

Opcodes' tables

Compact

This file is free to modify and re-use, with no restrictions, even commercially.
it's an OpenOffice document.

1 grab its content via: *svn export https://corkami.googlecode.com/svn/trunk/oOo/opcodes_tables_compact*
2 rezip with subdirectories as opcodes_tables_compact.ods

opcodes tables are available as compact and complete form.

Feb 21, 2012
Ange Albertini 2012 <http://corkami.com>

inspired by the work of Daniel Plohmann
Creative Commons Attribution 3.0 Unported License

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	nop	aconst_ null	iconst_*					lconst_*		fconst_*			dconst_*			
1x	*ipush b* ₁ s* ₂		ldc* *_1*_w*_2*_2*_w*_2		*load i* l* f* d* a* ¹					iload_* *_1*_2*_3			lload_*... *_0*_1			
2x	...lload_* *_2*_3		fload_* *_0*_1*_2*_3			dload_* *_0*_1*_2*_3			aload_* *_0*_1*_2*_3			*aload... i* l*				
3x	...*aload f* d* a* b* c* s*					*store i* l* f* d* a* ¹					istore_* *_0*_1*_2*_3			lstore_*... *_0		
4x	...lstore_* *_1*_2*_3			fstore_* *_0*_1*_2*_3			dstore_* *_0*_1*_2*_3			astore_* *_0*_1*_2*_3			*astore... i*			
5x	...*astore l* f* d* a* b* c* s*					pop* *_2		dup* *_x1*_x2*_2*_2*_x1*_2*_x2					swap			
6x	*add i* l* f* d*			*sub i* l* f* d*			*mul i* l* f* d*			*div i* l* f* d*						
7x	*rem i* l* f* d*			*neg i* l* f* d*			*shl i* l*		*shr i* l*		*ushr i* l*	*and i* l*				
8x	*or i* l*		*xor i* l*	iinc ₂	i2* *_i*_f*_d		l2* *_i*_f*_d		f2* *_i*_l*_d		d2*... *_i*_l*_d					
9x	...d2* *_f*_b*_c*_s	i2* *_i*_f*_d		lcmp	fcmp* *_l*_g		dcmp* *_l*_g		if* *_eq*_ne*_lt*_ge*_gt*_le			if_icmp*.. *_eq*_2				
Ax	...if_icmp* *_ne*_lt*_ge*_gt*_le					if_acmp* *_eq*_ne		goto ₂	jsr ₂	ret ₁	*switch ^v table* lookup*		*return... i* l* f* d*			
Bx	...*return a* *		*static ₂ get* put*		*field ₂ get* put*		invoke* *_virtual*_2*_special*_2*_static*_2*_interface*_4					xxunusedxxx1	new ₂	*newarray *_1*_a*_2	array length	athrow
Cx	check cast ₂	instance of ₂	monitor* *enter*exit		(pref.) wide	multi anewarray ₃		if* *_null*_nonnull		goto_w ₄	jsr_w ₄	break point				
Fx														impdep* *_1*_2		

misc	arithmetic	flow
constants	logical	method
immediates	conversion	object
locals	comparison	system
stack	conditional	undefined

JVM (Java)

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	nop	break	ldarg.* *0 *1 *2 *3				ldloc.* *0 *1 *2 *3				stloc.* *0 *1 *2 *3			ldarg* *.s *.a.s		
1x	starg.s	ldloc *.s *.a.s		stlocl.s	ldnull	*m1	ldc.i4.* *0 *1 *2 *3 *4 *5 *6 *7 *8 *.s									
2x	ldc.* *i4 *i8 *r4 *r8				dup	pop	jmp	call* * *i		ret	br* *.s *false.s *true.s			b*.s... eq ge		
3x	...b*.s			b*.un.s				br			b* *eq *ge *gt *le *lt					
4x	b*.un ne ge gt le lt				switch	ldind.* ... *i1 *u1 *i2 *u2 *i4 *u4 *i8 *j *r4 *r8										
5x	...ldind.* *ref	stind.* *ref *i1 *i2 *i4 *i8 *r4 *r8				add	sub	mul	div* * *.un		rem* * *.un		and			
6x	or	xor	shl	shr * *.un		neg	not	conv.* *i1 *i2 *i4 *i8 *r4 *r8 *u4 *u8							callvirt	
7x	cpobj	ldobj	ldstr	newobj	cast class	isinst	conv.r.un	unbox		throw	ldfld* * *.a		stfld	ldsfld* * *.a		
8x	stsfld	stobj	conv.ovf.i*.un 1 2 4 8				conv.ovf.u*.un 1 2 4 8				conv.ovf.*.un i u		box	newarr	ldlen	ldelema
9x	ldelem.* *i1 *u1 *i2 *u2 *i4 *u4 *i8 *j *r4 *r8 *ref											stelem.*... *i *i1 *i2 *i4 *i8				
Ax	...stelem.* *r4 *r8 *ref			ldelem	stelem	unbox.any										
Bx	conv.ovf.* *i1 *u1 *i2 *u2 *i4 *u4 *i8 *u8															
Cx	ref anyval		ck finite	mkref any												
Dx	ldtoken	conv.* *u2 *u1		conv.ovf.* *i *u		add.ovf* * *.un		mul.ovf* * *.un		sub.ovf* * *.un		end finally	leave* * *.s		stind.i	
Ex	conv.u															
FE																
0x	arglist	*eq	*gt	c* *gt.un *lt *lt.un		ld*ftn * virt		ldarg* * *.a		starg	ldloc* * *.a		stloc	localloc		
1x	end filter	un aligned.	volatile.	tail.	init obj	constrained.	cpblk	init blk	no.	re throw	sizeof	ref anytype	read only.			

