



# Présentation

Intervenant

Auditoire

# Plan

- Domaines d'application
- Le FPGA
- Les pièges
- Cas concrets d'utilisation
- Outils
- Conclusions / Questions

# Domaines d'application

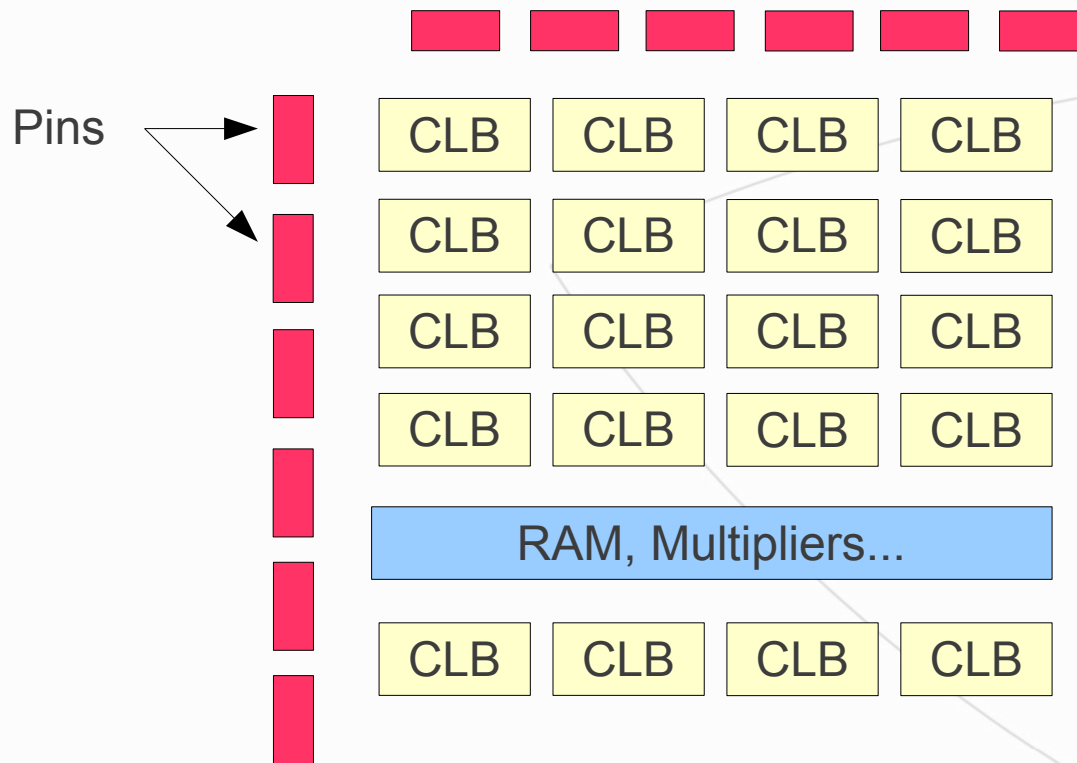
- Traitement de signal  
Radar, médical, vidéo
- Contrôle / commande  
Asservissement, contrôle moteur
- Gestion de trafic réseau
- Finance

# Le FPGA

- Composant électronique
- Différent d'un processeur et d'un DSP
- Machine parallèle par nature
- Synchrones et déterministe

\* IP: Intellectual property

# Le FPGA: structure



CLB: Configurable Logical Blocks  
(ALU, Combinatorial functions,  
registers...)

