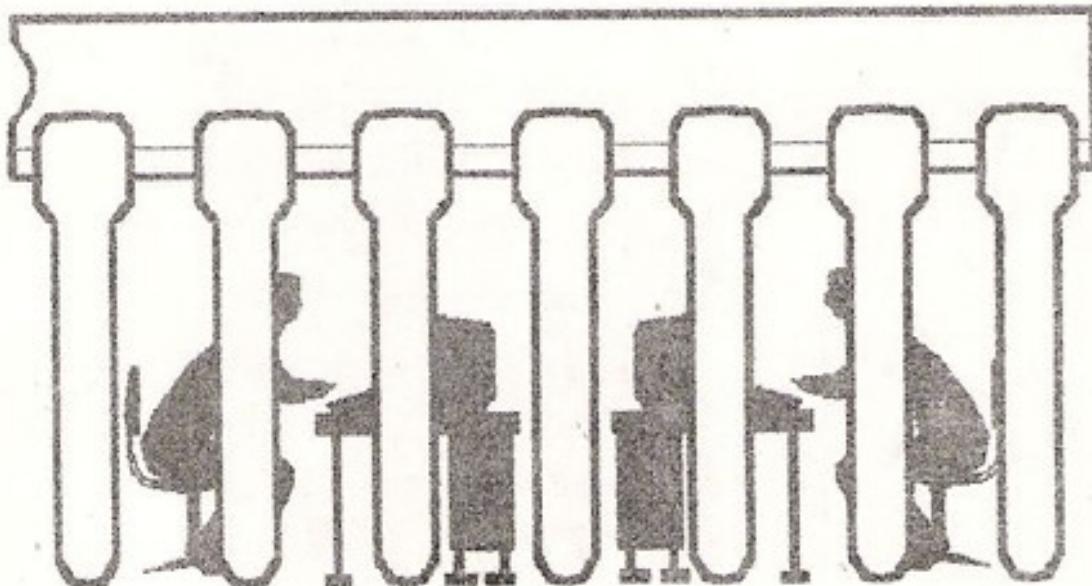


MIKROPROCESOR **6502**

POPIS
INSTRUKCÍ



ATARI KLUB OLOMOUC

Mikroprocesor 6502

Atari klub Olomouc, březen 1992

Mikroprocesor 6502 firmy MOS TECHNOLOGY se řadí spolu s mikroprocesorem ZILOG Z-80 mezi nejpoužívanější osmibitové mikroprocesory. Je použit v mikropočítacích firem APPLE, COMMODORE, ROCKWELL, SYNERTEK, ATARI a dalších.

Vnitřní strukturu mikroprocesoru si můžete prohlédnout na obr. 1. Základem je aritmetická jednotka ALU, která provádí veškeré aritmetické a logické funkce. Dále jsou zde 4 osmibitové registry A, S, X a Y. Dominantní postavení má registr A, tzv. akumulátor. Do něj jsou ukládány výsledky aritmeticko-logických operací. Zároveň slouží jako zdroj jednoho z operandů pro prováděné funkce. Registr S nazýváme ukazatelem zásobníku. Registry X a Y označujeme jako indexregistry. Používají se zejména při některých způsobech adresování, můžeme je však využít i jako zdroj druhého operandu pro funkce prováděné ALU. Zvláštní postavení má registr F, tzv. stavový registr. Jeho jednotlivé bity lze testovat a na základě jejich hodnoty provádět větvení programu. Tyto bity jsou nastavovány podle výsledků operací prováděných v ALU.

Posledním z registrů, který nás může zajímat z pozice programátora, je programový čítač. Ten obsahuje adresu instrukce, která se má vykonat. Adresová sběrnice je "široká" 16 bitů, což znamená, že maximální rozsah přímo adresovatelné paměti je 65 536 byte.

Organizace paměti

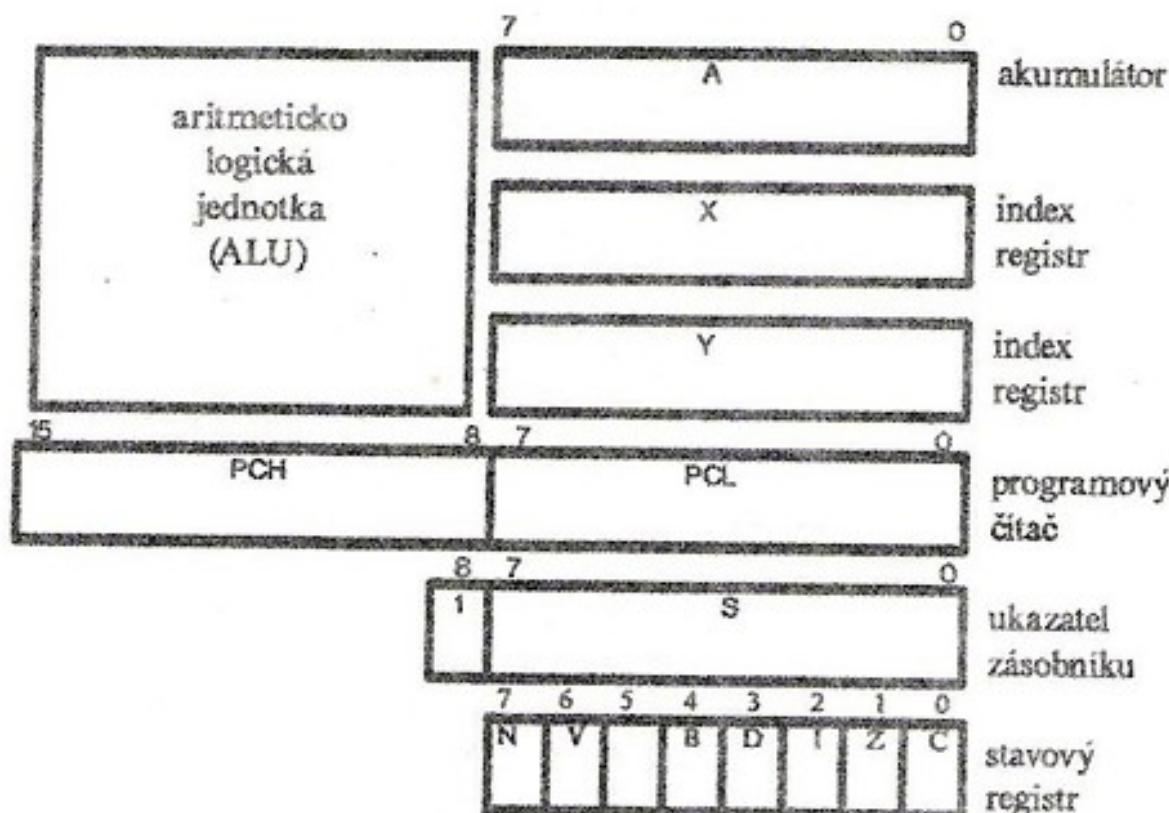
Mikroprocesor 6502 může adresovat paměť v rozsahu 64 kB (hexa \$0000 - \$FFFF). Paměť se dělí na tzv. stránky. Jedna stránka má délku 256 byte. V praxi to znamená, že vyšší byte adresy určuje stránku paměti a nižší byte určuje adresu uvnitř stránky. Teoreticky bychom mohli kdekoli v paměti umístit program nebo data. Existují však paměťové lokace, které procesor nezbytně potřebuje ke své práci. Organizace

