COMS12600 Target ThumbV1 ISA Sub-set

Simon Hollis (simon@cs.bris.ac.uk) V2013.9

ARM Assembler		Opcode	Flags	V	Encoding															
Page	Format				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
221	`ADDI rdn, #i8`	ADDI	Y	N	0	0	1	1	0	rdn	rdn	rdn	i8	i8	i8	i8	i8	i8	i8	i8
223	`ADDR rd, rn, rm`	ADDR	Y	-	0	0	0	1	1	0	0	rm	rm	rm	rn	rn	rn	rd	rd	rd
225	`ADDSPI rdn, sp, #i8`	ADDSPI	N	N	1	0	1	0	1	rdn	rdn	rdn	i8	i8	i8	i8	i8	i8	i8	i8
225	`INCSP sp, #i7`	INCSP	N	N	1	0	1	1	0	0	0	0	0	i7	i7	i7	i7	i7	i7	i7
229	`ADDPCI rd, pc, #i8`	ADDPCI	N	N	1	0	1	0	0	rd	rd	rd	i8	i8	i8	i8	i8	i8	i8	i8
495	`SUBI rdn, #i8`	SUBI	Y	N	0	0	1	1	1	rdn	rdn	rdn	i8	i8	i8	i8	i8	i8	i8	i8
497	`SUBR rd, rn, rm`	SUBR	Y	-	0	0	0	1	1	0	1	rm	rm	rm	rn	rn	rn	rd	rd	rd
499	`DECSP sp, #i7`	DECSP	N	N	1	0	1	1	0	0	0	0	1	i7	i7	i7	i7	i7	i7	i7
359	`MULR rdmn, rn`	MULR	Y	N	0	1	0	0	0	0	1	1	0	1	rn	rn	rn	rdm	rdm	rdm
					15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
233	`ANDR rdn, rm`	ANDR	Y	-	0	1	0	0	0	0	0	0	0	0	rm	rm	rm	rdn	rdn	rdn
373	`ORR rdn, rm`	ORR	Y	-	0	1	0	0	0	0	1	1	0	0	rm	rm	rm	rdn	rdn	rdn
273	`EORR rdn, rm`	EORR	Y	-	0	1	0	0	0	0	0	0	0	1	rm	rm	rm	rdn	rdn	rdn
411	`NEGR rd, rn`	NEGR	Y	-	0	1	0	0	0	0	1	0	0	1	rn	rn	rn	rd	rd	rd
					15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
333	`LSLI rd, rm, #i5`	LSLI	Y	N	0	0	0	0	0	i5	i5	i5	i5	i5	rm	rm	rm	rd	rd	rd
335	`LSLR rdn, rm`	LSLR	Y	-	0	1	0	0	0	0	0	0	1	0	rm	rm	rm	rdn	rdn	rdn
337	`LSRI rd, rm, #i5`	LSRI	Y	N	0	0	0	0	1	i5	i5	i5	i5	i5	rm	rm	rm	rd	rd	rd
339	`LSRR rdn, rm`	LSRR	Y	-	0	1	0	0	0	0	0	0	1	1	rm	rm	rm	rdn	rdn	rdn
235	`ASRI rd, rm, #i5`	ASRI	Y	N	0	0	0	1	0	i5	i5	i5	i5	i5	rm	rm	rm	rd	rd	rd
					15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
347	`MOVI rd, #i8`	MOVI	Y	N	0	0	1	0	0	rd	rd	rd	i8	i8	i8	i8	i8	i8	i8	i8
363	`MOVNR rd, rm`	MOVNR	Y	-	0	1	0	0	0	0	1	1	1	1	rm	rm	rm	rd	rd	rd
349	`MOVRSP sp, rm`	MOVRSP	N	-	0	1	0	0	0	1	1	0	1	0	rm	rm	rm	1	0	1
					15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
287	`LDRI rt, [rn, #i5]`	LDRI	N	N	0	1	1	0	1	i5	i5	i5	i5	i5	rn	rn	rn	rt	rt	rt
291	`LDRR rt, [rn, rm]`	LDRR	N	-	0	1	0	1	1	0	0	rm	rm	rm	rn	rn	rn	rt	rt	rt
287	`LDRSPI rt, [sp, #i8]`	LDRSPI	N	N	1	0	0	1	1	rt	rt	rt	i8	i8	i8	i8	i8	i8	i8	i8
289	`LDRPCI rd, [pc, #i8]`	LDRPCI	N	N	0	1	0	0	1	rd	rd	rd	i8	i8	i8	i8	i8	i8	i8	i8
473	`STRI rt, [rn, #i5]`	STRI	N	N	0	1	1	0	0	i5	i5	i5	i5	i5	rn	rn	rn	rt	rt	rt
475	`STRR rt, [rn, rm]`	STRR	N	-	0	1	0	1	0	0	0	rm	rm	rm	rn	rn	rn	rt	rt	rt
473	`STRSPI rt, [sp, #i8]`	STRSPI	N	N	1	0	0	1	0	rt	rt	rt	i8	i8	i8	i8	i8	i8	i8	i8
389	`PUSH {lr}`	PUSH	N	-	1	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0
387	`POP {pc}`	POP	N	-	1	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0
					15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
239	`BU <label>`	BU	N	Y	1	1	1	0	0	i11	i11	i11	i11	i11	i11	i11	i11	i11	i11	i11
239	`B{EQ,NE,LT,GT} <label:`	BC	N	Y	1	1	0	1	con	con	con	con	i8	i8	i8	i8	i8	i8	i8	i8
248	`BL <label>`	BL1	N	Y	1	1	1	1	0	i11	i11	i11	i11	i11	i11	i11	i11	i11	i11	i11
248	*`No extra mnemonic`*	BL2	N	Y	1	1	1	1	1	i11	i11	i11	i11	i11	i11	i11	i11	i11	i11	i11
250	`BR rm`	BR	N	-	0	1	0	0	0	1	1	1	0	0	rm	rm	rm	0	0	0
503	`SVC #i8`	SVC	N	Y	1	1	0	1	1	1	1	1	i8	i8	i8	i8	i8	i8	i8	i8

Key: ***Flags*** Are the condition flags set by this instruction?

V Is the immediate to be interpreted as signed?

rn Source register 1

rm Source Register 2

rd Destination register

rnd Source and destination register

rt Target register (= destination)

iX Immediate with bit-length '***X***'

cond Condition codes: “EQ, NE, LT, GT” only need to be supported. Encodings on p.20

Note

Immediate is multiplied by 4

Immediate is multiplied by 4

Immediate is multiplied by 4

Immediate is multiplied by 4

rd<- (0 – Rn) ; = RSB rn #i where i=0

Modified MOVR

Immediate is multiplied by 4

Immediate is multiplied by 4

Immediate is multiplied by 4

Push lr to stack

Pop stack to pc

“EQ, NE, LT, GT” only need to be supported.

Modified BX with reduced register range

prefix clash with BC '11011111' means do SVC instead

8