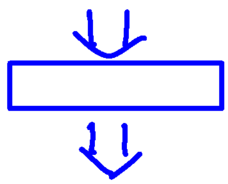
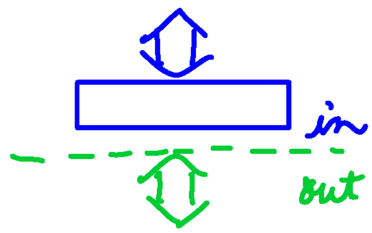
Common Digital Logic Components

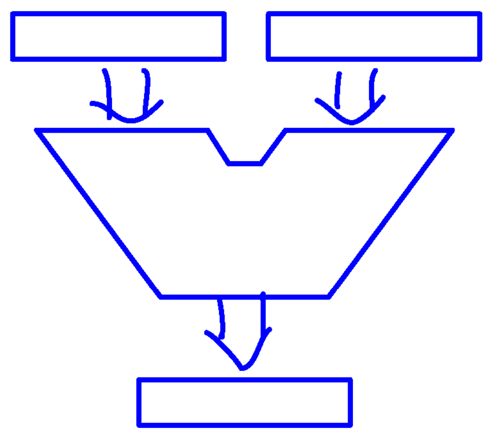
Register:



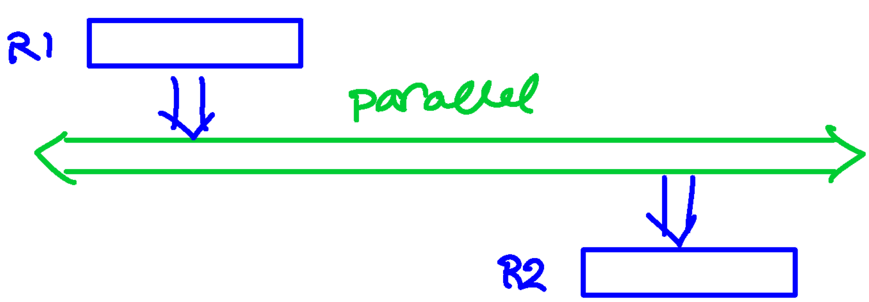
Port:



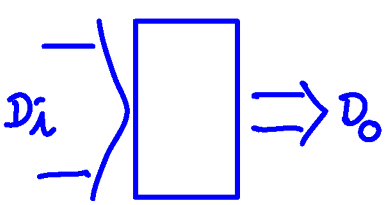
ALU:



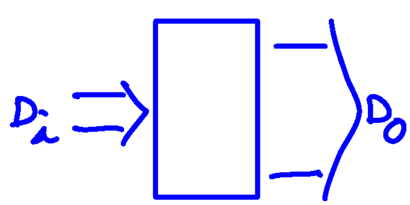
Bus:



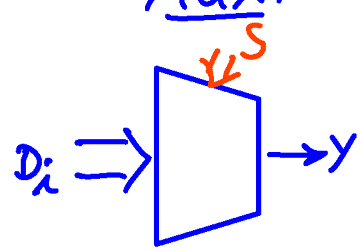
Encoder:



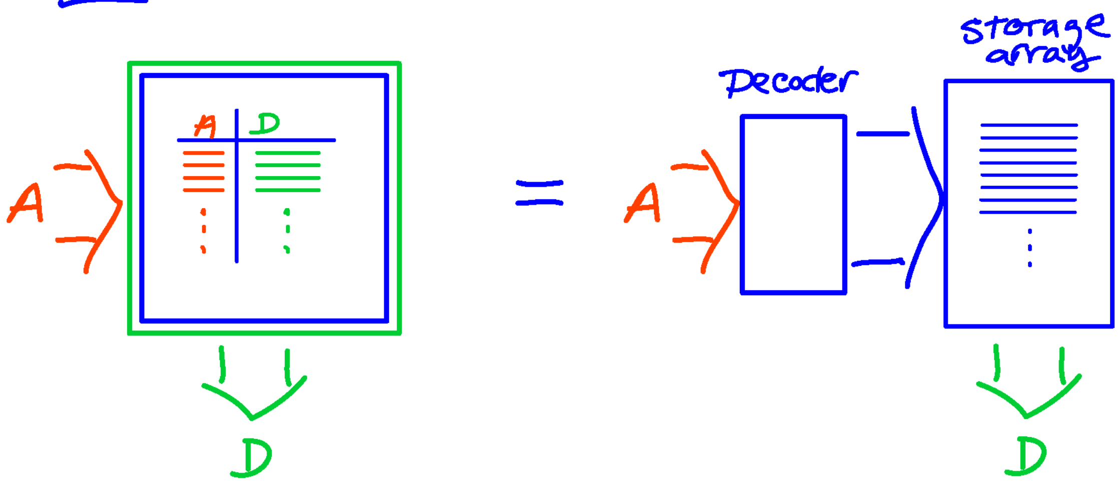
Decoder:



