Basic Computer Architecture

CSCE 496/896: Embedded Systems

Witawas Srisa-an

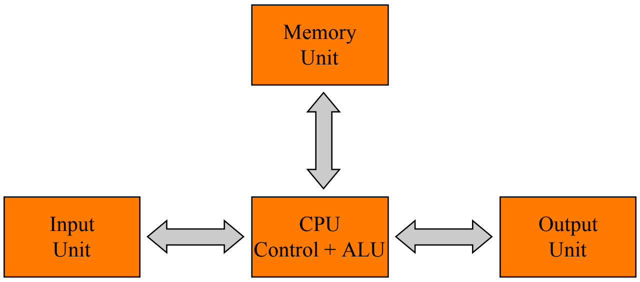
Review of Computer Architecture

- Credit: Most of the slides are made by Prof. Wayne Wolf who is the author of the textbook.
- I made some modifications to the note for clarity.
  - Assume some background information from CSCE 430 or equivalent

von Neumann architecture

- Memory holds data and instructions.
- Central processing unit (CPU) fetches instructions from memory.
    - Separate CPU and memory distinguishes programmable computer.
- CPU registers help out: program counter (PC), instruction register (IR), general-purpose registers, etc.

von Neumann Architecture



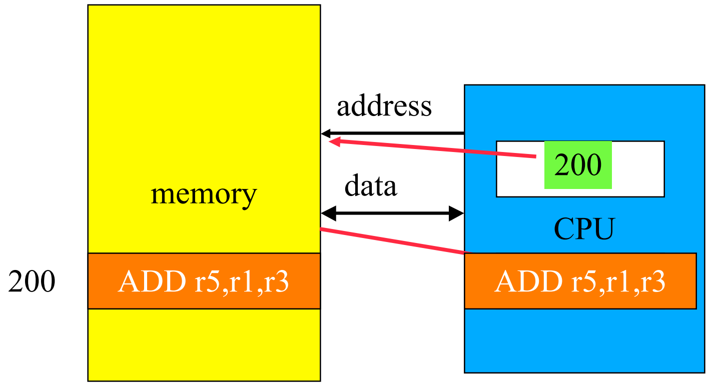
Memory Unit

Input Unit

CPU Control + ALU

Output Unit

