Livelli di astrazioni

Numerazione binaria | Algebra di Boole

Reti combinatorie e sequenziali

Unità firmware

P

Linguaggi ASM

Memoria

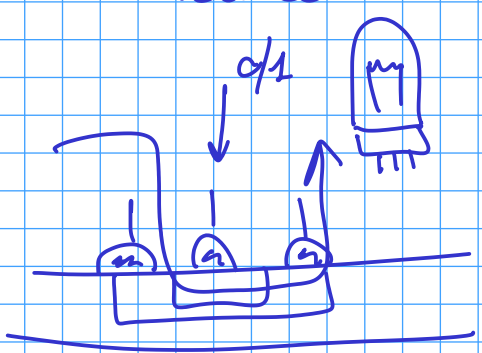
pipeline | superscalari | multicore

Esame { SCRITTO / ORALE } { prove intermedie } { progetto VERILOG }

Ricevimento Gio 15-18

Storia tecnologie

'50 '60 valvole



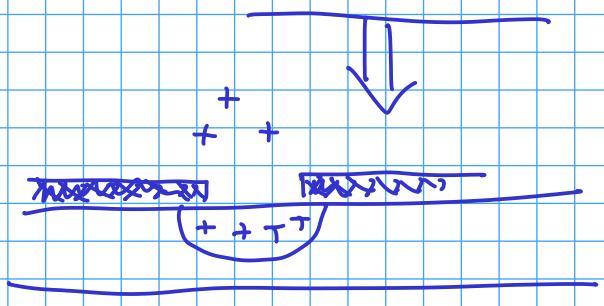
CEP

20 nm

14 nm

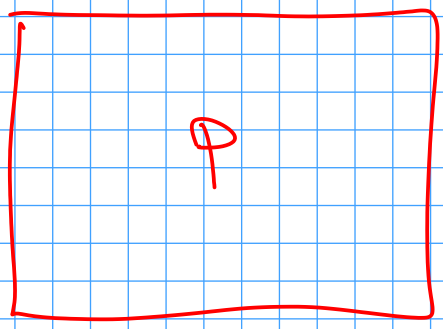
10 7 nm ?

