

DATORARKITEKTUR

Processor



Dr. Andreas de Blanche
andreas.de-blanche@hv.se

Datorarkitektur, 2022-08-18

Moore's Law

The complexity for minimum component costs has increased at a rate of roughly a factor of two per year. Certainly over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years.

- Gordon Moore, 1965

G. Moore, "Cramming more components onto integrated circuits", Electronics Magazine, McGraw-Hill, April 19, 1965.



Gordon Moore, 2004

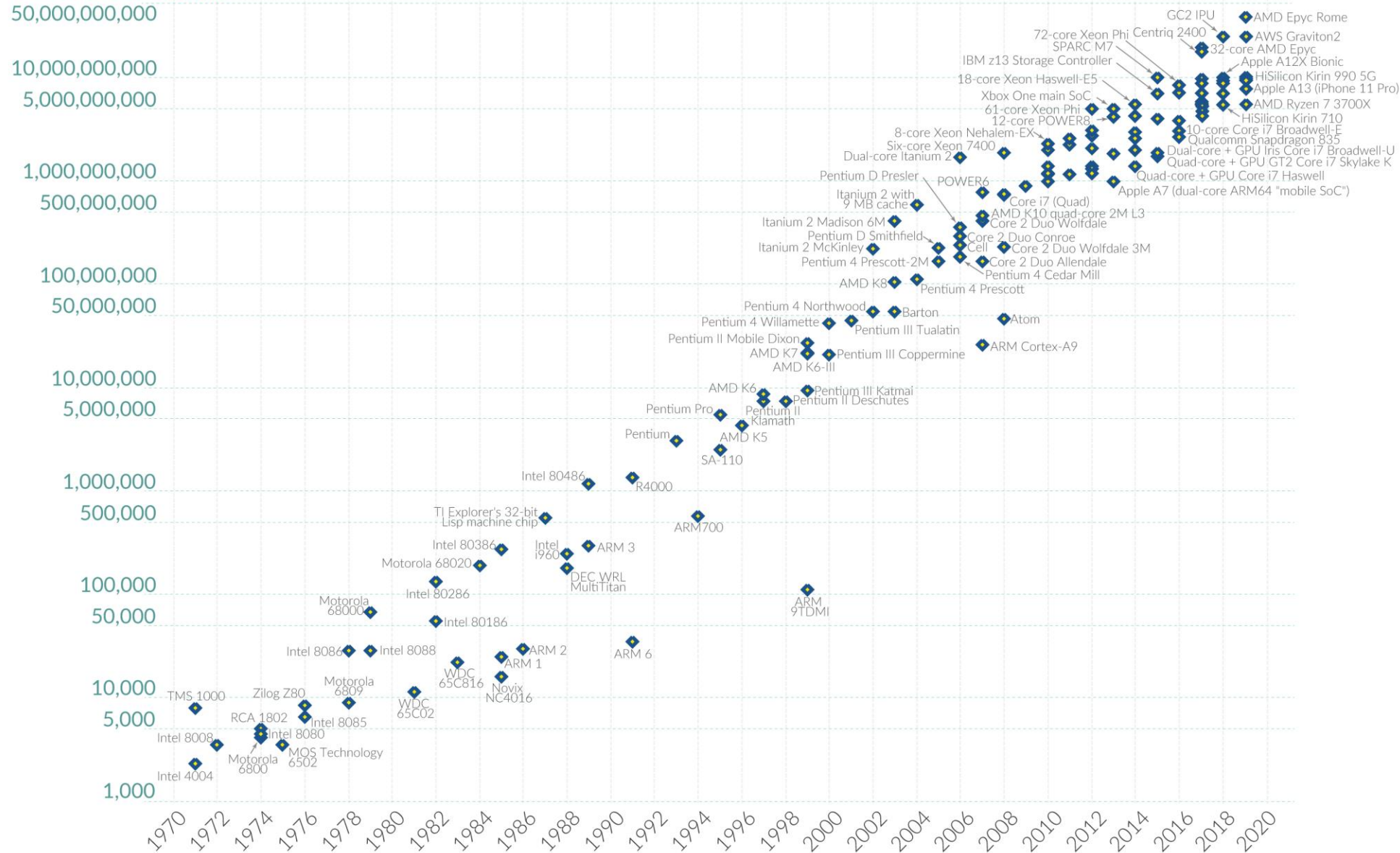
Moore's Law: The number of transistors on microchips doubles every two years

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

Our World
in Data



Transistor count



Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

OurWorldinData.org – Research and data to make progress against the world's largest problems.

Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

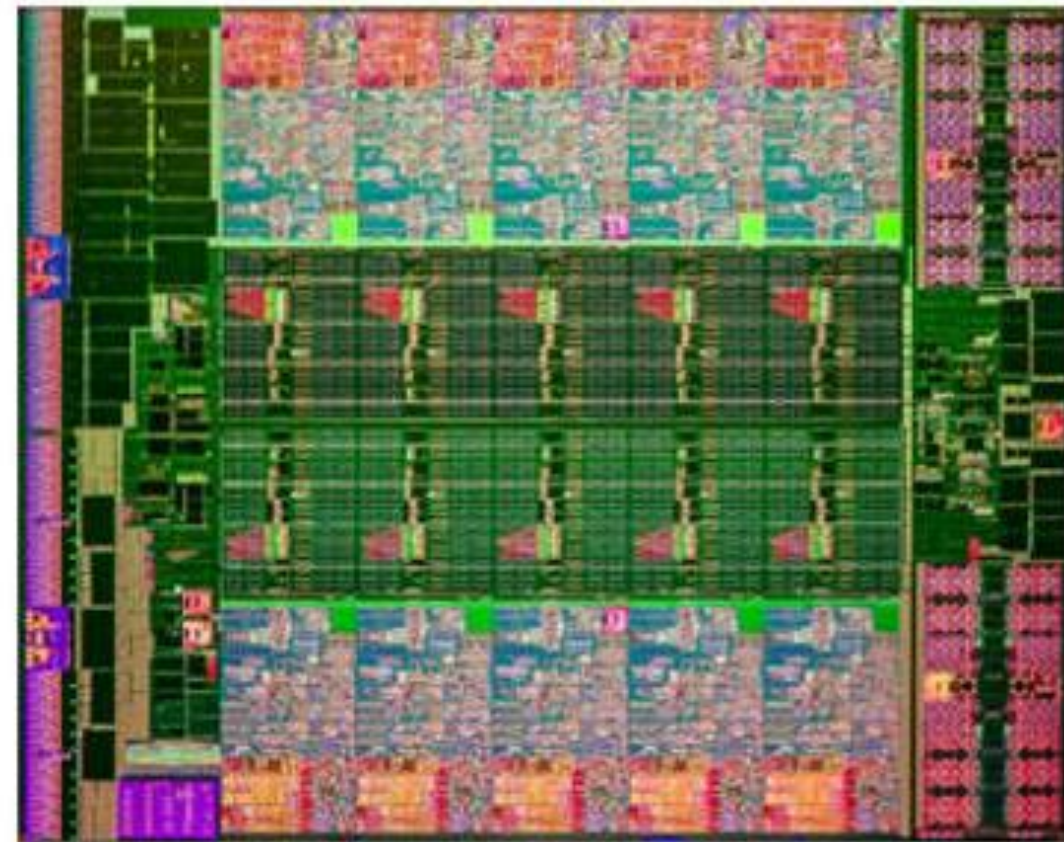
Dr. Andreas de Blanche
DAK101, processor, 2022

TRANSISTOR BUDGET

Trots att en modern processor består av miljarder transistorer så är det inte nog!