CPU + memory



address

data

memory

200

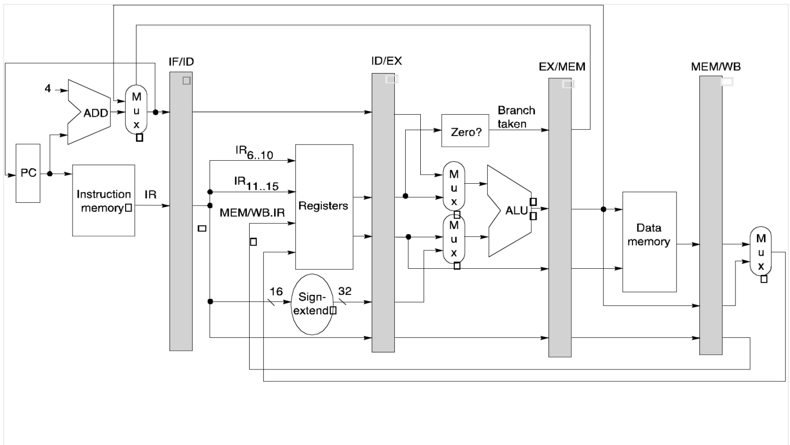
ADD r5,r1,r3

200

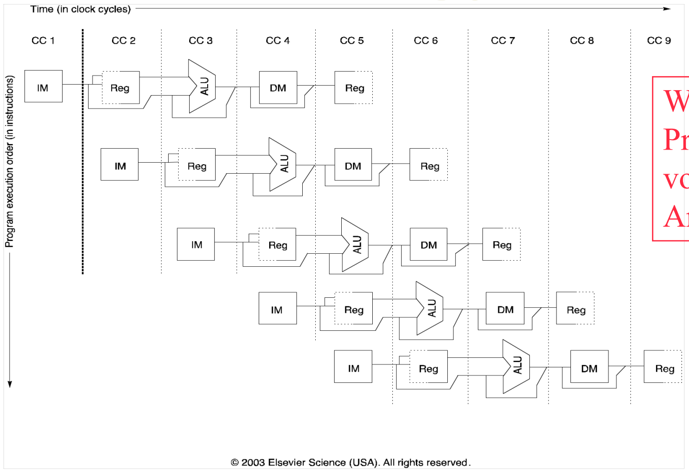
CPU

ADD r5,r1,r3

Recalling Pipelining



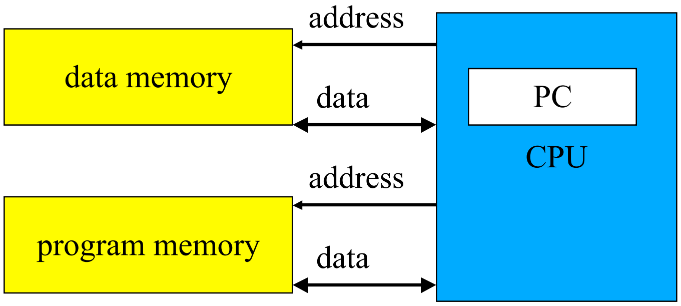
Recalling Pipelining



© 2003 Elsevier Science (USA). All rights reserved.

What is a potential Problem with von Neumann Architecture?

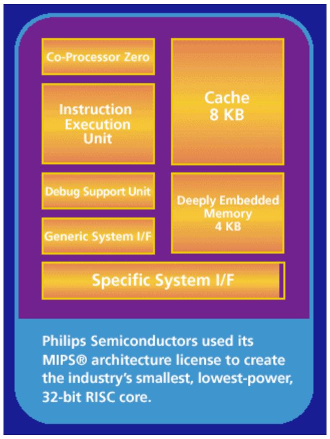
Harvard architecture



von Neumann vs. Harvard

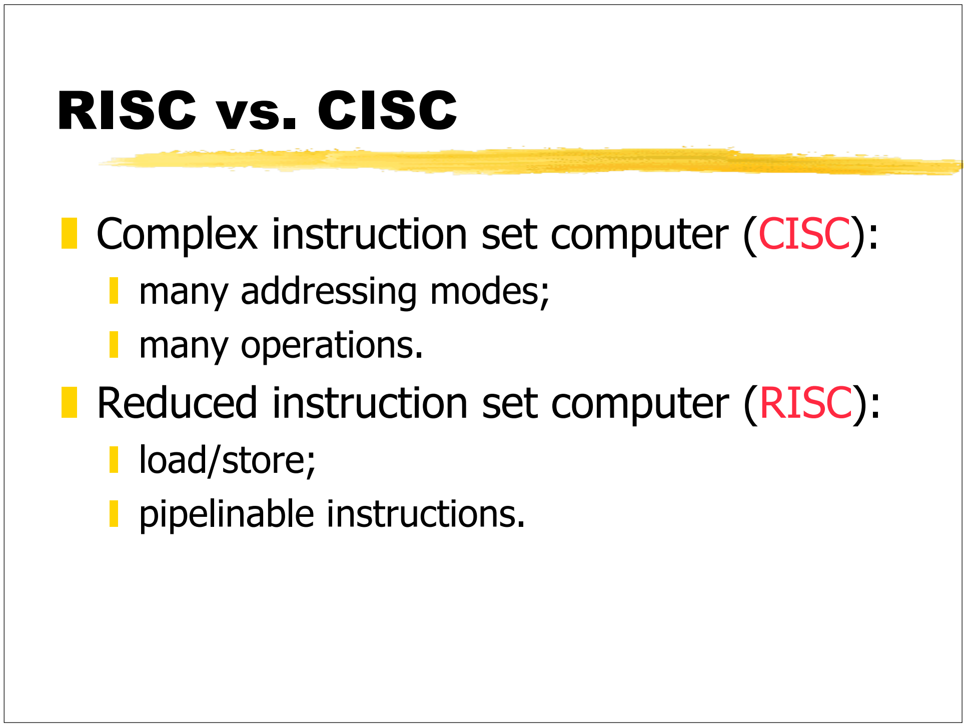
- Harvard can’t use self-modifying code.
- Harvard allows two simultaneous memory fetches.
- Most DSPs (e.g Blackfin from ADI) use Harvard architecture for streaming data:
  - greater memory bandwidth.
  - different memory bit depths between instruction and data.
  - more predictable bandwidth.