transistor



Z80 16 bit 1 MHz

Legge di Moore



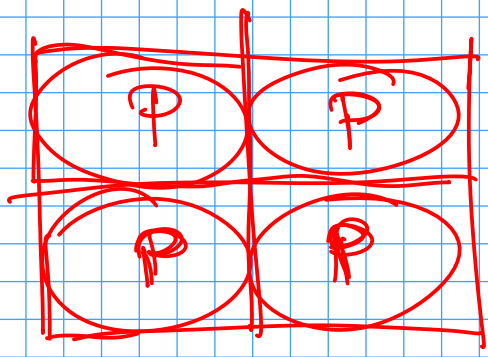
area nel chip X%

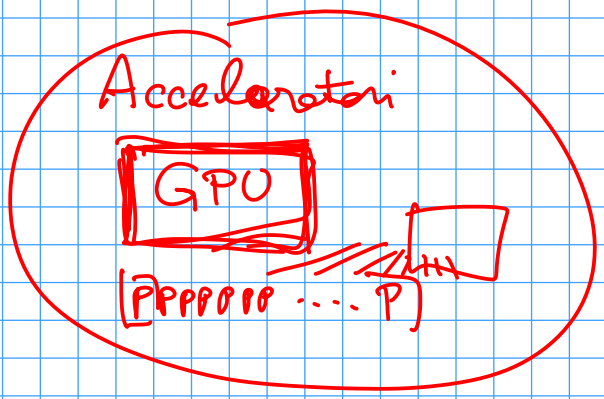
miglioramento prestazioni

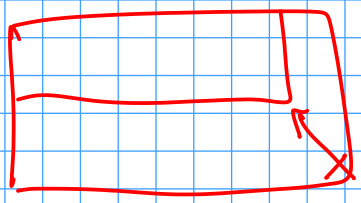
15%

Y%

Z%







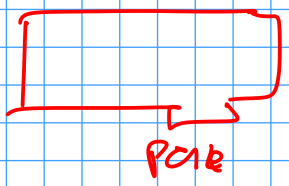
1/4

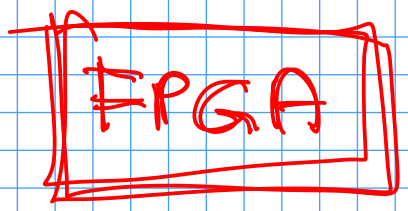
Intel XEON PHI

64 core x86

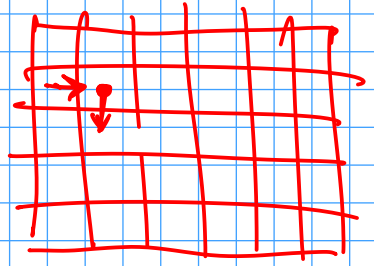
60

KNL





Field Programmable Gate Array



Processori

16 core x socket

2 thread

64

dati

48

GPU

o(1000) "core"

o(10) "SM"

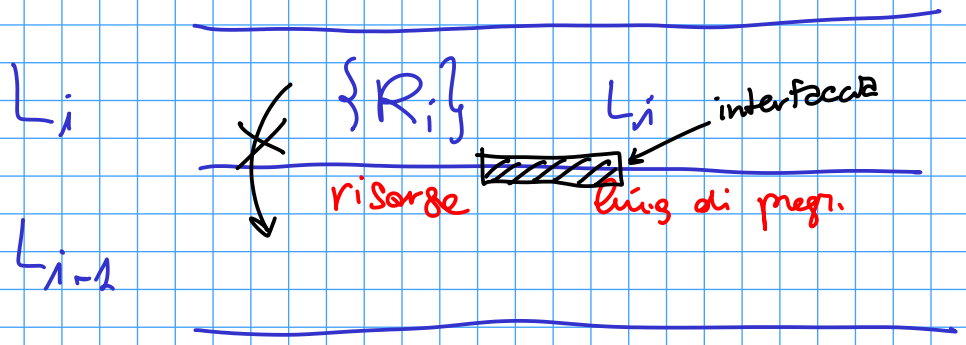
FPGA

1M 10M

celle riconfigurabili

20 - 14 nm ⇒ 10 nm

Livelli di astrazione

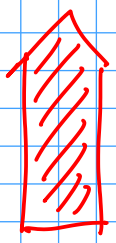


Macchina virtuale

MV_i

MV'_{i-1} } { MV''_{i-1}

astrazione (virtualizzazione)

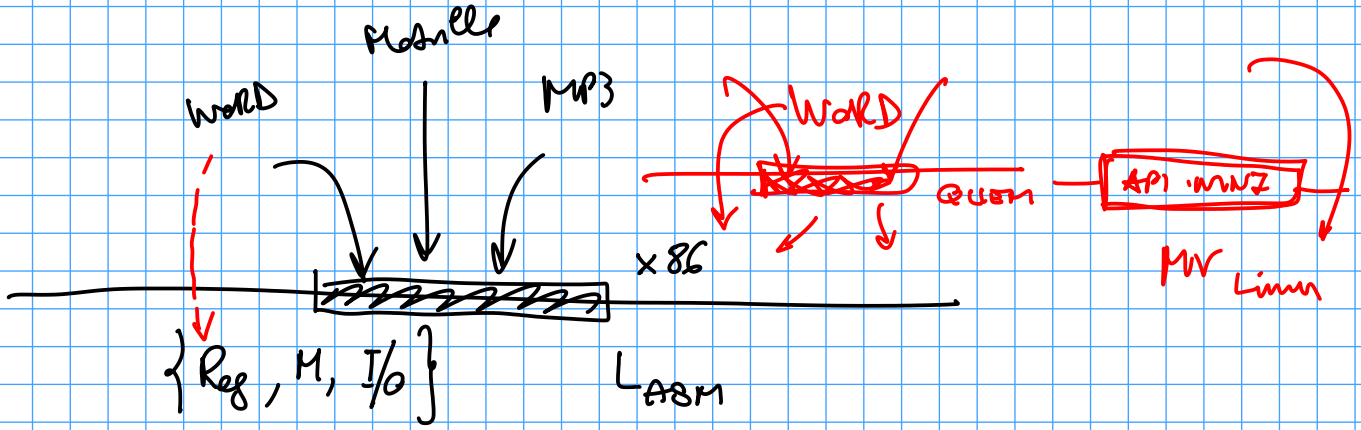


concretizzazione (emulazione)

