ACCUMULATOR, ARITHMETIC & LOGIC INSTRUCTIONS

Mnemonic	Description	words	Cycle	Opcode
ABS	Absolute Add Value of ACC	1	1	1011 1110 0000 000D
ADD	Add to acc with shift of 0 to 15 direct or indirect	1	1	0010 SHFT 1AAA AAAA
	- Add to ACC with shift 0 to 15, Long Immediate	2	2	1011 1111 1001 SHFT +1 word
	- Add to ACC with shift of 16 direct or indirect	1	1	0110 0001 1AAA AAA
	Add to ACC, short immediate	1	1	1011 1000 IIII IIII
ADDC	Add to ACC with carry, direct or indirect	1	1	0110 0000 1AAA AAAA
ADDS	Add to Low ACC with Sign - extension suppressed	1	1	0110 0010 1AAA AAAA
ADDT	Add to ACC with shift (0→15) specified by TREG, direct or Indirect	1	1	
AND	AND ACC with data value direct, or indirect	1	1	
	AND with ACC shift of 0 to 15 Long Immediate	2	2	

CMPL Complement ACC

LACC Load ACC with shift of 0 to 15

Load ACC will shift of 0 to 15, Long immediate

LACL — Load Low word of ACC, direct or indirect

LACT — Load ACC with shift (0 to 15) specified by TREG direct or Indirect

NEG — Negate ACC

OR — OR ACC with data value, direct or Indirect

OR with ACC with shift of 0 to 15, Long immediate

ROL — Rotate ACC left

ROR — Rotate ACC right

SACH — Store ~~two~~ high ACC with shift of 0 to 7 direct or indirect

SACL — Store Low ACC with shift of 0 to 7 direct or

SFL — Shift ACC left

SFR — Shift ACC right

SUB — Subtract from ACC with shift of 0 to 15 direct or indirect

SUBB — Subtract from ACC with borrow, direct or indirect

SUBC — Conditional subtract, direct

SUBS — Subtract from ACC with Sign-extension Suppressed, direct or Indirect

SUBT — Subtract from ACC with shift (0 to 15) specified by TREG, direct or indirect

XOR — Exclusive OR ACC with data value, direct
Exclusive OR with ACC with shift of 0 to 15
Long immediate
Exclusive OR with ACC with shift of 16,
Long Immediate

ZALR Zero Low ACC and Load high ACC with rounding, direct or indirect

ADRK Add Constant to Current AR short immd

BANZ Branch on Current AR not 0, direct
Cycle — 4 (Conditional true, 2 conditional - false

CMPR — Compare Current AR with ARO

LAR — Load specified AR from specified data Location, direct or indirect
* Load specified AR with Constant
Long Immediate
*

MAR Modify Current AR and/OR ARP, Indirect (performs no operation when direct)

SAR — Store specified AR to specified data location, direct or indirect

SBRK — Subtract Constant from Current AR, short immediate

APAC — Add PREG to ACC

LPH — Load high PREG, direct or indirect

LT — Load TREG, direct or indirect

LTA — Load TREG and accumulate previous product, direct or indirect

LTD — Load TREG, accumulate previous product, and move data, direct or indirect

LTS — Load TREG and subtract previous product

MAC — Multiply and accumulate, direct or indirect

MACD — Multiply and accumulate with data move

MPY — Multiply TREG by data value, direct or indirectMultiply TREG by 13-bit constant, short imm

MPYA — Multiply and accumulate previous product

MPYS — Multiply and subtract previous

MPYU — Multiply unsigned direct or indirect

PAC — Load ACC with PREG

SPAC — Subtract PREG from ACC

SPH — Store high PREG, direct or indirect

SPL — Store Low PREG, direct or indi

SPM — Set product shift mode

SQRA — Square and accumulate previous product

SQRS — Square and Subtract previous product

B — Branch unconditionally, indirect (2) (4)

BACC — Branch to address specified by ACC (1) (4)

BANZ — Branch on Current AR not 0, indirect

BCND — Branch Conditionally (2) (4)

CALA — Call Subroutine at Location specified by ACC

CALL — Call Subroutine, indirect (2) (4)

CC — Call Conditionally (2) (4)

INTR — Soft interrupt

NMI — Nonmaskable interrupt

RET — Return from Subroutine

RETC — Returns Conditionally

TRAP — Software interrupt (1) (4)

BIT — Test bit, direct or indirect

BITT — Test bit specified by TREG, direct or indirect

CLRC — clear C bit
clear CNF bit
clear INTM bit
clear OVM bit
clear SXM bit
Clear TC bit
clear XF bit

IDLE — Idle until interrupt

LDP — Load data page pointer
Load data page pointer, short immediate

LST — Load status register ST0, direct or Indirect
ST1

NOP — No operation

POP — POP top of stack to Low ACC

POPD — pop top of stack to data

PSHD — Push data memory value on stack, direct

PUSH — Push Low ACC onto stack

RPT — Repeat next instruction

SETC — Set C bit
Set CNF bit

SPM — Set product shift mode

SST – store status register STO, direct
STI

BLDD – Block move from data memory to data memory

BLPD – Block move from program memory to data memory

DMOV – Data move in data memory

IN – Input data from I/O location

OUT – output data to port, direct or indirect

SPLK – Store Long immediate to data memory Location

TBLR – Table read, direct

TBLW – Table write, direct or indirect

ACC — The accumulator

AR — The auxiliary register

ARX — A 3-bit value used in the LAR and SAR instructions to designate which auxiliary register will be Loaded (LAR) or have its contents stored (SAR)

BITX — A 4-bit value (Called the bit code) that determines which which bit of a designated data memory value will be tested by the BIT instruction

CM — A 2-bit value. The CMPR instruction performs a comparison specified by the value of CM

— if CM = 00, test whether Current AR = ARO

if CM = 01, test whether Current AR < ARO

if CM = 10, test whether Current AR > ARO

if CM = 11, test whether Current AR ≠ ARO

INTR# — A 5-bit value representing a number from 0 to 31. The INTR instruction uses instruction uses this number to change program control to one of the 32 interrupt vector addresses.

PM – A 2-bit value copied into the PM bits of status register ST1 by the SPM instruction

SHF – A 8-bit left-shift value

SHFT – A 4-bit left-shift value

TP = A 2-bit value used by the conditional execution instructions to represent four conditions

$\overline{BIO}$ pin Low	TP = 00
TC bit = 1	TP = 01
TC bit = 0	TP = 10
no condition	TP = 11

ZLVC ZLVC – Two 4-bit fields – each representing the following conditions:

ACC = 0	Z
ACC < 0	L
Overflow	V
Carry	C

A conditional instruction contains two of these 4-bit fields. The 4-LSB field of the instruction is a mask field. A 1 in the corresponding mask bit indicates that condition is being tested. For example, to test for ACC ≥ 0, the Z and L fields are set and the V and C

fields are not set. The Z field is set to test to test the condition ACC = 0, and the L field is ~~set to test~~ reset to test the condition ACC ≥ 0. The second 4-bit field (bits 4-7) indicates the state of the conditions to test. The conditions possible with these eight bits are shown in the destriptions of the BCND, CC and RETC instructions.

+1 word — The second word of a 2-word opcode. This second word contains a 16-bit constant. Depending on the instruction, this constant is a Long immediate value, a program memory address, or an address for an I/O ports or an I/O-mapped register.