ECOLE IN2P3 : Techniques de base des acquisitions multi-détecteurs

l'électronique du détecteur à la mesure
III – Composants et algorithmes

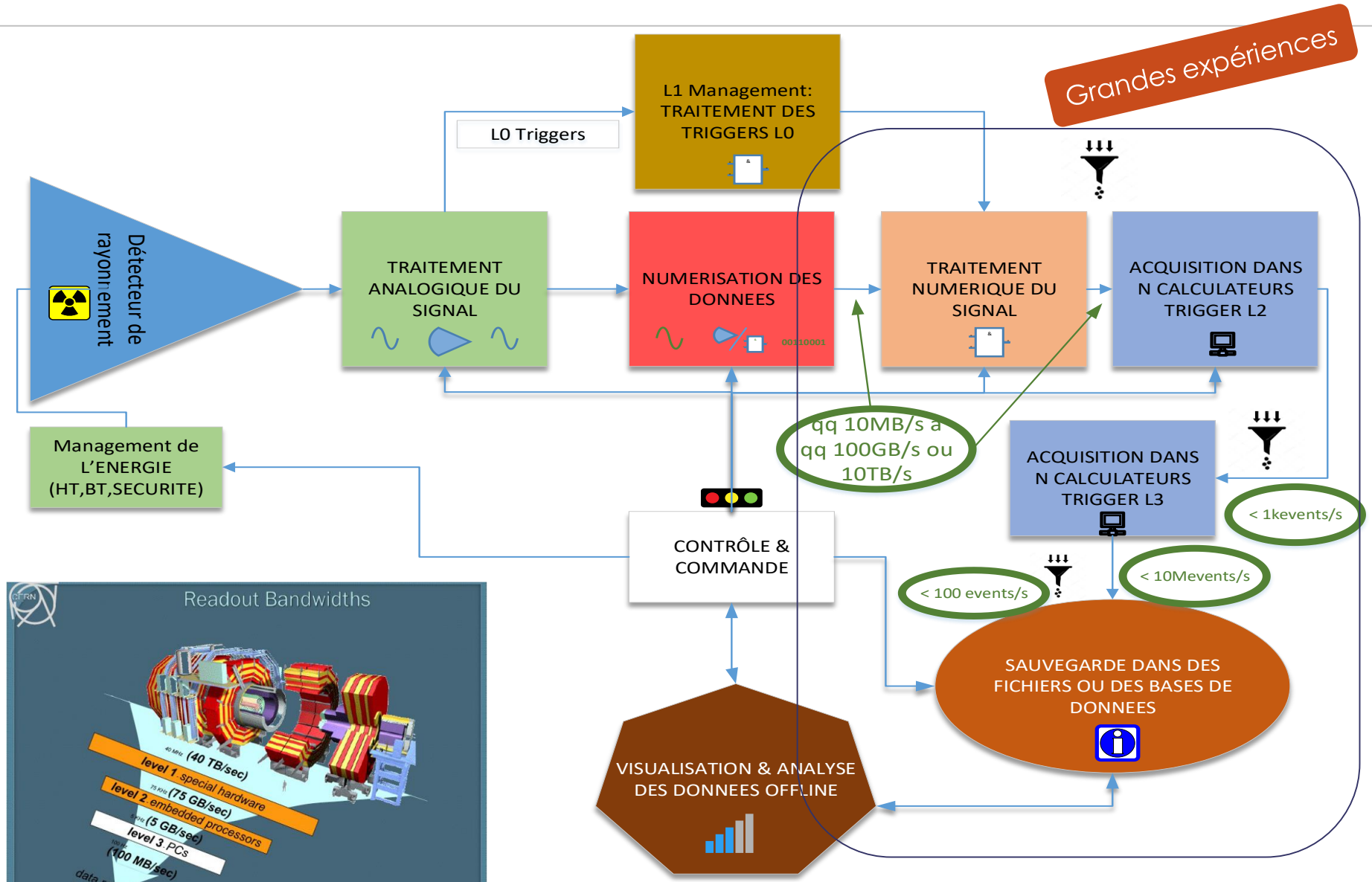
LP2i Laboratoire de Physique des 2 infinis Bordeaux

Frédéric Druillole – LP2IB -

Sommaire

- La bataille de l'informatique
- L'architecture numérique dans nos instruments
  - Logique programmable
  - Processeurs
  - Synchronisation
- Le processeur
  - ALU et fonctionnalité
  - Le problème de la mémoire
  - Les technologies de l'information
- Et apres…
- FPGA
- Démonstration:
  - Système DAQ 1 voie

Un Système d'acquisition actuel multicanaux



La bataille de l'informatique

SCIENCES EXPERIMENTALES | SCIENCES THEORIQUES | SCIENCES CALCULATOIRES

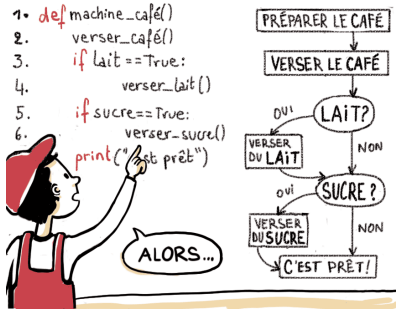
1600 1950

Informatique : traitement automatisé de l'information

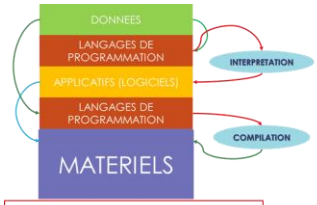
Représentation de l'information

Algorithmes

- Symbolisme
- connectivisme
- Probabilisme
- Analogisme



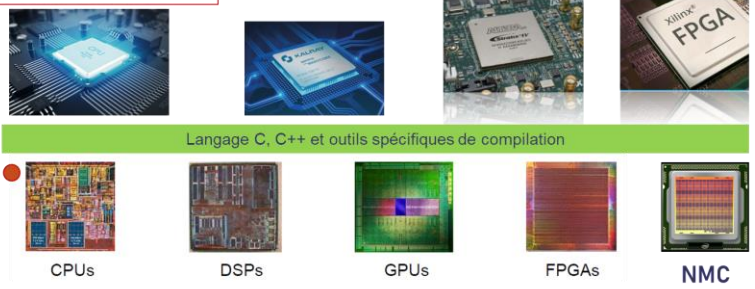
Matériels (Microprocesseurs)
Architecture de Von Neumann



Ordinateur actuel
Central Processing Unit

Ordinateur quantique
Langage spécifique

Calcul haute performance



STRUCTURES DE DONNEES

Fichiers
Base de données

- ☺ Table de hachage
- ☺ Listes
- ☺ Listes chainées
- ☺ Vecteurs
- ☺ Pile / Tas
- ☺ Tableau N dimension (Tenseur)

➔ Force brute
➔ Prog. Dynamique
…

➔ Diviser et régner
➔ Glouton
…

La bataille de l'informatique

SCIENCES EXPERIMENTALES → SCIENCES THEORIQUES → SCIENCES CALCULATOIRES → **SCIENCES PILOTEES PAR LES DONNEES**

1600 — 1950 — 2000

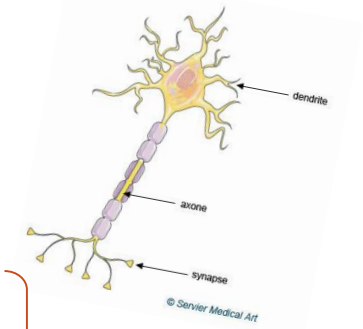
Approche inférentielle (Inductive)

Données d'entrée
Données de sortie


Modèle d'inférence

MESURES
FAITS
Informations

Lois
Règles



- connectivisme
- \+ Probabilisme
- \+ Analogisme

Algorithmes apprenants

vs

Approche déductive

Données d'entrée
ALGORITHMES

Données de sortie

EXPERTS:
LOIS
Règles

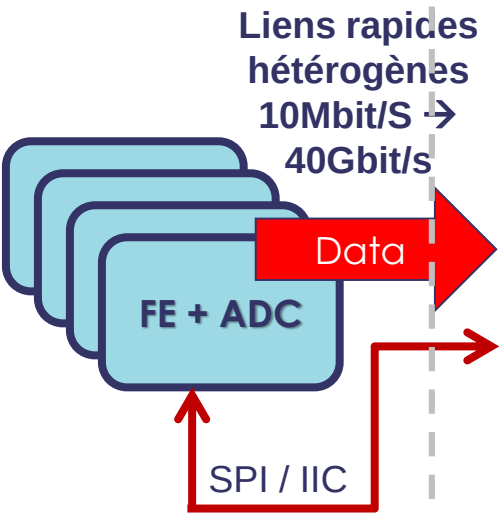
Traitement de cas particuliers:
Logiciels spécialisés

Symbolisme

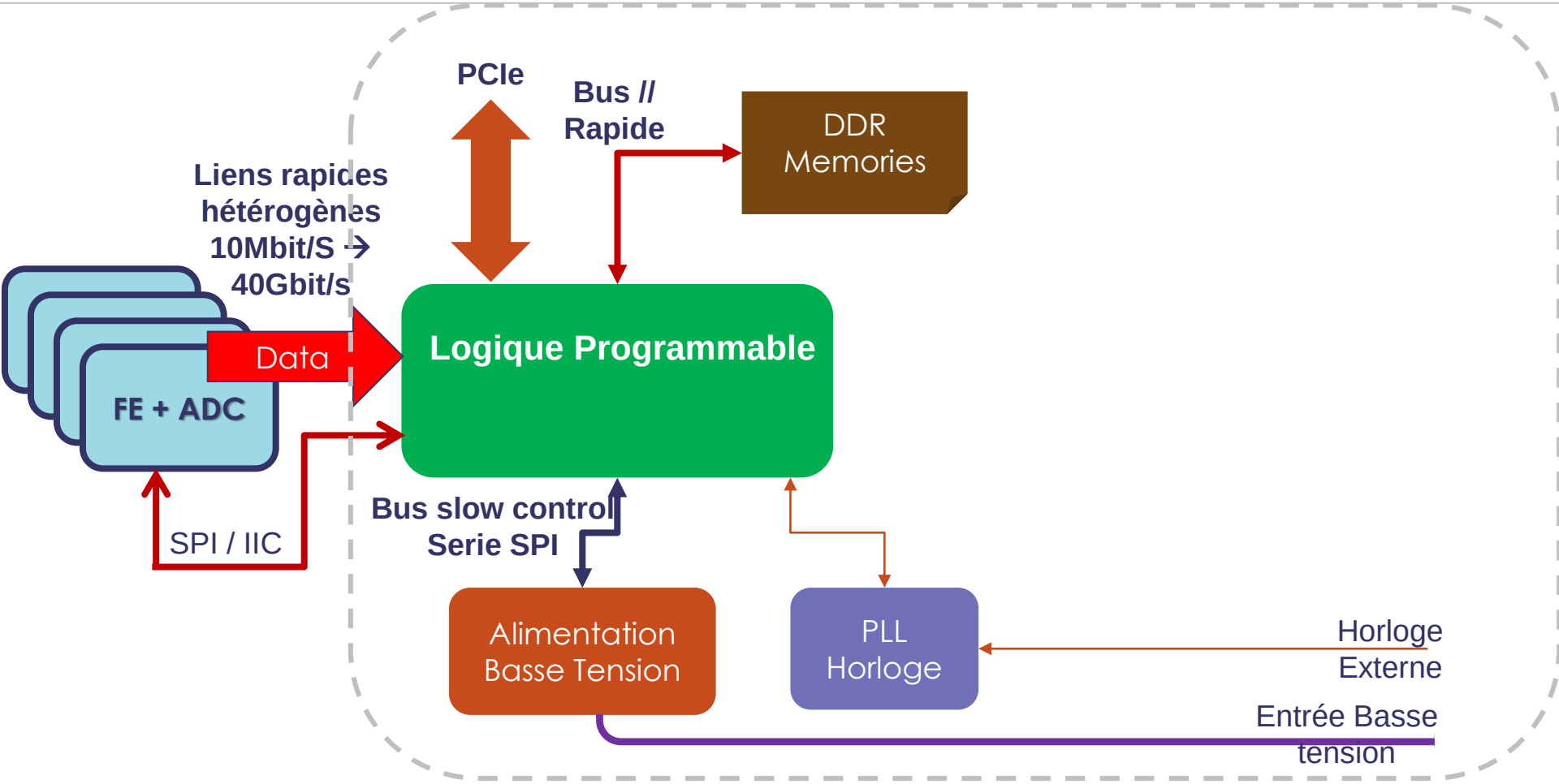
Boucles
Tests logiques

```
 if (strncmp(cmd, "ADS:SPI:REGLIST", 15) == 0) {
if (Verbose > 0) {
xil_printf("**************************************\r\n");
xil_printf("* User R2D2 OWEN SPI LIST REGISTER   *\r\n");
xil_printf("**************************************\r\n");
}
if (sscanf(cmd, "ADS:SPI:REGLIST %s", str_array) != 1) {
sprintf(resp, " ERROR: ADS:SPI:REGLIST command failed !! \r\n");
er = -6;
return XST_FAILURE;
} else {
spiwordvalue = (u16_t*)realloc(spiwordvalue, 16*sizeof(u16_t));
strcpy(cp_array, str_array);
nb_parameter = sscanTabInt(cp_array, spiwordvalue);
xil_printf("spidata: nombre de parametres : %d\n", nb_parameter);
xil_printf("spi reg =[ ");
            for (int i= 0; i < nb_parameter; i++) {
            xil_printf("%x, ", spiwordvalue[i]);
            }
            xil_printf(" ]\r\n");
            sprintf(resp, "spi data list: : %s\r\n", str_array);
}
```

