

Instruction Set Architecture 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".

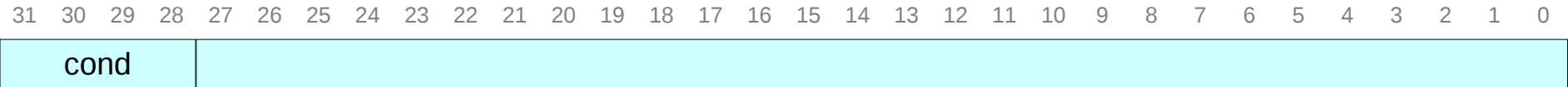
Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using LibreOffice.

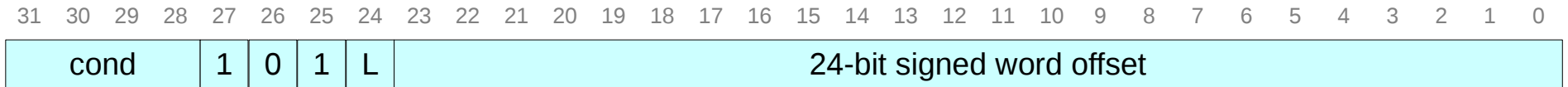
Based on

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

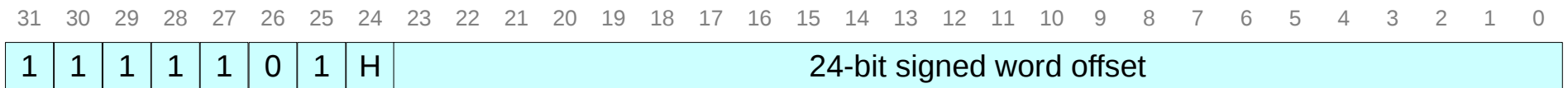
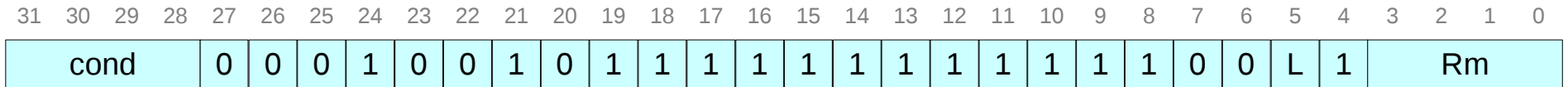
Condition Code



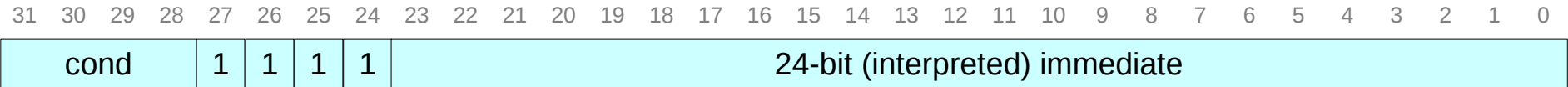
Branch and Branch with Link (B, BL)



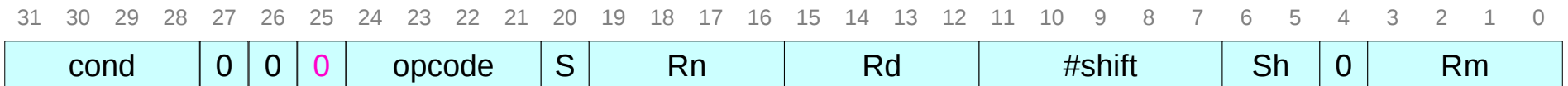
Branch, Branch with Link and eXchange (BX, BLX)



SWI (Software Interrupt)



Data Processing Instructions



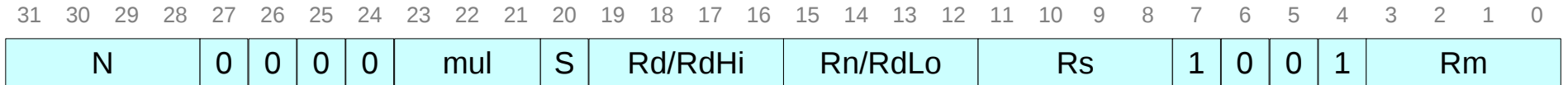
Data Processing Instructions

31 30 29 28 27 26 25 24 23 22 21 20

cond	0	0	#	0000	S
cond	0	0	#	0001	S
cond	0	0	#	0010	S
cond	0	0	#	0011	S
cond	0	0	#	0100	S
cond	0	0	#	0101	S
cond	0	0	#	0110	S
cond	0	0	#	0111	S
cond	0	0	#	1000	S
cond	0	0	#	1001	S
cond	0	0	#	1010	S
cond	0	0	#	1011	S
cond	0	0	#	1100	S
cond	0	0	#	1101	S
cond	0	0	#	1110	S
cond	0	0	#	1111	S

