UNIVERSITY OF REGINA ARM QUICK REFERENCE CARD

Rev 1.2 Sep 2014

ENCODINGS	31-28	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	notes or examples
Data Proc[1]	cond	0	0	I	OPCODE				S	Rn				Rd				Operand 2												See Operand 2
Multiply[2]	cond	0	0	0	0	0	0	A	S	Rd				(Rn for MLA)				Rs				1	0	0	1	Rm				Rd←Rm*Rs(+Rn)
Swap[3]	cond	0	0	0	1	0	B	0	0	Rn				Rd				0	0	0	0	1	0	0	1	Rm				Rd←M[Rn]; M[Rn]←Rm
Load/Store[4]	cond	0	1	R	P	U	B	W	LS	Rn				Rd				12-bit Unsigned Offset / Reg Sh												Reg Sh as Operand 2
Block L/S	cond	1	0	0	P	U	Sp	W	LS	Rn				Register List (15..0)																
Branch	cond	1	0	1	L	24b 2's Comp Branch Offset. << 2, then SE to 32 to add to PC																								PC is PC+8, prefetched
BX	cond	0	0	0	1	0	0	1	0	SBO[5]												0	0	0	1	Rm				T←Rm[0], PC←Rm[31:1]<<1
SWI	cond	1	1	1	1	SWI comment Field (ignored by Processor)																								

* Coprocessor transfer and halfword instructions are not included on this table

1	if Rd=R15, CPSR <- SPSR, doesn't write to R15.
2	Rd ≠ Rm; Don't use R15
3	Rd≠Rn;Rm≠Rn Don't use R15
4	Rm ≠ R15. R15 is PC+8
5	SBO = should be 1s

Operand 2	11	10	9	8	7	6	5	4	3	2	1	0	Op2 =	EG
Immediate	Align				unsigned Imm								Imm ror 2*Align	ADD Rd,Rn,#25
Imm Sh	Imm Shamt (u)					Sh*		0	Rm				Rm sh by ImmShamt	ADD Rd,Rn,Rm,LSR #4
Reg Sh	Rw				0	Sh*		1	Rm				Rm sh by Rw	ADD Rd,Rn,Rm,LSL Rw

*Sh(shift)				
00	01	10	11	ROR #0
LSL	LSR	ASR	ROR	RRX

ADDRESSING MODES
Literal
Register Indirect
Register Indirect w Offset
Autoindexing Preindexed
PC-Relative

PSEUDOINSTRUCTIONS / ALIASES	
ADR Rd, L	Rd ← address of L
LDR Rd, L	Rd ← memory at L
MOV Rd, V	RD = 32b literal V
PUSH rglst	STMDB sp!, reglist
POP rglst	LDMIA sp!, reglist
NOP	no operation
get comment field SWI	LDR R0,[R14,#-4]

CONTROL BITS		
DP M Sw	I	Op 2: imm/¬reg
	S	Write to CPSR
	A	Accumulate
Sw L/S	B	byte/¬word
L/S	R	Op 2: reg/¬imm
	P	update ptr: pre/¬post
	U	update ptr: inc/¬dec
	W	write back pointer
	LS	load/¬store
BLS	Sp	Restore PSR / banked
B	L	Link (R14 <- PC before

OPCODE		RESULT	
0000	AND	Rn AND operand2	
0001	EOR	Rn EOR operand2	
0010	SUB	Rn - operand2	
0011	RSB	operand2 - Rn	
0100	ADD	Rn + operand2	
0101	ADC	Rn + operand2 + carry	
0110	SBC	Rn - operand2 + carry - 1	
0111	RSC	operand2 - Rn + carry - 1	
1000	TST	as AND*	Result not written; CPSR written. See "test" for assembler syntax.
1001	TEQ	as EOR*	
1010	CMP	as SUB*	
1011	CMN	as ADD*	
1100	ORR	Rn OR operand2	
1101	MOV	operand2 (Rn is ignored)	
1110	BIC	Rn AND NOT operand2 (Bit clr)	
1111	MVN	NOT operand2 (Rn is ignored)	

