

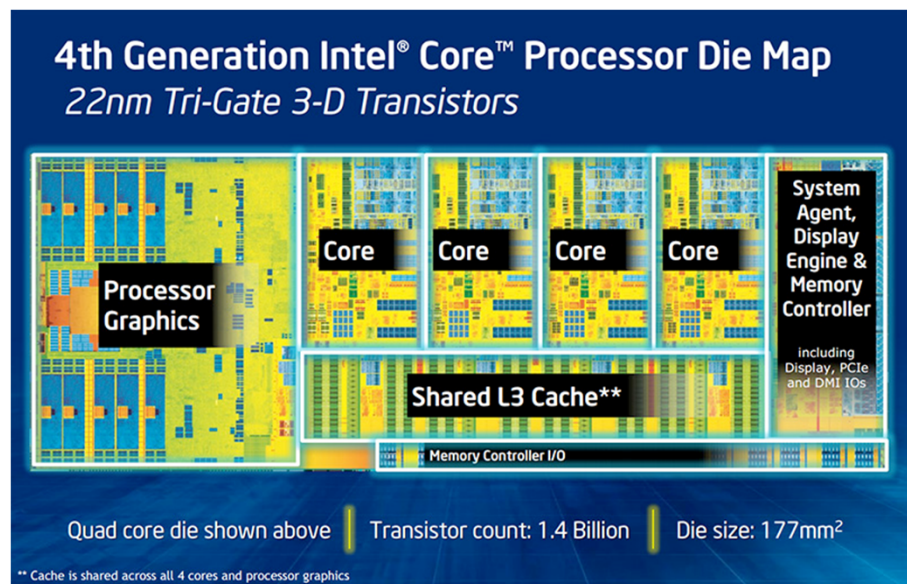
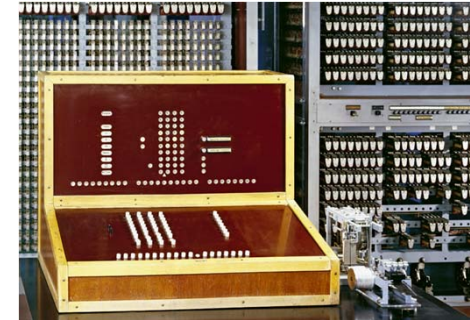
Computer Architectures

Computer Engineering 1

**CT Team: A. Gieriet, J. Gruber, R. Gübeli, M. Meli, M. Rosenthal,
A. Rüst, J. Scheier, M. Thaler**

Agenda

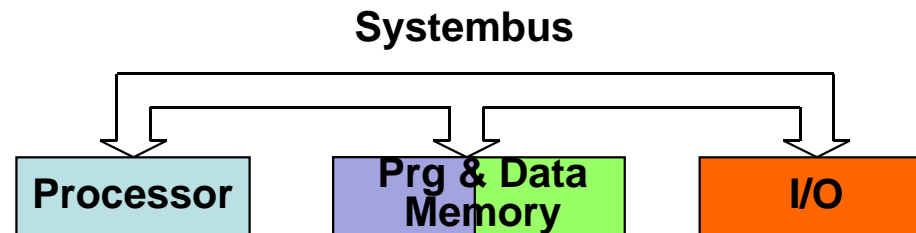
- Computer Architectures
- Von Neumann / Harvard
- RISC versus CISC
- Architecture Matrix



- **At the end of this lesson you will be able**
 - to explain different types of computer architectures
 - to understand the difference between von Neumann and Harvard architecture
 - to understand RISC and CISC paradigms

