

Lista rozkazów mikrokontrolerów rodziny '51.

W zestawieniu stosuje się następujące oznaczenia:

Rr - rejestry R0-R7.

direct - adres wewnętrznej [pamięci danych](#) (128B) i [rejestrów SFR](#).

@Ri - adres pośredni w rejestrze indeksowym R0 lub R1.

bit - bity [pamięci danych](#) i [SFR](#) adresowalne bezpośrednio.

#data - stała 8-bitowa.

#data16 - stała 16-bitowa.

addr16 - adres 16-bitowy (obszar 64kB).

addr11 - adres 11-bitowy (w stronie 2kB).

rel - adres względny -128...+127.

/bit - negacja bitu.

(n) - rozkaz wykonywany w n cyklach maszynowych. Brak oznaczenia oznacza 1 cykl.

Operacje arytmetyczne:

ADD - dodawanie.

Działanie: $A=A+dana$.

Dozwolone argumenty: A,Rr A,direct A,@Ri A,#data

Modyfikowane znaczniki: C, AC, OV.

ADDC - dodawanie z przeniesieniem.

Działanie: $A=A+dana+C$.

Dozwolone argumenty: A,Rr A,direct A,@Ri A,#data

Modyfikowane znaczniki: C, AC, OV.

SUBB - odejmowanie z pożyczką.

Działanie: $A=A-dana-C$.

Dozwolone argumenty: A,Rr A,direct A,@Ri A,#data

Modyfikowane znaczniki: C, AC, OV.

INC - inkrementacja.

Działanie: $argument=argument+1$

Dozwolone argumenty: A Rr direct @Ri DPTR₍₂₎

DEC - dekrementacja.

Działanie: $argument=argument-1$.

Dozwolone argumenty: A Rr direct @Ri

MUL - mnożenie liczb 8-bitowych, wynik 16-bitowy.

Działanie: $A*B$, do A zapisywana młodsza część wyniku, do B starsza.

Dozwolone argumenty: AB

Modyfikowane znaczniki: OV, bit C zerowany.

DIV - dzielenie liczb 8-bitowych, 8-bitowy wynik i 8-bitowa reszta.

Działanie: A/B , A - wynik, B - reszta.

Dozwolone argumenty: AB

Modyfikowane znaczniki: OV, bit C zerowany.

DA - poprawka dziesiętna.

Działanie: wykonuje poprawkę dziesiętną liczby 8-bitowej, przekształcając ją do postaci 2 cyfr w kodzie BCD.

Dozwolone argumenty: A

Modyfikowane znaczniki: C.

Operacje logiczne:

ANL - iloczyn logiczny AND.

Działanie: wykonuje logiczny iloczyn na odpowiednich bitach argumentów, wynik zapisuje do pierwszego argumentu.

Dozwolone argumenty: A,Rr A,direct A,@Ri A,#data direct,A direct,#data₍₂₎

ORL - suma logiczna OR.

Działanie: wykonuje sumę logiczną odpowiednich bitów argumentów, wynik zapisuje do pierwszego argumentu.

Dozwolone argumenty: A,Rr A,direct A,@Ri A,#data direct,A direct,#data₍₂₎

XRL - suma mod 2 (XOR).

Działanie: wykonuje sumowanie mod 2 odpowiednich bitów argumentu, wynik zapisuje do pierwszego z nich.

Dozwolone argumenty: A,Rr A,direct A,@Ri A,#data direct,A direct,#data₍₂₎

CLR - zerowanie (bajtowe).

Działanie: zeruje wszystkie bity argumentu.

Dozwolone argumenty: A.

CPL - negowanie (bajtowe).

Działanie: neguje wszystkie bity argumentu.

Dozwolone argumenty: A.

RL - obrót w lewo.

Działanie: przesuwa w lewo zawartość akumulatora.

Dozwolone argumenty: A.

RR - obrót w prawo.

Działanie: przesuwa w prawo zawartość akumulatora.

Dozwolone argumenty: A.

RLC - obrót w lewo przez C.

Działanie: przesuwa w lewo zawartość akumulatora, na miejsce najmłodszego bitu wpisywana jest dotychczasowa zawartość bitu C, bit najstarszy jest wpisywany jako nowa wartość C.

Dozwolone argumenty: A.

RRC - obrót w prawo przez C.

Działanie: przesuwa w prawo zawartość akumulatora, na miejsce najstarszego bitu wpisywana jest dotychczasowa zawartość bitu C, bit najmłodszy jest wpisywany jako nowa wartość C.

Dozwolone argumenty: A.

SWAP - zamiana 4 bitów bajtu.

Działanie: zamienia miejscami 4 starsze i 4 młodsze bity bajtu.

Dozwolone argumenty: A.

Operacje logiczne na bitach:**CLR - zerowanie.**

Działanie: zeruje podany bit.

Dozwolone argumenty: C bit.

SETB - ustawienie bitu.

Działanie: przypisuje danemu bitowi wartość 1.

Dozwolone argumenty: C bit.

CPL - negacja.

Działanie: neguje podany bit.

Dozwolone argumenty: C bit.

ANL₍₂₎ - iloczyn logiczny AND.

Działanie: wykonuje iloczyn logiczny dwóch bitów, wynik zapisuje w pierwszym.

Dozwolone argumenty: C,bit C,/bit.

ORL₍₂₎ - suma logiczna OR.

Działanie: wykonuje sumę logiczną dwóch bitów, wynik zapisuje w pierwszym.

Dozwolone argumenty: C,bit C,/bit.

MOV - kopiuj.

Działanie: wpisuje do pierwszego bitu wartość zapisaną w drugim.

