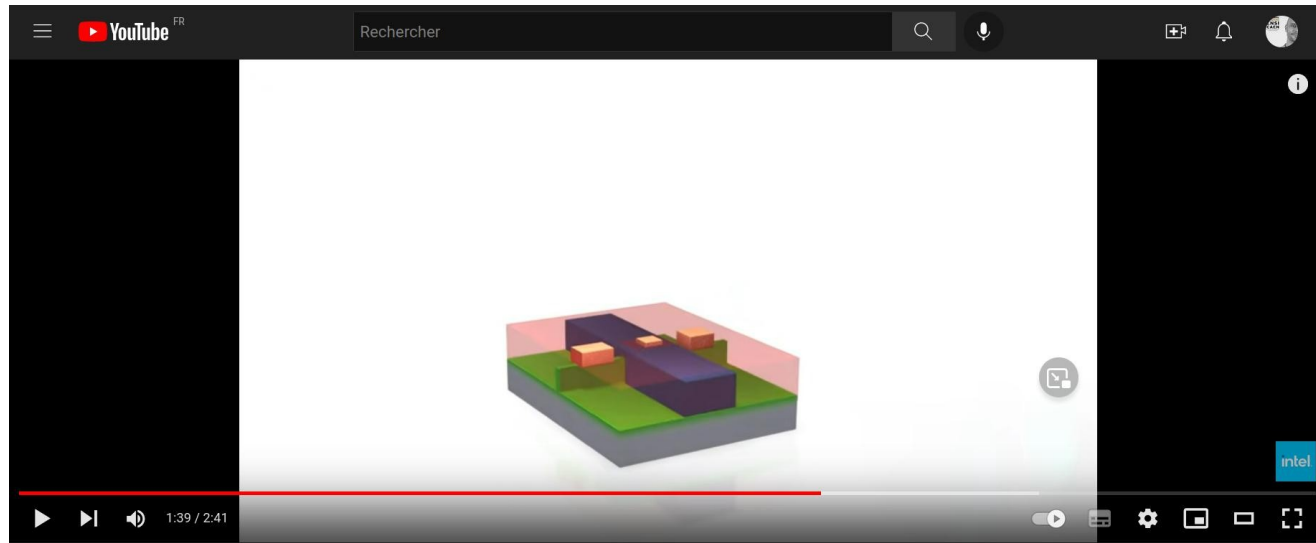


Systemes Embarqués



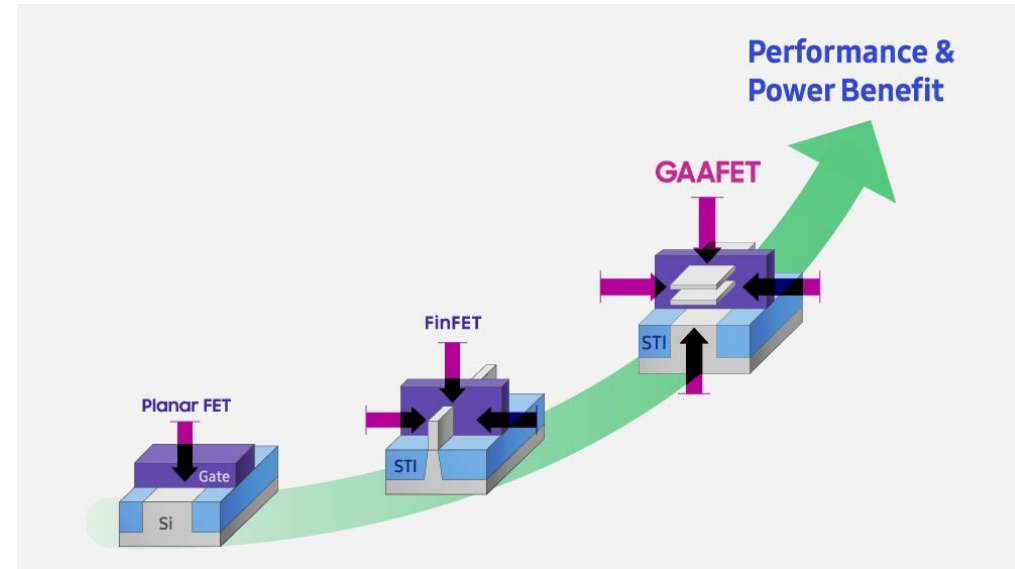
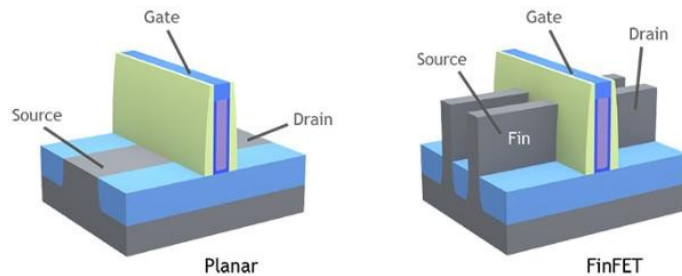
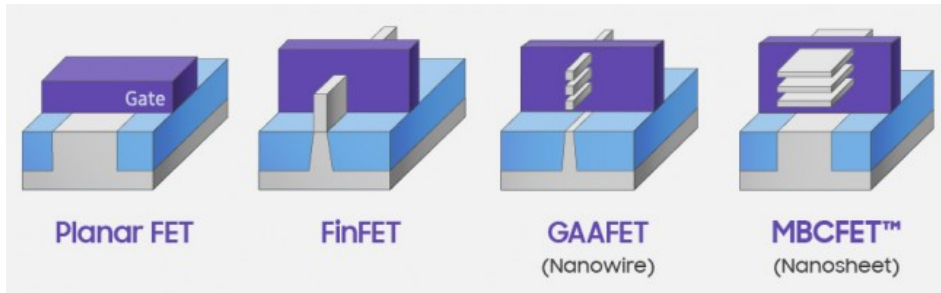
2021-2022