paměti mikroprocesoru 6502 je uvedena na obr. 2. Z obrázku vyplývá, že zásobník může mít maximální délku 256 byte. Pro naprostou většinu aplikací je jeho délka postačují, v mnoha případech s velkou rezervou. Zásobník najdeme v první stránce paměti. Při tvorbě programů na to musíte brát ohled, aby nedocházelo ke kolizím mezi programem a zásobníkem.

V posledních šesti paměťových buňkách nalezneme adresy začátků programů, které zpracovávají odezvu na tři vnější signály:

adresa	přerušující signál
FFFAH, FFFFH	NMI - nemaskovatelné přerušení
FFFFCH, FFFDH	IRQ - maskované přerušení
FFFEH, FFFFH	RES - reset procesoru

Obr. 1 - Vnitřní struktura mikroprocesoru 6502



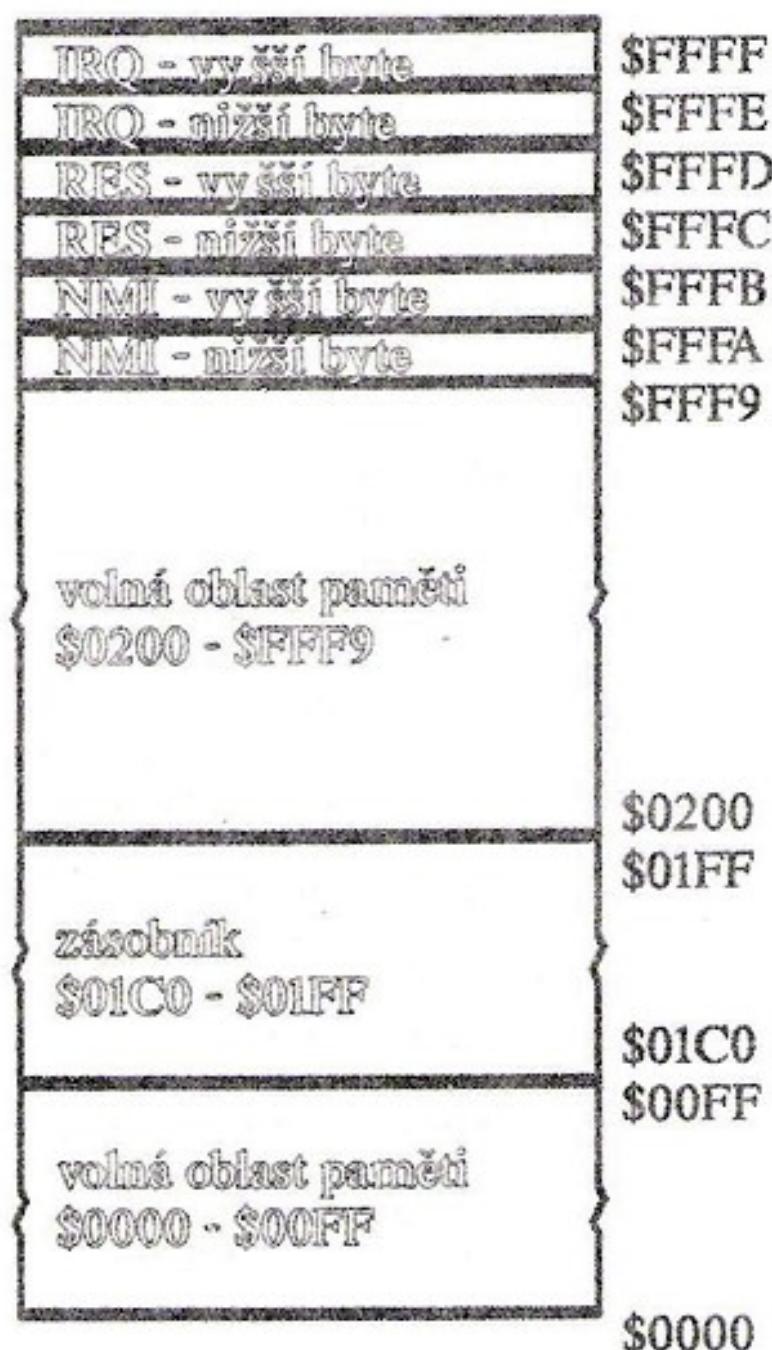
Způsoby adresování operandů instrukcí

Procesor 6502 umožňuje určit operand pomocí několika způsobů adresování:

1. IMMEDIATE - přímé určení. Operand je obsažen ve druhém bytu instrukce.
2. ABSOLUTE - Adresa operandu je obsažena ve druhém (nižším) a třetím (vyšším) bytu instrukce.
3. ZERO PAGE - Adresace paměti v rozsahu OOH - FFH, tj. v nulté stránce paměti. Pro určení adresy tedy stačí jen jeden byte. Tento byte najdeme ve druhém bytu instrukce.
4. ACCUMULATOR - instrukce operující implicitně s akumulátorem.
5. Z-PAGE INDEXED - Druhý byte instrukce je sečten s registrém X nebo Y. O přenosu do vyššího bytu se neuvažuje. Tento součet je potom použit jako adresa lokace v nulté stránce, kde se nachází příslušný operand.
6. ABSOLUTE INDEXED - K adrese určené druhým a třetím bytem instrukce je přičten registr X nebo Y. Výsledek je použit jako adresa operandu v paměti.
7. INDEXED INDIRECT - Druhý byte je sečten s registrém X. Výsledek je adresa v nulté stránce, kde je umístěna adresa operandu.
8. INDIRECT INDEXED - Druhý byte instrukce dává adresu v nulté stránce paměti, kde je umístěna taková adresa, ze které vznikne adresa operandu po přičtení registru Y.
9. RELATIVE - K programovému čítači je přičtena hodnota druhého bytu instrukce. Výsledkem je nová adresa. Přičítaný byte je vyjádřen v druhém doplňku - jeho rozsah je tedy od -128 do 127.
10. INDIRECT - Druhý a třetí byte instrukce obsahují adresu v paměti, kde je uložena konečná adresa.

III. IMPLIED - určení operandů je součástí instrukce.

Obr.2 - Organizace paměti mikroprocesoru 6502



Struktura registru příznaků

Jednotlivým bitům registru příznaků jsou přiřazeny tyto významy:

- bit 0 CARRY (CY) - Tento bit je nastavován tehdy, když dochází-li při aritmetických operacích k přenosu sedmého bitu. Při logických funkcích je do něj možné přesunout obsah nultého nebo sedmého bitu.
- bit 1 ZERO (Z) - Jestliže výsledek některých aritmetických a logických funkcí je roven nule, potom se nastaví tento bit.
- bit 2 IRQ DISABLE (I) - Pomocí tohoto bitu je možné povolit přerušení (bit je roven nule), nebo jej zakázat.
- bit 3 DECIMAL MODE (D) - Je-li nastaven tento bit, pak je operand reprezentován jako dvě desítková čísla, která jsou vyjádřena v kódu BCD. Také pro instrukce sčítání a odčítání se provádí příslušné korekce tak, aby výsledek byl opět v BCD tvaru.
- bit 4 BRK COMMAND (B) - Po nastavení tento bit indikuje provedení instrukce BRK.
- bit 6 OVERFLOW (V) - Dochází-li k přenosu ze šestého bitu, potom je tento bit nastavován.
- bit 7 NEGATIVE (N) - Je-li číslo záporné (sedmý bit je roven jedné), je tento bit nastaven.

Bity 0, 6, 7 lze testovat a pomocí jejich hodnot větvit program.

Stručný popis jednotlivých instrukcí

Instrukce přesunů

Obsahy registrů A, X a Y mohou být naplňovány pomocí instrukcí LDA, LDX, LDY a uchovávány pomocí instrukcí STA, STX a STY. Operandy mohou být adresovány mnoha způsoby. O tom se můžete přesvědčit, podíváte-li se do seznamu všech instrukcí, kterou uvádíme v příloze.

Při práci s indexregistry X a Y si musíte dát pozor na použití indexovaného adresování. K indexování totiž není možné použít tentýž registr, se kterým se zrovna pracuje. Např. instrukce:

LDX HODN,Y
LDY HODN,X

jsou správné, ale instrukce:

LDX HODN,X
LDY HODN,Y

jsou nefunkční a nebudou vykonány.

Instrukce LDA, LDX a LDY neovlivňují příznakové bity C, I, D, ale mění příznaky Z a N v závislosti na přenášených datech.

Instrukce STA může použít libovolné způsoby adresace paměti. Při používání instrukcí STX a STY musí být použito pouze adresování nulté stránky, nebo adresování absolutní a nulté stránky s indexováním. Tyto instrukce (STA, STX a STY) neovlivňují žádné příznaky.

Další skupinou instrukcí přesunů jsou přesuny mezi registry A, X, Y a S. Těchto instrukcí je celkem šest: TAX, TAY, TSX, TXA, TXS a TYA. Druhé písmeno v mnemonickém zápisu instrukce označuje zdrojový registr a třetí písmeno označuje cílový registr. Tyto instrukce také ovlivňují příznaky N a Z.

Aritmetické instrukce

Do mikroprocesoru 6502 jsou implementovány dva druhy aritmetických operací: ADC pro sčítání a SBC pro odčítání.

Obě instrukce pracují vždy s přenosem - bit CARRY (CY), proto je nutné před každým sčítáním tento bit nulovat a před každým odčítáním nastavit (pokud není uvažován přenos z předcházející operace).

Operand je přičten (odečten) k akumulátoru, jehož hodnota je vzápěti přepsána výsledkem této operace. V závislosti na výsledku operace se nastavují příslušné bity v registru příznaků.

Při vykonávání aritmetických funkcí může procesor pracovat ve dvou módech - binárním nebo decimálním. V binárním módu reprezentuje každý byte číslo v rozsahu 0 až 255 v přímém módu, nebo v rozsahu -128 až 127, pokud použijete zobrazení v druhém doplňku. V desítkovém módu je každý byte rozdělen na dvě poloviny. Každá polovina určuje decimální číslo 0 - 9. V tomto módu tedy byte nabývá hodnoty 0 až 99. Např. má-li byte hodnotu 01101001, potom v binárním módu to reprezentuje hodnotu 106 (decimálně) a v decimálním módu hodnotu 69 (decimálně).

Hodnota bitu D v registru příznaků určuje, v jakém módu procesor zrovna pracuje. Je-li D rovno jedné, potom je nastaven decimální mód. Tento bit můžete nastavovat, nebo nulovat programově, a to pomocí instrukcí SED (pro D=1) a CLD (pro D=0).

Logické instrukce

Procesor 6502 má implementovány tyto základní logické funkce:

logický součin AND

logický součet ORA

exclusive OR EOR

Jednotlivé funkce jsou prováděny bit po bitu mezi obsahem akumulátore a operandem. Výsledek je poté uložen do akumulátoru. Podle typu výsledku jsou v registru příznaků nastavovány bity Z (je-li výsledek operace roven 0) a N (kopíruje sedmý bit výsledku).