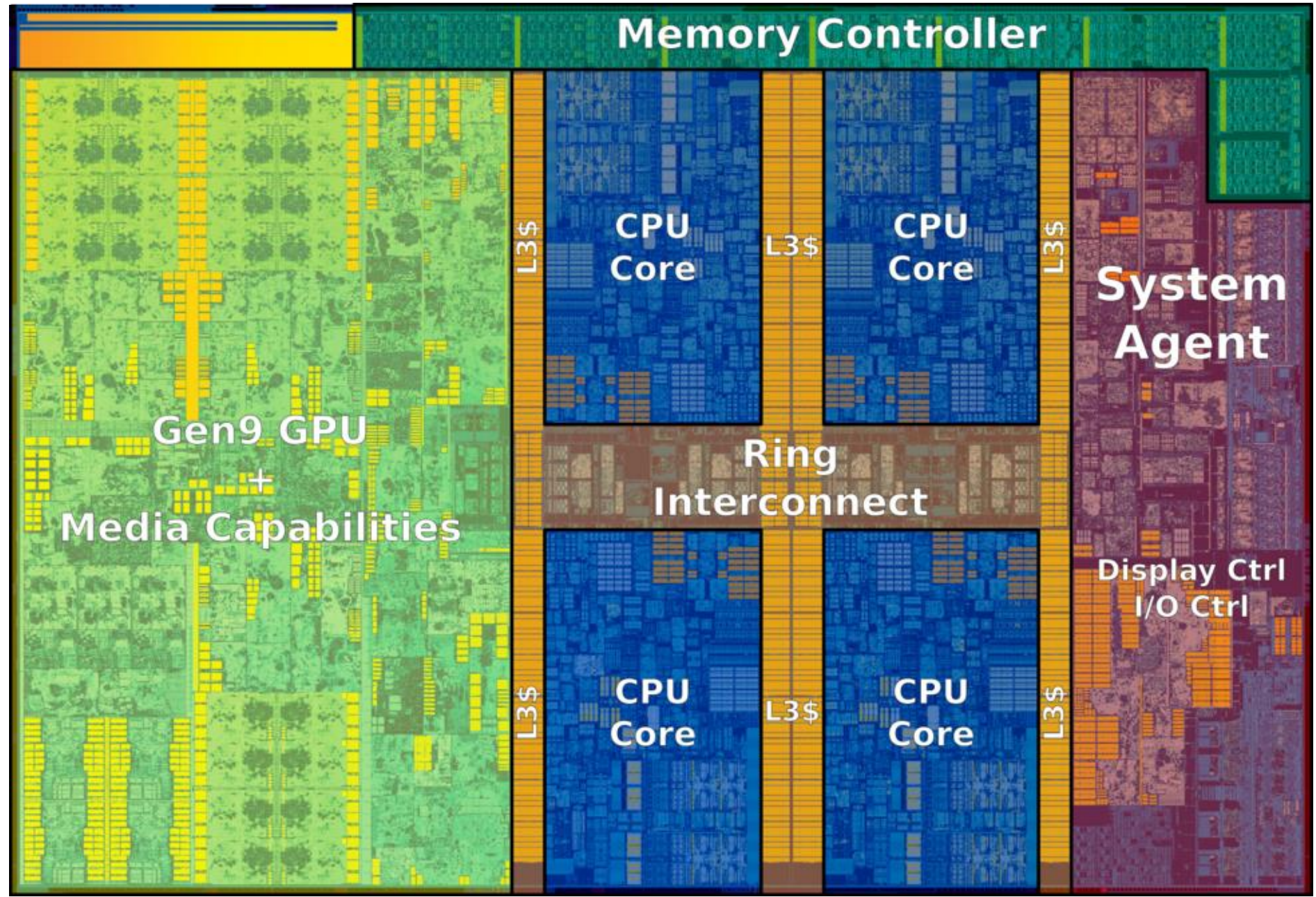
- En stor del går dock åt till cache.
- Multipla cores

Anledningen till att “saker” inte kommer med är ofta att det inte fanns transistorer till det.



Xeon E5-2600 v2 processor by Intel

Quad Core GT2 Skylake by Intel



DATORARKITEKTUR



I en processor



INSTRUCTION SET ARCHITECTURE ISA



Ett datorprogram består av väldigt många små instruktioner.

En instruktion är ett binärt värde. Ex:

01010101 01010101 01010101 01010111

Denna är delad i opkod och operand(er).

HUR GÖR PROCESSORN?



Fetch

Decode

Execute

- Fetch: hämta en instruktion
- Decode: avkoda instruktionen
- Execute: kör instruktionen

Instruktion är till exempel

- Fyra räknesätten
- NOT, AND, OR, XOR
- Kopiera data
- Jämföra
- kontrollfunktioner

INSTRUKTIONER OCH DATA



Fetch

L1 Cache

Decode

Register file

Execute

Instruktionerna är lagrade i minnet (dvs programmet)

Instruktionerna arbetar på data från register eller minne.

Ett register är en lagringsplats i processorn, vanligtvis några byte stort, ex; 64 bitar.

Instruktioner läser från och skriver till register och minnesadresser.

KLOCKCYKLER

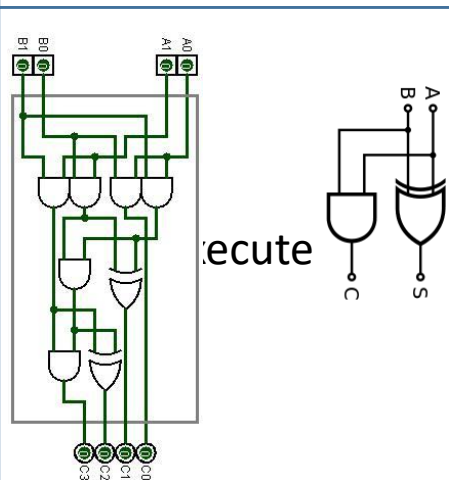
Fetch

L1 Cache

Decode

2*2bit

1+1 bit



Register file

Processorn arbetar efter en pulsgenerator, en klocka.

Fetch-decode-execute är byggt med digitalteknik som uppdateras varje gång det kommer en klockpuls.

Olika instruktioner tar olika vägar genom processorns logik.

Den längsta vägen genom processorn bestämmer hur kort en klockcykel kan vara.

Alla transistorer på vägen måste hinna uppdateras.

KLOCKCYKLER

Säga att en instruktion tar en klockcykel att genomföra.

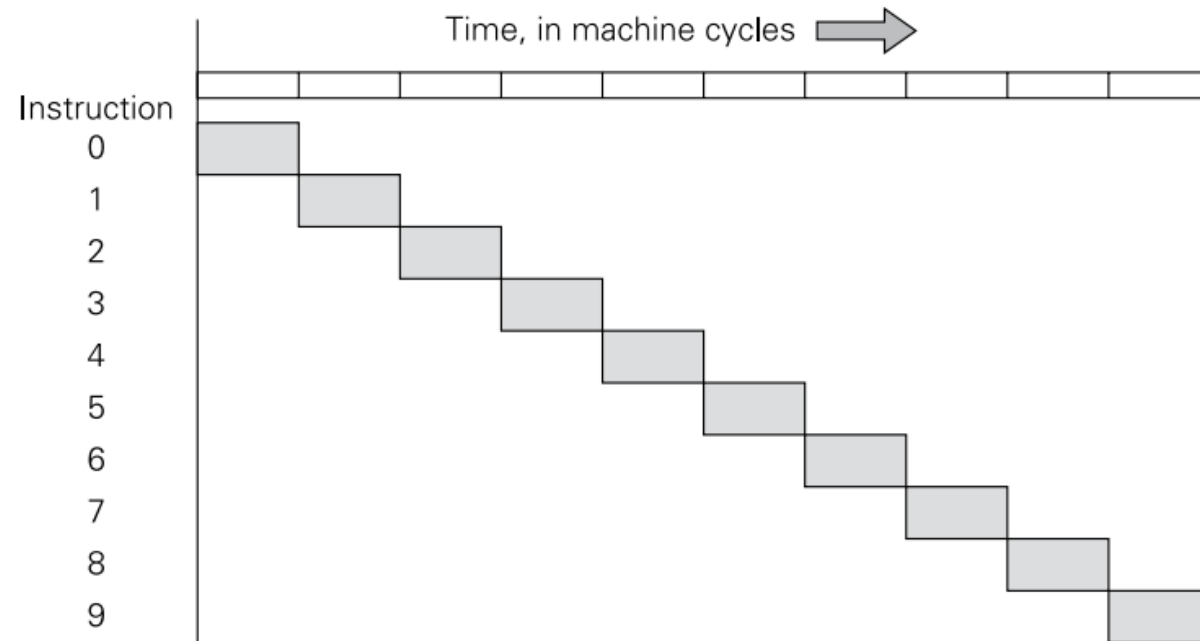


FIGURE 4-8: Single-cycle machine instructions

Den längsta vägen genom processor är 0.001 sekund.

Max processor frekvens: 1000 Hz.

