PURDUE UNIVERSITY DEPARTMENT OF PHYSICS

Physics 53600

Electronics Techniques for Research

Now in PowerPoint!

Spring 2020 Semester

Prof. Matthew Jones

Digital Electronics

- So far, we have analyzed transistors that are biased so that they are in the *active region*
- They act as voltage or current controlled current sources

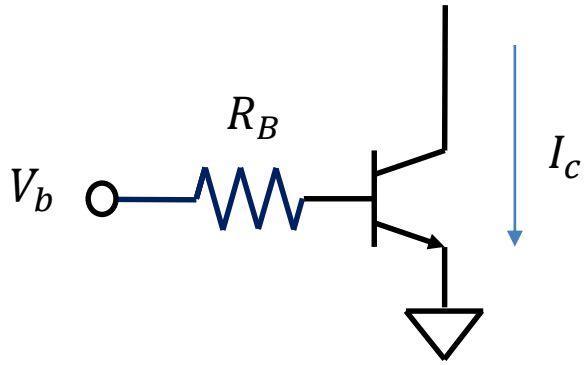
$$I_D = I_{DSS}\left(1 + \frac{V_G}{|V_P|}\right)^2$$

$$I_C = \beta I_B$$

- But transistors can also be used like a switch…

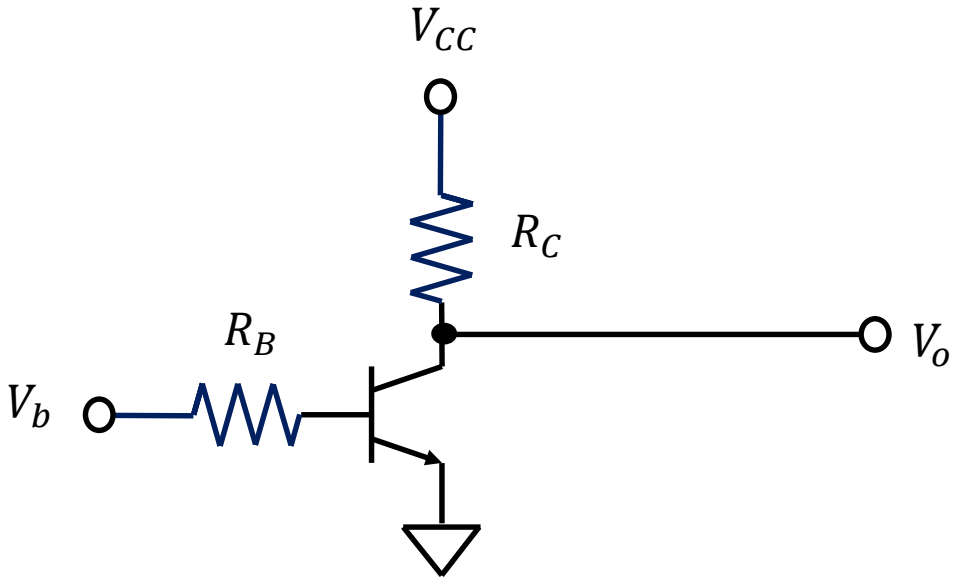
Transistors as a Switch

- Typical common emitter amplifier:



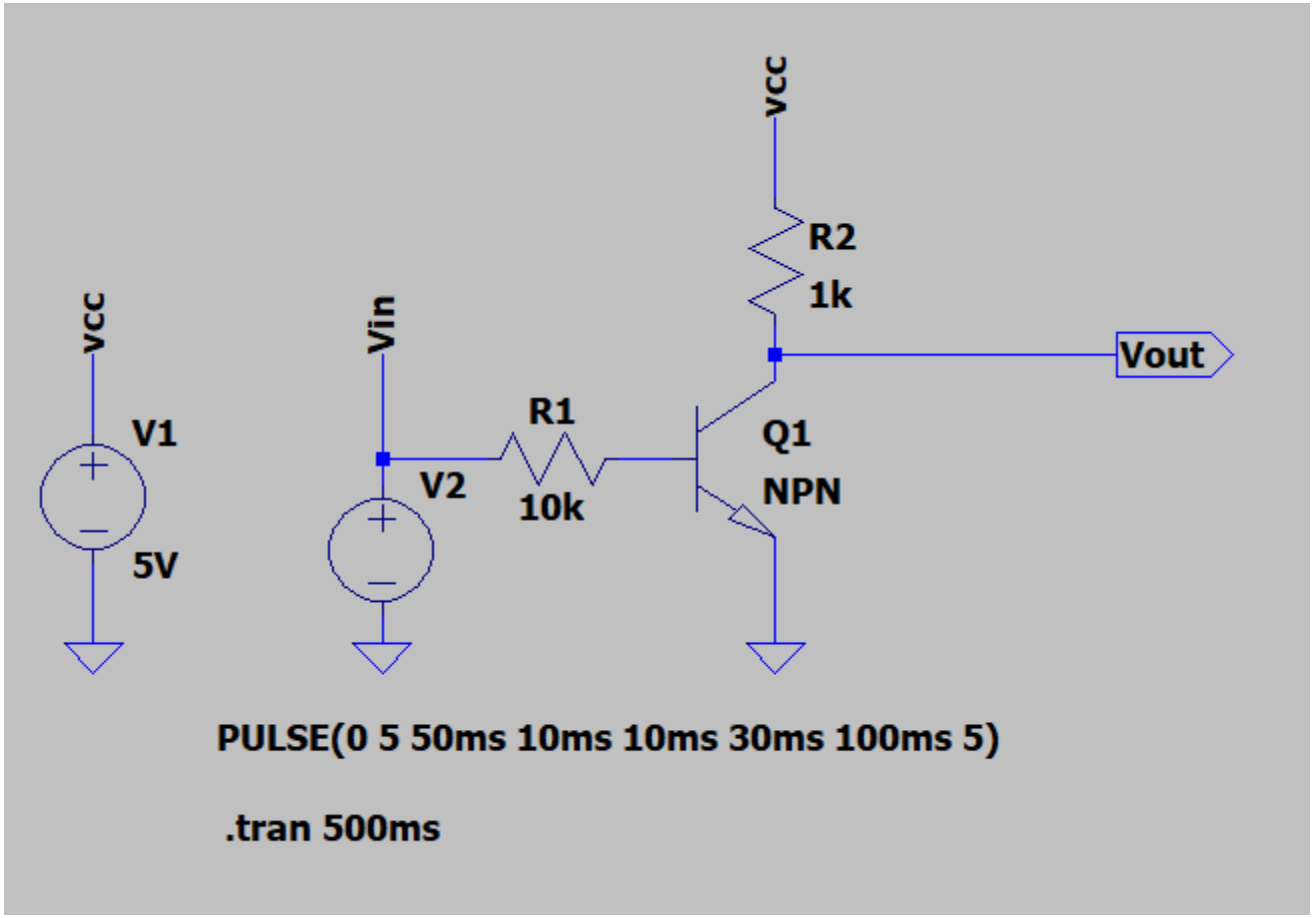
- When $V_b < V_{be} \approx 0.7\ V$, the base-emitter junction is not forward biased and no current flows.
- When $V_b > V_{be}$, the base-emitter junction is forward biased and a large current flows.

