

# Instrukční soubor mikroprocesorů 8051 / 8052

	A	Rx	a <sub>8</sub>	@Rr	#d <sub>8</sub>	DPTR	@DPTR	C	AC	OV	P	vysvětlení	
<b>&lt; přesuny &gt;</b>													
MOV	A,	-	-	1/1	2/1	1/1	2/1	-	-	-	-	P	(A) ← ( )
MOV	Rx,	-	1/1	-	2/2	-	2/1	-	-	-	-	P	(Rx) ← ( )
MOV	a <sub>8</sub> ,	-	2/1	2/2	3/2	2/2	3/2	-	-	-	-	-	(a <sub>8</sub> ) ← ( )
MOV	@Rr,	-	1/1	-	2/2	-	2/1	-	-	-	-	-	((Rr)) ← ( )
Přesun bytu z paměťového místa na jiné paměťové místo ve vnitřní datové paměti. Instrukce MOV přesune obsah zdrojového bytu do cílového bytu bez ovlivnění jakýchkoliv příznaků.													
MOV	DPTR,#d <sub>16</sub>	3/2	-	-	-	-	-	-	-	-	-	-	(DPTR) ← d <sub>16</sub>
Instrukce přesune obsah druhého a třetího bytu instrukce do ukazatele dat (DPTR). Druhý byt do DPH a třetí byt do DPL.													
MOVC	A,@A+	-	-	-	-	-	1/2	-	-	-	-	P	(A) ← ((A)+(DPTR)) <sub>c</sub>
MOVC	A,@A+PC	1/2	-	-	-	-	-	-	-	-	-	P	(A) ← ((A)+(PC)) <sub>c</sub>
Instrukce přesune byte z programové paměti (operační kód nebo konstantu) do střadače. Adresa místa, jehož obsah se přesouvá, získáme jako 16-bitový součet obsahu střadače (8 bitů) a ukazatele dat DPTR nebo čítače instrukcí PC. Pro případ čítače instrukcí je jeho obsah před provedením instrukce inkrementován (ukazuje na následující instrukci).													
MOVX	A,	-	-	-	1/2	-	1/2	-	-	-	-	P	(A) ← (( )) <sub>x</sub>
MOVX	,A	-	-	-	1/2	-	1/2	-	-	-	-	-	(( )) <sub>x</sub> ← (A)
Instrukce přesune byte z/do střadače do/z vnější paměti dat. Instrukce mohou využívat 16-bitovou nebo 8-bitovou nepřímou adresu. V prvním případě se vysílá adresa uložená v DPTR na bránu P2 (DPH) a bránu P0 (DPL). V druhém případě se vysílá na bránu P0 adresa uložená v registru R0 nebo R1 a na bráně P2 zůstává hodnota naposledy zapsaná.													
PUSH	a <sub>8</sub>	-	-	2/2	-	-	-	-	-	-	-	-	zásobník ← (a <sub>8</sub> )
Instrukce přičte jedničku k ukazateli zásobníku a potom uloží obsah adresovaného místa do vrcholu zásobníku (zásobník je vždy ve vnitřní datové paměti).													
POP	a <sub>8</sub>	-	-	2/2	-	-	-	-	-	-	-	-	(a <sub>8</sub> ) ← zásobník
Instrukce vyzvedne obsah vrcholu zásobníku a uloží jej na adresované paměťové místo. Pak odečte od ukazatele zásobníku jedničku.													
XCH	A,	-	-	1/1	2/1	1/1	-	-	-	-	-	P	(A) ↔ ( )
Instrukce vymění (navzájem prohodí) obsah střadače a určeného registru nebo adresovaného paměťového místa.													
XCHD	A,	-	-	-	1/1	-	-	-	-	-	-	P	(A <sub>3-0</sub> ) ↔ ( ) <sub>3-0</sub>
Instrukce vymění (navzájem prohodí) obsah nižšího "půlbytu" střadače s nepřímo adresovaným paměťovým místem vnitřní paměti RAM.													
<b>&lt; aritmetické operace &gt;</b>													
INC		-	1/1	-	-	-	-	-	-	-	-	P	( ) ← ( ) + 1
INC		-	-	1/1	2/1	1/1	-	1/2	-	-	-	-	( ) ← ( ) + 1
Instrukce přičte k obsahu adresovaného paměťového místa jedničku. Po zvětšení hodnoty FFH dojde k přetečení na hodnotu 00H. Inkrementace obsahu výstupní brány zvětšuje obsah přečtený z registru brány a nikoliv ze vstupně/výstupních vodičů. Instrukce INC DPTR přičte jedničku k registrovému páru DPH a DPL, které vytváří 16-bitový ukazatel datové paměti DPTR. Dojde-li při přičítání k přetečení u registru DPL (FFH → 00H) potom je přičtena jednička k registru DPH. Přičítání jedničky k DPTR probíhá modulo 2 <sup>16</sup> (FFFFH+1 → 0000H). Instrukce INC DPTR je jedinou 16-bitovou instrukcí v instrukčním souboru procesoru.													
