Aug 18th

Note Title 18-08-2011

Assembly Instructions [ARM ARCHITECTURE REFERENCE MANUAL]

Data Processing Instructions

Arithmetic: ADD, SUB, MUL, UDIV

Shift: LSL, LSR, ASR

Logical: ORR, AND

Data Movement: MOV, MVN

Three formats: Register, Immediate, Shifted

Data Transfer Instructions (DT)

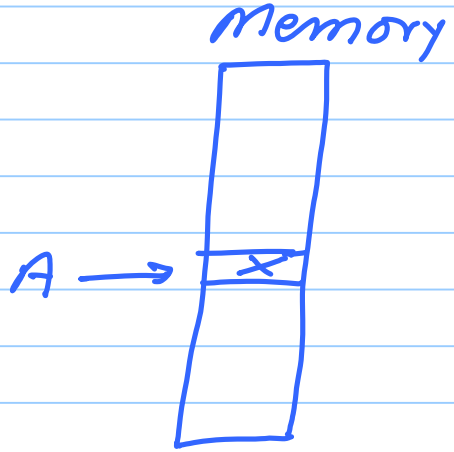
MEMORY <--(DT)--> Registers

LDR , STR

LDR (Load Register)

LDR r_1, [r_2]

1) Read the value stored in register r_2
2) Let the value be A
3) Read memory at location, A
4) Let the value be X

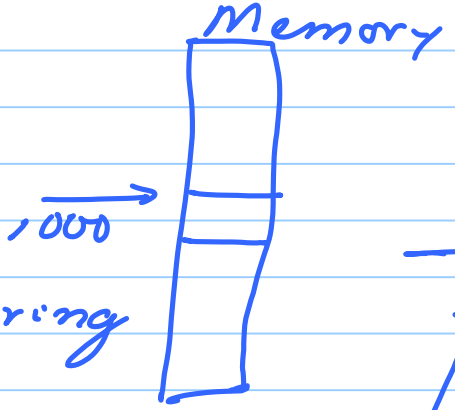


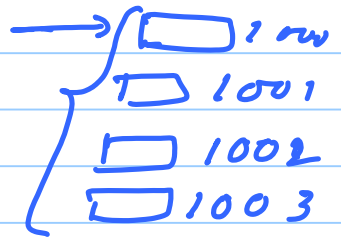
5) Save x in register r_1

LDR <destination reg.> [Address]

$r_2 = 1000$

LDR r_1, [r_2]





Unless specified otherwise we are always loading or storing an integer (4 bytes)

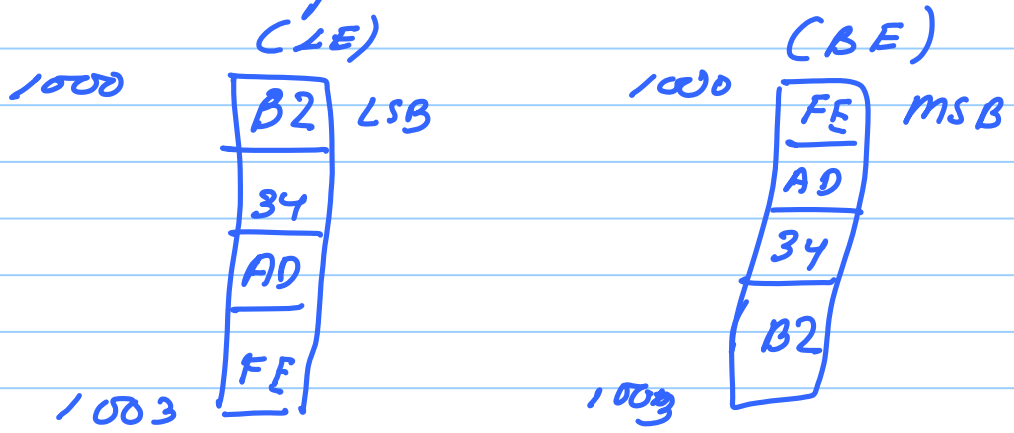
- Int - 4
- Short - 2
- Char - 1
- Long / Double - 8
- Float - 4

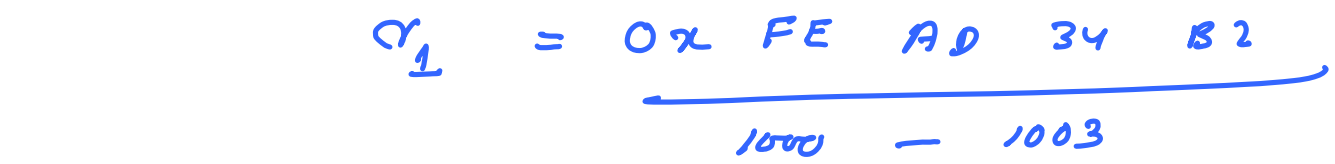
Little Endian vs. Big Endian

$i = 0x \underbrace{FE}_{MSB} AD\ 34 \underbrace{B2}_{LSB}$

Little Endian → LSB is stored in the lowest position
(ARM, INTEL x86)

Big Endian → MSB is stored in the lowest position.
(sparc, IBM, HP)





STR

STR r_1, [r_2]

[Reverse of a Load]

x y

MEM[y] = x

Different variants of LDR / STR Instructions

Formats / Addressing Modes

1) LDR r_1, [r_2] Register Format

Address = r_2

(least count is always bytes)

2) LDR r_1, [r_2, #16]

Address = r_2 + 16 [Immediate Format]

3) LDR r_1, [r_2, r_3]

Address = $r_2 + r_3$ [Register offset]