Today’s Processors



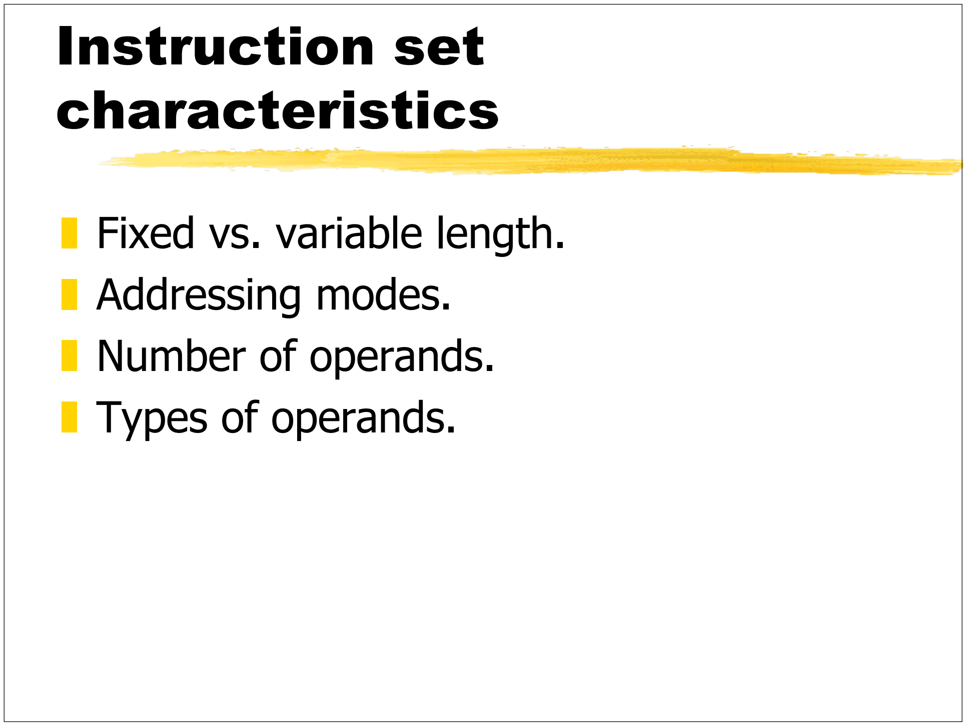
Philips Semiconductors used its MIPS® architecture license to create the industry’s smallest, lowest-power, 32-bit RISC core.

Harvard or von Neumann?



RISC vs. CISC

- Complex instruction set computer (CISC):
  - many addressing modes;
  - many operations.
- Reduced instruction set computer (RISC):
  - load/store;
  - pipelinable instructions.

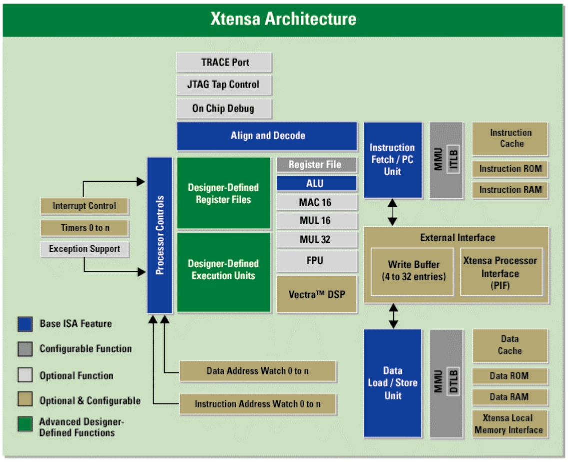


Instruction set characteristics

- Fixed vs. variable length.
- Addressing modes.
- Number of operands.
- Types of operands.

Tensilica Xtensa

- RISC based variable length
  - But not CISC



Programming model

- Programming model: registers visible to the programmer.
- Some registers are not visible (IR).

Multiple implementations

- Successful architectures have several implementations:
  - varying clock speeds;
  - different bus widths;
  - different cache sizes, associativities, configurations;
  - local memory, etc.

Assembly language

- One-to-one with instructions (more or less).
- Basic features:
  - One instruction per line.
  - Labels provide names for addresses (usually in first column).
  - Instructions often start in later columns.
  - Columns run to end of line.

