

Extension : ARM

gl-24

1. Motivation:

Les téléphones, les ordinateurs, l'IOT...L'architecture ARM est très prépondérante. Ainsi développer un compilateur qui marche dans un très grand nombre de machines nous paraît très intéressant. Cela nous permettra aussi d'apprendre beaucoup plus sur l'architecture et de prendre en considération les nuances entre les différentes machines.

2. Contenu attendu:

- Ajout de l'option [-arm] pour le compilateur decac. Cela permet de compiler des fichiers .deca vers un assembleur compatible avec une machine ARMv7 32 bits
- Paquetage arm.pseudocode et arm.pseudocode.instructions : qui contiennent les classes gérantes des spécificités liées à la mémoire de l'architecture ARM et aux instructions.

ARM	Description
R0	General Purpose
R1-R5	General Purpose
R6-R10	General Purpose
R11 (FP)	Frame Pointer
R12	Intra Procedural Call
R13 (SP)	Stack Pointer
R14 (LR)	Link Register
R15 (PC)	<- Program Counter / Instruction Pointer ->
CPSR	Current Program State Register/Flags

Instruction	Description	Instruction	Description
MOV	Move data	EOR	Bitwise XOR
MVN	Move and negate	LDR	Load
ADD	Addition	STR	Store
SUB	Subtraction	LDM	Load Multiple
MUL	Multiplication	STM	Store Multiple
LSL	Logical Shift Left	PUSH	Push on Stack
LSR	Logical Shift Right	POP	Pop off Stack
ASR	Arithmetic Shift Right	B	Branch
ROR	Rotate Right	BL	Branch with Link
CMP	Compare	BX	Branch and exchange
AND	Bitwise AND	BLX	Branch with Link and exchange
ORR	Bitwise OR	SWI/SVC	System Call

- Exécution :
Pour assembler et lier le fichier nous utiliserons le paquetage gcc-arm-gnueabi et pour exécuter nous utiliserons l'émulateur des systèmes Qemu.
Dans les mesures de possible, nous essayerons d'exécuter le compilateur sur un ordinateur avec ARM comme architecture natif (Un Raspberry Pi par exemple)
- Nous pouvons vous garantir un compilateur ARM pour le sous langage sans objet pour Deca. Mais nous essayerons de faire le maximum possible en fonction du temps pour la partie objet.