Poněkud zvláštní je instrukce BIT, která provádí logický součet hodnot akumulátore a oprandu. Je podobná instrukci

AND, ale na rozdíl od ní se výsledek nikam neukládá. Výsledek ale nastavuje příznak Z (je-li výsledek nulový), příznak N (kopíruje hodnotu sedmého bitu operandu) a příznak V (kopíruje hodnotu šestého bitu operandu). Pomocí této instrukce se můžete dozvědět o vlastnosti X operandu, aniž byste museli porušit obsah akumulátoru nebo kteréhokoliv registru.

Mezi logické operace také řadíme instrukce porovnávání - CMP, CPX a CPY. Instrukce CMP porovnává obsah akumulátore a operandu. Podle výsledku nastavuje jednotlivé bity v registru příznaků. Nastavení probíhá podle této tabulky:

Příznaky:	Z	N	C
akumulátor je větší než operand	0	0	1
akumulátor je roven operandu	1	0	1
akumulátor je menší než operand	0	1	0

Instrukce CPX a CPY jsou podobné instrukci CMP. Liší se pouze v tom, že komparace se provádí s registrém X, respektivně registrém Y, místo akumulátoru.

Mezi další logické instrukce lze zařadit i instrukce posunů ROL a ROR, které provádí rotaci operandů vlevo nebo vpravo. Instrukce ROL přesouvá obsah CY do bitu 0 a obsah bitu 7 se přesouvá do CY, zatímco instrukce ROR přesouvá obsah CY do bitu 7 a obsah bitu 0 se přesouvá do CY.

Příklad:

	76543210	CY
před ROL	11100010	0
po ROL	11000100	1

	76543210	CY
před ROR	00100010	1
po ROR	10010001	0

Obdobou instrukce ROL je instrukce ASL. Tato instrukce funguje na stejném principu, pouze s tím rozdílem, že do bitu 0 operandu vsnuje vždy nulu bez ohledu na stav bitu CY. Komplementérní k ASL je instrukce LSR, která hodnoty bitů operandu posunuje vpravo a do sedmého bitu vsouvá nulu.

Instrukce inkrementací a dekrementací

Pro přičítání nebo odčítání jedničky od operantu jsou v procesoru instrukce inkrementace INC, INX, INY a dekrementace DEC, DEX, DEY. Instrukce INC a DEC pracují s operandy uloženými v paměti. Možné adresace operandů si můžete vyhledat ze seznamu instrukcí, který je uveden v příloze. Instrukce INX, INY, DEX a DEY se týkají indexregisterů X a Y. Podle dosažených výsledků jsou ovlivňovány příznaky Z a N v registru příznaků.

Prázdná operace

Procesor může vykonávat i tzv. prázdnou operaci. Zde se nevykonává žádná funkce, obsahy všech registrů zůstávají zachovány. Této instrukce se využívá v těch částech programů, které mají trvat přesně stanovenou dobu.

Instrukce skoků

Naprostá většina programů se neskládá ze sekvenčního řazení instrukcí (za sebou). Instrukce skoků proto umožňují předávat řízení (běh) programu na různé paměťové lokace. Skoky dělíme do dvou skupin - na nepodmíněné a podmíněné.

Při nepodmíněném skoku je řízení programu předáno na adresu, která je určena v instrukci, a to bez ohledu na stav programu. Tento typ instrukce označujeme jako JMP. Adresu můžeme určit dvěma způsoby: absolutně a nepřímo. Při absolutním adresování určuje druhý a třetí byte instrukce adresu, kam má být předáno řízení programu, zatímco při nepřímém adresování adresu, na které je uložena cílová adresa skoku.

Při podmíněném skoku je předání řízení podmíněno stavem registru příznaků. Pokud zadaná podmínka skoku není splněna, pokračuje se instrukcí, která následuje za instrukcí podmíněného skoku. Tato skupina instrukcí vyžaduje pouze relativní způsob adresování. Při relativním adresování určuje druhý byte instrukce relativní vzdálenost, kde se předává

řízení. Skákat lze maximálně o 128 bytů vzad nebo o 127 bytů vpřed, a to vzhledem k hodnotě programového čítače. Hodnota relativního skoku je tedy vyjádřena ve druhém doplňku.

Používání relativních skoků je velmi výhodné. Části programů, kde použijeme tyto relativní skoky, jsou nezávislé na umístění v paměti, a proto je můžeme libovolně přesouvat v libovolném paměťovém prostoru. Délka výsledného kódu je přitom kratší než při použití skoků s absolutním adresováním.

Instrukcí pro podmíněné skoky je celkem osm:

BCC ... skok při CY=0	BCS ... skok při CY=1
BNE ... skok při Z=0	BEQ ... skok při Z=1
BPL ... skok při N=0	BMI ... skok při N=1
BVC ... skok při V=0	BVS ... skok při V=1

K instrukcím skoků se přiřazuje i instrukce JSB, která se používá pro skok do podprogramu. Podrobněji se k ní ještě vrátíme.

Instrukce pro práci s registrum příznaků

Některé bity registru příznaků můžeme přímo programově ovládat. Lze je tedy nastavit, nebo nulovat bez provedení aritmetických nebo logických funkcí. Patří sem tyto instrukce:

CLC ... pro CY=0	SEC ... pro CY=1
CLD ... pro D=0	SED ... pro D=1
CLI ... pro I=0	SEI ... pro I=1
CLV ... pro V=0	

Instrukce pro práci se zásobníkem a podprogramy

Často se opakující sekvenči stejných příkazů můžeme v programu zapsat jen jednou, a to formou podprogramu. Instrukce pro volání podprogramu má mnemonické označení JSR. Je velmi podobná instrukci JMP, ale na rozdíl od ní vyžaduje i tzv. adresu návratu. Je to vlastně adresa místa, které

následuje za instrukcí volání podprogramu. Tato adresa se uloží do zásobníku, který je adresován pomocí registru S.