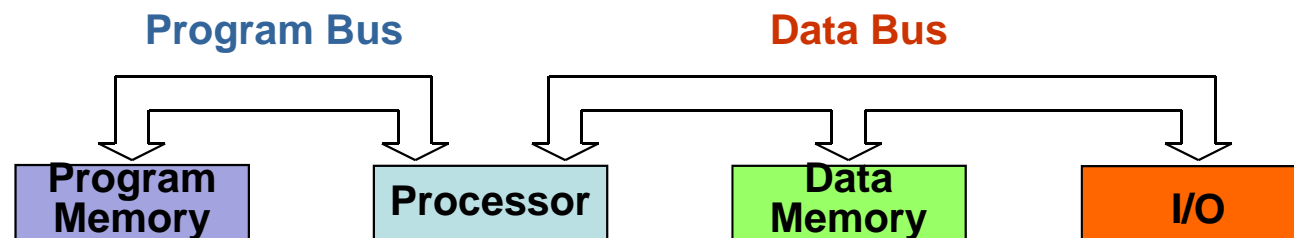
■ von Neumann Architecture

- Same memory holds program and data
- Single bus system between CPU and memory



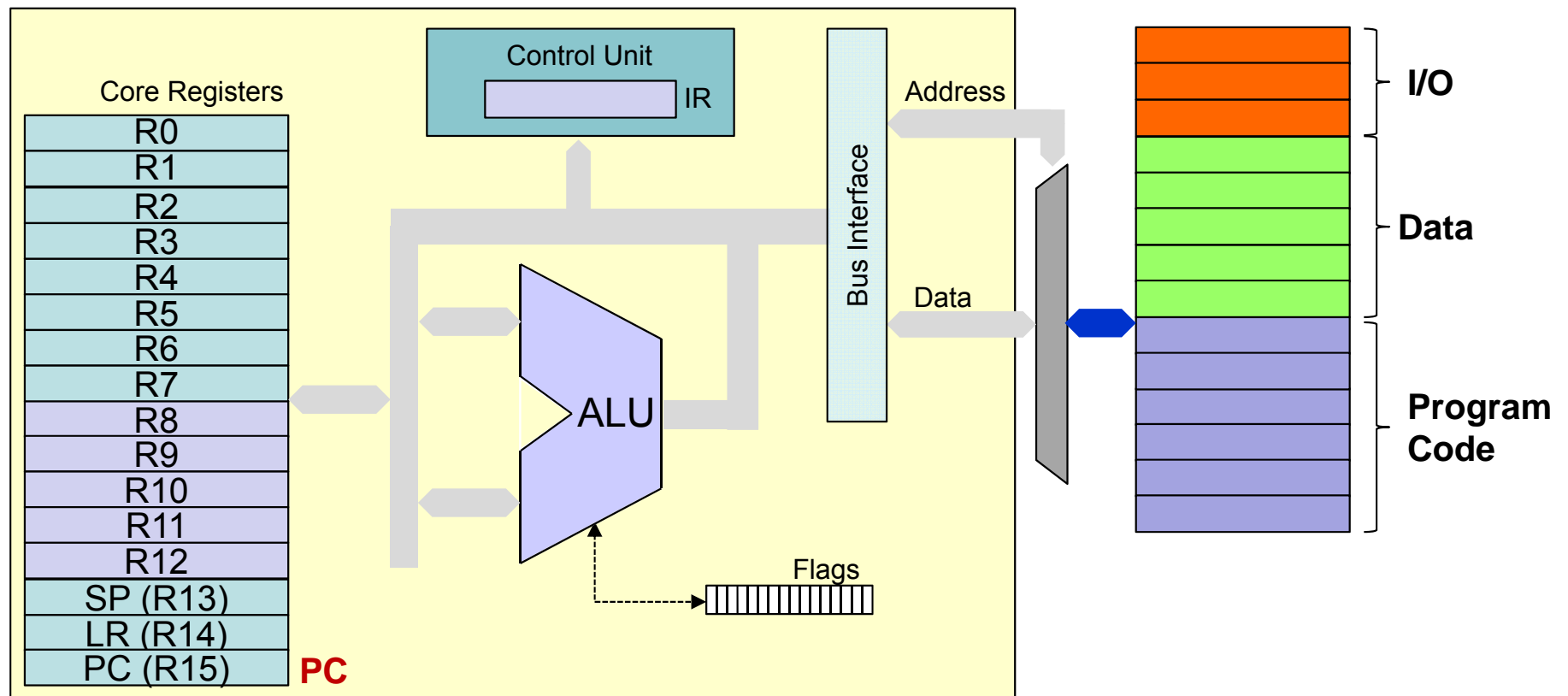
■ Harvard Architecture

- "Mark I" at Harvard University (Howard Aiken, 1939-44)
- Separate memories for program and data
- Two sets of address/data buses between CPU and memory



