15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Mnemonic	Instruction	
0	0	0	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	BL	Branch with Link	FLOW-OF-CONTROL
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	BRA	Branch Always	
0	1	0	0	0	0	0	0	R/C	W/B	S/C	S/C	S/C	D	D	D	ADD	Add: DST = DST + SRC/CON	REG/CON-REG
0	1	0	0	0	0	0	1	R/C	W/B	S/C	S/C	S/C	D	D	D	ADDC	Add: DST = DST + (SRC/CON + Carry)	
0	1	0	0	0	0	1	0	R/C	W/B	S/C	S/C	S/C	D	D	D	SUB	Subtract: DST = DST + (¬SRC/CON + 1)	
0	1	0	0	0	0	1	1	R/C	W/B	S/C	S/C	S/C	D	D	D	SUBC	Subtract: DST = DST + (¬SRC/CON + Carry)	
0	1	0	0	0	1	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	0	0	0	1	0	1	R/C	W/B	S/C	S/C	S/C	D	D	D	CMP	Compare: DST – SRC/CON	
0	1	0	0	0	1	1	0	R/C	W/B	S/C	S/C	S/C	D	D	D	XOR	Exclusive OR: DST = DST ⊕ SRC/CON	
0	1	0	0	0	1	1	1	R/C	W/B	S/C	S/C	S/C	D	D	D	AND	AND: DST = DST & SRC/CON	
0	1	0	0	1	0	0	0	R/C	W/B	S/C	S/C	S/C	D	D	D	OR	OR: DST = DST | SRC/CON	
0	1	0	0	1	0	0	1	R/C	W/B	S/C	S/C	S/C	D	D	D	BIT	Bit test: DST & (1 << SCR/CON)	
0	1	0	0	1	0	1	0	R/C	W/B	S/C	S/C	S/C	D	D	D	BIC	Bit clear: DST = DST & ~(1 << SRC/CON)	
0	1	0	0	1	0	1	1	R/C	W/B	S/C	S/C	S/C	D	D	D	BIS	Bit set: DST = DST | (1 << SRC/CON)	
0	1	0	0	1	1	0	0	0	W/B	S	S	S	D	D	D	MOV	DST = SRC	
0	1	0	0	1	1	0	0	1	0	S	S	S	D	D	D	SWAP	Swap SRC and DST	
0	1	0	0	1	1	0	1	0	W/B	0	0	0	D	D	D	SRA	Shift DDD right (1 bit) arithmetic	REG/CON-REG
0	1	0	0	1	1	0	1	0	W/B	0	0	1	D	D	D	RRC	Rotate DDD right (1 bit) through carry	
0	1	0	0	1	1	0	1	0	0	0	1	1	D	D	D	SWPB	Swap bytes in DDD	
0	1	0	0	1	1	0	1	0	0	1	0	0	D	D	D	SXT	Sign extend byte to word in DDD	
0	1	0	0	1	1	0	1	1	0	0	0	0	PR	PR	PR	SETPRI	Set current priority	CPU
0	1	0	0	1	1	0	1	1	0	0	1	SA	SA	SA	SA	SVC	Control passes to address specified in vector[SA]	
0	1	0	0	1	1	0	1	1	0	1	V	SLP	N	Z	C	SETCC	Set PSW bits (1 = set)	
0	1	0	0	1	1	0	1	1	1	0	V	SLP	N	Z	C	CLRCC	Clear PSW bits (1 = clear)	
0	1	0	1	0	0	C	C	C	C	T	T	T	F	F	F	CEX	Conditional execution	
0	1	0	1	1	0	PRPO	DEC	INC	W/B	S	S	S	D	D	D	LD	DST = mem[SRC plus addressing]	LD-ST
0	1	0	1	1	1	PRPO	DEC	INC	W/B	S	S	S	D	D	D	ST	mem[DST plus addressing] = SRC	
0	1	1	0	0	B	B	B	B	B	B	B	B	D	D	D	MOVL	DST.Low byte = BBBBBBBB; DST.High byte unchanged	CHG REG
0	1	1	0	1	B	B	B	B	B	B	B	B	D	D	D	MOVLZ	DST.Low byte = BBBBBBBB; DST.High byte = 00000000	
0	1	1	1	0	B	B	B	B	B	B	B	B	D	D	D	MOVLS	DST.Low byte = BBBBBBBB; DST.High byte = 11111111	
0	1	1	1	1	B	B	B	B	B	B	B	B	D	D	D	MOVH	DST.Low byte unchanged; DST.High byte = BBBBBBBB	
1	0	OFF	OFF	OFF	OFF	OFF	OFF	OFF	W/B	S	S	S	D	D	D	LDR	DST = mem[SRC + sign-extended 7-bit offset]	LD-ST
1	1	OFF	OFF	OFF	OFF	OFF	OFF	OFF	W/B	S	S	S	D	D	D	STR	mem[DST + sign-extended 7-bit offset] = SRC	