L1 Cache

Register file

Fetch

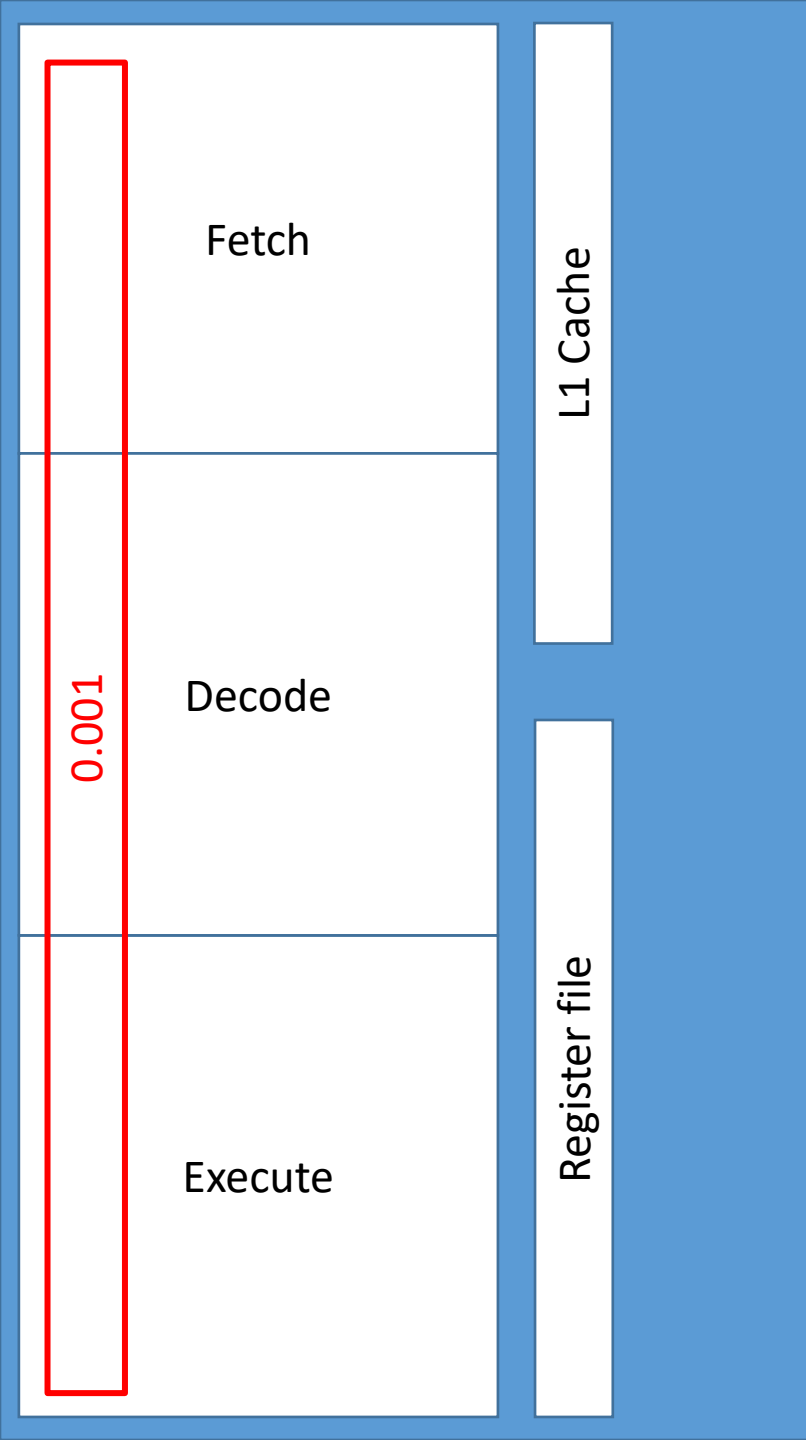
Decode

Execute

0.001

PIPELINING

Hur får vi upp frekvensen?



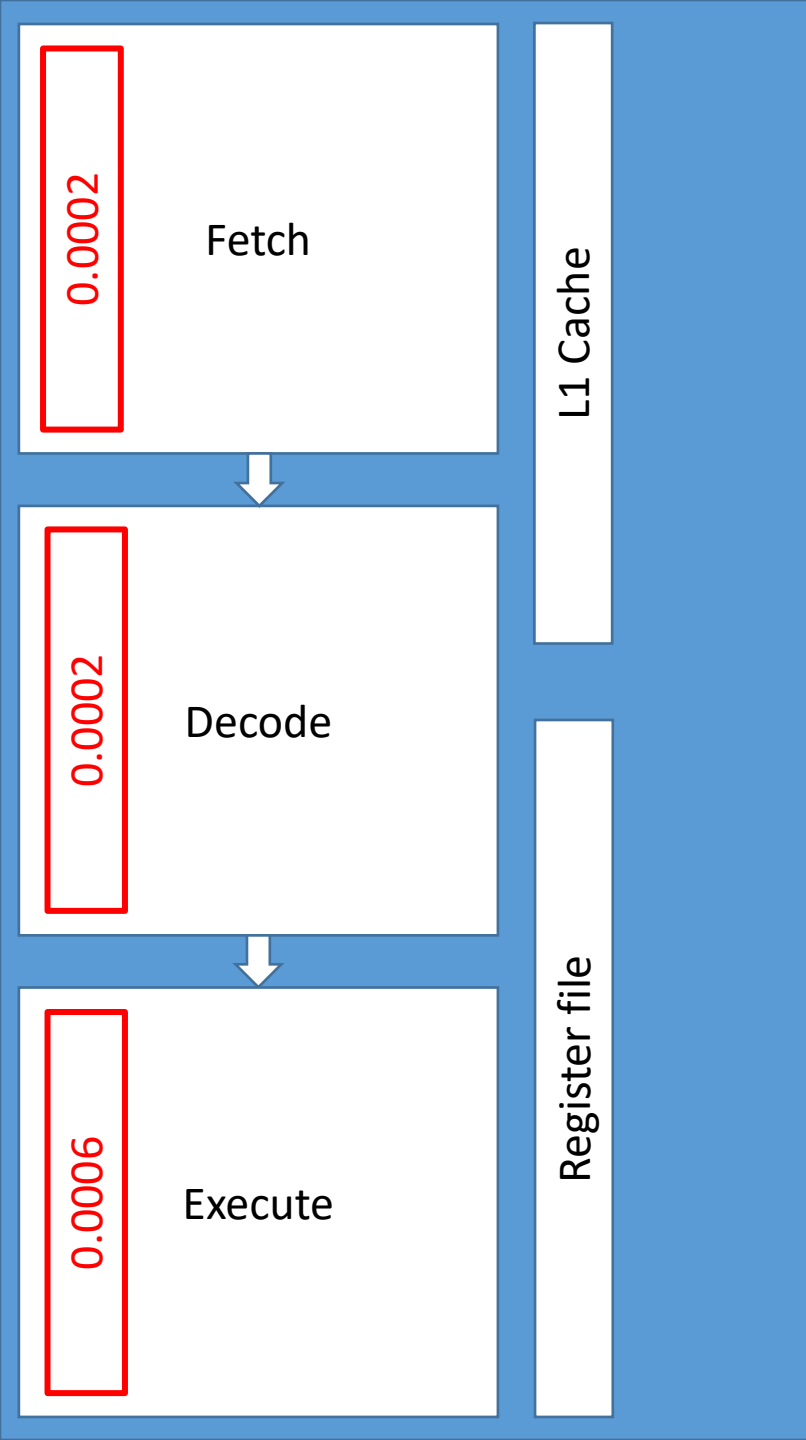
PIPELINING

Hur får vi upp frekvensen?

Vi delar in den i flera steg.

Den längsta vägen blir nu delad i tre delar, där den längsta subdelen blir ny längsta väg.

Ny frekvens: 1666.6 Hz



PIPELINING

Förutom en ny frekvens på 1666.6 Hz

Kan ha 4 instruktioner “in flight” samtidigt.

