

# ASModee Asynchronous RISC V Core

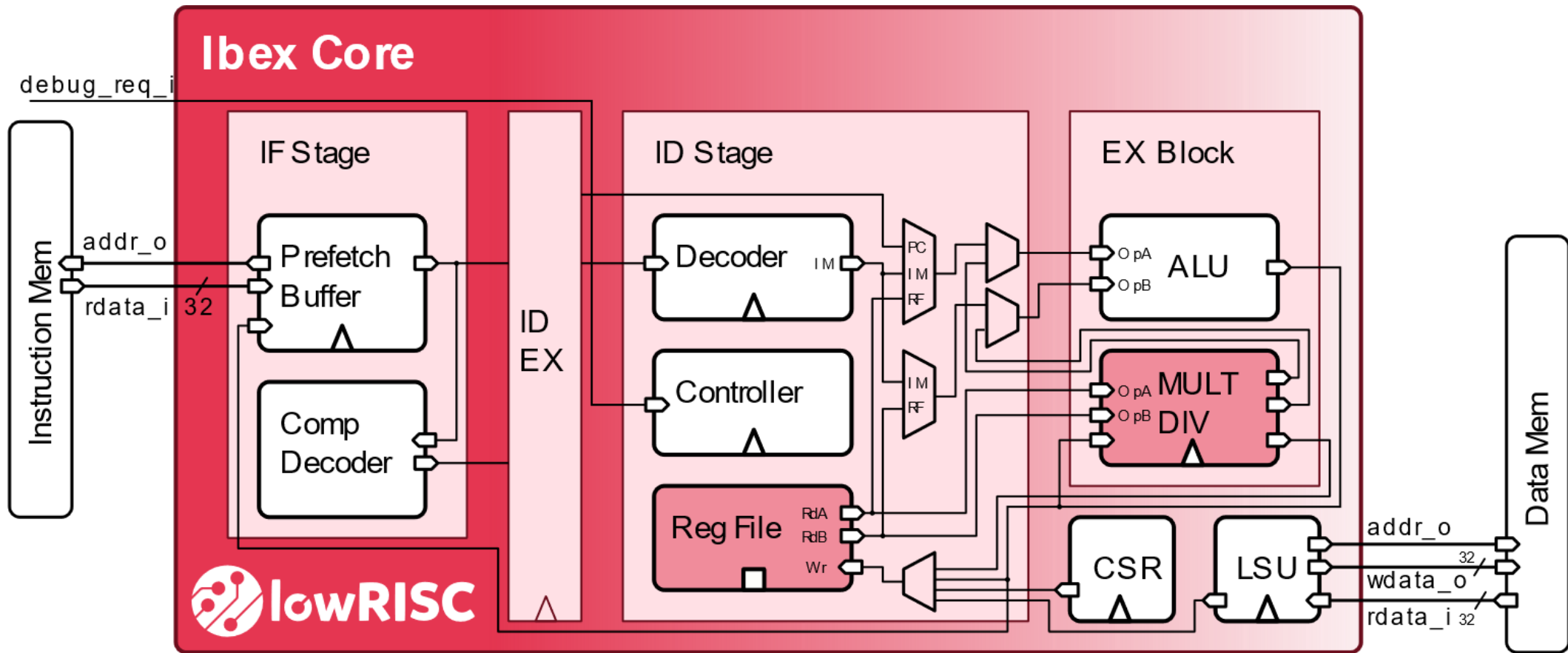
---

THIRION NATHAN, LESAGE XAVIER

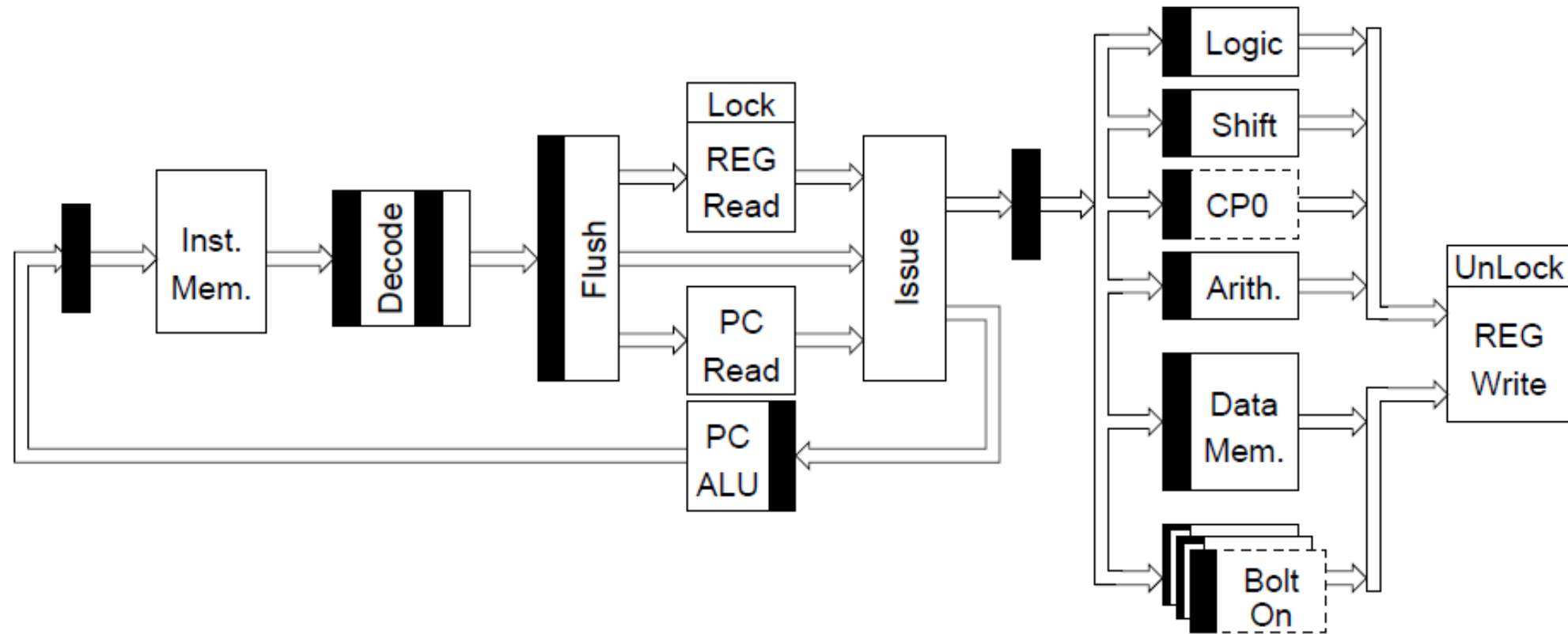
04/05/2020



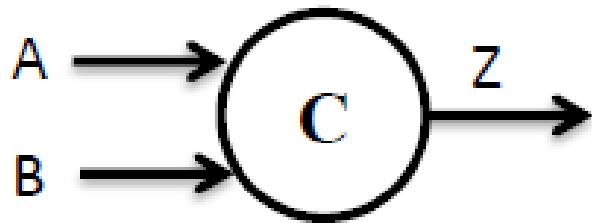