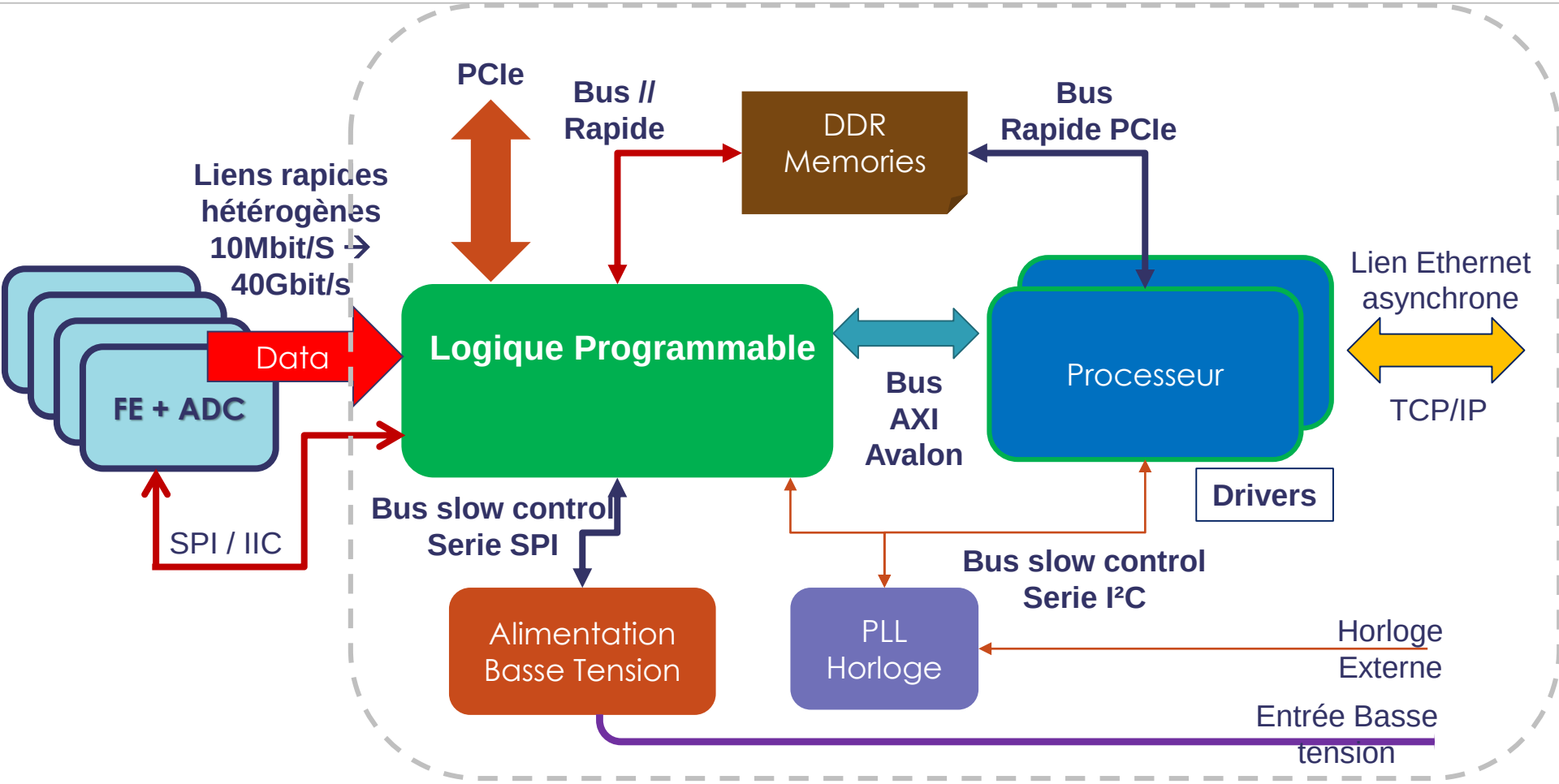
Architecture numérique



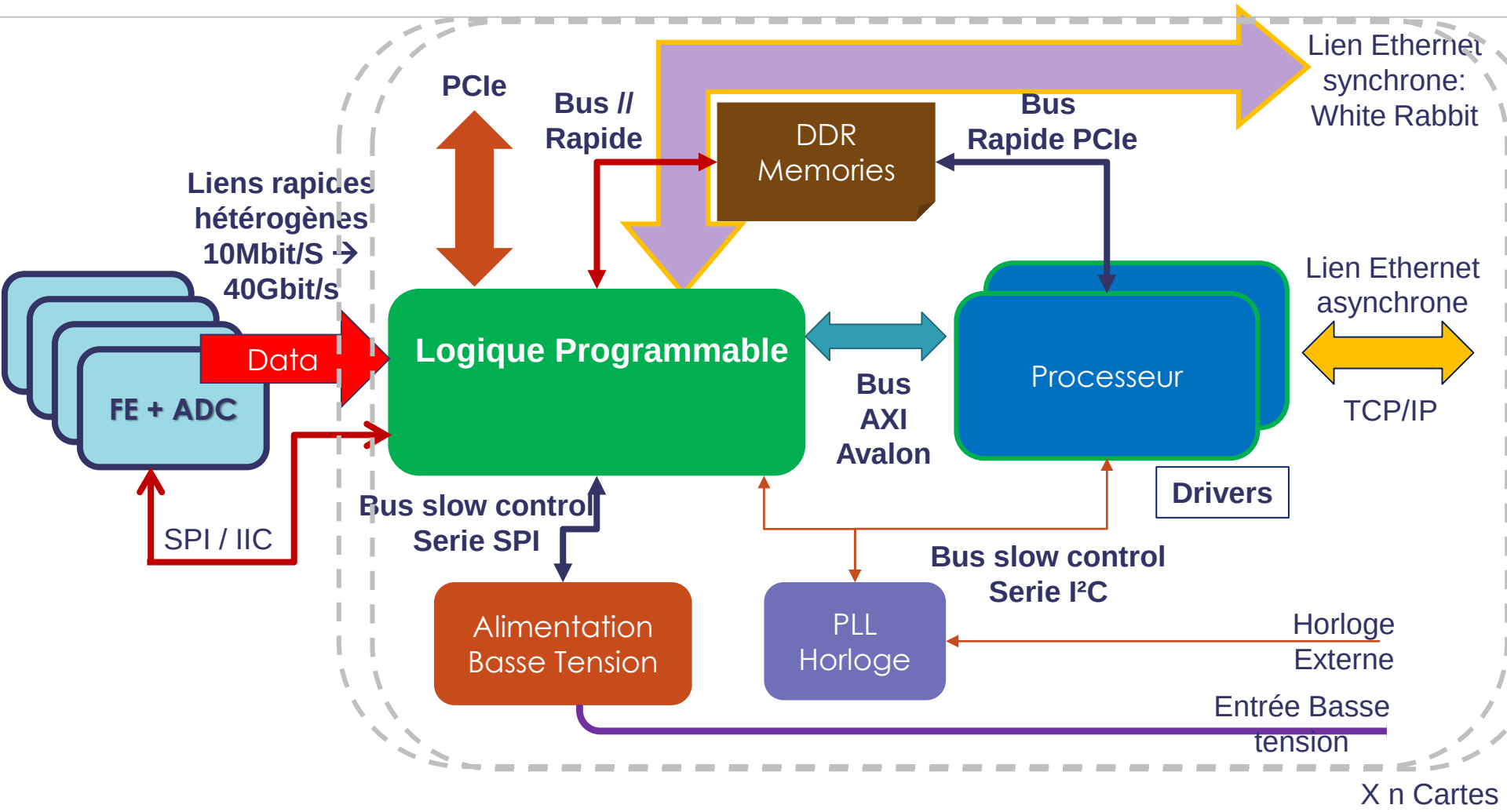
Architecture numérique



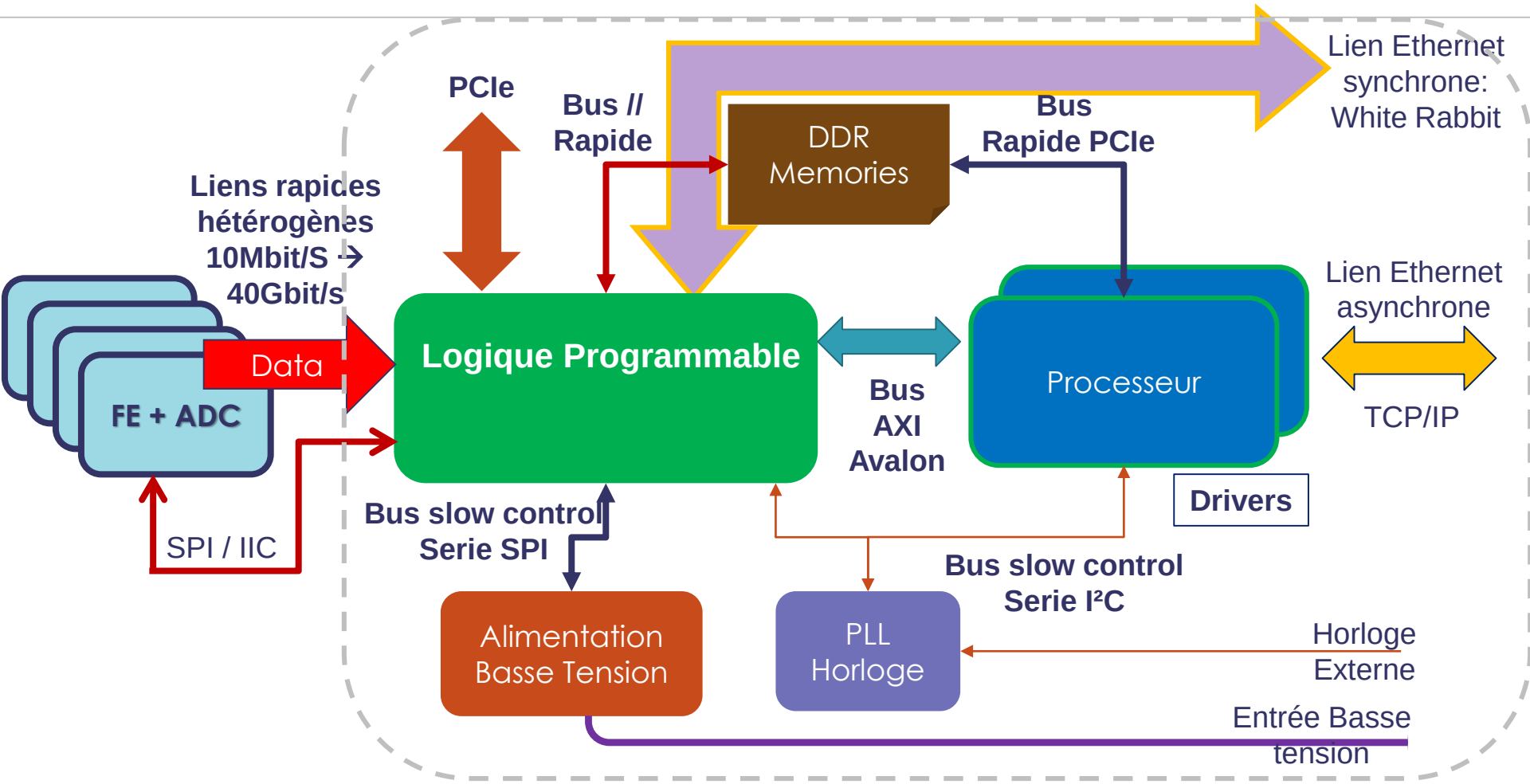
Architecture numérique



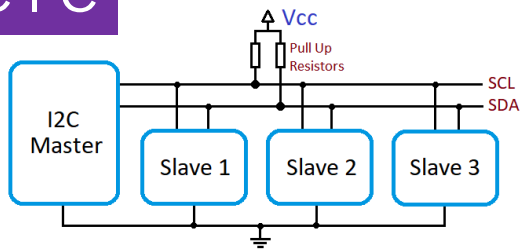
Architecture numérique

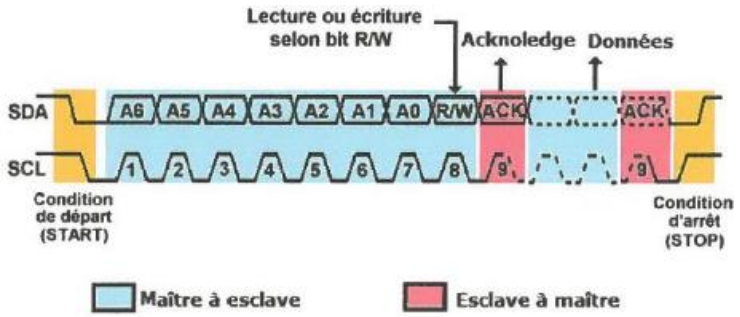


Architecture numérique

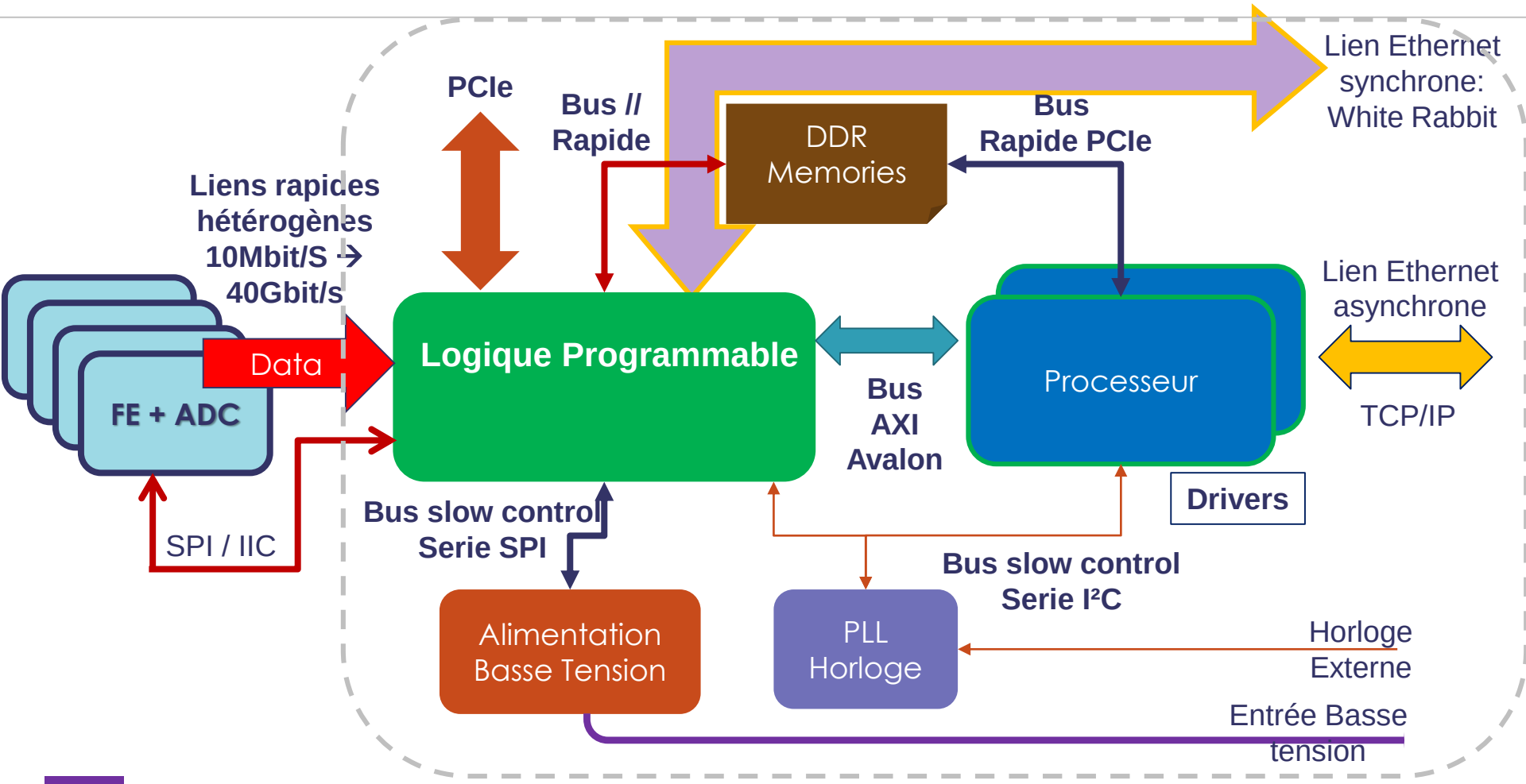


IIC I²C

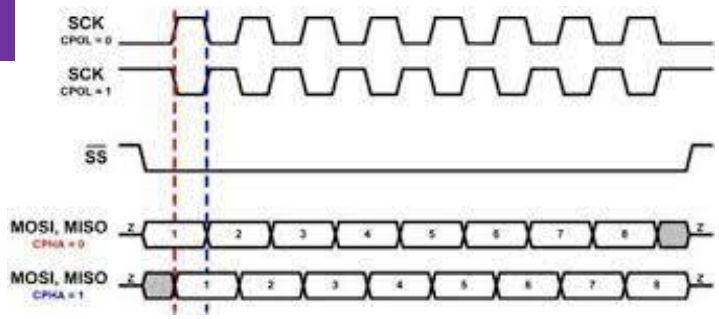


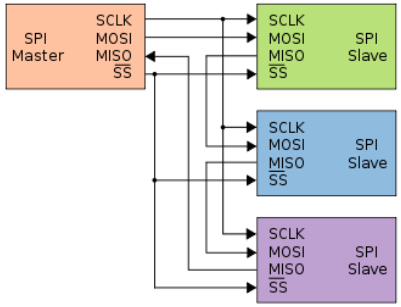


Architecture numérique

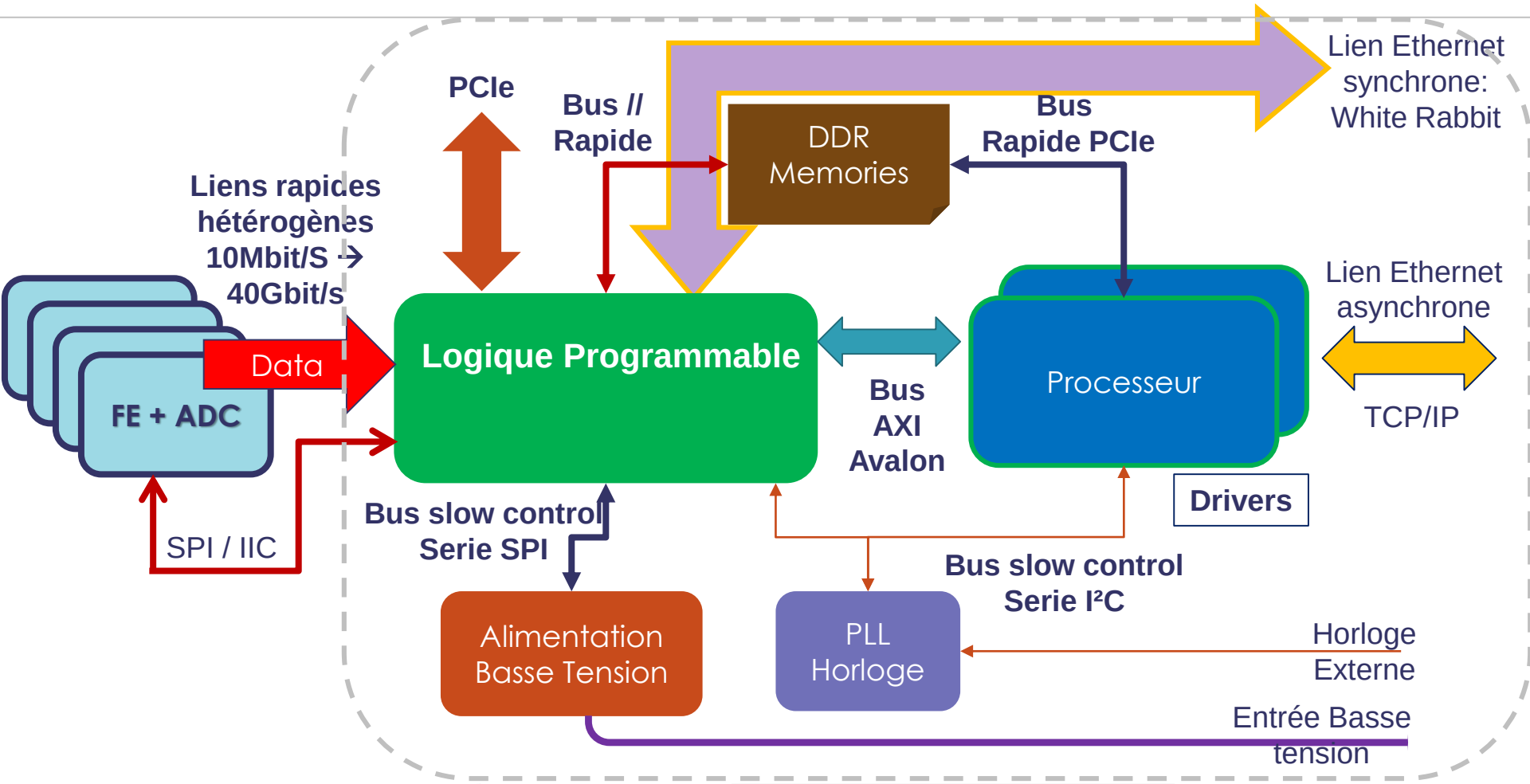


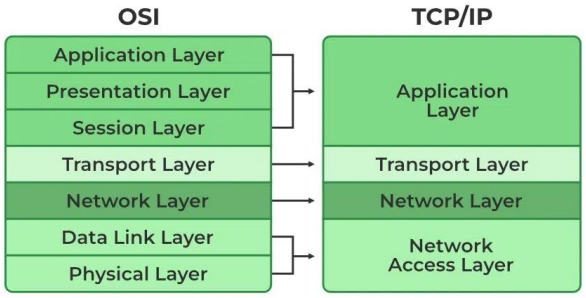
SPI

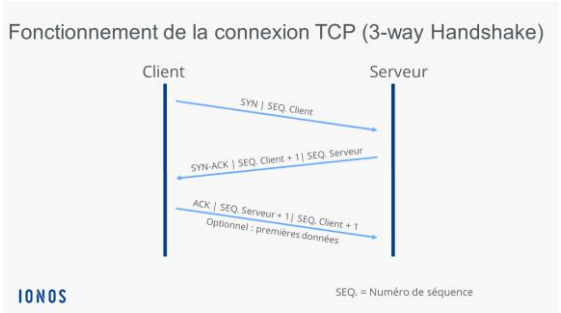




Architecture numérique







Architecture numérique

PCIe

Bus // Rapide

DDR Memories

Bus Rapide PCIe

Lien Ethernet synchrone: White Rabbit

Liens rapides hétérogènes 10Mbit/S → 40Gbit/s

FE + ADC

Data

Logique Programmable

Bus AXI Avalon

Processeur

Lien Ethernet asynchrone

TCP/IP

Drivers

SPI / IIC

Bus slow control Serie SPI

Bus slow control Serie I²C

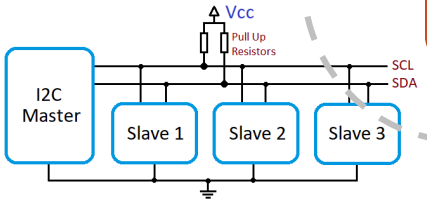
Alimentation Basse Tension

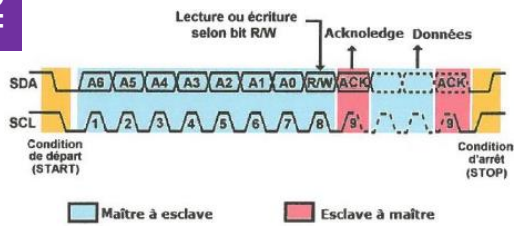
PLL Horloge

Horloge Externe

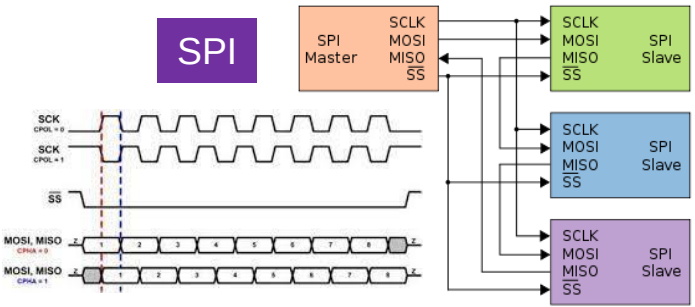
Entrée Basse tension

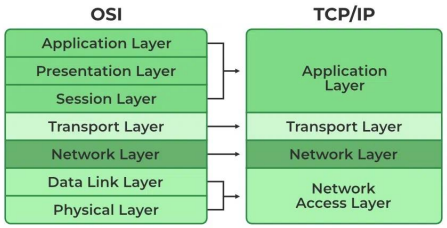
IIC I²C



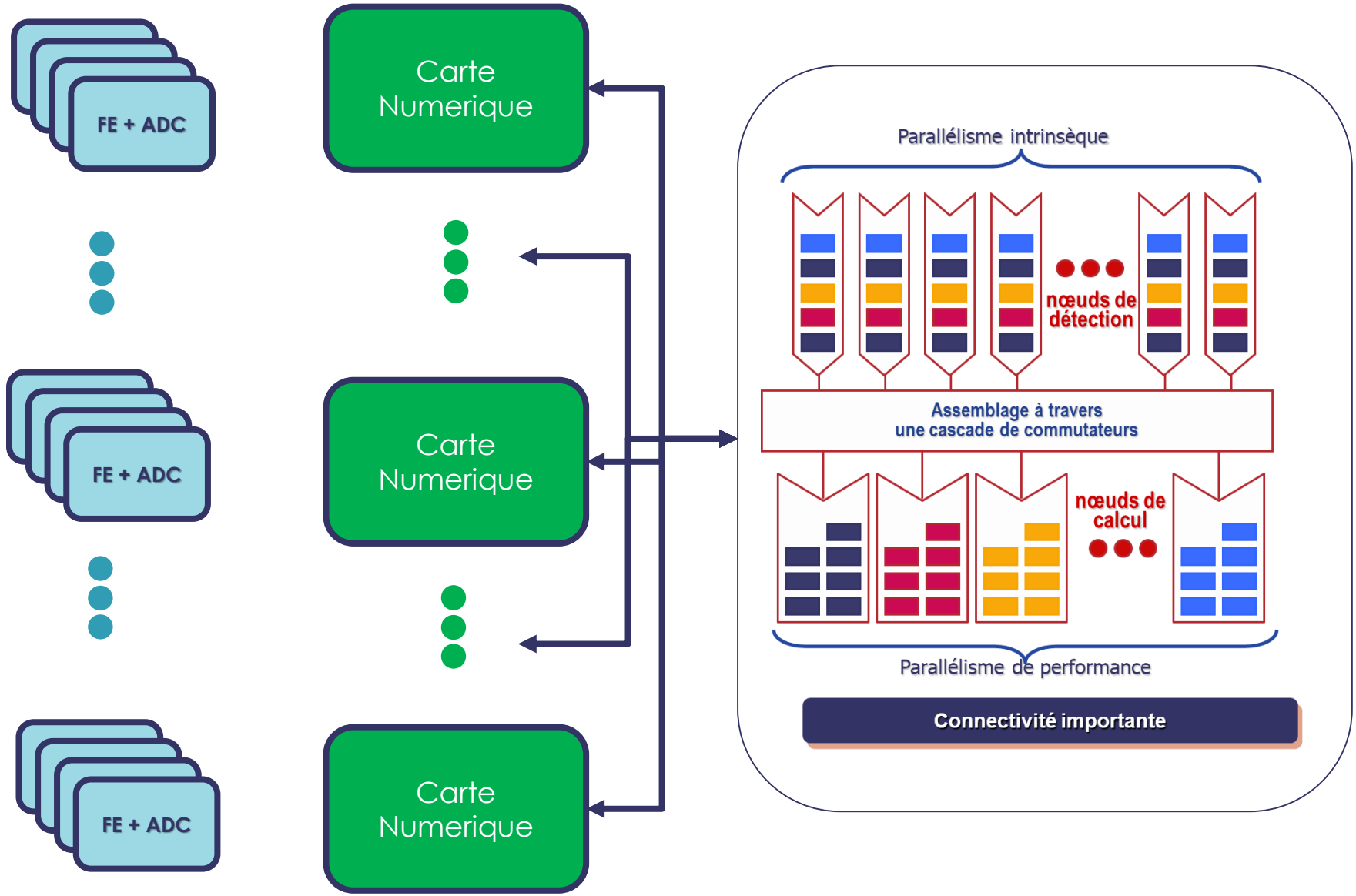


SPI

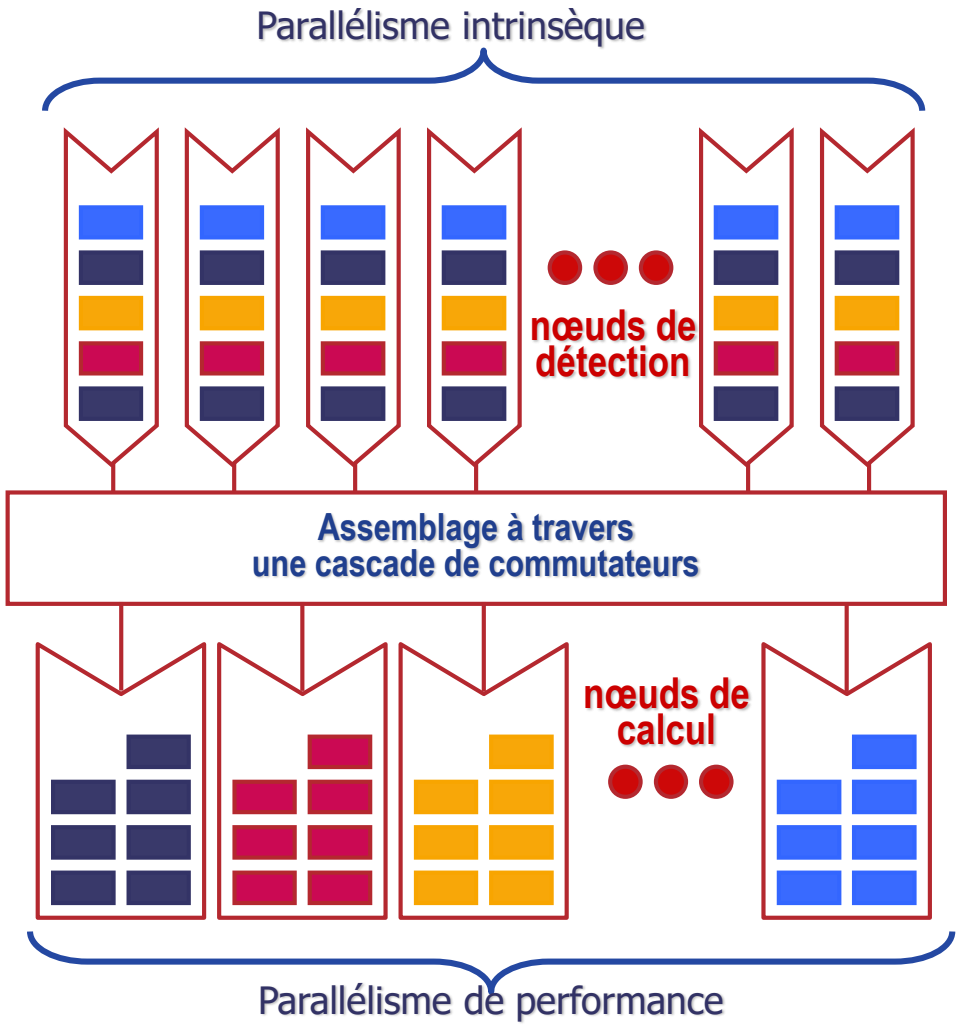




Architecture système



Comment est-ce utilisé ?



- Matériels & Firmware
- ADC
- TAS (CEB & SNR)

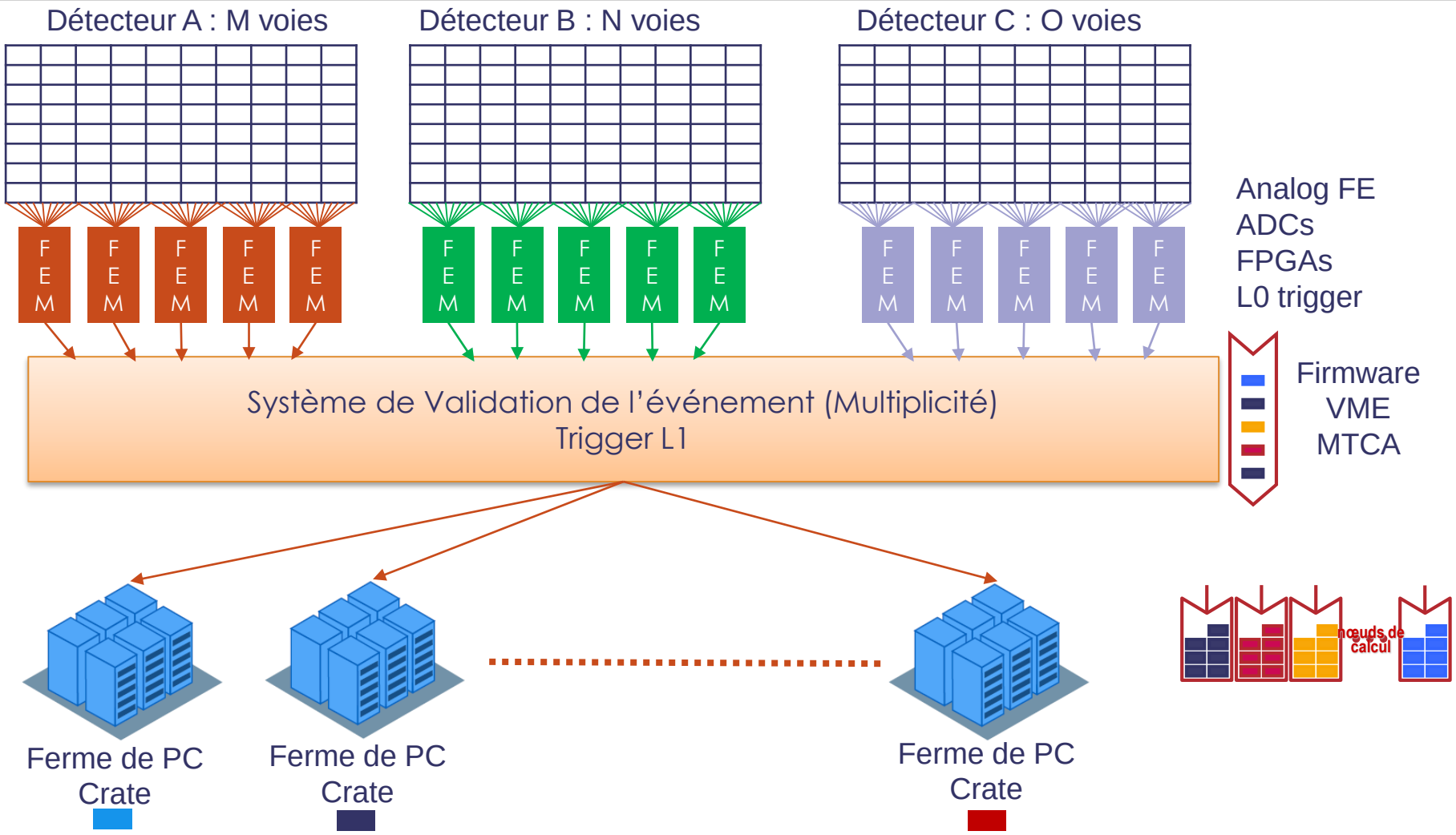
Les aspects numériques :

- Matériels
- Firmware et logiciels
- « Temps réel »
- Distribués
- Hétérogènes
- Projets longs & évolutifs

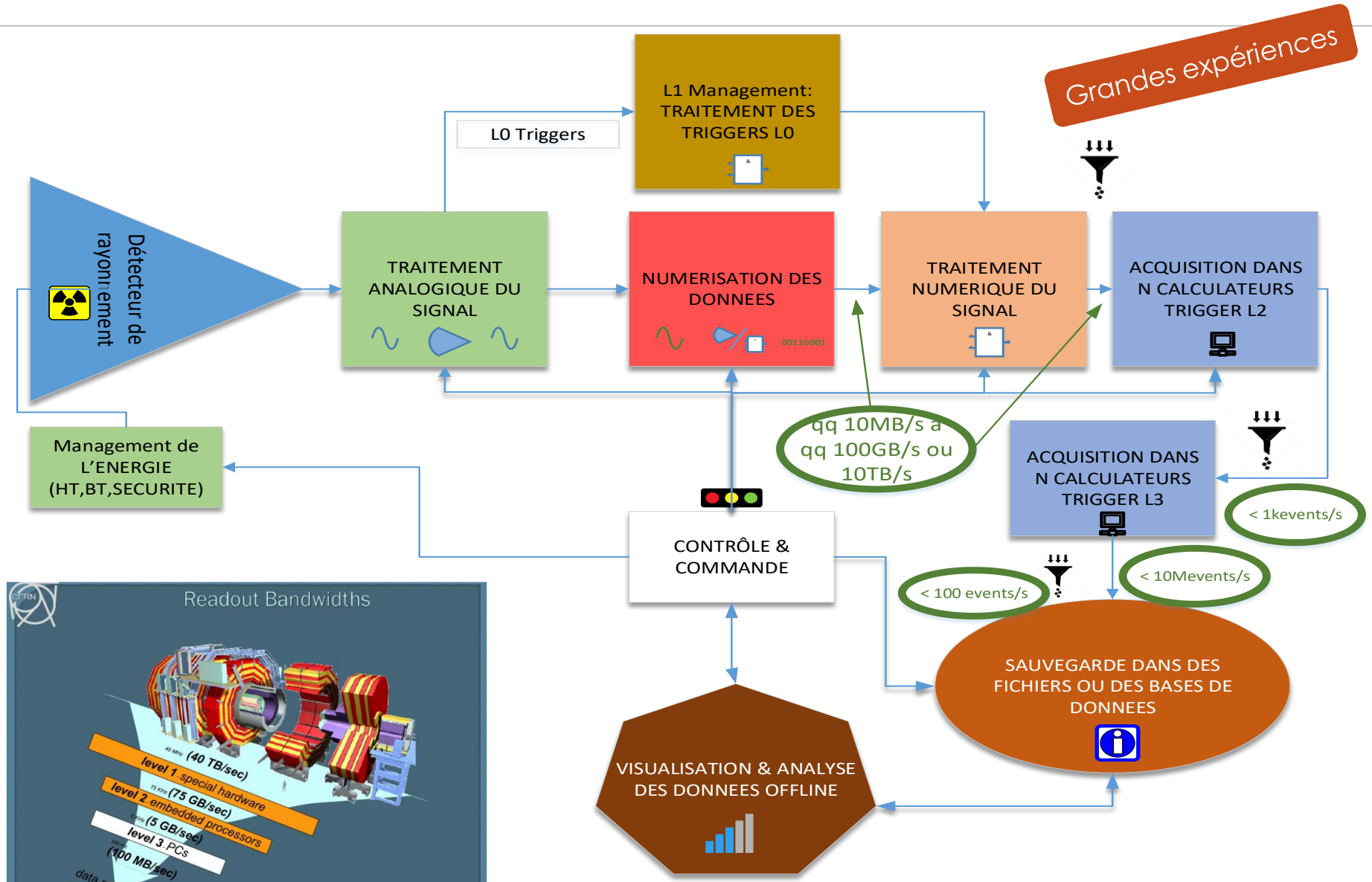
- Logiciels
- Grille de calculs (DataCenter)

Connectivité importante

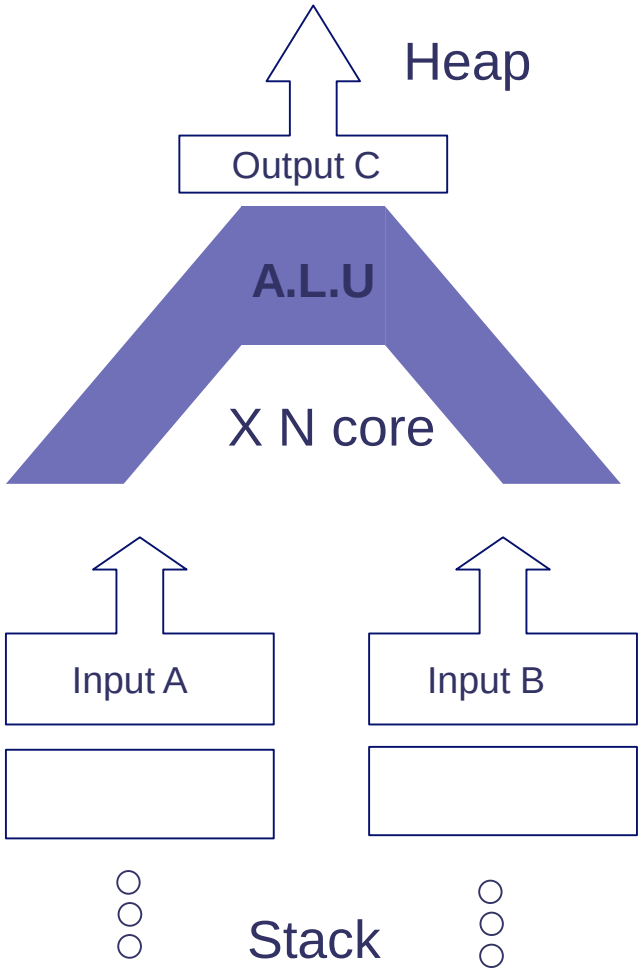
L'architecture d'acquisition de données multi-détecteurs