Livello delle APPL. {Memoria, Librerie ...} L = { C/C++ Java Python

Livello Hardware {registri, componenti memoria, reti di calcolo}

L = "regole fisiche"



APPLICAZIONI — R = {"oggetti", ..} — L = {C, C++, Java, Occam, C#, F#, ..}

MACCHINA A PROCESSI — R = {condizioni di comunicazione, messaggi} — L = linguaggio concorrente

MACCHINA ~~ASSEMBLER~~ — R = {locazioni di memoria, registri, I/O ..} — L = linguaggio assembler

MACCHINA FIRMWARE — R = {registri, operatori, strutture di interconnessione} — L = μ-programmazione (un linguaggio)

MACCHINA HARDWARE — R = {componenti fisici} — L = ?

dsm

PROC formale — {operazioni esterne da implementare}

↓

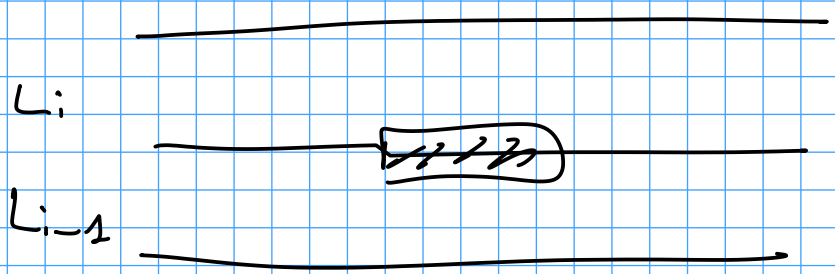
fw { μ-programma

↓

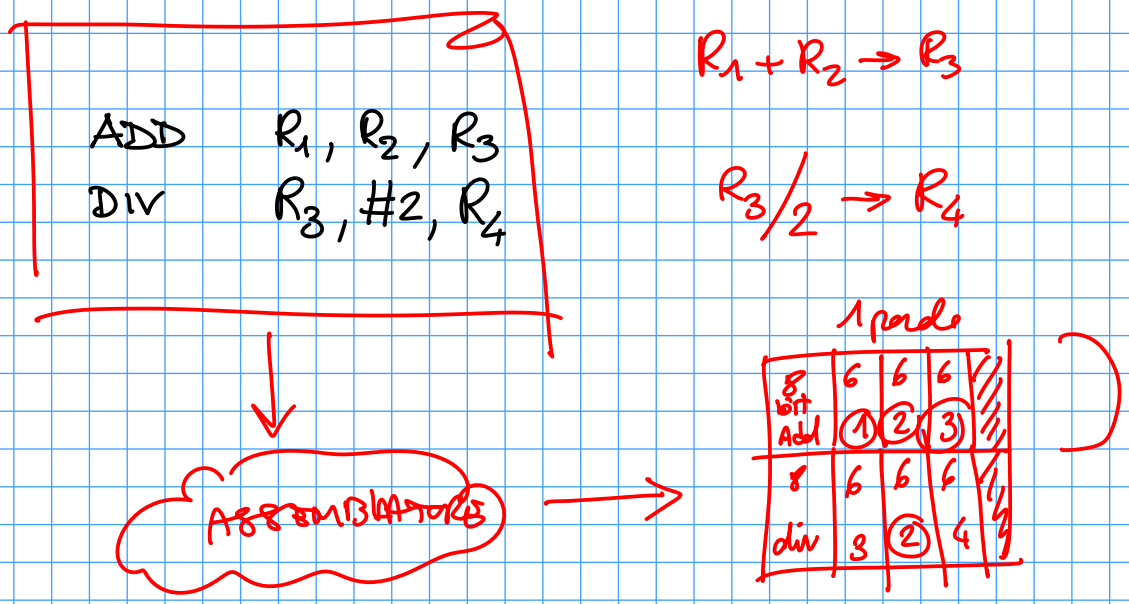
2 reti sequenziali

↓

componenti logici



} compilazione / interpretazione


ADD R1, R2, R3
DIV R3, #2, R4
R1 + R2 → R3
R3/2 → R4
1 parola
8 bit Add
6 6 6
1 2 3
8
6 6 6
div
3 2 4