# OBELICS Task 3.2 – LAPP contribution

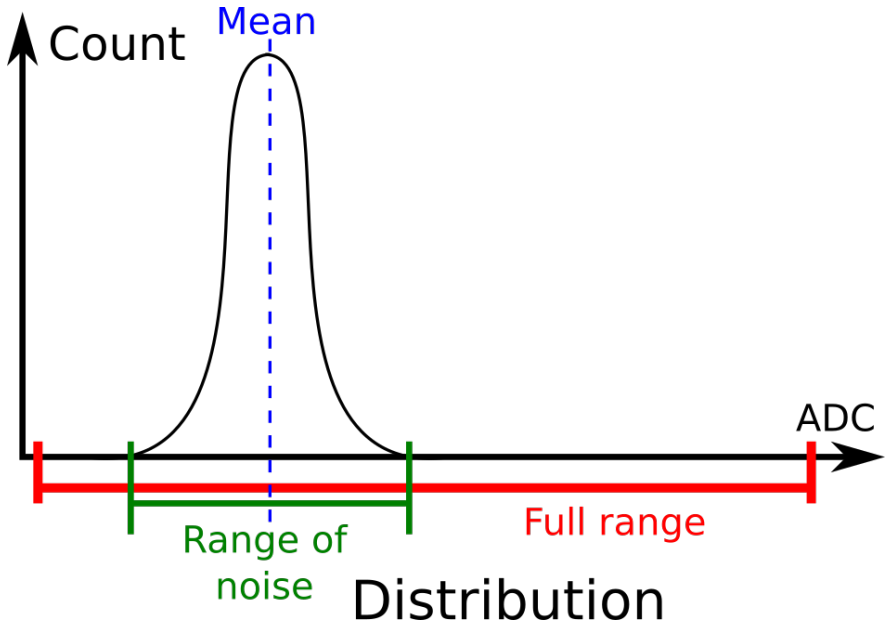
Thomas Vuillaume, on behalf of LAPP

{Pierre Aubert, Thomas Vuillaume, Jean Jacquemier, Gilles Maurin, Armand Fiasson, Giovanni Lamanna}