Un Système d'acquisition actuel multicanaux



Processeurs

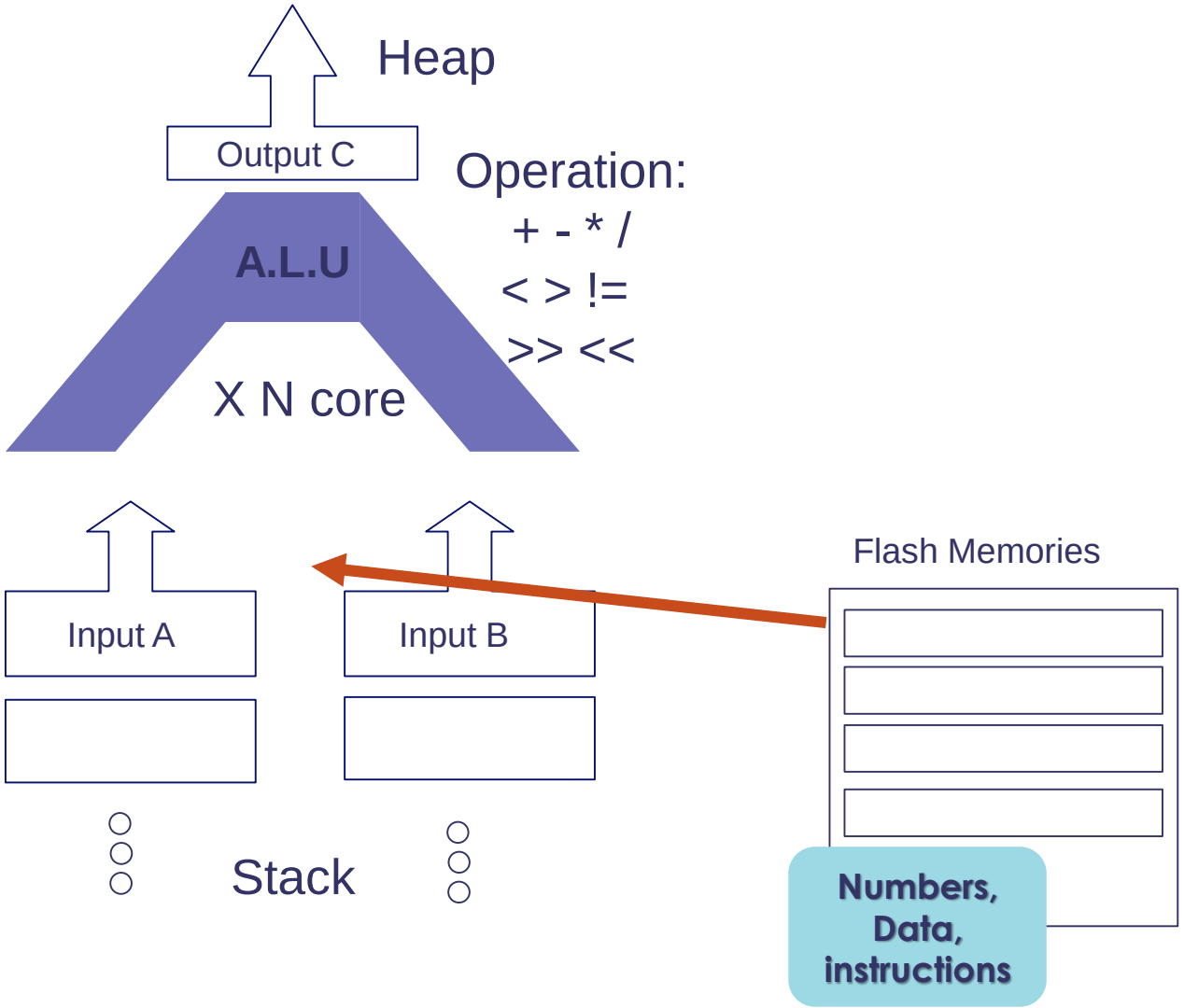


Value : Binary items (Bit)
00 : 0
01 : 1
10 : 2
11 : 3

Operations:
Arithmetic : + - * /
Test : < > !=
Bit shift : >> <<
Boolean algebra :
AND, OR, NOT

