Terminology

Tera	10^{12}
Giga	10^{9}
Mega	10^{6}
Kilo	10^{3}
1 GHz	$1x10^{9}$ Hz
1 TB	$1x10^{12}$ Byte
1 Byte	8 Bits

Conversions

Binary	Hex	Dec
1111	0F	15

Hex

In Hex, each digit represents 4 binary bits (i.e. 1111 1111 is FF)

2's Complement

0111	7
1000	-8
1111	-1

Logic Gates

AND	A B
OR	A+B
NOT	A
NAND	A B
NOR	A+B
XOR	A B

XOR Truth Table

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Basic Identities

Name (Law)	AND form	OR form
Identity	1A = A	0+A = A
Null	0A = 0	1+A = 1
Idempotent	AA = A	A+A = A
Inverse	AA = 0	A+A = 1
Commutative	AB = BA	A+B = B+A
Associative	(AB)C = A(BC)	(A+B)+C = A+(B+C)
Distributive	A+BC = (A+B)(A+C)	A(B+C) = AB+AC
Absorption	A(A+B) = A	A+AB = A
DeMorgan's	AB = A+B	A+B = AB

Boolean Concepts

Addition	Subtraction
1+1 = 10	0-1 = 1 (borrow method)

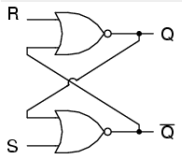
All other operations straightforward.

Boolean Concepts II

Carry out: Carry out of leftmost bit

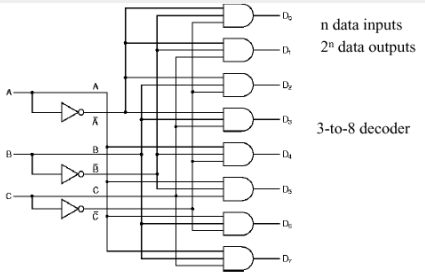
Overflow: Result too large or small to fit into bits

S-R Latch



S	R	Q	$\overline{Q}$
0	0	latch	latch
0	1	0	1
1	0	1	0
1	1	0	0

Decoder



Other Sequential Circuits

Multiplexer

Demultiplexer

Decoder

Encoder

Priority Encoder

Byte Ordering

Bytes in a word can be numbered from left to right or right to left. Big-endian: Most significant byte (byte containing most significant bit) is stored first and following bytes are stored in decreasing significance order. Little-endian is the opposite.

Memory Hierarchy

In order of increasing access time, storage capacity, and bits you get per dollar spent: Registers; Cache; Main Memory; Magnetic or solid state disk; Tape or Optical Disk.

Cache

Component that stores data to speed up future search requests. Cache hit - data can be found vs Cache miss - data cannot be found. Hit rate - % of accesses resulting in cache hits. Relatively small due to cost. Locality of reference - Temporal locality: reuse of specific data, and/or resources, within a relatively small time duration. Spatial locality: use of data elements within relatively close storage locations. Tradeoff: Larger caches have better hit rates but longer latency. Small fast caches backed up by larger, slower caches. Multi-level caches: check fastest L1 cache first; if it hits, proceeds at HS. If L1 misses, L2 is checked, and so on, before accessing external memory..

Cache Mapping

Cache type	Hit Ratio	Search speed
Direct Mapped	Good	Best
N-Way Set Associative	Very good, better as N increases	Good, worse as N increases
Fully associative	Best	Moderate

Replacement Algorithms

Optimize management of cache - chooses which items to discard in a full cache

Each is a tradeoff between latency and hit ratio

Common: LRU, MRU, RR, 2-way set, Direct-mapped

Write Policy

Write-through	Write-back
Write is done synchronously both to cache and main mem	Write initially only to cache; write to main memory postponed until cache blocks containing data are about to be modified/replaced
ADV: Simpler and more reliable, mem is always up-to-date	ADV: Faster, uses less memory bandwidth

Write Policy (cont)

DISADV: Write is slower, every write requires main memory access, uses more memory bandwidth	DISADV: More complex, must track "dirty" locations

Mean access time

mean access time = c + (1-h)m
c - cache access time
h - hit ratio
m - main memory access time

Data Dependencies

True data dependency (RAW)	Occurs when an instruction depends on result of previous instruction
WAR	Occurs when an instruction requires a value that is later updated
WAW	Occurs when the ordering of instructions will affect the final output value of a variable

CISC vs. RISC

CISC	RISC
Emphasis on hardware	Emphasis on software
Includes multi-clock complex instructions	Single-clock, reduced instruction only
Memory-to-memory: "LOAD" and "STORE" incorporated in instructions	Register-to-register: "LOAD" and "STORE" are independent instructions
Small code sizes, high cycles per second	Low cycles per second, large code sizes
MULT	LOAD LOAD PROD STORE

Performance equation

time/program = time/cycle x cycles/instruction x instructions/program

CISC minimzes instructions per program at cost of cycles per instruction

RISC reduces cycles per instruction at cost of number of instructions per program

State Machines

Moore Machines: output is a function of the state

Mealy Machines: output is a function of the state and the input

Opcode

Portion of a machine language that specifies the operation to be performed

Addressing modes

Immediate

Direct

Indirect

Reg Direct

Reg Indirect

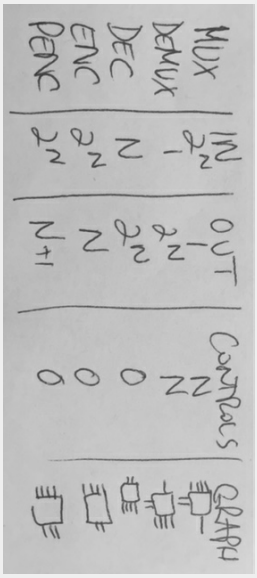
Reg Base-Ind

Reg Base-Scaled Ind

Basic ISA Classes

Accumulator	Stack	Gen Purp Reg	Load/Store
1 or 1+x address	0 address	2 or 3 address	3 address

Circuits



Speedup Calculation

Speedup = $(NT^1)/[K+(N-1)]T$

Where N is # of instructions, T^1 is time for stage 1 CPU to complete instruction

K is # of stages for multistage and T is time for step + latch