# Outline

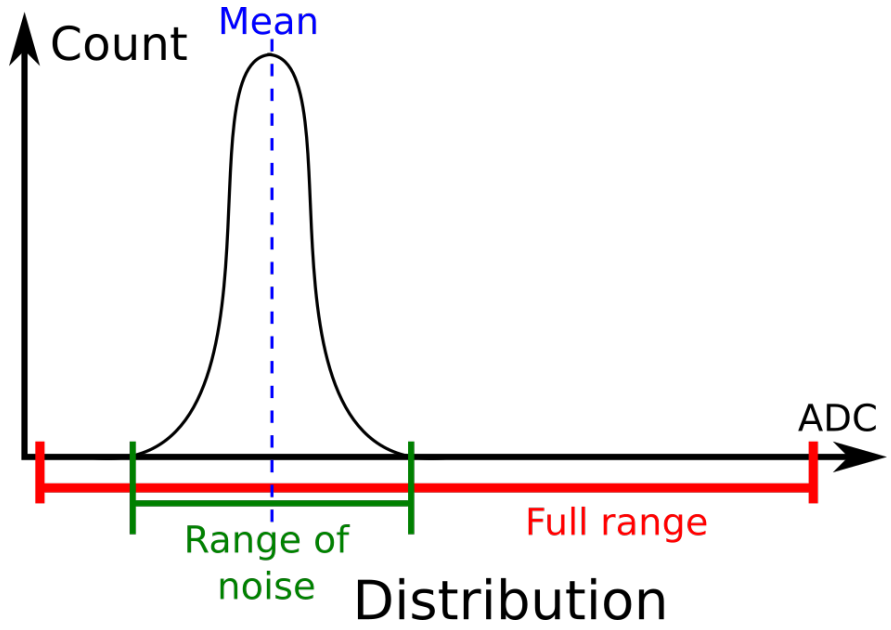
# Data lossless compression

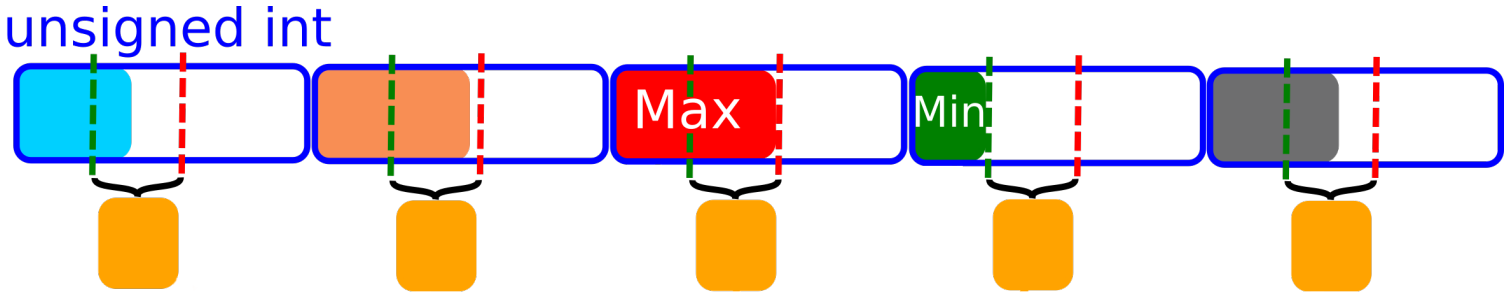
Digitalization : 16 - 24 bits
short : 16 bits
unsigned int 32 bits



# Data lossless compression

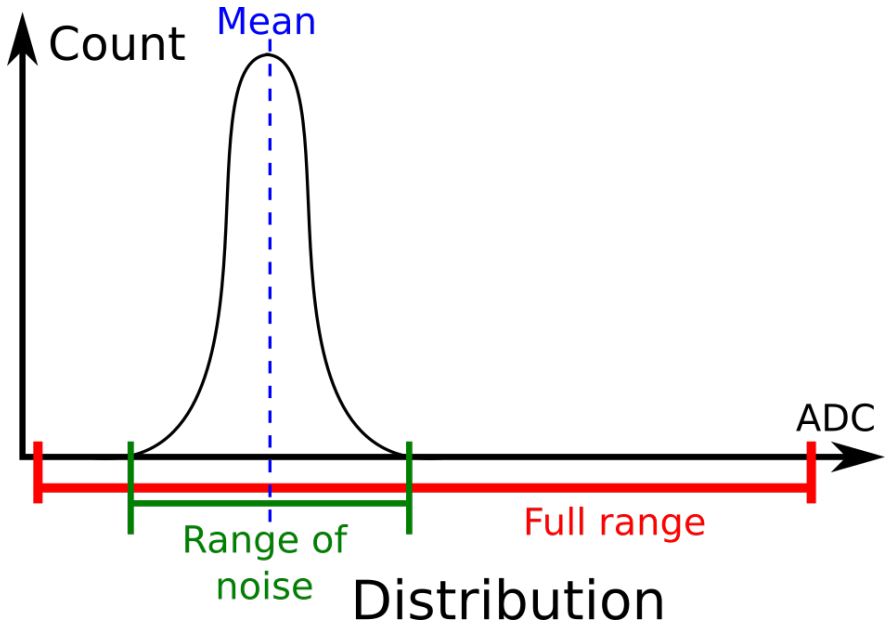
Digitalization : 16 - 24 bits
short : 16 bits
unsigned int 32 bits

