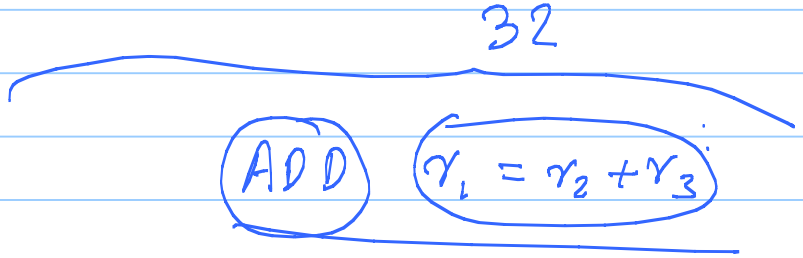
Aug 12th.

Note Title 12-08-2011

1) Instruction Format




Data Processing Instrs.

F-2 (0 → DP, 1 - DT, 2 - Branch)

Opcode-4 ($2^4 = 16$ types of DP Instr.)

x > 1
x < 1

```
if(x==1)
    y = 2+ z;
```

```
CMP   x, #1
ADDEQ  y, z, #2
```

↑ Predicated Instr.

CMP, x, #1 → Result → CPSR

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

Please set cpsr

$x = y - z$

$(x < y)$ & $(y < z)$

N Z C O

-1 0

LT → N

LE → N & Z

EQ → Z

GT → (!N) & (!Z)

GE → Z & (!N)

SUB → Don't set

1 bit for S → flag

(21)

$r_5 = r_2 + r_3$

$r_2 = r_3 + 5$

I

①

Left with 20 bits.

$r_2 = r_3 + 5$

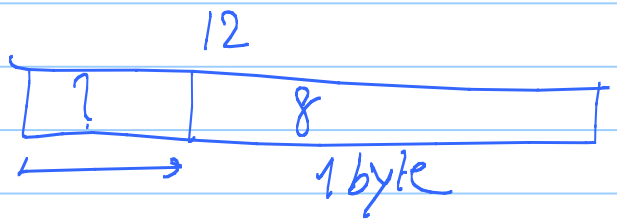
4 4 12

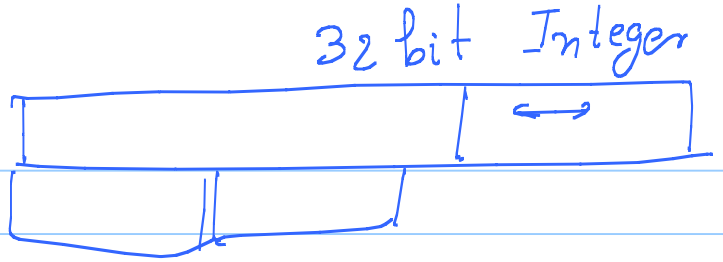
$r_2 = r_3 + r_4$

4 bits.

Last 12 bits.

12 bits for immediate.



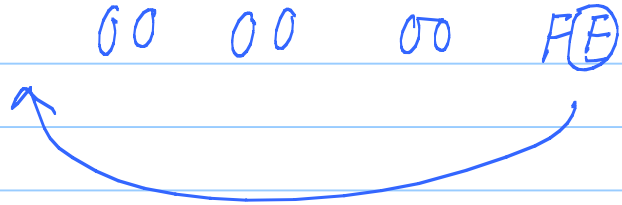



Use the rest 4 bits to push.
(ROR)
the 8 bit.

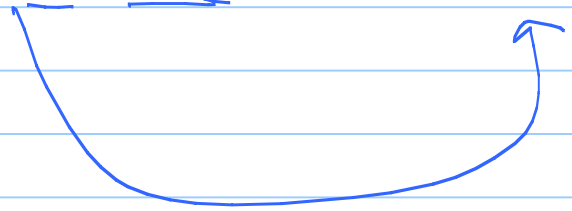
4 → i

8 bits → x

final number: x << (2i)
(almost correct)
(See the book)




ROR Shift




(Right Rotate)

MOV $r_3$, 0x E0 00 00 0F

(This is sufficient).

# Addressing Modes

Immediate: $r_1 = r_2 + 5$.

Register: $r_1 = r_2 + r_3$.

Shift:

ADD $r_1, r_2, r_3,$ LSL #2

Load:

LDR $r_1, [r_5]$ (Register)

LDR, $r_1, [r_5, \#10]$ (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 = $r_2 + r_3 << 2$

pre- indexed.

register.

LDR $r_1$, [$r_0$, #4]!

$r_0 = r_0 + 4$

address = $r_0$

LDR $r_1$, [$r_0$, $r_2$, LSL #2]!

$r_0 = r_0 + r_2 << 2$

address $= r_0$

```
for ( i=0;  i < n;  i++)
    A[i] = 5;
```

↑ $r_0$