Inst. 0 skrivs

Inst. 1 exekveras

Inst. 2 avkodas

Inst. 3 hämtas

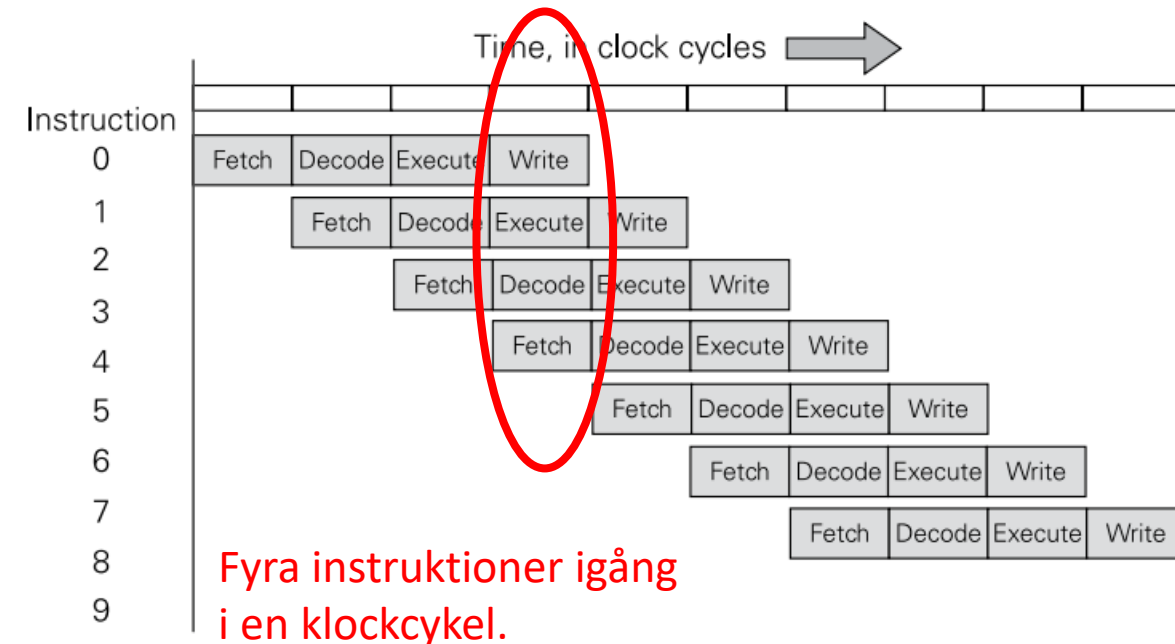
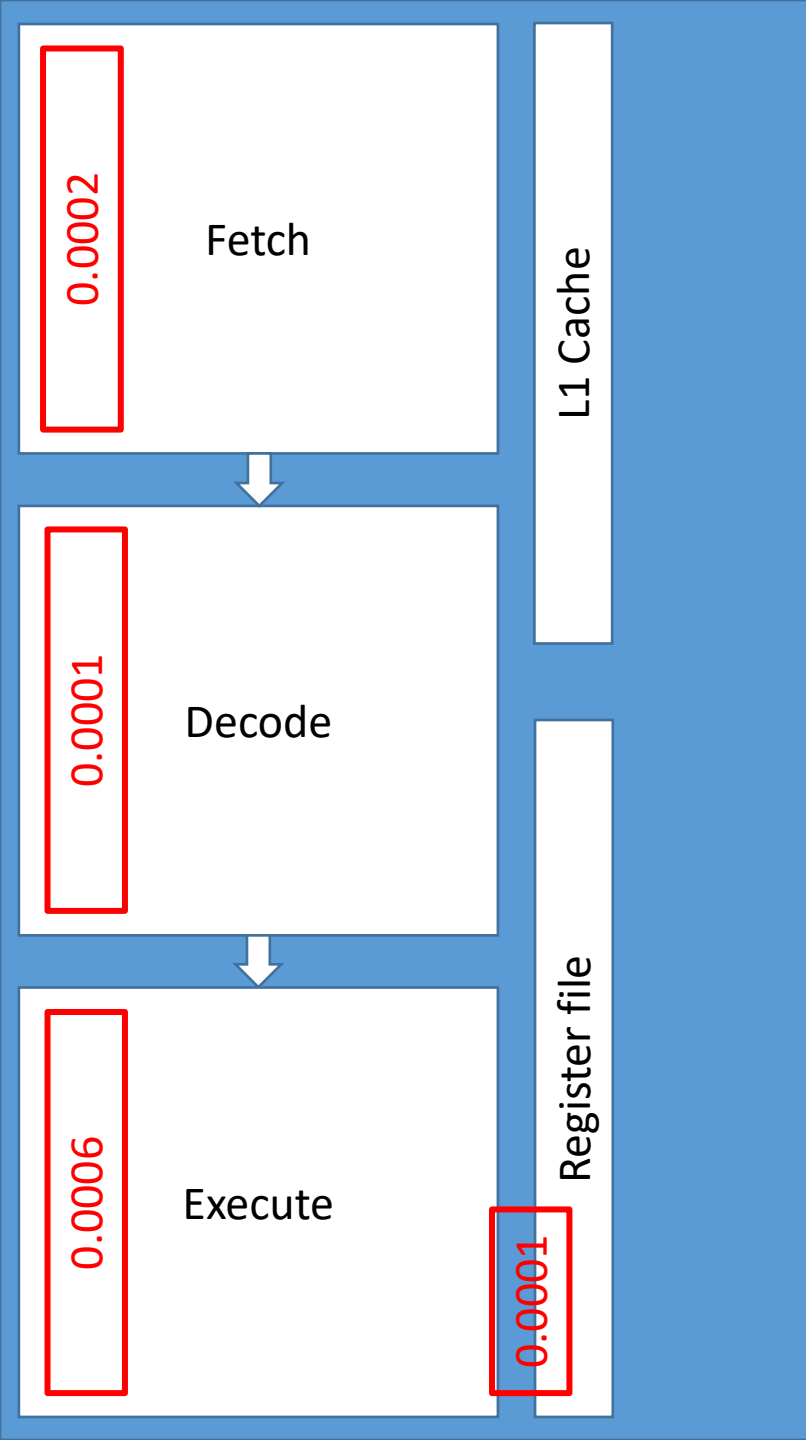


FIGURE 4-9: Overlapping instruction execution

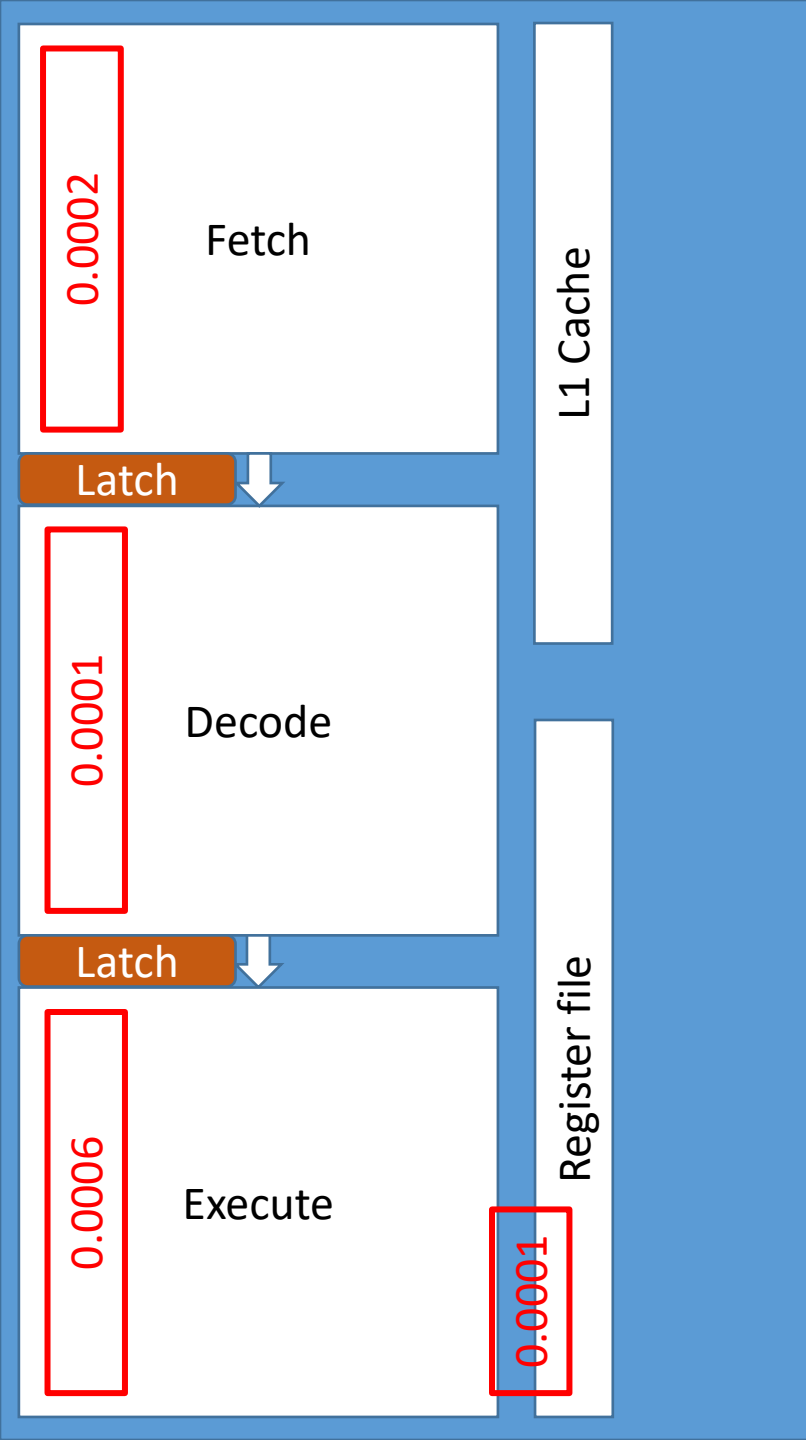


PIPELINING & LATCHES



För att skapa en pipeline behöver vi lägga till “minne” mellan de olika stegen.