Dozwolone argumenty: C,bit bit,C₍₂₎.

Przesyłanie danych:

MOV - kopiuj.

Działanie: wpisuje do pierwszego operandu zawartość drugiego.

Dozwolone argumenty: A,Rr A,direct A,@Ri A,#data Rr,A Rr,direct₍₂₎ Rr,#data direct,A direct,Rr₍₂₎ direct,direct₍₂₎ direct,@Ri₍₂₎ direct,#data₍₂₎ @Ri,A @Ri,direct₍₂₎ @Ri,data DPTR,#data16₍₂₎.

MOVC₍₂₎ - kopiuj z pamięci programu.

Działanie: kopiuje bajt kodu (zawartości pamięci programu).

Dozwolone argumenty: A,@A+DPTR A,@A+PC.

MOVX₍₂₎ - kopiuj z/do zewnętrznej pamięci danych.

Działanie: kopiuje bajt z/do komórki zewnętrznej pamięci danych wskazanej rejestrem indeksowym.

Dozwolone argumenty: A,@Ri A,@DPTR @Ri,A @DPTR,A.

XCH - zamień.

Działanie: wymienia między sobą zawartość dwóch rejestrów.

Dozwolone argumenty: A,Rr A,direct A,@Ri.

XCHD - zamień młodsze 4 bity.

Działanie: wymienia między dwoma rejestrami zawartość ich 4 młodszych bitów.

Dozwolone argumenty: A,@Ri.

PUSH₍₂₎ - przesłanie bajtu na stos.

Działanie: inkrementuje zawartość rejestru SP i kopiuje zawartość komórki pod adres wskazywany przez SP.

Dozwolone argumenty: direct.

POP₍₂₎ - pobranie bajtu ze stosu.

Działanie: kopiuje zawartość komórki wskazywanej przez SP do komórki podanej w wywołaniu i dekrementuje zawartość SP.

Dozwolone argumenty: direct.

Skoki₍₂₎:

ACALL - wywołanie podprogramu w stronie 2kB.

Działanie: zapisuje na stos zawartość PC i wykonuje skok do adresu początku procedury.

Dozwolone argumenty: addr11.

LCALL - wywołanie podprogramu.

Działanie: zapisuje na stos zawartość PC i wykonuje skok do adresu początku procedury.

Dozwolone argumenty: addr16.

RET - powrót z procedury.

Działanie: pobiera ze stosu zawartość licznika rozkazów i przekazuje sterowanie do znajdującego się tam adresu.

RETI - powrót z procedury obsługi przerwania.

Działanie: pobiera ze stosu zawartość licznika rozkazów i przekazuje sterowanie do znajdującego się tam adresu.

LJMP - rozkaz skoku w obszarze 64kB.

Działanie: wpisuje do licznika rozkazów podany adres.

Dozwolone argumenty: addr16.

AJMP - rozkaz skoku w stronie 2kB.

Działanie: zastępuje 11 młodszych bitów PC wartością podaną w wywołaniu.

Dozwolone argumenty: addr11.

SJMP - rozkaz skoku w obszarze 256B.

Działanie: następuje zmiana zawartości PC o wartość w kodzie U2 podaną w wywołaniu.

Dozwolone argumenty: rel.

JMP - rozkaz skoku w trybie pośrednim.

Działanie: przepisuje do PC wartość będącą sumą zawartości rejestru indeksowego i bazowego.

Dozwolone argumenty: @A+DPTR.

JZ - skok warunkowy.

Działanie: wykonuje skok, jeżeli zawartość akumulatora równa jest 0.

Dozwolone argumenty: rel.

JNZ - skok warunkowy.

Działanie: wykonuje skok, jeżeli zawartość akumulatora jest różna od 0.

Dozwolone argumenty: rel.

JC - skok warunkowy.

Działanie: wykonuje skok, jeżeli bit przeniesienia C jest ustawiony.

Dozwolone argumenty: rel.

JNC - skok warunkowy.

Działanie: wykonuje skok, jeżeli bit przeniesienia C jest wyzerowany.

Dozwolone argumenty: rel.

JB - skok warunkowy.

Działanie: wykonuje skok, jeżeli dany bit jest ustawiony.

Dozwolone argumenty: bit,rel.

JNB - skok warunkowy.

Działanie: wykonuje skok, jeżeli dany bit jest wyzerowany.

Dozwolone argumenty: bit,rel.

JBC - skok warunkowy.

Działanie: gdy dany bit jest ustawiony, to nastąpi skok i wyzerowanie bitu.

Dozwolone argumenty: bit,rel.

CJNE - skok warunkowy, zależny od wyniku porównania.

Działanie: następuje porównanie operandów, jeżeli są różne, to następuje skok.

Dozwolone argumenty: A,direct,rel A,#data,rel Rr,#data,rel @Ri,#data,rel.

Modyfikowane znaczniki: C (ustawiany, jeżeli pierwszy operand mniejszy od drugiego).

DJNZ - skok warunkowy, zależny od wyniku dekrementacji.

Działanie: następuje dekrementacja zawartości operandu, jeżeli zawartość nie osiągnęła zera, to nastąpi skok.

Dozwolone argumenty: Rr,rel direct,rel.

Inne**NOP – nic nie rób (no operation)**

Jego wykonanie nie wywołuje żadnych efektów, zajmuje jedynie czas 1 cyklu maszynowego. Często jest on wykorzystywany jako rezerwaacja miejsca pamięci przeznaczzonego na późniejszą rozbudowę programu, lub jako niewielkie (1 cykl) opóźnienie w wykonywaniu programu.