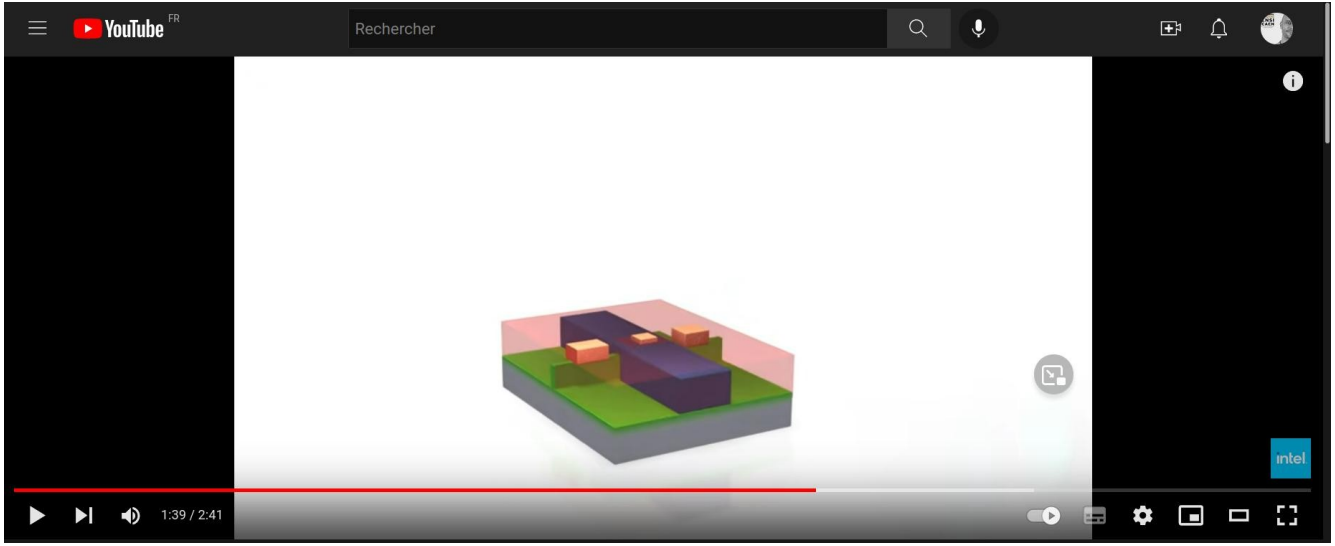
Architecture des ordinateurs

ENSI CAEN

ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

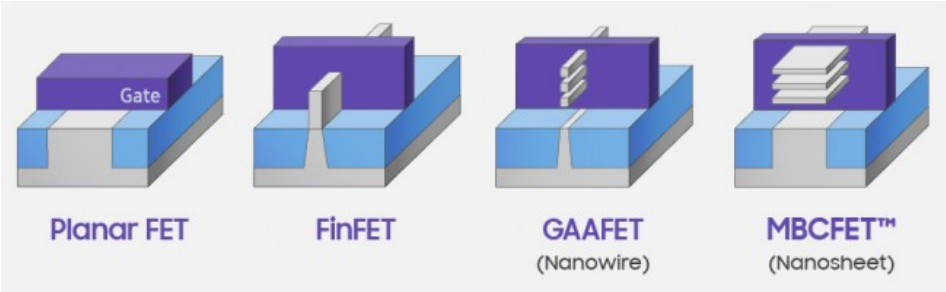
2021-2022

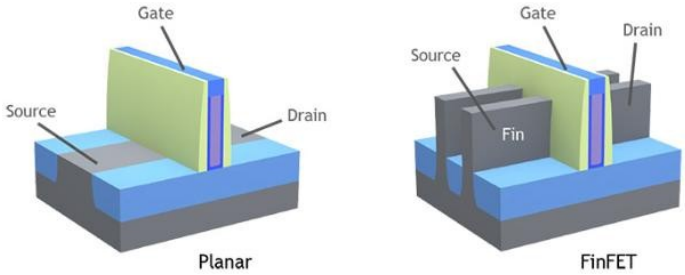
Du transistor MOS au Processeur !

