MikroLeo Instruction Set

Encoding of MikroLeo Instruction Word (16-bits)						Mnemonic	Instructions with all possibilities	Operation	Affected Flags
ROMH (8-bit)				ROML (8-bit)					
High Nibble (HiNB)			Decoder Address	MAddr	Operand/ LAddr				
MICRO2_IN	AMODE	MOD	MICRO						
b15	b14	b13:b12	b11:b8	b7:b4	b3:b0				
0	x	0	0	x	n	LDI	LDI ACC,n	ACC ← n	ZF
		1					LDI RA,n	RA ← n	-
		2					LDI RB,n	RB ← n	
		3					LDI RC,n	RC ← n	
	x	0	1	x	n	NAND	NAND ACC,n	ACC ← ACC NAND n	ZF
		1			x		NAND ACC,RA	ACC ← ACC NAND RA	
		2					NAND ACC,RB	ACC ← ACC NAND RB	
	0	3		MAddr	LAddr		NAND ACC,RAM[Addr]	ACC ← ACC NAND RAM[RC:MAddr:LAddr]	
	1			x	x			ACC ← ACC NAND RAM[RC:RB:RA]	
	0	0	2	MAddr	LAddr	LDW	LDW ACC,RAM[Addr]	ACC ← RAM[RC:MAddr:LAddr]	ZF
	1			x	x			ACC ← RAM[RC:RB:RA]	
	x	1	3	x	x	LDA	LDA RA	ACC ← RA	ZF
		2					LDA RB	ACC ← RB	
		3					LDA RC	ACC ← RC	
		0	4	x	n	OUTA	OUTA n	OUTA ← n	-
		1			x		OUTA ACC	OUTA ← ACC	
		2					OUTA RA	OUTA ← RA	
	0	3		MAddr	LAddr		OUTA RAM[Addr]	OUTA ← RAM[RC:MAddr:LAddr]	
	1			x	x			OUTA ← RAM[RC:RB:RA]	
	x	0	5	x	n	OUTB	OUTB n	OUTB ← n	-
		1			x		OUTB ACC	OUTB ← ACC	
		2					OUTB RA	OUTB ← RA	
	0	3		MAddr	LAddr		OUTB RAM[Addr]	OUTB ← RAM[RC:MAddr:LAddr]	
	1			x	x			OUTB ← RAM[RC:RB:RA]	
	x	0	6	x	n	OUTC	OUTC n	OUTC ← n	-
		1			x		OUTC ACC	OUTC ← ACC	
		2					OUTC RA	OUTC ← RA	
	0	3		MAddr	LAddr		OUTC RAM[Addr]	OUTC ← RAM[RC:MAddr:LAddr]	
	1			x	x			OUTC ← RAM[RC:RB:RA]	
	x	1	7	x	x	LDR	LDR RA	RA ← ACC	-
		2					LDR RB	RB ← ACC	
		3					LDR RC	RC ← ACC	
		0	8	x	n	CMP	CMP ACC,n	ACC - n	CF, ZF
		1			x		CMP ACC,RA	ACC - RA	
		2					CMP ACC,RB	ACC - RB	
	0	3		MAddr	LAddr		CMP ACC,RAM[Addr]	ACC - RAM[RC:MAddr:LAddr]	
	1			x	x			ACC - RAM[RC:RB:RA]	
	x	0	9	x	n	OUTD	OUTD n	OUTD ← n	-
		1			x		OUTD ACC	OUTD ← ACC	
		2					OUTD RA	OUTD ← RA	
	0	3		MAddr	LAddr		OUTD RAM[Addr]	OUTD ← RAM[RC:MAddr:LAddr]	
	1							OUTD ← RAM[RC:RB:RA]	
	0	0	Ah	MAddr	LAddr	STW	STW RAM[Addr],ACC	RAM[RC:MAddr:LAddr] ← ACC	-
	1			x	x			RAM[RC:RB:RA] ← ACC	
	x	0	Bh	x	n	SUB	SUB ACC,n	ACC ← ACC - n	CF, ZF
		1			x		SUB ACC,RA	ACC ← ACC - RA	
		2					SUB ACC,RB	ACC ← ACC - RB	
	0	3		MAddr	LAddr		SUB ACC,RAM[Addr]	ACC ← ACC - RAM[RC:MAddr:LAddr]	
	1			x	x			ACC ← ACC - RAM[RC:RB:RA]	
	0	0	Ch	MAddr	LAddr	JPI	JPI RC:MA[7:4]:LA[3:0]	PC ← [RC:MAddr:LAddr]	-
	1			x	x			PC ← [RC:RB:RA]	
	0	0	Dh	MAddr	LAddr	JPC	JPC MA[7:4]:LA[3:0]	If CF=1, PC ← [PCH:MAddr:LAddr]	
	1			x	x			If CF=1, PC ← [PCH:RB:RA]	
	0	0	Eh	MAddr	LAddr	JPZ	JPZ MA[7:4]:LA[3:0]	If ZF=1, PC ← [PCH:MAddr:LAddr]	
	1			x	x			If ZF=1, PC ← [PCH:RB:RA]	
	x	0	Fh	x	n	ADD	ADD ACC,n	ACC ← ACC + n	CF, ZF
		1			x		ADD ACC,RA	ACC ← ACC + RA	
		2					ADD ACC,RB	ACC ← ACC + RB	
	0	3		MAddr	LAddr		ADD ACC,RAM[Addr]	ACC ← ACC + RAM[RC:MAddr:LAddr]	
	1			x	x			ACC ← ACC + RAM[RC:RB:RA]	
1	x	0	x			INA	INA	ACC ← INA	ZF
		1				INB	INB	ACC ← INB	
		2				INC	INC	ACC ← INC	
		3				IND	IND	ACC ← IND	

LA[3:0] => can be RA or Operand (LAddr), depends on the AMODE bit.
MA[7:4] => can be RB or MAddr, depends on the AMODE bit.
Address = Addr => RC:MA[7:4]:LA[3:0]
If AMODE=0, Addr=RC:Maddr:LAddr
If AMODE=1, Addr=RC:RB:RA
x = don't care
MAddr = m, so the names are interchangeable
Operand = n, so the names are interchangeable

AMODE = Addresing mode (b14)
MOD = Modifier bits (b13:b12)
MAddr = Medium Address (b7:b4)
LAddr = Low Address (b3:b0)
MICRO = Instruction (b11:b8)
OPCODE = AMODE:MOD:MICRO