

# ISA Overview (1A)

---

Copyright (c) 2014 - 2018 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Please send corrections (or suggestions) to [youngwlim@hotmail.com](mailto:youngwlim@hotmail.com).

This document was produced by using LibreOffice.

# Based on

---

ARM System-on-Chip Architecture, 2<sup>nd</sup> ed, Steve Furber

<b>Rn</b>	1 <sup>st</sup> Operand Reg / Base Reg
<b>Rm</b>	2 <sup>nd</sup> Operand Reg / Operand Reg / Offset Reg / Source Reg
<b>Rd</b>	Source / Destination Reg
<b>S</b>	Set Condition Codes / Signed / Restore PSR and force user bit
<b>P</b>	Pre/Post Index
<b>U</b>	Up/Down
<b>B</b>	Unsigned Byte/Word
<b>H</b>	Half-word Address
<b>L</b>	Link / Load/Store
<b>W</b>	Write-back (auto-index)
<b>Opcode</b>	4-bit op codes
<b>Sh</b>	Shift type
<b>R</b>	CPSR/SPSR

Branch and Branch with Link (B, BL)  
Branch, Branch with Link and eXchange (BX, BLX)  
Data Processing Instructions  
Single Word and Unsigned Byte Transfer Instructions  
Half-word and Signed Byte Transfer Instructions  
Multiple Register Transfer Instructions  
Status Register to General Register Transfer Instructions  
General Register to Status Register Transfer Instructions  
Software Interrupt (SWI)  
Swap Memory and Register Instructions  
Multiply Instructions  
Count Leading Zeros (CLS)

- Coprocessor Instructions
- Coprocessor Data Operations
- Coprocessor Data Transfers
- Coprocessor Register Transfers
- Breakpoint Instruction (BKPT)
- Unused Instruction Space

# Assembler Format

## Branch and Branch with Link (B, BL)

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

## Branch, Branch with Link and eXchange (BX, BLX)

B{L}X {<cond>} Rm

BLX <target address>

## Data Processing Instructions

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

<op> {<cond>} {S} Rd, Rm, {<shift>}

## Single Word and Unsigned Byte Transfer Instructions

LDR|STR {<cond>} {B} Rd, {Rn, <offset>} {!}

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

LDR|STR {<cond>} {B} Rd, LABEL

## Half-word and Signed Byte Transfer Instructions

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

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

# B{L} {<cond>}

Branch and Branch with Link (B, BL)

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

L : the branch and link

<cond> : condition codes, AL if omitted

<target address> : a label in the assembler code

The assembler will generate the offset  
target address – branch instruction address + 8



# B{L}X {<cond>}, BLX <target address>

Branch, Branch with Link and eXchange (BX, BLX)

B{L}X {<cond>} Rm

BLX <target address>

L : the branch and link

<cond> : condition codes, AL if omitted

<target address> : a label in the assembler code

The assembler will generate the offset  
target address – branch instruction address + 8

# B{L}X {<cond>}, BLX <target address>

## Data Processing Instructions

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

<op> {<cond>} {S} Rd, Rm, {<shift>}

L : the branch and link

<cond> : condition codes, AL if omitted

<target address> : a label in the assembler code

The assembler will generate the offset  
target address – branch instruction address + 8

