Instruction Encoding

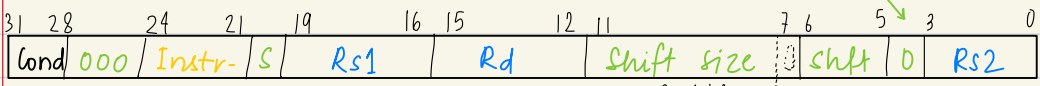
- what do i include?
  - → instruction type
  - → operands
  - → condition for conditional execution
  - → other flags

A Arithmetic & logic Instruction

ADD, ADC, SUB, SBC, RSB, AND, EOR, RSC, ORR, BIC

① 4 bits each

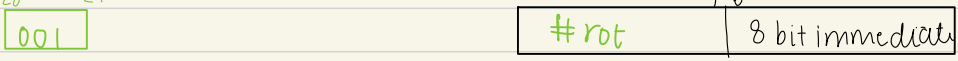
XXXCCS Rd, Rs1, Rs2, {shift #}

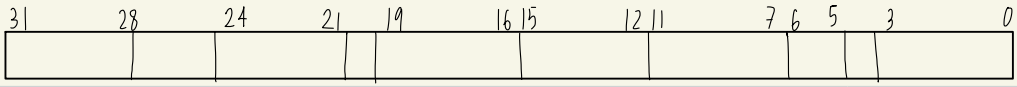


uction type — flag — dest — Rshift

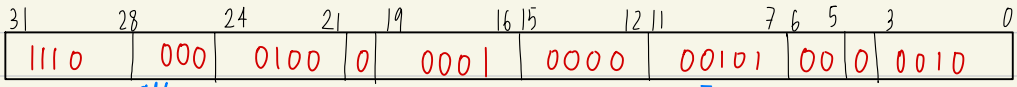
② {shift Rshift} — 1

③ Rd, Rs1, #immediate (rotated)





1 ADD R0, R1, R2, LSL #5



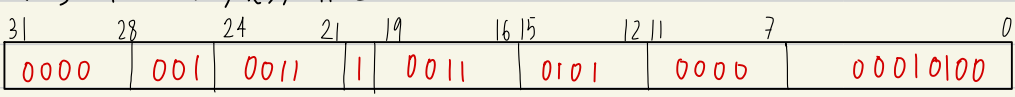
no condition
all reg
add
no flag set
first source R1
dest. R0
5
LSL
second source R2
shift w/ immediate

if no shift, 3→6 = 0 (as is)
and 7→11 : 0

2 SUBS R4, R3, R2, LSR R5

no condition
all reg
sub
flag set
source 1 R3
dest
reg R5
LSR
source 2
shift w/ reg

3. RSBEQS R5, R3, #20

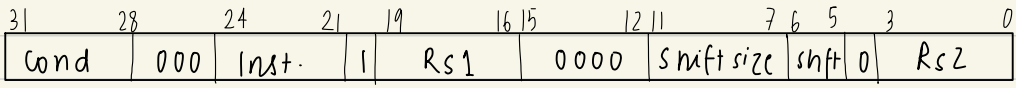


EQ cond.
immediate add. in 3rd.
RSB
flag set
source 1
dest
no rotate
number

B Comparison Instructions

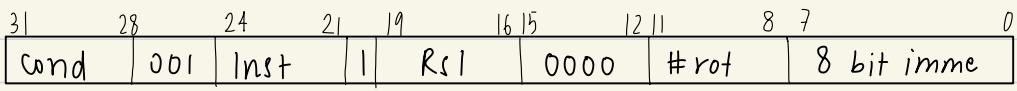
TST, TEQ, CMP, CMN

XXXcc Rs1, Rs2, {shft #}



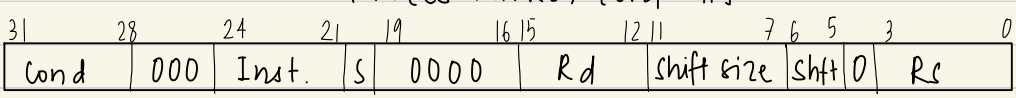
↑ has to be source 1 no dest source 2

XXXcc Rs1, #immediate (rotated)

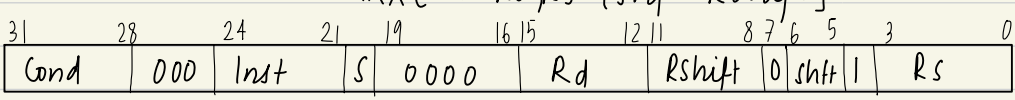


C MOV instructions.