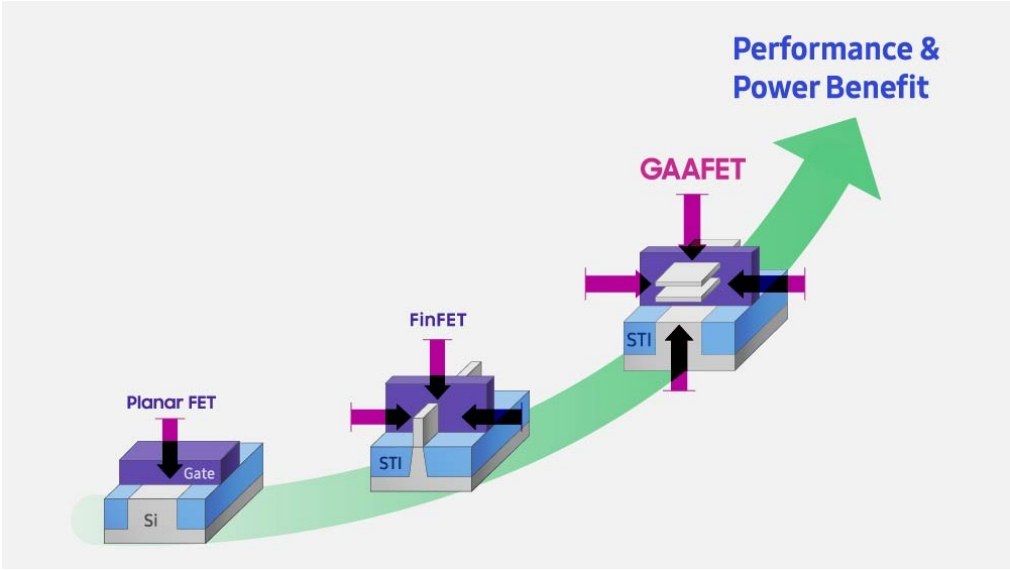


https://www.youtube.com/watch?v=d9SWNLZvA8g

Les technologies de transistors MOSFET







Les portes logiques (technologie FinFET – exemple Intel Trigate à notre époque)

YES

INPUT	OUTPUT
A	
0	0
1	1

NOT

INPUT	OUTPUT
A	
0	1
1	0

AND

INPUT		OUTPUT
A	B	
0	0	0
1	0	0
0	1	0
1	1	1

OR

INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	1

XOR

INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	0

NAND

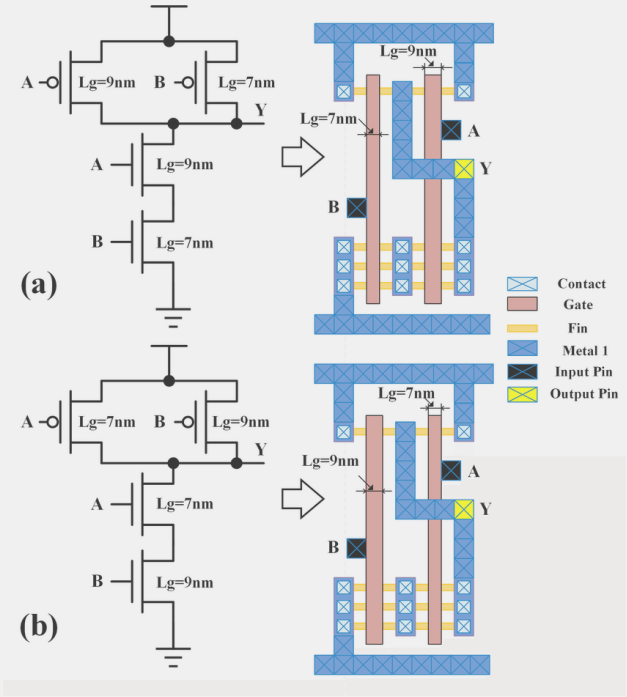
INPUT		OUTPUT
A	B	
0	0	1
1	0	1
0	1	1
1	1	0

