Table 17: X-Makina Instruction Set (Light-grey denotes opcode bits)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Mnemonic	Instruction
1	0	0	0	0	PRPO	DEC	INC	0	W/B	S	S	S	D	D	D	**LD**	Load DST from mem[SRC plus addressing]
1	0	0	0	1	PRPO	DEC	INC	0	W/B	S	S	S	D	D	D	**ST**	Store SRC in mem[DST plus addressing]
1	1	0	OFF	OFF	OFF	OFF	OFF	OFF	W/B	S	S	S	D	D	D	**LDR**	Load DST from mem[SRC + sign-extended offset]
1	1	1	OFF	OFF	OFF	OFF	OFF	OFF	W/B	S	S	S	D	D	D	**STR**	Store SRC in mem[DST + sign-extended offset]
1	0	0	1	0	B	B	B	B	B	B	B	B	D	D	D	**MOVL**	DST.Low byte ← BBBBBBBB; High byte unchanged
1	0	0	1	1	B	B	B	B	B	B	B	B	D	D	D	**MOVLZ**	DST.Low byte ← BBBBBBBB; Zero high byte
1	0	1	0	0	B	B	B	B	B	B	B	B	D	D	D	**MOVH**	DST.High byte ← BBBBBBBB ; Low byte unchanged
0	0	0	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	**BL**	Branch with Link
0	0	1	0	0	0	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	**BEQ/BZ**	Branch if equal or zero
0	0	1	0	0	1	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	**BNE/BNZ**	Branch if not equal or not zero
0	0	1	0	1	0	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	**BC/BHS**	Branch if carry/higher or same (unsigned)
0	0	1	0	1	1	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	**BNC/BLO**	Branch if no carry/lower (unsigned)
0	0	1	1	0	0	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	**BN**	Branch if negative
0	0	1	1	0	1	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	**BGE**	Branch if greater or equal (signed)
0	0	1	1	1	0	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	**BLT**	Branch if less (signed)
0	0	1	1	1	1	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	**BAL**	Branch Always (unconditionally)
0	1	1	0	0	0	0	0	R/C	W/B	S/C	S/C	S/C	D	D	D	**ADD**	Add: DST ← DST + SRC/CON
0	1	1	0	0	0	1	0	R/C	W/B	S/C	S/C	S/C	D	D	D	**ADDC**	Add: DST ← DST + (SRC/CON + Carry)
0	1	1	0	0	1	0	0	R/C	W/B	S/C	S/C	S/C	D	D	D	**SUB**	Subtract: DST ← DST + (¬SRC/CON + 1)
0	1	1	0	0	1	1	0	R/C	W/B	S/C	S/C	S/C	D	D	D	**SUBC**	Subtract: DST ← DST + (¬SRC/CON + Carry)
0	1	1	0	1	0	0	0	R/C	W/B	S/C	S/C	S/C	D	D	D	**DADD**	Decimal add: DST ← DST + (SRC/CON + Carry)
0	1	1	0	1	0	1	0	R/C	W/B	S/C	S/C	S/C	D	D	D	**CMP**	Compare: DST – SRC/CON
0	1	1	0	1	1	0	0	R/C	W/B	S/C	S/C	S/C	D	D	D	**XOR**	Exclusive or: DST ← DST ⊕ SRC/CON
0	1	1	0	1	1	1	0	R/C	W/B	S/C	S/C	S/C	D	D	D	**AND**	Logical AND: DST ← DST & SRC/CON
0	1	1	1	0	0	0	0	R/C	W/B	S/C	S/C	S/C	D	D	D	**BIT**	Bit test: DST & SCR/CON
0	1	1	1	0	0	1	0	R/C	W/B	S/C	S/C	S/C	D	D	D	**BIC**	Bit clear: DST ← DST & ~SRC/CON
0	1	1	1	0	1	0	0	R/C	W/B	S/C	S/C	S/C	D	D	D	**BIS**	Bit set: DST ← DST | SRC/CON
0	1	1	1	0	1	1	0	R/C	W/B	S/C	S/C	S/C	D	D	D	**MOV**	DST ← SRC/CON
0	1	1	1	1	0	0	0	0	0	S	S	S	D	D	D	**SWAP**	Swap SRC and DST
0	1	1	1	0	0	0	1	0	W/B	0	0	0	D	D	D	**SRA**	Shift DDD right (1 bit) arithmetic
0	1	1	1	0	0	1	1	0	W/B	0	0	0	D	D	D	**RRC**	Rotate DDD right (1 bit) through carry
0	1	1	1	0	1	0	1	0	0	0	0	0	D	D	D	**SWPB**	Swap bytes in DDD
0	1	1	1	0	1	1	1	0	0	0	0	0	D	D	D	**SXT**	Sign extend byte to word in DDD