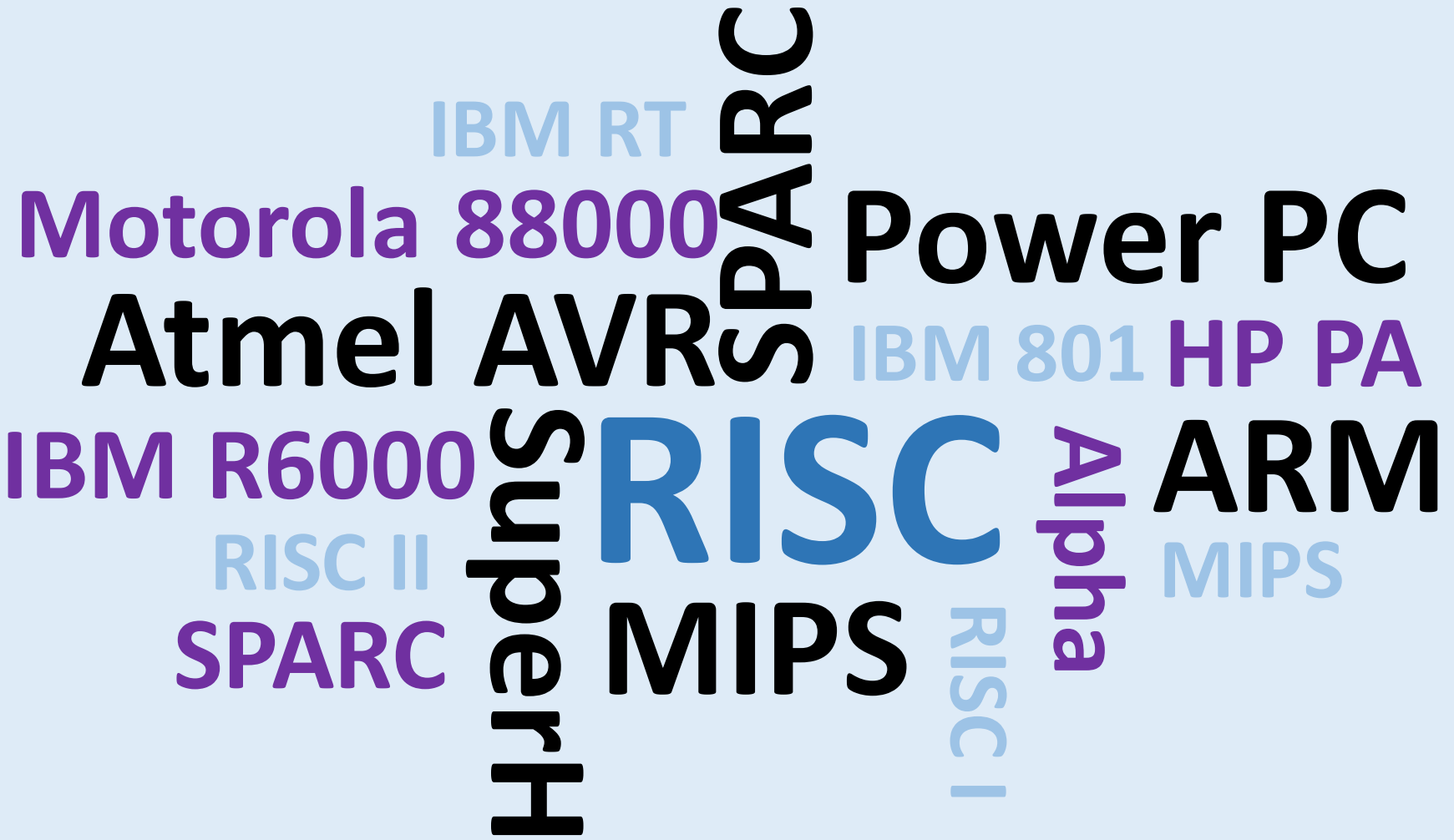
RISC

Reduced Instruction Set Computers



In this lesson:

- Introduce the principles of the RISC and CISC architectures
- Overview of the Berkeley RISC I microprocessor architecture
- Comparison of RISC and CISC architectures
- History of the development of RISC architectures

Reduced Instruction Set Computers (RISC)

Principle:

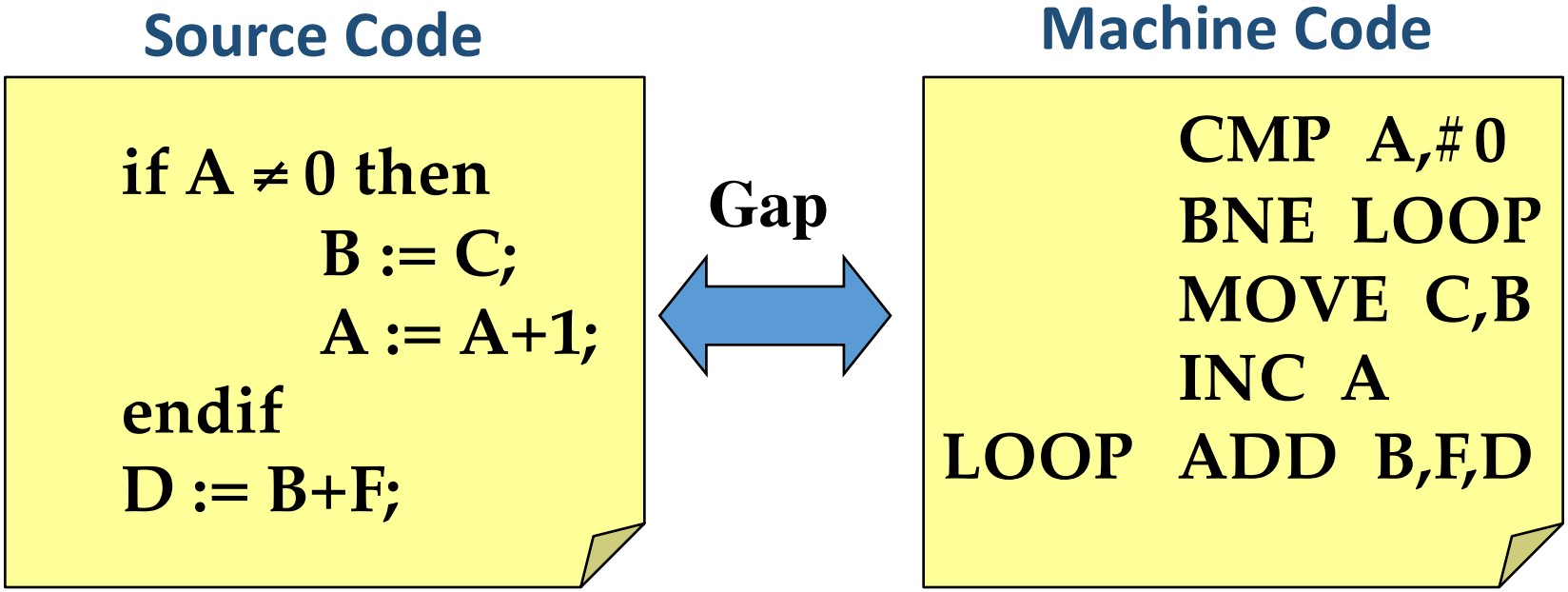
Support high level language structures that are frequently used and have a considerable impact in program execution time. The support to these structures is based on primitives architectural elements that facilitate the implementation of a fast, efficient and simple processor.

Complex Instruction Set Computers (CISC)

Principle:

Provide support for high level languages moving complex function to the hardware. Facilitate the development of compilers closing the “semantic gap”.

Semantic Gap



Semantic gap: How close is the machine code to the source code.