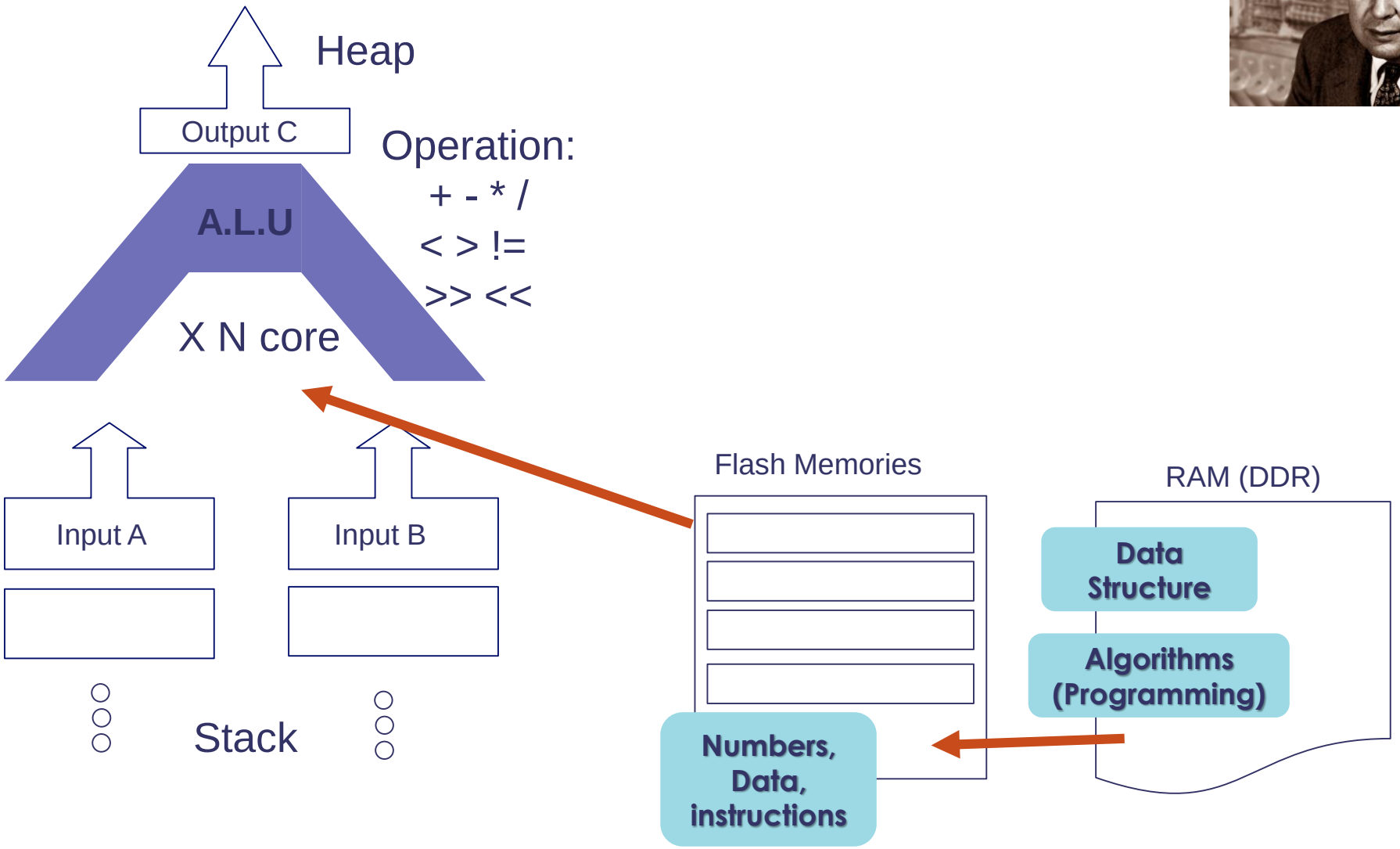
➔ One instruction in each clock period

Processeurs



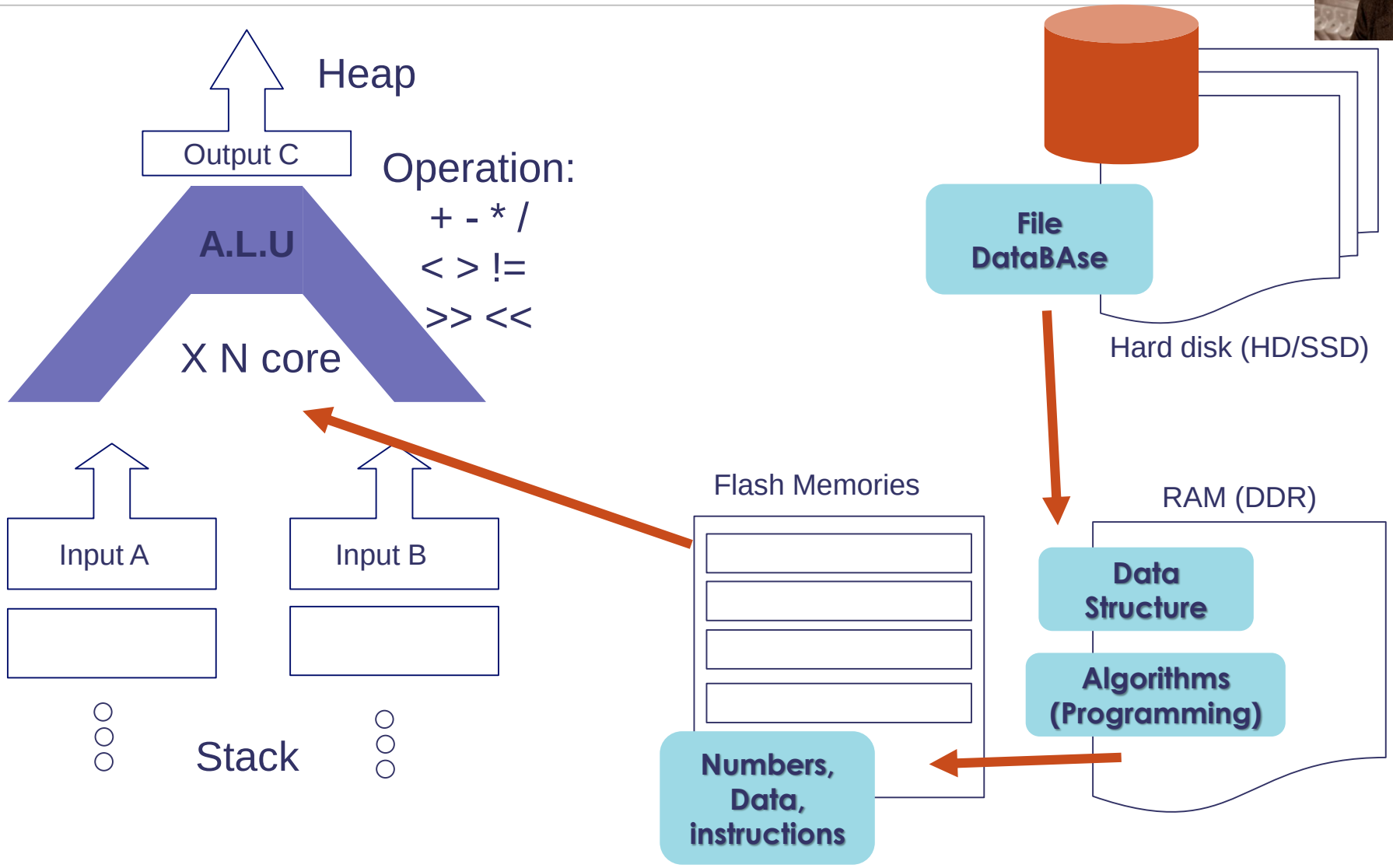
➔ One instruction in each clock period

Processeurs



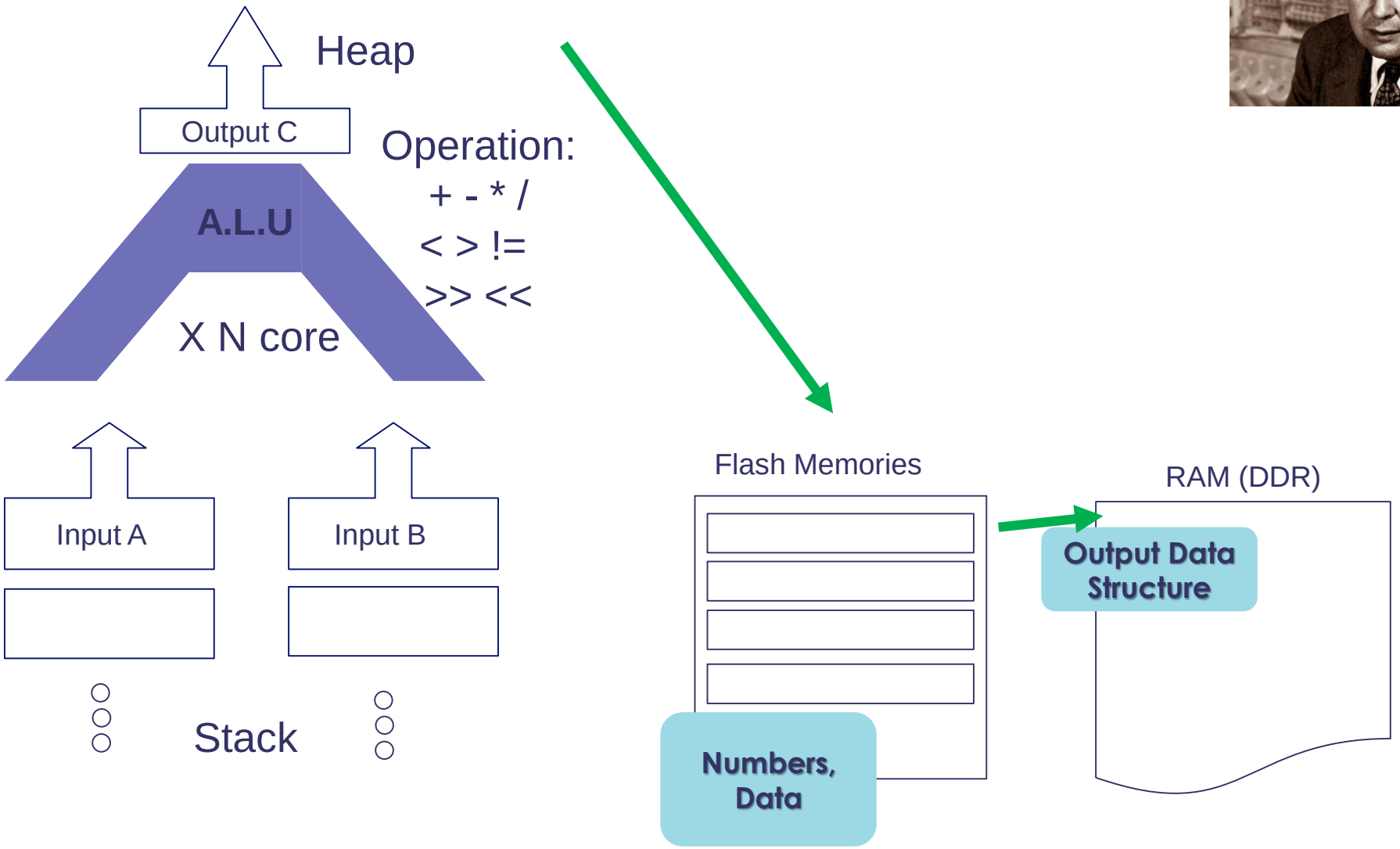
➔ One instruction in each clock period

Processeurs



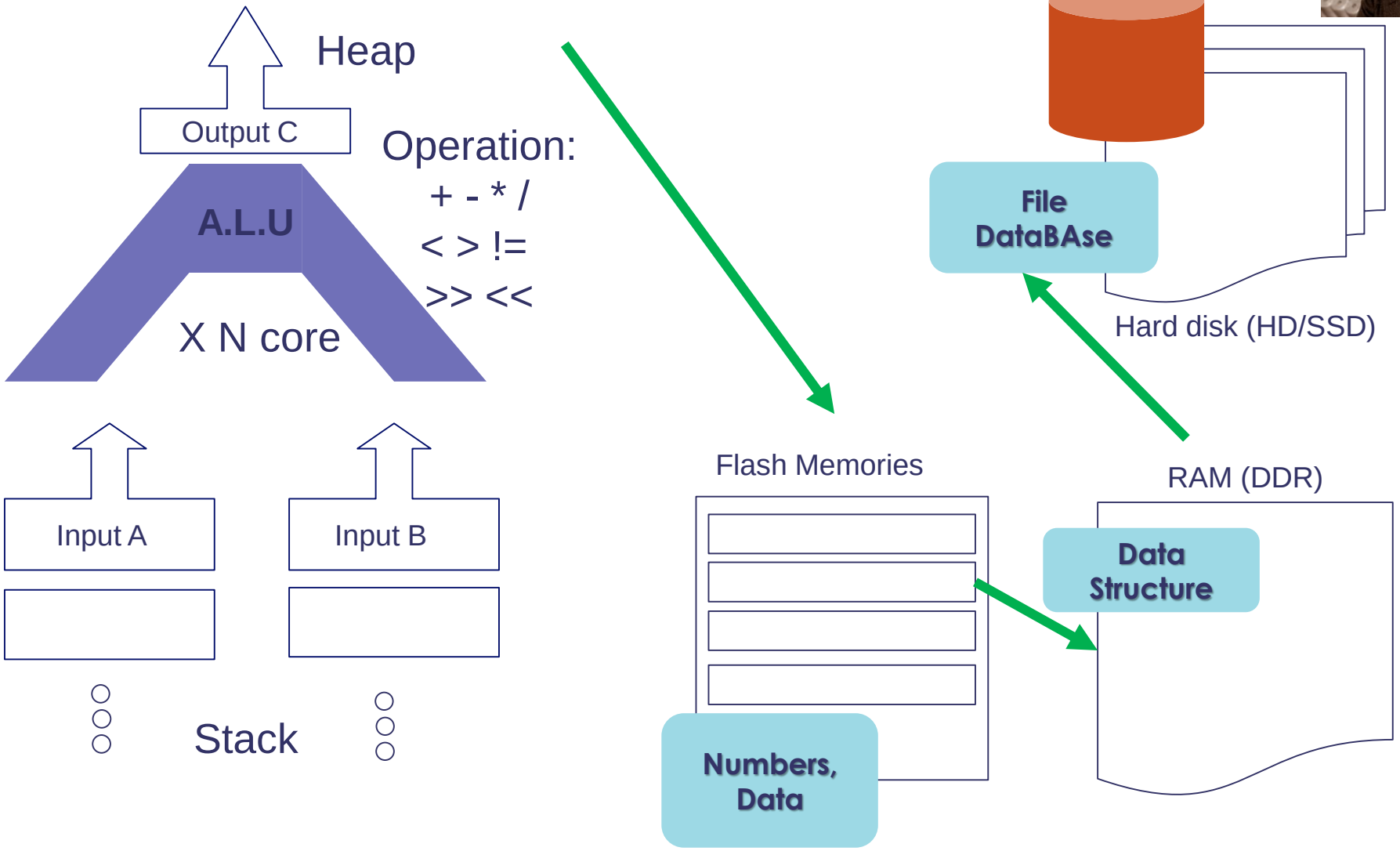
➔ One instruction in each clock period

Processeurs



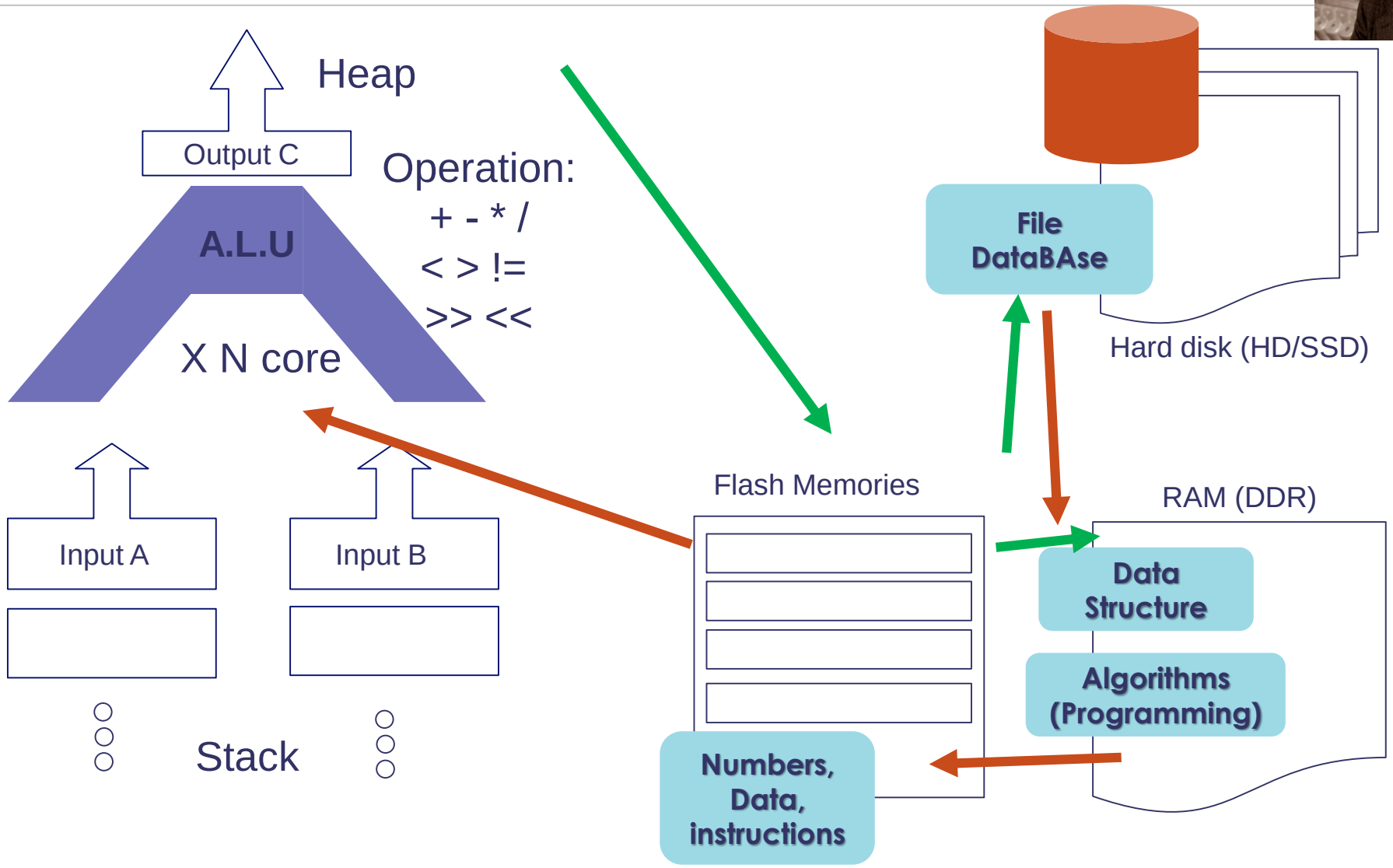
➔ One instruction in each clock period

Processeurs



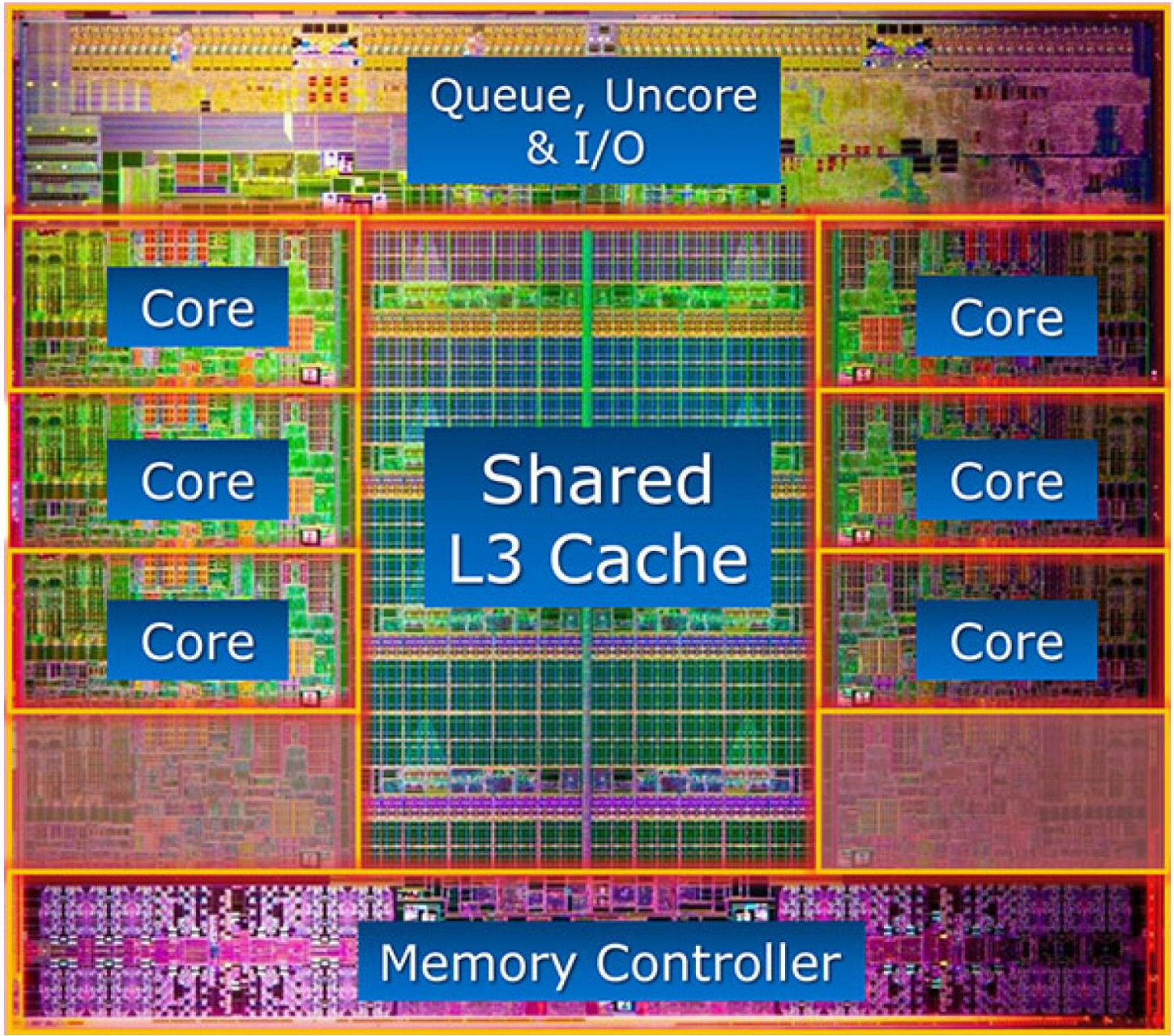
➔ One instruction in each clock period

Processeurs

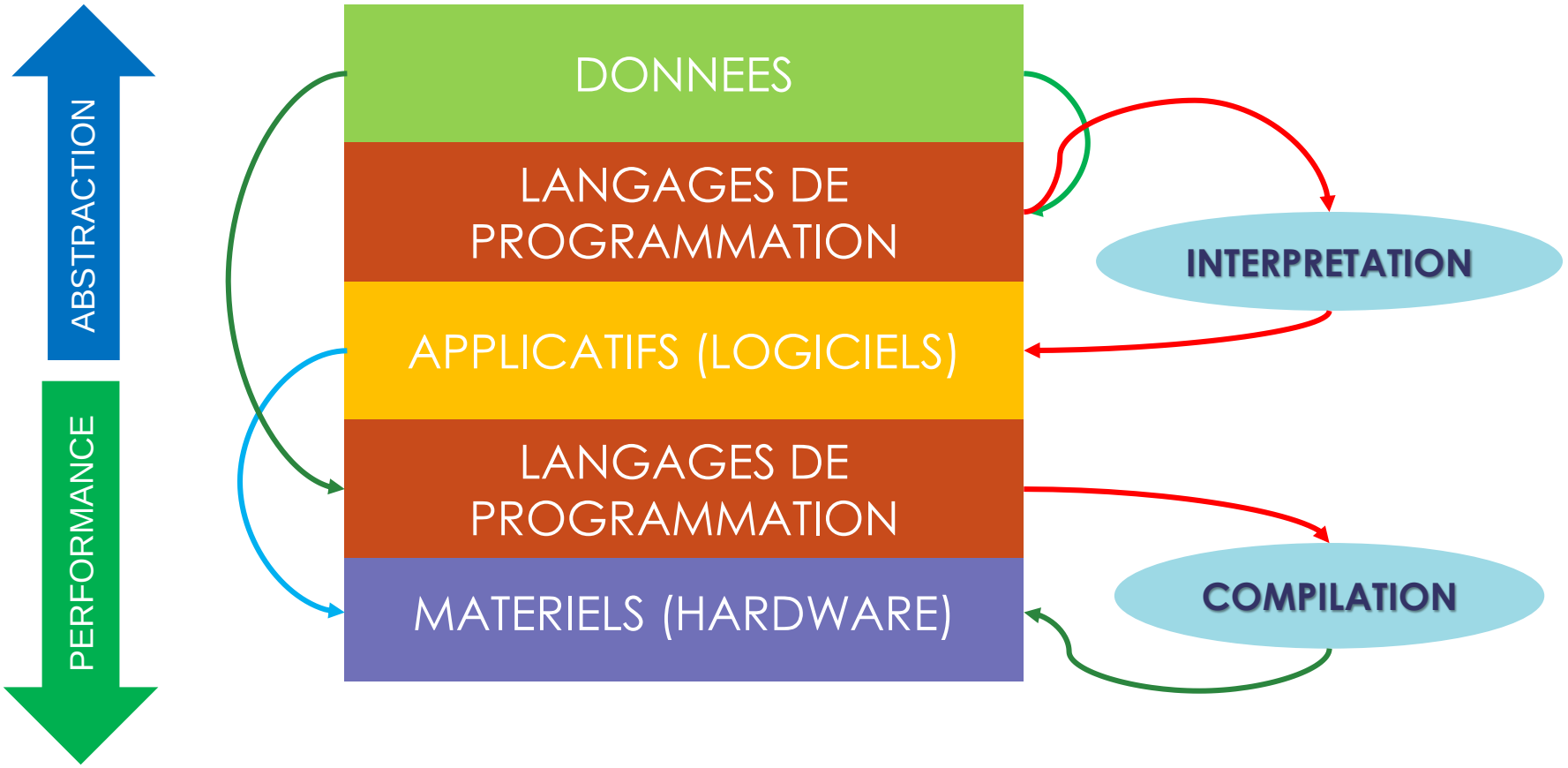


➔ One instruction in each clock period

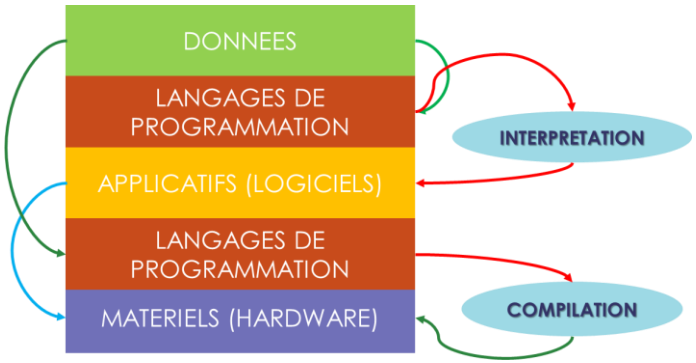
Processeurs



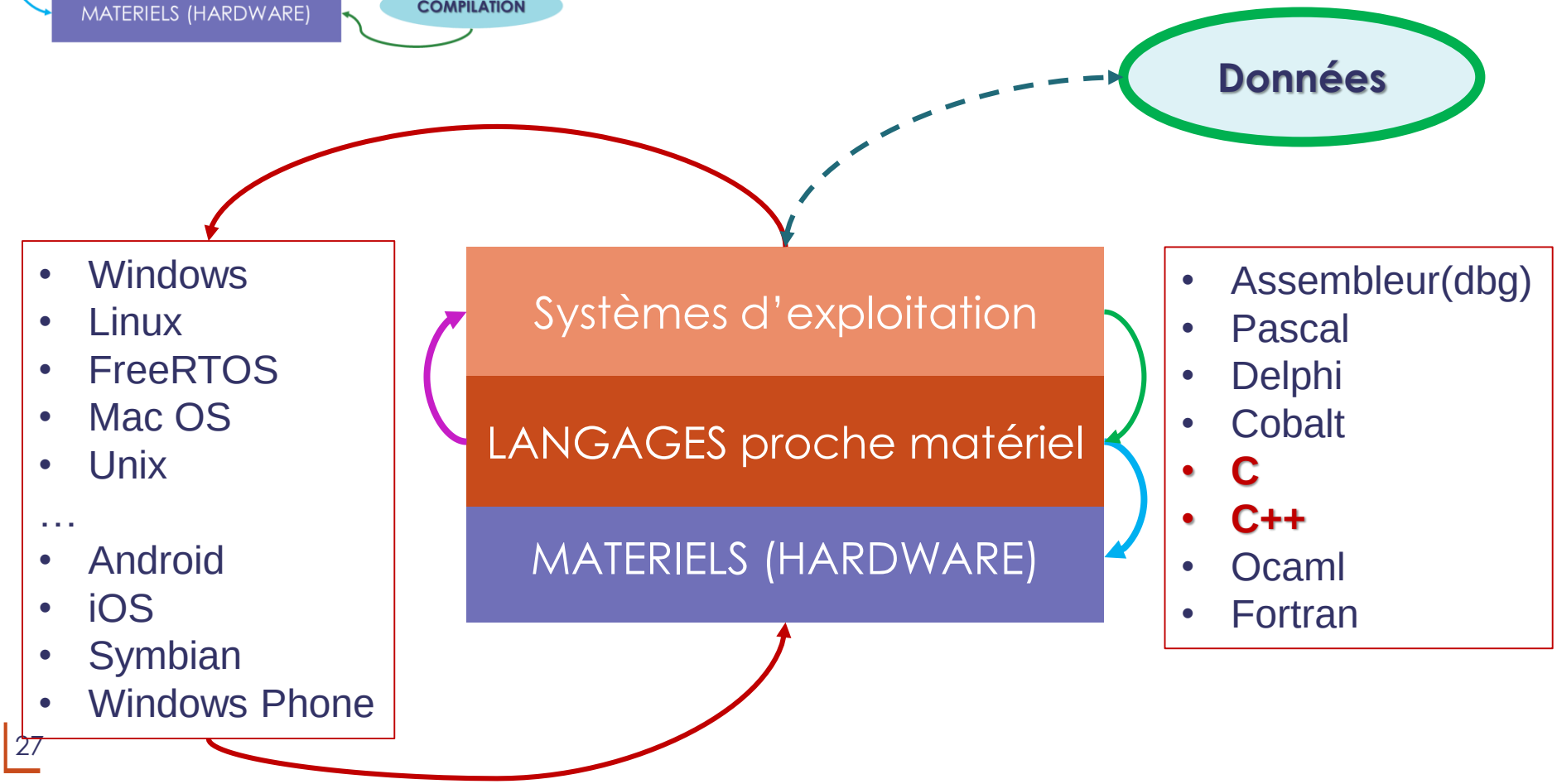
Un mot de la technologie de l'information



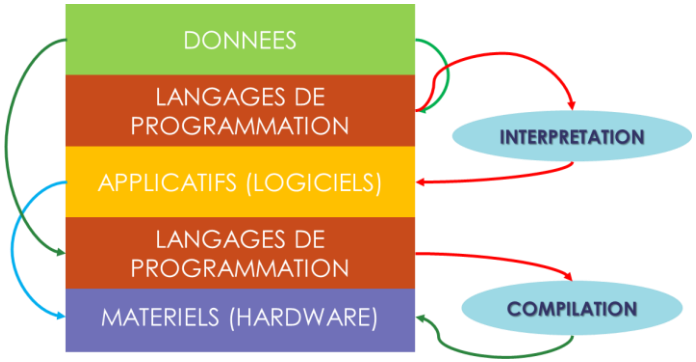
Un mot de la technologie de l'information



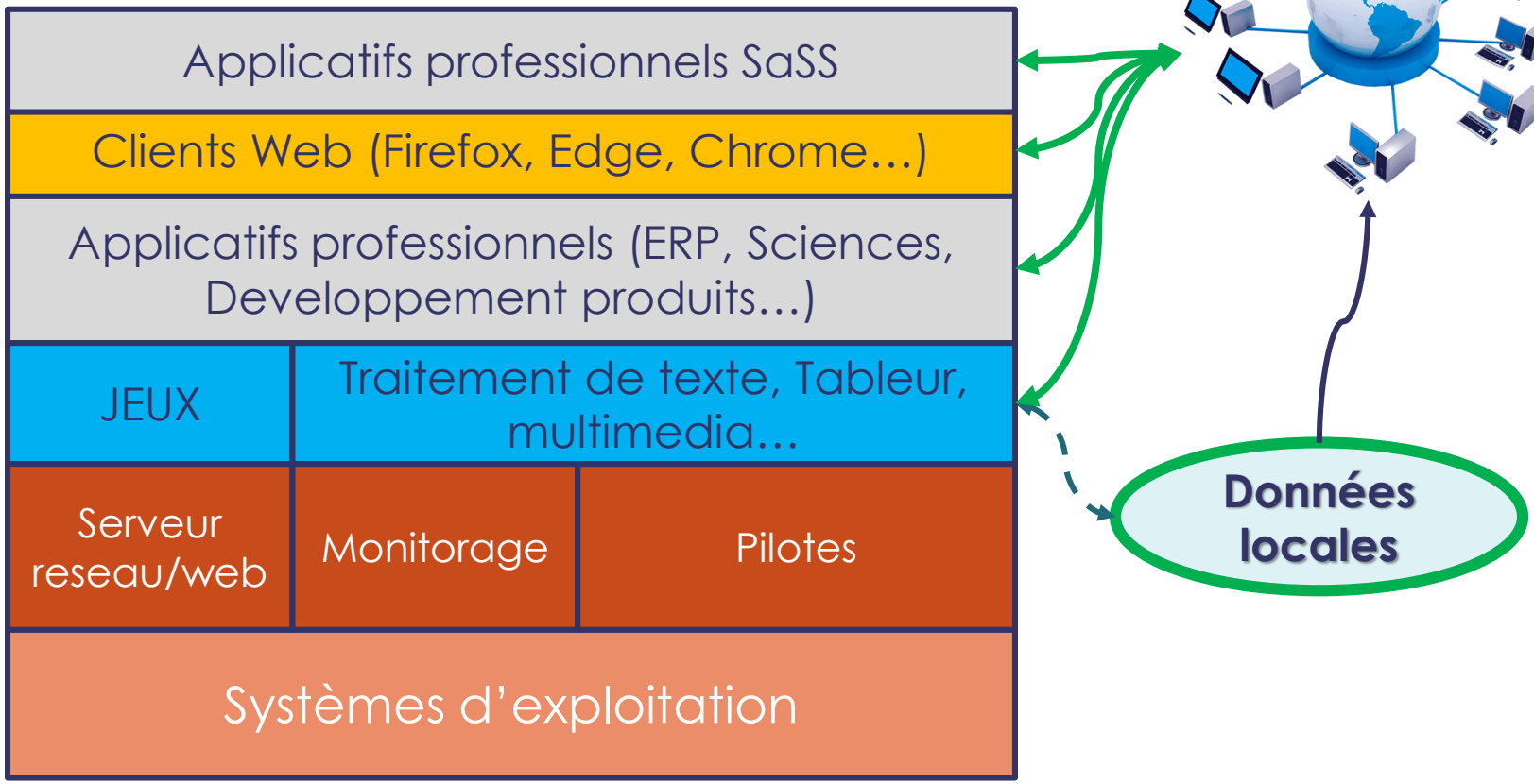
➔ Langages adressant directement tous types de matériels (CPU, µController etc…)
➔ Permet de créer des SE/OS



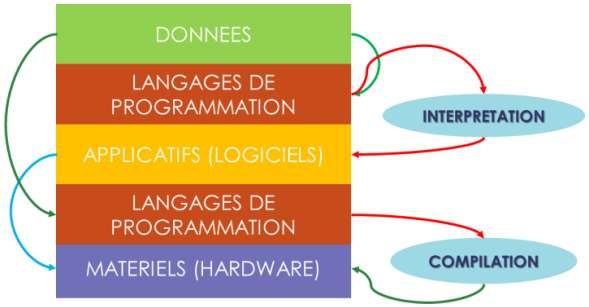
Un mot de la technologie de l'information



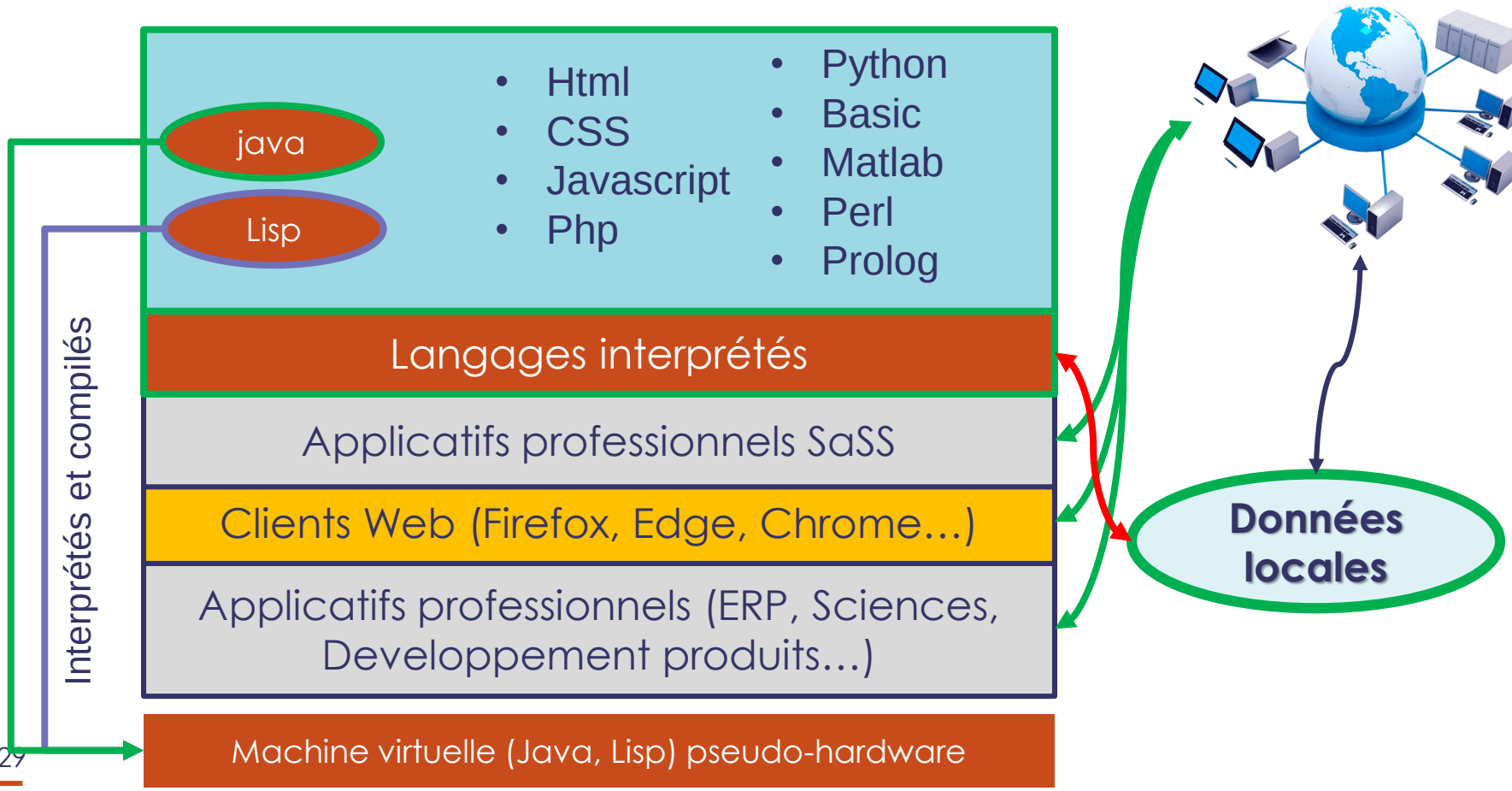
➔ Les langages permettent de créer tous type de logiciels. Ces logiciels peuvent être indépendants des données qu'ils traitent (Traitement de texte, Tableur, Pres….) ou bien aller les chercher sur le reseau…



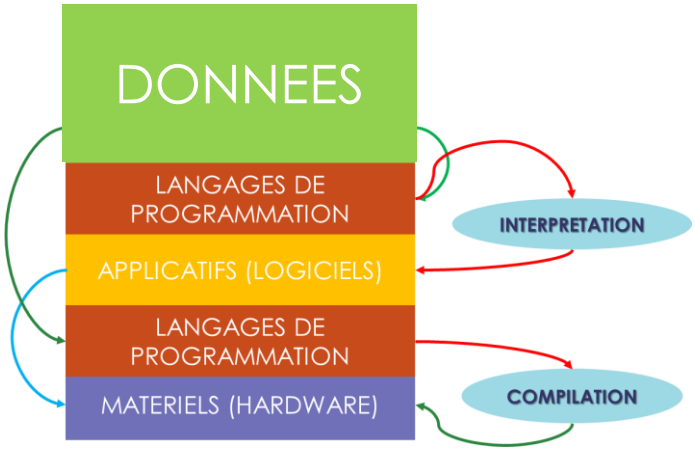
Un mot de la technologie de l'information



➔ Les langages interprétés permettent de créer des applications très liées à l'information au données. Les aspects matériels sont peu pris en compte, ce sont des langages de haut niveau (loin du matériel)



Un mot de la technologie de l'information



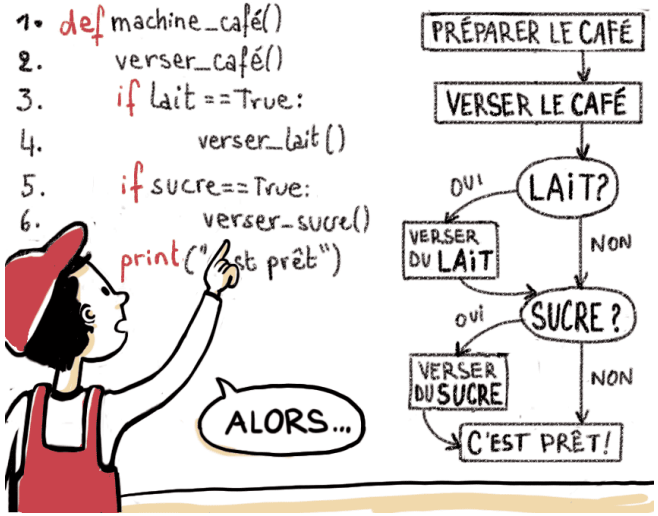
➔ Les données sont la moelle épiniere de l'informatique. Elles doivent être stockés, être préparées à l'analyse et traitées, transformées pour enrichir leur valeur.

STRUCTURES DE DONNEES

- Base de données
- Table de hachage
- Listes
- Listes chainées
- Vecteurs
- Pile / Tas

ALGORITHMIE

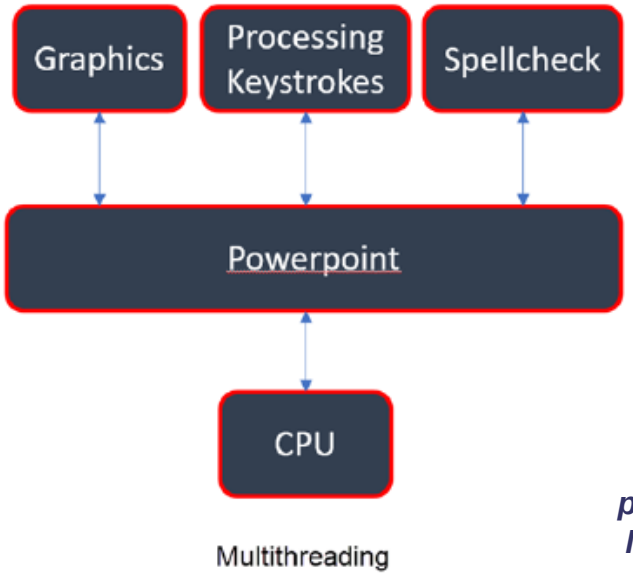
- Symbolisme
- connectivisme
- Probabilisme
- Analogisme

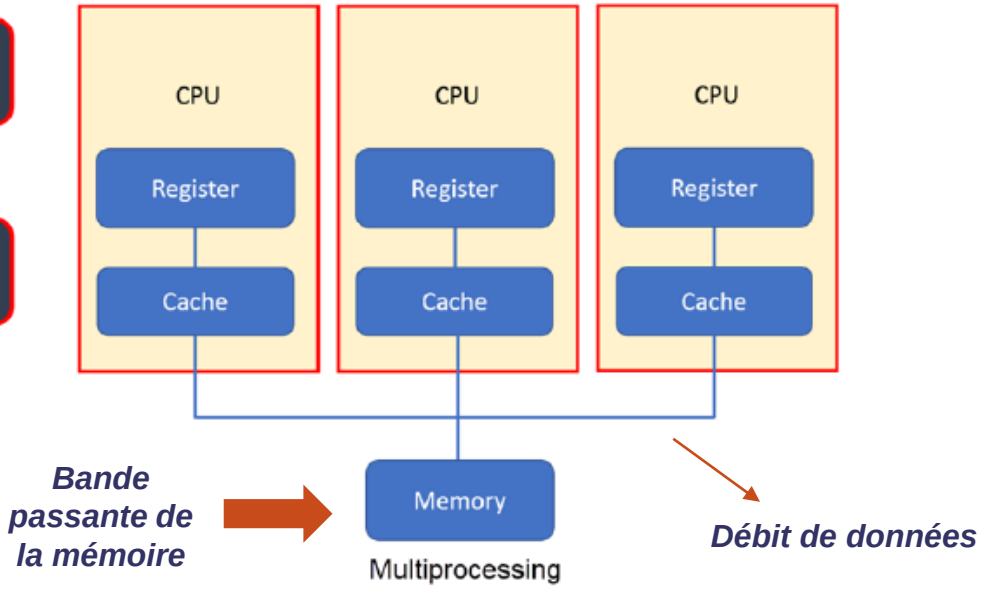


- **Diviser et régner**
- **Glouton**
- **Force brute**
- **Prog. Dynamique**

…

CPU: Modernisation de l' Architecture de Von Neumann



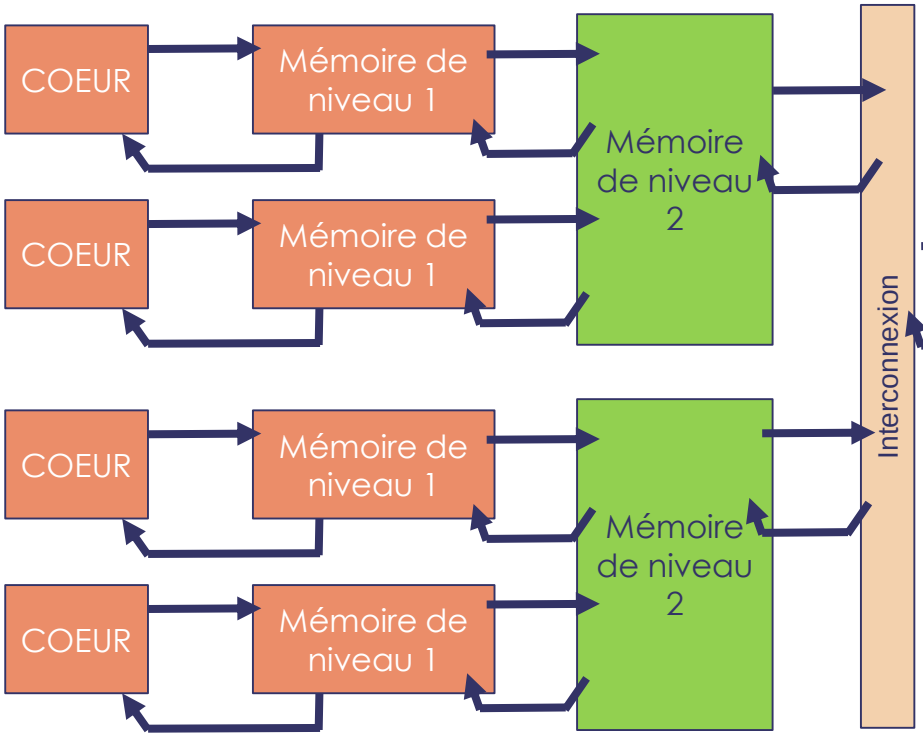


Partage des ressources sous forme de tâches en parallele

Plusieurs cœurs (UAL) mais une seule mémoire

Les performaces de calculs dependent de la bande passante de la mémoire (DDR4 (Gbit/s)) et du débit de données entre les différents cœurs et la mémoire

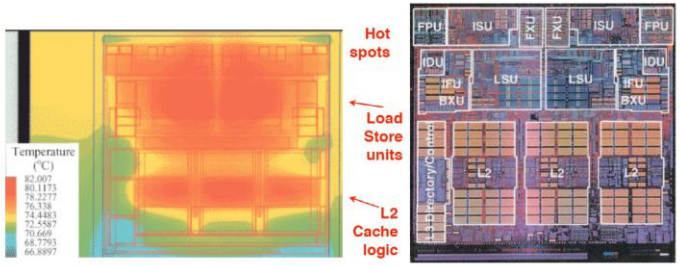
CPU: Modernisation de l' Architecture de Von Neumann

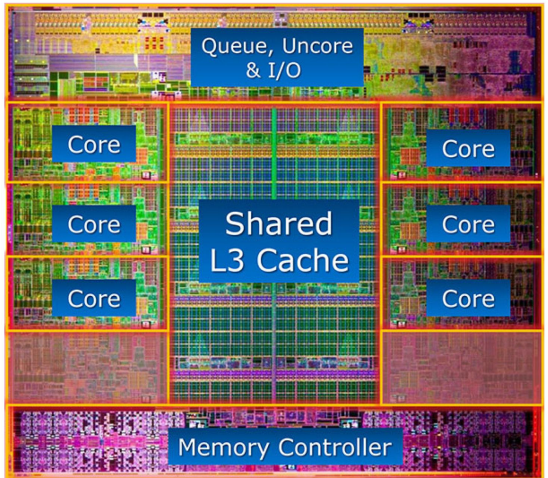


Le débit des données et la bande passante mémoire entraine une consommation importante de puissance:

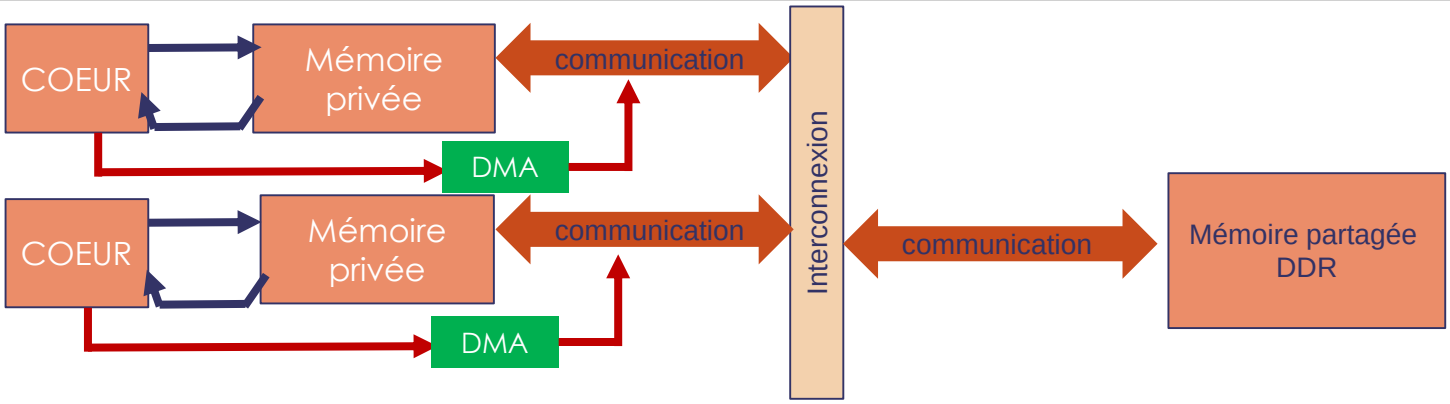
Pw ppl f²

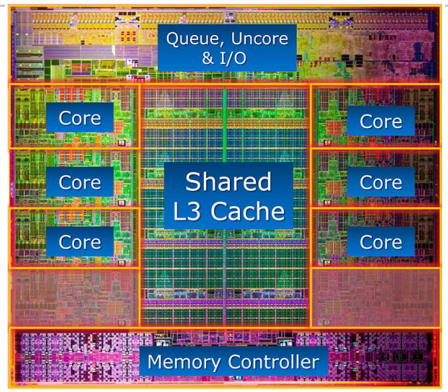
Importance de la hiérarchie mémoire





CPU: Modernisation de l' Architecture de Von Neumann





La latence de la mémoire est un mur physique:

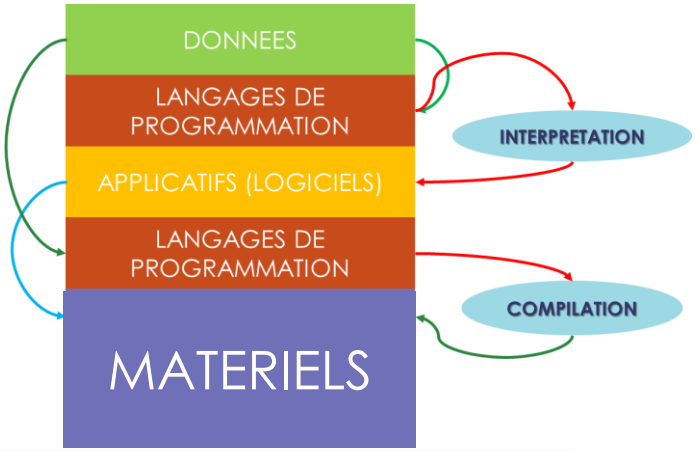
- ➔ C'est une barriere fondamentale pour l'amélioration des performances de calculs
- ➔ On essaye de mixer les types de mémoire pour améliorer la latence (DMA, cache, execution en parallele)
- ➔ multi-processing: on exécute des thread locallement en attendant la liberation de la mémoire

on doit augmenter la bande passante de la mémoire pour compenser le temps de latence

➔ Calcul massivement parallèle

Soulager le CPU par du co-processing dédié à un certain type de calcul

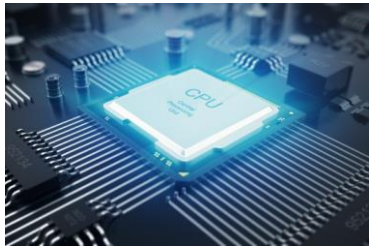
Un mot de la technologie de l'information



Ordinateur actuel
Central Processing Unit

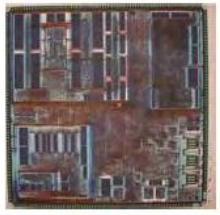
Ordinateur quantique
Langage spécifique

Calcul haute performance

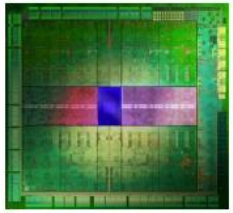


Langage C, C++ et outils spécifiques de compilation

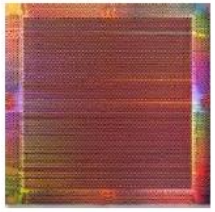
CPUs



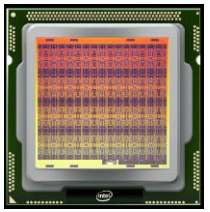
DSPs
microcontroller



GPUs



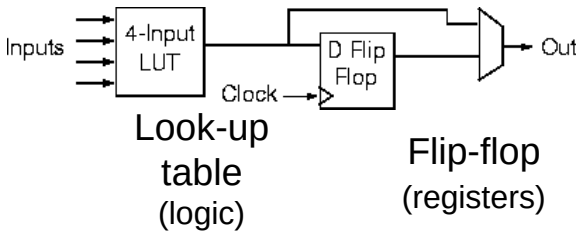
FPGAs



NMC

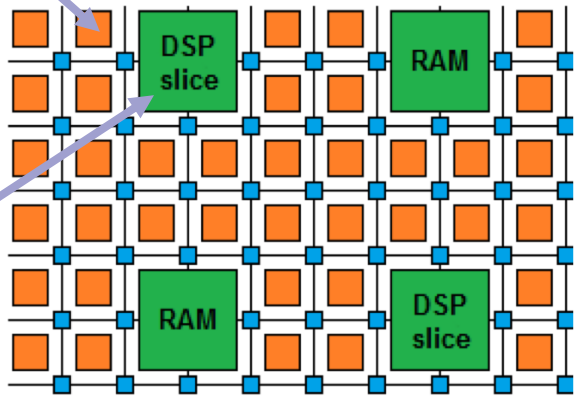
Architecture Full Programmable Gate Array (FPGA)

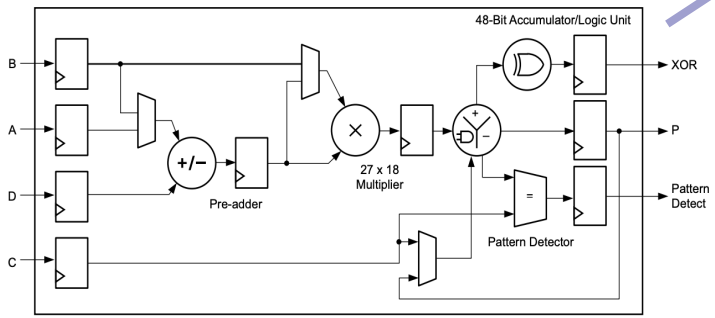
- Le FPGA est un circuit logique reprogrammable
- Logic cells / Look Up Tables créent des fonctions arbitraries et des entrées logiques: operations booléennes, arithmétiques et petites mémoires
- Registre Flip-Flops enregistre les données selon l'impulsion d'horloge
- **DSPs (Digital Signal Processor)** sont specialisés dans les multiplications et les operation sarithmétiques. Plus rapide et plus efficacies que les CLB (LUT)

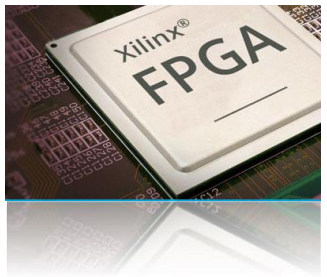


CLB/LUT

diagramme FPGA







What are FPGAs?

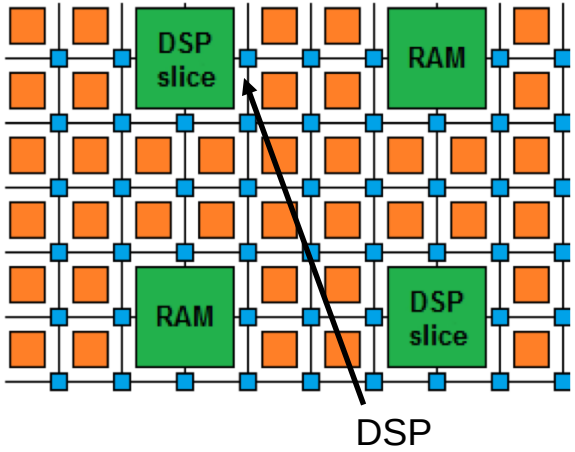
Field Programmable Gate Arrays are reprogrammable integrated circuits

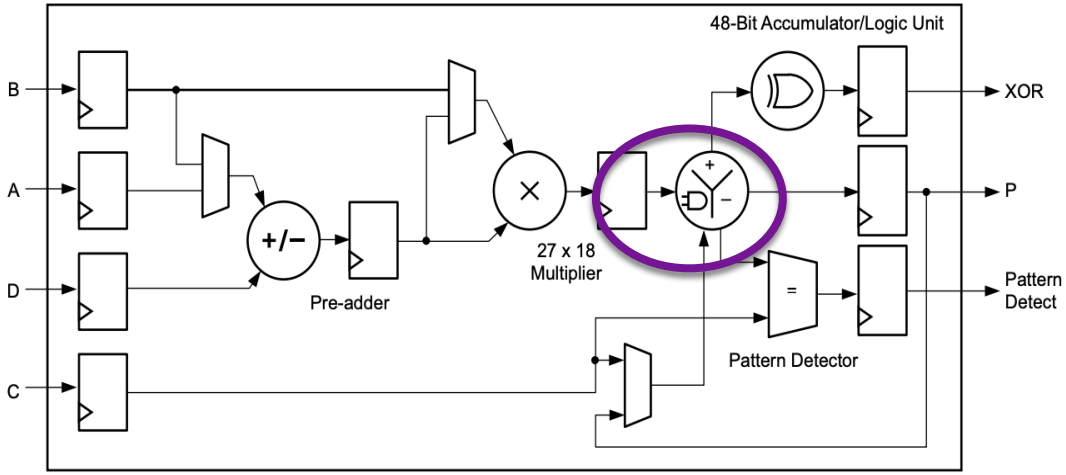
DSPs (Digital Signal Processor) are specialized units for multiplication and arithmetic

Faster and more efficient than using LUTs for these types of operations

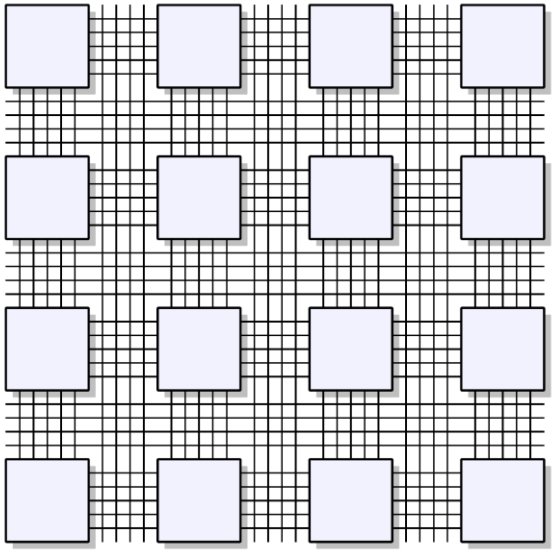
And for Neural Nets, DSPs are often the most scarce

FPGA diagram

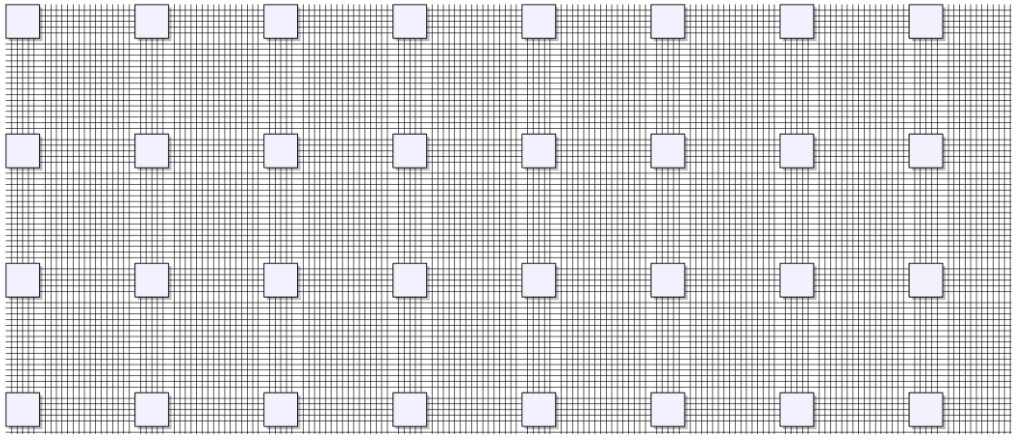




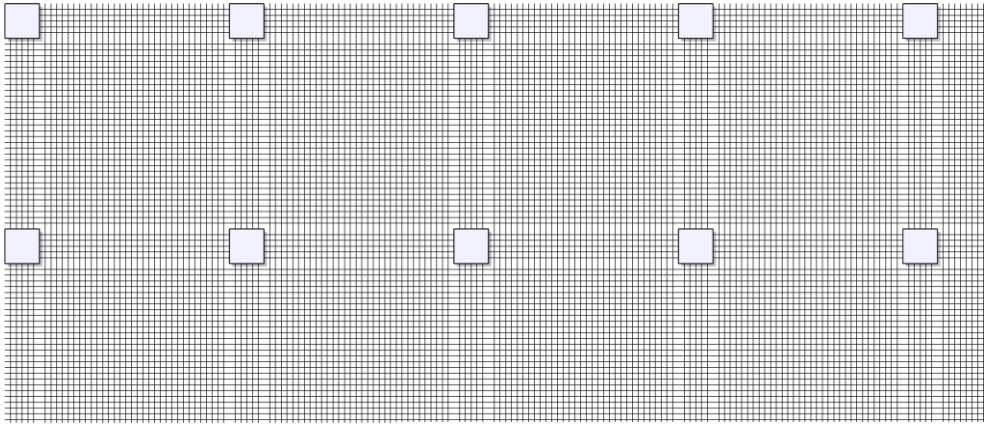
Architecture Full Programmable Gate Array (FPGA)



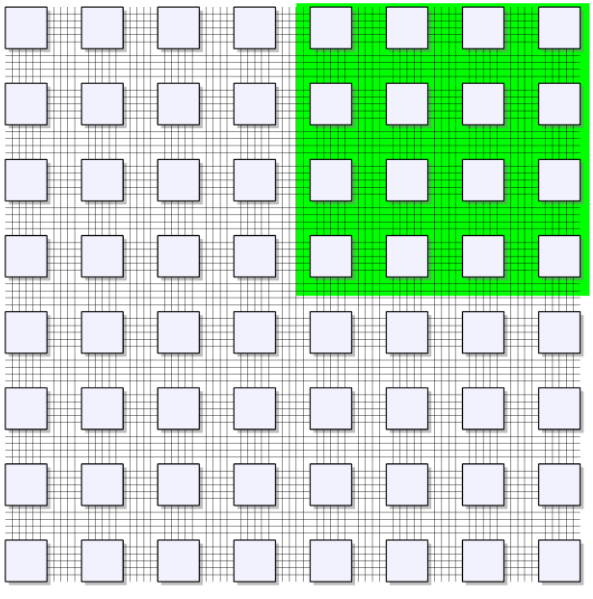
1990

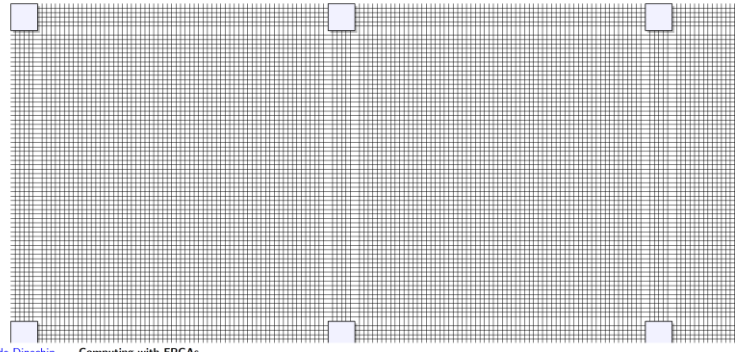


2001
2001



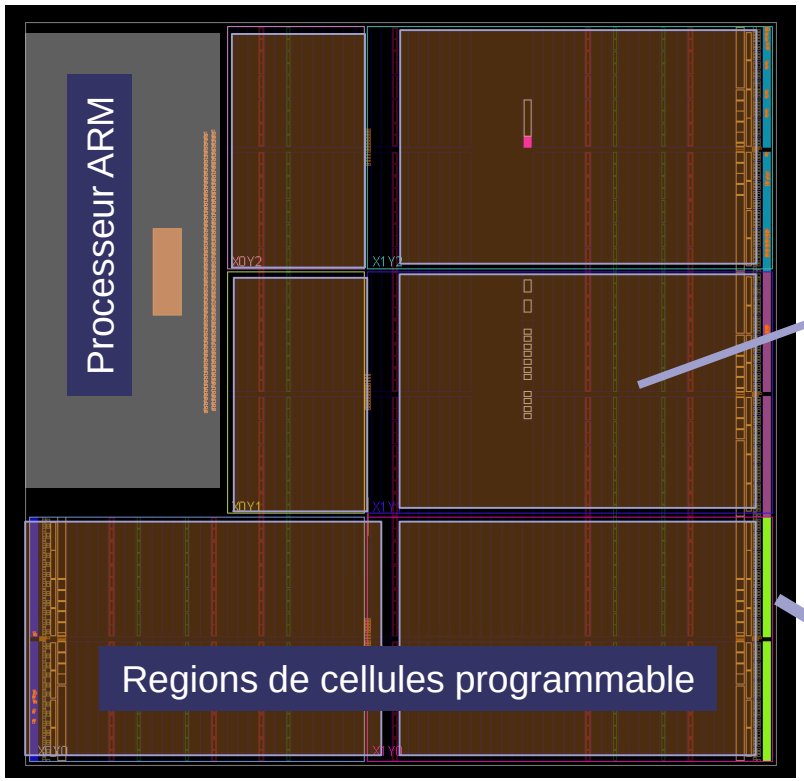
1996

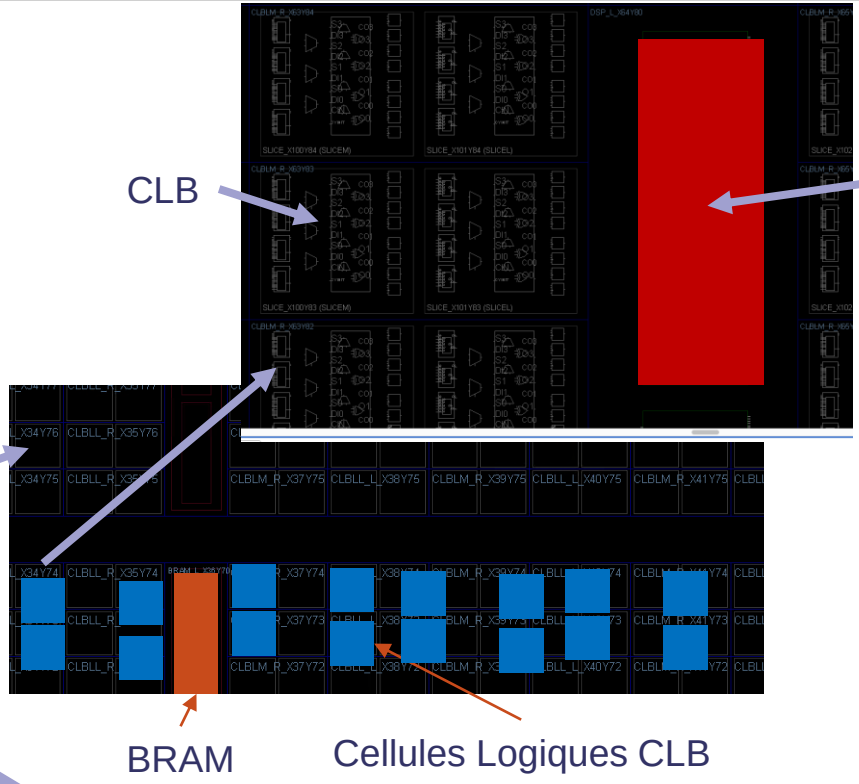


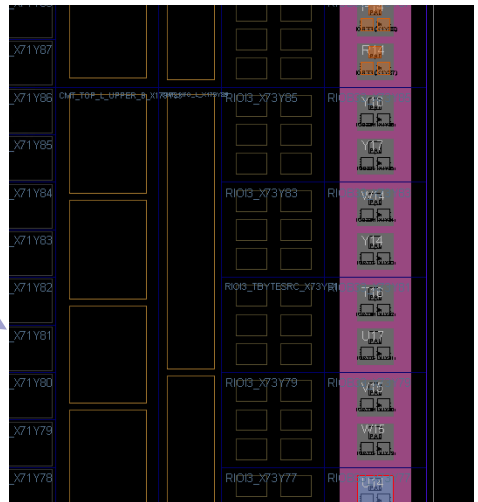


Architecture Full Programmable Gate Array (FPGA)

AMD Xilinx Zynq (Zc020)



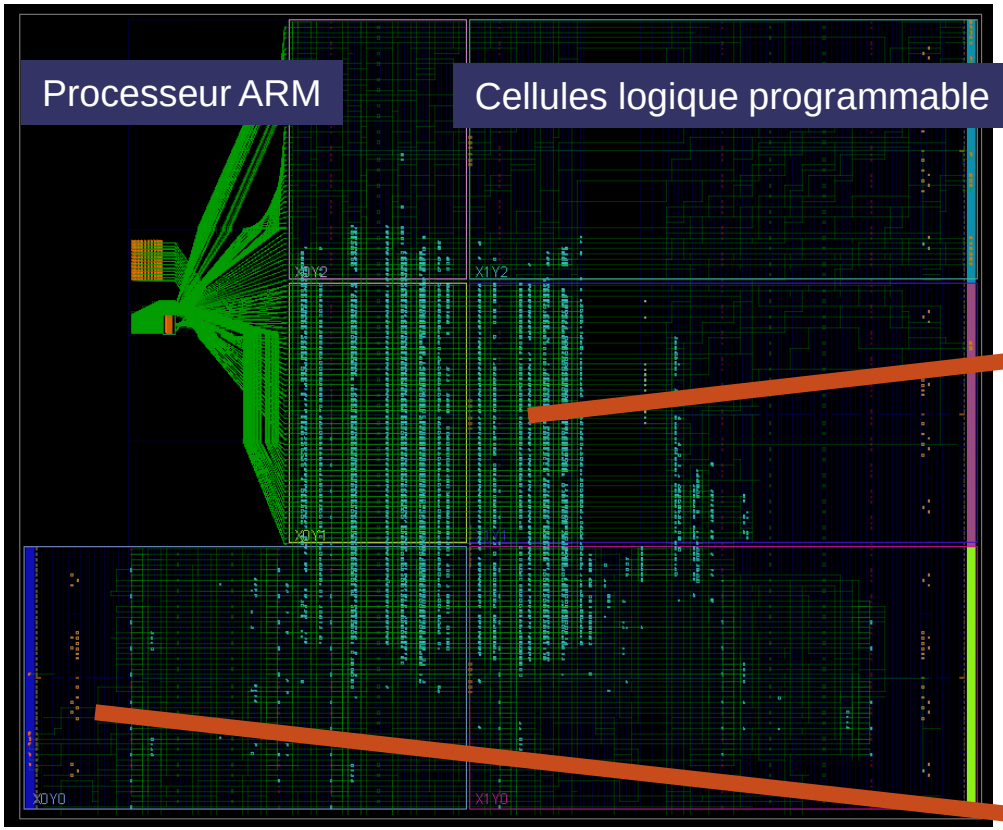


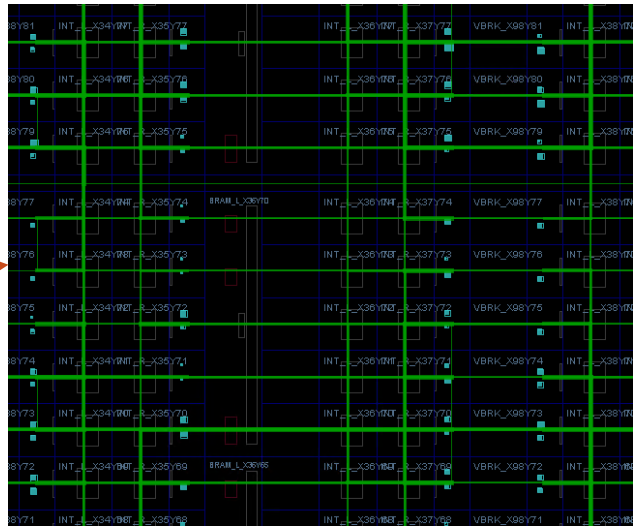


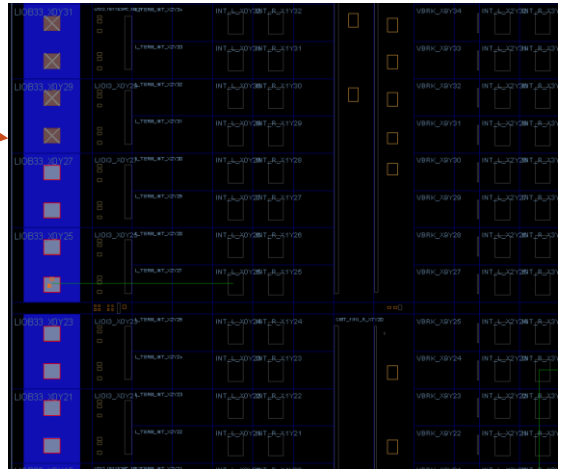
IO Programmable avec buffer

Architecture Full Programmable Gate Array (FPGA)

AMD Xilinx Zynq (Zc020)







How are FPGAs programmed?