Outputen från fetch i cykel 1 behöver sparas så den blir indata till decode i cykel 2, etc.....



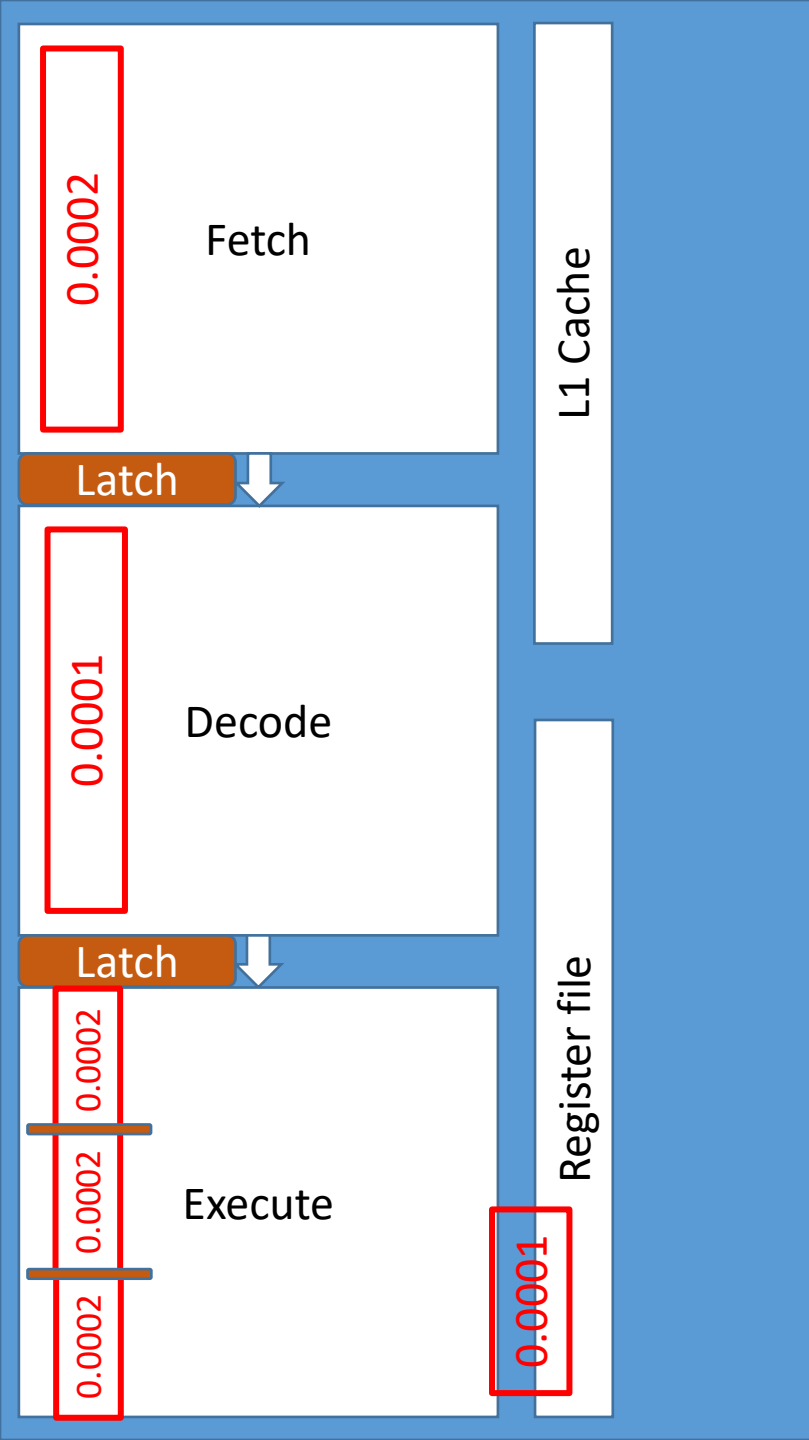
DJUPA PIPELINES

Nästa steg är att dela in de kvarvarande stegen i flera understeg.

Med 3 execute steg och allting annat lika kan klockfrekvensen bli: 5 000 Hz / 5 KHz

- Vilket är fem gånger snabbare än det första exemplet.

- ARM A53
- Intel Xeon E5-2650 v4
- Intel Pentium 4
- 8 stegs pipeline
- 16 stegs pipeline
- 31 stegs pipeline