# RISC V - Ibex



# ARISC – Jens Sparsø

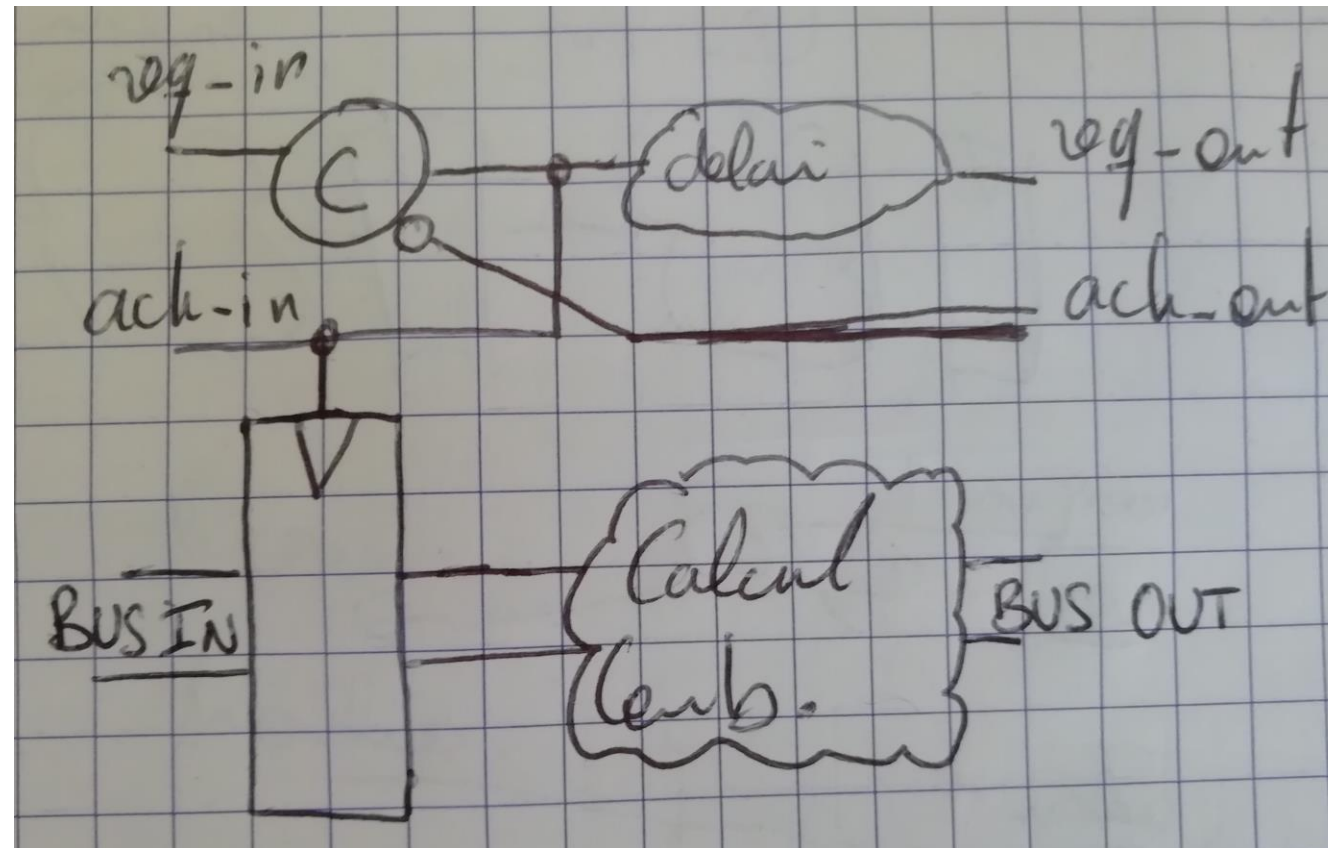


# Muller's gate

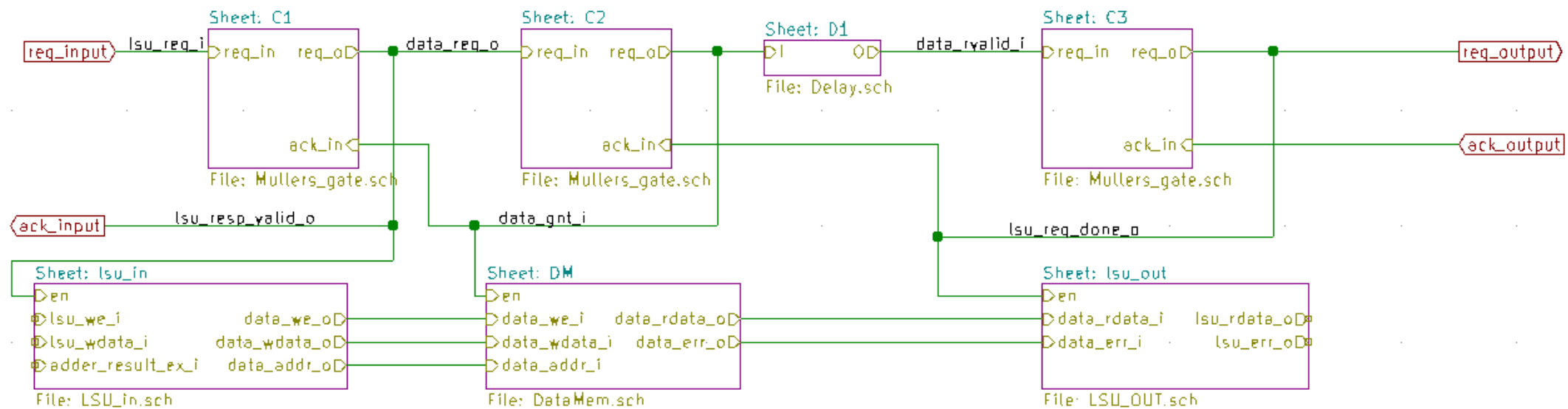


F	R	Z
0	0	0
0	1	<i>Memory</i>
1	0	<i>Memory</i>
1	1	1

# Architecture d'un bloc



# Asynchrone - Handshake



# Flot et Méthodologie

---

Conception de Blocs combinatoires fonctionnels

- vérification en banc de test

Actualisation des données en entrée sur un signal enable (cf Architecture d'un Bloc)