# Assembler Format

## Multiple Register Transfer Instructions

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

## Status Register to General Register Transfer Instructions

MRS {<cond>} Rd, CPSR|SPSR

## General Register to Status Register Transfer Instructions

MSR {<cond>} CPSR\_f|SPSR\_f, #<32-bit immediate>

MSR {<cond>} CPSR\_<field>|SPSR\_<field>, Rm

## Software Interrupt (SWI)

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

## Swap Memory and Register Instructions

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

# Assembler Format

## Multiply Instructions

MUL {<cond>} {S} Rd, Rm, Rs

MLA {<cond>} {S} Rd, Rm, Rs, Rn

<mul> {<cond>} {S} RdHi, RdLo, Rm, Rs

UMULL, UMLAL, SMULL, SMLAL

## Count Leading Zeros (CLZ)

CLZ {<cond>} Rd, Rm

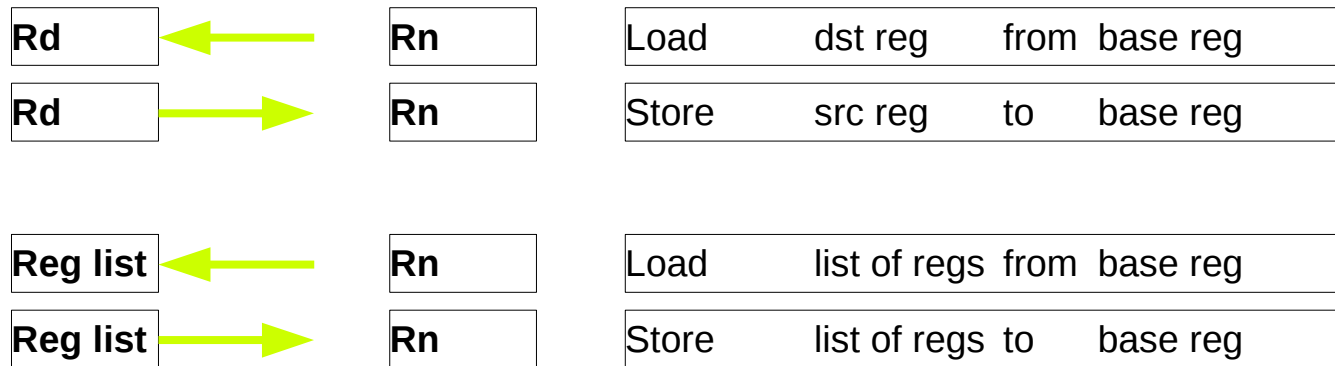
- Coprocessor Instructions
- Coprocessor Data Operations
- Coprocessor Data Transfers
- Coprocessor Register Transfers
- Breakpoint Instruction (BKPT)
- Unused Instruction Space

Rn            1<sup>st</sup> Operand Reg / Base Reg  
 Rd            Source/Destination Reg  
 Rm            2<sup>nd</sup> Operand Reg / Source Reg / Offset Reg / Operand Reg

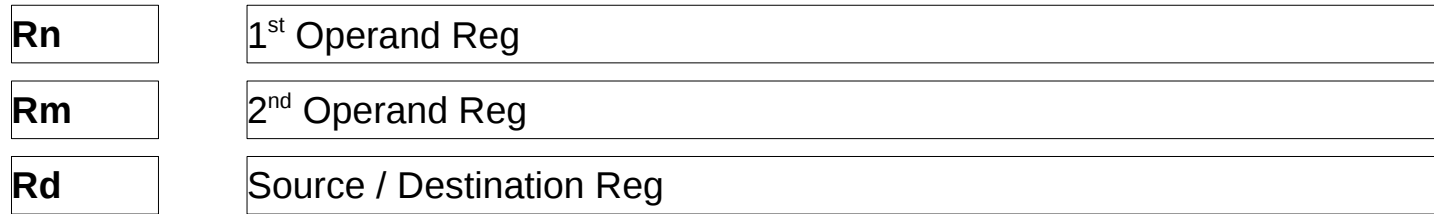
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
cond		0	0	#	opcode				S	Rn				Rd				Operand 2														
cond		0	0	0	1	0	1	1	0	0	0	0	0	Rd				0	0	0	0	0	0	0	1	Rm						
cond		0	1	#	P	U	B	W	L	Rn				Rd				offset														
N		0	0	0	P	U	#	W	L	Rn				Rd				offsetH		1	S	H	1	offsetL								
cond		1	0	0	P	U	S	W	L	Rn				Register list																		
cond		0	0	0	1	0	B	0	0	Rn				Rd				0	0	0	0	1	0	0	1	Rm						
cond		0	0	0	1	0	R	0	0	1	1	1	1	Rd				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
cond		0	0	0	1	0	R	1	0	field				1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	Rm			

# Data Transfer

<b>Rn</b>	1 <sup>st</sup> Operand Reg / Base Reg
<b>Rm</b>	2 <sup>nd</sup> Operand Reg / Operand Reg / Offset Reg / Source Reg
<b>Rd</b>	Source / Destination Reg



# Data Processing





## References

- [1] <ftp://ftp.geoinfo.tuwien.ac.at/navratil/HaskellTutorial.pdf>
- [2] <https://www.umiacs.umd.edu/~hal/docs/daume02yaht.pdf>