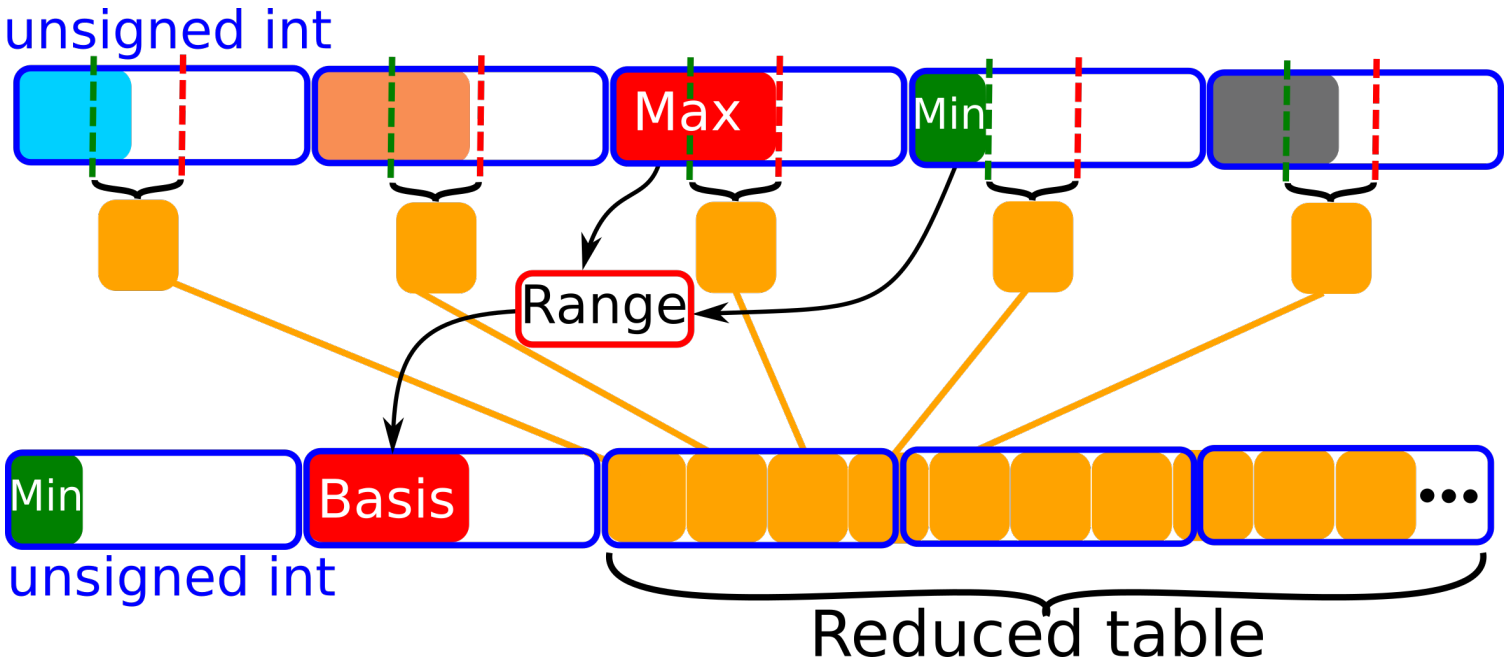


# Data lossless compression

Digitalization : 16 - 24 bits
short : 16 bits
unsigned int 32 bits



# Raw data Compression Ratio and speed (Lossless compression)

## Test file run 497 (Paranal, Gamma Monte-Carlo CTA PROD_3)

- 475 MB
- Up to 99% ADC values

| | Compression ratio | Time | File size (MB) |
|---|---|---|---|
| LZMA (7z) | 4.84 | 7 min 48.636 s | 98 |
| Advanced Polynomial Compression | 3.74 | 3.7 s | 127 |

# Raw data Compression Ratio and speed (Lossless compression)

## Test file run 497 (Paranal, Gamma Monte-Carlo CTA PROD_3)

- 475 MB
- Up to 99% ADC values

| | Compression ratio | Time | File size (MB) |
|---|---|---|---|
| LZMA (7z) | 4.84 | 7 min 48.636 s | 98 |
| Advanced Polynomial Compression | 3.74 | 3.7 s | 127 |
| **Advanced Polynomial Compression + LZMA** | **4.84** | **24.646 s** | **98** |

Same ratio but faster ($\times 19$) compression !

