DME CPU - ISA (based on Magic1.6)

		Bit														
OPCODE	**Descr**	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ldi tgt, s10	load s10b immediate	0	0	0	signed 10-bit imm										tgt		
br s13	br s13b offset	0	0	1	signed 13-bit pc relative offset												
	not used atm	0	1	0	not used												
	not used atm	0	1	1	not used												
ldw tgt, sw7(BP)	load word from base BP with s7b offset into tgt2	1	OPCODE						7-bit signed							tgt2	
ldb sb7(r5),tgt	load byte from base BP with s7b offset into tgt2	1	OPCODE						7-bit signed							tgt2	
stw sw7(r5), src	store word to base BP with s7b offset from tgt2	1	OPCODE						7-bit signed							tgt2	
stb src,sw7(r5)	store byte to base BP with s7b offset from tgt2	1	OPCODE						7-bit signed							tgt2	
stw0 sw7(r5), r0	store word to base BP with s7b offset from r0	1	OPCODE						7-bit signed							tgt2	
stb0	store byte to base BP with s7b offset from r0	1	OPCODE						7-bit signed							tgt2	
add op1,op2,res	ADD op1 to op2 and store in tgt	1	OPCODE						op1			op2				tgt	
sub op1,op2,res	SUB op1 from op2 and store in tgt	1	OPCODE						op1			op2				tgt	
and op1,op2,res	AND op1 and op2 and store in tgt	1	OPCODE						op1			op2				tgt	
or op2,op2,res	OR op1 and op2 and store in tgt	1	OPCODE						op1			op2				tgt	
skip.c op1,op2, cond	skip next instr if op1 and op2 meet cond	1	OPCODE						op1			op2				cond	
addskp.z op1,op2,res	tgt = op1 + op 2; if 0 then skip next instr	1	OPCODE						op1			op2				tgt	
addskp.nz op1,op2,res	tgt = op1 + op 2; if !0 then skip next instr	1	OPCODE						op1			op2				tgt	
add imm,op2,res	ADD imm to op2 and store in tgt	1	OPCODE						imm			op2				tgt	
sub imm,op2,res	SUB imm from op2 and store in tgt	1	OPCODE						imm			op2				tgt	
and imm,op2,res	AND imm and op2 and store in tgt	1	OPCODE						imm			op2				tgt	
or imm,op2,res	OR imm and op2 and store in tgt	1	OPCODE						imm			op2				tgt	
skip.c imm,op2, cond	skip next instr if imm and op2 meet cond	1	OPCODE						imm			op2				cond	
addskp.zimm imm,op2,res	tgt = imm + op 2; if 0 then skip next instr	1	OPCODE						imm			op2				tgt	
addskp.nzimm imm,op2,res	tgt = imm + op 2; if !0 then skip next instr	1	OPCODE						imm			op2				tgt	
ldwb (base,idx),tgt	load word from base + offset idx into tgt	1	OPCODE						base			idx				tgt	
ldbb (base,idx),tgt	load byte from base + offset idx into tgt	1	OPCODE						base			idx				tgt	
stwb src,(base,idx)	store word to base + offset idx from tgt	1	OPCODE						base			idx				tgt	
stbb src,(base,idx)	store byte to base + offset idx from tgt	1	OPCODE						base			idx				tgt	
sext reg, reg	sign extend src into tgt	1	OPCODE						src			not used				tgt	
not used		1	OPCODE														
addih u7,res	add u7b as 7 high bits into reg tgt2	1	OPCODE						7-bit unsigned							tgt2	
push src	push value from reg tgt to RAM address SP	1	OPCODE						not used							tgt	
pop tgt	load reg tgt with value at RAM address SP	1	OPCODE						not used							tgt	
br.r tgt	br to address in reg tgt	1	OPCODE						not used							tgt	
syscall	raise trap SYSCALL	1	OPCODE						not used								
reti	return to user space	1	OPCODE						not used								
push.u	push user bank reg tgt to stack	1	OPCODE						not used							tgt	
brk	break cpu execution (for debugging)	1	OPCODE						not used								
lcr tgt	load control reg to tgt	1	OPCODE						not used							tgt	
wcr src	write tgt to control reg	1	OPCODE						not used							tgt	
wpte lpage, ppage	write pte logical page, physical page	1	OPCODE						lpage			ppage				not used	
lpte lpage	load pte logical page into tgt	1	OPCODE						lpage			not used				tgt	
wptb	write tgt to PTB register	1	OPCODE						tgt			not used					
lptb	load PTB to tgt	1	OPCODE						not used							tgt	
wivec	write tgt to interrupt vector register	1	OPCODE						not used							tgt	
shl cnt,src,tgt	shift src left count places, result in tgt	1	OPCODE						count				src			tgt2	
shr cnt,src,tgt	shift src right count places, result in tgt	1	OPCODE						count				src			tgt3	
shl.r src, cnt, tgt	shift src left reg places, result in tgt	1	OPCODE						reg			src				tgt	
shr.r src, cnt, tgt	shift src right reg places, result in tgt	1	OPCODE						reg			src				tgt	
pop.u	pop stack value into user reg tgt	1	OPCODE						not used							tgt	
lcr.u	load user bank cr	1	OPCODE						not used							tgt	
scr.u	store user bank cr	1	OPCODE						not used							tgt	
rsi	reset all traps in irq encoder	1	OPCODE						not used								