misc	immediates	arithmetic	fields	prefix
system	stack	logical	array	
args	method	conversion	references	
locals	conditional	object	flow	
constants	indirects	exception	comparison	undefined

Common Intermediate Language (.Net)

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF		
0x	nop	move*		move-wide*			move-object*			move-result*		move-exception	return*					
		*	/from16	/16	*	/from16	/16	*	/from16	/16	*	-wide	-object		-void	*		
1x	return*		const						const-wide			const-string	const-class	monitor		check		
	-wide	-object	/4	/16	*	/high16	/16	/32	*	/high16	*	-jumbo		enter	exit	cast		
2x	instance of	array length	new*		filled-new-array		fill-array-data	throw	goto			*-switch		cmp*-float		cmp*-double		
			*instance	*array	*	-range			*	/16	/32	packed*	sparse*	l	g	l		
3x	cmp*-double		cmp-long		if-*				if-*z									
	g	long	eq	ne	lt	ge	gt	le	eq	ne	lt	ge	gt	le				
4x					aget						aput...							
					*	-wide	-object	-bool	-byte	-char	-short	*	-wide	-object	-bool	-byte		
5x	...aput		iget						iput									
	-char	-short	*	-wide	-object	-bool	-byte	-char	-short	*	-wide	-object	-bool	-byte	-char	-short		
6x					sget				sput						invoke-*...			
	*	-wide	-object	-bool	-byte	-char	-short	*	-wide	-object	-bool	-byte	-char	-short	virtual	super		
7x	...invoke-*			invoke-*/range						neg*		not*	neg*	not*	neg*			
	-direct	-static	-interface		virtual	super	-direct	-static	-interface		-int	-int	-long	-long	-float			
8x	...neg*		int-to-*		long-to-*		float-to			double-to-*		int-to-*						
	-double	long	float	double	int	float	double	int	long	double	int	long	float	byte	char	short		
9x											*-int						*-long...	
	add	sub	mul	div	rem	and	or	xor	shl	shr	ushr	add	sub	mul	div	rem		
Ax	...*-long						*-float						*-double					
	and	or	xor	shl	shr	ushr	add	sub	mul	div	rem	add	sub	mul	div	rem		
Bx											*-int/2addr						*-long/2addr...	
	add	sub	mul	div	rem	and	or	xor	shl	shr	ushr	add	sub	mul	div	rem		
Cx	...*-long/2addr						*-float/2addr						*-double/2addr					
	and	or	xor	shl	shr	ushr	add	sub	mul	div	rem	add	sub	mul	div	rem		
Dx									*-int/lit16								*-int/lit8	
	add	sub	mul	div	rem	and	or	xor	add	sub	mul	div	rem	and	or	xor		
Ex					*-int/lit8												execute	inline
	shl	shr	ushr															
Fx	invoke-direct-empty		iget-*quick				iput-*quick			invoke-*quick								
			*	-wide	-object	*	-wide	-object	virtual	virtual/range	super	super/range						

misc	object	conversion
moves	flow	arithmetic
method	conditional	
literals	transfer	
system	logical	undefined