Conception d'un contrôleur asynchrone universelle (réglage des délais simple)

- Forks et Joins adaptées

Mise en place du circuit de requête asynchrone et de l'initialisation

Connection des blocs au contrôleur asynchrone

Définition d'une mémoire d'instruction pour tester le core

# Résultats & Perspectives

---

POF:

Architecture fonctionnelle et adaptable

Exécution de programme assembleur simple

Gestion des exceptions

Différents modes d'exécution (Debug)

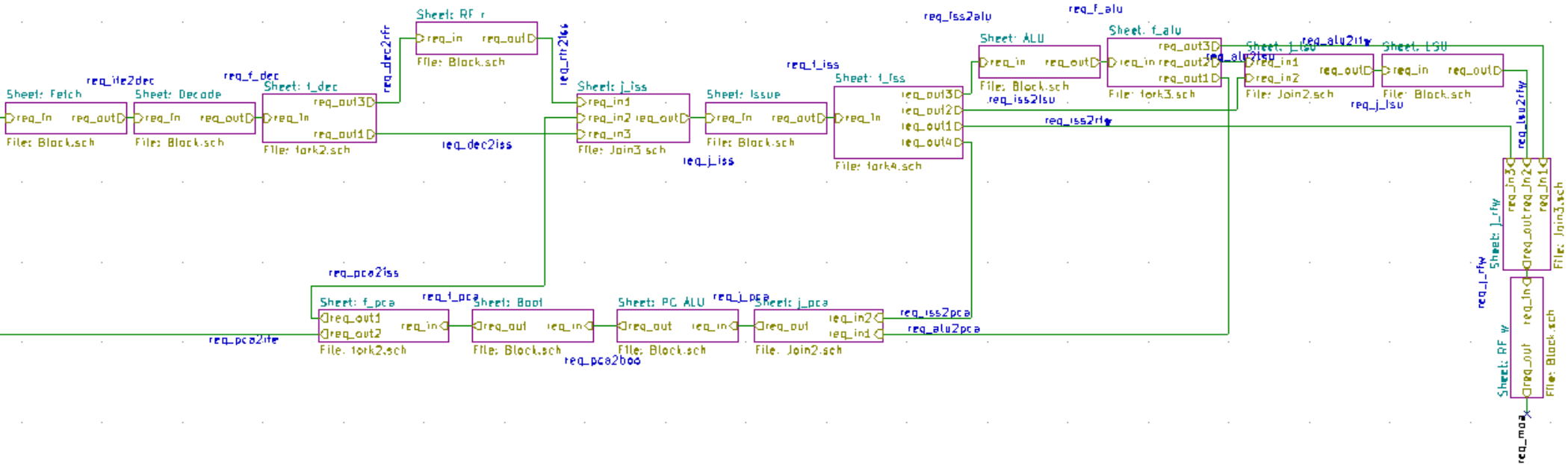
Gestion des CSR

Ajout de fonctionnalités supplémentaires (p.  
ex. Mult/Div)

