Formulas

1. M1

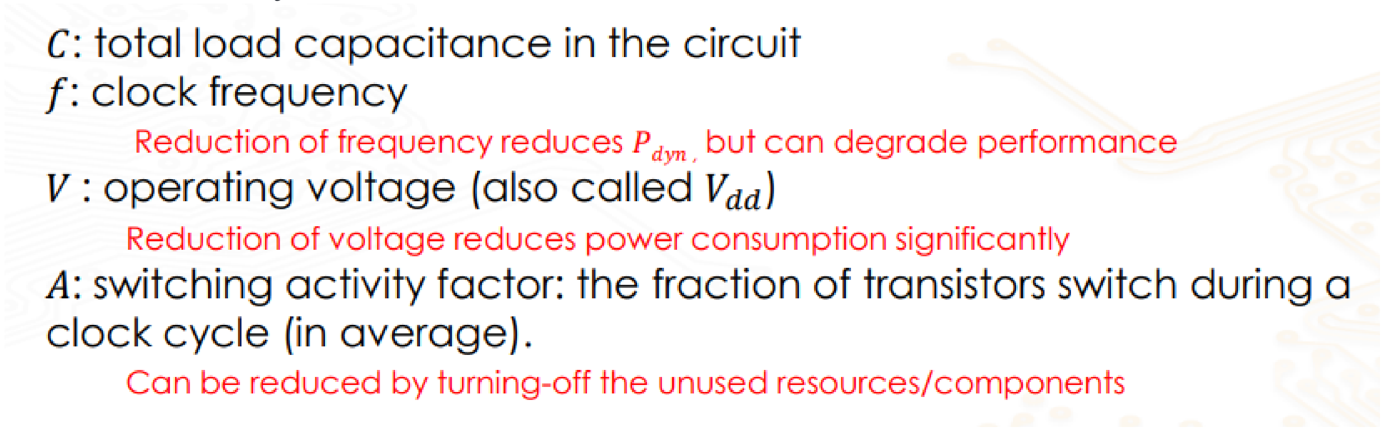
$$\text{Performance} = \frac{1}{\text{execution time}}$$

$$\text{Execution Time} = \text{IC x CPI x T}$$

$$\text{Speedup} = \frac{\text{Perf}_A}{\text{Perf}_B} = \frac{\text{Time}_B}{\text{Time}_A} = \frac{T_{\text{original}}}{T_{\text{enhanced}}} = \frac{1}{(1-E)+E/S}$$

- E is the fraction of the process that got speed up by a factor of S
- $\text{Speedup}_{\text{max}} = \frac{T_s+n\times T_p}{T_s+T_p}$ for Gustafson's Law

Dynamic Power $P_{dyn} = ACV^2f$



C: total load capacitance in the circuit
f: clock frequency
Reduction of frequency reduces P_{dyn}, but can degrade performance
V : operating voltage (also called V_{dd})
Reduction of voltage reduces power consumption significantly
A: switching activity factor: the fraction of transistors switch during a clock cycle (in average).
Can be reduced by turning-off the unused resources/components

Static / Leakage Power $P_{st} = VI_{leak}$

Maximum operating frequency $f_{max} \propto [V - V_{th}]^2/V$

2. M4

$$\text{Steady State CPI} = \frac{\text{number of instructions + number of stalls}}{\text{number of instructions}}$$