- Vectorizable : possible faster reduction
- Can be used for any data format

# Vectorizable ?

## CPU Recent Architectures

| Architecture | Instruction Set | CPU | Nb float Computed at the same time |
|---|---|---|---|
| SSE4 | 2006 | 2007 | 4 |
| AVX | 2008 | 2011 | 8 |
| AVX 512 | 2013 | 2016 | 16 |

Easy adaptation for coming architectures

# Data format

## What is good for an HPC data format ?

- Allows CPU data prefetching
  - With contiguous data
  - Guaranty data locality
  - Cache friendly
- Allows vectorization
  - With aligned data for vectorizable computation
- Must be as simple as possible (user friendly)
  - Simple types (float, int, double, unsigned int, ...)
  - Tables
  - Allows the simpler solution for complex problem

# LAPP Sorted HPC Autotuned Data Optimized Kernel format

## Generated Data format for CTA

Describes :

- All the telescopes
- All the Cameras
- The CTA PROD_3 Monte-Carlo
- Allows Layouts and Masks
- DL0, DL1, DL2

Allows :

- Allows CPU data prefetching
  - With contiguous data
  - Guaranty data locality
  - Cache friendly
- Vectorization of algorithms
  - Calibration, Cleaning, Hillas

And it's user friendly

# Configuration file example

```
#include "PRecoQuality.h"
#include "PRunHeader.h"

///@brief Describe a reconstructed event
PRecoEvent{
	///Id of the event
	size_t eventId;
	///Timestant of the event
	double timeStamp;
	///Azimuth of the event
	float angleAzimuth;
	///Altitude of the event
	float angleAltidude;
	///High of first interation
	float depthFirstInteration;
	///Recontructed energy
	float energy;
	///Impact parameter on x axis
	float coreX;
	///Impact parameter on y axis
	float coreY;
	///Describe the quality of the event (GOOD) or at which time the telescope has been rejected
	PRecoQuality::PRecoQuality recoQuality;
}

///@brief Describe a file of reconstructed events
PTableRecoEvent{
	///Header of the file
	PRunHeader header;
	///Table of all reconstructed events
	Table(PRecoEvent, nbEvent) tabEvent;
}
```

