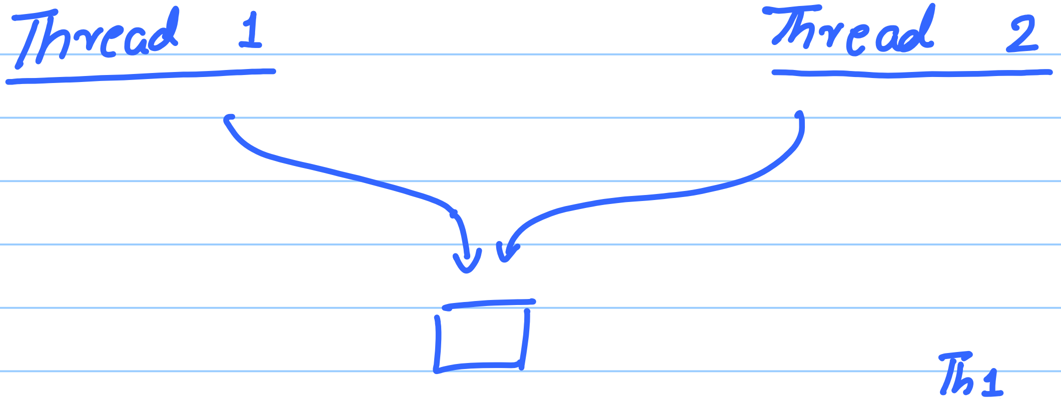
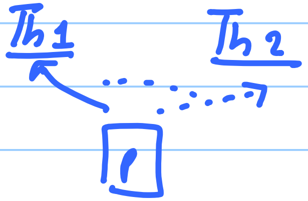
Nov. 14

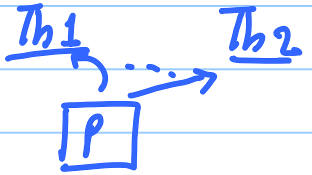


Fine Grained Multi-Threading →



Switches every alternate cycle.

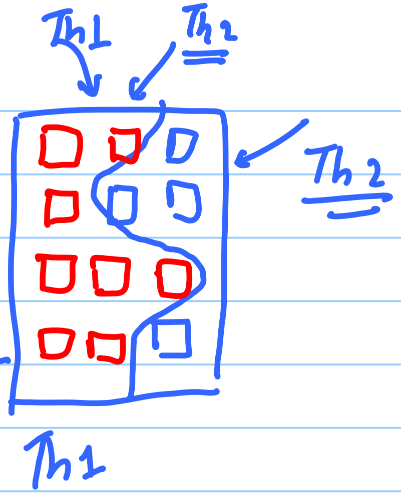
Coarse Grained Multi-Threading →



Switch every 'n' cycles.

Multiple Issue Processor

Simultaneous Multi-threading



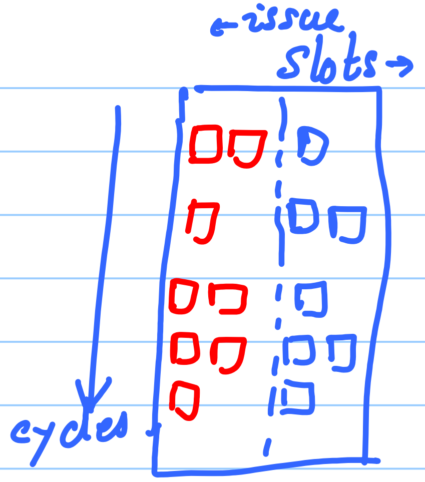
- Forward across pipelines
- Stalling & forwarding logic is complicated.

Hyper-Threaded.

PENTIUM CORE i7 (HT)

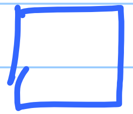
Assume:

4 insts. per cycle

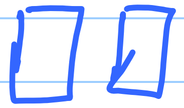


HT MODE IN INTEL PROCESSORS

NON HT MODE



HT MODE



Flynn's Classification

S → Single M → Multiple

I → Instruction D → Data.

SISD
|
Traditional sequential processor.

SIMD
- Graphics Proc.
- Vector Proc
- ⋮

MISD

Very Rare.

X

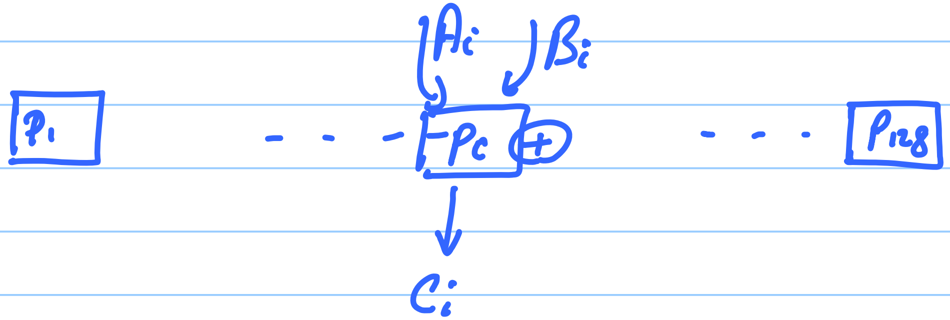
MIMD
↓
Multi-processor
or
Multi-Threaded

Vector Processor

$$C[1 \ldots 128] = A[1 \ldots 128] + B[1 \ldots 128]$$

Addition is being done parallely.

$$C_i = A_i + B_i$$

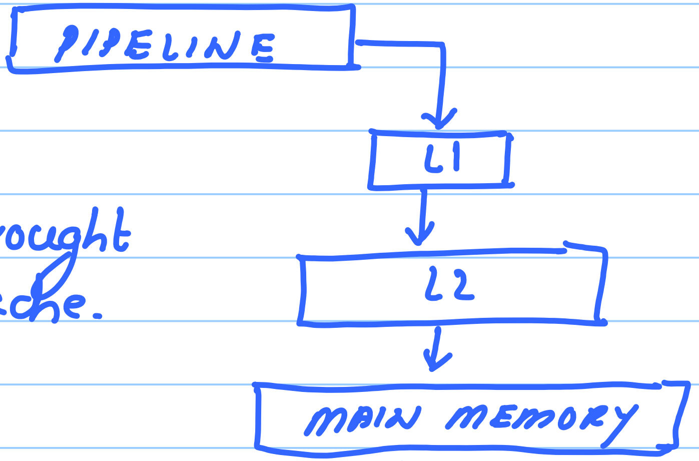


$I \to \oplus$

$.+ \to$ Vector Add

$$C = A .+ B$$

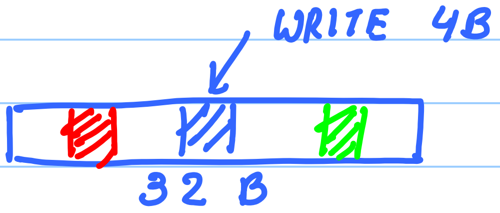
Cache Overview



Read → Data is brought to the L1 cache.

Write → (Read ++ or Read + something more)

Typical cache block: 32B



1) First, get the block to the L1 Cache

2) Do the write.

WB

Write-Back

1) Do not propagate any writes to the lower level in normal situations.

2) When the block is evicted
→ check if it has been written to
If yes
→ Propagate the change to the

WT

Write-Through

1) Every write is written to the lower level

2) Requires a lot of bandwidth

3) Evictions are cheap.

lower level

3) Saves bandwidth

4) Evictions are expensive