Performance

Runtime Equation

Run Time = Instructions × Cycles per Instruction × Clock Cycle Time

$$= I \times CPI \times CCT$$

Amdahl's Law

$$\text{Speedup} = \frac{1}{1 - F + \frac{F}{E}}$$

F is the fraction of *execution time* a portion of the program runs.
E is the enhancement applied to the portion of the program F.

Powers of 2

$$2^0 = 1$$
$$2^1 = 2$$
$$2^2 = 4$$
$$2^3 = 8$$
$$2^4 = 16$$
$$2^5 = 32$$
$$2^6 = 64$$
$$2^7 = 128$$
$$2^8 = 256$$
$$2^9 = 512$$
$$2^{10} = 1024 = 1kbyte$$
$$2^{20} = 1Mbyte$$
$$2^{30} = 1Gbyte$$

Hex to Binary Conversion

$$0x0 = 0000$$
$$0x1 = 0001$$
$$0x2 = 0010$$
$$0x3 = 0011$$
$$0x4 = 0100$$
$$0x5 = 0101$$
$$0x6 = 0110$$
$$0x7 = 0111$$
$$0x8 = 1000$$
$$0x9 = 1001$$
$$0xa = 1010$$
$$0xb = 1011$$
$$0xc = 1100$$
$$0xd = 1101$$
$$0xe = 1110$$
$$0xf = 1111$$

Cache

Average Memory Access Time

$AMAT$ = Hit Time + Miss Rate × Miss Penalty

Fast Access L1 Cache

Max size of L1 Cache = Page Size × Associativity

Register Usage Convention

Register	Role in the procedure call standard
r15 (pc)	The Program Counter
r14 (lr)	The Link Register
r13 (sp)	The Stack Pointer
r12 (ip)	The Intra-Procedure-call, caller-saved
r11	callee-saved
r10	callee-saved
r9	callee-saved
r8	callee-saved
r7	callee-saved
r6	callee-saved
r5	callee-saved
r4	callee-saved
r3	Argument register 4, caller-saved
r2	Argument register 3, caller-saved
r1	Argument register 2, caller-saved
r0	Argument register 1, return value, caller-saved

Example Cycle Timings

Instruction Sequence	Cycles
`add r1, r2, r3` `add r4, r1, r5`	2
`add r1, r2, r3` `add r4, r5, r1 asl #2`	3 r1 is early reg
`add r1, r2, r3` `str r1, [r4]`	2
`add r1, r2, r3` `str r4, [r1]`	3 r1 is early reg
`ldr r1, [r2]` `add r4, r1, r5`	4
`ldr r1, [r2]` `ldr r4, [r1]`	5 r1 is early reg