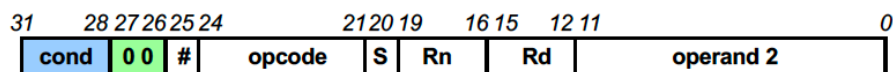


Codage binaire des instructions du ARM7TDMI

■ Instructions de traitement (1)



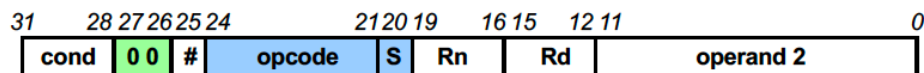
- *Cond*: l'instruction est exécutée si le registre d'état CPSR vérifie la condition spécifiée

Asm	Cond
EQ	0000
NE	0001
CS/HS	0010
CC/LO	0011
MI	0100
PL	0101

Asm	Cond
VS	0110
VC	0111
HI	1000
LS	1001
GE	1010
LT	1011

Asm	Cond
GT	1100
LE	1101
AL	1110
NV	1111

■ Instructions de traitement (2)



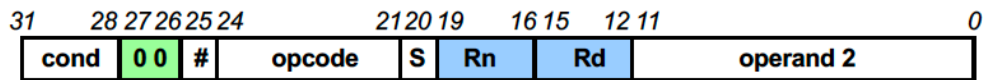
- *S = 1* : affecte CPSR
- *Opcode* : code de l'opération à effectuer

Asm	Opcode
AND	0000
EOR	0001
SUB	0010
RSB	0011
ADD	0100
ADC	0101

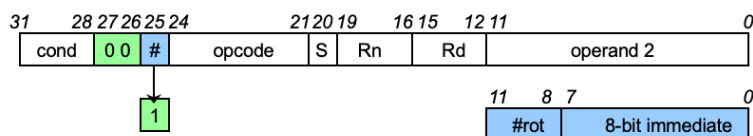
Asm	Opcode
SBC	0110
RSC	0111
TST	1000
TEQ	1001
CMP	1010
CMN	1011

Asm	Opcode
ORR	1100
MOV	1101
BIC	1110
MVN	1111

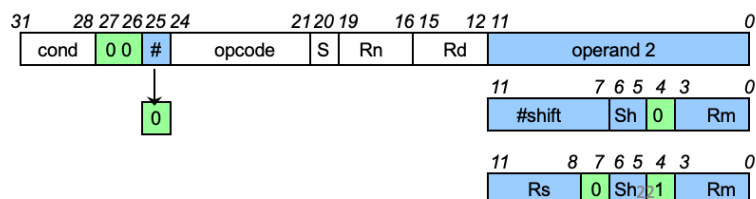
■ Instructions de traitement (3)



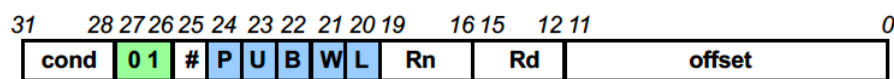
- $Rd \rightarrow$ numéro du registre de destinations
 - 4 bits pour sélection parmi les 16 registres possibles
- $Rn \rightarrow$ numéro du registre qui sert de premier opérande
 - 4 bits pour sélection parmi les 16 registres possibles
- $operand2 \rightarrow$ relié à l'entrée B
 - possibilité de rotation/décalage



- Si le bit # est à 1, $operand2$ est une valeur littérale
 - Exemple : `ADD R0, R1, #10` $\rightarrow R0 = R1 + 10$
- Si le bit # est à 0, $operand2$ est un registre (Rm) sur lequel on applique (ou non) une rotation/décalage
 - Exemple : `ADD R0, R2, R3` $\rightarrow R0 = R2 + R3$

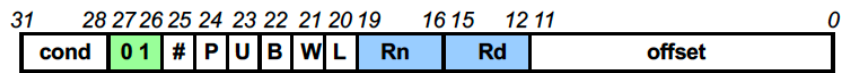


- Instructions de transfert
 - Lecture/écriture
 - Données accédées
 - Mots, demi-mots, octets
 - Signées/non signées
 - Mode d'accès (pré/post indexé, +/- offset, write back)
 - Transfert simple/multiple
 - Registre source/destination, liste de registres
 - Registre de base
 - Offset
 - Condition
 - Exécution conditionnelle d'un transfert
- Instructions de transfert de mots ou d'octets non-signés (LDR, STR, LDRB, STRB)



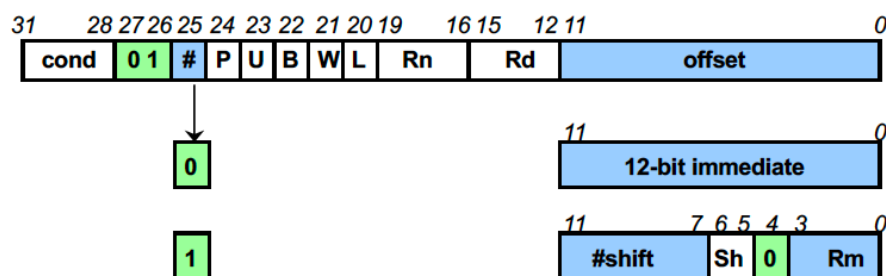
- *P* : pre/post index
 - 1 pré-indexé, 0 post-indexé
- *U* : up/down
 - 1 + offset, 0 -offset
- *B* : byte/word
 - M1 accès 8 bits, 0 accès 32 bits
- *W* : write-back
 - Si P=1, W=1 adressage pré-indexé automatique
- *L* : load/store
 - 1 load, 0 store

- Instructions de transfert de mots ou d'octets non-signés (LDR, STR, LDRB, STRB)



- $Rd \rightarrow$ registre source (si $L=0$, store) ou destination (si $L=1$, load)
- $Rn \rightarrow$ registre de base

- Instructions de transfert de mots ou d'octets non-signés (LDR, STR, LDRB, STRB)



- Offset: soit un littéral non signé sur 12 bits, soit un registre d'index (Rm) éventuellement décalé sur un nombre constant de bits ($\#shift$)

Instructions de branchement relatif

- Branch (B)
- Branch and Link (BL)



- Offset : déplacement signé sur 24 bits
- L : Link (0 branch, 1 branch and link)
- Effets:
 - B: $PC \leftarrow PC + \text{offset}$
 - BL: $r14 \leftarrow PC - 4$; $PC = PC + \text{offset}$
 - r14 : Link Register (contient l'adresse de retour)

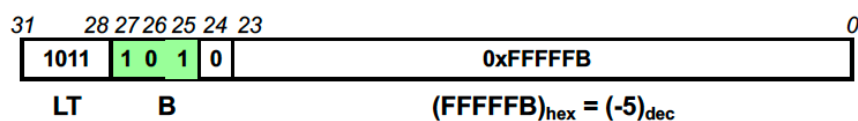
Instructions de branchement relatif

- Exemple traduction d'une boucle *for*
- Utilisation de labels en langage d'assemblage: le déplacement est calculé automatiquement par l'assembleur

```

                                MOV     r4, #0           @ tmp=0
                                MOV     r5, #0           @ i=0
loop:                          ADD     r4, r4, r5       @ tmp+=i
                                ADD     r5, r5, #1       @ i++
                                CMP     r5, #5
                                BLT     loop             @ i<5 : réitérer
                                ...

```



Représentation binaire: BAFFFFFFB