ARM A53 PIPELINE

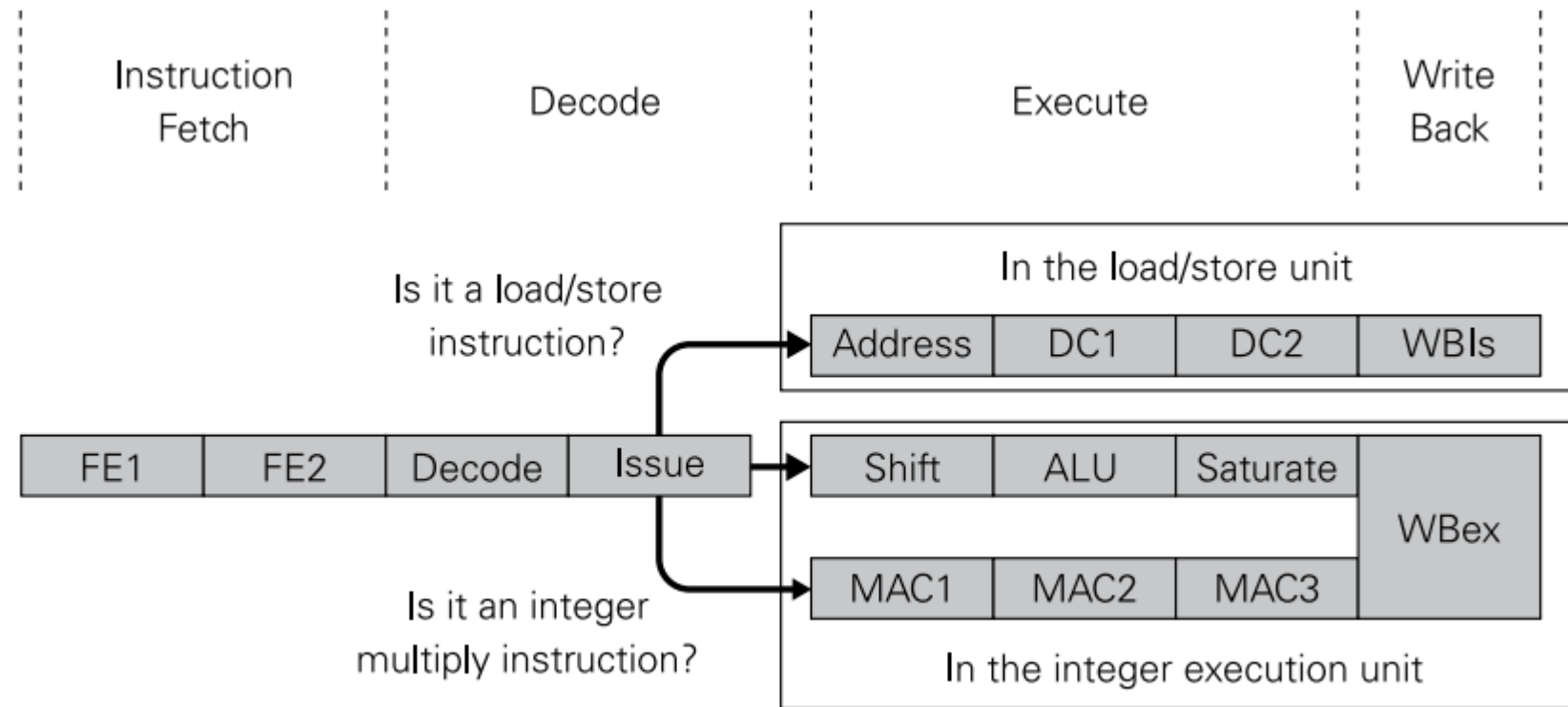
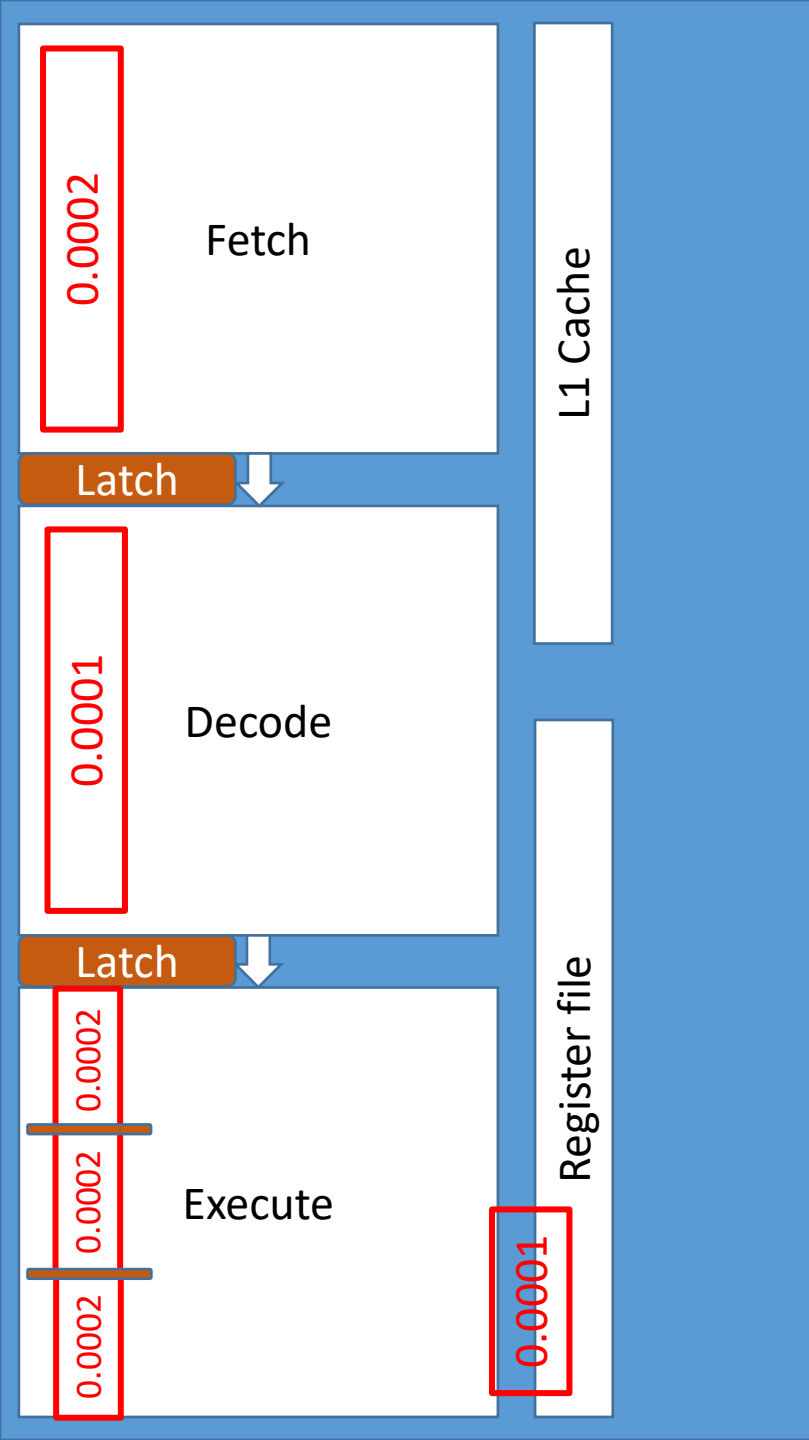
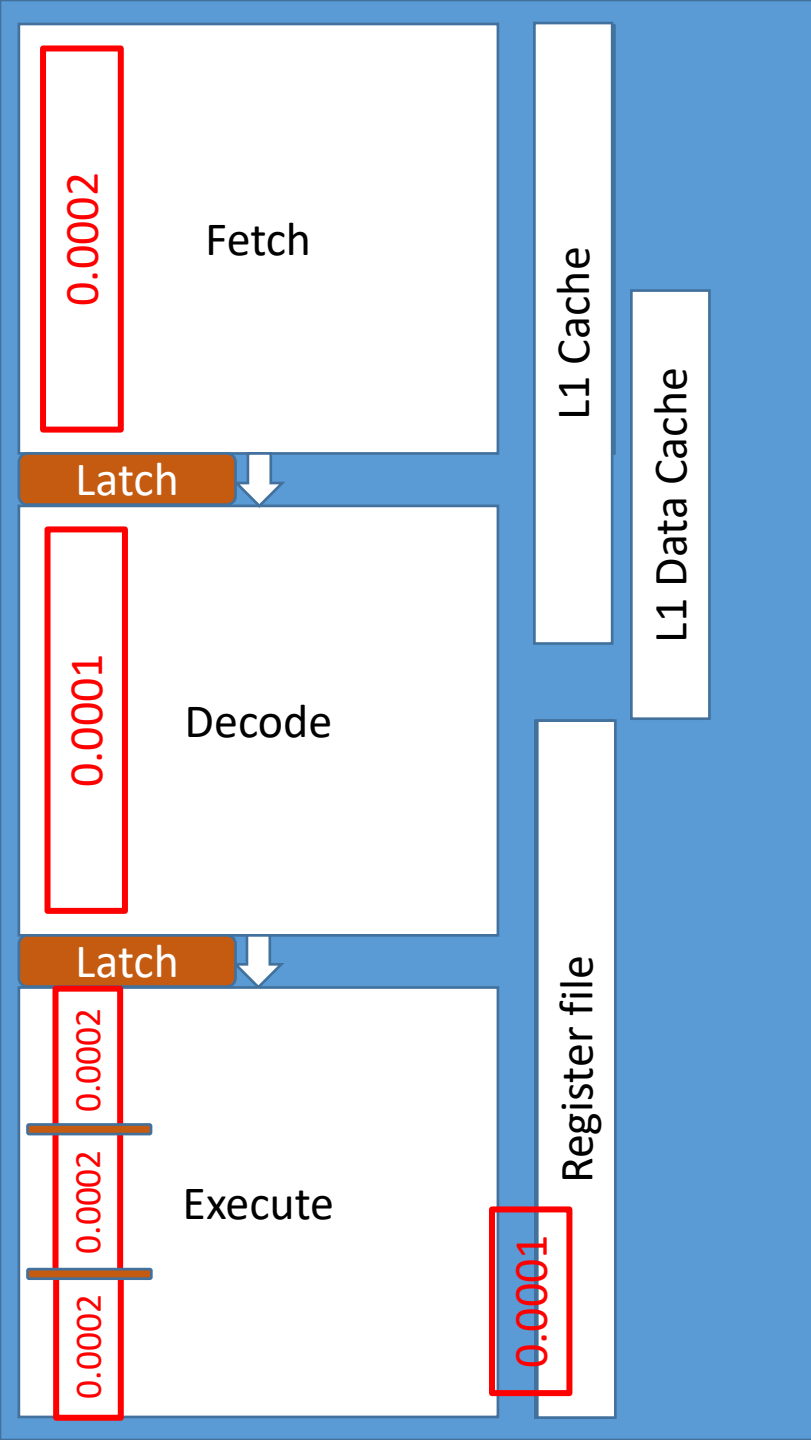


FIGURE 4-12: The ARM11 pipeline



PIPELINE HAZARDS

Control

Data

Structural

SUPERSCALAR PIPELINE



Vart och ett av stegen i pipelinen kan hantera flera instruktioner samtidigt.