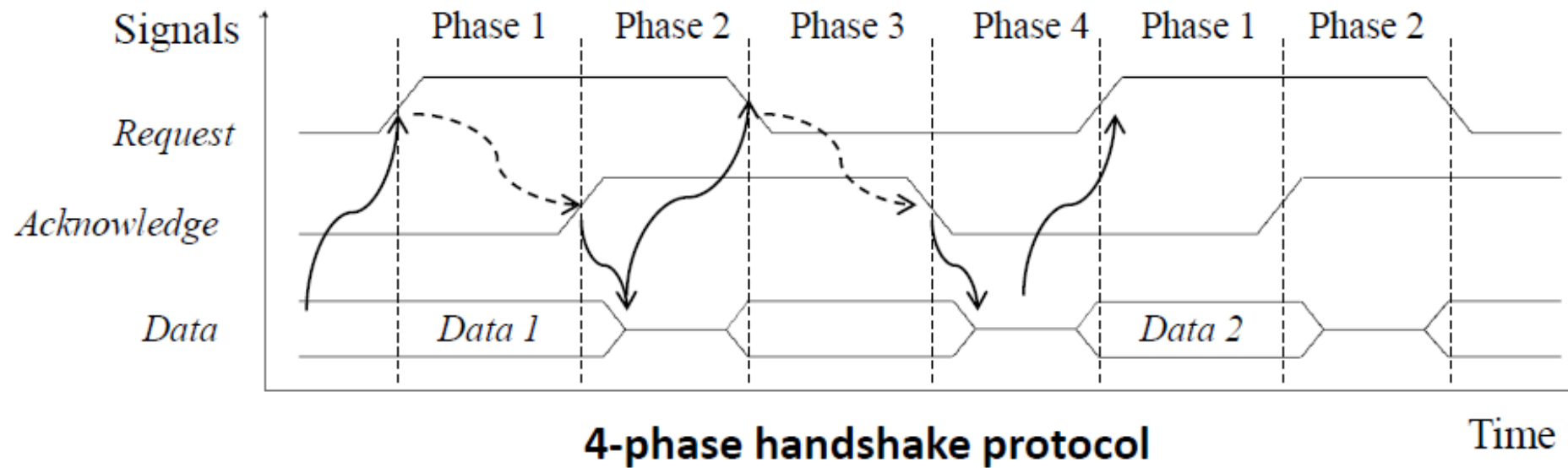
Revoir Fork et Join en Split et Merge



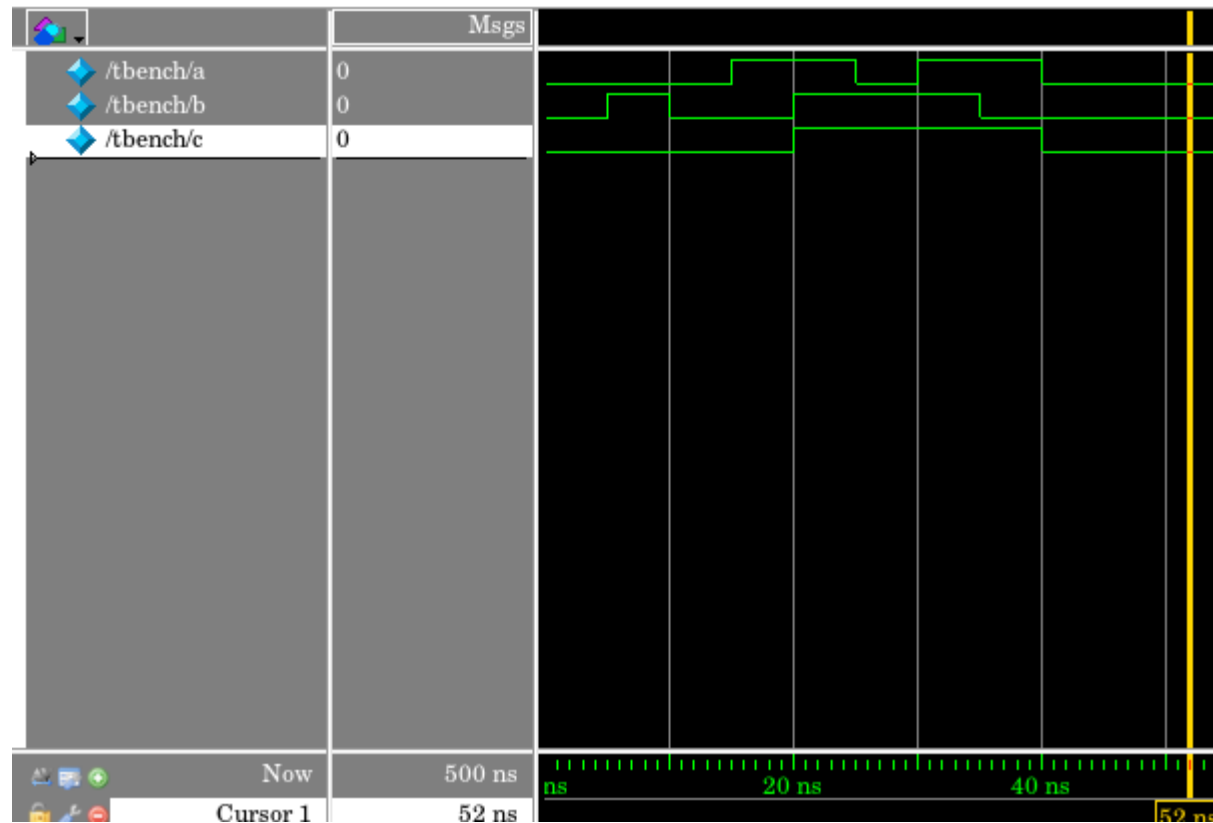
# Conclusion



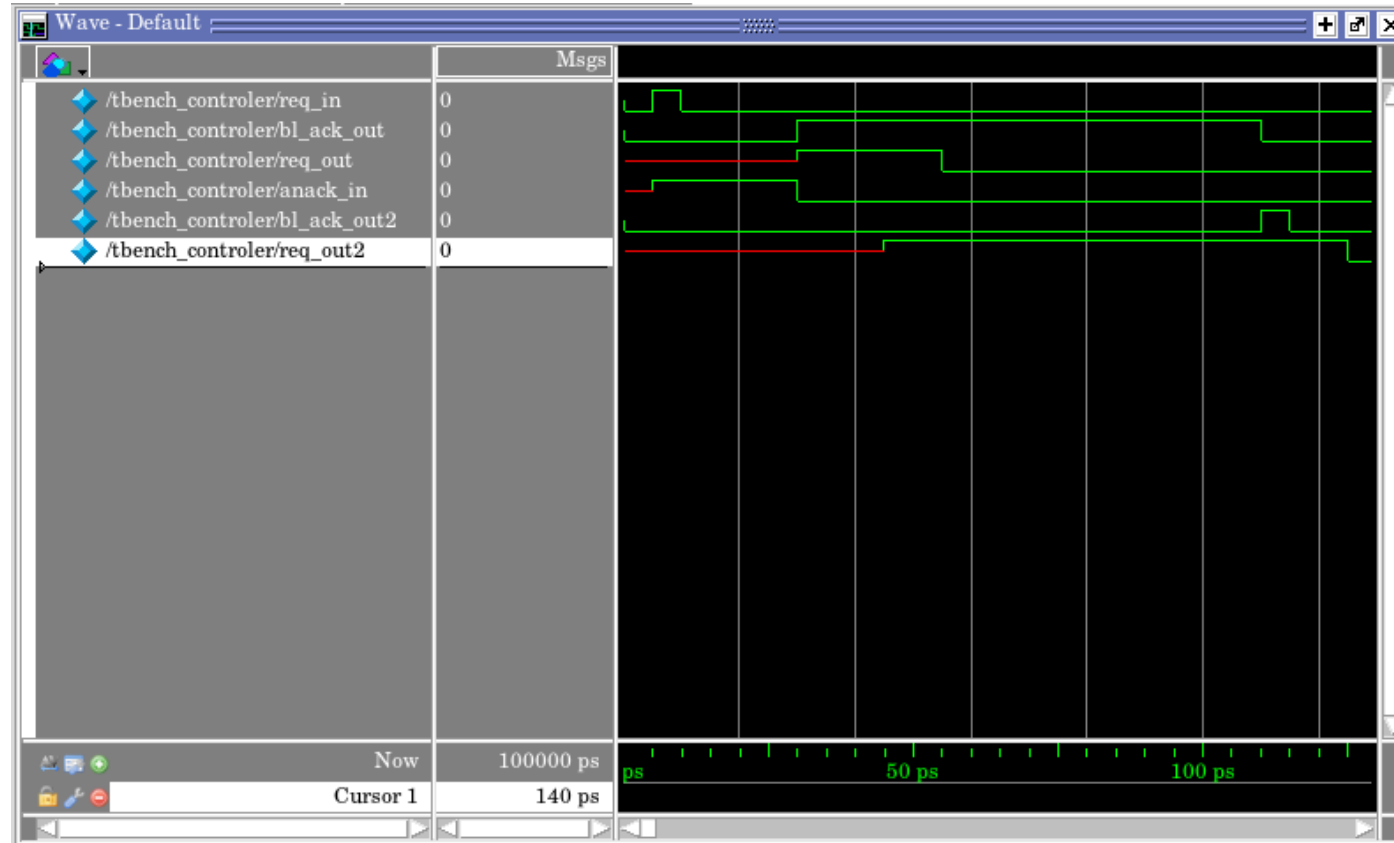
# Annexe A: 4-Phase Handshake



# Annexe B: Chronogramme C-element



# Annexe C: Chronogramme 2 controleurs



# Annexe D: Chronogramme contrôleur et BUS

