

PROCESSEUR 16 BITS

Architecture des ordinateurs – Université CY Cergy Paris

21 Avril 2020

Equipe: MAJDALANI Tatiana, TLEMCANI Ramy

SOMMAIRE

1. Le projet et ses objectifs
2. Description des éléments réalisés
3. Exemple du fonctionnement de l'exécution d'une instruction
4. Extensions apportées au processeur
5. Conclusion

Le projet et ses objectifs

- **Le projet :**

Processeur 16 bits conçu en langage VHDL grâce au logiciel Quartus Prime Lite 18.1.

- **Les objectifs :**

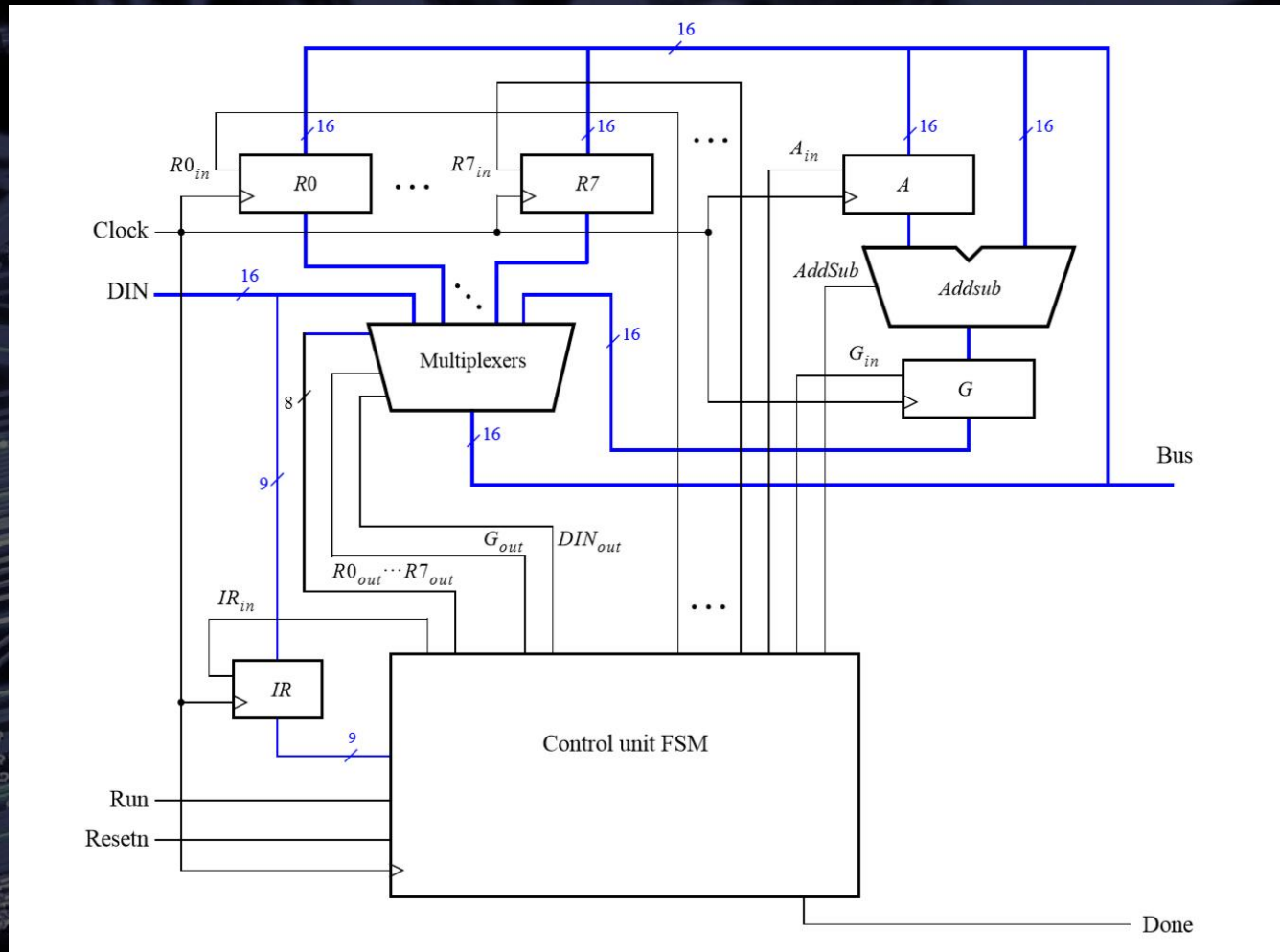
1. Implémentation modules nécessaires
2. Réalisation ensemble des connexions entre modules
3. Configuration signaux de commande
4. Encodage de l'instruction
5. Réalisation protocole de test du processeur.
6. Implémentation de l'extension
7. Faire fonctionner processeur avec réalisations de calculs et rangement des données dans registres.

Description des éléments réalisés

Notre processeur est composé de :

Modules nécessaires	Fonctions de ces modules
FullAdder	Permet de faire l'additionneur
Additionneur/soustracteur	Faire les opérations : + et -
RegNBits (composant générique sur N bits)	Utile pour la conception des différents registres de notre processeur.
Registre généraux R0...R15 (sur 16 bits chacun)	Ils servent à stocker des données issues de la mémoire et des résultats des opérations effectuées par l'ALU
Registre A (sur 16bits)	Registre accumulateur permettant le stockage de la donnée de l'ALU.
Registre RG (sur 16 bits)	Registre servant à prendre en mémoire le résultat.
Registre d'instruction RI (sur 12 bits)	Permet la récupération et la sauvegarde des instructions.
Multiplexeur principal 32 vers 1	Permet le transfert de données d'un registre vers un autre ,il sert à choisir la donnée qui circulera sur le bus.
Unité arithmétique et logique	Effectue des calculs arithmétiques et logiques.
Unité de contrôle (FSM)	Décode l'instruction, permet d' émettre des signaux de contrôle pour réaliser l'instruction.

Exemple du fonctionnement de l'exécution d'une instruction



Add R0,R1
 $R0 \leftarrow [R0] + [R1]$

Extensions apportées au processeur

✓ Processeur 16 Bits

➤ Tableau d'instruction du processeur :

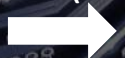
Opération	Code opération
mv Rx,Ry	0100
mvi Rx, #DI	0101
Add Rx,Ry	1000
Sub Rx,Ry	1001
Mul Rx,Ry (non faite)	1010

➤ Ajouts d'instructions :

Opération	Code opération
Jump	0010
Load	0110
store	0111
OR	1100
AND	1101
NOT	1110

➤ Améliorations du processeur:

- Passage nombre registres généraux à 16 (R0...R15) au lieu de 8



processeur plus puissant

- Nombre de bit entrant dans FSM monté à 12



plus d'instructions possibles.

Conclusion

➤ Ce qui marche:

- Manipuler des données stockées dans des registres mémoire de 16 bits
- Le calcul d'opération logique ou arithmétique sur des données
- Implémentation d'extensions (Jump, Load, Store, AND, OR, NOT)
- Ajout ROM réussi

➤ Perspectives:

- Passage du processeur à 32 ou 64 bits
- Utilisation des nombres flottants
- Ajout RAM
- Faire fonctionner extension Jump, Load, Store
- Résoudre bug de la FSM afin qu'elle prenne en compte les signaux RUN et reset.