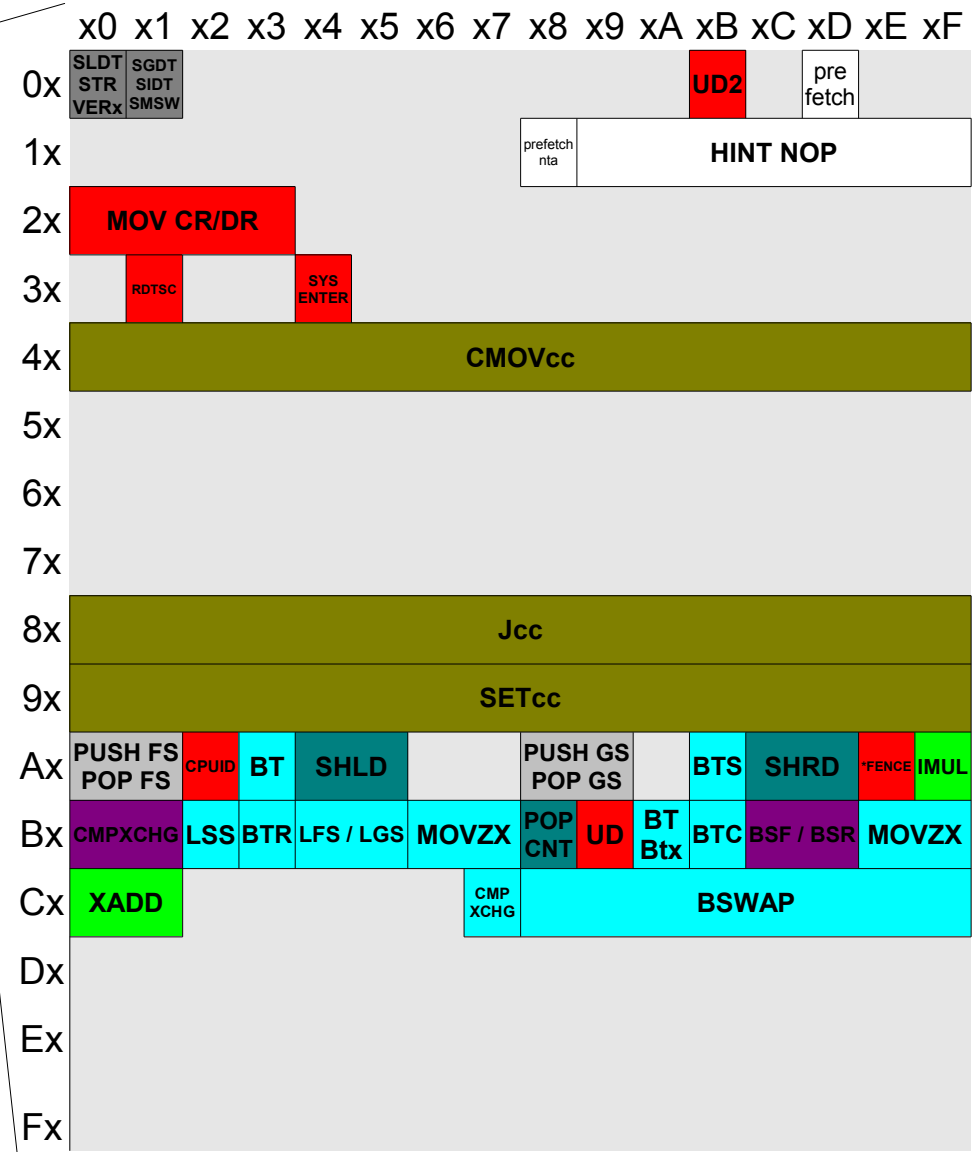
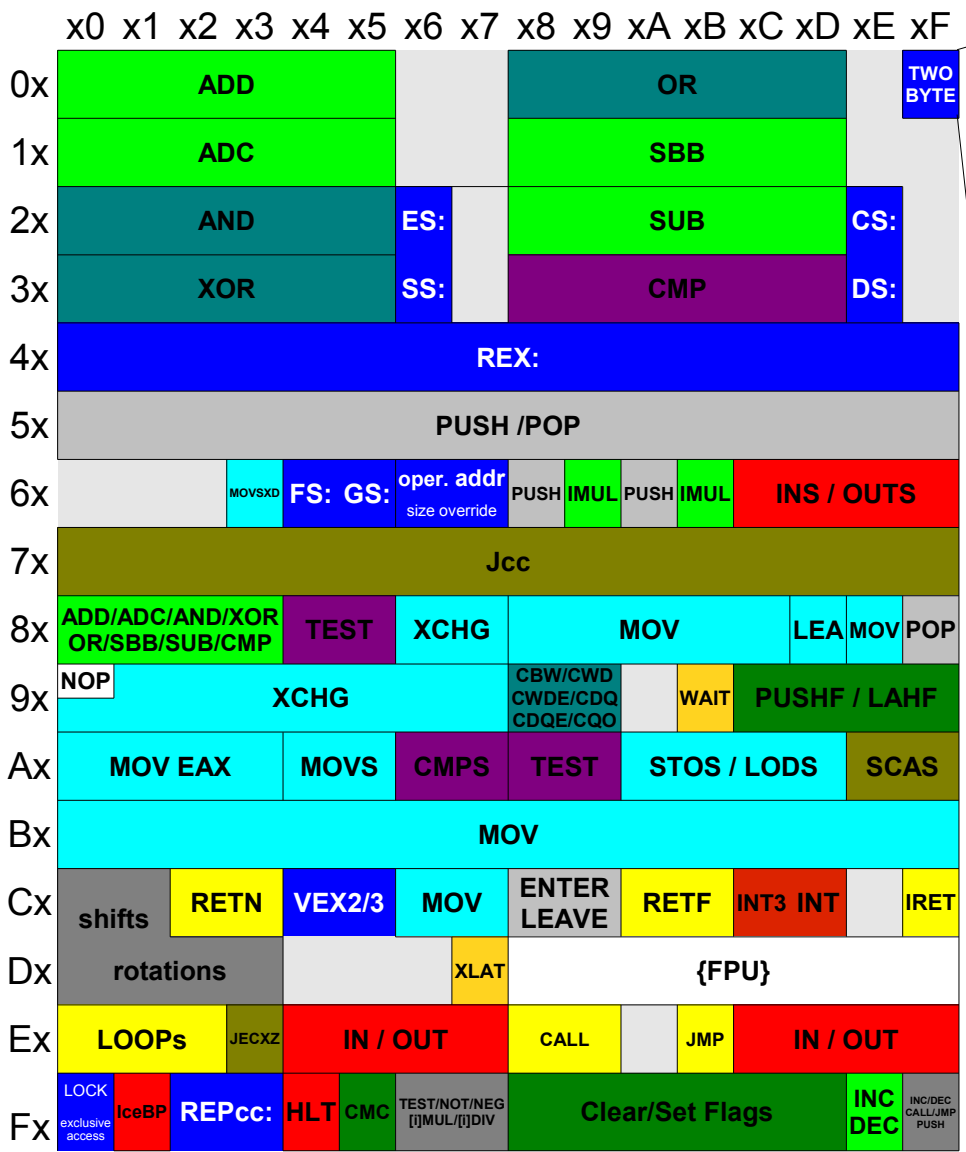
Dalvik Virtual Machine (android)