ARM assembly language example

```
label1    ADR r4,c
          LDR r0,[r4] ; a comment
          ADR r4,d
          LDR r1,[r4]
          SUB r0,r0,r1 ; comment
```

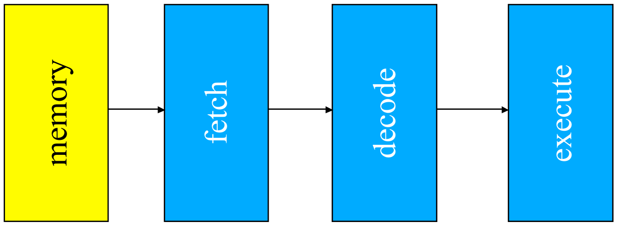
destination

Pseudo-ops

- Some assembler directives don't correspond directly to instructions:
  - Define current address.
  - Reserve storage.
  - Constants.

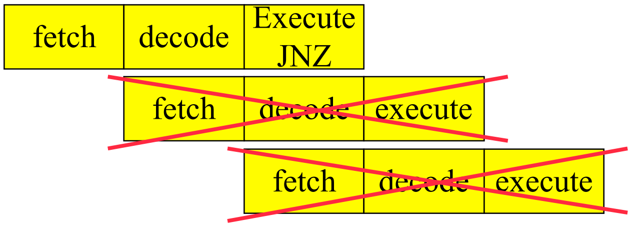
Pipelining

- Execute several instructions simultaneously but at different stages.
- Simple three-stage pipe:



Pipeline complications

- May not always be able to predict the next instruction:
  - Conditional branch.
- Causes bubble in the pipeline:



Superscalar

- RISC pipeline executes one instruction per clock cycle (usually).
- Superscalar machines execute multiple instructions per clock cycle.
  - Faster execution.
  - More variability in execution times.
  - More expensive CPU.

Simple superscalar

- Execute floating point and integer instruction at the same time.
  - Use different registers.
  - Floating point operations use their own hardware unit.
- Must wait for completion when floating point, integer units communicate.

Costs

- Good news---can find parallelism at run time.
  - Bad news---causes variations in execution time.
- Requires a lot of hardware.
  - n^2 instruction unit hardware for n-instruction parallelism.

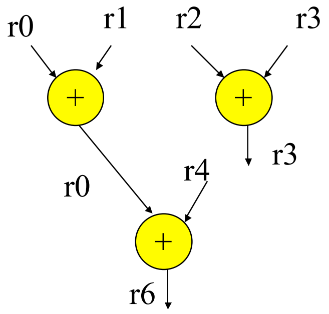
Finding parallelism

- Independent operations can be performed in parallel:

  ADD r0, r0, r1

  ADD r3, r2, r3

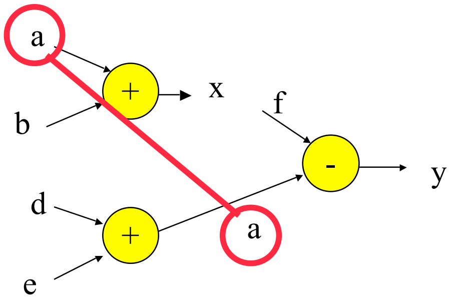
  ADD r6, r4, r0



Pipeline hazards

- Two operations that have data dependency cannot be executed in parallel:

```
x = a + b;
a = d + e;
y = a - f;
```



Order of execution

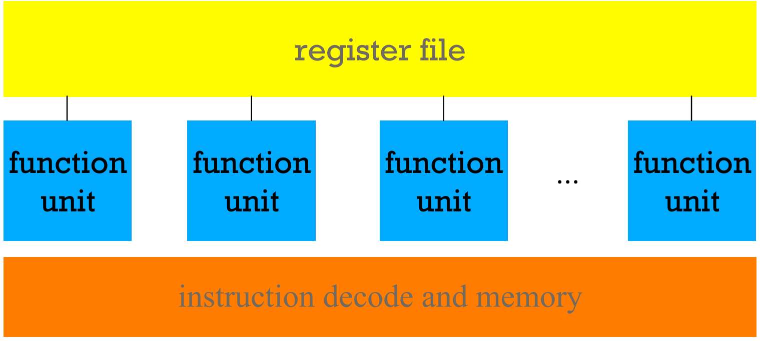
- In-order:
  - Machine stops issuing instructions when the next instruction can’t be dispatched.
- Out-of-order:
  - Machine will change order of instructions to keep dispatching.
  - Substantially faster but also more complex.

