

# Modèles et techniques en programmation parallèle hybride et multi-cœurs

Utilisation des accélérateurs de calcul

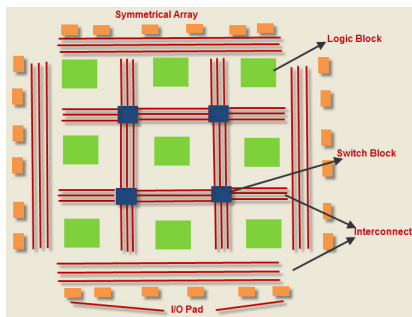
Marc Tajchman

CEA - DEN/DM2S/STMF/LMES

02/01/2021

# Circuits FPGA

Les circuits **FPGA** (**F**ield-**P**rogrammable **G**ate **A**rray) est constitué d'un ensemble de circuits électroniques (Logic Block dans la figure) connectés entre eux et dont les paramètres et les connexions (Switch Block, Interconnect) peuvent être changées par l'utilisateur.



Exemple de matrice de circuits FPGA

La façon d'utiliser les FPGA dépend de chaque dispositif.

En général, l'utilisateur écrit la description du système avec un langage HDL (Hardware Description Language).

Exemple:

```
library ieee;
use ieee.std_logic_1164.all;

entity E is
port (
    I1:in std_logic;
    I2:in std_logic;
    O:out std_logic
);
end E;
architecture rtl of E is
signal and_gate: std_logic;
begin
    and_gate <= I1 and I2;
    O <= and_gate;
end rtl;
```

On compile ensuite le code en langage HDL qui configure les circuits électroniques et on obtient un système qui peut exécuter seulement l'algorithme contenu dans le programme HDL.

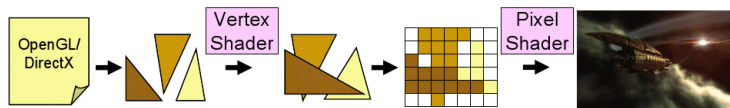
L'exécution est très rapide (pas de décodage des instructions) mais pour exécuter autre chose, il faut recompiler un autre code HDL.

Utilisé principalement pour des dispositifs très spécialisés (applications embarquées dans les domaines aéronautique, médical, audio, gps, etc)

Catégorie voisine : **ASIC** (Application-Specific Integrated Circuit), encore plus rapides, mais non programmables, les circuits qui exécutent l'algorithme choisi, sont fixés une fois pour toute.

# Utilisation des GPU

Les GPU (**G**raphics **P**rocessing **U**nits ou cartes graphique) ont été conçus pour faire le plus rapidement possible les calculs nécessaires à l'affichage : affichage pixels couleurs, tracé de formes, projection 3D sur l'écran, etc.



Pour programmer les traitements graphiques par les GPU, plusieurs interfaces spécialisées existent (OpenGL, DirectX, Metal, ...).

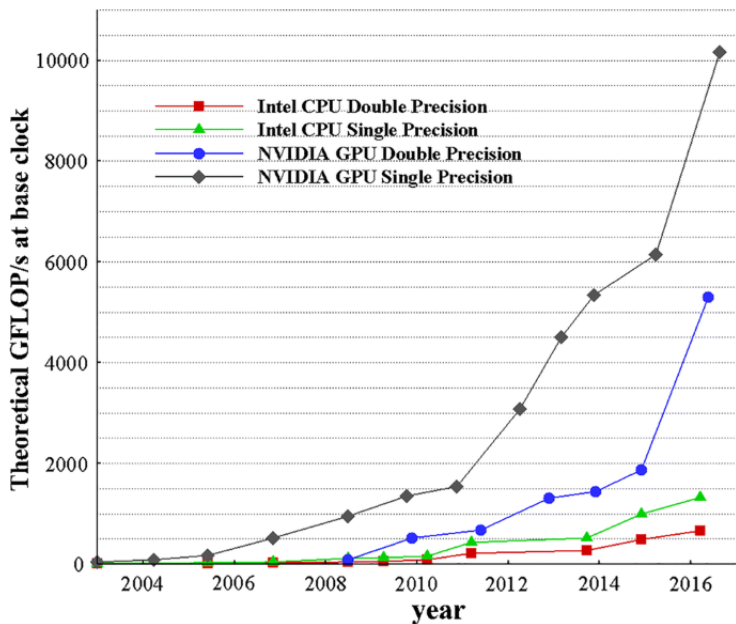
Ces traitements sont souvent très parallélisables et donc les constructeurs de cartes graphiques y incluent beaucoup d'unités de calcul (ou cœurs) (plusieurs milliers actuellement).

Principales différence entre les CPU et les GPU (situation actuelle) :

	CPU	GPU
nombre de cœurs	☹ ~ 10 à 30	☺ ~ 1000 à 10000
chaque cœur	☺ + performant	☹ – performant
jeu d'instructions	☺ plus général	☹ spécialisé calcul
bande passante	☹ – importante	☺ + importante
latence	☺ + petite	☹ + grande

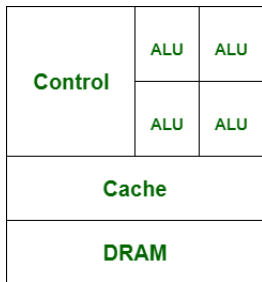
Ici, latence : durée de l'initialisation des transferts mémoire - processeur, bande passante : vitesse des transferts mémoire - processeur.

La puissance de calcul des GPU dépassent celle des CPU : les cœurs GPU sont moins puissants que les cœurs CPU mais (beaucoup) plus nombreux.

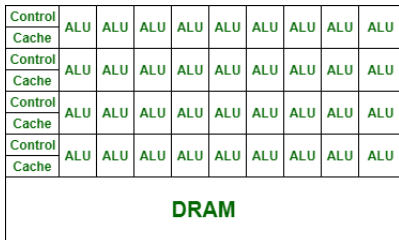


### Structure interne comparée (schématique):

- ▶ ALU : cœur
- ▶ Control : contrôleur mémoire
- ▶ Cache : mémoire cache
- ▶ DRAM : mémoire de travail



CPU



GPU



Sur une seule machine, pour :

calculs séquentiels ou faiblement // : utiliser le CPU

calculs fortement // : utiliser le GPU

A adapter à chaque cas !!

Les résultats ne seront pas toujours exactement les mêmes sur CPU et sur GPU:

- ▶ Jusque  $\sim 2000$ , les GPU (Nvidia) calculaient seulement en simple précision (32 bits) : pas besoin de 15 décimales pour calculer la couleur de pixels
- ▶ Depuis, on a du 64 bits (double précision) sur GPU, **mais** pas toujours les mêmes précisions entre le GPU et le CPU pour les additions/multiplications/divisions/..., les modes d'arrondi ne sont pas toujours identiques