Lecture 17 Digital Electronics

Big shift: digital vs. analog

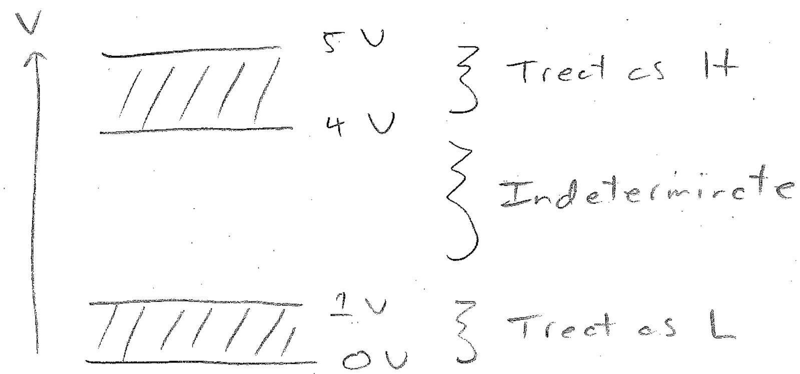
Idea of digital: reduce signals to two discrete states

- on/off
- high/low.
- true/false
- 1/0

Typically use +5V/0V

Main advantage: noise immunity

Typical setup:



Any voltage in 4-5 V range treated as H, small fluctuations don't matter

So we can manipulate & store information with little concern for errors

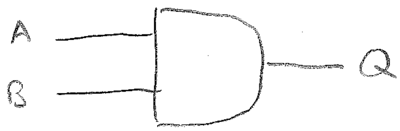
Each H/L signal called a <u>bit</u>

8 bits = 1 byte

Manipulate digital signals using <u>gates</u>

Typically represent standard logic operations

AND gate



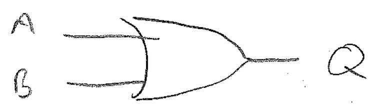
Truth Table:

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

Q is "true" if and only if A & B both true

Write $Q = A \text{ AND } B = A \& B = A \cdot B = AB$

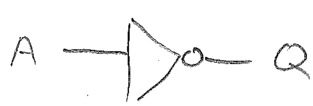
OR gate:



A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

$Q = A | B = A + B$

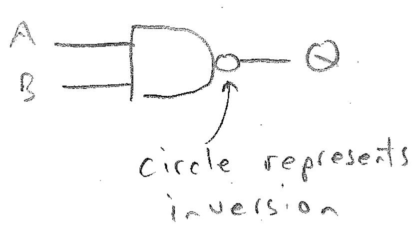
NOT gate:



A	Q
0	1
1	0

$Q = \sim A = A^* = \overline{A}$

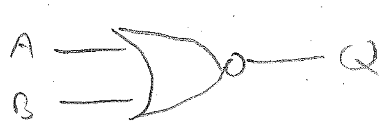
NAND gate:



A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

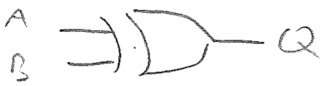
$Q = \overline{AB}$

NOR gate:



A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

XOR (Exclusive Or):



A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

$Q = A \oplus B$

$= \bar{A}B + A\bar{B}$

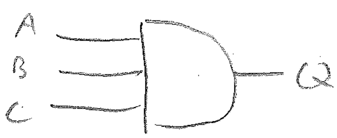
Buffer

A	Q
A	Q

$Q = A$

Also multiple input versions of most gates

3AND:



$Q = 1$ only if $A = B = C = 1$

$Q = ABC$

etc.

Note that theorems from mathematical logic apply here:

De Morgan's Laws:

$$\overline{A+B} = \bar{A} \cdot \bar{B}$$

$$\overline{AB} = \bar{A} + \bar{B}$$

Implies

Typically many ways to implement a given gate

Fewer components is better

Can reduce any operation to basic gates

addition, multiplication, etc.

In fact can reduce all operations to NAND gates

but not very efficient.

Implementation

Generally use transistors, either bipolar or FET

See idea in lab

Chip naming scheme: 74 AB XYZ

- 74 → logic gate
- AB → family
- XYZ → ID for gate type

Gate type = numbers

00	=	NAND	(quad = 4 gates/chip)
02	=	NOR	(quad)
04	=	NOT	(hex = 6 gates/chip)
08	=	AND	(quad)
32	=	OR	(quad)
86	=	XOR	(quad)

Many more

Family = different implementations

No code	=	plain bipolar — TTL "transistor-transistor logic"
LS	=	improved TTL
ALS	=	even better TTL
HC	=	plain FETs: CMOS "complementary MOSFETs"
HCT	=	CMOS-TTL interfacing
AC	=	improved CMOS

etc

So

74LS08 = bipolar and gates

74HC32 = FET or gates

Also another family of CMOS gates

CD4 XXX

MC14 XXX

Mostly more special purpose circuits.

Some notable differences between CMOS & TTL:

- TTL inputs float high when unconnected

  TTL outputs only source a little current (few mA) in H state

  Can sink a lot (few 10's of mA) in L state

- CMOS more symmetric:

  Floating inputs are indefinite

  H & L outputs both source/sink 10s of mA

In general CMOS usually better, but more expensive & more fragile

Explore more in lab