VLIW architectures

- Very long instruction word (VLIW) processing provides significant parallelism.
- Rely on compilers to identify parallelism.

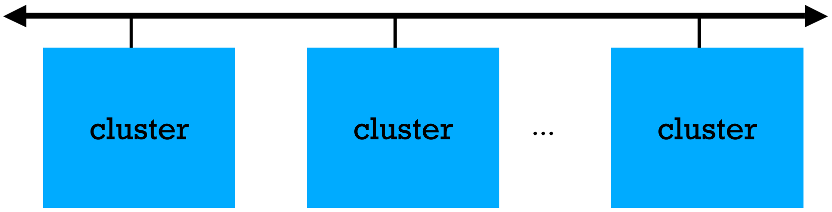
What is VLIW?

- Parallel function units with shared register file:



VLIW cluster

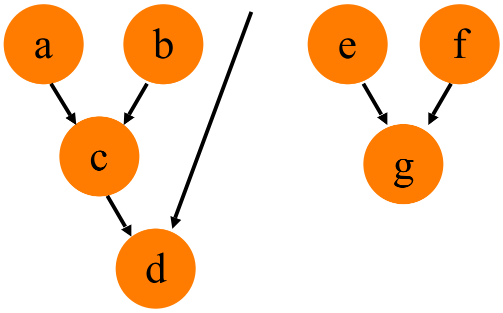
- Organized into clusters to accommodate available register bandwidth:



VLIW and compilers

- VLIW requires considerably more sophisticated compiler technology than traditional architectures---must be able to extract parallelism to keep the instructions full.
- Many VLIWs have good compiler support.

Scheduling



expressions

a	b	e
f	c	nop
d	g	nop

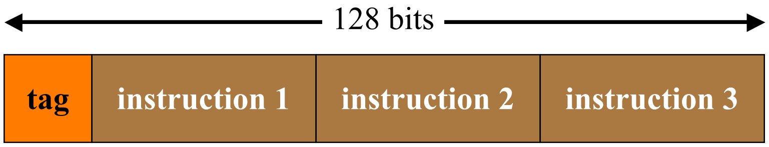
instructions

EPIC

- EPIC = Explicitly parallel instruction computing.
- Used in Intel/HP Merced (IA-64) machine.
- Incorporates several features to allow machine to find, exploit increased parallelism.

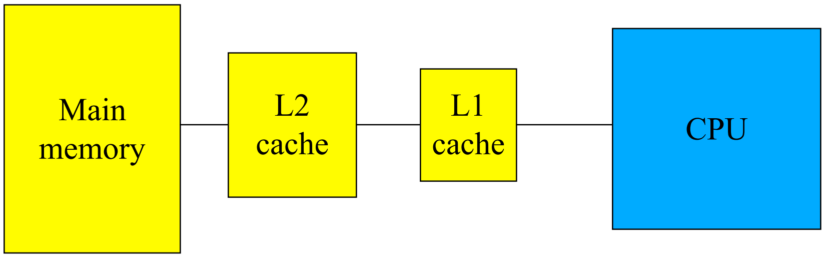
IA-64 instruction format

- Instructions are bundled with tag to indicate which instructions can be executed in parallel:



Memory system

- CPU fetches data, instructions from a memory hierarchy:



Memory hierarchy complications

- Program behavior is much more state-dependent.
  - Depends on how earlier execution left the cache.
- Execution time is less predictable.
  - Memory access times can vary by 100X.

Memory Hierarchy Complication

	Pentium 3-M	Pentium 4-M	Pentium M
Core	P6 (Tualatin 0.13µ)	Netburst (Northwood 0.13µ)	"P6+" (Banias 0.13µ, Dothan 0.09µ)
L1 Cache (data + code)	16Kb + 16Kb	8Kb + 12Kµops (TC)	32Kb + 32Kb
L2 Cache	512Kb	512Kb	1024Kb
Instructions Sets	MMX, SSE	MMX, SSE, SSE2	MMX, SSE, SSE2
Max frequencies (CPU/FSB)	1.2GHz 133MHz	2.4GHz 400MHz (QDR)	2GHz 400MHz (QDR)
Number of transistors	44M	55M	77M, 140M
SpeedStep	2nd generation	2nd generation	3rd generation

End of Overview

- Next class: Altera Nios II processors