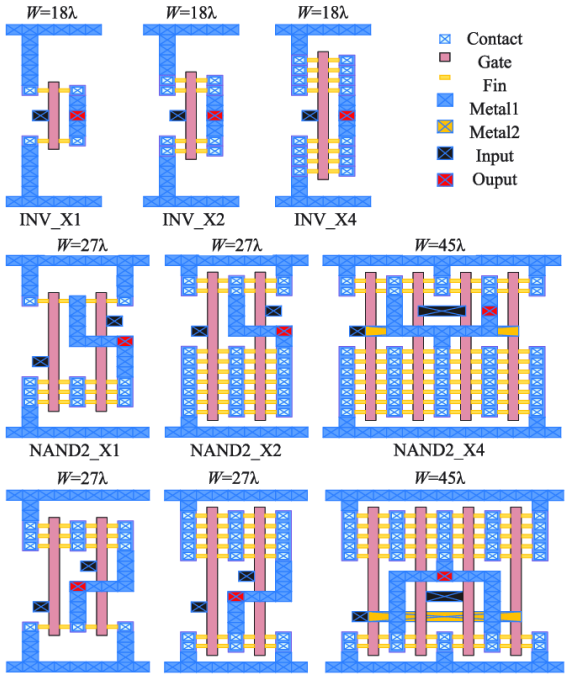
NOR

INPUT		OUTPUT
A	B	
0	0	1
1	0	0
0	1	0
1	1	0

XNOR

INPUT		OUTPUT
A	B	
0	0	1
1	0	0
0	1	0
1	1	1





La bascule D : Exemple mémoire SRAM à 1bit à 6 FinFET – Static RAM

D Flip-flop

Symbol

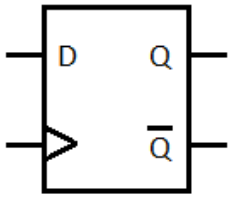
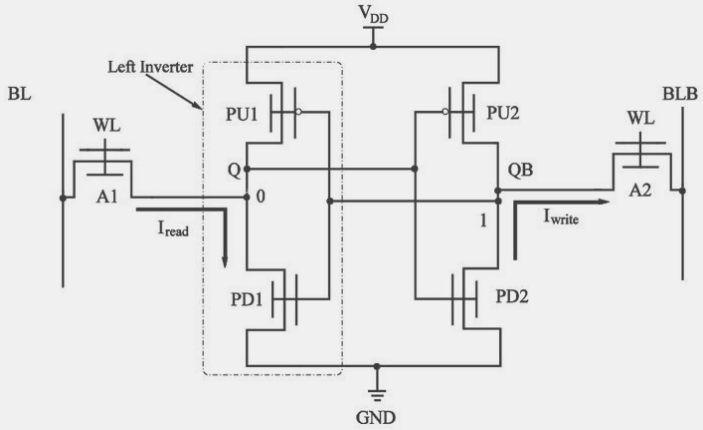
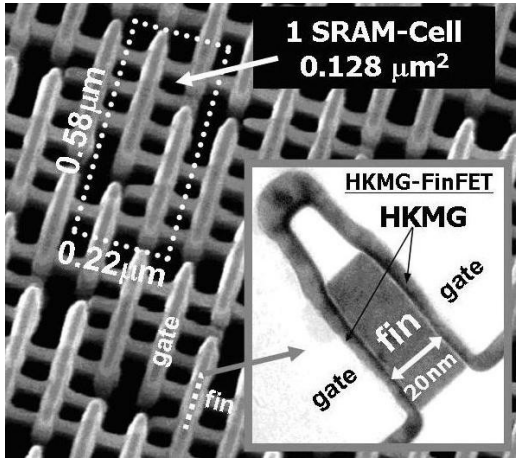


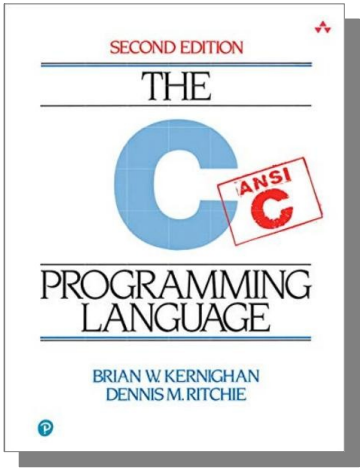
Table of truth:

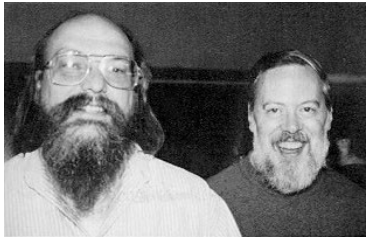
clk	D	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
0	1	Q	$\overline{Q}$
1	0	0	1
1	1	1	0





Langage C : Analyse de programmes simples (fonction, variable, pointeur, etc) !





```c
/* ANSI C standard syntax - 1989 */
void function_1 (void) ;
int function_2 (int a, int b, int c);

int main(void)
{

    function_1();

return 0;
}

void function_1 (void)
{
    int ret_1;

    ret_1 = function_2 (1, 2, 3);
}

/* K&R C original syntax - 1978 */
int function_2 (a_2, b_2, c_2)
int a_2;
int b_2;
int c_2;
{
    return a_2 + b_2 + c_2;
}
```

```c
#include <stdio.h>
#include <sys/resource.h>

void goto_segmentation_fault (char* pt_stack_bottom);

int main(void)
{
    char* pt_bottom_of_stack = (char*) &pt_bottom_of_stack;

    /*
    struct rlimit myprogram_ressources;
    myprogram_ressources.rlim_cur = 16000000;
    setrlimit (RLIMIT_STACK, &myprogram_ressources);
    */

    goto_segmentation_fault (pt_bottom_of_stack);

return 0;
}

void goto_segmentation_fault (char* pt_bottom_of_stack)
{
    char current_top_of_stack;

    printf("\rSize of stack = %lu ", pt_bottom_of_stack - &current_top_of_stack);

    fflush(stdout);

    goto_segmentation_fault (pt_bottom_of_stack);
}

```