Du transistor MOS au Processeur !



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


Les technologies de transistors MOSFET



Pré-requis


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

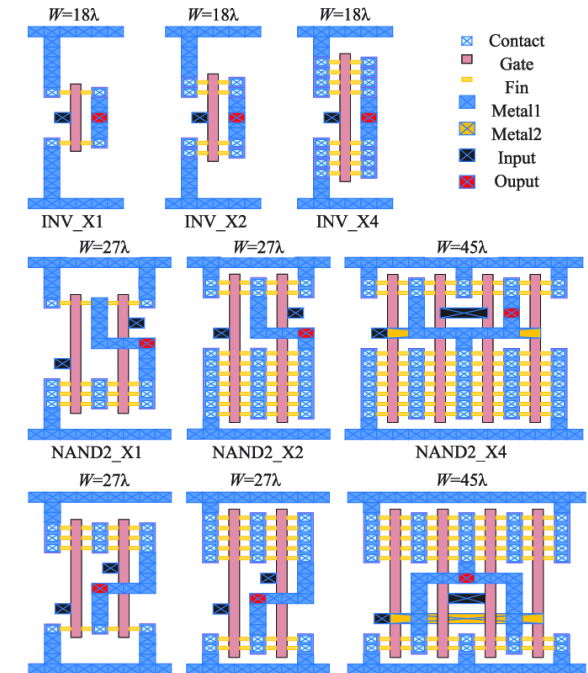
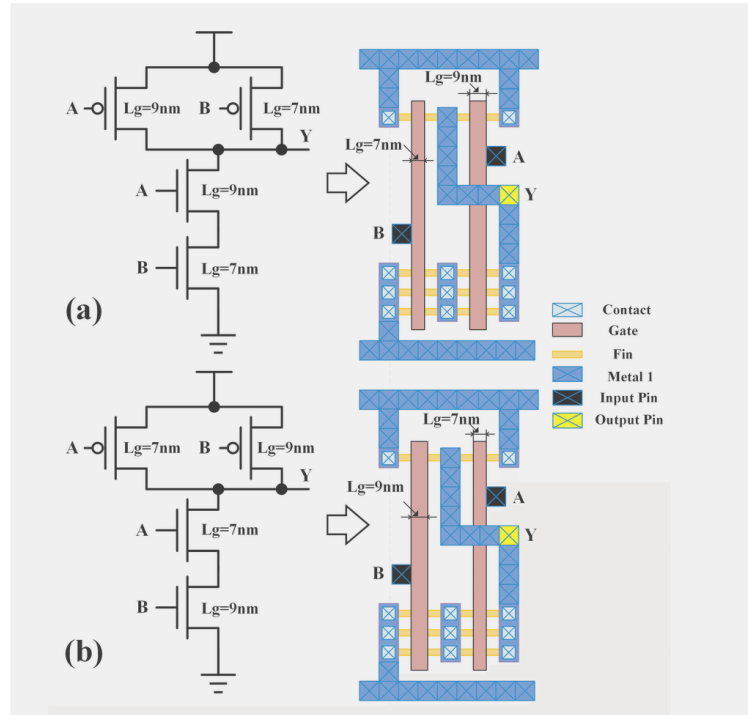