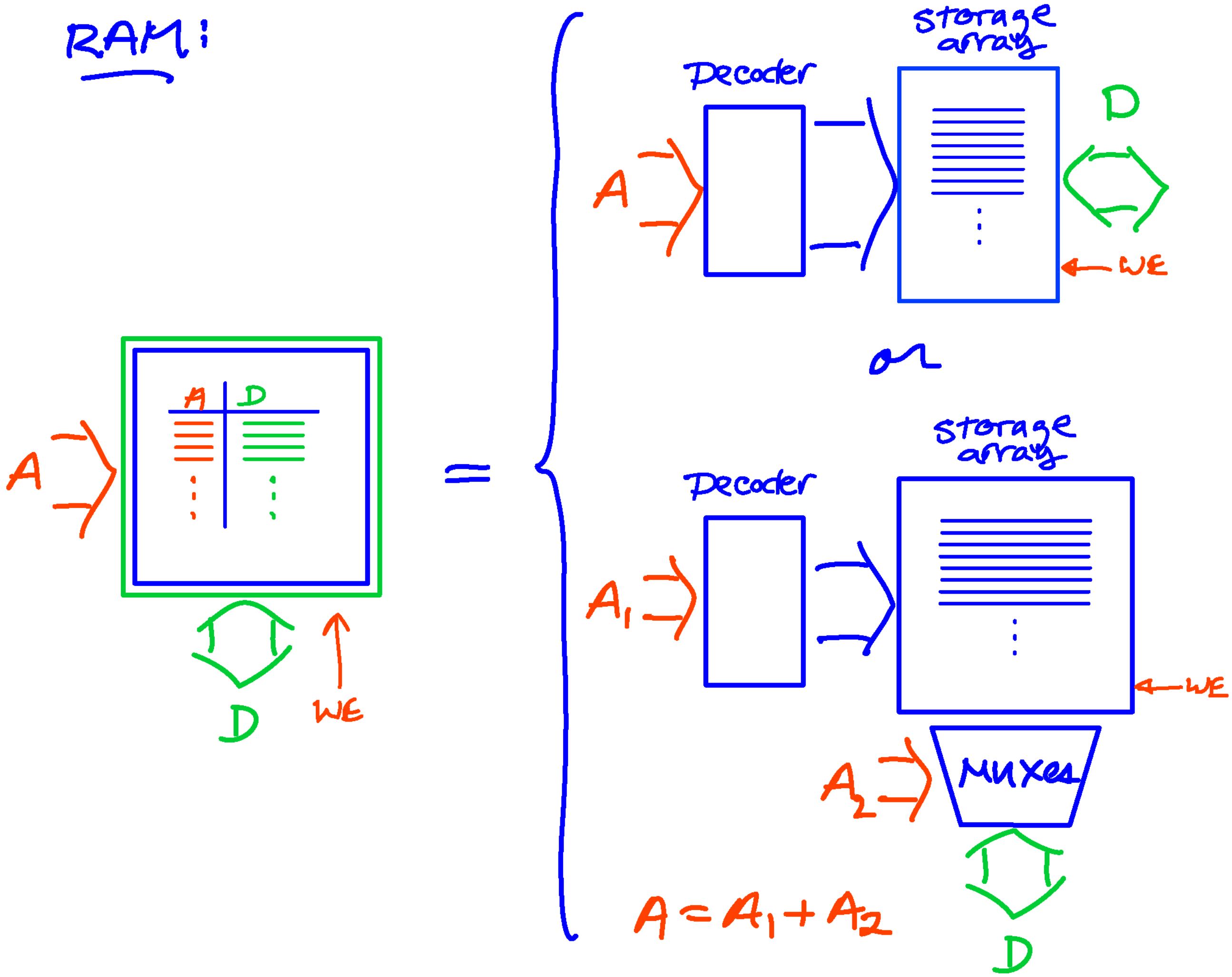
MUX:



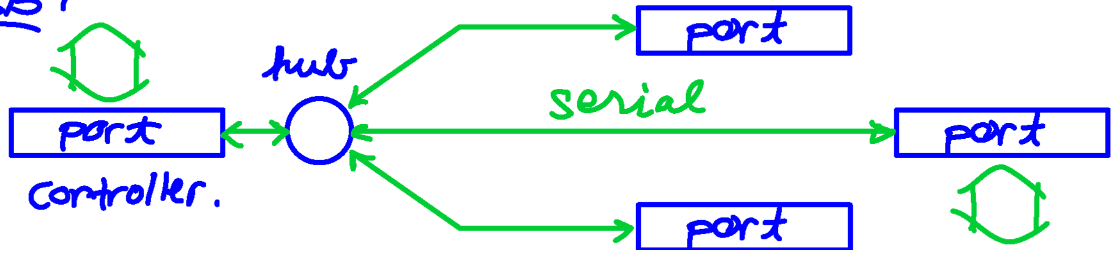
ROM:


A
A
D
=
A
Decoder
storage array
D

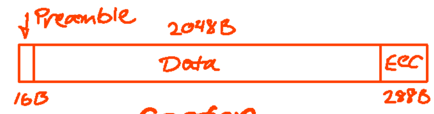
RAM:

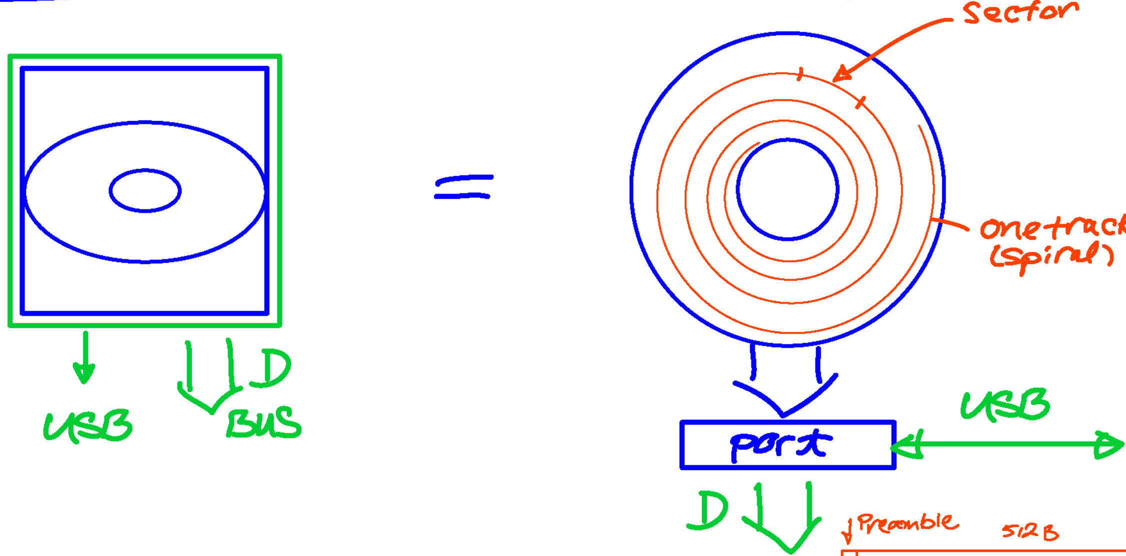

A
A
D
D
WE
=
A
Decoder
storage array
D
WE
or
A_1
Decoder
storage array
WE
A_2
Muxes
D
$A = A_1 + A_2$

USB:

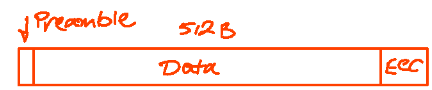

hub
port
port
serial
port
port
controller.