DEC	A	-	1/1	-	-	-	-	-	-	-	-	P	(A) ← (A) - 1
DEC		-	-	1/1	2/1	1/1	-	-	-	-	-	-	( ) ← ( ) - 1
Instrukce odečte jedničku od obsahu adresovaného paměťového místa. Po zmenšení hodnoty 00H dojde k podečtení na hodnotu FFH. Dekrementace obsahu výstupní brány zmenšuje obsah přečtený z registru brány a nikoliv ze vstupně/výstupních vodičů.													
ADD	A,	-	-	1/1	2/1	1/1	2/1	-	C	AC	OV	P	(A) ← (A)+( )
Instrukce přičte obsah adresovaného bytu ke střadači a výsledek v něm ponechá.													
ADDC	A,	-	-	1/1	2/1	1/1	2/1	-	C	AC	OV	P	(A) ← (A)+(C)+( )
Instrukce přičte obsah adresovaného bytu a příznakový bit přenosu C ke střadači a výsledek v něm ponechá.													
DA	A	-	1/1	-	-	-	-	-	C	AC	-	P	dokad.korekce po +
Instrukce koriguje obsah střadače po binárním sčítání dvou dekadických čísel vyjádřených v BCD kódu tak, aby výsledek opět tvořil dvě čtyřbitová BCD čísla. Je-li hodnota na nižších čtyřech bytech >9 nebo AC=1, potom se ke střadači přičte hodnota 6. Je-li hodnota na vyšších čtyřech bitech >9 nebo AC=1, potom se ke střadači přičte hodnota 60H.													
SUBB	A,	-	-	1/1	2/1	1/1	2/1	-	C	AC	OV	P	(A) ← (A)-(C)-( )
Instrukce odečte obsah adresovaného bytu od střadače včetně příznaku přenosu (výpůjčky) a výsledek v něm ponechá. Je-li při výpočtu vyžadována v bitu b <sub>7</sub> výpůjčka, je příznakový bit C nastaven, v opačném případě je vynulován.													
MUL	AB	1/4	-	-	-	-	-	-	0	-	OV	P	(A),(B) ← (A) * (B)
Instrukce vynásobí dvě osmibitová čísla bez znaménka uložená ve střadači a registru B. Je-li součin větší než hodnota 255 (FFH), nastaví se příznakový bit přetečení OV=1. V opačném případě se OV vynuluje.													
DIV	AB	1/4	-	-	-	-	-	-	0	-	OV	P	(A),(B) ← (A) / (B)
Instrukce provádí celočíselné dělení obsahu střadače s obsahem registru B. Celá část podílu zůstává ve střadači, zbytek (nikoliv desetinná část) zůstává v registru B. Při dělení nulou se nastaví příznak přetečení OV=1.													
<b>&lt; logické operace &gt;</b>													
ANL	A,	-	-	1/1	2/1	1/1	-	-	-	-	-	P	(A) ← (A) AND ( )
ANL	A,	-	-	-	-	-	2/1	-	-	-	-	P	(A) ← (A) AND ( )
ANL	a <sub>8</sub> ,	-	2/1	-	-	-	3/2	-	-	-	-	-	(a <sub>8</sub> ) ← (a <sub>8</sub> ) AND ( )
Instrukce ANL provede logický součin mezi odpovídajícími bity cílového a zdrojového bytu a výsledek uloží do cílového bytu. Je-li cílovým bytem výstupní brána (přímá adresa), pak se operace provede mezi výstupním registrem a zdrojovým bytem (nikoliv vstupními signály).													
ORL	A,	-	-	1/1	2/1	1/1	2/1	-	-	-	-	P	(A) ← (A) OR ( )
ORL	a <sub>8</sub> ,	-	2/1	-	-	-	-	-	-	-	-	-	(a <sub>8</sub> ) ← (a <sub>8</sub> ) OR ( )
ORL	a <sub>8</sub> ,	-	-	-	-	-	3/2	-	-	-	-	-	(a <sub>8</sub> ) ← (a <sub>8</sub> ) OR ( )
Instrukce ORL provede logický součet mezi odpovídajícími bity cílového a zdrojového bytu a výsledek uloží do cílového bytu. Pro operaci s výstupní bránou (přímá adresa) se operace provede mezi výstupním registrem a zdrojovým bytem (nikoliv vstupními signály).													
XRL	A,	-	-	1/1	2/1	1/1	2/1	-	-	-	-	P	(A) ← (A) XOR ( )
XRL	a <sub>8</sub> ,	-	2/1	-	-	-	3/2	-	-	-	-	-	(a <sub>8</sub> ) ← (a <sub>8</sub> ) XOR ( )