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

Address = $r_2 + r_3 << 2$ [Scaled register offset]

:

Some more addressing modes

(We will look at them later)

LDR/STR → Load & Store Integers

LDRB/STRB → Load / Store 1 byte

LDRH/STRH → Load / Store 2 bytes

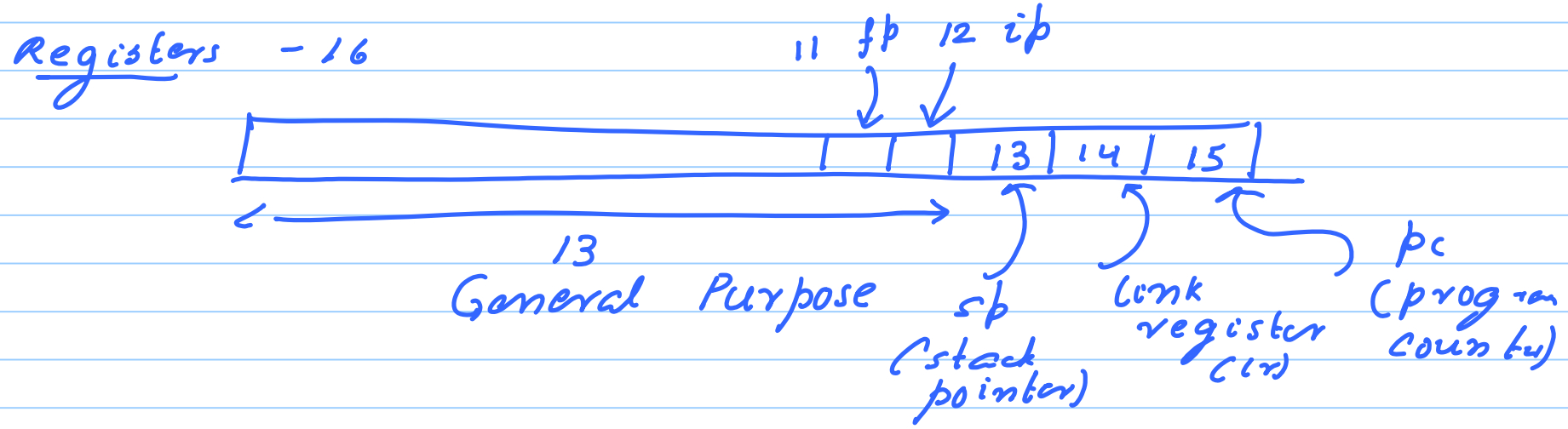
MEMORY WORD → 4 byte.

✓ Data Processing & Data Transfer.

✗ Control Instruction

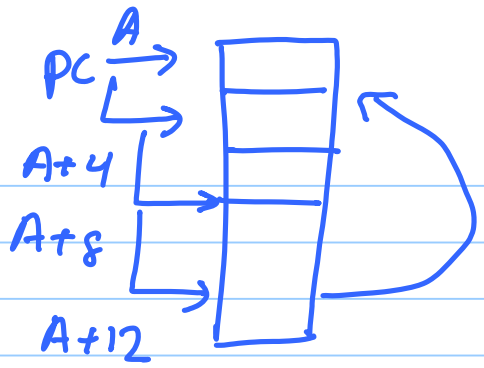
Branches, Jumps, Procedure calls, Returns

Transfer of Control Flow.



pc → currently executing instruction.
(program counter)

One instr. → 32 bits
(4 bytes)

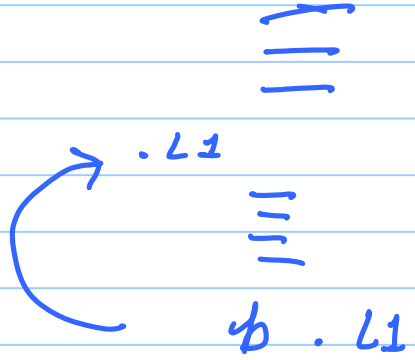


[goto, for, while]

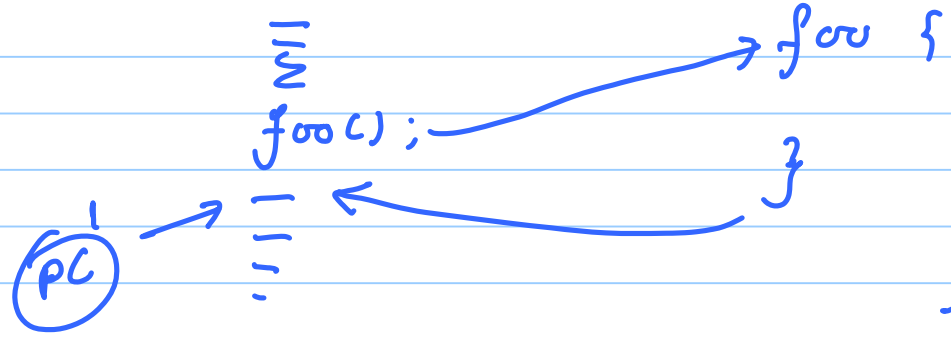
Instruction that implements transfer of control : b (branch)

≡ Assembly

b <name of label> (goto, if)



Procedure Call



bl
(branch and link

bl <name of the function>

lr ← the value of (pc)' is stored.

(code

```
foo{};
pc:1000→ a=b+c;←
```

```
bl    foo    (lr = 1000)
return: mov   pc, lr
```

Tutorial : Saturday.

Basic Overview of Assembly Code

[Some examples]

(1) Mini - Homework

TA: Abhishek Sagar

Next Week: Wednesday. 11 - 12:30

Saturday : 11 - 12:30