Hardware Description Languages

HDLs are programming languages which describe electronic circuits

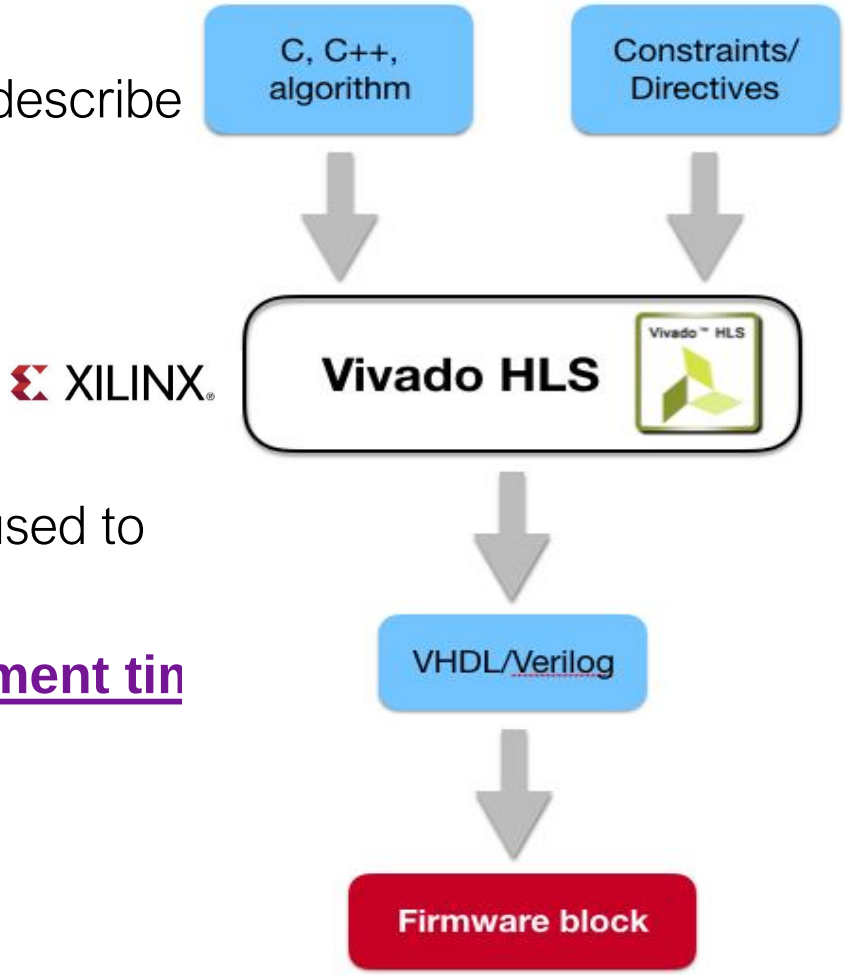
High Level Synthesis

Compile from C/C++ to VHDL

Pre-processor directives and constraints used to optimize the design

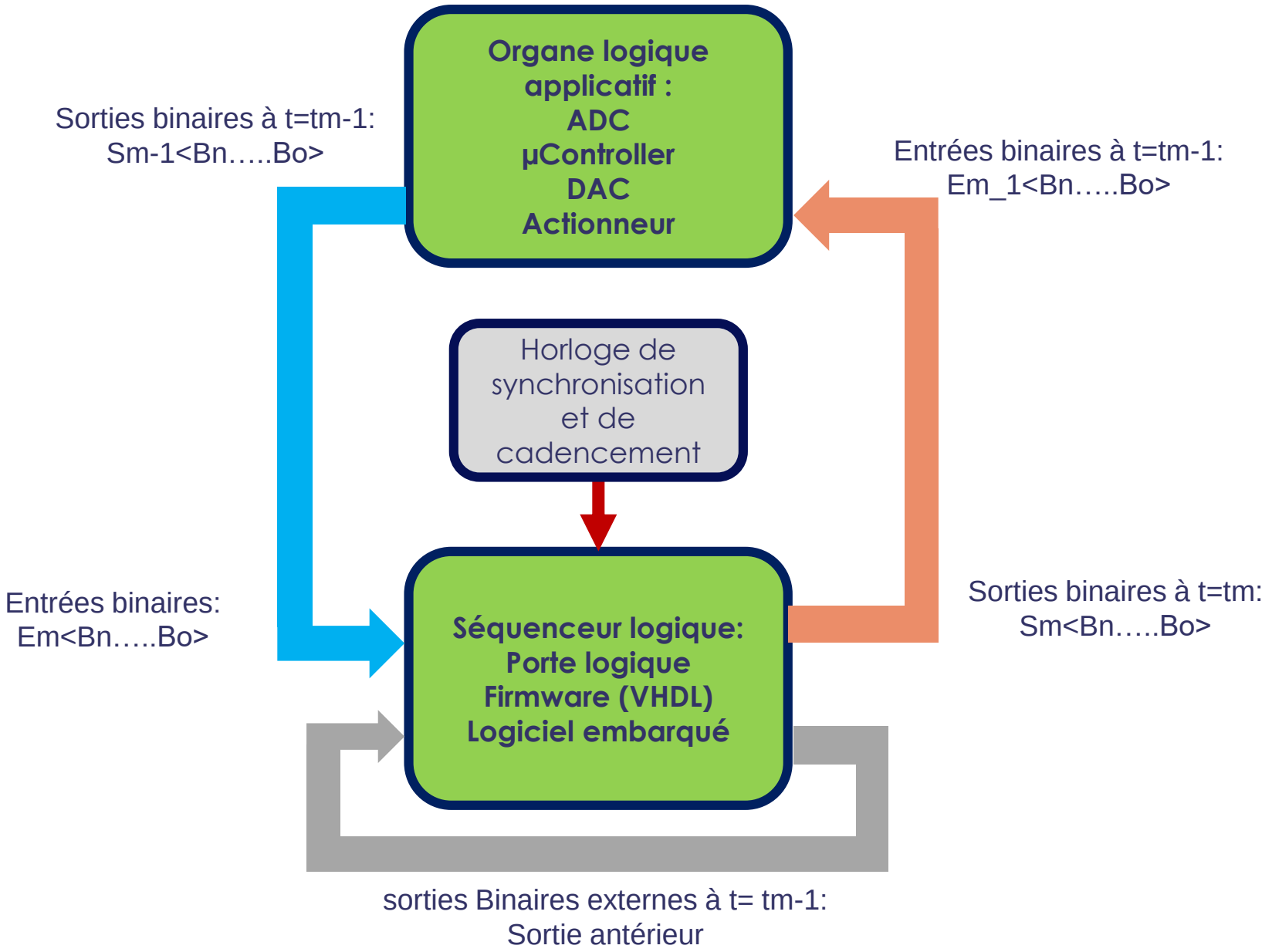
Drastic decrease in firmware development tin

Today we'll use Xilinx Vivado HLS [*]

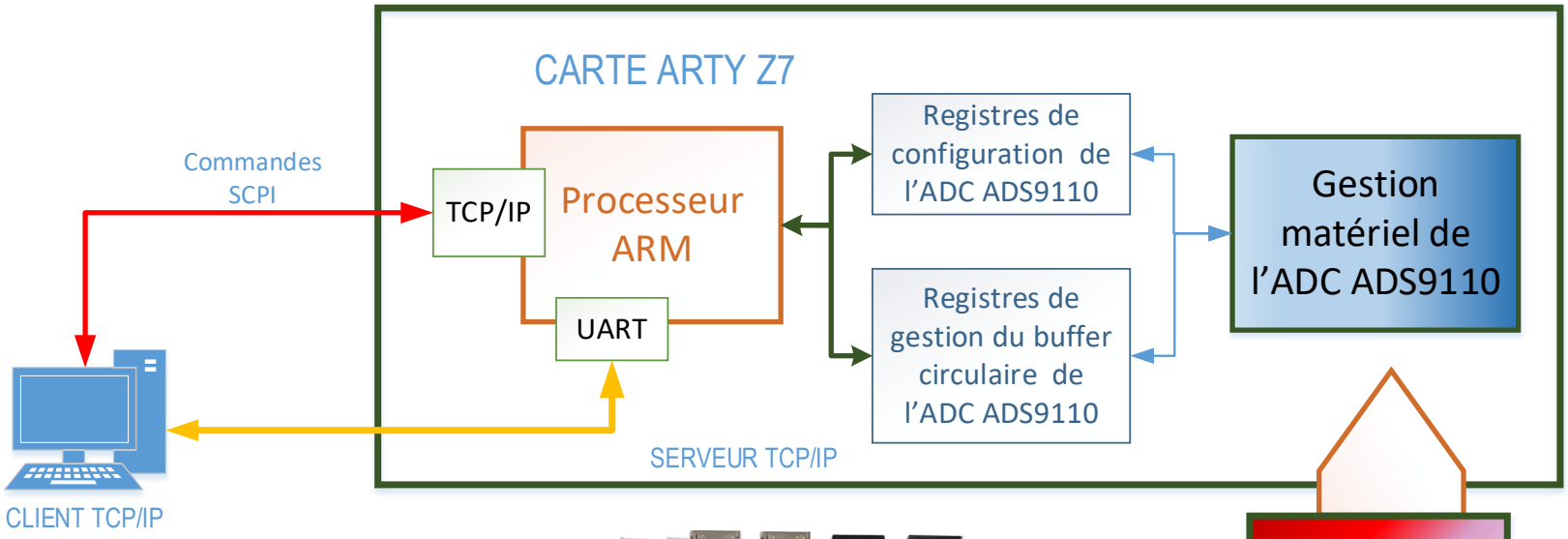


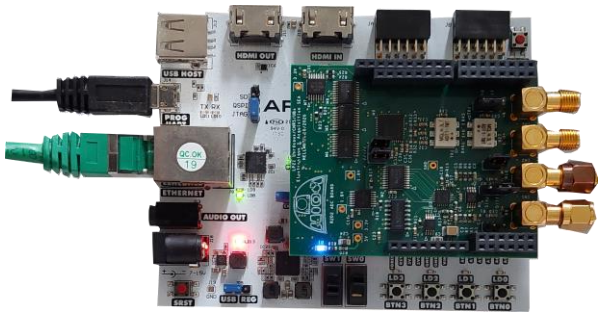
[*]
https://www.xilinx.com/support/documentation/sw_manuals/xilinx2020_1/ug902-vivado-high-level-synthesis.pdf

Les séquenceurs



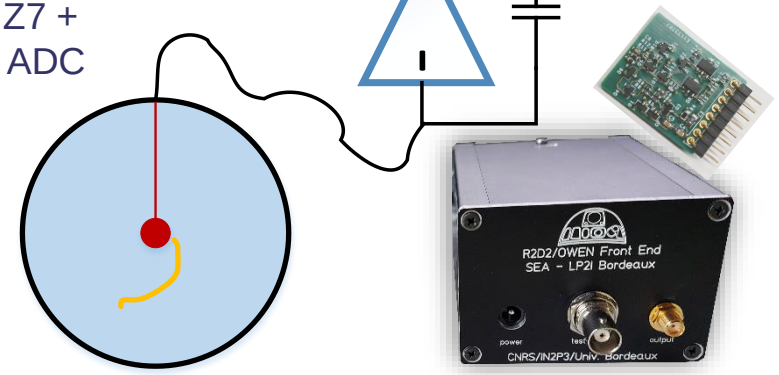
Système d'acquisition : Architecture





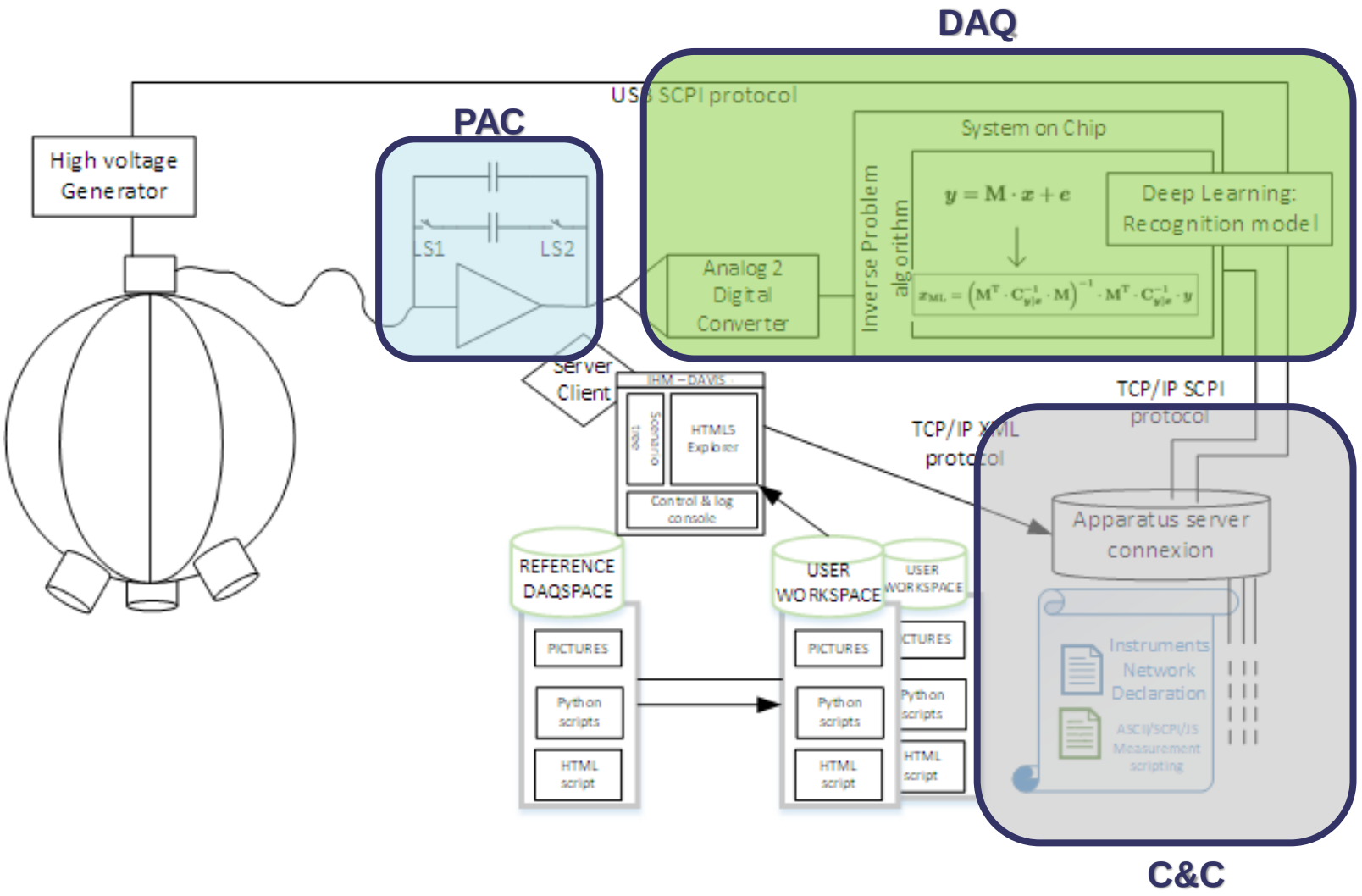
Carte ARTY Z7 + Mezzanine 2 ADC

Carte fille 2 voies d'acquisition

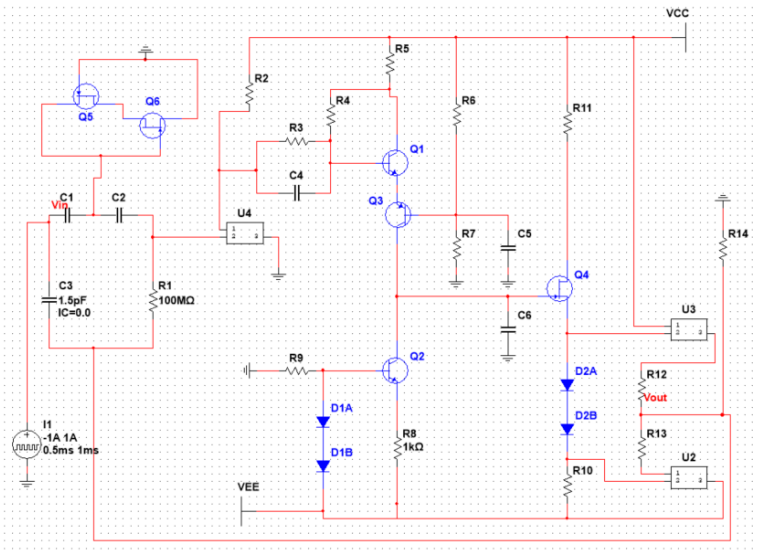


- Architecture client/Serveur
- Gére l'ADC ADC9110 (SAR)
  - Configuration
  - Acquisition
- Gère trigger L0
- DAQ dataStream (FiFo 1 evenement)

Synoptique du système

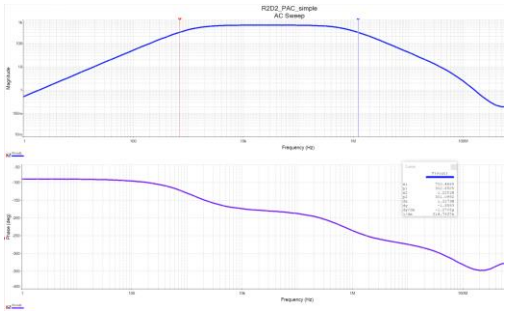


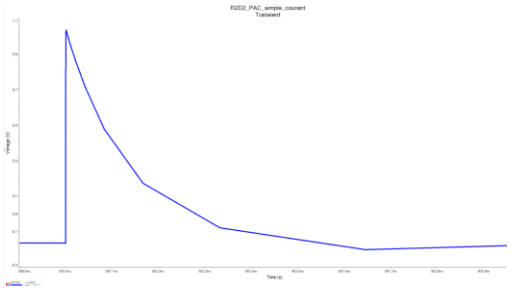
Le préamplificateur de charge

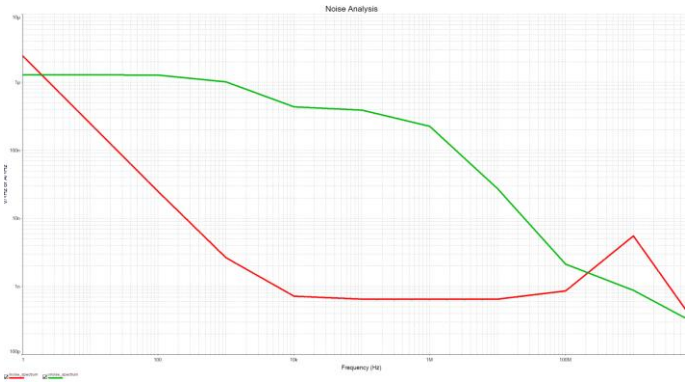


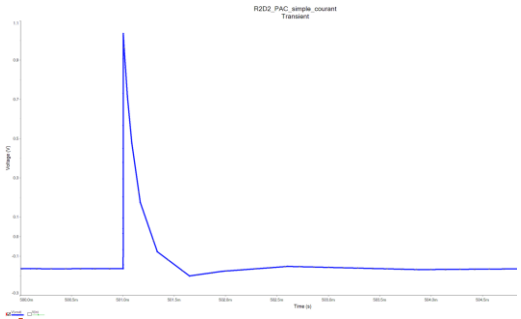
PAC cascode + Ampli classe A
Cf = 1,5pF
Vout = +4,3V/-3,4V
Treset = 360µs
BP = 1,2MHz
Noise = 418µVrms

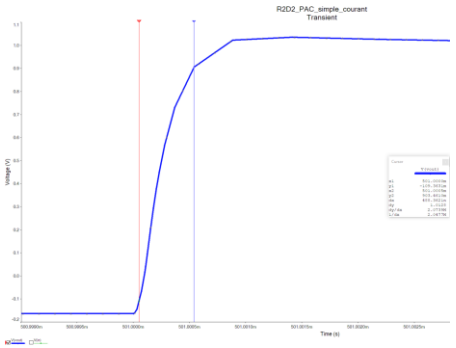
BODE



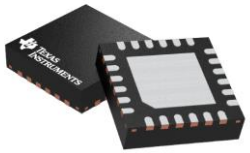


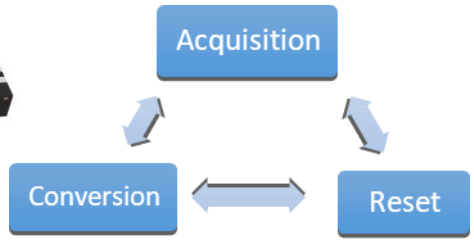


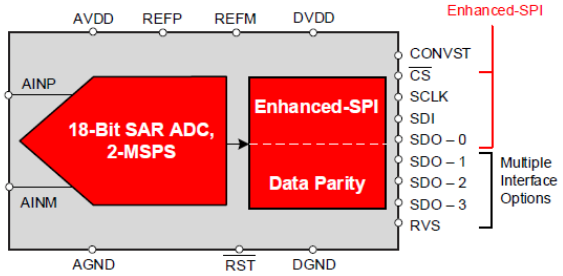


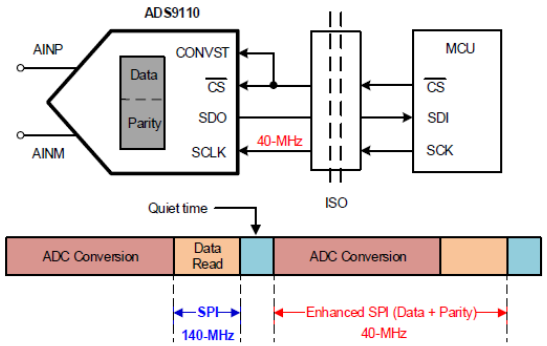


Système d'acquisition : ADC AD9110









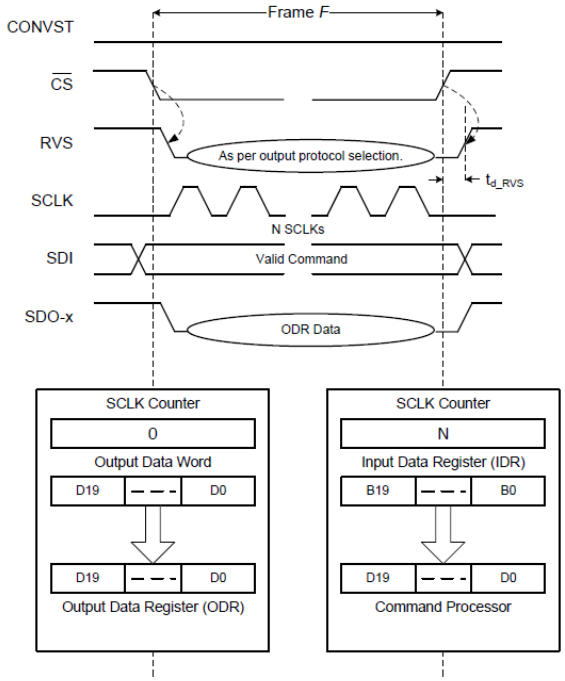


Figure 46. Data Transfer Frame

Configuration & Acquisition en même temps

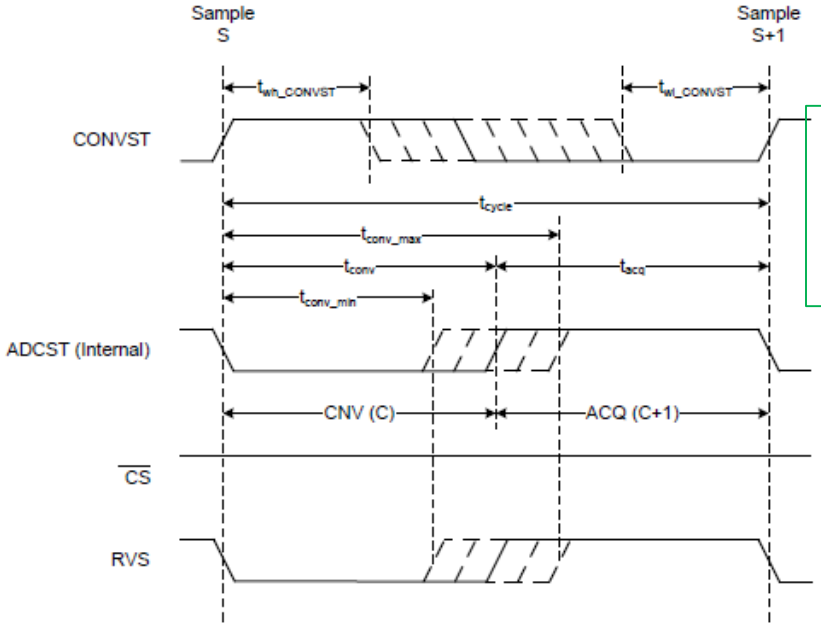


Figure 43. Typical Conversion Process

Un mode de lecture sélectionné : Echantillon sur 4 voies à 40MHz

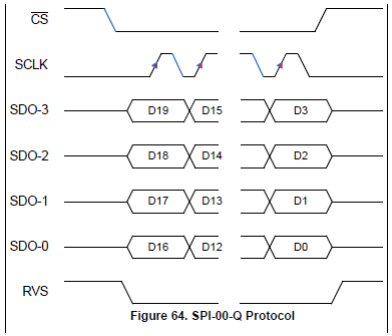
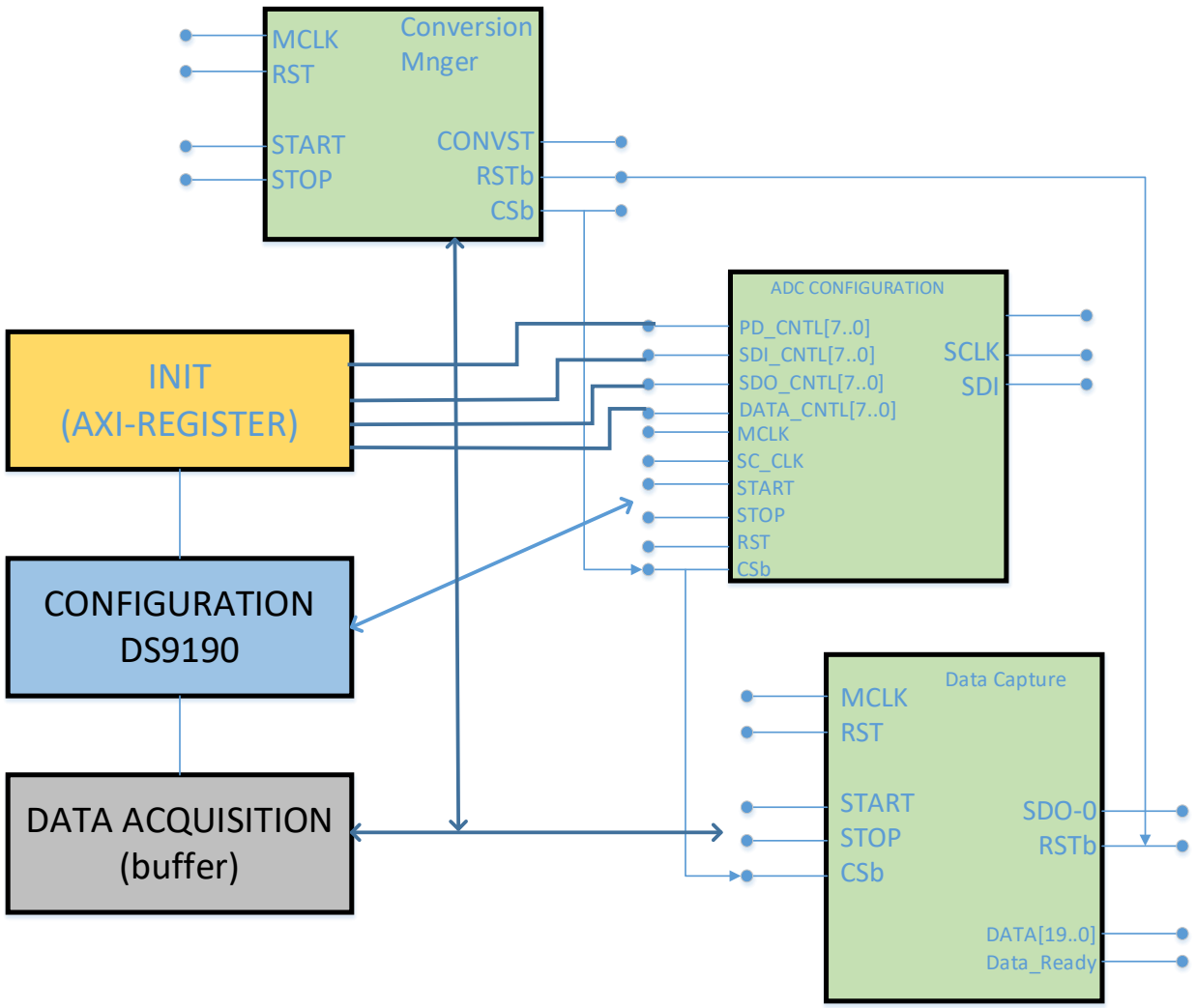