Instrukce XRL provede operaci XOR (exclusive OR) mezi odpovídajícími bity cílového a zdrojového bytu a výsledek uloží do cílového bytu. Pro operaci s výstupní bránou (přímá adresa) se operace provede mezi výstupním registrem a zdrojovým bytem (nikoliv vstupními signály).

CLR	A	-	1/1	-	-	-	-	-	-	-	-	-	P	(A) ← (0)
Instrukce vynuluje obsah střadače.														
CPL	A	-	1/1	-	-	-	-	-	-	-	-	-	-	(A) ← NOT (A)
Instrukce neguje každý bit střadače a vytváří tak jeho jednotkový doplněk.														
<b>&lt; posuny &gt;</b>														
RL	A	-	1/1	-	-	-	-	-	-	-	-	-	-	rotace vlevo
Instrukce osmibitové logické rotace střadače o jednu pozici vlevo.														
RLC	A	-	1/1	-	-	-	-	-	C	-	-	-	P	rotace vlevo přes C
Instrukce devítibitové logické rotace střadače a příznaku přenosu o jednu pozici vlevo.														
RR	A	-	1/1	-	-	-	-	-	-	-	-	-	-	rotace vpravo
Instrukce osmibitové logické rotace střadače o jednu pozici vpravo.														
RRC	A	-	1/1	-	-	-	-	-	C	-	-	-	P	rotace vpravo přes C
Instrukce devítibitové logické rotace střadače a příznaku přenosu o jednu pozici vpravo.														
SWAP	A	-	1/1	-	-	-	-	-	-	-	-	-	-	(A <sub>0-3</sub> ) ↔ (A <sub>4-7</sub> )
Instrukce prohodí navzájem obsah nižšího a vyššího "půlbytu" střadače. Instrukce je shodná s osmibitovou rotací o čtyři bity vpravo nebo vlevo.														