# Conclusion

## Summary

- Data compression (4.84, 19× faster)
- Data format generator

https://gitlab.in2p3.fr/CTA-LAPP/PLIBS_8

# Backups

# CPU Architecture

# CPU Architecture

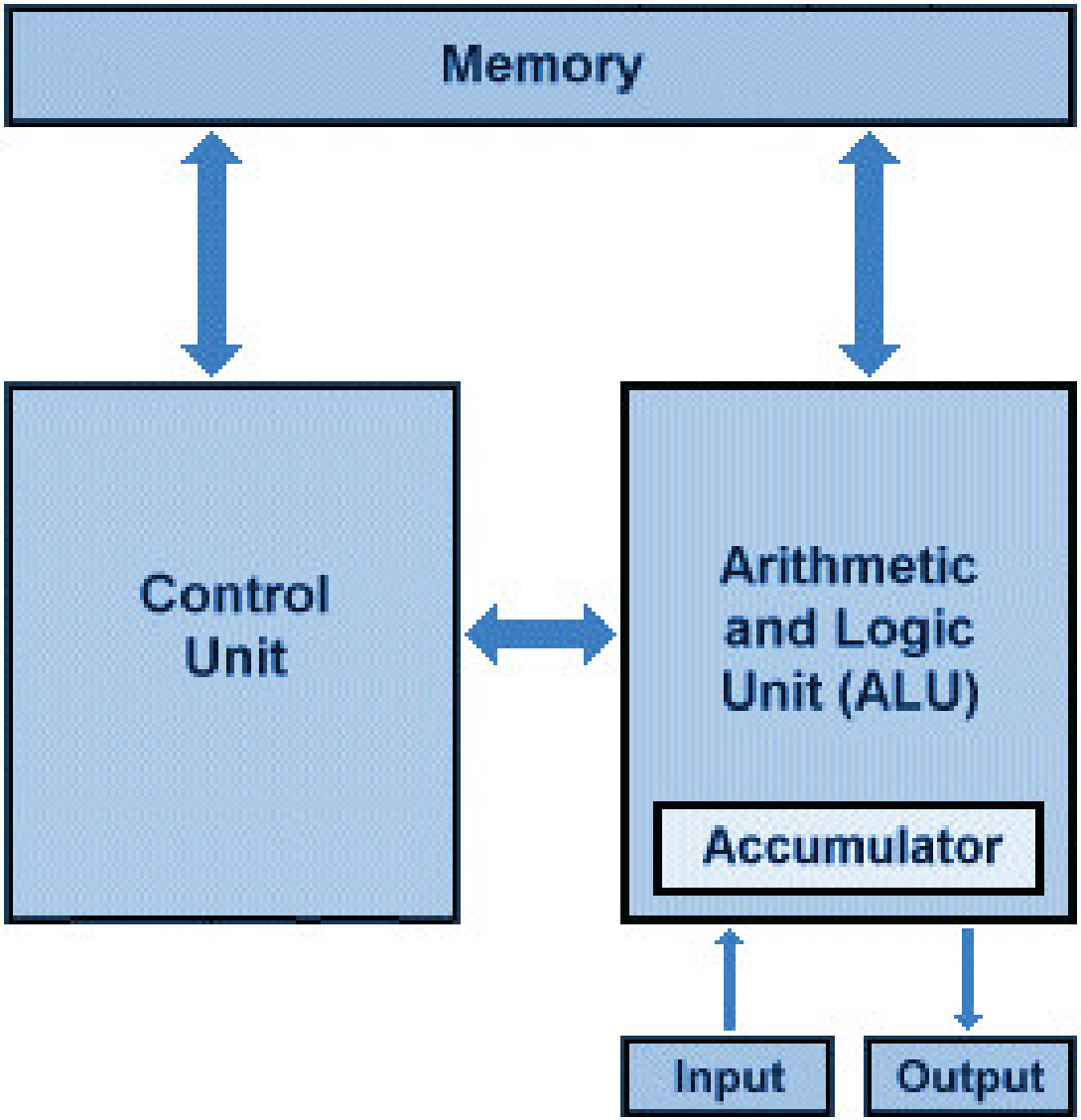
## Von Neumann architecture 1945

### Definition

Cycle : basis unit of time in a CPU

### Time

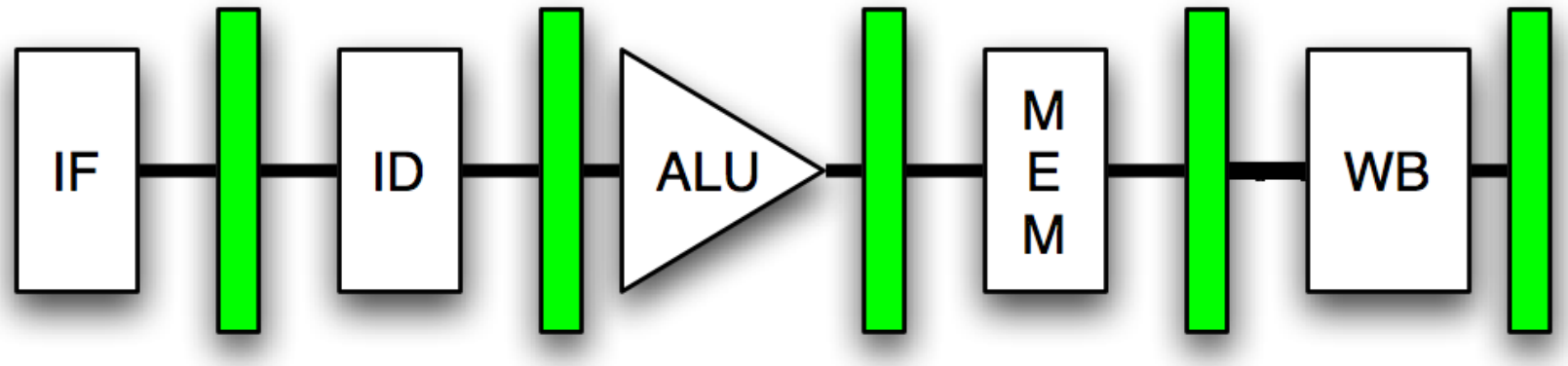
- 1 cycle per elementary operation (load, store, add, ...)
- 4 cycles per whole operation ($c = a + b$)



