Very abstracted view of instruction execution
Very abstracted view of instruction execution

Memory

CPU

Control

ALU

PC

state

Data path

Program

Data

0x10093826

Sub $7,$8,$9

control

Registers

0x10093826

000..., 00119

001... 0001

(addi $4,$12,33)

0x21840021

(addi $4,$12,33)

0x21840021

(addi $4,$12,33)
Superscalar Processors

• MIPS 3000: scalar processor, i.e., one instruction at a time in pipeline

• Newer processors expand the concept in:
  – Width: there are several pipelines from the EX stage on hence the name *superscalar*
  – Depth: Each pipeline has more stages

• The pipeline consists of:
  – A *front-end* (IF + ID) that can fetch and decode several instructions concurrently
  – A *back-end* (EX + MEM) that consists of several pipelines
  – The WB stage must be such that the processor state is modified according to the original program order.
Two Types of Superscalar

• In-order processors:
  – Instructions leave the front-end in strict program order
  – All dependencies are resolved at the last stage of the front-end
  – Good performance relies on optimized compilers

• Out-of-order processors
  – Instructions can execute and complete their execution out-of-order
  – However, need to replace (extend) the WB stage by a *Commit* stage that ensures that results are stored in the process state in-order
  – Good performance relies on extensive hardware logic
DEC Alpha 21164 (in-order)
Alpha 21164 Pipeline

IF and ID

Front-end

S0  S1  S2  S3

S4  S5  S6

Integer

S4  S5  S6  S7  S8

Floating-point

S4  S5  S6  S7  S8  S9  S10  S11  S12

L1 Cache access  L2 cache access

EX, Mem and WB

Back-end
High-level View of the P6 Microarchitecture
Block Diagram of the P6 Microarchitecture