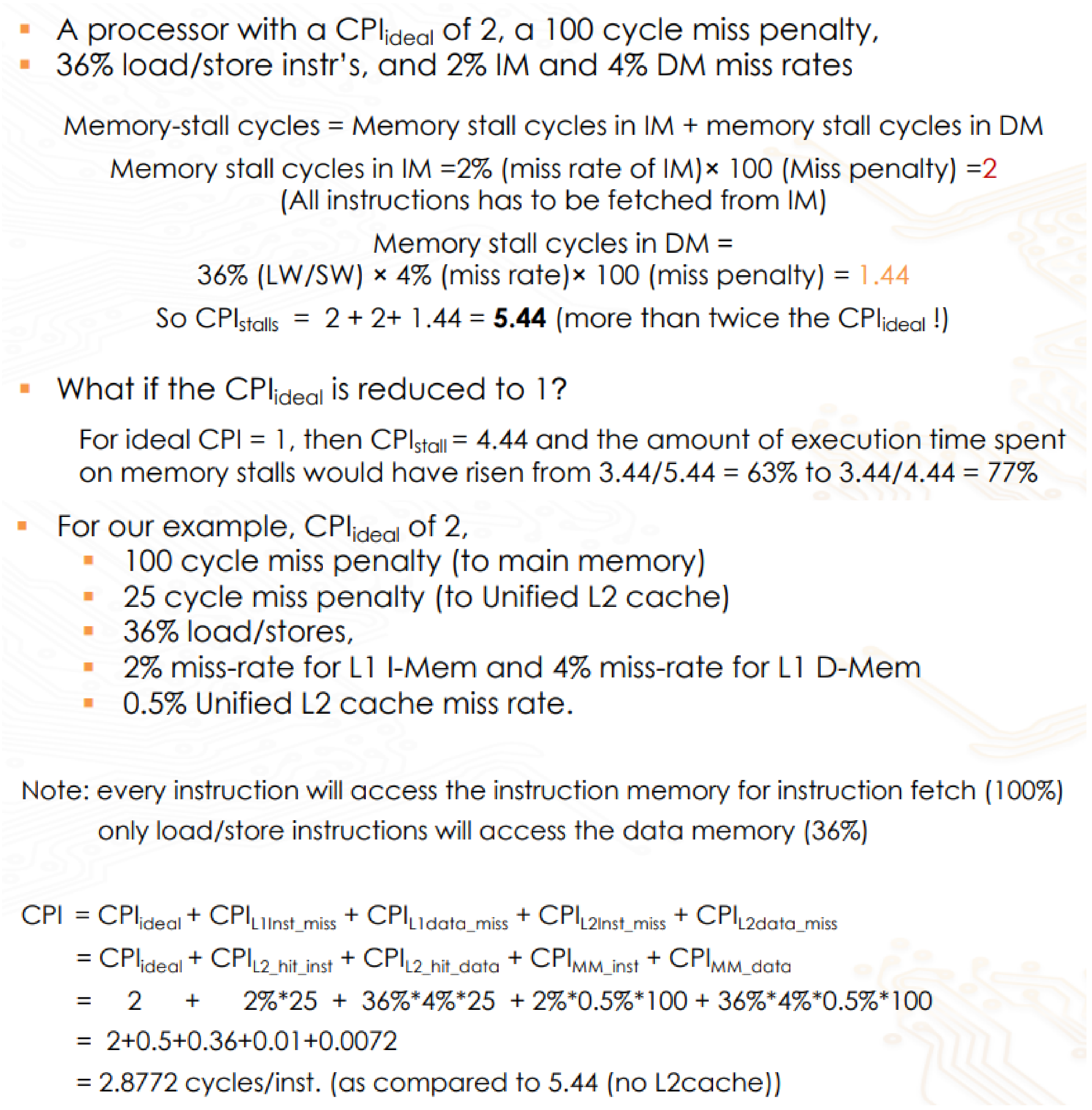
3. M5

- Size of main memory = 2 ^ number of bits for addressing
  - e.g. main memory size = 2GB, number of bits for addressing = 31

$$\text{cache size} = (1\text{ bit} + t\text{ bits} + \text{BS bytes}) \times E \times S$$

- Can also be Block size $\times$ bytes used for tag, LRU, dirty bits (number of bits / 8) $\times$ no. of blocks
- t = tag, BS = blocksize, E = cache lines in 1 set, S = number of sets
- Number of bits in `offset` is given by number of bytes in 1 block (if byte addressable)

- Number of bits in `set` is given by size of cache / size of set
- Number of bits in `tag` is the remaining bits
- **Other special stuff**
  - Number of bits in a cache line = "#tag + #valid + #data" (data bits = block size)
  - Cache Size = "#cachelines x #bits in cacheline"
  - Cache Overhead = Cache Size / Data size in cache (excludes tag and valid bits)
- Number of sets = number of blocks / set associativity (2 if 2-way set associative)
- Execution time $= \text{IC} \times \text{CPI} \times \text{Cycle time}$
  - If cache hit costs are included, $\text{CPI} = \text{CPI}_{\text{ideal}} + \text{Memory-stall cycles}$
  - Memory-stall cycles = memory accesses / program $\times$ miss rate $\times$ miss penalty



- A processor with a $\text{CPI}_{\text{ideal}}$ of 2, a 100 cycle miss penalty,
- 36% load/store instr's, and 2% IM and 4% DM miss rates

Memory-stall cycles = Memory stall cycles in IM + memory stall cycles in DM

Memory stall cycles in IM =2% (miss rate of IM)× 100 (Miss penalty) =2

(All instructions has to be fetched from IM)

Memory stall cycles in DM =

36% (LW/SW) × 4% (miss rate)× 100 (miss penalty) = 1.44

So $\text{CPI}_{\text{stalls}}$ = 2 + 2+ 1.44 = **5.44** (more than twice the $\text{CPI}_{\text{ideal}}$!)

- What if the $\text{CPI}_{\text{ideal}}$ is reduced to 1?

  For ideal CPI = 1, then $\text{CPI}_{\text{stall}}$ = 4.44 and the amount of execution time spent on memory stalls would have risen from 3.44/5.44 = 63% to 3.44/4.44 = 77%

- For our example, $\text{CPI}_{\text{ideal}}$ of 2,
  - 100 cycle miss penalty (to main memory)
  - 25 cycle miss penalty (to Unified L2 cache)
  - 36% load/stores,
  - 2% miss-rate for L1 I-Mem and 4% miss-rate for L1 D-Mem
  - 0.5% Unified L2 cache miss rate.

Note: every instruction will access the instruction memory for instruction fetch (100%)

only load/store instructions will access the data memory (36%)

$$\begin{aligned}
\text{CPI} &= \text{CPI}_{\text{ideal}} + \text{CPI}_{\text{L1Inst\_miss}} + \text{CPI}_{\text{L1data\_miss}} + \text{CPI}_{\text{L2Inst\_miss}} + \text{CPI}_{\text{L2data\_miss}} \\
&= \text{CPI}_{\text{ideal}} + \text{CPI}_{\text{L2\_hit\_inst}} + \text{CPI}_{\text{L2\_hit\_data}} + \text{CPI}_{\text{MM\_inst}} + \text{CPI}_{\text{MM\_data}} \\
&= 2 + 2\%*25 + 36\%*4\%*25 + 2\%*0.5\%*100 + 36\%*4\%*0.5\%*100 \\
&= 2+0.5+0.36+0.01+0.0072 \\
&= 2.8772 \text{ cycles/inst. (as compared to 5.44 (no L2cache))}
\end{aligned}$$

Average Memory Access Time (AMAT)

- Average time required to access memory considering both hits and misses
    - Time for hit $+$ Miss Rate $\times$ Miss Penalty