

### 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

**Source Code**

```
if A ≠ 0 then
  B := C;
  A := A+1;
endif
D := B+E;
```

Gap

↔

**Machine Code**

```
CMP A,#0
BNE LOOP
MOVE C,B
INC A
LOOP ADD B,E
```

**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).*

7

## 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**)

8

## 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

9

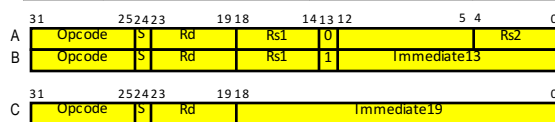
## Berkeley's RISC I Architecture

- 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

10

## RISC I Instruction Set

Type	Instructions	Operation	Format
Arithmetic & Logic	ADD, ADDC, SUB, SUBC, SUBR, SUBRC, AND, OR, XOR	$Rd \leftarrow Rs1 \text{ op } S2$	A and B
Shift	SLL, SRL, SRA	$Rd \leftarrow Rs1 \text{ shifted by } S2$	A and B
Load	LDL, LDSU, LDSS, LDBU, LDDBS	$Rd \leftarrow M[Rs1 + S2]$	A and B
Store	STL, STS, STB	$M[Rs1 + S2] \leftarrow Rd$	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 size	5 variable size	10 fixed size
Addressing Modes	4	13	19

12

## RISC: What's left

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

13

## Fallacies about RISC Architectures

- Reduced number of instructions
- Simple instructions

*Not reduced instruction set computers by any standard*

14

## First RISC Architectures

### Pioneers

- IBM 801
- Berkeley RISC I
- Stanford MIPS

### First Commercial Generation

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

15

## History of RISC Development

### Pre RISC Era

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

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

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

IBM completes the IBM 801 project

1964

1975

1976

1980

16

## History of RISC Development

### Beginning of RISC Era

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

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

John Hennessy, from Stanford, presents the MIPS microprocessor

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

1981 1982 1983 1985

17

## History of RISC Development

### Beginning of Commercial RISC Era

IR2000, the first commercial MIPS microprocessor is developed. Eventually used in Silicon Graphics workstations.

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

HP sells computers with the Precision Architecture (HP PA)

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

1985 1986 1987

18

## History of RISC Development

### The 90s

IBM introduces the Power architecture (R6000 microprocessor).

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

PowerPC is born. Developed by Apple, IBM, Motorola consortium

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.

1990 1992 1994 1997

19

## 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 Japanese (Hitachi SuperH)

Most with 64-bit architectures.

2016

20

## 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 Sequoia
Hitachi SuperH	Consumer electronics
Atmel AVR	Xbox handheld controllers and BMW cars
IBM R6000	servers and supercomputers

21

## 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.

22