Návrat z podprogramu provádí instrukce RTS. Tato instrukce vyjme ze zásobníku návratovou adresu, na kterou předá další řízení programu. Tím se běh programu vrátí zpět za instrukci JSR, kterou byl daný podprogram vyvolán. Pozor, instrukce JSR používá pouze absolutní adresování.

Procesor 6502 má implementovanou ještě jednu instrukci pro návrat z podprogramu. Je to instrukce RTI, která se používá pro návrat z přerušení. Je-li běh procesoru přerušen, předá se řízení programu na specifikované adresy, které můžeme najít na konci přímo adresovatelného paměťového prostoru. Tato operace je podobná instrukci JSR, ale navíc se zde ukládá do zásobníku stavový registr. Návrat z přerušení zabezpečí právě instrukce RTI, která rovněž obnoví stavový registr.

Potřebujeme-li pro práci s podprogramem registry, jejichž obsah je však důležitý pro hlavní program, musíme jejich obsah uložit. Před návratem z podprogramu pak musíme opět obsah registrů obnovit. Uložení do zásobníku provádí instrukce PHA a PHP, pomocí nichž lze uchovat obsah akumulátora a stavového registru. Registry X a Y můžeme uložit pouze nepřímo, a to pomocí akumulátoru. Obnovu registrů na původní hodnotu provádí instrukce PLA a PLP. Ty přímo obnovují akumulátor a stavový registr, nepřímo přes akumulátor indexregistry.

Přerušení

Mikroprocesor 6502 umožňuje přerušení běhu programu vnějšími signály. Tyto signály jsou tři:

RESET - má nejvyšší prioritu zpracování. Pomocí něj se procesor dostává do výchozího stavu, tj. začíná pracovat od adresy, která je umístěna na paměťových lokacích \$FFFC (nižší byte) a \$FFFFD (vyšší byte).

NMI - namaskovatelné přerušení, má druhou nejvyšší prioritu.

Přijde-li tento signál, začíná program pracovat od adresy, jejíž hodnota je uložena v paměti na adresách \$FFFFAH (nižší byte) a SFFFFBH (vyšší byte). Zároveň se ukládá i obsah programového čítače a registru příznaků.

IRQ - žádost o přerušení. Toto přerušení může procesor programově ovládat, a to nastavením (přerušení zakázáno), nebo nulováním (přerušení povoleno) bitu I v registru příznaků. Po akceptování žádosti přerušení se uloží programový čítač a registr příznaků. Zároveň se nastaví bit I, tím se zakáže další přerušení, a nuluje se bit B, jehož nulová hodnota signalizuje přerušení. Programové řízení je pak předáno na adresu, která je uložena v paměťových lokacích \$FFFFEH (nižší byte) a FFFFH (vyšší byte). Po obsloužení je nutné znova povolit přerušení nulováním bitu I v registru příznaků.

Příloha I

Stručný seznam instrukcí

název	ovlivňuje	činnost
ADC	NV....ZC	součet operandu s akumulátorem a CY
AND	N....Z.	logický součin (AND) obsahu paměti a akumulátoru
ASL	N....ZC	aritmetický posun vlevo o 1 bit
BCC	podmíněný skok při CY=0
BCS	podmíněný skok při CY=1
BEQ	podmíněný skok při Z=1
BIT	NV....Z.	test bitu akumulátoru a paměti
BNI	podmíněný skok při N=1
BNE	podmíněný skok při Z=0
BPL	podmíněný skok při N=0
BRK	...B.I...	programové přerušení (BREAK)
BVC	podmíněný skok při V=0
BVS	podmíněný skok při V=1
CLCC	nulování příznaku carry (CY=0)
CLDD...	nulování příznaku decimal mode (D=0)
CLII..	nulování příznaku interrupt (I=0)
CLV	.V.....	nulování příznaku overflow (V=0)
CMP	N....ZC	porovnání obsahu paměti s akumulátorem
CPX	N....ZC	porovnání obsahu paměti s X-registrem
CPY	N....ZC	porovnání obsahu paměti s Y-registrem
DEC	N....Z.	zmenšení obsahu paměti o 1
DEX	N....Z.	zmenšení obsahu X-registru o 1
DEY	N....Z.	zmenšení obsahu Y-registru o 1
EOR	N....Z.	"exclusive or" obsahu paměti a akumulátoru
INC	N....Z.	zvýšení obsahu paměti o 1
INX	N....Z.	zvýšení obsahu X-registru o 1
INY	N....Z.	zvýšení obsahu Y-registru o 1
JMP	nepodmíněný skok na adresu
JSR	volání podprogramu
LDA	N....Z.	naplnění akumulátoru z paměti
LDX	N....Z.	naplnění index registru X z paměti
LDY	N....Z.	naplnění index registru Y z paměti
LSR	N....ZC	logický posun vpravo o 1 bit

název	ovlivňuje	činnost
NOP	prázdná operace
ORA	N.....Z.	logický součet (OR) obsahu paměti s akumulátorem
PHA	uložení akumulátoru do zásobníku
PHP	uložení registru příznaků do akumulátoru
PLA	N.....Z.	naplnění akumulátoru ze zásobníku
PLP	NV+BDIZC	naplnění stavového registru ze zásobníku
ROL	N.....ZC	rotace vlevo o 1 bit
ROR	N.....ZC	rotace vpravo o 1 bit
RTI	NV+BDIZC	návrat z přerušení
RTS	návrat z podprogramu
SBC	NV....ZC	odečtení obsahu paměti od akumulátoru s přenosem
SECC	nastavení CY=1
SEDD...	nastavení D=1
SEII...	nastavení I=i
STA	uložení obsahu akumulátore do paměti
STX	uložení obsahu indexregistru X do paměti
STY	uložení obsahu indexregistru Y do paměti
TAX	N.....Z.	přenos akumulátore do indexregistru X
TAY	N.....Z.	přenos akumulátore do indexregistru Y
TSX	N.....Z.	přenos ukazatele zásobníku do registru X
TXA	N.....Z.	přenos obsahu X-registru do akumulátoru
TXS	N.....Z.	přenos obsahu X-registru do ukazatele zásobníku
TYA	N.....Z.	přenos obsahu Y-registru do akumulátoru