Tous les blocs sont  
interconnectés par différents  
types de bus

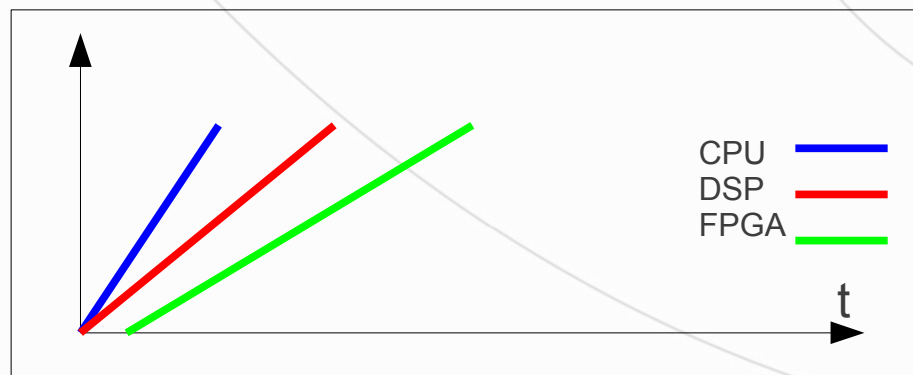
\*FPGA: Field Programable Gate Array

# Le FPGA: grandes caractéristiques

- Jusqu'à 2M de CLB
- Jusqu'à 5000 GMACS (FIR symétrique)
- Jusqu'à 68Mbits de ram interne
- Fréquence de fonctionnement lente comparée aux CPU / DSP
- Supporte des Mémoires externes
- Bus rapides pour l'interfaçage à des ADC/DACs/FPGA
- Dispose de peu de périphériques

# Les pièges (1)

- FPGA = processeur
  - Revoir les algorithmes développés sur CPU / DSP
- Simple à mettre en oeuvre



# Les pièges (2)

- Traitement flottant  $\Rightarrow$  à proscrire
  - Gestion des erreurs d'arrondi
- Code asynchrone
- Développement de périphériques complexes (ethernet, USB etc)



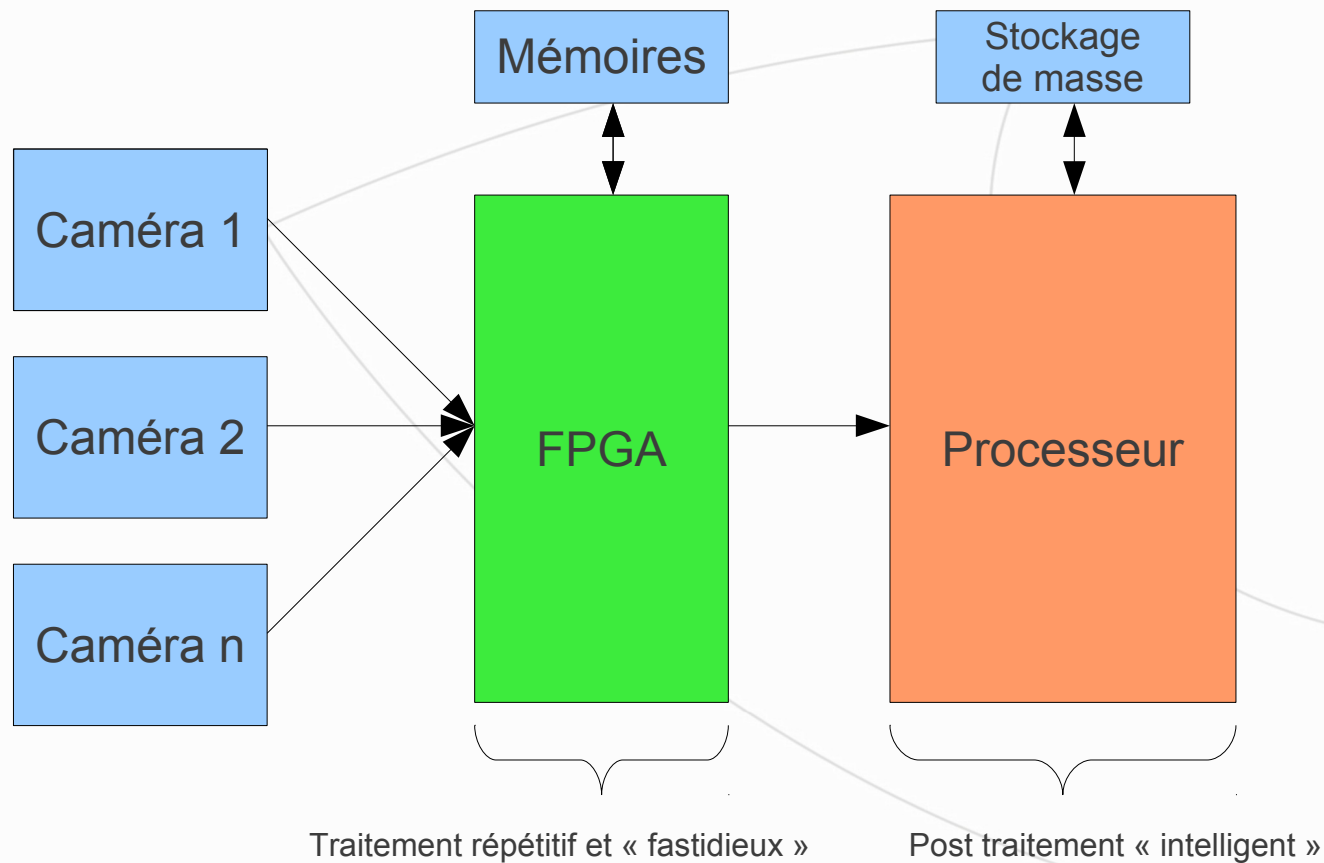
# Les pièges (3)

