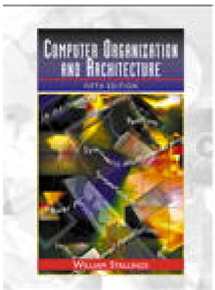
Campus de Gualtar
4710-057 Braga

UNIVERSIDADE DO MINHO
ESCOLA DE ENGENHARIA

Departamento de
Informática



Computer Organization and Architecture

5th Edition, 2000

by William Stallings

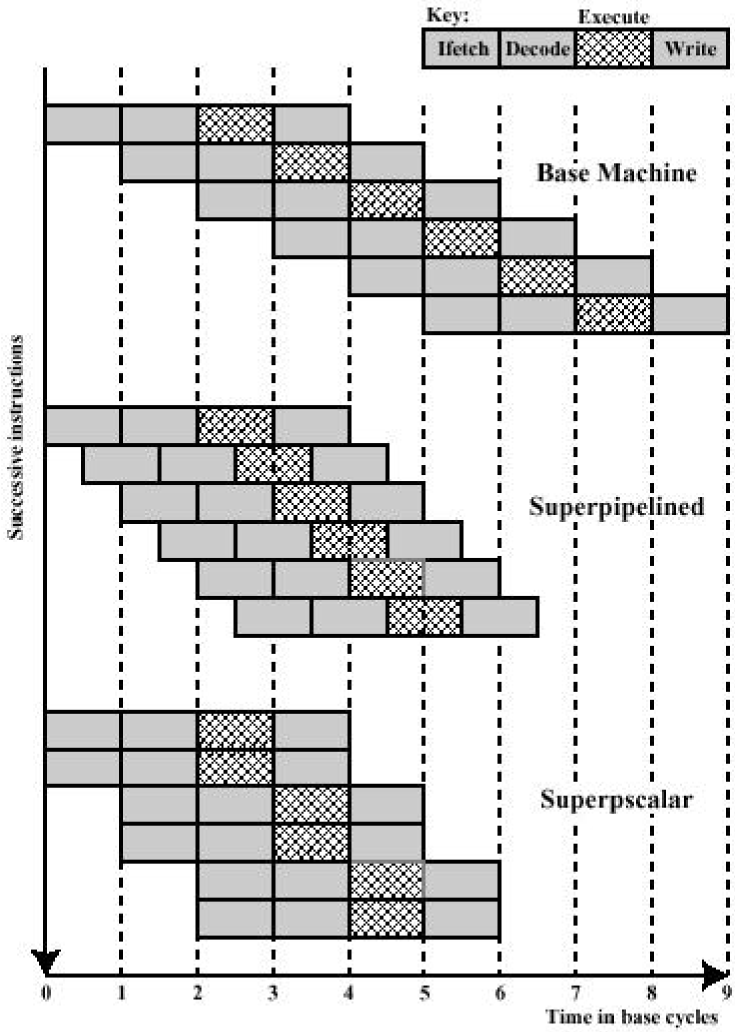
Table of Contents

III. THE CENTRAL PROCESSING UNIT.

8. ...
9. ...
10. ...
11. ...
12. …
13. Instruction-Level Parallelism and Superscalar Processors. (5-May-01)

Overview (13.1)

- Superscalar refers to a machine that is designed to improve the performance of the execution of scalar instructions
    - This is as opposed to vector processors, which achieve performance gains through parallel computation of elements of homogenous structures (such as vectors and arrays)
    - The essence of the superscalar approach is the ability to execute instructions independently in different pipelines, and in an order different from the program order.
    - In general terms, there are multiple functional units, each of which is implemented as a pipeline, which support parallel execution of several instructions.
- Superscalar vs. Superpipelined
    - Superpipeline falls behind the superscalar processor at the start of the program and at each branch target.



- Limitations
  - o Superscalar approach depends on the ability to execute multiple instructions in parallel.
  - o Instruction-level parallelism refers to the degree to which, on average, the instructions of a program can be executed in parallel.

- Fundamental limitations to parallelism (to which we apply compiler-based optimization and hardware techniques)
  - o True data dependency
    - ▪ Also called flow dependency or write-read dependency
    - ▪ Caused when one instruction needs data produced by a previous instruction
  - o Procedural dependency
    - ▪ Usually caused by branches, i.e. the instructions following a branch (taken or not taken) cannot be executed until the branch is executed
    - ▪ Variable length instructions cause a procedural dependency because the instruction must be at least partially decoded (to determine its length) before the next instruction can be fetched.
  - o Resource conflicts
    - ▪ A competition of two or more instructions for the same resource oat the same time.
    - ▪ Resources include memories, caches, buses, register-file ports, and functional units.
    - ▪ Similar to data dependency, but can be
      - ▪ overcome by duplication of resources
      - ▪ minimized by pipelining the appropriate functional unit (when an operation takes a long time)
  - o Output dependency
    - ▪ Only occurs when instructions may be completed out of order
    - ▪ Occurs when two instructions both change the same register or memory location, and a subsequent instruction references that data. The order of those two instructions must be preserved.
  - o Antidependency
    - ▪ Only occurs when instructions may be issued out of order
    - ▪ Similar to a true data dependency, but reversed
    - ▪ Instead of the first instruction producing a value that the second instruction uses, the second instruction destroys a value that the first instruction uses