Příloha 2

Seznam instrukcí mikroprocesoru 6502

(podle mnemonického označení)

Kód	Název instrukce	Způsob adresace	Hodinové cykly
109	\$6D	ADC	absolutní 4
125	\$7D	ADC	absolutní, X 4+1
121	\$79	ADC	absolutní, Y 4+1
105	\$69	ADC	bezprostřední 2
97	\$61	ADC	(nepřímé, X) 6
101	\$65	ADC	nultá stránka 3
117	\$75	ADC	nultá stránka, X 4
45	\$2D	AND	absolutní 4
61	\$3D	AND	absolutní, X 4+1
57	\$39	AND	absolutní, Y 4+1
41	\$29	AND	přímé 2
33	\$21	AND	(nepřímé, X) 6
49	\$31	AND	(nepřímé), Y 5+1
37	\$25	AND	nultá stránka 3
53	\$35	AND	nultá stránka, X 4
14	\$0E	ASL	absolutní 6
30	\$1E	ASL	absolutní, X 7
10	\$0A	ASL	akumulátor 2
6	\$06	ASL	nultá stránka 5
22	\$16	ASL	nultá stránka, X 6
144	\$90	BCC	relativní 2+1+1
176	\$B0	BCS	relativní 2+1+1
240	\$F0	BEQ	relativní 2+1+1
44	\$2C	BIT	absolutní 4
36	\$24	BIT	nultá stránka 3
48	\$30	BMI	relativní 2+1+1

Kód	Název instrukce	Způsob adresace	Hodinové cykly
208 \$D0	BNE	relativní	2+1+1
16 \$10	SPL	relativní	2+1+1
0 \$00	BRK	implicitní	7
80 \$50	BVC	relativní	2+1+1
112 \$70	BVS	relativní	2+1+1
24 \$18	CLC	implicitní	2
216 \$D8	CLD	implicitní	2
88 \$58	CLI	implicitní	2
184 \$B8	CLV	implicitní	2
205 \$CD	CMP	absolutní	4
221 \$DD	CMP	absolutní, X	4+1
217 \$D9	CMP	absolutní, Y	4+1
201 \$C9	CMP	bezprostřední	2
193 \$C1	CMP	(nepřímý, X)	6
209 \$D1	CMP	(nepřímý), Y	5+1
197 \$C5	CMP	nultá stránka	3
213 \$D5	CMP	nultá stránka, X	4
236 \$EC	CPX	absolutní	4
224 \$E0	CPX	bezprostřední	2
228 \$E4	CPX	nultá stránka	3
204 \$CC	CPY	absolutní	4
192 \$C0	CPY	bezprostřední	2
196 \$C4	CPY	nultá stránka	3
206 \$CE	DEC	absolutní	6
222 \$DE	DEC	absolutní, X	7
198 \$C6	DEC	nultá stránka	5
214 \$D6	DEC	nultá stránka, X	6
202 \$CA	DEX	implicitní	2
136 \$B8	DEY	implicitní	2
77 \$4D	EOR	absolutní	4
93 \$6D	EOR	absolutní, X	4+1

Kód	Název instrukce	Způsob adresace	Hodinové cykly
89	\$59	EOR	absolutní, Y 4+1
73	\$49	EOR	bezprostřední 2
65	\$41	EOR	(nepřímý, X) 6
81	\$51	EOR	(nepřímý), Y 5+1
69	\$45	EOR	nultá stránka 3
85	\$55	EOR	nultá stránka, X 4
238	\$EE	INC	absolutní 6
254	\$FE	INC	absolutní, X 7
230	\$E6	INC	nultá stránka 5
246	\$F6	INC	nultá stránka, X 6
232	\$E8	INX	implicitní 2
200	\$C8	INY	implicitní 2
76	\$4C	JMP	absolutní 3
108	\$6C	JMP	(nepřímý) 5
32	\$20	JSR	absolutní 6
173	\$AD	LDA	absolutní 4
189	\$BD	LDA	absolutní, X 4+1
185	\$B9	LDA	absolutní, Y 4+1
169	\$A9	LDA	bezprostřední 2
161	\$A1	LDA	(nepřímý, X) 6
177	\$B1	LDA	(nepřímý), Y 5+1
165	\$A5	LDA	nultá stránka 3
181	\$B5	LDA	nultá stránka, X 4
174	\$AE	LDX	absolutní 4
190	\$BE	LDX	absolutní, Y 4+1
162	\$A2	LDX	bezprostřední 2
166	\$A6	LDX	nultá stránka 3
182	\$B6	LDX	nultá stránka, Y 4
172	\$AC	LDY	absolutní 4
188	\$BC	LDY	absolutní, X 4+1
160	\$AO	LDY	bezprostřední 2

Kód		Název instrukce	Způsob adresace	Hodinové cykly
164	\$A4	LDY	nultá stránka	3
180	\$B4	LDY	nultá stránka, X	4
78	\$4E	LSR	absolutní	6
94	\$5E	LSR	absolutní, X	7
74	\$4A	LSR	akumulátor	2
70	\$46	LSR	nultá stránka	5
86	\$66	LSR	nultá stránka, X	6
234	\$EA	NOP	implicitní	2
13	\$0D	ORA	absolutní	4
29	\$1D	ORA	absolutní, X	4+1
25	\$19	ORA	absolutní, Y	4+1
9	\$09	ORA	bezprostřední	2
1	\$01	ORA	(nepřímý, X)	6
17	\$11	ORA	(nepřímý), Y	5+1
5	\$05	ORA	nultá stránka	3
21	\$15	ORA	nultá stránka, X	4
72	\$48	PHA	implicitní	3
8	\$08	PHP	implicitní	3
104	\$68	PLA	implicitní	4
40	\$28	PLP	implicitní	4
46	\$2E	ROL	absolutní	6
62	\$3E	ROL	absolutní, X	7
42	\$2A	ROL	akumulátor	2
38	\$26	ROL	nultá stránka	5
54	\$26	ROL	nultá stránka, X	6
110	\$6E	ROR	absolutní	6
126	\$7E	ROR	absolutní, X	7
106	\$6A	ROR	akumulátor	2
102	\$66	ROR	nultá stránka	5
118	\$76	ROR	nultá stránka, X	6
64	\$40	RTI	implicitní	6