# CPU Architecture

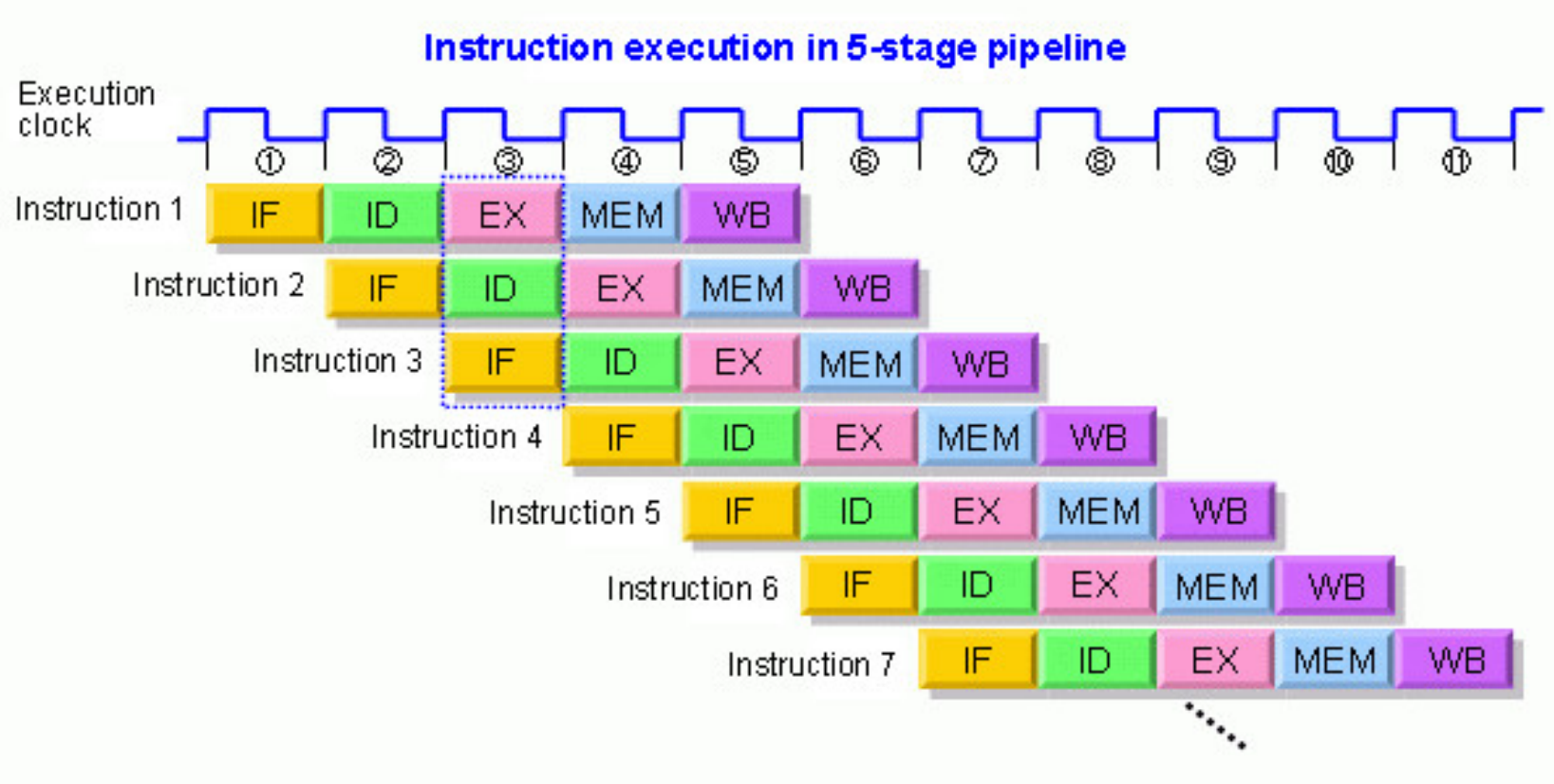
## Pipeline approach

- IF : Instruction Fetch
- ID : Instruction Decode
- ALU : Execution
- MEM : Memory
- WB : Write Bytes



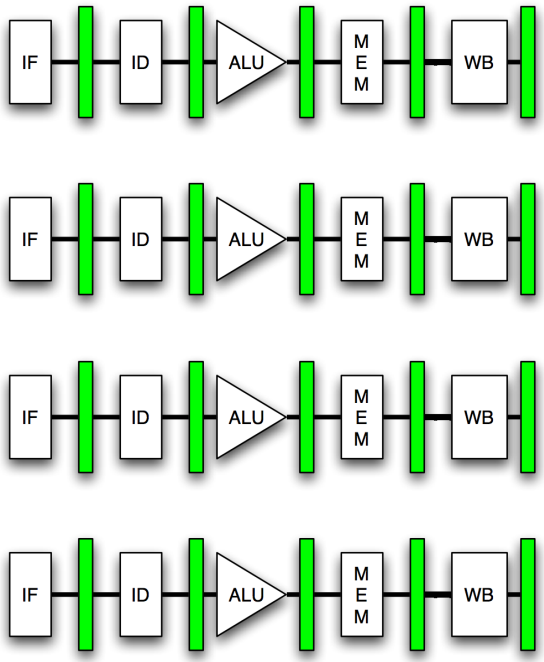
# CPU Architecture evolution

## Pipeline using

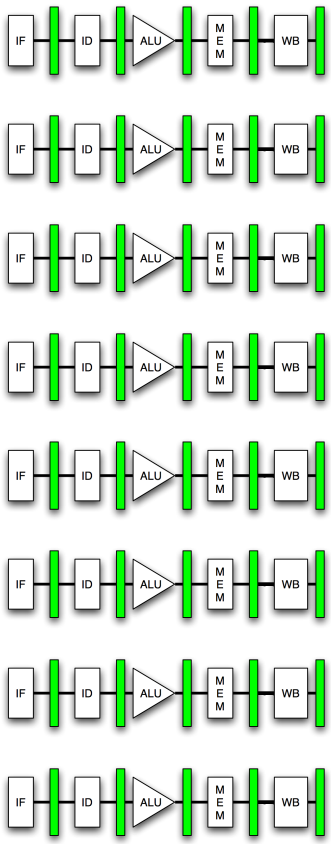


# CPU Recent Architectures

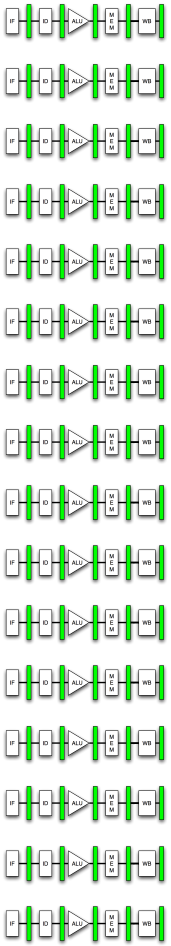
SSE4 4 floats
Instruction set : 2006
CPU : 2007




AVX 8 floats
Instruction set : 2008
CPU : 2011




AVX 512 16 floats
Instruction set : 2013
CPU : 2016



**Data format**

Efficent only if data are contiguous