Transistors as a Switch

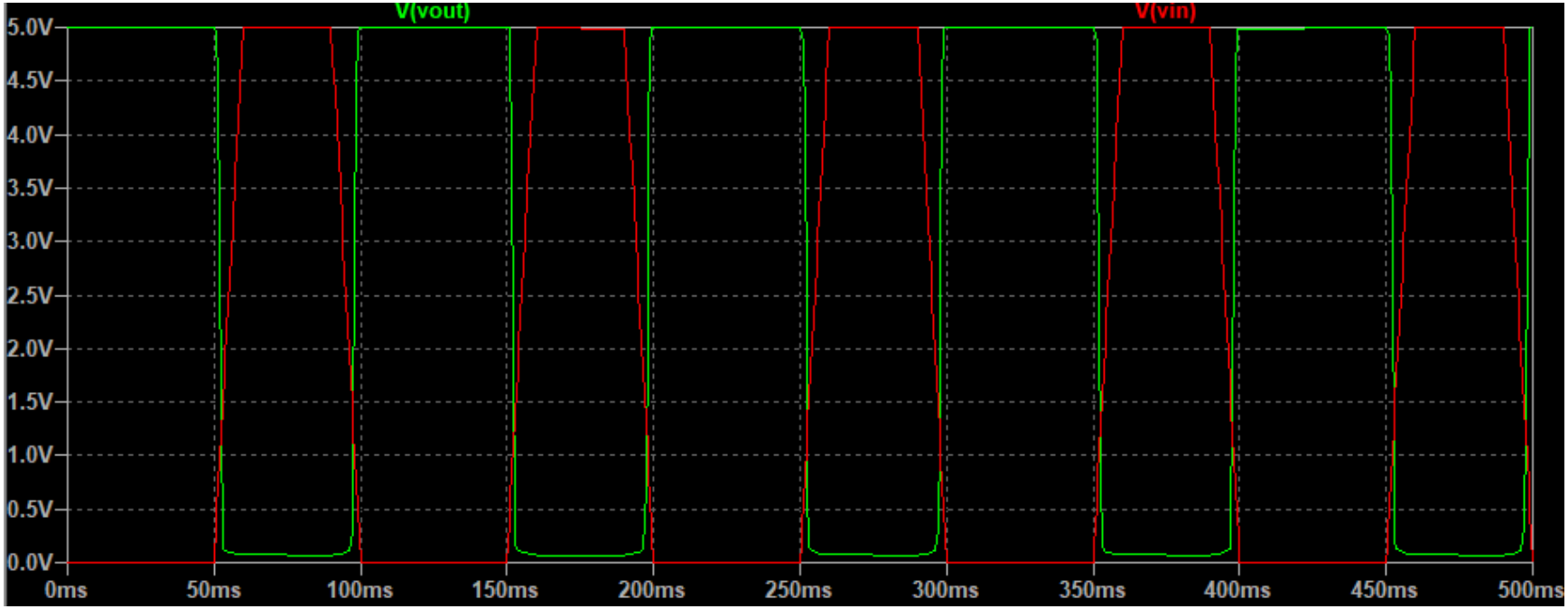


- When no current flows through R_C, $V_o = V_{CC}$
- When a large current flows through R_C, then $V_o \approx 0V$

Transistors as a Switch



Transistors as a Switch



- Because $R_E \approx 0$, the gain is very large
- The transition between conducting and non-conducting occurs very rapidly
- This circuit acts as an “inverter”

Logical Values as Voltages

- Voltages in this circuit can be anything from ground potential to V_{CC}.
- Nevertheless, we can use voltages to represent logical values of true and false.
- One (of many) conventions (TTL):

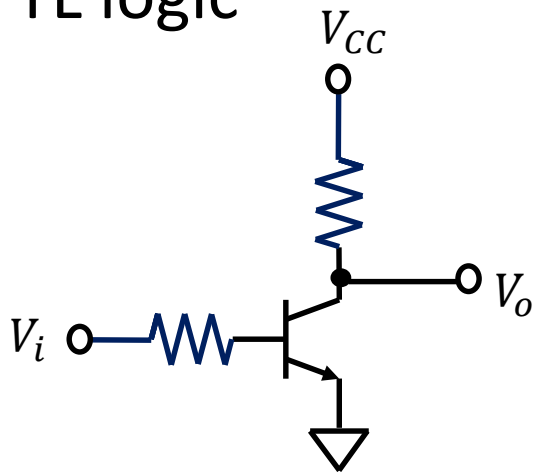
$$V_{in} < 0.8\ V = V_{IL}\ \text{(false)}$$

$$V_{in} > 2\ V = V_{IH}\ \text{(true)}$$

- The region between 0.8 and 2 V should be avoided except during transitions between true and false.