Kód	Název instrukce	Způsob adresace	Hodinové cykly
96 \$60	RTS	implicitní	6
237 \$ED	SBC	absolutní	4
253 \$FD	SBC	absolutní, X	4+1
249 \$F9	SBC	absolutní, Y	4+1
233 \$E9	SBC	bezprostřední	2
225 \$E1	SBC	(nepřímý, X)	6
241 \$F1	SBC	(nepřímý), Y	5+1
229 \$E5	SBC	nultá stránka	3
245 \$F5	SBC	nultá stránka, X	4
56 \$39	SEC	implicitní	2
248 \$F8	SED	implicitní	2
120 \$78	SEI	implicitní	2
141 \$8D	STA	absolutní	4
157 \$9D	STA	absolutní, X	5
153 \$99	STA	absolutní, Y	5
129 \$81	STA	(nepřímý, X)	6
145 \$91	STA	(nepřímý), Y	6
133 \$85	STA	nultá stránka	3
149 \$95	STA	nultá stránka, X	4
142 \$8E	STX	absolutní	4
134 \$86	STX	nultá stránka	3
150 \$96	STX	nultá stránka, Y	4
140 \$8C	STY	absolutní	4
132 \$82	STY	nultá stránka	3
148 \$94	STY	nultá stránka, X	4
170 \$AA	TAX	implicitní	2
168 \$A8	TAY	implicitní	2
186 \$BA	TSX	implicitní	2
138 \$8A	TXA	implicitní	2
154 \$9A	TXS	implicitní	2
152 \$98	TYA	implicitní	2

Příloha 3

Seznam instrukcí procesoru 6502

(podle strojového kódu)

Kód	Název instrukce	Způsob adresace
0 \$00	BRK	implicitní
1 \$01	ORA	(nepřímý, X)
5 \$05	ORA	nultá stránka
6 \$06	ASL	nultá stránka
8 \$08	PHP	implicitní
9 \$09	ORA	bezprostřední
10 \$0A	ASL	akumulátor
13 \$0D	ORA	absolutní
14 \$0E	ASL	absolutní
16 \$10	BPL	relativní
17 \$11	ORA	(nepřímý), Y
21 \$15	ORA	nultá stránka, X
22 \$16	ASL	nultá stránka, X
24 \$18	CLC	implicitní
25 \$19	ORA	absolutní, Y
29 \$1D	ORA	absolutní, X
30 \$1E	ASL	absolutní, X
32 \$20	JSR	absolutní
33 \$21	AND	(nepřímý, X)
36 \$24	BIT	nultá stránka
37 \$25	AND	nultá stránka
38 \$26	ROL	nultá stránka
40 \$28	PLP	implicitní
41 \$29	AND	bezprostřední
42 \$2A	ROL	akumulátor
44 \$2C	BIT	absolutní
45 \$2D	AND	absolutní
46 \$2E	ROL	absolutní
48 \$30	BMI	relativní
49 \$31	AND	(nepřímý), Y
53 \$35	AND	nultá stránka, X

Kód	Název instrukce	Způsob adresace
54	\$36	ROL
56	\$38	SEC
57	\$39	AND
61	\$3D	AND
62	\$3E	ROL
64	\$40	RTI
65	\$41	EOR
69	\$45	EOR
70	\$46	LSR
72	\$48	PHA
73	\$49	EOR
74	\$4A	LSR
76	\$4C	JMP
77	\$4D	EOR
78	\$4E	LSR
80	\$50	BVC
81	\$51	EOR
85	\$55	EOR
86	\$56	LSR
88	\$58	CLI
89	\$59	EOR
93	\$5D	EOR
94	\$5E	LSR
96	\$60	RTS
97	\$61	ADC
101	\$65	ADC
102	\$66	ROR
104	\$68	PLA
105	\$69	ADC
106	\$6A	ROR
108	\$6C	JMP
109	\$6D	ADC
110	\$6E	ROR
112	\$70	BVS
117	\$75	ADC
118	\$76	ROR
120	\$78	SEI

Kód	Název instrukce	Způsob adresace
121 \$79	ADC	absolutní, Y
125 \$7D	ADC	absolutní, X
126 \$7E	ROR	absolutní, X
129 \$81	STA	(nepřímý, X)
132 \$84	STY	nultá stránka
133 \$85	STA	nultá stránka
134 \$86	STX	nultá stránka
136 \$88	DEY	implicitní
138 \$8A	TXA	implicitní
140 \$8C	STY	absolutní
141 \$8D	STA	absolutní
142 \$8E	STX	absolutní
144 \$90	BCC	relativní
145 \$91	STA	(nepřímý), Y
148 \$94	STY	nultá stránka, X
149 \$95	STA	nultá stránka, X
150 \$96	STX	nultá stránka, Y
152 \$98	TYA	implicitní
153 \$99	STA	absolutní, Y
154 \$9A	TXS	implicitní
157 \$9D	STA	absolutní, X
160 \$A0	LDY	bezprostřední
161 \$A1	LDA	(nepřímý, X)
162 \$A2	LDX	bezprostřední
164 \$A4	LDY	nultá stránka
165 \$A5	LDA	nultá stránka
166 \$A6	LDX	nultá stránka
168 \$A8	TAY	implicitní
169 \$A9	LDA	bezprostřední
172 \$AC	LDY	absolutní
173 \$AD	LDA	absolutní
174 \$AE	LDX	absolutní
176 \$B0	BCS	relativní
177 \$B1	LDA	(nepřímý), Y
180 \$B4	LDY	nultá stránka, X
181 \$B5	LDA	nultá stránka, X
182 \$B6	LDX	nultá stránka, Y