AND	Logical bit-wise AND	$Rd := Rn \text{ AND } Op2$
EOR	Logical bit-wise XOR	$Rd := Rn \text{ EOR } Op2$
SUB	Subtract	$Rd := Rn - Op2$
RSB	Reverse Subtract	$Rd := Op2 - Rn$
ADD	Add	$Rd := Rn + Op2$
ADC	Add with carry	$Rd := Rn + Op2 + C$
SBC	Subtract with carry	$Rd := Rn - Op2 + C - 1$
RSC	Reverse subtract with carry	$Rd := Op2 - Rn + C - 1$
TST	Test	$Rn \text{ AND } Op2$
TEQ	Test equivalence	$Rn \text{ EOR } Op2$
CMP	Compare	$Rn - Op2$
CMN	Compare negated	$Rn + Op2$
ORR	Logical bit-wise OR	$Rd := Rn \text{ OR } Op2$
MOV	Move	$Rd := Op2$
BIC	Bit clear	$Rd := Rn \text{ AND NOT } Op2$
MVN	Nive begated	$Rd := \text{NOT } Op2$

Multiply Instructions

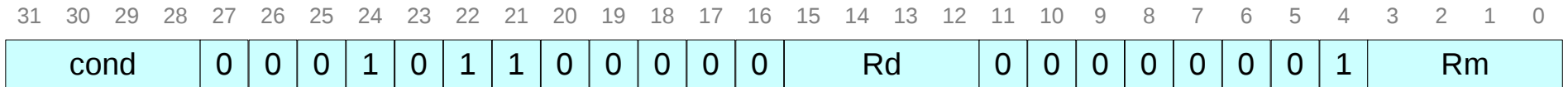


Multiply Instructions

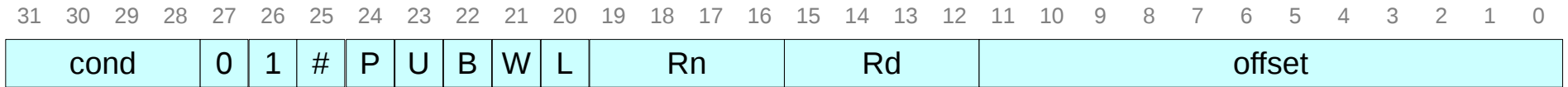
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N				0	0	0	0	mul	S	Rd/RdHi				Rn/RdLo				Rs				1	0	0	1	Rm					

0	0	0	MUL	Multiply (32-bit result)	Rd := (Rm * Rs)[31:0]
0	0	1	MLA	Multiply-accumulate (32-bit result)	Rd := (Rm * Rs)[31:0]
1	0	0	UMULL	Unsigned multiply long	RdHi.RdLo := Rm * Rs
1	0	1	UMLAL	Unsigned multiply-accumulate long	RdHi.RdLo += Rm * Rs
1	1	0	SMULL	Signed multiply long	RdHi.RdLo := Rm * Rs
1	1	1	SMLAL	Signed multiply-accumulate long	RdHi.RdLo += Rm * Rs

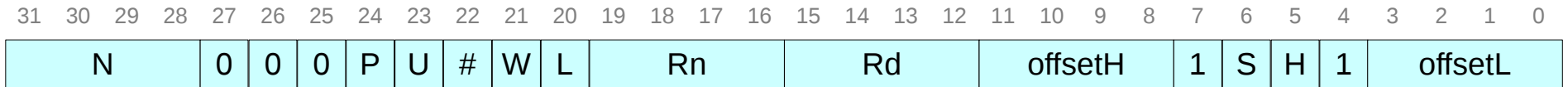
CLZ (Count leading zeros)



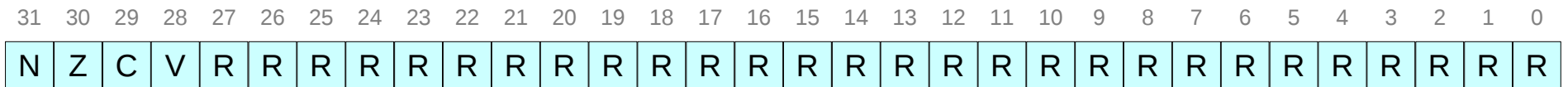
Single word and unsigned byte data transfer instructions



ARM Exception Handling



ARM Exception Handling



References

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