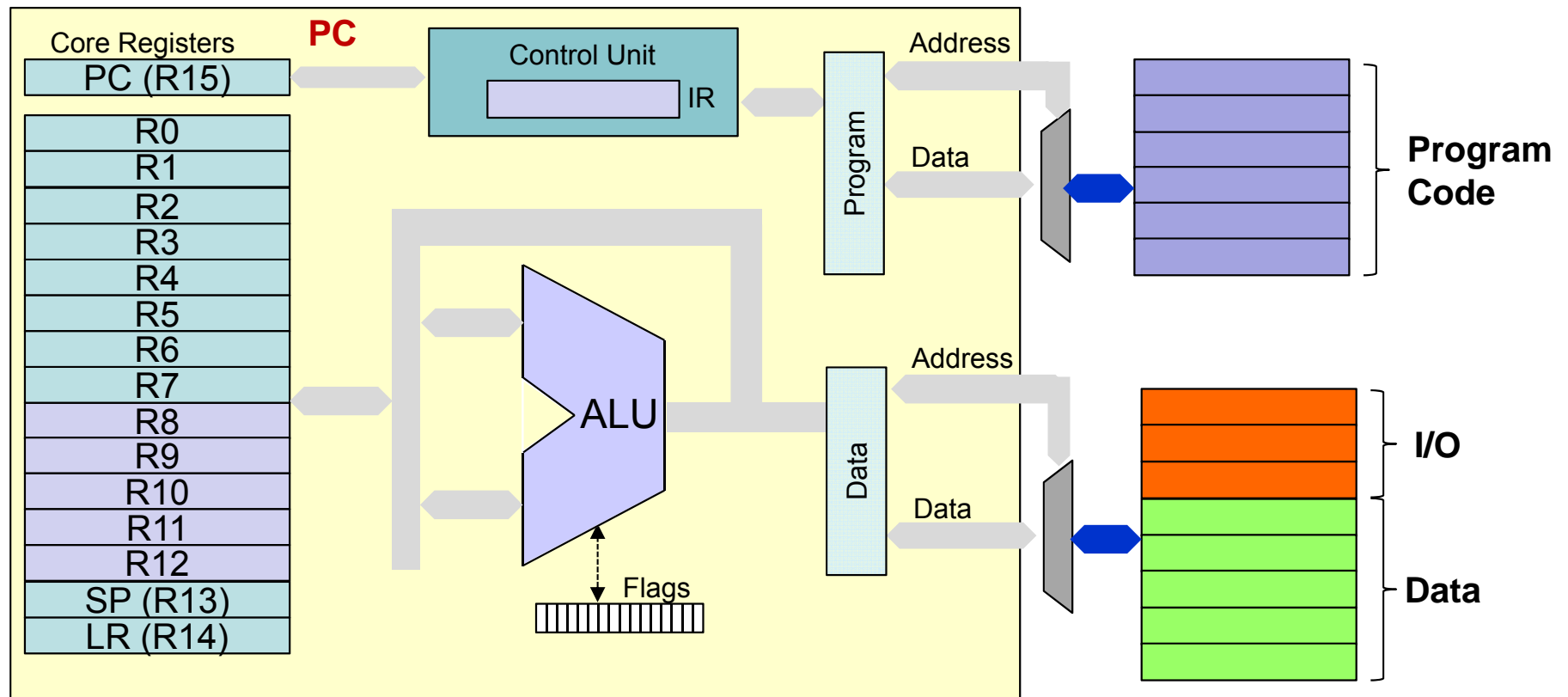
■ von Neumann Architecture

- Example: ARM Cortex-M0



■ Harvard Architecture

- Example: ARM Cortex-M3/M4



■ First CPUs

- Small address bus, small data bus (8 bit)
- Only few registers
- No cache, pipelining, co-processors, multi-core
- Machine code as finite state machine
 - Shift, multiply, etc. in loops

■ Consequences for machine code

- Differing complexity
- Different length
- Variable execution time
- Individual addressing / arguments

