC and RISC becomes evident through the basic computer performance equation:

$$\text{CPU Time} = \frac{\text{seconds}}{\text{program}} = \frac{\text{instructions}}{\text{program}} \times \frac{\text{avg. cycles}}{\text{instruction}} \times \frac{\text{seconds}}{\text{cycle}}$$

RISC systems shorten execution time by reducing the *clock cycles per instruction* (i.e. simple instructions take less time to interpret)

CISC systems shorten execution time by reducing the *number of instructions per program*

Example for RISC vs. CISC

Consider the the program fragments:

CISC

```
mov ax, 10
mov bx, 5
mul bx, ax
```

RISC

```
       mov ax, 0
       mov bx, 10
       mov cx, 5
Begin  add ax, bx
       loop Begin
```

The total clock cycles for the CISC version might be:

(2 movs × 1 cycle) + (1 mul × 30 cycles) = 32 cycles

While the clock cycles for the RISC version is:

(3 movs × 1 cycle) + (5 adds × 1 cycle) + (5 loops × 1 cycle) = 13 cycles

RISC vs. CISC Summary

RISC

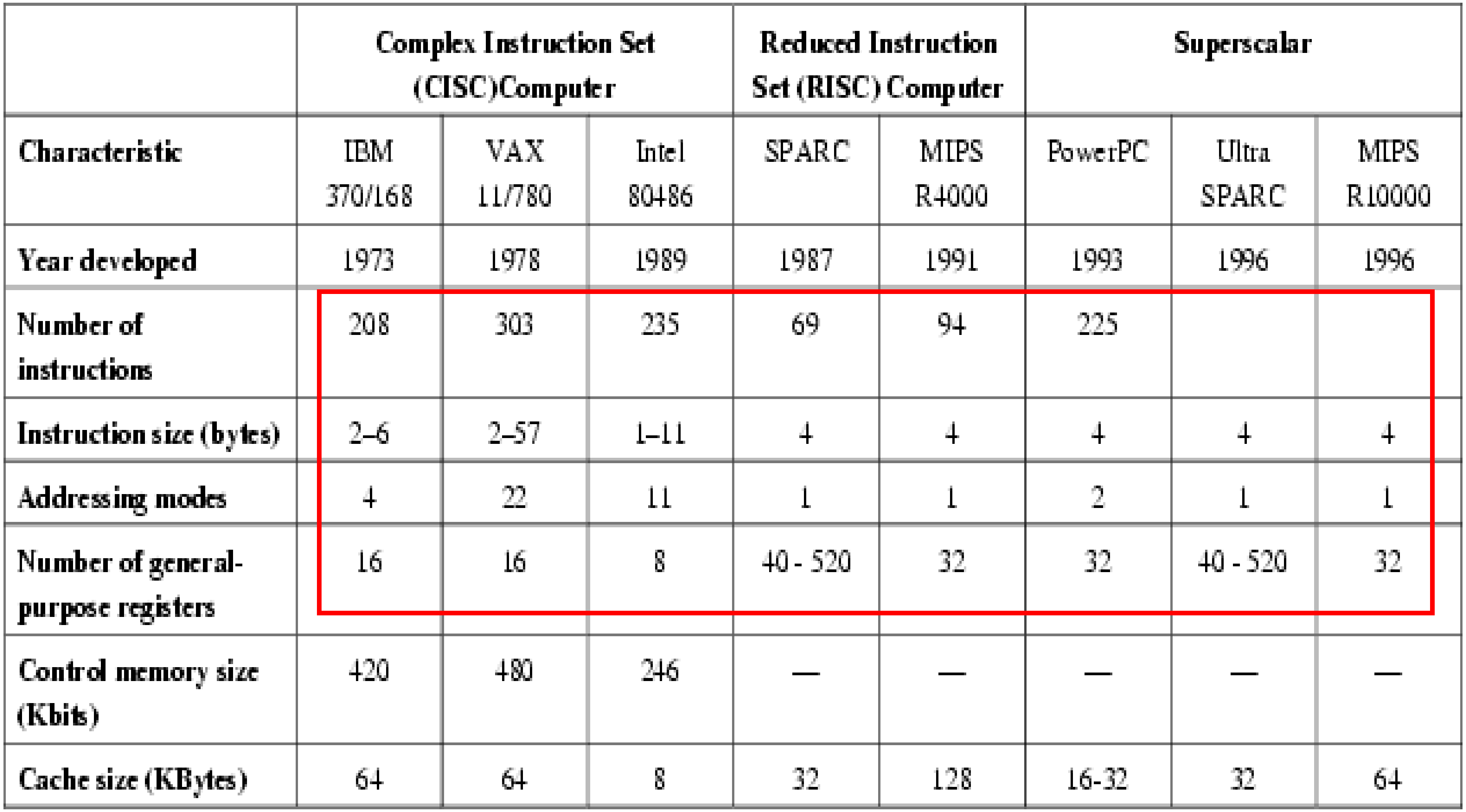
- Simple instructions, few in number
- Fixed length instructions
- Complexity in compiler
- Only `LOAD`/`STORE` instructions access memory
- Few addressing modes