<b>&lt; bitové operace &gt;</b>														
		b <sub>8</sub>	/b <sub>8</sub>	C	NC	B	NB	BC	C	AC	OV	P	vysvětlení	
CLR		-	2/1	-	1/1	-	-	-	*	-	-	-	-	( ) ← 0
Instrukce vynuluje adresovaný bit. Ovlivňuje: C,AC,F0,RS1,RS0,OV,P jen je-li adresován.														
SETB		-	2/1	-	1/1	-	-	-	*	-	-	-	-	( ) ← 1
Instrukce nastaví přímo adresovaný bit na log.1.														
CPL		-	2/1	-	1/1	-	-	-	*	-	-	-	-	( ) ← NOT ( )
Instrukce neguje adresovaný bit. Ovlivňuje: C,AC,F0,RS1,RS0,OV,P jen je-li adresován.														
ANL	C,	-	2/2	2/2	-	-	-	-	C	-	-	-	-	(C) ← (C) AND ( )
Logický součin příznaku přenosu C s přímo adresovaným bitem. Výsledek operace se uloží do příznaku C. Je-li před adresou bitu lomítko, potom hodnota bitu bude před operací negována. Adresovaný bit lze adresovat jenom přímou adresou.														
ORL	C,	-	2/2	2/2	-	-	-	-	C	-	-	-	-	(C) ← (C) OR ( )
Logický součet příznaku přenosu C s přímo adresovaným bitem. Výsledek operace se uloží do příznaku C. Je-li před adresou bitu lomítko, potom hodnota bitu bude před operací negována. Adresovaný bit lze adresovat jenom přímou adresou.														
MOV	C,b <sub>8</sub>	2/1	-	-	-	-	-	-	C	-	-	-	-	(C) ← (b <sub>8</sub> )
MOV	b <sub>8</sub> ,C	2/2	-	-	-	-	-	-	-	-	-	-	-	(b <sub>8</sub> ) ← (C)
Instrukce přesune hodnotu mezi příznakem přenosu C a daným bitem.														
J	r <sub>8</sub>	-	-	-	2/2	2/2	-	-	-	-	-	-	-	(PC) ← (PC)+2+if( ,r <sub>8</sub> ,0)
Instrukce JC testuje příznak přenosu a v případě jeho nastavení (log.1) provede skok na adresu, kterou vypočte jako součet čítače instrukcí a relativní adresy.														
Instrukce JNC testuje příznak přenosu a v případě jeho nulovosti (log.0) provede skok na adresu, kterou vypočte jako součet čítače instrukcí a relativní adresy.														
J	b <sub>8</sub> ,r <sub>8</sub>	-	-	-	-	-	3/2	3/2	-	-	-	-	-	(PC) ← (PC)+3+if( ,r <sub>8</sub> ,0)
Instrukce JB testuje adresovaný bit a v případě jeho nastavení (log.1) provede skok na adresu, kterou vypočte jako součet čítače instrukcí a relativní adresy.														
Instrukce JNB testuje adresovaný bit a v případě jeho nulovosti (log.0) provede skok na adresu, kterou vypočte jako součet čítače instrukcí a relativní adresy.														
J	b <sub>8</sub> ,r <sub>8</sub>	-	-	-	-	-	-	3/2	-	-	-	-	-	(PC) ← (PC)+3+if( ,r <sub>8</sub> ,0),(b <sub>8</sub> ) ← 0
Instrukce testuje adresovaný bit a v případě jeho nastavení (log.1) provede skok na adresu, kterou vypočte jako součet čítače instrukcí a relativní adresy a vynuluje testovaný bit.														

<b>&lt; skoky &gt;</b>														
	A	Rx	@Rr	a <sub>8</sub>	Z	NZ			C	AC	OV	P		
ACALL	a <sub>11</sub>	2/2	-	-	-	-	-	-	-	-	-	-	-	{SP}←(PC+2).(PC)-a <sub>11</sub> ,SP=SP+2
Volání podprogramu uvnitř 2kB adresovaného prostoru (11-bitová adresa). Instrukce uloží návratovou adresu do zásobníku s tím, že nejprve uloží nižší a potom vyšší byte. Volaný podprogram musí ležet uvnitř 2kB stránky, ve které leží instrukce následující po instrukci ACALL.														
LCALL	a <sub>16</sub>	3/2	-	-	-	-	-	-	-	-	-	-	-	{SP}←(PC+3).(PC)-a <sub>16</sub> ,SP=SP+2
Instrukce vykoná nepodmíněně volání podprogramu z přímo uvedené adresy ve svém druhém (vyšší byte) a třetím (nižší byte) bytu. Před uložením přečtené adresy do čítače instrukcí (PC) se uloží návratová adresa (současný stav PC=adresa následující instrukce) do zásobníku.														
RET		1/2	-	-	-	-	-	-	-	-	-	-	-	(PC) ← (SP), SP=SP - 2
Instrukce návratu z podprogramu vyzvedne ze zásobníku dva byty a uloží je do čítače instrukcí (PC). Nejprve vyjme vyšší byte a potom nižší byte a odečte od ukazatele zásobníku hodnotu dvě.														
RETI		1/2,	-	-	-	-	-	-	-	-	-	-	-	(PC) ← (SP), SP=SP - 2
Instrukce návratu z obsluženého podprogramu přerušení vyzvedne ze zásobníku dva byty a uloží je do čítače instrukcí (PC). Nejprve vyjme vyšší a potom nižší byte a odečte od ukazatele zásobníku hodnotu dvě. Nakonec povolí přijetí žádosti o přerušení se stejnou nebo nižší úrovní priority. Stavové slovo se automaticky neobnovuje.														
AJMP	a <sub>11</sub>	2/2	-	-	-	-	-	-	-	-	-	-	-	(PC) <sub>0-10</sub> ← a <sub>11</sub>
Krátký nepodmíněný skok na adresu uvnitř 2kB stránky.														
LJMP	a <sub>16</sub>	3/2	-	-	-	-	-	-	-	-	-	-	-	(PC) ← a <sub>16</sub>
Instrukce vykoná nepodmíněný skok na adresu přímo uvedenou ve svém druhém (vyšší byte) a třetím (nižší byte) bytu. Adresa může ležet kdekoli v 64kB adresovém prostoru.														
SJMP	r <sub>8</sub>	2/2	-	-	-	-	-	-	-	-	-	-	-	(PC) ← (PC) + 2 + r <sub>8</sub>
Instrukce realizuje krátký nepodmíněný skok na definovanou adresu, která se vypočítá jako součet čítače instrukcí ukazujícího na adresu následující instrukce a posunu (-128;127) určeného relativní adresou.														
JMP	@A+DPTR	1/2	-	-	-	-	-	-	-	-	-	-	-	(PC) ← (A) + (DPTR)