} for different instructions

■ Increasing demands

- Address bus and data bus size
- Machine code needs more space for addressing
- More variants of instructions

→ More and more complex instructions (CISC)

■ Problem

- Most instructions were not handled by compilers

■ Introduction of RISC

- 1974 – John Cocke (IBM) proved:
80% of work was done using only 20% of the instructions
- The first reduced instruction set (RISC) projects
 - IBM 801 machine (1974)
 - Berkeley's RISC-I and RISC-II processors (1980)
 - Stanford's MIPS processor (1981)
- 1986 – announcement of first commercial RISC chip
 - Well known RISC processors: ARM, MIPS, PowerPC, SPARC

Remark: Word “CISC” was introduced with “RISC”

■ RISC versus CISC

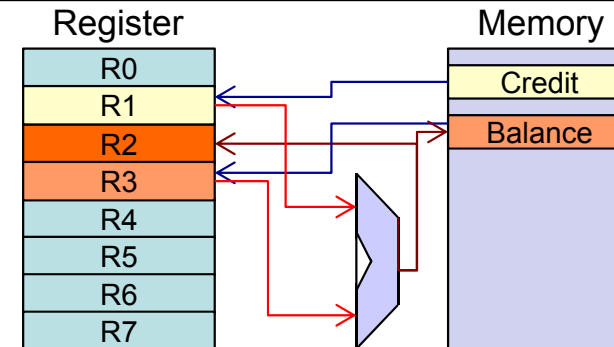
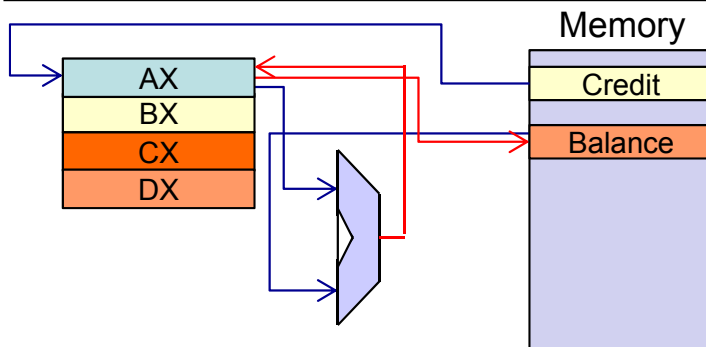
- RISC: LOAD / STORE Architecture
 - Memory → Register → Memory
 - Data processing instructions: only on Registers

CISC: x86

```
MOV  AX, Credit
ADD  AX, Balance
MOV  Balance, AX
```

RISC: ARM Cortex-M0

```
LDR  R0, =Credit
LDR  R1, [R0]
LDR  R0, =Balance
LDR  R3, [R0]
ADDS R2, R1, R3
STR  R2, [R0]
```



Complex Instruction Set Computer (CISC)

- Traditional memory access
- Complex addressing
- High code density
- Most compilers support only sub set of instructions
 - Powerful instructions are often not used at all
- Often require manual optimization of assembly code for embedded systems
- Program needs to wait for external memory

Reduced Instruction Set Computer (RISC)

- ↔ ■ Load-store architecture
- ↔ ■ Only simple addressing
- ↔ ■ More lines of code
- ↔ ■ Reduced instruction set
 - Requires less hardware
 - Allows higher clock rates
 - Use Silicon area for more registers
- ↔ ■ Allow effective compiler optimization with limited, generic instructions
- ↔ ■ Program works on registers at full-speed

■ RISC versus CISC

- Advantages RISC
 - Simple Hardware: for more registers/cache on silicon area
 - ▶ Less accesses to memory
 - Several data paths possible (Superscalar Computers)
 - Easy pipelining
 - Higher clock frequencies
 - ▶ Less hardware involved per cycle
- Advantages CISC
 - Less program memory needed with complex instructions
 - Short programs may work faster with less memory accesses
- CISC and RISC architectures are more and more indistinguishable

■ Differentiation of architectures

	Von Neumann	Harvard
RISC	ARM Cortex-M0 Series	ARM Cortex-M3/M4 Series (Harvard Cache Architecture)
CISC	x86	Specialized Processors (e.g. DSP)