CISC

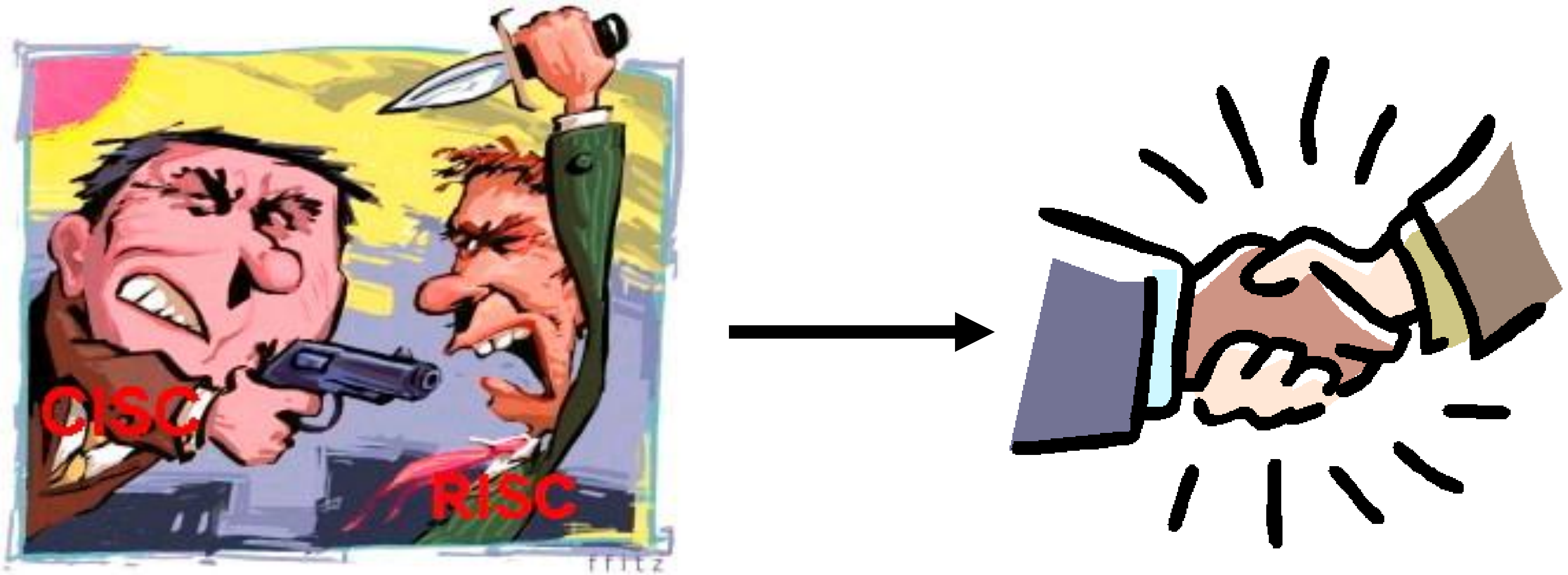
- Many complex instructions
- Variable length instructions
- Complexity in microcode
- Many instructions can access memory
- Many addressing modes

CISC	**RISC**
Emphasis on hardware	Emphasis on software
Includes multi-clock complex instructions	Single-clock, reduced instruction only
Memory-to-memory: "LOAD" and "STORE" incorporated in instructions	Register to register: "LOAD" and "STORE" are independent instructions
Small code sizes, high cycles per second	Low cycles per second, large code sizes

Comparison of Processors



	Complex Instruction Set (CISC)Computer			Reduced Instruction Set (RISC) Computer		Superscalar		
Characteristic	IBM 370/168	VAX 11/780	Intel 80486	SPARC	MIPS R4000	PowerPC	Ultra SPARC	MIPS R10000
Year developed	1973	1978	1989	1987	1991	1993	1996	1996
Number of instructions	208	303	235	69	94	225		
Instruction size (bytes)	2–6	2–57	1–11	4	4	4	4	4
Addressing modes	4	22	11	1	1	2	1	1
Number of general-purpose registers	16	16	8	40 - 520	32	32	40 - 520	32
Control memory size (Kbits)	420	480	246	—	—	—	—	—
Cache size (KBytes)	64	64	8	32	128	16-32	32	64


CISC
RISC
fritz