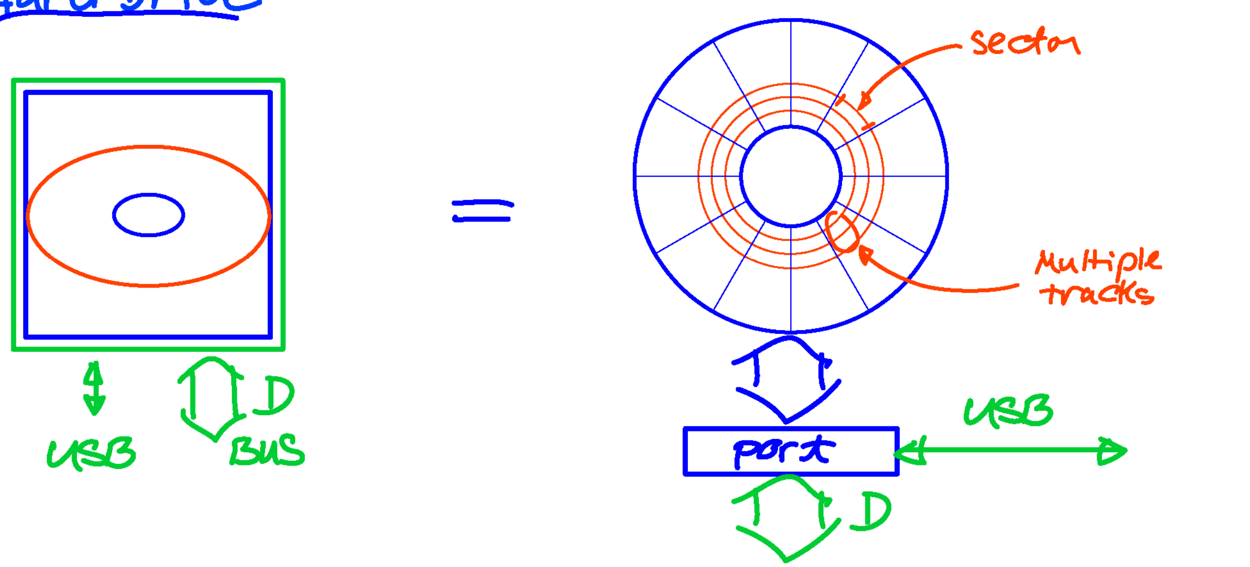
Optical Drive (CD/DVD ROM)



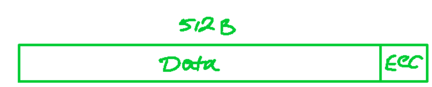


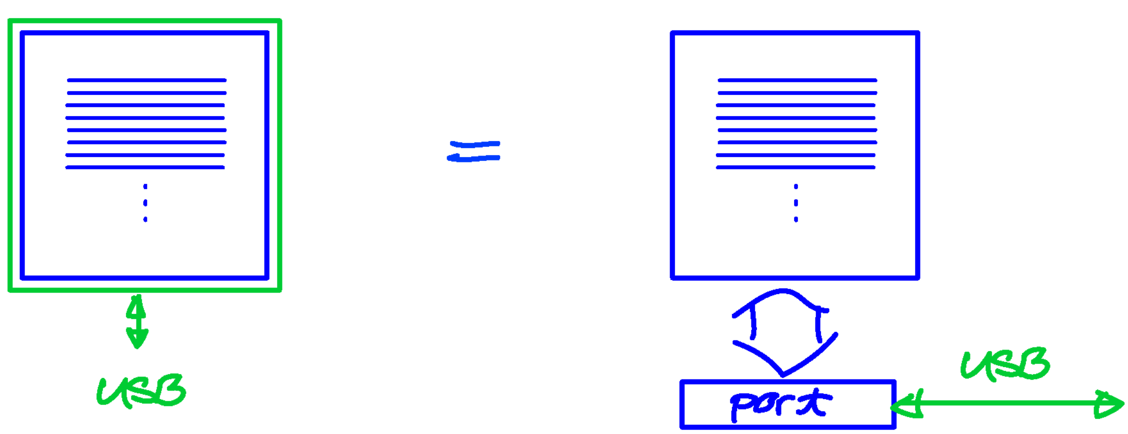
Hard Drive



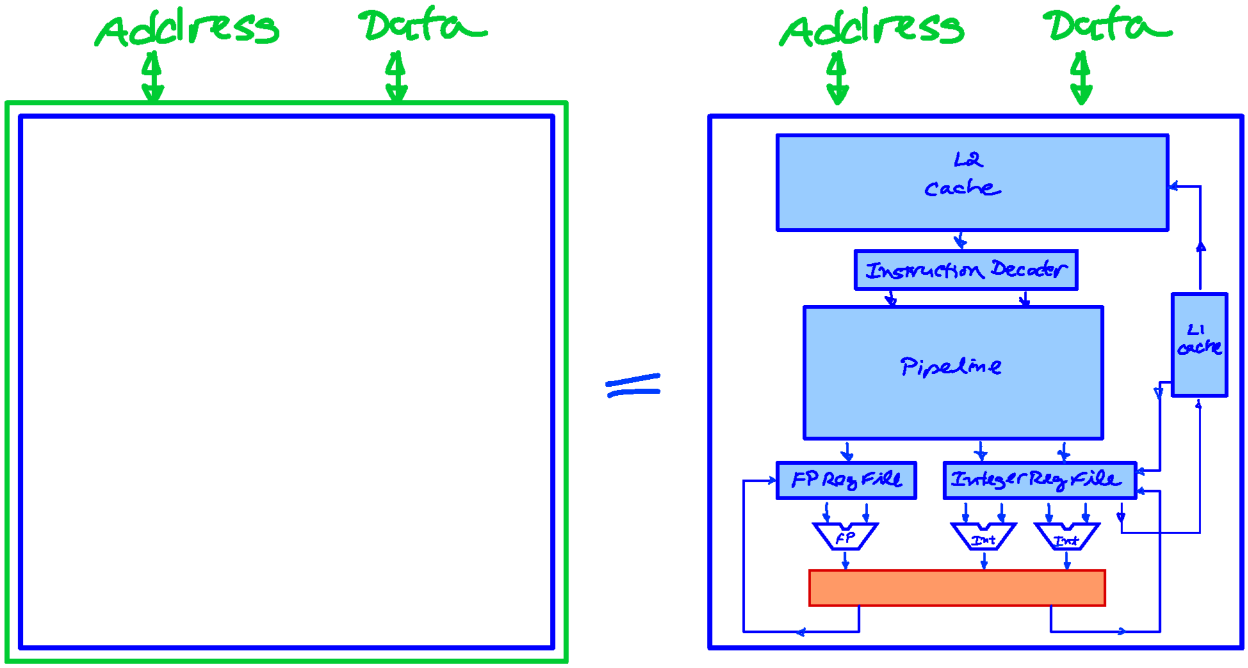


USB Drive (also Flash or SSD)

