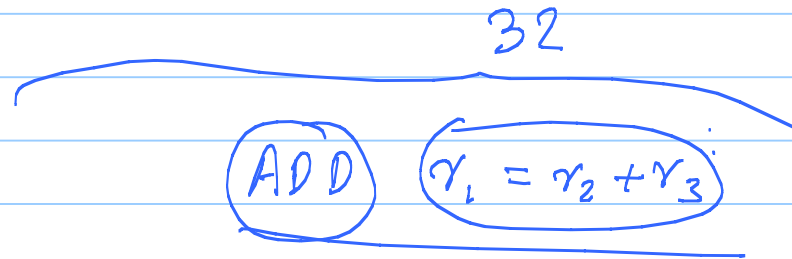


Aug 12th.

Note Title

12-08-2011

1) Instruction Format



Data Processing Insts.

F-2 (0 → DP, 1-DT, 2-Branch)

opcode-4 ($2^4 = 16$ types of DP Insts.)

$x > 1$
 $x < 1$
 if ($x == 1$)

$y = z + z;$

CMP $x, \#1$

ADDEQ $y, z, \#2$

↑
 Predicated
 Instr.

CMP, $x, \#1$

CPSR ✓ result.

$1 \ x > 1$
 $0 \ x == 1$ } EQ
 $-1 \ x < 1$ }

Cond bits

4
 16 possible conditions.

EQ HI

NE HO

LE

LT 14 - ALWAYS

GE

GT

4

left = 26 - 4

22

CPSR

N C Z O

↑

↑

$(x=y-z)$
 SUBS x, y, z
 ↑

$(x < y) \ ? \ (y < z)$
 $N \quad z \quad C \quad 0$
 $-1 \quad 0$

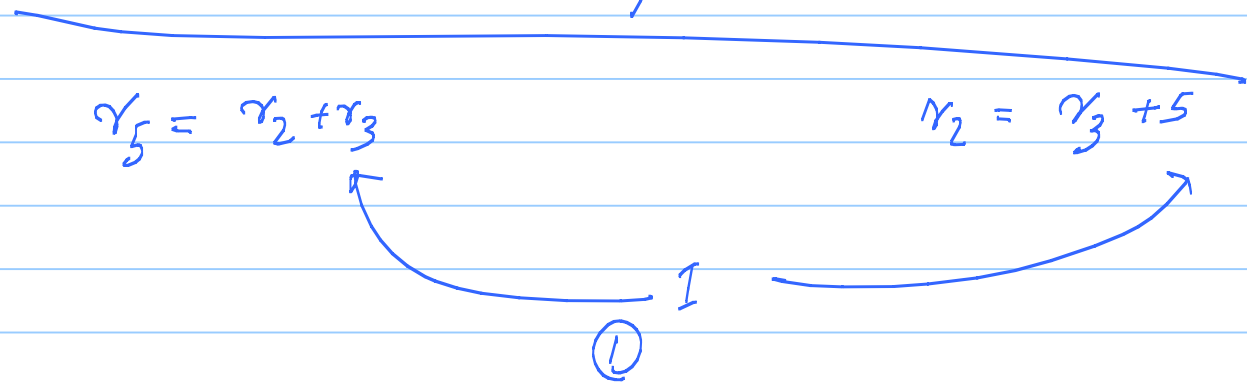
$LT \rightarrow N$
 $LE \rightarrow N \& Z$
 $EQ \rightarrow Z$
 $GT \rightarrow (!N) \&$
 $GE \rightarrow Z \& (!Z)$

Please
set cpsr

SUB → Don't set

1 bit for S → flag

(21)



Left with 20 bits.

$$r_2 = r_3 + 5$$

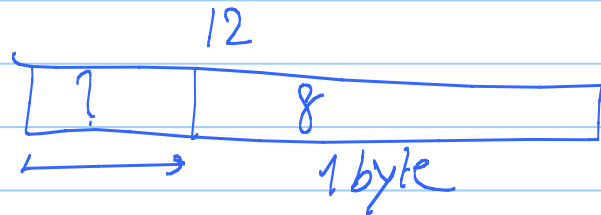
Diagram showing bit allocation for the equation above. An arrow points from the number 4 below r_2 to the number 4 below r_3 . Another arrow points from the number 4 below r_3 to the number 5 in the equation. A third arrow points from the number 12 below r_3 to the number 5 in the equation.

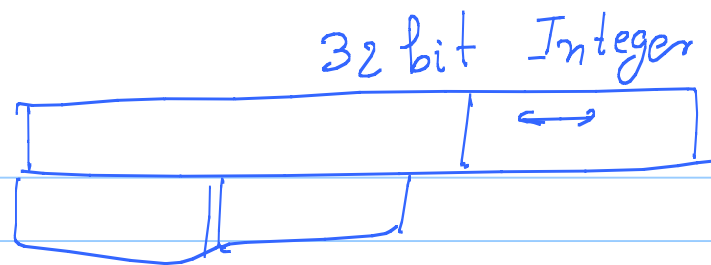
$$r_2 = r_3 + r_4$$

Diagram showing bit allocation for the equation above. An arrow points from the number 4 below r_3 to the number 4 below r_4 . A bracket under the number 4 below r_4 is labeled "4 bits".

Last 12 bits.

12 bits for immediate.





Use the rest 4 bits to push.
the 8 bits. (ROR)

$4 \rightarrow i$

8 bits $\rightarrow x$

final number: $x \ll (2i)$
(almost correct)
(See the book)

00 00 00 FE
ROR
Shift

EO 00 00 0F
(Right Rotate)

MOV r3, 0x EO 00 00 0F
(This is sufficient).

Addressing Modes

Immediate: $r_1 = r_2 + 5.$

Register: $r_1 = r_2 + r_3.$

Shift: `ADD r1, r2, r3, LSL #2`

Load:

`LDR r1, [r5] (Register)`

`LDR, r1, [r5, #10] (offset)`

register-offset.

LDR r_1 , [r_2 , r_3] (address = $r_2 + r_3$)

shifted reg. offset

LDR r_1 , [r_2 , r_3 , LSL#2]

address = $r_2 + r_3 \ll 2$

pre-indexed.

LDR r_1 , [r_0 , #4]!

register.

$r_0 = r_0 + 4$

address = r_0

DDR $\gamma_1, [\gamma_0, \gamma_2, \text{LSL} \#2]!$

$$\gamma_0 = \gamma_0 + \gamma_2 \ll 2$$

add ~~rep~~ = γ_0

for ($i=0; i < n; i++$)

$A[i] = 5;$

\nearrow
 γ_0