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF	
0x	ADD				PUSH/POP ES/SS		OR				PUSH CS	TWO BYTE					
1x	ADC						SBB				PUSH DS		POP DS				
2x	AND				ES: DAA		SUB				CS: DAS						
3x	XOR				SS: AAA		CMP				DS: AAS						
4x	INC / DEC																
5x	PUSH / POP																
6x	PUSHA	POPA	BOUND	ARPL	FS: GS: oper. addr size override		PUSH	IMUL	PUSH	IMUL	INS / OUTS						
7x	Jcc																
8x	ADD/ADC/AND/XOR OR/SBB/SUB/CMP		TEST		XCHG		MOV				LEA MOV POP						
9x	NOP		XCHG				CBW/CWD CWDE/CDQ		CALL	WAIT	PUSHF / LAHF						
Ax	MOV EAX		MOVS		CMPS		TEST		STOS / LODS			SCAS					
Bx	MOV																
Cx	shifts		RETN		LES / LDS		MOV		ENTER LEAVE		RETF		INT3 INT INTO		IRET		
Dx	rotations		AAM AAD		SALC XLAT		{FPU}										
Ex	LOOPS		JECXZ		IN / OUT				CALL / JMP				IN / OUT				
Fx	LOCK exclusive access	IceBP	REPcc:		HLT CMC		TEST/NOT/NEG [i]MUL/[i]DIV				Clear/Set Flags				INC DEC		INC/DEC CALL/JMP PUSH

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF				
0x	SLDT STR VERx	SGDT SIDT SMSW					UD2		pre fetch											
1x	prefetch nta		HINT NOP																	
2x	MOV CR/DR																			
3x	RDTSC				SYS ENTER															
4x	CMOVcc																			
5x																				
6x																				
7x																				
8x	Jcc																			
9x	SETcc																			
Ax	PUSH FS POP FS		CPUID		BT		SHLD				PUSH GS POP GS		BTS		SHRD		*FENCE	IMUL		
Bx	CMPXCHG		LSS		BTR		LFS / LGS		MOVZX		POP CNT		UD		BT Btx		BTC		BSF / BSR	MOVZX
Cx	XADD						CMP XCHG		BSWAP											
Dx																				
Ex																				
Fx																				

misc	arithmetic	flow
memory/reg	logical	group
	flags	prefix
obsolete	comparison	system
stack	conditional	hidden

usermode only, no extra instruction set



misc	arithmetic	flow
memory/reg	logical	group
	flags	prefix
obsolete	comparison	system
stack	conditional	hidden

usermode only, no extra instruction set

Android 1 <http://source.android.com/tech/dalvik/dalvik-bytecode.html>
2 http://pallergabor.uw.hu/androidblog/dalvik_opcodes.html

.Net 1 <http://www.ecma-international.org/publications/standards/Ecma-335.htm> p355-469
2 <http://www.asukaze.net/etc/cil/opcode.html>

Java http://java.sun.com/docs/books/jvms/second_edition/html/Instructions.doc.html

0x

x0

Sub**L**eq

SubLeq