- Développement “Quick & Dirty”
  - **Simulation obligatoire**
- Développement “tout en un”
  - Penser fonctionnel
- Aspects parallèle et temporels
  - Pas naturel pour les développeurs

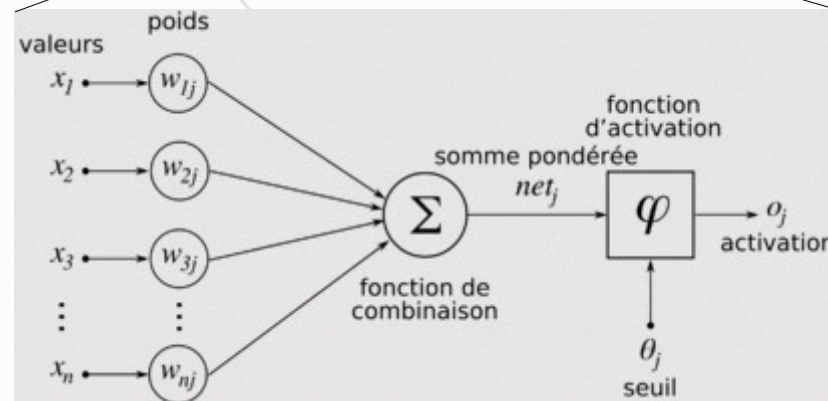
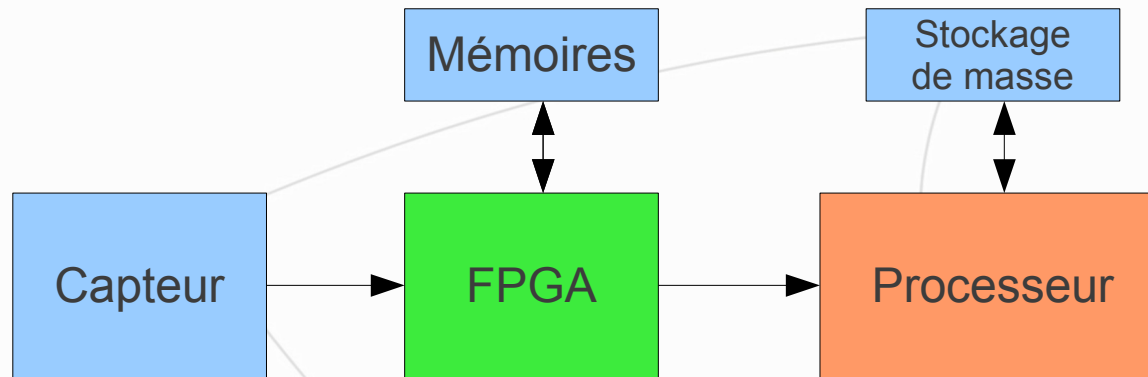
# Cas concrets d'utilisation