Closing the Semantic Gap on VAX-11

- **Case instruction** – for implementation of Case statements
- **Character string instructions** – for handling character strings
- **Call instruction** – for handling subroutines and functions
- **Compare and branch instructions** – for if/else statements

RISC-CISC Controversy

CISC supporters:

Microprocessors can be more efficient if they support complex instructions mapping high level language structures (close semantic gap).

RISC supporters:

Microprocessors can run faster by supporting simple instruction to implement complex high level language structures faster (open semantic gap).

Controversy resolved

Compatibility, Cost, Size and Energy Consumption decide

- The large majority of microprocessors installed in commercial products are RISC

- On the desktop market the large majority of microprocessors installed are CISC (**Intel® 64 and IA-32 Architectures**)

RISC vs CISC Characteristics (In the beginning)

	RISC	CISC
Number of Instructions	relatively small/simple	large/complex
Number of Data Type	small	relatively large
Registers	many general purpose	most have many general purpose
Memory Size	4G Bytes	4G Bytes or less
Memory access	only with load/store instructions	with many instructions
Operations (operands)	immediate or register	immediate, register or memory
Instruction Formats	few, fixed size	several, variable size
Addressing Modes	few	many

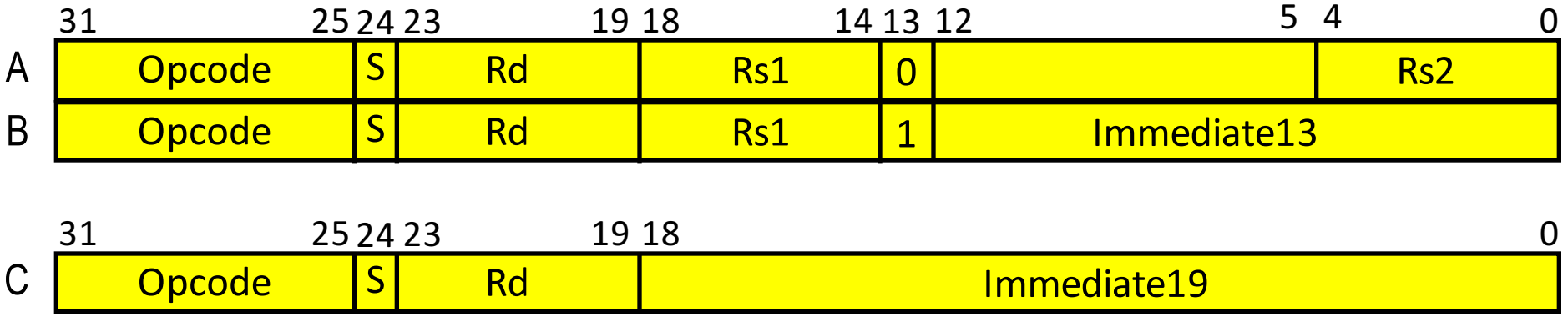
Berkeley's RISC I Architecture

- 31 instructions
  - *Delayed branch*
  - *Subroutines supported with Register Windows mechanism*
- Architectural memory of 2^{32} bytes
- Data types: byte, short word (16 bits), long word (32 bits)
- Addressing mode:

  Immediate, Register, Base-Index, Base-Displacement, Relative
- Registers
  - *32 general purpose visible registers (R0=0)*
  - *PC*
  - *PSW*
  - *Register Windows*

RISC I Instruction Set

Type	Instructions	Operation	Format
Arithmetic & Logic	ADD, ADDC, SUB, SUBC, SUBR, SUBRC, AND, OR, XOR	Rd ← Rs1 op S2	A and B
Shift	SLL, SRL, SRA	Rd ← Rs1 shifted by S2	A and B
Load	LDL, LDSU, LDSS, LDBU, LDDBS	Rd ← M[Rs1 + S2]	A and B
Store	STL, STS, STB	M[Rs1+S2] ← Rd	A and B
Control	JMP, JMPR, CALL, CALLR, RET, RETINT, CALLINT, GTLPC	Various	A, B, o C
Other	LDHI, GETPSW, PUTPSW	Various	A or C