Logic Families

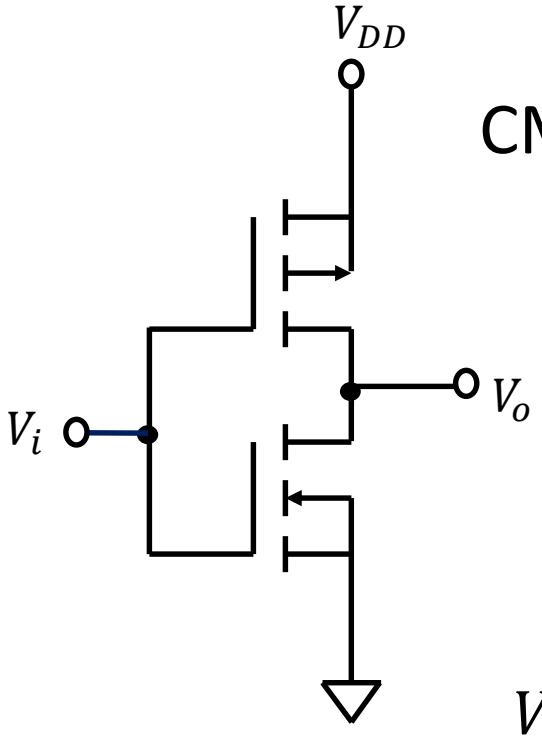
TTL logic



$V_{IH} = 2.0\ V$
$V_{IL} = 0.8\ V$

$V_{OH} = 3.3\ V$
$V_{OL} = 0.35\ V$

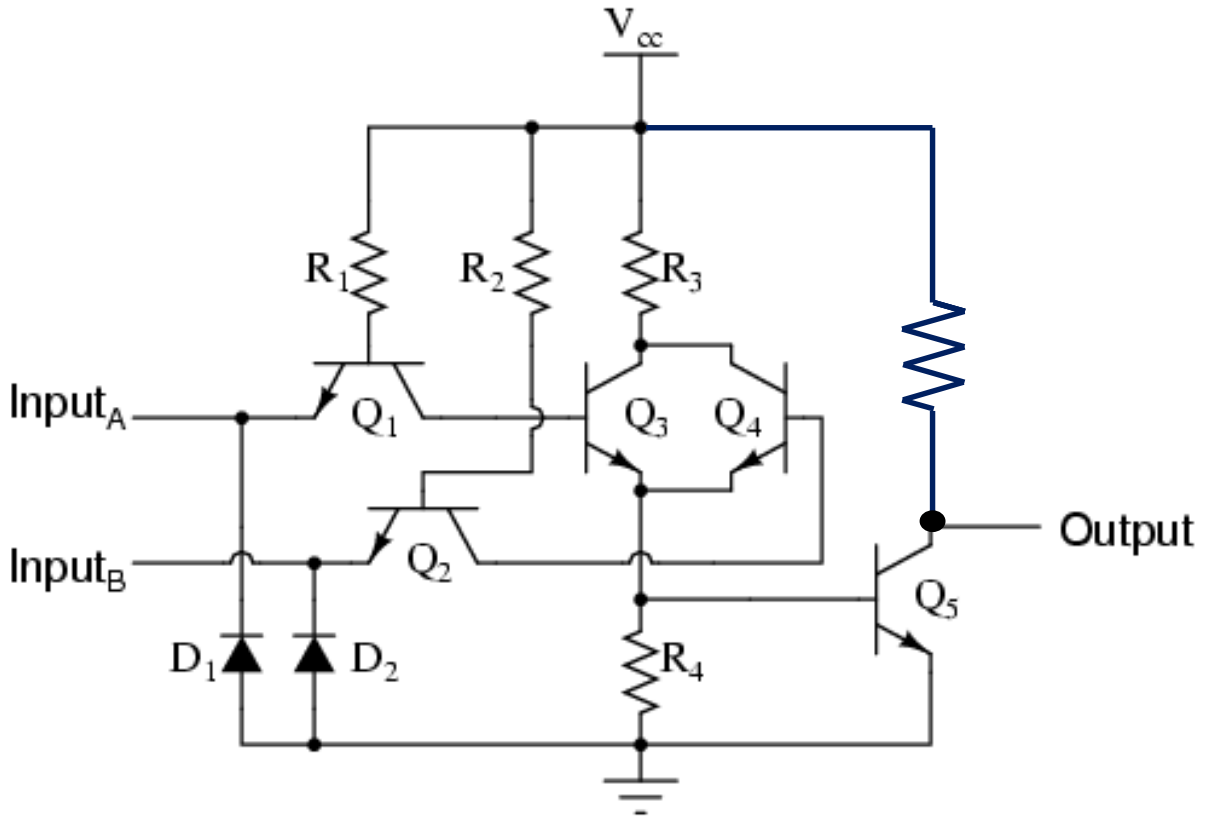
CMOS logic



$V_{IH} = 3.7\ V$
$V_{IL} = 1.3\ V$

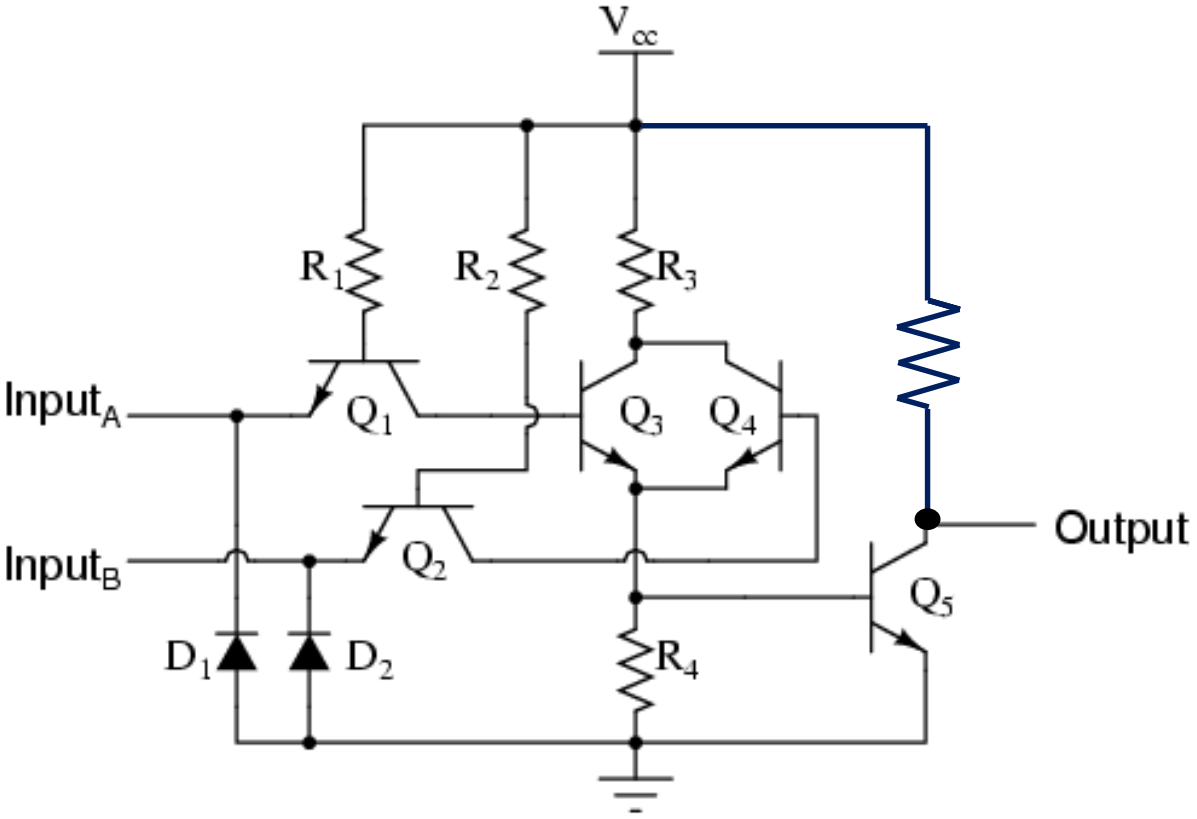
$V_{OH} = 4.7\ V$
$V_{OL} = 0.2\ V$

Another Digital Circuit



- Suppose both inputs are at ground potential.
- Then current is pulled out of the bases of Q3 and Q4
- Q3 and Q4 don't conduct any current
- The base of Q5 is at ground potential and the output is at V_{CC}

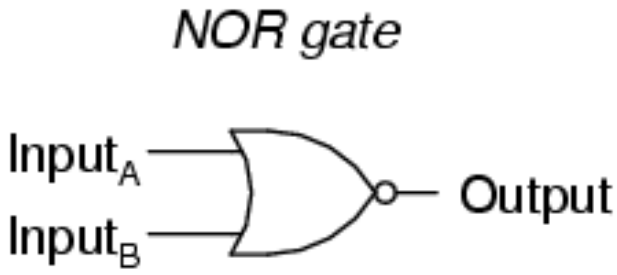
Another Digital Circuit



- Suppose either inputs is at V_{CC}
- Then current flows into the base of Q3 or Q4
- Current flows through R4 which raises the voltage at the base of Q5
- The output is pulled to near ground potential

