
Simulateur de processeur

Projet de prog5 L3 Informatique

I) Organisation

A) Le travail en équipe

Le groupe est composée de :

- Vizzini Jeremy
- Finiel Jérémie
- Souchet Julie
- Heckmann Victor.

Dès le premier jour, nous avons partagé nos informations personnelles et nous avons pris la décision d'utiliser Git et Github pour partager notre code. Nous avons fait le choix de ne pas travailler à distance mais de se rencontrer chaque jour dans les bâtiments de la fac pour pouvoir plus facilement se parler.

B) Organisation des taches

Le travail a été globalement réparti par fichiers :

- arm_instruction.c : tous
- arm_data_processing.c : Julie
- arm_branch_others.c : Jérémie
- arm_load_store.c & memory.c & arm_exception.c : Jérémy
- tests pour toutes les instructions : Victor

Lors du débogage , nous avons cependant amené à travailler en dehors de nos taches respectives.

II) Structure du code

A) Organisations des fichiers

Nous avons suivi le squelette fourni. Nous avons ni enlevé ni rajouté de fichier. Certains fichiers fournis comme arm_core.c, arm_constants.c ont comme même été modifié pour certains ajouts (ajout des exceptions, fonctions d'accès facilités pour l'accès aux flags, nouvelles constantes, etc)

B) Fonctionnalités ajoutées

Fichiers	Ajouts
arm_branch_others	<ul style="list-style-type: none">• Branchements B, BL• MRS, MSR
arm_constant	<ul style="list-style-type: none">• Constantes registres (PC, LR, SP)• Bits de CPSR (E, T, ...)• NO_EXCEPTION
arm_core.c	<ul style="list-style-type: none">• Fonctions d'accès aux flags• Prise en charge des exceptions
arm_data_processing	<ul style="list-style-type: none">• AND, EOR, SUB, RSB, ADD, ADC, SBC, RSC, TST, TEQ, CMP, CMN, ORR, MOV, BIC, MVN
arm_exception	<ul style="list-style-type: none">• Traitements des exceptions
arm_instruction	<ul style="list-style-type: none">• Exception UNDEFINED_INSTRUCTION• Traitement du champ de conditions• Redirection des instructions vers les fichiers correspondants
arm_load_store	<ul style="list-style-type: none">• LDR / STR {S/B/T/D/H/ SH}• LDM / STM {1, 2, 3}
memory	<ul style="list-style-type: none">• lecture / écriture {byte / half / word}
util	<ul style="list-style-type: none">• Macros de manipulation bits à bits

C) Ecriture des tests

Les tests ne sont pas automatisés et nous avons écrit des fichiers exemples pour chaque instruction (et leurs variantes).

Voici la liste des fichiers tests :

- example1.s
- example2.s
- example3.s
- example4.s
- exampleB_BL.s
- exampleHandler.s
- exampleLDM_STM.s
- exampleLDRB.s
- exampleLDRH.s
- exampleLDR.s
- exampleLDRSH_LDRSB_LDRD_STRD.s
- exampleMOV_MVN.s
- exampleMRS.s
- exampleMSR.s
- example_operations_bin.s
- example_operations.s
- exampleSTRB.s
- exampleSTRH.s
- exampleSTR.s
- exampleTST_TEQ_CMP_CMN.s

III) Journal des tâches

Jour :	Tâches effectués :
1	<ul style="list-style-type: none">• Prise en main de Git• Prise en main de la structure fournie• memory.c : implémentation des accès mémoire
2	<ul style="list-style-type: none">• Création de tests pour les load et store• arm_instruction.c : Décodage en vérifiant les conditions• arm_branch_others.c : B, BL et MRS• arm_data_processing.c : Début du décodage des valeurs immédiates
3	<ul style="list-style-type: none">• Suite et fin des tests pour les load et store• Nettoyage du git avec .gitignore• arm_load_store.c : Debut de l'implémentation de LDR / STR• arm_data_processing.c : Ajout du calcul de la 2nde opérande• arm_branch_others : Implémentation de MSR
4	<ul style="list-style-type: none">• Création de tests pour les branchements• arm_load_store.c : Implémentation de LDR/STR {B, T}• arm_data_processing.c : ajout de la maj des flags
5	<ul style="list-style-type: none">• Intégration et première compilation et essais• Correction des bugs à l'aide des tests précédemment écrits• arm_load_store.c : factorisation du code
6	<ul style="list-style-type: none">• Ecriture de tests et correction des bugs• Ecriture d'un script pour faciliter l'exécution des tests• Lecture de la documentation pour les exceptions
7	<ul style="list-style-type: none">• Ecriture de tests et correction des bugs• Création de tests pour les exceptions• arm_exception.c : Implémentation des exceptions
8	<ul style="list-style-type: none">• Fin des tests de toutes les instructions et validation
9	<ul style="list-style-type: none">• Création de programme de test pour la soutenance• Ecriture du compte rendu final
10	<ul style="list-style-type: none">• Contrôle écrit et soutenance