Figure 64. SPI-00-Q Protocol

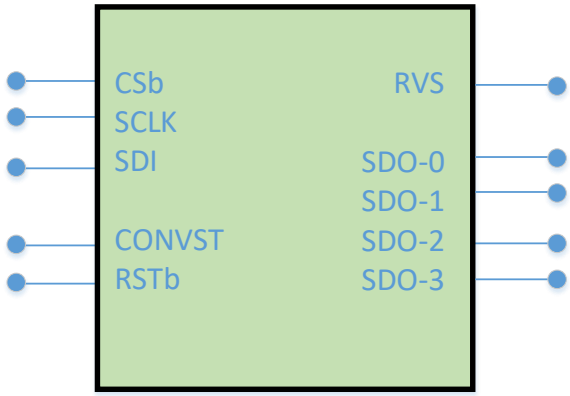
Conversion piloté par commande externe

Système d'acquisition: Firmware ADC

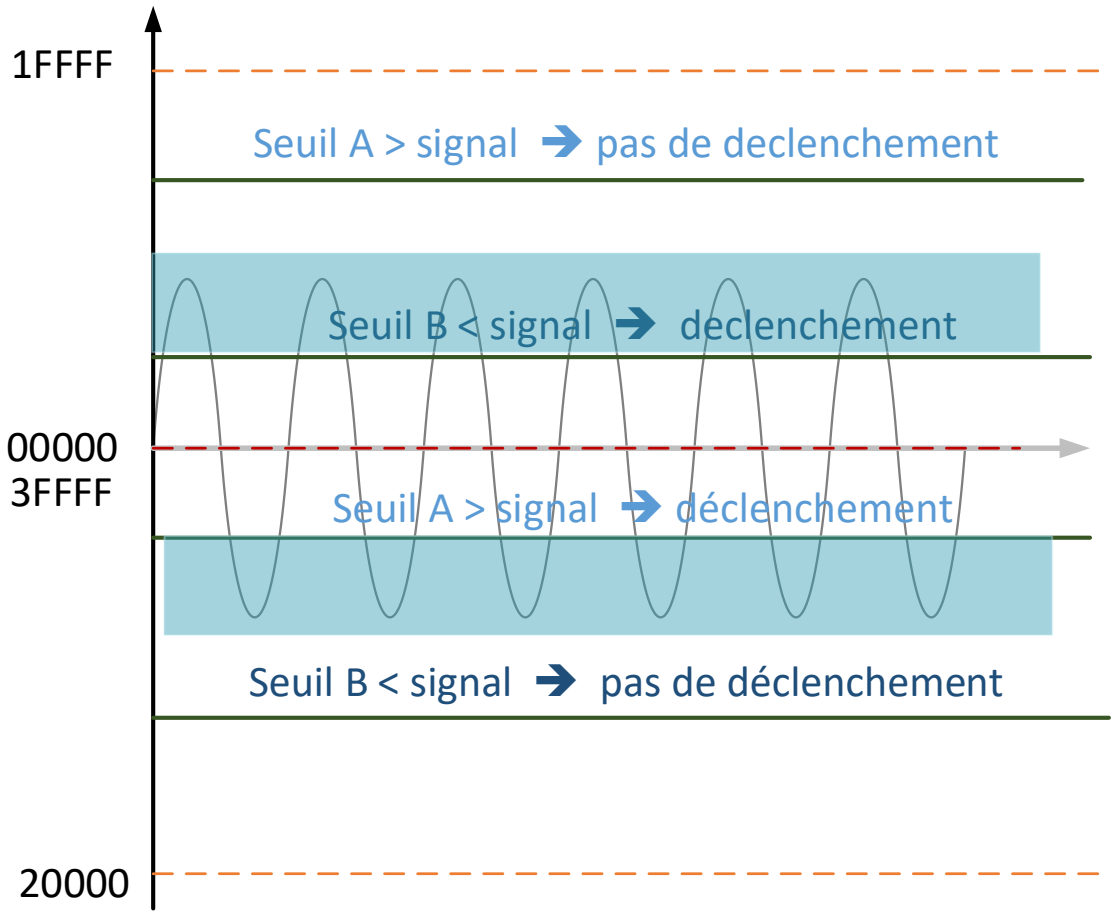
µC ← Architecture firmware ADC → ADC



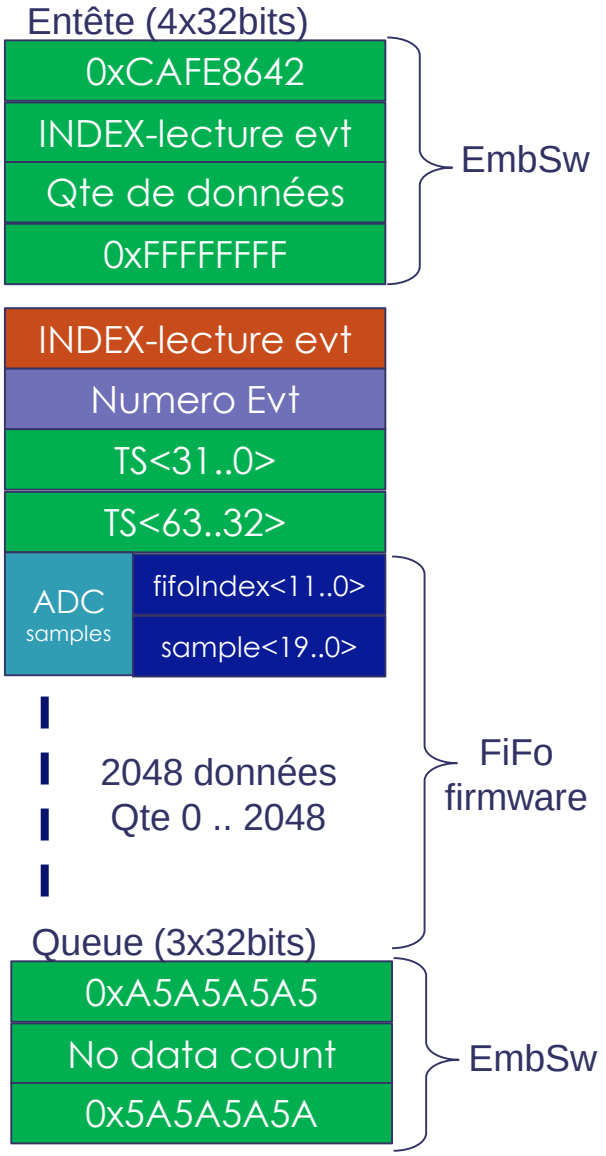
AD9110 → interface firmware



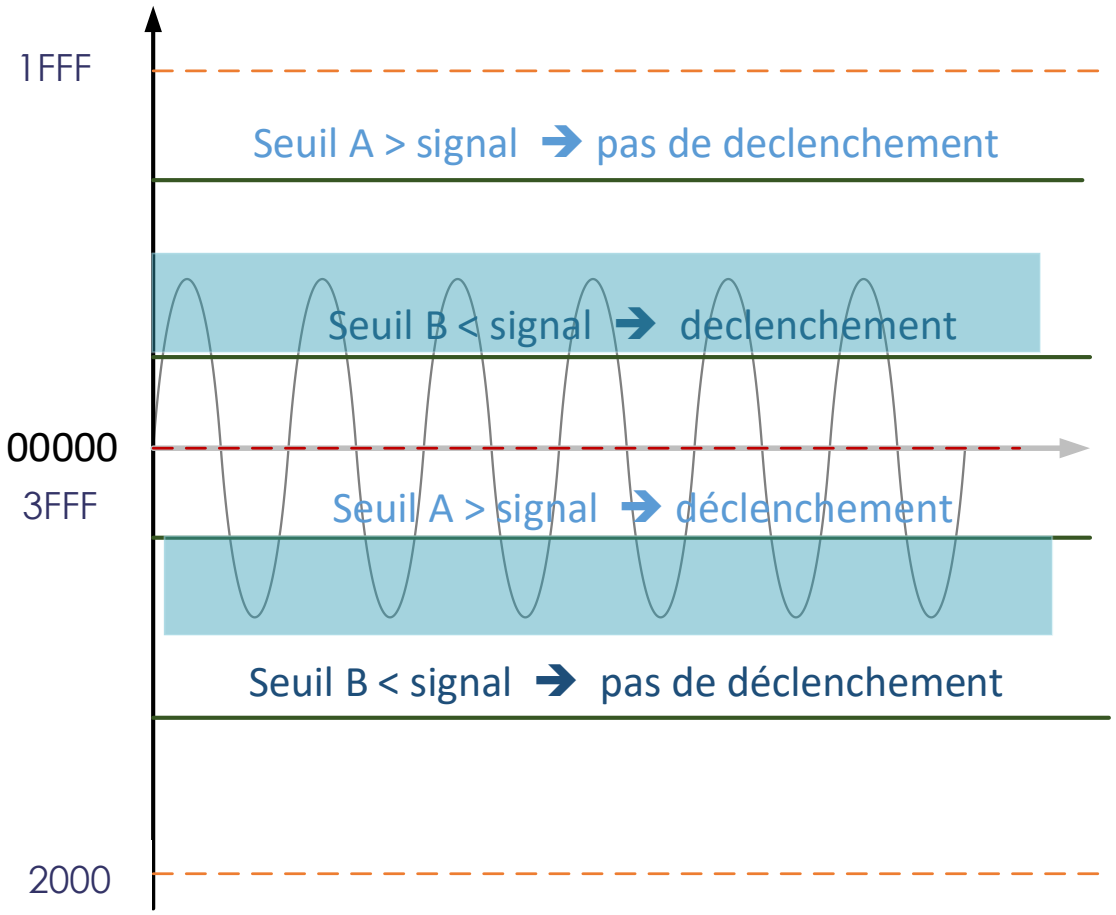
Système d'acquisition: Trigger & Format AD9110



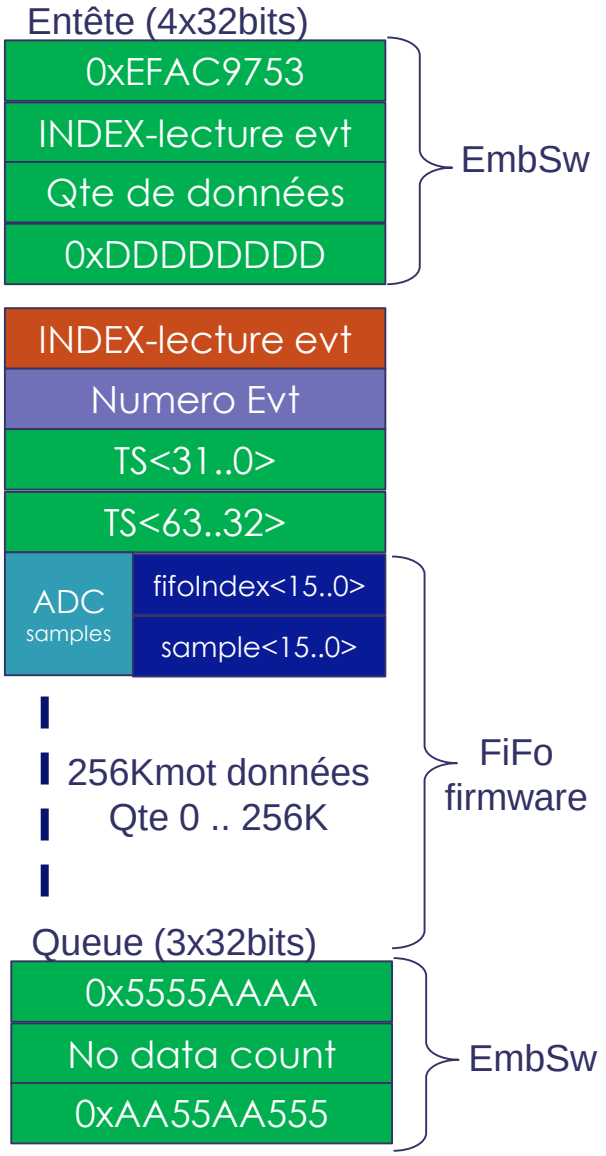
Creation :
- Numero d'évenement 32 bits
- Etiquetage en temps 64bits x 5ns



Système d'acquisition: Trigger & Format ADS41b49

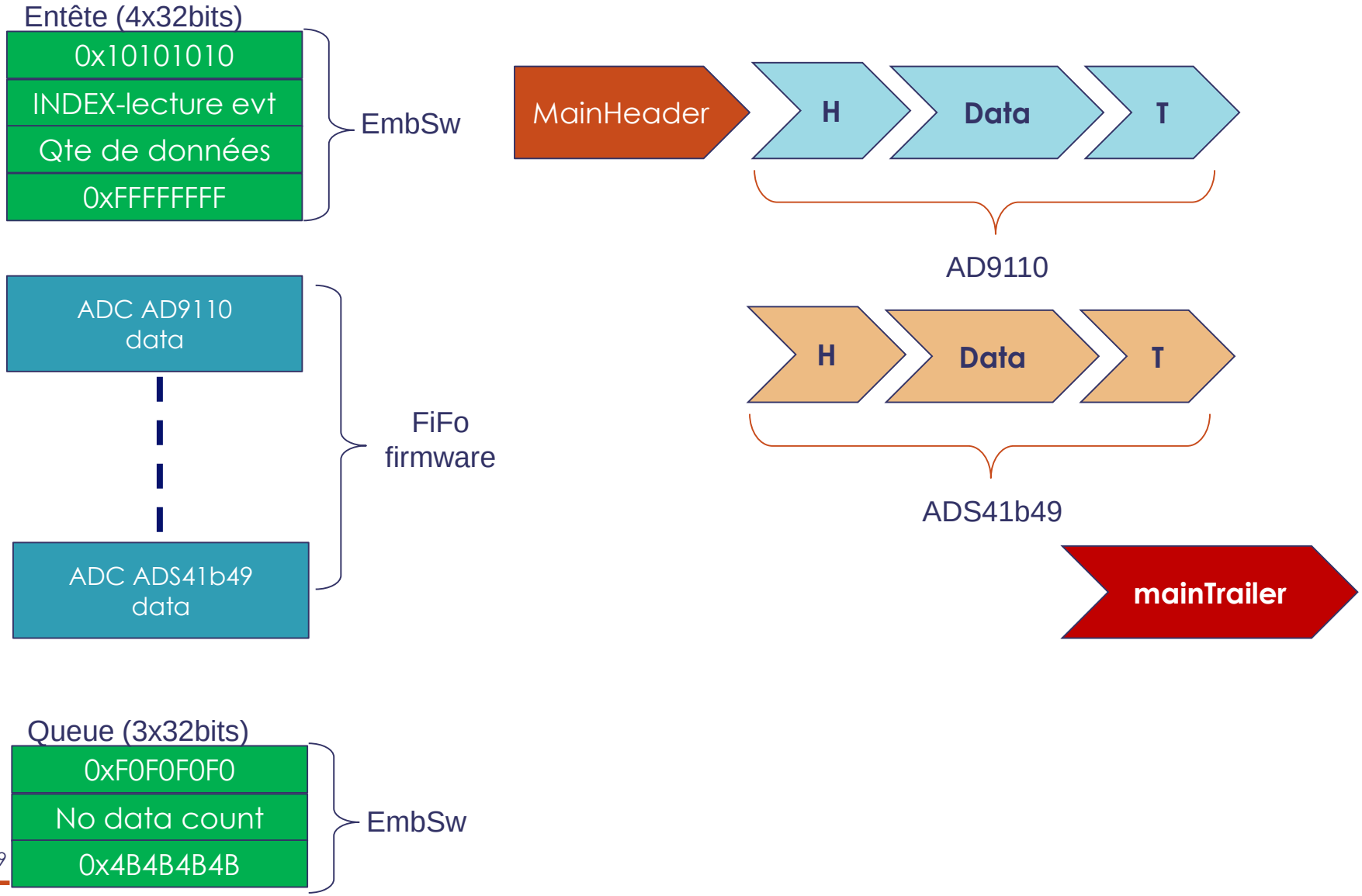


Creation :
- Numero d'évenement 32 bits
- Etiquetage en temps 64bits x 5ns

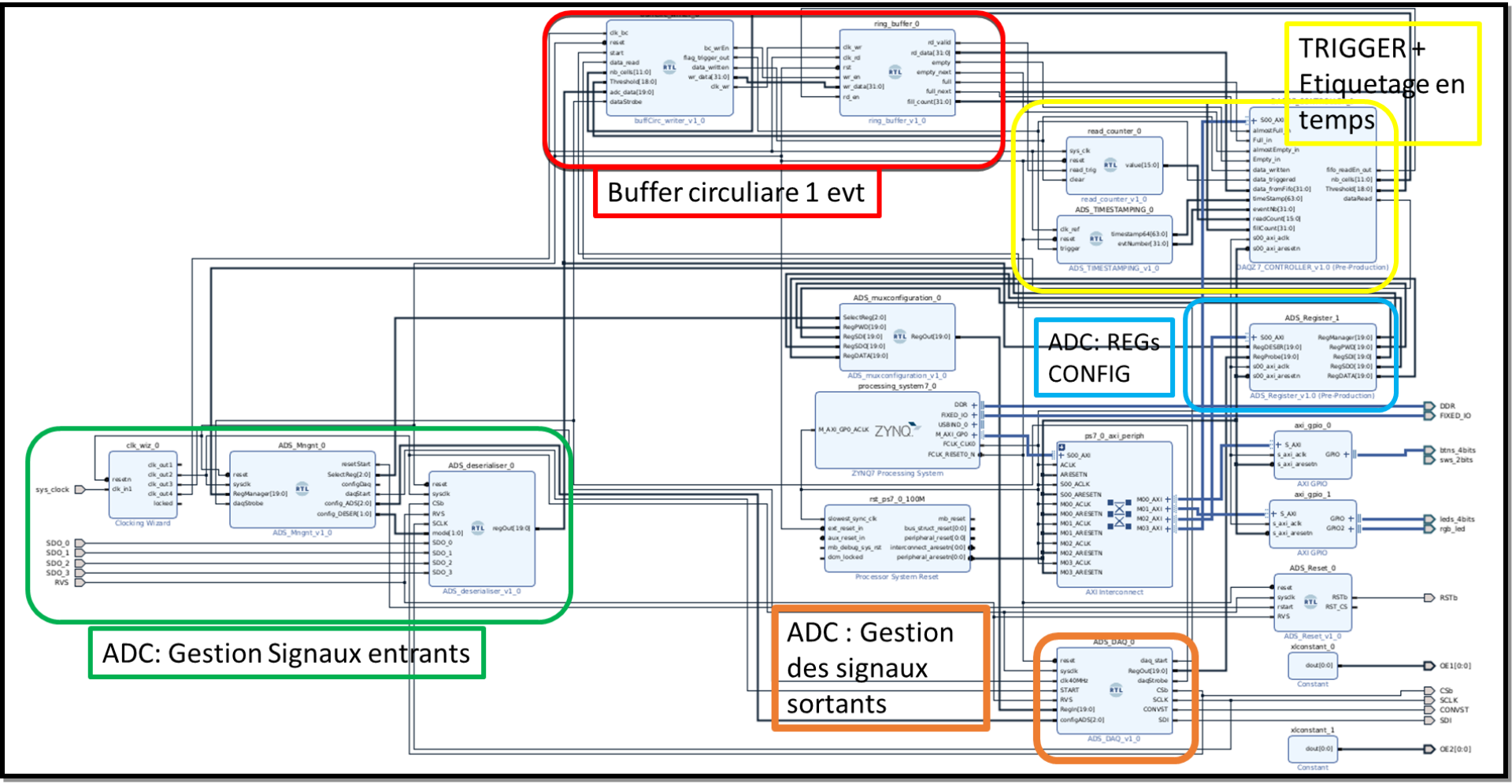


Système d'acquisition: Format DATA AD9110 & ADS41b49

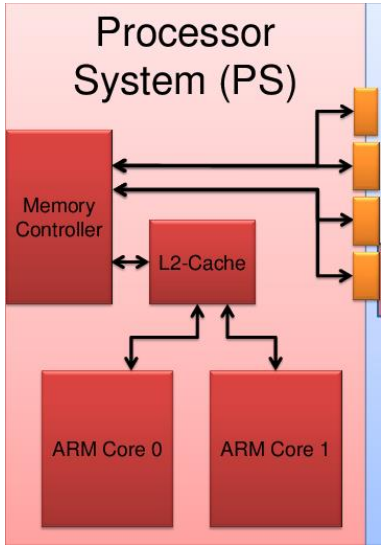
➔ Les données sont encapsulées dans un entête et une queue d'information



Système d'acquisition: Firmware global



Les séquenceurs: Exemple



ASPECT LOGICIEL : Machine d'état

Chargement des données de configuration de l'ADS9110 dans le processeur embarqué

Configuration de l'ADS9110

Demarrage de l'acquisition : START = '1'

Start = '1'

Numerisation de l'entrée en sauvegarde dans le buffer circulaire

Arrêt de l'acquisition : START = '0'

'0'

Event Triggered ?

Registre Data_written

'1'

Arrêt de l'ecriture dans le buffer circulaire
Lecture des données stockées dans la buffer

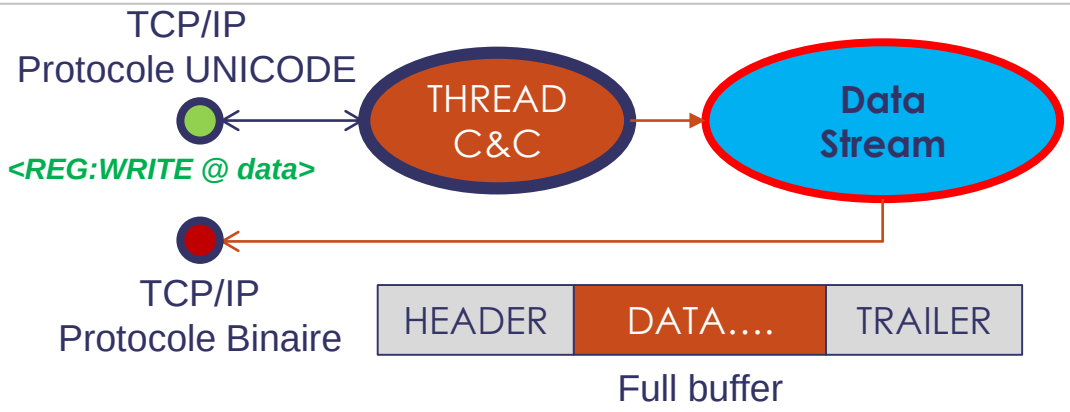
'0'

Fin Lecture du buffer Circulaire

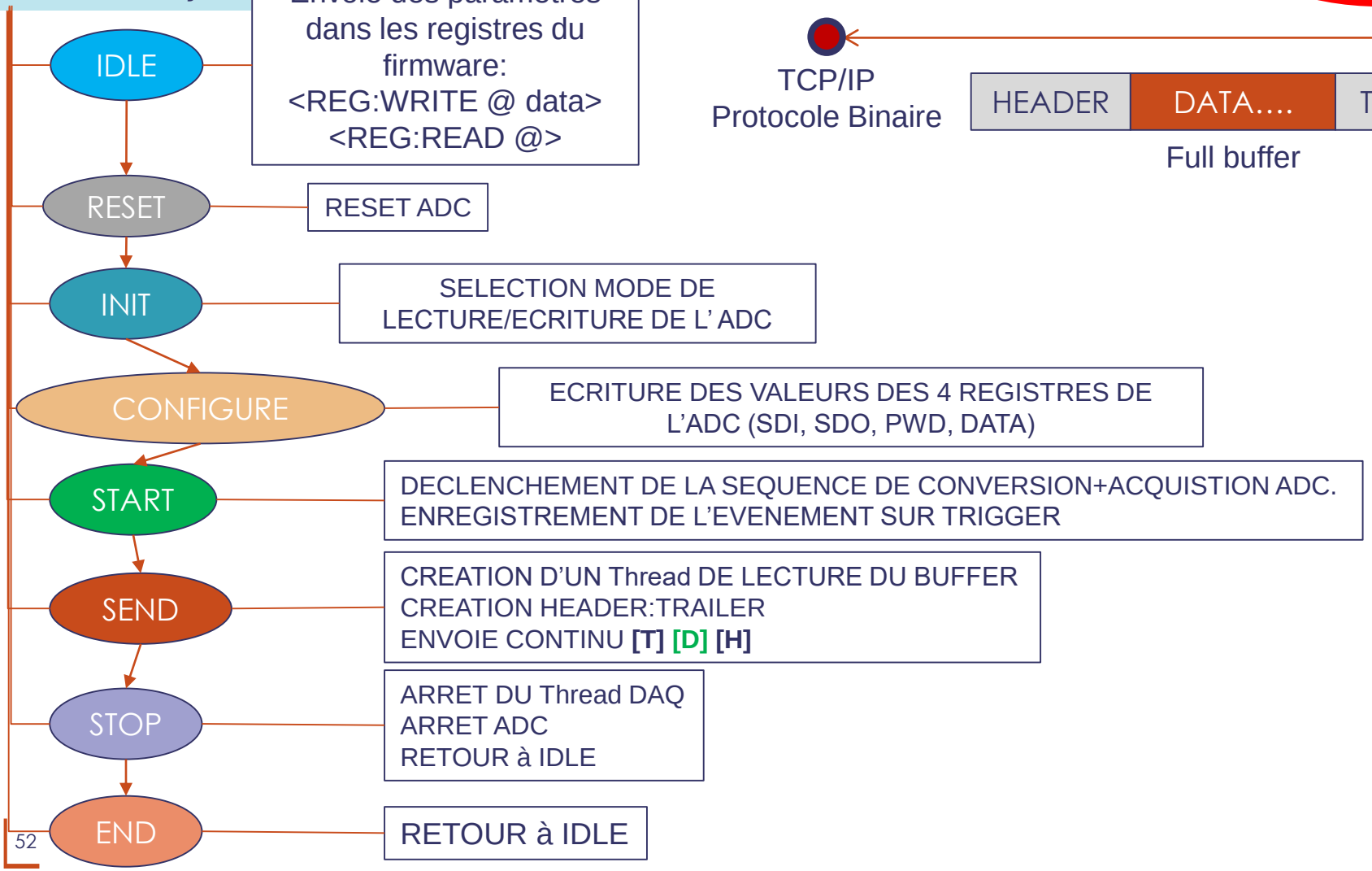
Registre Data_read

Système d'acquisition: logiciel embarqué

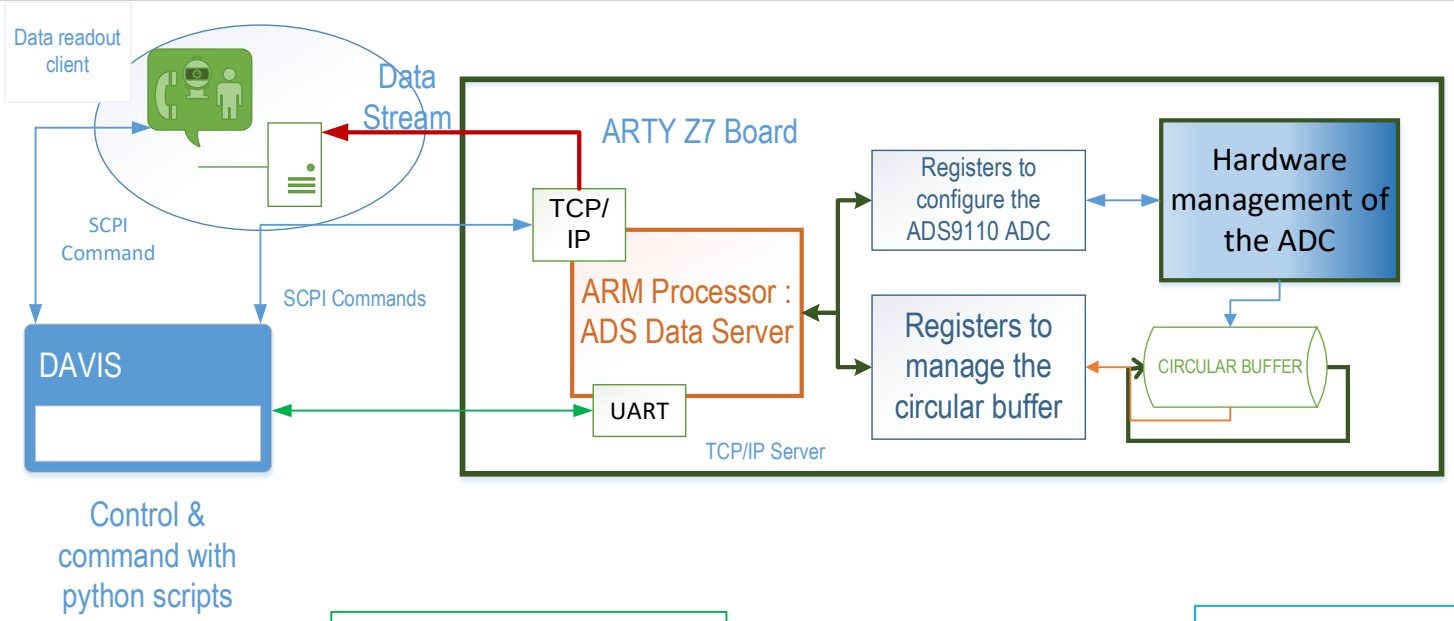
OS: Free RTOS (Xilinx – Vitis)
Langage C