NOR Gate

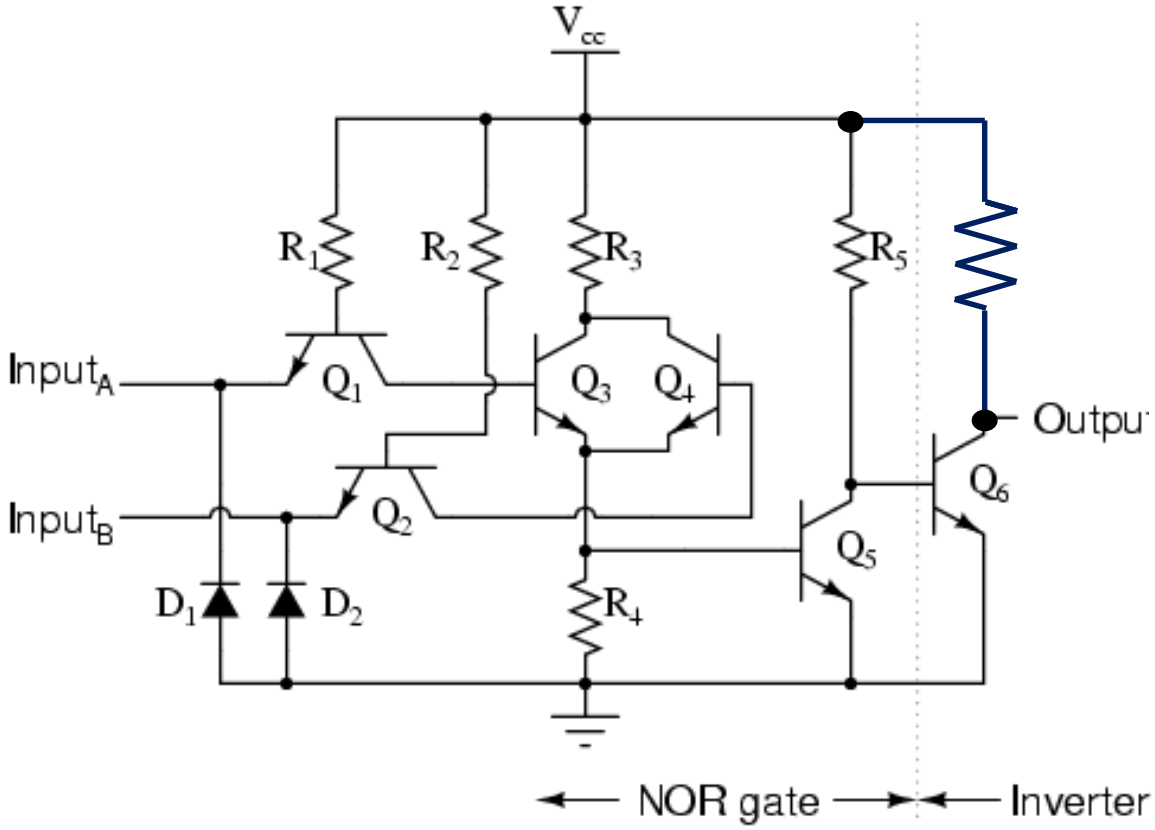
- This circuit performs the following logical operation where 0 = false (ground), 1 = true (V_{CC})



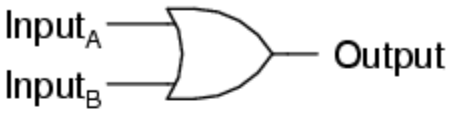
A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

OR Gate

- If we add an inverter at the output we get an OR gate:

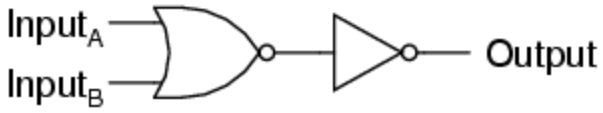


OR gate



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

Equivalent circuit



Boolean Algebra

- Boolean algebra uses variables that can only be true/false or 1/0.
- The “OR” operation is denoted with a plus “+”
- The “AND” operation is denoted with a dot “·”
- The negation operation is denoted with a bar above the expression
- This can be much easier than writing out a truth table for all possible inputs

