ISA Overview (1A)

Young Won Lim
5/28/18

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using LibreOffice.

Based on

ARM System-on-Chip Architecture, 2nd ed, Steve Furber

ARM

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

ARM

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 Instrucitons
Multiply Instructions
Count Leading Zeros (CLS)

Coprocessor Instructions
Coprocessor Data Operations
Coprocessor Data Transfers
Copressor 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 Instrucitons

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 (CLS)

CLZ {<cond>} Rd, Rm

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

Rn	1st Operand Reg / Base Reg
Rd	Source/Destination Reg
Rm	2nd 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
cond				0	0	0	1	0	R	1	0	field				1	1	1	1	0	0	0	0	0	0	0	0	Rm			

Data Transfer

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

Rd	←	**Rn**	Load	dst reg	from base reg
Rd	→	**Rn**	Store	src reg	to base reg
Reg list	←	**Rn**	Load	list of regs	from base reg
Reg list	→	**Rn**	Store	list of regs	to base reg

Data Processing

Rn	1st Operand Reg
Rm	2nd Operand Reg
Rd	Source / Destination Reg

References

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