Instrukce nepřímého nepodmíněného skoku na adresu určenou 16-bitovým součtem obsahu střadače (8 bitů bez znaménka) s obsahem ukazatele datové paměti DPTR.

J	$r_8$	-	-	-	-	-	2/2	2/2	-	-	-	-	$(PC) \leftarrow (PC)+3+if(,r_8,0)$
---	-------	---	---	---	---	---	-----	-----	---	---	---	---	-------------------------------------

Instrukce JZ testuje obsah střadače a v případě je nulovosti provede skok na adresu, kterou vypočte jako součet čítače instrukcí a relativní adresy.

Instrukce JNZ testuje obsah střadače a v případě je nenulovosti provede skok na adresu, kterou vypočte jako součet čítače instrukcí a relativní adresy.

CJNE	A, $a_8,r_8$	3/2	-	-	-	-	-	-	C	-	-	-	$(PC) \leftarrow (PC)+3+if((A)=(a_8),r_8,0)$
CJNE	,# $d_8,r_8$	-	3/2	3/2	-	3/2	-	-	C	-	-	-	$(PC) \leftarrow (PC)+3+if((,)=d_8,r_8,0)$

Instrukce porovná střadač nebo registr Rr nebo byte adresovaný registry R0,R1 se zdrojovým bytem (adresa,data). V případě jejich nerovnosti provede relativní skok (maximálně o +127 nebo -128 bytů) na požadovanou adresu. Adresa skoku se vypočte přičtením relativního posunu k čítači instrukcí, který byl nejprve třikrát inkrementován v důsledku čtení prováděné instrukce. Je-li cílový byte menší než zdrojový, je zároveň nastaven příznak přenosu C=1, jinak se nuluje C=0.

DJNZ	, $r_8$	-	-	3/2	3/2	-	-	-	-	-	-	-	$(PC) \leftarrow (PC)+3+if((,)=r_8,0),$ $(,)\leftarrow(,)-1$
------	---------	---	---	-----	-----	---	---	---	---	---	---	---	---

Instrukce odečte od adresovaného paměťového místa jedničku a zjistí, zda je výsledek nulový. Je-li výsledek nenulový, provede skok na definovanou adresu.

**< ostatní >**

NOP	1/1	-	-	-	-	-	-	-	-	-	-	-	nic nevykonává
-----	-----	---	---	---	---	---	---	---	---	---	---	---	----------------

Kromě čítače instrukcí neovlivňuje instrukce žádné registry a příznaky.

A	...střadač	1/1	...počet slabik/cyklů
Rx	...registr R0+R7	c	...paměť programu
@Rr	...nepřímá adresa v R0, R1(obsah registru určuje adresu místa s jehož obsahem se bude pracovat)	x	...externí paměť dat
$a_8, a_{11}, a_{16}$	...přímá adresa	*	...ovlivňuje, je-li operandem
# $d_8, #d_{16}$	...bezprostřední data (8, 16 bitů)	{SP}	...ukládání do (SP+1) a (SP+2), vybírání z (SP) a (SP-1)
$r_8$	...relativní adresa		
$b_8$	...bitová adresa		
(/ $b_8$ )	...negovaný obsah buňky na adrese $b_8$		

Co je cílový a zdrojový operand:

kód instrukce cílový byte,zdrojový byte,příp. 3.operand  
např. MOV cílový byte (bit), zdrojový byte(bit)