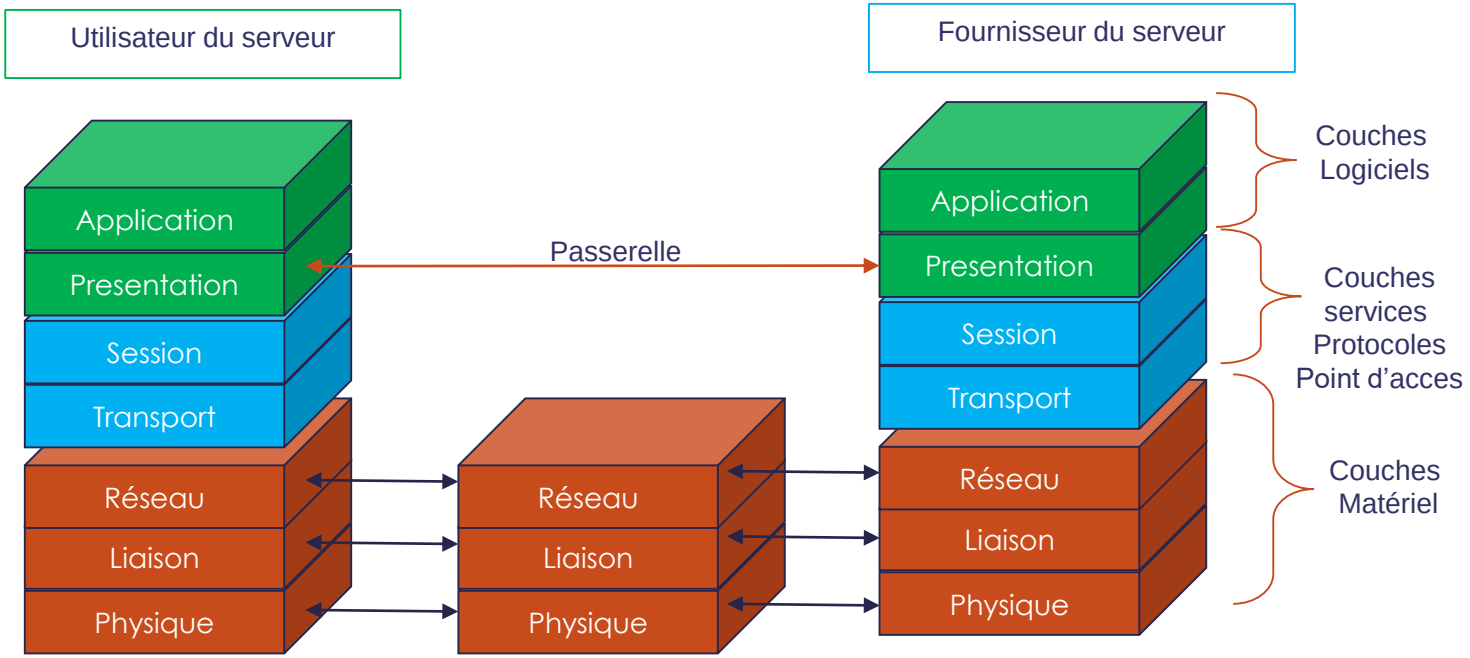


Commande reçue

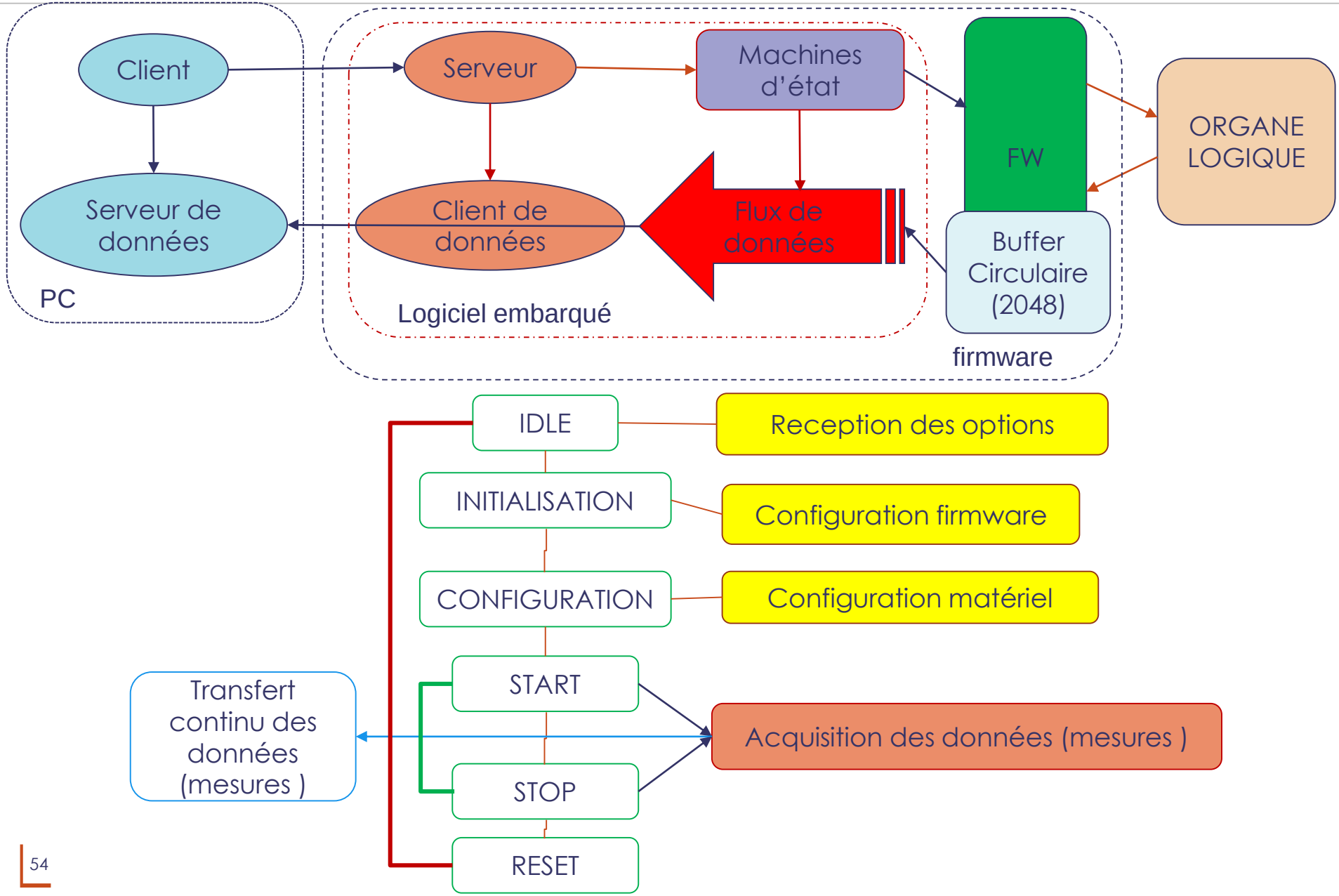


Les séquenceurs: Exemple

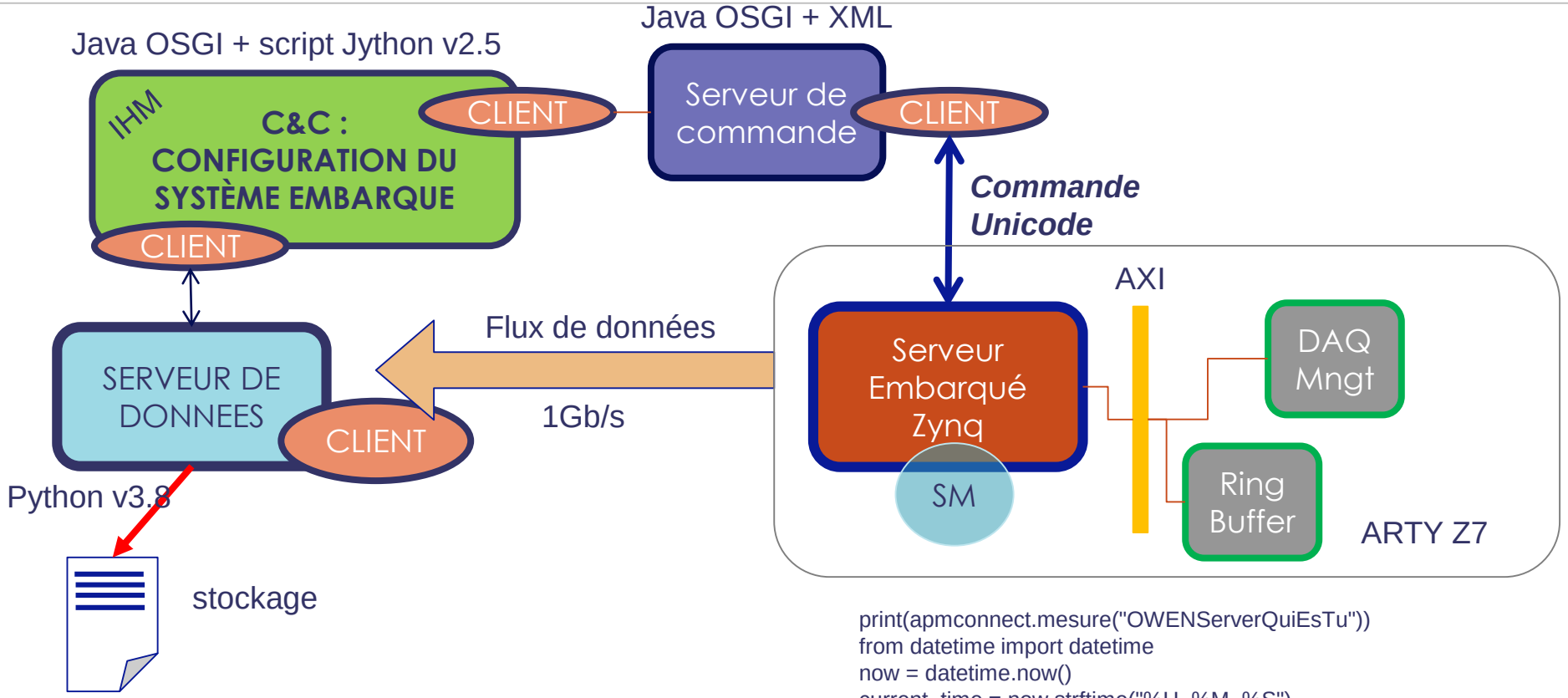




Les séquenceurs: Exemple



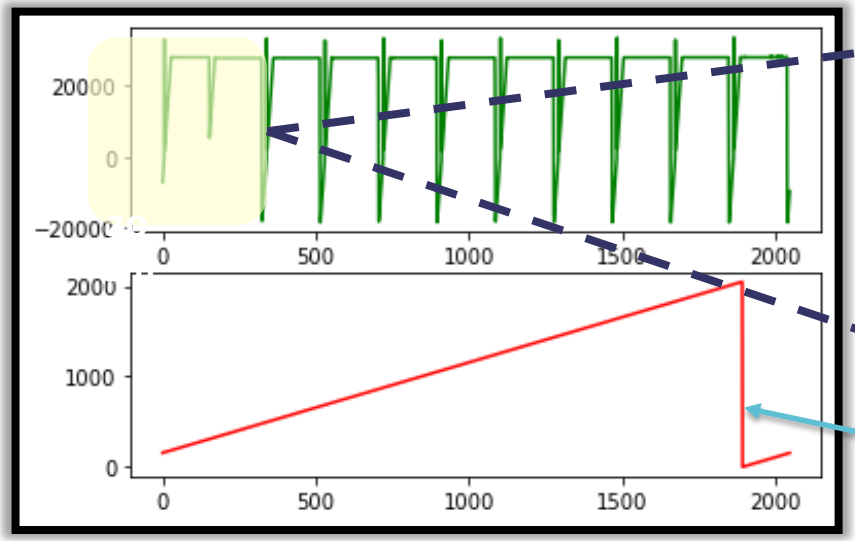
Système d'acquisition : logiciel C&C



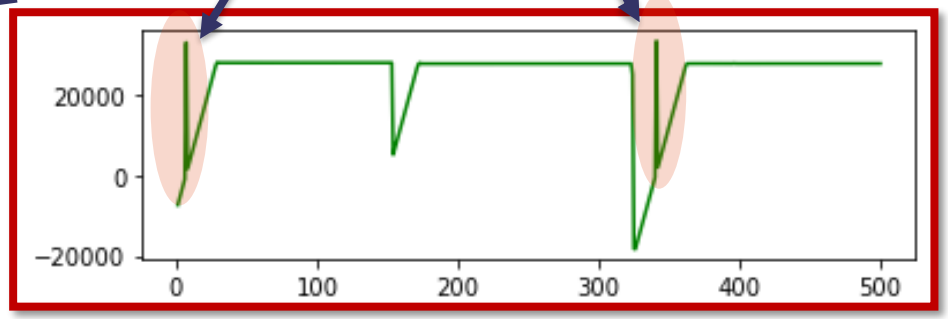

```
print(apmconnect.mesure("OWENWRITEREG ID=9 data=0xCAFEA5A5"))
print(apmconnect.mesure("OWENWRITEREG ID=11 data=0x5A5AFEDC"))
print(apmconnect.mesure("OWENREADREG ID=9"))
print(apmconnect.mesure("OWENMODE state=0"))
print(apmconnect.mesure("OWENTHRESHOLD thres=0x50032"))
print(apmconnect.mesure("OWENNBCELLS NB=2200"))

print(apmconnect.mesure("OWENSTATE state=RESETTING"))
print(apmconnect.mesure("OWENSTATE state=INITIALISING"))
print(apmconnect.mesure("OWENSTATE state=CONFIGURING"))
print(apmconnect.mesure("OWENSTATE state=RUNNING"))
```

```
print(apmconnect.mesure("OWENServerQuiEsTu"))
from datetime import datetime
now = datetime.now()
current_time = now.strftime("%H_%M_%S")
DataPath = "D\\\DATA"
file2Store = DataPath+"\\\OWENDATA_"+current_time+'.bin'
print(apmconnect.mesure("OWENServerFile fname="+file2Store))
print(apmconnect.mesure("OWENServerDatarequest qty=3"))
print(apmconnect.mesure("OWENServerStart val=TRUE"))
print(apmconnect.mesure("OWENServerThread"))
…..
print(apmconnect.mesure("OWENServerStart val=FALSE"))
print(apmconnect.mesure("OWENServerDataRecv"))
print(apmconnect.mesure("OWENSTATE state=STOPPING"))
print(apmconnect.mesure("OWENSTATE state=ENDING"))
print(apmconnect.mesure("OWENSTATE state=WAITING"))
```

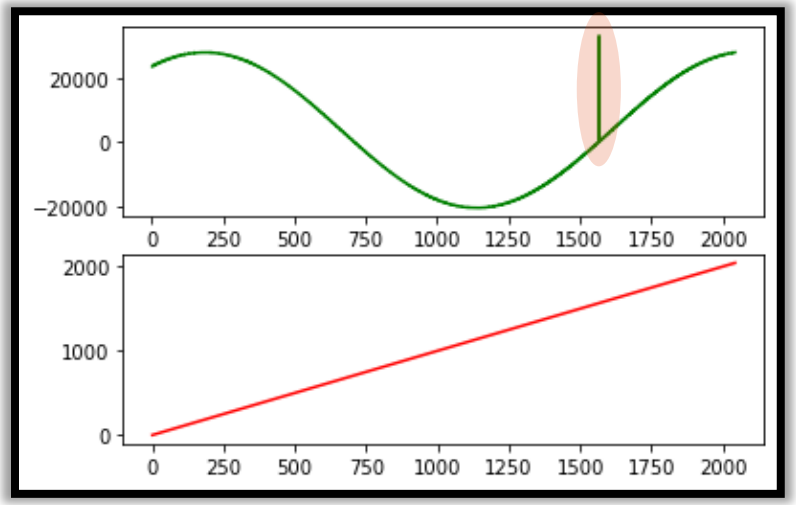
Résultats préliminaires:

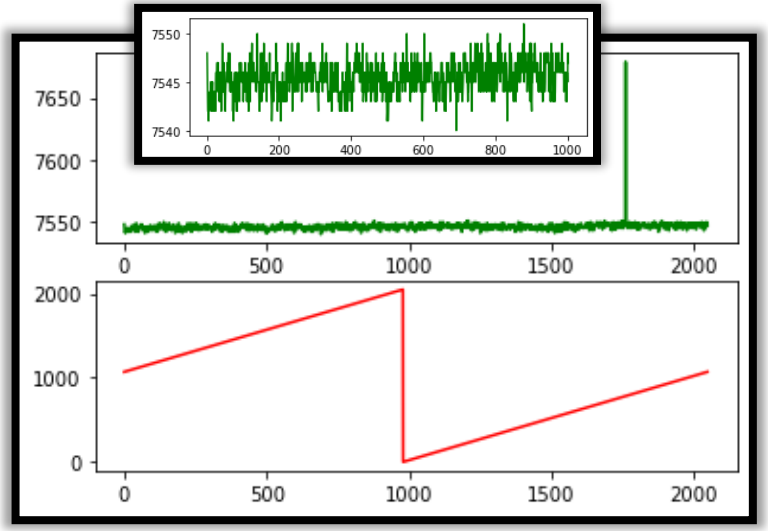


Signal échantillonné



Vérification de la cohérence des données avec indexes (2048 ech.) des échantillons reçus

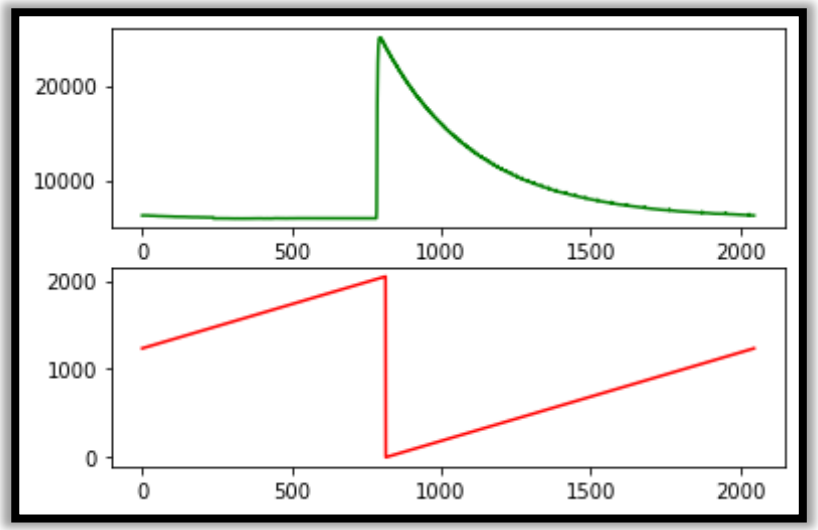


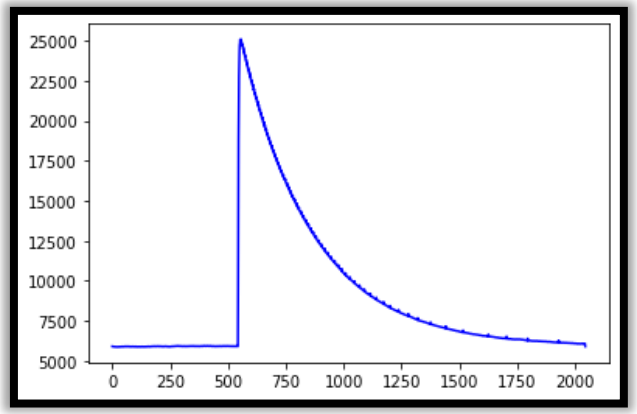


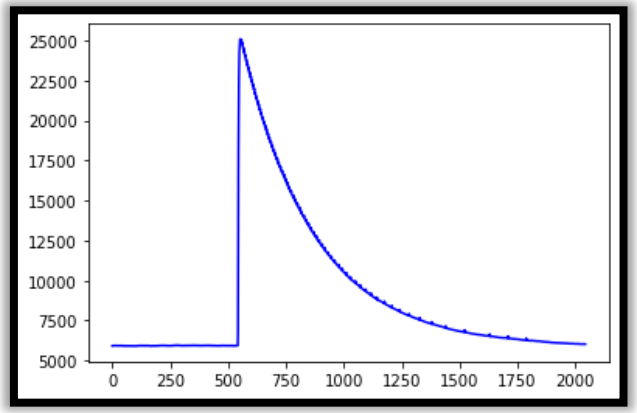
Bruit: +/- 3 LSB

Résultats préliminaires:

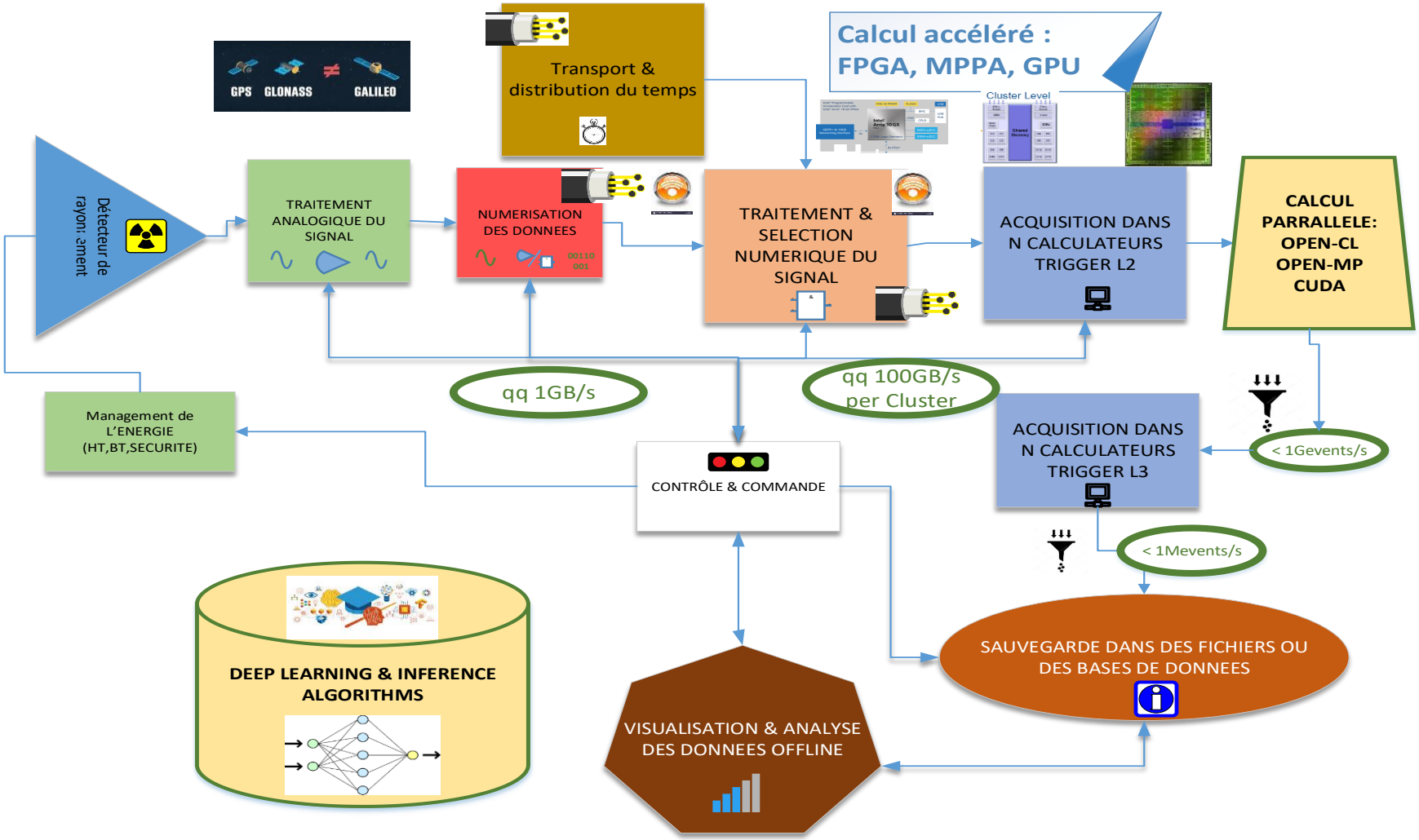
Signal avec PAC + test d'injection au générateur







Les Technologies émergentes pour les DAQ



Les points de rupture :

- Transmission Optique et RF des FE
- Distribution d'horloges précises (qq ps)
- Calcul accéléré et massivement parallèle
- Algorithmes IA (*embarqués*)