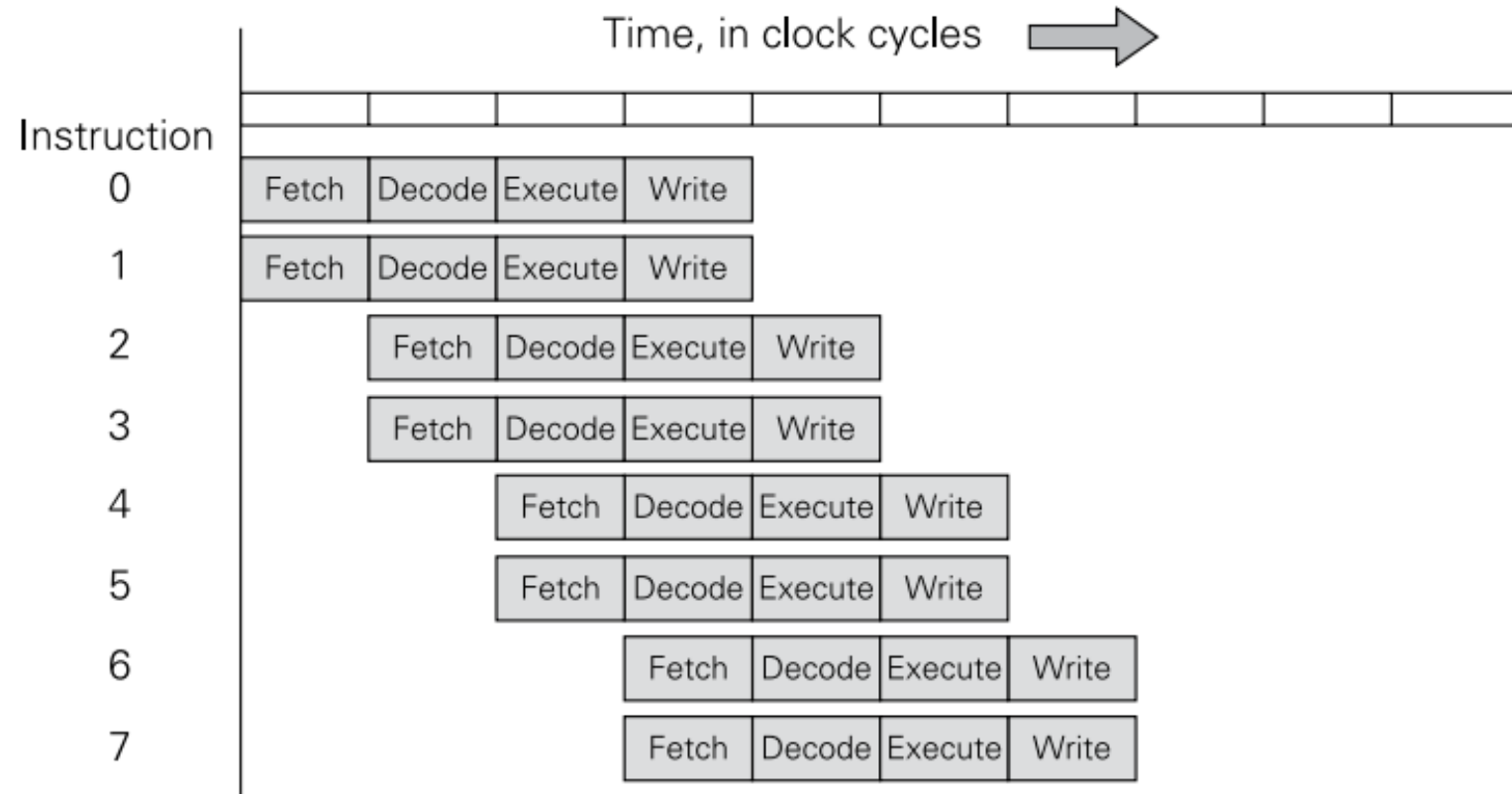
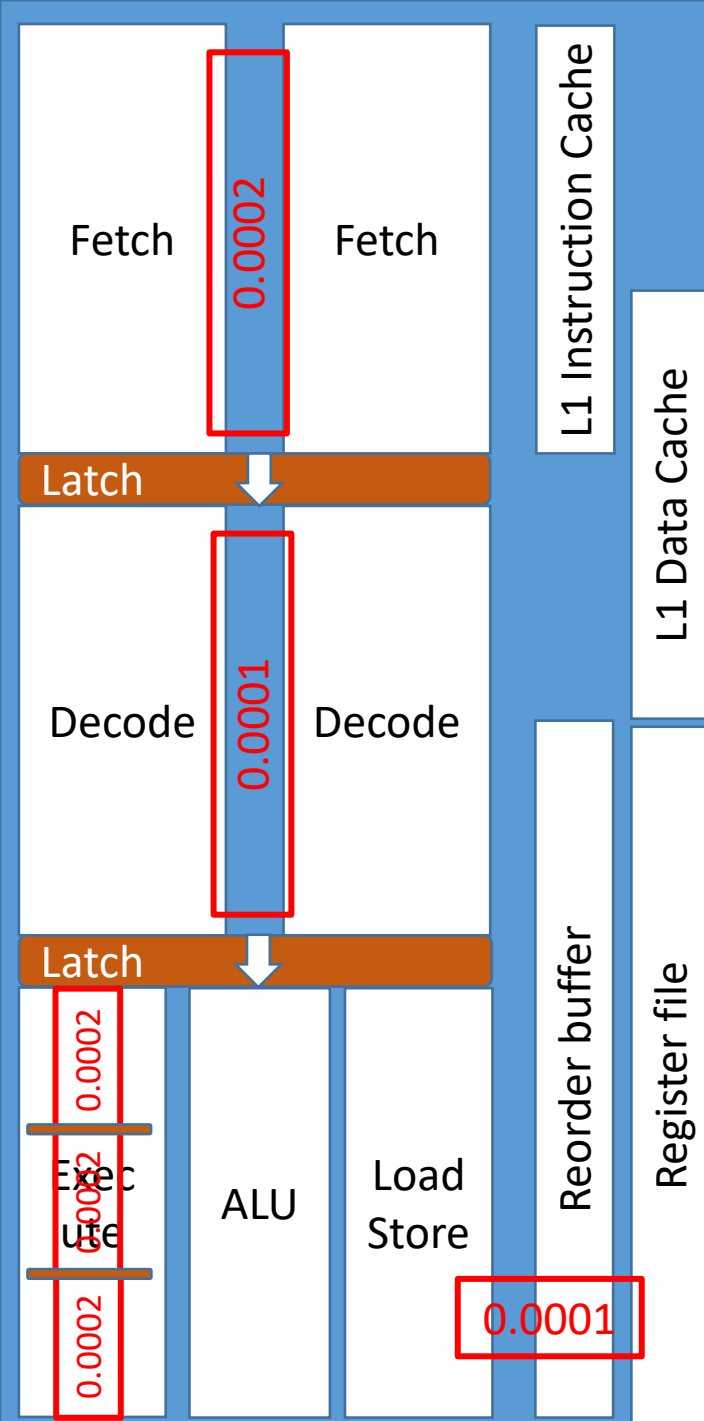


FIGURE 4-13: Superscalar execution

<https://www.bit-tech.net/reviews/tech/cpus/intel-core-i7-nehalem-architecture-dive/5/>