XXXccS Rd, Rs, {shft #}

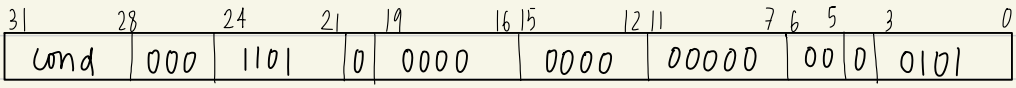


XXXccS Rd Rs {shft Rshift}

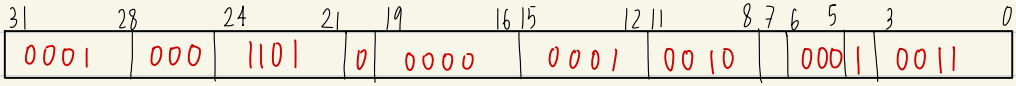


LSL R1, R2, #5 converted to MOV R1, R2, LSL #5

1. MOV R0, R5

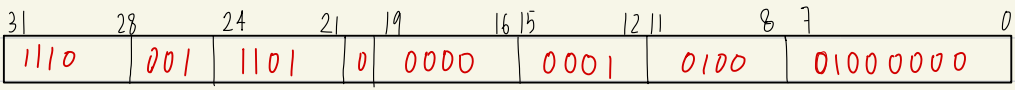


2. MOVNE R1, R3, LSL R2



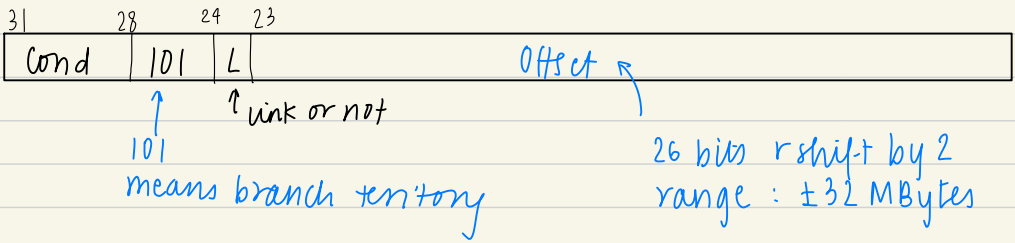
cond | all reg | MOV | not set | dest=0 | source R1 | reg R2 | LSL reg | reg | source R3

3. MOV R1, #0x40000000



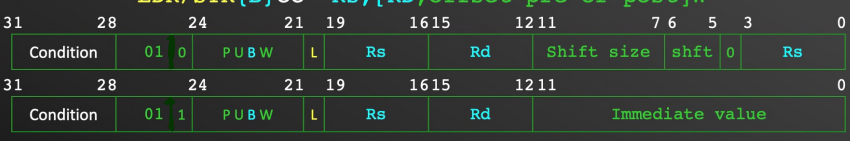
nocond | immediate | MOV | not set | dest=0 | R1

D. BRANCH



E. LDR/STR

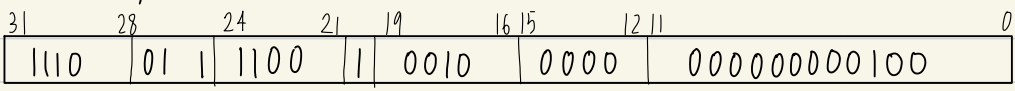
LDR/STR{B}CC Rs,[RD,offset pre or post]W



- L bit determines load (1) or store (0)
- P bit determines indexing pre (1) or post (0)
- U determines offset direction add (1) or subtract (0)offset
- B determines byte (1) or word (0) access
- W is for write back (!) of the offset

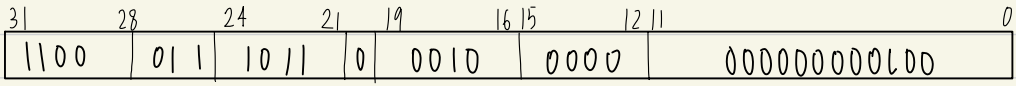
PUBW

1. LDR R0, [R2, #4]



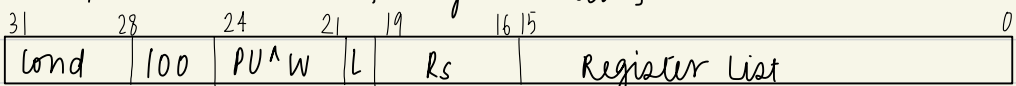
pre
add offset load
word
no write back

2. STRB R0, [R2, #-4]!



PUBW

F LDM/STMFXCC Rsw, {register list}



^ don't care

one hot
for each reg.

PU	Instruction
10	STMFD
01	LDMFD
00	STMFA
11	LDMFA