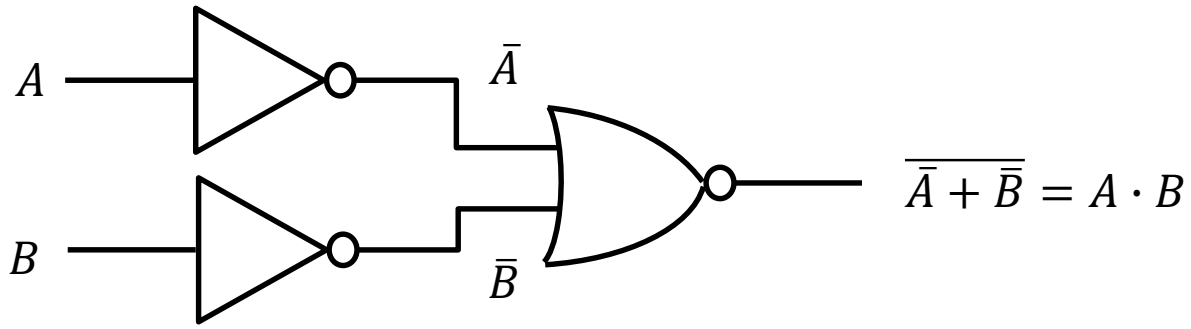
Boolean Algebra

- De Morgan's laws:

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

- We can use these to make other logic operations using NOR gates and inverters

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



Combinatorial Logic

- With a sufficient number of inverters and NOR gates, arbitrarily complex Boolean algebraic expressions can be implemented

$$o_j = f_j(i_1, i_2, \ldots, i_N)$$

- Multiple outputs $(1, \ldots, M)$ can be functions of multiple inputs $(1, \ldots, N)$

Digital Representation of Information

- Digital representation of information has advantages:
  - Immunity to noise
  - Arbitrarily complex (more or less) operations
  - Formal analysis techniques (other than differential equations)
- Boolean logic represents two possible states by a single voltage level (one “bit”)
- More complex information can be represented by multiple “bits”

Digital Representation of Information

- Unsigned integers (non-negative integers):

n	b_2	b_1	b_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

- In general, $n = \sum_k b_k 2^k$

Digital Representation of Information

- How many bits are needed?

$$N = \lfloor \log_2 n_{max} \rfloor + 1$$

$$\lfloor x \rfloor = \text{largest integer} \leq x$$

- Example:
  - To represent integers ≤ 15,

$$\log_2 15 = \frac{\log_e 15}{\log_e 2} = 3.907$$

  - Therefore, $N = 4$

Digital Representation of Information

$n_{max} = 2^N - 1$	N
15	4
255	8
1023	10
65,535	16
1,048,575	20
2,147,483,648	31
…	
9,223,372,036,854,775,807	63

Digital Representation of Information

- How to represent negative numbers?
- One's compliment:
  - Suppose $n = 0100$
  - Then $\bar{n} = 1011$ (one's compliment)
- Two's compliment
  - Suppose $n = 0100$. Then $-n = \bar{n} + 1 = 1100$
- Example: $4 + (-4) = 0$

$$\begin{array}{r} 0100 \\ 1100 \\ \hline 0000 \end{array}$$

Digital Representation of Information

- More complicated example: $n = -105$
  - 105 needs 7 bits: $\lceil \log_2 105 \rceil + 1 = 7$

$$105 = 64 + 32 + 8 + 1$$
$$= 2^6 + 2^5 + 2^3 + 2^0$$

  - One bit is needed for the minus sign
  - Binary representation:

$$105 = \underline{0}1101001$$

  - One's compliment:

$$\overline{105} = \underline{1}0010110$$

  - Two's compliment:

$$-105 = \underline{1}0010111$$

Binary Addition

- Addition can be performed using combinatorial logic

a	b	c (carry in)	sum	carry
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

Floating Point Representation

- Consider an arbitrary decimal fraction:

$$k = 1.38065 \times 10^{-23}$$

- This number has three parts:
  - An algebraic sign: (+)
  - The mantissa: 1.38065
  - The exponent: -23
- The binary representation is similar:

$$2^{-76} = 1.32349 \times 10^{-23}$$

$$2^{-77} = 6.6174449 \times 10^{-24}$$

$$2^{-78} = 3.3087225 \times 10^{-25}$$

…

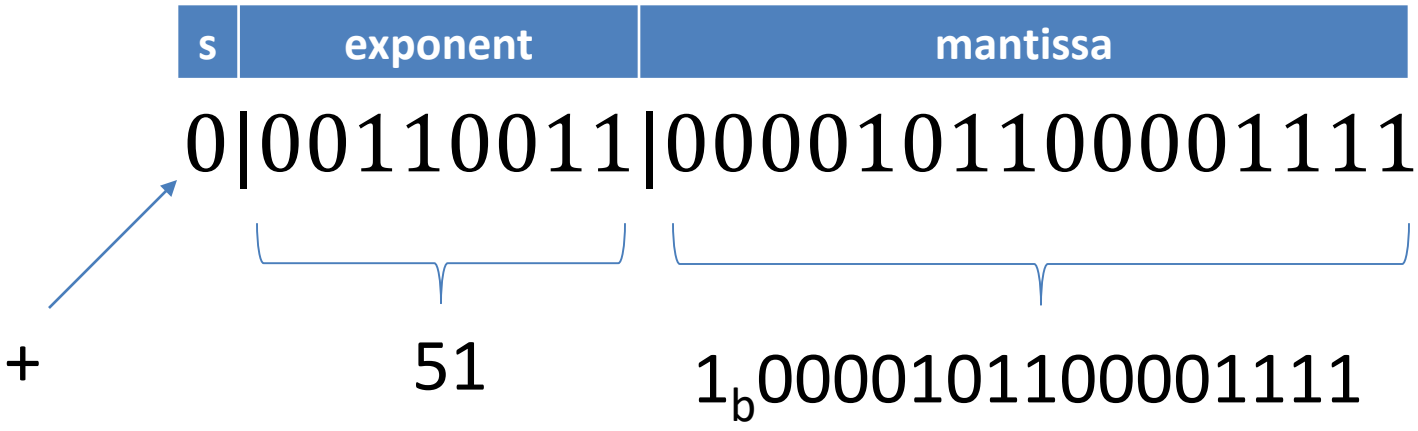
Floating Point Representation

$$k = 2^{-76} + 2^{-81} + 2^{-83} + 2^{-84} + 2^{-89} + \cdots$$
$$= (1_b 0000101100001111) \times 2^{-76}$$

- In this case, the exponent can be represented by an 8-bit integer.
- Instead of using 2's compliment representation for the exponent we can just add a "bias" to all exponents to make them positive integers.
  - If the bias was chosen to be 127 then the exponent would be (-76) + 127 = 51

Floating Point Numbers

- Notice that in this representation, there is always a 1 to the left of the binary point
- We don't need to store this 1 because it is always present in any number
- Representing $k = 1.38065 \times 10^{-23}$:



Floating Point Numbers

- Reversing the process:

  The true exponent is

$$e = 51 - 127 = -76$$

  Then,

$$k = (2^0 + 2^{-5} + 2^{-7} + 2^{-8} + 2^{-13} + \cdots) \times 10^{-76}$$
$$= 1.3806606 \times 10^{-23}$$

- Better accuracy can be achieved by using more than 16-bits in the mantissa
- IEEE 754 standard:

name	exponent	bias	mantissa	sign
binary32	8 bits	127	23 bits	1 bit
binary64	11 bits	1023	52 bits	1 bit

Representing Text Characters

- This requires defining a “code” which maps each character to a unique integer.
- There is no unique way to do this and historically, there are several codes that have been defined

Character	ASCII	Radix 50	EBCDIC
“@”	64	-	124
“A”	65	1	193
“B”	66	2	194
…			
“Z”	90	26	233

Representation of Characters

- ASCII: “American Standard Code of Information Interchange”
- Radix 50: Could store 3 characters in 16 bits (ASCII could store only 2)
- EBCDIC: “Extended Binary Coded Decimal Interchange Code”
- In practice, ASCII is by far the most common today.

Representation of Characters

- ASCII works well for Latin characters, but other languages have more than 255 characters.
- Unicode uses 16 bits to represent a character

Hindu-Arabic Numeral	Chinese/ Japanese numeral	Unicode
1:	一	&#x4e00;
2:	二	&#x4e8c;
3:	三	&#x4e09;
4:	四	&#x56db;
5:	五	&#x4e94;
6:	六	&#x516d;
7:	七	&#x4e03;
8:	八	&#x516b;
9:	九	&#x4e5d;
10:	十	&#x5341;
11:	十一	十一
12:	十二	十二
20:	二十	二十
50:	五十	五十
100:	百	百 (Japanese: hyaku, Chinese: bai)
1000:	千	千 (Japanese: sen, Chinese: qian)
10,000:	万	万 (Japanese: man)
10,000:	萬	萬 (Chinese: wan)
10^8:	億	億 (Japanese: oku)
10^8:	亿	亿 (Chinese: yi)
10^{12}:	兆	兆 (Japanese: chou, Chinese: jhao)