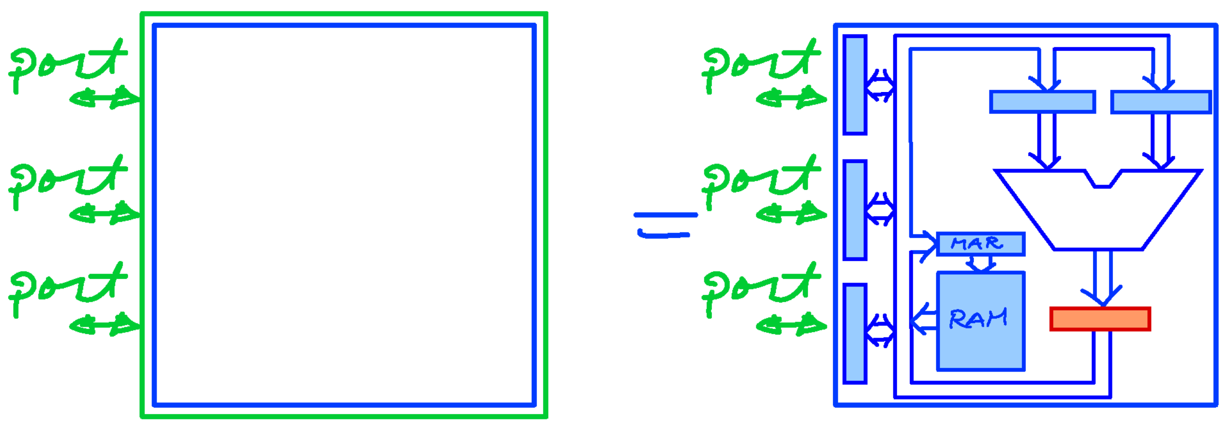




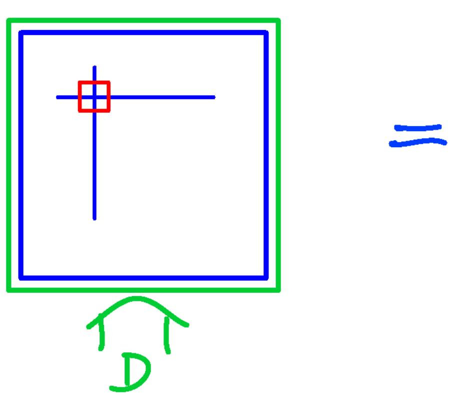
Microprocessor

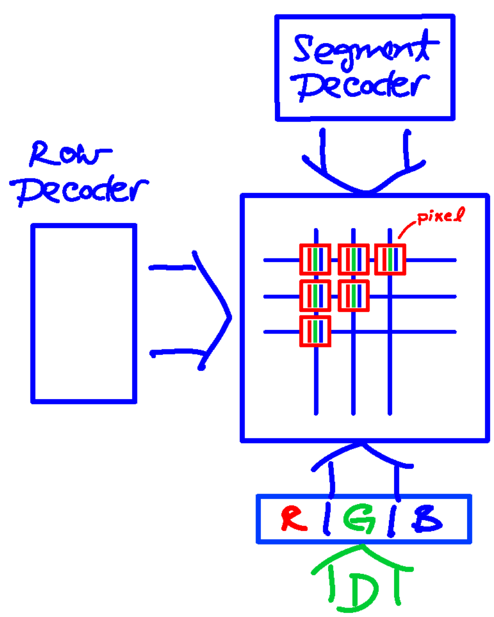


Microcontroller



LCD





Number Systems Table

Decimal	Binary	Octal	Hexadecimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

Red 1 and 2 = "Carry"
A,B,C,D,E,F = extra hex digits

Important number conversions to remember:
$(10)_{10} = (1010)_2 = (A)_{16}$
$(11)_{10} = (1011)_2 = (B)_{16}$

Juxapositional Notation

N = number

$= (a_{n-1}\ a_{n-2}\ \ldots\ a_2\ a_1\ a_0\ .\ a_{-1}\ a_{-2}\ a_{-3}\ \ldots\ a_{-m})_r$

Integer | *Fraction*

radix point

a_i = *digits*

r = *radix*

$0 \leq a_i < r$, *always holds.*

Example: decimal number

$(353.12)_{r=10}$

= 3 hundreds
5 tens
3 ones
1 tenth
2 hundreths

Example: binary number

$(1010.01)_{r=2}$

= ?

How do we expand this number?

<u>Clue</u>: look at polynomial representation.

Polynomial Representation

$(353.12)_{r=10}$

$$
\begin{aligned}
&= 3 \times 10^{2} + \\
&\quad 5 \times 10^{1} + \\
&\quad 3 \times 10^{0} + \\
&\quad 1 \times 10^{-1} + \\
&\quad 2 \times 10^{-2}
\end{aligned}
$$

$(1010.01)_{r=2}$

$$
\begin{aligned}
&= 1 \times 2^{3} + \\
&\quad 0 \times 2^{2} + \\
&\quad 1 \times 2^{1} + \\
&\quad 0 \times 2^{0} + \\
&\quad 0 \times 2^{-1} + \\
&\quad 1 \times 2^{-2}
\end{aligned}
$$

General Polynomial:

$$
\begin{aligned}
N &= \text{number} \\
&= \sum_{i=-m}^{n-1} a_i\, r^i \\
&= a_{n-1}\, r^{n-1} + a_{n-2}\, r^{n-2} + \ldots + a_2\, r^2 + a_1\, r^1 + a_0\, r^0 \\
&\quad + a_{-1}\, r^{-1} + a_{-2}\, r^{-2} + a_{-3}\, r^{-3} + \ldots + a_{-m}\, r^{-m}
\end{aligned}
$$

$$N = N_I + N_F$$

Conversion to Base 10

N_α to N_{10}, by polynomial subsitution:

$(101101)_2 = ?_{10}$

$$
\begin{aligned}
= \ & 1 \times 2^5 + \\
& 0 \times 2^4 + \\
& 1 \times 2^3 + \\
& 1 \times 2^2 + \\
& 0 \times 2^1 + \\
& 1 \times 2^0
\end{aligned}
$$

$= (45)_{10}$

$(247.1)_8 = ?_{10}$

$$
\begin{aligned}
= \ & 2 \times 8^2 + \\
& 4 \times 8^1 + \\
& 7 \times 8^0 + \\
& 1 \times 8^{-1}
\end{aligned}
$$

$= (167.125)_{10}$

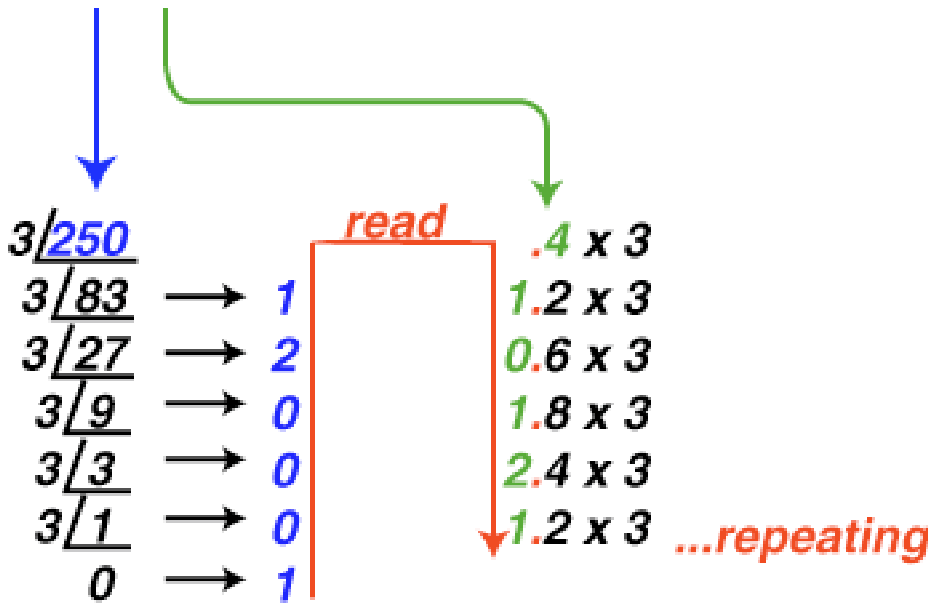
$(1AB)_{16} = ?_{10}$

$$
\begin{aligned}
= \ & 1 \times 16^2 + \\
& A \times 16^1 + \\
& B \times 16^0 \qquad \text{(Recall: } A = 10,\ B = 11\text{)}
\end{aligned}
$$

$= (427)_{10}$

Conversion to Base α

$(250.4)_{10} = (?)_3$



$(250.4)_{10} = (10021.10121012...)_3$

2^k Conversions

Octal: $2 \rightarrow 8 = 2^3$, in groups of $k = 3$

Hex: $2 \rightarrow 16 = 2^4$, in groups of $k = 4$

Example: Convert $(010101100)_2$ to base 8 and 16

Solution:

Add zeros

$$000(\overbrace{010}^{2}\,\overbrace{101}^{5}\,\overbrace{100}^{4})_2 \rightarrow (254)_8$$

$$\underbrace{0000}_{0}\,\underbrace{1010}_{A}\,\underbrace{1100}_{C} \rightarrow (AC)_{16}$$

Example: Convert $(110.110)_2$ to base 8 and 16

Add zeros

Solution:

Add zeros

$$0(\overbrace{110}^{6}.\overbrace{110}^{6})_2 0 \rightarrow (6,6)_8$$

$$\underbrace{0110}_{6}.\underbrace{1100}_{C} \rightarrow (6,C)_{16}$$

Adding and Subtracting Numbers

$$
\begin{array}{rr}
 & 1\,1\,1 \\
A & 00110 \\
+B & +\,01011 \\
\hline
C & x\;10001
\end{array}
\qquad
\begin{array}{r}
1 \\
+1 \\
\hline
10
\end{array}
\qquad
\begin{array}{r}
00110 \\
+\,01111 \\
\hline
10101
\end{array}
\qquad
\begin{array}{r}
10110 \\
+\,01111 \\
\hline
100101
\end{array}
$$

$\boxed{\ } =$ sign bit

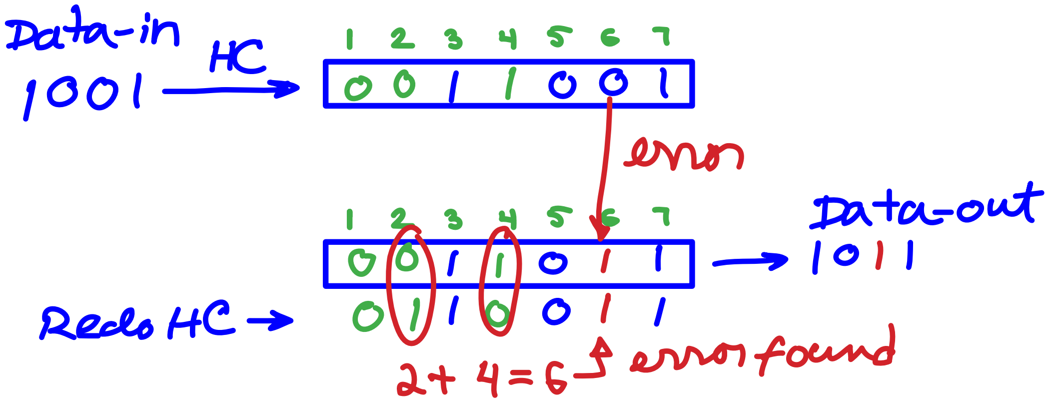
$$
\begin{array}{lrl}
 & N = 6 = & 00110 \\
\text{Invert:} & & 11001 \quad \text{step 1} \\
\text{Add + 1:} & & +1 \quad \text{step 2} \\
\hline
 & N' \;-6 = & 11010
\end{array}
\qquad
\begin{array}{rl}
7 = & 00111 \\
 & 11000 \\
 & +1 \\
\hline
-7 = & 11001
\end{array}
$$

