ARM Machine Language

Young W. Lim

June 15, 2016

Data Processing

```
 4-bit : Condition
 2-bit : 00
 1-bit : I
 4-bit : OPCODE
 1-bit : S

 4-bit : Rn
 4-bit : Rd
12-bit : OPERAND-2
```

- Arithmetic Operations
- Comparison Operations
- Logical Operations
- Data Movement

```
<op>{<cond>}{S} Rd, Rn, #<32-bit immediate>
<op>{<cond>}{S} Rd, Rn, Rm
```

Multiply

```
4-bit : Condition
2-bit : 00
1-bit : 0
4-bit : 000A
1-bit : S

4-bit : Rd
4-bit : Rn
4-bit : Rs
4-bit : 1001
4-bit : Rm
```

- MUL
- MLA

```
MUL{<cond>}{S} Rd, Rm, Rs      ; Rd = (Rm * Rs)
MLA{<cond>}{S} Rd, Rm, Rs, Rn  ; Rd = Rn + (Rm * Rs)
```

Long Multiply

```
4-bit : Condition
2-bit : 00
1-bit : 0
4-bit : 01UA
1-bit : S

4-bit : Rd-High
4-bit : Rd-Low
4-bit : Rn
4-bit : 1001
4-bit : Rm
```

- UMULL, SMULL
- RdHi, RdLo = (Rm*Rs)
- SMLAL, SMLAL
- RdHi, RdLo = (Rm*Rs) + RdLo

```
{U,S}MULL{<cond>}{S} RdLo, RdHi, Rm, Rs
{U,S}MLAL{<cond>}{S} RdLo, RdHi, Rm, Rs
```

Swap

```
4-bit : Condition
2-bit : 00
1-bit : 0
4-bit : 10B0
1-bit : 0

4-bit : Rn
4-bit : Rd
4-bit : 0000
4-bit : 1001
4-bit : Rm
```

```
SWP{<cond>}{B} Rd, Rm, [Rn]  ; Rd<=[Rn], [Rn]<=Rm
```

Load/Store - Byte/Word

```
 4-bit : Condition
 2-bit : 01
 1-bit : I
 4-bit : PUBW
 1-bit : L

 4-bit : Rn
 4-bit : Rd
12-bit : OFFSET

LDR|STR{<cond>}{B} Rd, [Rn, <offset>]{!}
LDR|STR{<cond>}{B}{T} Rd, [Rn], <offset>
LDR|STR{<cond>}{B} Rd, LABEL
```

Load/Store Multiple

```
 4-bit : Condition
 2-bit : 10
 1-bit : 0
 4-bit : PUBW
 1-bit : L

16-bit : Register List
```

LDS|STM{<cond>}<add mode>
Rn{!}, <registers>

Halfword Transfer Imm Offset

```
4-bit : Condition
2-bit : 00
1-bit : 0
4-bit : PU1W
1-bit : L

4-bit : Rn
4-bit : Rd
4-bit : Offset 1
4-bit : 1SH1
4-bit : Rm
```

```
LDR|STR{<cond>}H|SH|SB Rd, [Rn, <offset>]{!}
LDR|STR{<cond>}H|SH|SB Rd, [Rn], <offset>
```

Halfword Transfer Reg Offset

```
4-bit : Condition
2-bit : 00
1-bit : 0
4-bit : PUOW
1-bit : L

4-bit : Rn
4-bit : Rd
4-bit : 0000
4-bit : 1SH1
4-bit : Rm
```

Branch

B{L}{<cond>} <target address>

```
 4-bit : Condition
 2-bit : 10
 1-bit : 1
 1-bit : L

24-bit : Branch Offset
```

Branch Exchange

```
4-bit : Condition
2-bit : 00
1-bit : 0
4-bit : 1001
1-bit : 0

4-bit : 1111
4-bit : 1111
4-bit : 1111
4-bit : 0001
4-bit : Rn
```

B{L}X{<cond>} Rm
BLX <target address>

Coprocessor Data Transfer

```
4-bit : Condition
2-bit : 11
1-bit : 0
4-bit : PUNW
1-bit : L

4-bit : Rn
4-bit : CRd
4-bit : CPNum
8-bit : Offset
```

```
LDC|STC{<cond>}{L} <CP#>, CRd, [Rn, <offset>}{!}
LDC|STC{<cond>}{L} <CP#>, CRd, [Rn], <offset>
```

Coprocessor Data Op

```
4-bit : Condition
2-bit : 11
2-bit : 10
4-bit : OP-1

4-bit : CRn
4-bit : CRd
4-bit : CPNum
3-bit : OP-2
1-bit : 0
4-bit : Rm
```

```
CDP{<cond>} <CP#>, <Cop1>, CRd, CRn, CRm{, <Cop2>}
```

Coprocessor Reg Transfer

```
4-bit : Condition
2-bit :
2-bit :
3-bit : OP-1
1-bit : L

4-bit : CRn
4-bit : CRd
4-bit : CPNum
3-bit : OP-2
1-bit : 1
4-bit : Rm
```

```
MRC{<cond>} <CP#>, <Cop1>, Rd, CRn, CRm{, <Cop2>}
MCR{<cond>} <CP#>, <Cop1>, Rd, CRn, CRm{, <Cop2>}
```

Software Interrupt

```
 4-bit : Condition
 2-bit : 11
 2-bit : 11
 4-bit : OP-1

24-bit : SWI Number

SWI{<cond>} <24-bit immediate>
```

Encoding Load/Store Format

- P (Pointer adjust)
  - 0: post-index operation
  - 1: pre-index operation
- U (Pointer direction Up/Down)
  - 0: decrement pointer
  - 1: increment pointer
- B (Operand Size Byte/Word)
  - 0: word acsess
  - 1: byte access
- W (Pointer update Write-back)
  - 0: dont write back adjusted pointer
  - 1: write back adjusted pointer
- L (Data direction Load/Store)
  - 0: store in memory
  - 1: load into register

Encoding Other Formats

- I (Immediate operand)
  - 0: operand 2 is a register
  - 1: operand 2 is an immediate number
- S (Set condition codes)
  - 0: do not alter condition codes
  - 1: set condition codes
- A (Accumulate)
  - 0: multiply only
  - 1: multiply and accumulate
- N (Transfer Length)

Reference

[1] D. Harris, Digital Design and Computer Architecture", 2nd ed.
[2] D.A. Patterson & J.H. Hennessy, Computer Organization and Design (ARM ed)