Instructions et conditions implémentées

Liste des instructions implémentées

nop, add, adds, mul, muls, sub, subs, sdiv, udiv, mov, movs, mvn, mvns, lsl, lsls, lsr, lsrs, asr, asrs, ror, rors, rrx, rrxs, and, ands, eor, eors, orr, orrs, orn, orns, bic, bics, cmp, cmn, tst, teq, ldr, str

Détails sur la liste des instructions : les différents formats implémentés

nop

add rs, ra, rb	adds rs, ra, rb
add rs, ra, #imm	adds rs, ra, #imm
add rs, ra, lsl, rc	adds rs, ra, lsl, rc
add rs, ra, lsr, rc	adds rs, ra, lsr, rc
add rs, ra, asr, rc	adds rs, ra, asr, rc
add rs, ra, lsl, #imm	adds rs, ra, lsl, #imm
add rs, ra, lsr, #imm	adds rs, ra, lsr, #imm
add rs, ra, asr, #imm	adds rs, ra, asr, #imm
add rs, ra, ror, #imm	adds rs, ra, ror, #imm
add rs, ra, rrx	adds rs, ra, rrx

mul rs, ra, rb	muls rs, ra, rb
mul rs, ra, #imm	muls rs, ra, #imm
mul rs, ra, lsl, rc	muls rs, ra, lsl, rc
mul rs, ra, lsr, rc	muls rs, ra, lsr, rc
mul rs, ra, asr, rc	muls rs, ra, asr, rc
mul rs, ra, lsl, #imm	muls rs, ra, lsl, #imm
mul rs, ra, lsr, #imm	muls rs, ra, lsr, #imm
mul rs, ra, asr, #imm	muls rs, ra, asr, #imm
mul rs, ra, ror, #imm	muls rs, ra, ror, #imm
mul rs, ra, rrx	muls rs, ra, rrx

sub rs, ra, rb	subs rs, ra, rb
sub rs, ra, #imm	subs rs, ra, #imm
sub rs, ra, lsl, rc	subs rs, ra, lsl, rc
sub rs, ra, lsr, rc	subs rs, ra, lsr, rc
sub rs, ra, asr, rc	subs rs, ra, asr, rc
sub rs, ra, lsl, #imm	subs rs, ra, lsl, #imm
sub rs, ra, lsr, #imm	subs rs, ra, lsr, #imm
sub rs, ra, asr, #imm	subs rs, ra, asr, #imm
sub rs, ra, ror, #imm	subs rs, ra, ror, #imm
sub rs, ra, rrx	subs rs, ra, rrx

sdiv rs, ra, rb	udiv rs, ra, rb
mov rs, ra, rb	movs rs, ra, rb
mov rs, ra, #imm	movs rs, ra, #imm
mov rs, ra, lsl, rc	movs rs, ra, lsl, rc
mov rs, ra, lsr, rc	movs rs, ra, lsr, rc
mov rs, ra, asr, rc	movs rs, ra, asr, rc
mov rs, ra, lsl, #imm	movs rs, ra, lsl, #imm
mov rs, ra, lsr, #imm	movs rs, ra, lsr, #imm
mov rs, ra, asr, #imm	movs rs, ra, asr, #imm
mov rs, ra, ror, #imm	movs rs, ra, ror, #imm
mov rs, ra, rrx	movs rs, ra, rrx
mvn rs, ra, rb	mvns rs, ra, rb
mvn rs, ra, #imm	mvns rs, ra, #imm
mvn rs, ra, lsl, rc	mvns rs, ra, lsl, rc
mvn rs, ra, lsr, rc	mvns rs, ra, lsr, rc
mvn rs, ra, asr, rc	mvns rs, ra, asr, rc
mvn rs, ra, lsl, #imm	mvns rs, ra, lsl, #imm
mvn rs, ra, lsr, #imm	mvns rs, ra, lsr, #imm
mvn rs, ra, asr, #imm	mvns rs, ra, asr, #imm
mvn rs, ra, ror, #imm	mvns rs, ra, ror, #imm
mvn rs, ra, rrx	mvns rs, ra, rrx
lsl rs, ra, rb	lsls rs, ra, rb
lsl rs, ra, #imm	lsls rs, ra, #imm
lsr rs, ra, rb	lsrs rs, ra, rb
lsr rs, ra, #imm	lsrs rs, ra, #imm
asr rs, ra, rb	asrs rs, ra, rb
asr rs, ra, #imm	asrs rs, ra, #imm
ror rs, ra, #imm	rors rs, ra, #imm
rrx rs, ra	rrxs rs, ra
and rs, ra, rb	ands rs, ra, rb
and rs, ra, #imm	ands rs, ra, #imm
and rs, ra, lsl, rc	ands rs, ra, lsl, rc
and rs, ra, lsr, rc	ands rs, ra, lsr, rc
and rs, ra, asr, rc	ands rs, ra, asr, rc
and rs, ra, lsl, #imm	ands rs, ra, lsl, #imm
and rs, ra, lsr, #imm	ands rs, ra, lsr, #imm
and rs, ra, asr, #imm	ands rs, ra, asr, #imm
and rs, ra, ror, #imm	ands rs, ra, ror, #imm
and rs, ra, rrx	ands rs, ra, rrx
eor rs, ra, rb	eors rs, ra, rb

eor rs, ra, #imm	eors rs, ra, #imm
eor rs, ra, lsl, rc	eors rs, ra, lsl, rc
eor rs, ra, lsr, rc	eors rs, ra, lsr, rc
eor rs, ra, asr, rc	eors rs, ra, asr, rc
eor rs, ra, lsl, #imm	eors rs, ra, lsl, #imm
eor rs, ra, lsr, #imm	eors rs, ra, lsr, #imm
eor rs, ra, asr, #imm	eors rs, ra, asr, #imm
eor rs, ra, ror, #imm	eors rs, ra, ror, #imm
eor rs, ra, rrx	eors rs, ra, rrx