- Calcul temps réel dur
- Systèmes embarqués
- Contrôle commande rapide / critique

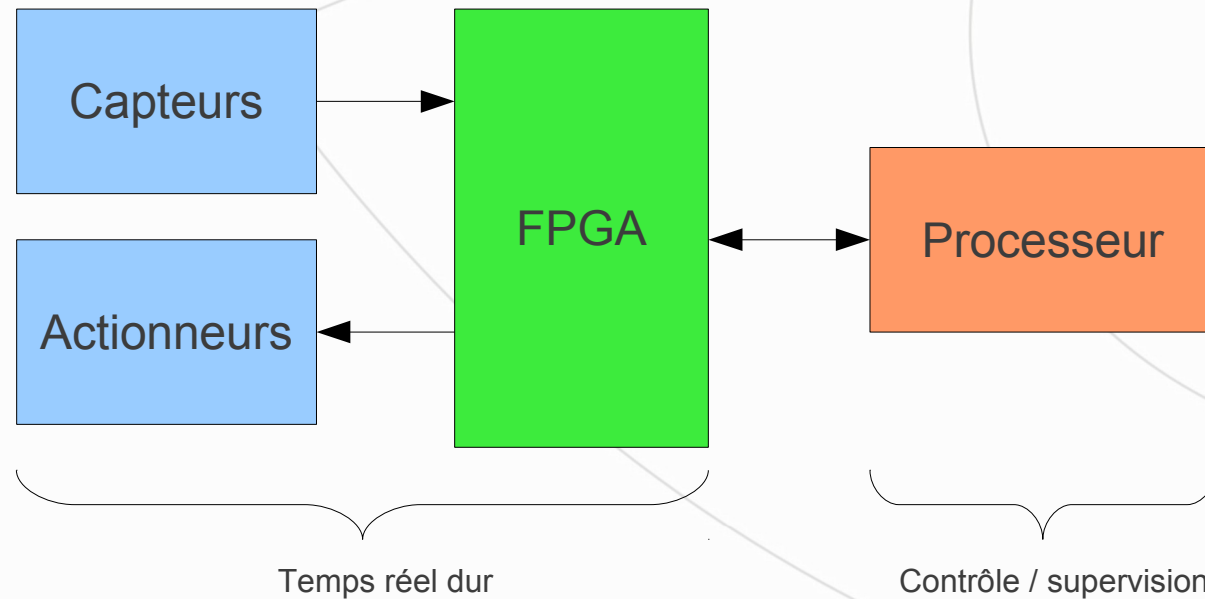
# Cas concrets d'utilisation: Traitement vidéo



# Cas concrets d'utilisation: Réseaux Neuronaux



# Cas concrets d'utilisation: Contrôle / Commande



# Outils

## Synthèse, placement, routage

- Langages spécifiques mais normalisés (VHDL, Verilog, System C...)
- Outils des fabricants de FPGA (Xilinx, Altera...)
- Outils open source: GHDL, gEDA
- Outils haut de gammes de sociétés spécialisés (ex: Mentor Graphics)

## Générateurs de code, RAD

- Matlab (HDLCoder)
- National Instruments (NI Labview FPGA)
- Open source ?

# Conclusion

- Adapté pour les systèmes embarqués nécessitant des temps de réponses déterminés et rapides (traitement en flux, contrôle/commande)
- Flexible et évolutif (cluster, chaînage)
- Outils encore à perfectionner pour simplifier et accélérer le développement

Questions ?