ASSEMBLER SYNTAX	
Branch	B{L}{cond} <expression>
Move	<MOV|MVN>{cond}{S} Rd,<Op2>
Test	<opcode>{cond} Rn,<Op2>
Data	<opcode>{cond}{S} Rd,Rn,<Op2>
Mult	MUL{cond}{S} Rd,Rm,Rs
Mul/Acc	MLA{cond}{S} Rd,Rm,Rs,Rn
B L/S	<LDM|STM>{cond}<BLS*> Rn{!},<Rlist>{^}
L/S	<LDR|STR>{cond}{B} Rd,<L/S Address>
Swap	<SWP>{cond}{B} Rd,Rm,[Rn]
SWI	SWI{cond} <expression>

! = Effective Address write-back to Rn

Batch L/S <BLS*>
FD|ED|FA|EA
IA|IB|DA|DB

F	Full Stack
E	Empty stack
A	Up Stack
D	Down Stack
IA	Inc After
IB	Inc Before
DA	Dec After
DB	Dec Before
^	include CPSR

<L/S Address>		Eff. Addr.	RN after inst'n	P	W	R
Register	[Rn]	Rn + 0	Rn + 0	0	0	0
Pre-indexed	[Rn,#<exp>]{!}	Rn + <exp>	Rn {+ <exp>}	1	!	0
	[Rn,{+|-}Rm{,<sh>}]{!}	Rn ± Rm <sh>	Rn {± Rm <sh>}	1	!	1
Post-indexed	[Rn],#<exp>	Rn	Rn + <exp>	0	1	0
	[Rn],{+|-}Rm{,<sh>}	Rn	Rn ± Rm <sh>	0	1	1

	REGISTER FILE*					
alias	User	SVC	Abor	UND	IRQ	FIQ
A1	r0					
A2	r1					
A3	r2					
A4	r3					
V1	r4					
V2	r5					
V3	r6					
V4	r7					
V5	r8					r8f
V6/SB	r9					r9f
V7/SL	r10					r10f
V8/FP	r11					r11f
IP	r12					r12f
SP	r13	r13s	r13a	r13u	r13i	r13f
LR	r14	r14s	r14a	r14u	r14i	r14f
PC	r15					
PSR	CPSR	SPSRs	SPSRa	SPSRu	SPSRi	SPSRf

* chips with coprocessor have c0-c15
* chips with FPU have s0-s32 and d0-d16

CPSR/SPSR									
31	30	29	28	27	26-8	7	6	5	4-0
N	Z	C	V	Q	unused	I	F	T	mode

CPSR/SPSR BITS	
N	negative
Z	zero
C	carry
V	overflow
Q	underflow
I	interrupt mask
F	fast intrpt mask
T	Thumb/¬ARM

CONDITION		Flags	Note	CONDITION		Flags	Note
0000	EQ	Z==1	Equal	1000	HI	C==1&&Z==0	> $^{(u)}$
0001	NE	Z==0	Not Equal	1001	LS	C==0||Z==1	<= $^{(u)}$
0010	HS/CS	C==1	>= $^{(u)}$ / C=1	1010	GE	N==V	>=
0011	LO/CC	C==0	< $^{(u)}$ / C=1	1011	LT	N!=V	<
0100	MI	N==1	minus(neg)	1100	GT	Z==0&&N==V	>
0101	PL	N==0	plus(pos)	1101	LE	Z==1||N!=V	<=
0110	VS	V==1	V set(ovfl)	1110	AL	always	
0111	VC	V==0	V clr				

(u) = unsigned

VECTOR TABLE				
Address	Exception	Mode	CPSR[4:0]	Priority
0x00000000	Reset	SVC	10011	1
0x00000004	Undefined Inst	UND	11011	6
0x00000008	SWI	SVC	10011	6
0x0000000C	Prefetch Abort	Abort	10111	5
0x00000010	Data Abort	Abort	10111	2
0x00000014	*Reserved*			
0x00000018	IRQ	IRQ	10010	4
0x0000001C	FIQ	FIQ	10001	3
		user	10000	

RETURN FROM EXCEPTION	
SVC,UND	MOVS pc,r14
IRQ,FIQ	SUBS pc, R14, #4
abort: repeat	SUBS pc, R14, #8
unstack(eg r4-r0)	LDMFD r13! {r0-r4,pc}
unstack,rstr CPSR	LDMFD r13! {r0-r4,pc}^