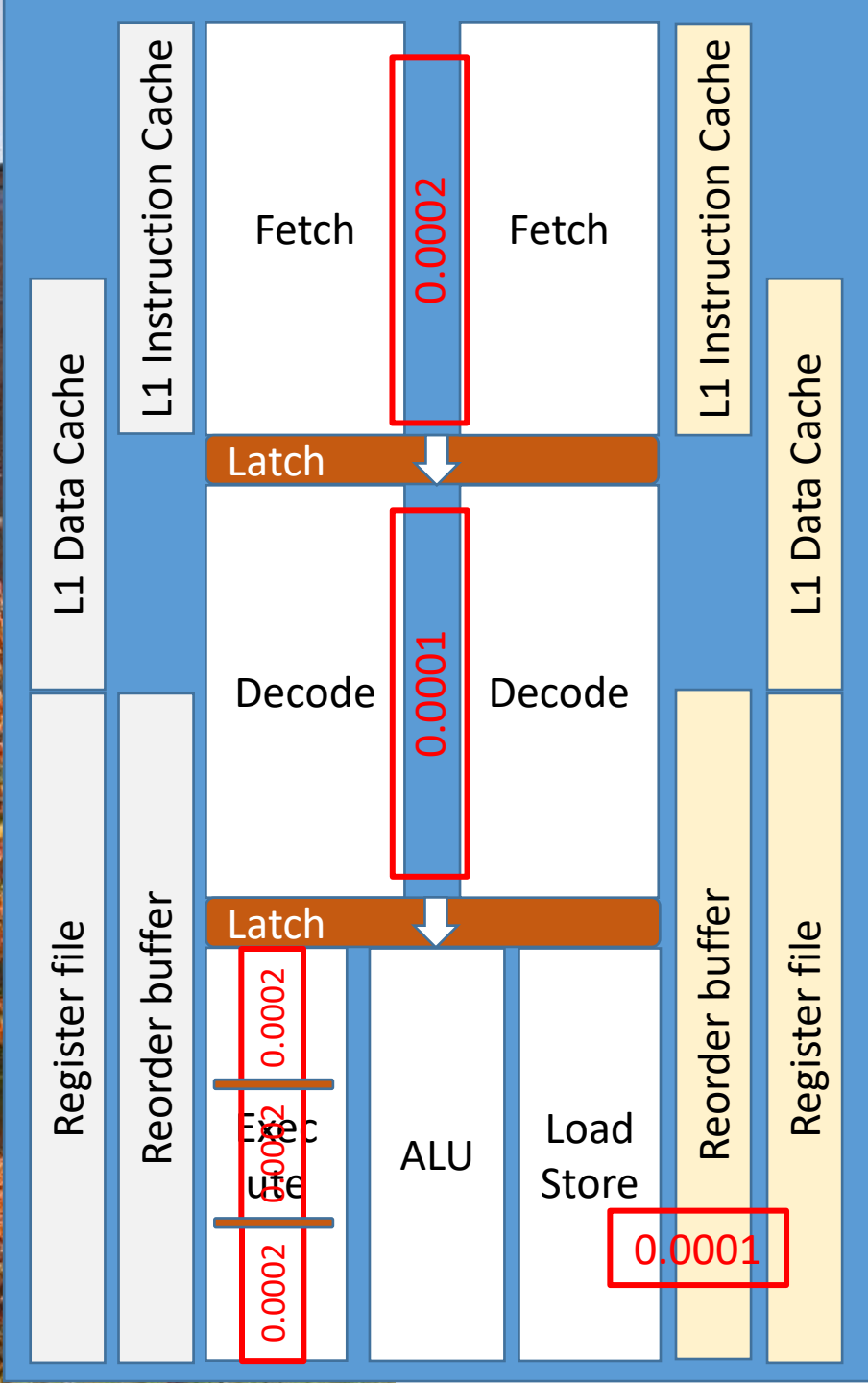
Simultaneous multithreading SMT



Hämtar och exekverar instruktioner från flera program samtidigt.

Exempel: Intel Hyperthreading, 2 way SMT

Fungerar bäst om program 1 och program 2 använder olika delar av processorn.



CISC VS. RISC

Complex Instruction Set Computing

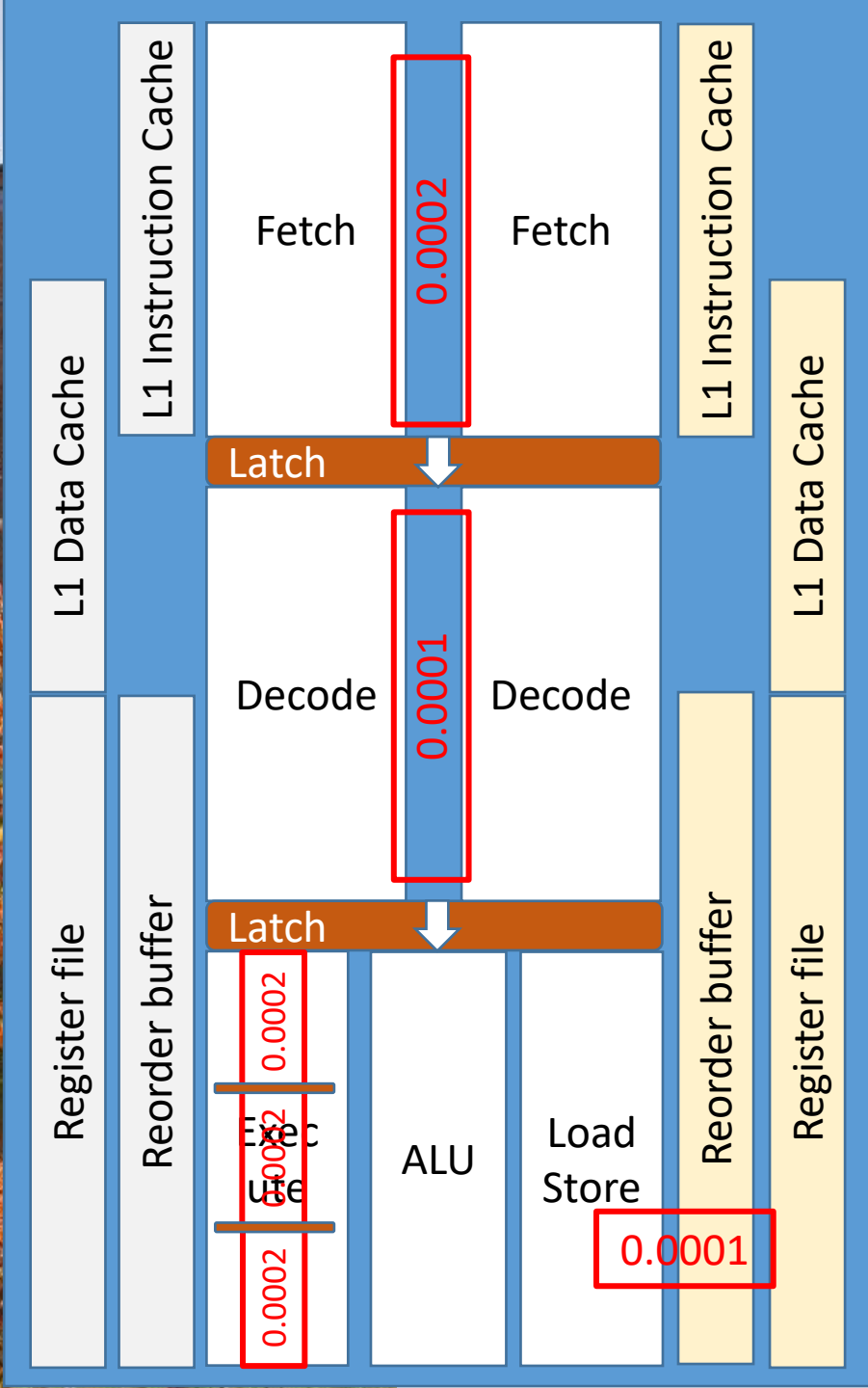
- Namnet är påhittat i efterhand
- Intel x86, x86_64
- Minne var dyrt, avancerade hårdvaruinstruktioner sparade in högnivåkod.
- 20% av instruktionerna stod för 80% av exekveringarna

Reduced Instruction Set Computing

- IBM POWER, SPARC, MIPS, ARM
- Få och enkla instruktioner
- Exekverar fler instruktioner för att göra samma sak.

Saker som kom från RISC

- Större registerfiler
- Load/store arkitektur
- Separata L1I och L1D
- Inga Legacy program



ENDIANNESS

- Det tidiga datorerna arbetade med 4 bitar och sedan 8 bitar.
 - De skrev 8 bitar (1 byte) till minnet åt gången.
 - Idag hanteras oftast 64-bitar (eller 32-bitar)
-
- Läser man in 4 byte – i vilken ordning skall de vara?
-
- Vanligast är little endian
-
- ARM är bi-endian

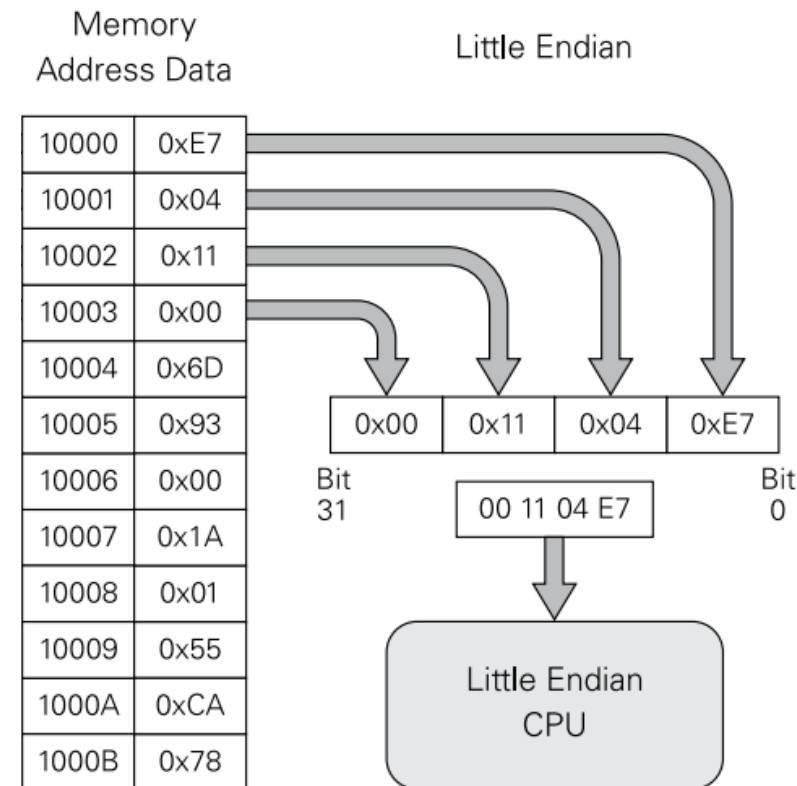


FIGURE 4-16: Big-endian vs. little-endian architectures