INPUT		OUTPUT
A	B	
0	0	1
1	0	0
0	1	0
1	1	0

XNOR



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



Pré-requis

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

D Flip-flop

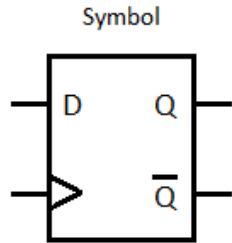
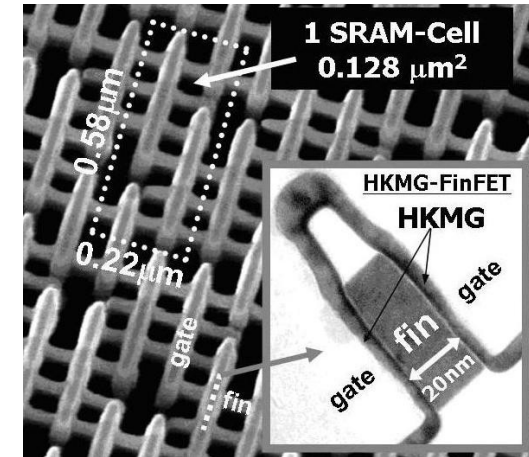
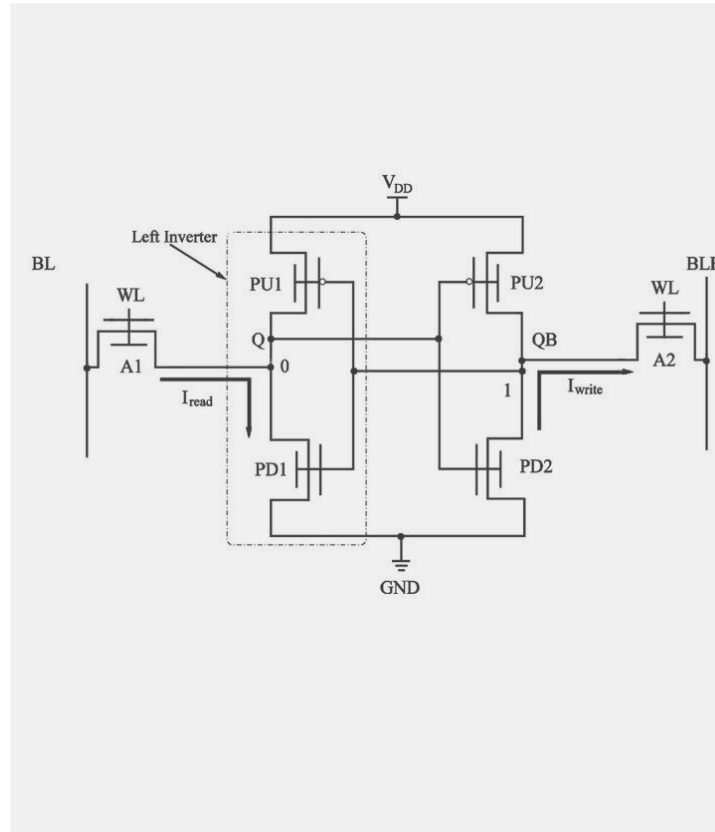
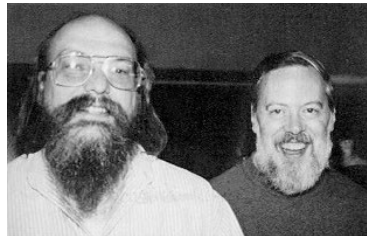
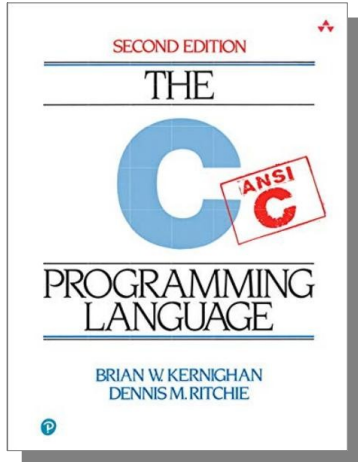


Table of truth:

clk	D	Q	\bar{Q}
0	0	Q	\bar{Q}
0	1	Q	\bar{Q}
1	0	0	1
1	1	1	0



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



```
1/* ANSI C standard syntax - 1989 */
2void function_1 (void) ;
3int function_2 (int a, int b, int c);
4
5int main(void)
6{
7    function_1();
8
9
10 return 0;
11}
12
13void function_1 (void)
14{
15    int ret_1;
16
17    ret_1 = function_2 (1, 2, 3);
18}
19
20/* K&R C original syntax - 1978 */
21int function_2 (a_2, b_2, c_2)
22int a_2;
23int b_2;
24int c_2;
25{
26    return a_2 + b_2 + c_2;
27}
```

```
1#include <stdio.h>
2#include <sys/resource.h>
3
4void goto_segmentation_fault (char* pt_stack_bottom);
5
6int main(void)
7{
8    char* pt_bottom_of_stack = (char*) &pt_bottom_of_stack;
9
10    /*
11     struct rlimit myprogram_ressources;
12     myprogram_ressources.rlim_cur = 16000000;
13     setrlimit (RLIMIT_STACK, &myprogram_ressources);
14     */
15
16    goto_segmentation_fault (pt_bottom_of_stack);
17
18 return 0;
19}
20
21void goto_segmentation_fault (char* pt_bottom_of_stack)
22{
23    char current_top_of_stack;
24
25    printf("\rSize of stack = %lu ", pt_bottom_of_stack - &current_top_of_stack);
26
27    fflush(stdout);
28
29    goto_segmentation_fault (pt_bottom_of_stack);
30}
31
```