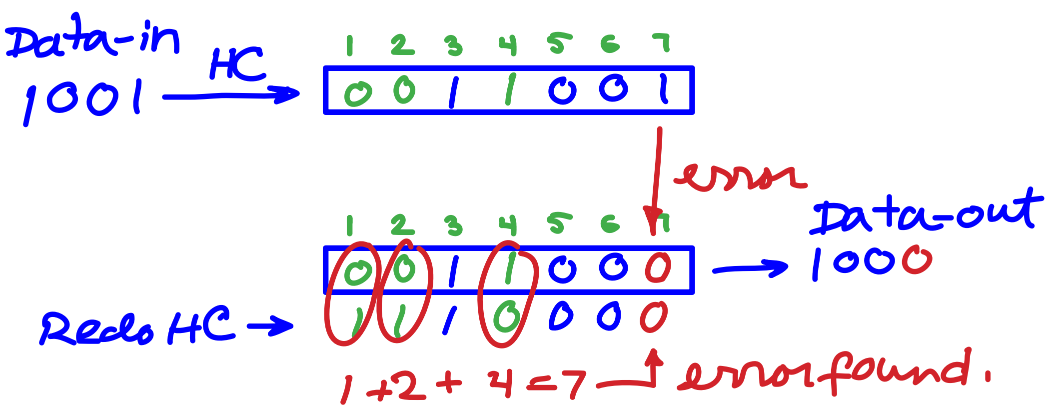
$$
\begin{array}{rl}
 & 1\,1\,1\,1 \\
7 = & 00111 \\
\underline{-6} = & 11010 \\
+1 = & 1\,00001
\end{array}
\qquad
\begin{array}{rl}
6 = & \\
\underline{-7} = & \\
-1 = &
\end{array}
\qquad
\begin{array}{rl}
 & 1\,1\,1\,1 \\
6 = & 00110 \\
\underline{-6} = & 11010 \\
0 & 1\,00000
\end{array}
$$

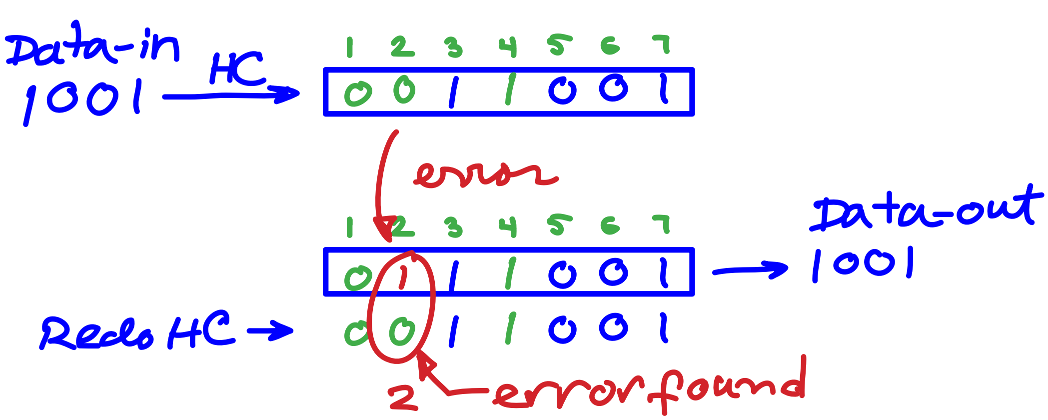
N′ = 11001 N′ = 01001

±N = ±N =

Hamming Code Error Correction





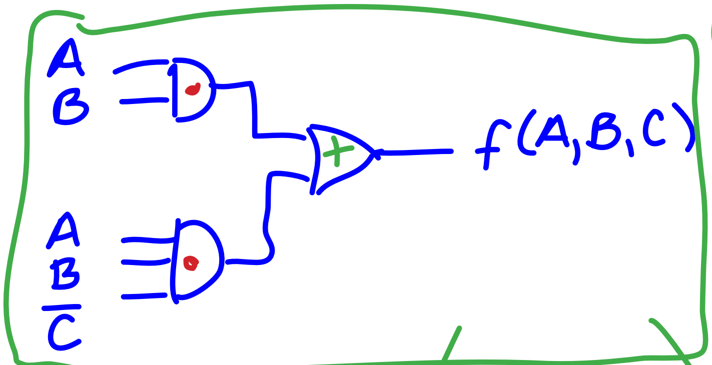


Simplification of Circuits using Boolean Algebra

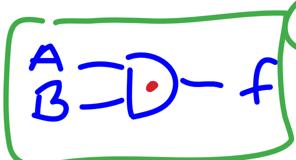
$$f(A,B,C) = \underbrace{A\cdot B}_{\text{products}} \overset{\text{sum}}{+} \underbrace{A\cdot B\cdot\overline{C}}_{\text{products}} = \text{SOP}$$

= sum of products

Circuit:



Simplifies



Equation:

Boolean Algebra (pattern matching)

$f = AB + AB\overline{C}$ ✓

$\underbrace{x + xy = x}_{\text{pattern}}$

Let $\begin{cases} x = AB \\ y = \overline{C} \end{cases}$

$x + xy = x = AB$

$\therefore f = \underline{\underline{AB}}$ ✓

Truth Table:

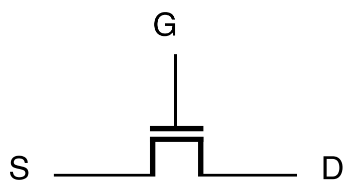
ABC	A·B	minterm $AB\overline{C}$	$f = AB + AB\overline{C}$
000	0	0	0 + 0 = 0
001	0	0	0 + 0 = 0
010	0	0	0 + 0 = 0
011	0	0	0 + 0 = 0
100	0	0	0 + 0 = 0
101	0	0	0 + 0 = 0
110	1	1	1 + 1 = 1
111	1	0	1 + 0 = 1