Architectures Comparison

	RISC I	VAX-11	ARM
Number of Instructions	31 simple	130 simple & complex	140+ simple & complex
Number of Data Type	3	9	3
Registers	32 GPR	16 GPR	16 GPR
Memory Size	4G Bytes	4G Bytes	4G Bytes
Memory access	Load/Store instructions	Many instructions	Load/Store instructions
Operations (operands)	Immediate or register	Immediate, register or memory	Immediate or register
Instruction Formats	3 fixed sixe	5 variable size	10 fixed size
Addressing Modes	4	13	19

RISC: What's left

- Large number of registers
- Operations with operands kept in registers
- Memory accesses exclusively with load/store instructions
- Fixed-size instructions

Fallacies about RISC Architectures

- Reduced number of instructions
- Simple instructions

Not reduced instruction set computers by any standard

First RISC Architectures

Pioneers

- IBM 801
- Berkeley RISC I
- Stanford MIPS

First Commercial Generation

- SPARC
- DEC Alpha
- HP Precision
- IBM Power
- IBM RT

History of RISC Development

Pre RISC Era

1964 — Seymour Cray develops the CDC6600 that exhibited a load/store architectures with two addressing modes (base/indexed and base/displacement).

1975 — IBM starts the IBM 801 project, considered to be the first RISC architecture.

1976 — The Cray I, developed by Cary Research Inc., featured a load/store architecture like the CDC6600

1980 — IBM completes the IBM 801 project

History of RISC Development

Beginning of RISC Era

1981 IBM produces the first RISC microprocessor, but does not make it public.

1982 David Paterson and Carlo H. Sequin, fromUC Berkeley, present the RISC I microprocessor and coin the RISC concept.

1983 John Hennessy, from Stanford, presents the MIPS microprocessor

1985 The ARM (Advanced RISC Machine) architecture is born in England.

History of RISC Development

Beginning of Commercial RISC Era

1985: R2000, the first comercial MIPS microprocessor is developed. Eventually used in Silicon Graphics workstations.

1986: IBM introduces the IBM RT, the first RISC commercial workstation (very short life).

1986: HP sells computers with the Precision Architecture (HP PA)

1987: Sun introduces workstations with SPARC microprocessors (based on Berkeley's RISC II architecture). First successful commercial RISC computer.

History of RISC Development

The 90s

1990 IBM introduces de Power architecture (R6000 microprocessor).

1992 Digital introduces Alpha, the fastest microprocessor then. Die relatively soon.

1994 PowerPC is born. Developed by Apple, IBM, Motorola consortion

1997 Hitachi introduces the SH-1 (Super H), a 16 bit microprocessor.

Atmel introduces the first AVR microprocessor, an 8-bit architecture developed by Alf-Egil Bogen[2] and Vegard Wollan, students of the Norwegian Institute of Technology.

History of RISC Development

2016

The most successful, alive and well:

One British (ARM)

Three from USA (MIPS, PowerPC and Power)

One Norwegian (Atmel AVR)

One Japannese (Hitachi SuperH)

Most with 64-bit architectures.

2016

Contemporary RISC Microprocessors

Microprocessor	Applications
ARM	PALM, Nintendo DS, Game Boy Advance, iPhone, iPod Touch, iPad, iPhone, Android smart phone
MIPS	PlayStation Portable and routers
PowerPC	Playstation 3, Xbox 360 y Wii, IBM Seqouia
Hitachi SuperH	Consumer electronics
Atmel AVR	Xbox handheld controllers and BMW cars
IBM R6000	servers and supercomputers

Lesson Outcomes

- Understand the difference between the original RISC and CISC principles.
- Know the key characteristics of the RISC I architectures
- Name the pioneer RISC architectures and the first commercial RISC microprocessors and their developers.
- Name some of the contemporary commercial RISC microprocessors.
- Know the fundamental characteristics that contemporary RISC architectures preserve (*What's left*).
- Have a good idea of the timeline of the development of RISC architectures.