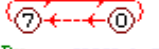
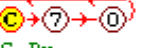


MOVLB d4 LDI BSR,#d4	ADDLW d8 ADD A,#d	SUBLW d8 SUB #d,A !результат в A!	ANDLW d8 AND A,#d	IORLW d8 OR A,#d	XORLW d8 XOR A,#d	INCF f,d,a INC Rn MOV A,Rn INC A	DECF f,d,a DEC Rn MOV A,Rn DEC A
LFSR f,d12 (0≤f≤2) LDI FSR,#d12	ADDWF f,d,a ADD Rn,A ADD A,Rn	SUBWF f,d,a SUB Rn,A MOV B,Rn SUB B,A XCH A,B	ANDWF f,d,a AND Rn,A AND A,Rn	IORWF f,d,a OR Rn,A OR A,Rn	XORWF f,d,a XOR Rn,A XOR A,Rn	INCFSZ f,d,a INC Rn MOV A,Rn INC A	DECFSZ f,d,a DEC Rn MOV A,Rn DEC A
MOVLW d8 MOV A,#d	ADDWFC f,d,a ADDC Rn,A ADDC A,Rn	SUBWFB f,d,a SUBB Rn,A MOV B,Rn SUBB B,A XCH A,B	CPFSEQ f,a CP Rn,A if Rn=A rstep	CPFSGT f,a CP Rn,A if Rn>A rstep	CPFSLT f,a CP Rn,A if Rn<A rstep	INFSNZ f,d,a INC Rn MOV A,Rn INC A	DCFSNZ f,d,a DEC Rn MOV A,Rn DEC A
MOVWF f,a MOV Rn,A	DAW только сложение! DA A	SUBFWB f,d,a MOV B,A SUBB B,Rn XCH Rn,B SUBB A,Rn		CALLW CALL PCLATU:PCLATH:A	PUSHL d8 MOV @FSR2,#d DEC FSR2	MOVWF [rel],adr MOV adr,@(FSR2+#re)	MOVSS [re],[re] _d MOV @(FSR2+#re) _d , @(FSR2+#re) _s
MOVWF f,d,a MOV Rn,Rn /MOV PORTn,PINn/ MOV A,Rn	MULLW d8 MUL A,#d результат в PRODH:PRODL	RLNCF f,d,a  RL Rn MOV A,Rn RL A	RRNCF f,d,a  RR Rn MOV A,Rn RR A	ADDULNK dk ADD FSR2,#d RET	ADDFSR r,dk ADD FSRr,#d	в режиме extended mode для команд с f,d,a/f,b,a/f,a при f=0-95(0x00-0x5f) мнемоника: КОМАНДА [f],d/b исполнение как: КОМАНДА @(FSR2+#f),d/b,0 при f=96-255(0x60-0xFF) мнемоника и исполнение команд стандартные	
MOVFF fs,fd MOV Rd,Rs 0≤Rd/Rs≤4095	MULWF f,a MUL A,Rn результат в PRODH:PRODL	RLCF f,d,a  RLC Rn MOV A,Rn RLC A	RRCF f,d,a  RRC Rn MOV A,Rn RRC A	SUBULNK dk SUB FSR2,#d RET	SUBFSR r,dk SUB FSR2,#d		
CLRF f,a CLR Rn	SETF f,a MOV Rn,0xFF	COMF f,d,a CPL Rn MOV A,Rn CPL A	SWAPF f,d,a SWAP Rn MOV A,Rn SWAP A	NEGF f,a MOV Rn,0x00-Rn	TSTFSZ f,a JZ Rn,rstep	NOP	PUSH MOV TOSU:TOSH:TOSL,PC
GOTO addr JMP addr	CALL addr,s CALL addr /CALL addr FAST/	BRA rela RJMP rela	RCALL rela RCALL rela	RETURN s RET /RET FAST/	RETFIE s RETI /RETI FAST/	RETLW d8 MOV A,#d RET	POP DEC STKPTR TOS=(STKPTR)
RESET = аппаратный сброс ИС	CLRWDI WDR	SLEEP	BCF f,b,a CLR Rn.b	BSF f,b,a SETB Rn.b	BTG f,b,a CPL Rn.b	BTFSC f,b,a JNB Rn.b rstep	BTFSS f,b,a JB Rn.b rstep
BC rel JC rel	BNC rel JNC rel	BZ rel JZ rel	BNZ rel JNZ rel	BOV rel JOV rel	BNOV rel JNOV rel	BN rel JN rel	BNN rel JNN rel
TBLRD* LPM TABLAT, TBLPTR	TBLRD** LPM TABLAT, TBLPTR+	TBLRD*- LPM TABLAT, TBLPTR-	TBLRD** LPM TABLAT, +TBLPTR	TBLWT* SPM TBLPTR, TABLAT	TBLWT** SPM TBLPTR+ TABLAT	TBLWT*- SPM TBLPTR- TABLAT	TBLWT** SPM +TBLPTR, TABLAT

*-принято по умолчанию (не требуется прямого указания в тексте команды)
a=0 access ram
a=1 bsr ram
d=0(W) результат в W
d=1(F) результат в F
-без альтернативы
s=0 обычный режим
s=1(FAST) "быстрый стек"
0<b<7
0<d4<15
0<d8<255
0<d12<4095
0<dk<63
0<f<255
0<adr<4095
-128<rel<127
-1024<rela<1023
r=0,1,2
rstep переход на +lкоманду
0<re<127 смещение относительно FSR2
extended instruction для включения этого режима требуется дополнительная установка флагов в регистрах конфигурации конкретной ИС + установить флажок "extend.mode" на вкладке проекта в MPASM/CL7/CL8 Suite
PUSH и POP касаются только счетчика адреса в 31-уровневом предикрементном/посткрементном аппаратном стеке программно доступна только верхушка стека: TOSU:TOSH:TOSL:/TOS/

Обращение к INDFr в качестве Rn выполняет обращение к регистру, адрес которого находится в соответствующем FSRr КОП INDFr,d,a = КОП @FSRr,d,a
!! Запись в PCL вызывает автоматическую загрузку PCLATU:PCLATH в PCU:PCN !!
варианты мнемоники КОП регистр=КОП регистр,l=КОП регистр,F (по умолчанию)
КОП регистр,0=КОП регистр,W

MOVWF модифицирует 2
согласно содержимого !!
регистра назначения

!при модификации PCL содержимое PCL.0 аппаратно приводится к 0!
!операции чтения PCL модифицируют содержимое PCLATU:PCLATH!

...INDFr ==>@FSRr ==> FSRr*
...POSTDECr ==>@FSRr ==> FSRr*-
...POSTINCc ==>@FSRr ==> FSRr*+
...PREINCc ==>@FSRr ==> FSRr*+
...PLUSWr ==>@FSRr+W ==> FSRr*+

при вычитании
C=1,Z=0->результат
положительный
C=2=1->результат=0
C=2=0->результат
отрицательный

d7	STATUS	d0
	z	o
	n	a
	u	u