2 ← same → 1

AND-OR : wx / yz = w.c + y.z

NAND - NAND : wx / yz = w.c + y.z

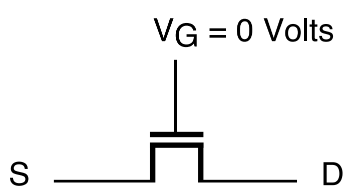
NMOS Transistors

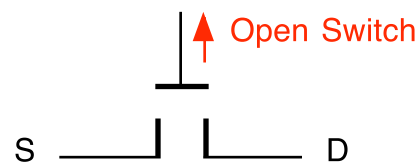


G = Gate
S = Source
D = Drain
(S and D are interchangeable)

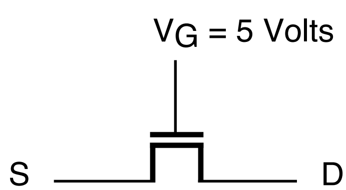
The NMOS transistor may be configured to operate in one of three different states, as determined by the **voltage** at the gate terminal V_G:

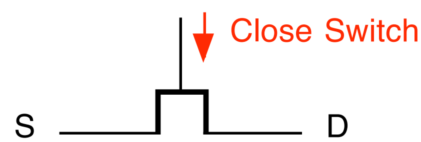
*1. **Off State** (Nonconducting)*



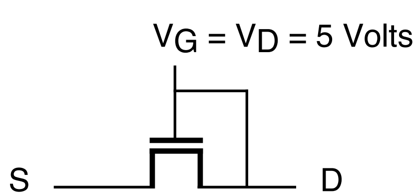


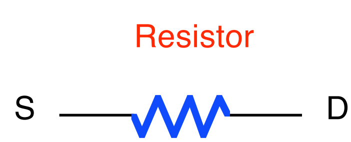
*2. **On State** (Conducting)*





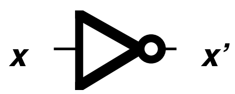
*3. **Resistive State** (Resistive)*

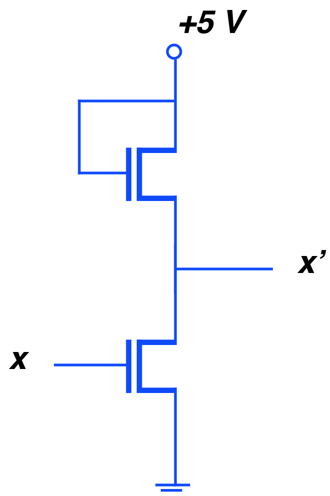




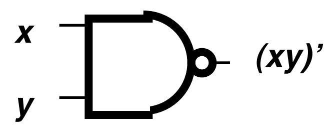
nMOS Based Logic Gates

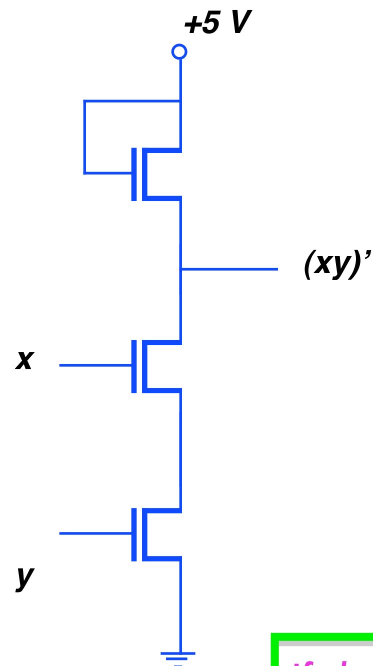
1. NOT Gate:





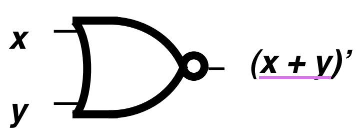
2. NAND Gate:

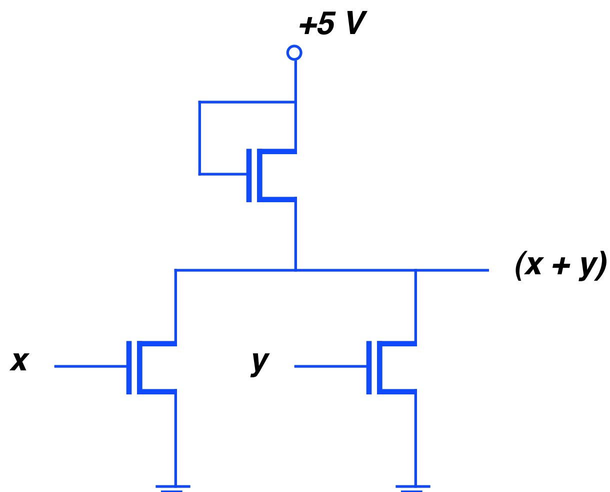




If change the y into x..then will get x'. x' = x'

3. NOR Gate:



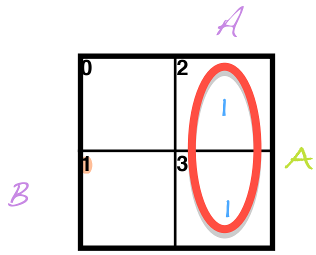


SOP = find minterm
POS = circle 0's & invert

Adjacency pattern :
x y + x y' = x

Karnaugh Maps

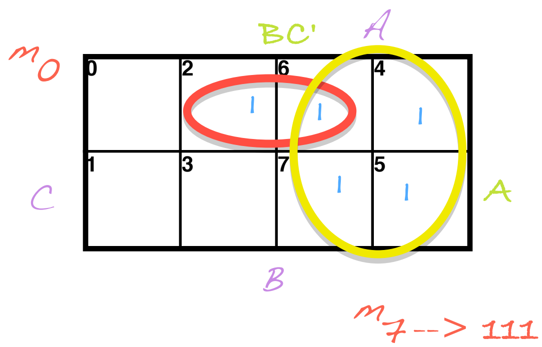
2 Variable Map:



$f = m2 + m3$
$= AB' + AB$
$= A$

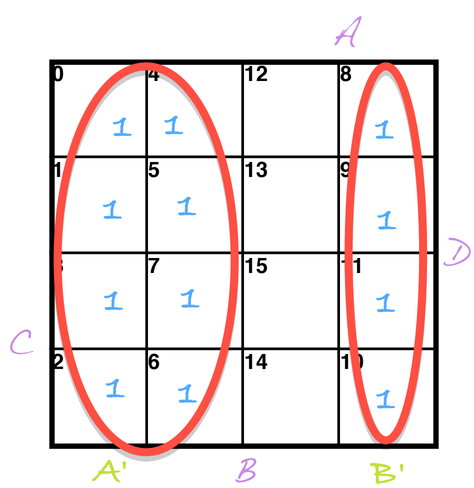
AB	f
0 0	0
0 1	1
1 0	1
1 1	0

3 Variable Map:



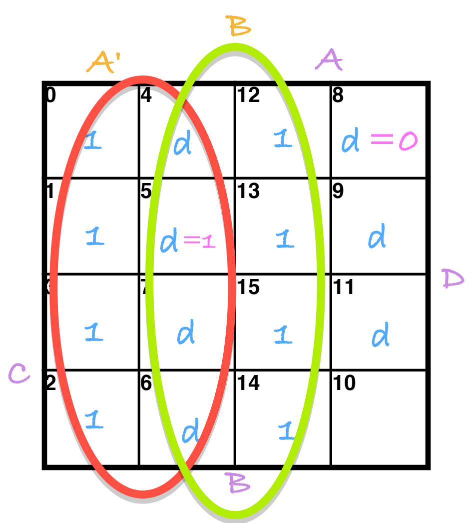
$f = m2 + m6$
$+ m4 + m5 + m7$
$= A + BC'$

4 Variable Map:



$f = \Sigma m(0,1,2,3,4,5,6,7,8,9,10,11)$
$= A' + B'$
$= A' \times B'$

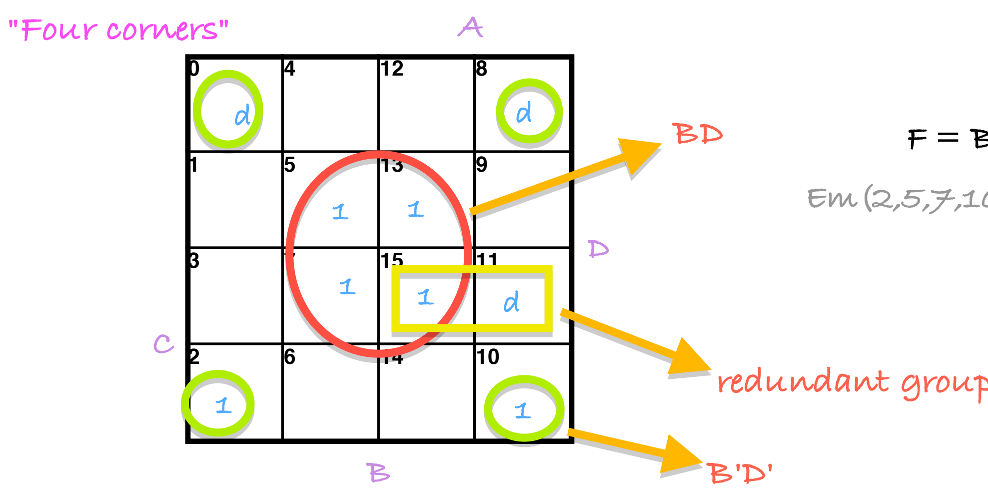
0	2	6	4
1	3	7	5



$d = x = \{0$
$1\}$

0	2	6	4
1	3	7	5

$xy' + xy = x$
(throw only y)

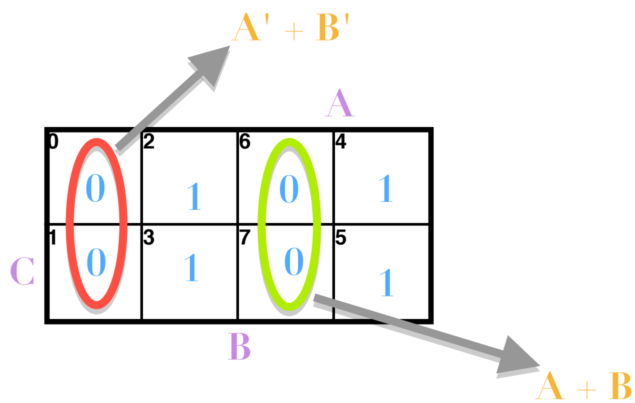


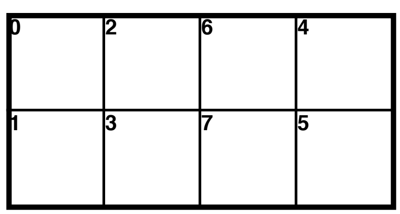
$F = BD + B'D'$

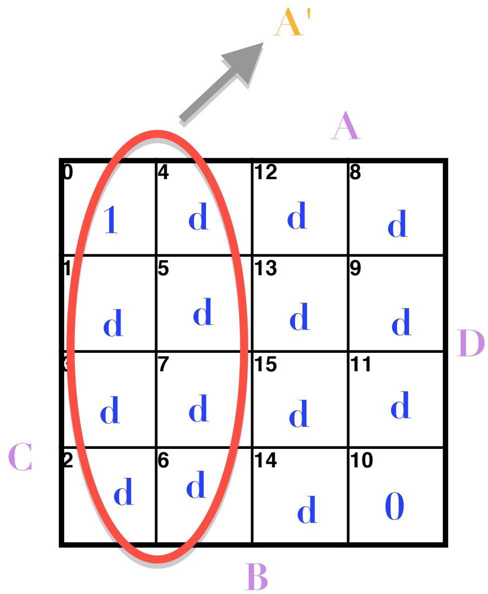
$\Sigma m(2,5,7,10,13,15) + \Sigma x(0,8,11)$

Identify the most simple POS.

Ans: (A + B)(A' + B')







The most simple expression is A'

0	4	12	8
1	5	13	9
3	7	15	11
2	6	14	10

5 Variable Map:

0	4	12	8
1	5	13	9
3	7	15	11
2	6	14	10

16	20	28	24
17	21	29	25
19	23	31	27
18	22	30	26

Maxterms: $f2(A,B,C) = (A+B+C)\ (A+B+C')\ (A'+B+C)\ (A'+B+C')$