ISA Assembler Format (3A)

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

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>

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

Branch and Branch with Link (B, BL)

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

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

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

Data Processing Instructions

Data Processing Instructions

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

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

Omitting Rn when the instruction is monadic (MOV, MVN)

Omitting Rd when the instruction only produces condition code

(CMP, CMN, TST, TEQ)

<shift> specifies the shift type (LSL, LSR, ASI, ASR, ROR, RRX)

Except RRX, the shift amount

By a 5-bit immediate (<shift>)

Bu a register (Rs)

References

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