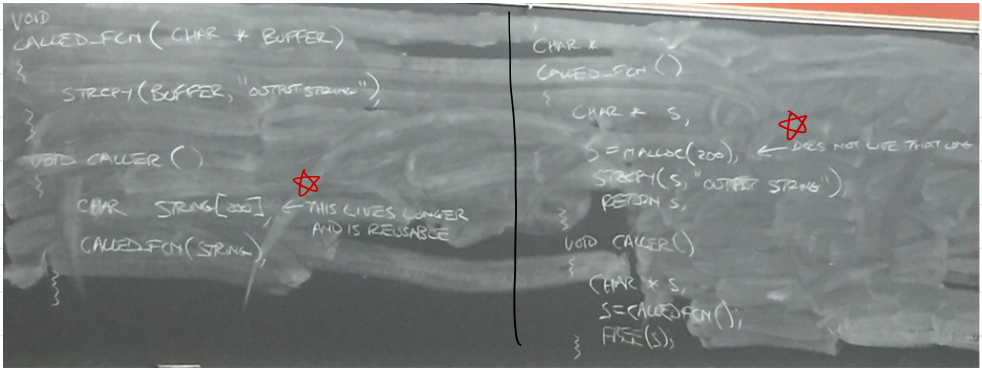
Ex)



08/07

char * strdup(const char *);

//malloc()s space for a copy of the arg string and copies the NUL-terminated arg string to the copy. Call is responsible for free()

strcpy (char * DST, const char * SRC)

//Copy to DST NUL-terminated SRC

#include <STDIO.H>

FILE * fopen(const char * pathname, const char * mode);

- "r" reading. Positioned at start of file.
- "r+" reading & writting. Also pos. at start of file.
- "w" Create new file or truncate existing file for writting
- "w+" open for reading or writting Create new file or truncate existing file
- "a" append. Create new file or at end of existing file.
- "a+" open for writting or reading. Create new file or at end of existing file.

size_T fread (void *ptr, size_T nelems, size_T elemsize, file*)

number of elements read

nelems * elemsize

EOF (end of file) and Error ⇒ return 0

int feof(file*);

1 if eof
0 otherwise

int ferror (file *);

1 if error
0 otherwise

size_t fwrite (const void *ptr, size_t nelems, size_t elemsize, file *)

↳ number of elements written

nelems * elemsize

int fseek (file *, long offset, int whence)

↳ offset from start of file or -1 for error

↳ seek_set
seek_cur
seek_end

//move around inside a file + rearrange position within the file

int fclose (file *); //flush output, close file

↳ 0 success
1 error

int fflush (file *);

//force all buffered output to the disk. If arg is 0, flush all open output streams

Transistors: Semiconductor device that amplifies and switches electronic signals

Semiconductor $\xrightarrow{\text{heat}}$ increase conductivity
Super conductor $\xrightarrow{\text{cool}}$ increase conductivity

*

- An electrical switch without moving parts
- Many transistors can be embedded in integrated circuits

* Microprocessors contain millions of transistors
* Logically, each transistor acts as a switch.

MOS (Metal Oxide Semiconductor) Transistor has 4 terminals:

① Source – current flows from

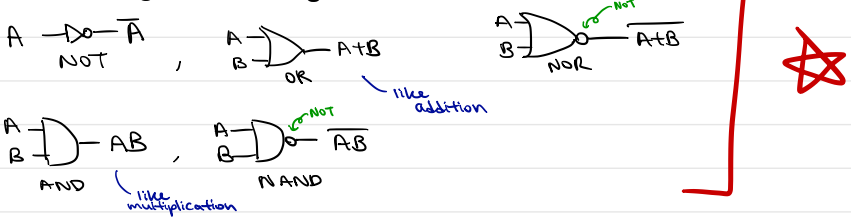
② Drain – current flows toward

③ Gate – has 0 or positive voltage

④ Body – substrate (underlying material)

	"0"	Illegal	"1"
Analog value	0 – 0.5 V		2.4 – 2.9 V

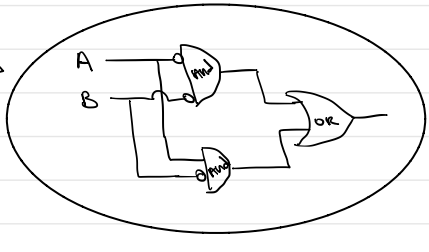
Basic Logic Gate Symbol:



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

$\overline{A}\overline{B} + A\overline{B} = 1$

OR'ed together all expressions with output 1.



Demorgan's Law:

To convert AND to OR (vice versa), invert inputs and output

Circuit Design & Logic Expressions

- Logical AND looks like multiplication.
- Logical OR looks like addition A+B+C
- Logical Negation: $\overline{A}$
- For N inputs, build a truth table with 2^N lines to account for all input possibilities

Truth Table:

A	B	C	F_1	F_2
0	0	0	1	1
0	0	1	1	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	0

Logic Expression

$F_1 = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + ABC$

$F_2 = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C$

Sum of products logic expression

Algebraic Rules for logic expressions

Commutative: $w+y = y+w$; $wy = yw$

Associative: $(w+y)+z = w+(y+z)$; $(wy)z = w(yz)$

Distributive: $w+yz = (w+y)(w+z)$; $w(y+z) = wy+wz$

Idempotent: $w+w = w$; $ww = w$

Involution: $\bar{\bar{w}} = w$

Complement: $w+\bar{w} = 1$; $w\bar{w} = 0$

Demorgan's Laws: $\overline{w+y} = \bar{w}\bar{y}$; $\overline{wy} = \bar{w}+\bar{y}$

Identities: $1+w = 1$; $0w = 0$

$0+w = w$; $1w = w$

$w+\bar{w}y = w+y$

*

DeMorgan's Law:

A pair of transformation rules that allow the expression of conjunctions (ANDs) and disjunctions (ORs) in terms of each other

1. The negation of a conjunction is the disjunction of the negations.

$\sim(P \wedge Q) \Leftrightarrow (\sim P) \vee (\sim Q)$

Negation — Conjunction AND — equivalent to — Disjunction

2. The negation of a disjunction is the conjunction of the negations.

$\sim(P \vee Q) \Leftrightarrow (\sim P) \wedge (\sim Q)$

$(P \wedge Q) \Leftrightarrow \sim(\sim P \vee \sim Q)$

$(P \vee Q) \Leftrightarrow \sim(\sim P \wedge \sim Q)$