orr rs, ra, rb	orrs rs, ra, rb
orr rs, ra, #imm	orrs rs, ra, #imm
orr rs, ra, lsl, rc	orrs rs, ra, lsl, rc
orr rs, ra, lsr, rc	orrs rs, ra, lsr, rc
orr rs, ra, asr, rc	orrs rs, ra, asr, rc
orr rs, ra, lsl, #imm	orrs rs, ra, lsl, #imm
orr rs, ra, lsr, #imm	orrs rs, ra, lsr, #imm
orr rs, ra, asr, #imm	orrs rs, ra, asr, #imm
orr rs, ra, ror, #imm	orrs rs, ra, ror, #imm
orr rs, ra, rrx	orrs rs, ra, rrx

orn rs, ra, rb	orns rs, ra, rb
orn rs, ra, #imm	orns rs, ra, #imm
orn rs, ra, lsl, rc	orns rs, ra, lsl, rc
orn rs, ra, lsr, rc	orns rs, ra, lsr, rc
orn rs, ra, asr, rc	orns rs, ra, asr, rc
orn rs, ra, lsl, #imm	orns rs, ra, lsl, #imm
orn rs, ra, lsr, #imm	orns rs, ra, lsr, #imm
orn rs, ra, asr, #imm	orns rs, ra, asr, #imm
orn rs, ra, ror, #imm	orns rs, ra, ror, #imm
orn rs, ra, rrx	orns rs, ra, rrx

bic rs, ra, rb	bics rs, ra, rb
bic rs, ra, #imm	bics rs, ra, #imm
bic rs, ra, lsl, rc	bics rs, ra, lsl, rc
bic rs, ra, lsr, rc	bics rs, ra, lsr, rc
bic rs, ra, asr, rc	bics rs, ra, asr, rc
bic rs, ra, lsl, #imm	bics rs, ra, lsl, #imm
bic rs, ra, lsr, #imm	bics rs, ra, lsr, #imm
bic rs, ra, asr, #imm	bics rs, ra, asr, #imm
bic rs, ra, ror, #imm	bics rs, ra, ror, #imm
bic rs, ra, rrx	bics rs, ra, rrx

cmp rs, ra, rb
cmp rs, ra, #imm
cmp rs, ra, lsl, rc
cmp rs, ra, lsr, rc
cmp rs, ra, asr, rc

cmp rs, ra, lsl, #imm
cmp rs, ra, lsr, #imm
cmp rs, ra, asr, #imm
cmp rs, ra, ror, #imm
cmp rs, ra, rrx

cmn rs, ra, rb
cmn rs, ra, #imm
cmn rs, ra, lsl, rc
cmn rs, ra, lsr, rc
cmn rs, ra, asr, rc
cmn rs, ra, lsl, #imm
cmn rs, ra, lsr, #imm
cmn rs, ra, asr, #imm
cmn rs, ra, ror, #imm
cmn rs, ra, rrx

tst rs, ra, rb
tst rs, ra, #imm
tst rs, ra, lsl, rc
tst rs, ra, lsr, rc
tst rs, ra, asr, rc
tst rs, ra, lsl, #imm
tst rs, ra, lsr, #imm
tst rs, ra, asr, #imm
tst rs, ra, ror, #imm
tst rs, ra, rrx

teq rs, ra, rb
teq rs, ra, #imm
teq rs, ra, lsl, rc
teq rs, ra, lsr, rc
teq rs, ra, asr, rc
teq rs, ra, lsl, #imm
teq rs, ra, lsr, #imm
teq rs, ra, asr, #imm
teq rs, ra, ror, #imm
teq rs, ra, rrx

ldr rs, [ra]

str rs, [ra]

Cas particulier des branchements :

Le branchement **b** a implicitement été implémenté dans le langage. Il suffit de faire deux transitions à partir d'un même état source, avec deux conditions inverses.

Exemple : b.eq <e3>

tran e0 → e1
	cond ne
	inst nop

tran e0 → e3
	cond eq
	inst nop

Liste des conditions implémentées

Signification des flags :
N : negative
Z : Zero
C : Carry (or Unsigned Overflow)
V : (Signed) Overflow

`eq`	Equal.	`Z==1`
`ne`	Not equal.	`Z==0`
`cs` **or** `hs`	Unsigned higher or same (or carry set).	`C==1`
`cc` **or** `lo`	Unsigned lower (or carry clear).	`C==0`
`mi`	Negative. The mnemonic stands for "minus".	`N==1`
`pl`	Positive or zero. The mnemonic stands for "plus".	`N==0`
`vs`	Signed overflow. The mnemonic stands for "V set".	`V==1`
`vc`	No signed overflow. The mnemonic stands for "V clear".	`V==0`
`hi`	Unsigned higher.	`(C==1) && (Z==0)`
`ls`	Unsigned lower or same.	`(C==0) || (Z==1)`
`ge`	Signed greater than or equal.	`N==V`
`lt`	Signed less than.	`N!=V`
`gt`	Signed greater than.	`(Z==0) && (N==V)`
`le`	Signed less than or equal.	`(Z==1) || (N!=V)`
`al` (or omitted)	Always executed.	None tested.

Le mot clé **al** n'a pas été ajouté, il a été remplacer par **true**. Le mot clé **false** a été ajouté.
Ainsi une transition avec la condition « true » sera toujours prise. Et un transition avec le mot clé «false » ne sera jamais pris.

L'**effet exact** des instructions sur les flags à été implémenté.