Kód	Název instrukce	Způsob adresace
184	\$B8	CLV
185	\$B9	LDA
186	\$BA	TSX
188	\$BC	LDY
189	\$BD	LDA
190	\$BE	LDX
192	\$C0	CPY
193	\$C1	CMP
196	\$C4	CPY
197	\$C5	CMP
198	\$C86	DEC
200	\$C8	INY
201	\$C9	CMP
202	\$CA	DEX
204	\$CC	CPY
205	\$CD	CMP
206	\$CE	DEC
208	\$D0	BNE
209	\$D1	CMP
213	\$D5	CMP
214	\$D6	DEC
216	\$D8	CLD
217	\$D9	CMP
221	\$DD	CMP
222	\$DE	DEC
224	\$EC	CPX
225	\$E1	SBC
228	\$E4	CPX
229	\$E5	SBC
230	\$E6	INC
232	\$E8	INX
233	\$E9	SBC
234	\$EA	NOP
236	\$EC	CPX
237	\$ED	SBC
238	\$EE	INC
240	\$F0	BEQ

Kód		Název instrukce	Způsob adresace
241	\$F1	SBC	(nepřímý), Y
245	\$F5	SBC	nultá stránka, X
246	\$F6	INC	nultá stránka, X
248	\$F8	SED	implicitní
249	\$F9	SBC	absolutní, Y
253	\$FD	SBC	absolutní, X
254	\$FE	INC	absolutní, X

Příloha 4

Přehled mnemonických názvů instrukcí procesoru 6502

ADC	Add Memory to Accumulator with Carry
AND	"AND" Memory with Accumulator
ASL	Shift Left One Bit (Memory or Accumulator)
BCC	Branch on Carry Clear
BCS	Branch on Carry Set
BEQ	Branch on Result Zero
BIT	Test Bits in Memory with Accumulator
NMI	Branch on Result Minus (Negative)
BNE	Branch on Result not Zero
BPL	Branch on Result Plus (Positive)
BRK	Force Break
BVC	Branch on Overflow Clear
BVS	Branch on Overflow Set
CLC	Clear Carry Flag
CLD	Clear Decimal Mode Flag
CLI	Clear Interrupt Disable Flag
CLV	Clear Overflow Flag
CMP	Compare Memory and Accumulator
CPX	Compare Memory and Index Register X
CPY	Compare Memory and Index Register Y
DEC	Decrement Memory by One
DEX	Decrement Index Register X by One
DEY	Decrement Index Register Y by One
EOR	"Exclusive-OR" Memory with Accumulator
INC	Increment Memory by One
INX	Increment Index Register X by One
INY	Increment Index Register Y by One
JMP	Jump to New Location
JSR	Jump to New Location Saving Return Address
LDA	Load Accumulator with Memory
LDX	Load Index Register X with Memory
LDY	Load Index Register Y with Memory
LSR	Shift Right One Bit (Memory or Accumulator)
NOP	No Operation

ORA	"OR" Memory with Accumulator
PHA	Push Accumulator on Stack
PHP	Push Processor Status on Stack
PLA	Pull Accumulator from Stack
PLP	Pull Processor Status from Stack
ROL	Rotate One Bit Left (Memory or Accumulator)
ROR	Rotate One Bit Right (Memory or Accumulator)
RTI	Return from Interrupt
RTS	Return from subroutine
SBC	Substract Memory from Accumulator with Borrow
SEC	Set Carry Flag
SED	Set Decimal Mode Flag
SEI	Set Interrupt Disable Flag
STA	Store Accumulator in Memory
STX	Store Index Register X in Memory
STY	Store Index Register Y in Memory
TAX	Transfer Accumulator to Index Register X
TAY	Transfer Accumulator to Index Register Y
TSX	Transfer Stack Pointer to Index Register X
TXA	Transfer Index Register X to Accumulator
TXS	Transfer Index Register X to Stack Pointer
TYA	Transfer Index Register Y to Accumulator

Obsah:

I.	Mikroprocesor 6502	1
	Organizace paměti	1
	Způsoby adresování operandů instrukcí	3
	Struktura registru příznaků	5
II.	Stručný popis jednotlivých instrukcí	6
	Instrukce přesunů	6
	Aritmetické instrukce	6
	Logické instrukce	7
	Instrukce inkrementaci a dekrementaci	9
	Prázdná operace	9
	Instrukce skoků	9
	Instrukce pro práci s registrém příznaků	10
	Instrukce pro práci se zásobníkem a podprogramy	10
	Přerušení	11
III.	Příloha 1	13
	Stručný seznam instrukcí	
	Příloha 2	15
	Seznam instrukcí 6502 (podle mnemoniky)	
	Příloha 3	20
	Seznam instrukcí 6502 (podle ML kódu)	
	Příloha 4	25
	Přehled mnemonických názvů instrukcí 6502	
IV.	Obsah	27

ATARI KLUB OLOMOUC

VÁM NABÍZÍ

Atari zpravodaj Olomouc 1/1991
kompletní ročník AZ Olomouc 1990
jednotlivá čísla AZ 5-6, 7-8/89, 3-4, 5-6 a 7-8/88
Makroknihovna - rutiny asembleru
Kniha her pro Atari - manuály 70 her
kazeta Kniha her pro Atari - výběr nejlepších 55 her
ze stejnojmenné příručky (TURBO 2000)
kazety i diskety

Clip-Art #1 The Print Shop a #2 The Newsroom
sada 35 standardních fontů
soubor 90 programů z AZ Olomouc 1989 a 1990
programy uvedené v AZ Olomouc 1/1991

Schůzky probíhají v místnostech autoškoly Na Šibeníku od 9.00 do 13.00 hod. Druhá sobota v měsíci - klub disket, třetí sobota v měsíci - 8-bit klub a čtvrtá sobota v měsíci - klub ST. Srdečně zveme.

Literaturu a programy si můžete objednat na adresě:
Atari klub Olomouc PS 137
772 11 Olomouc

Zásilky zasíláme dobirkou do tří týdnů.

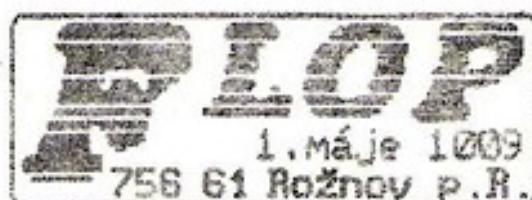
(c) 1991
by GIA Software
Atari 1040 STFM
DTP Calamus



Pro všechny uživatele ATARI XL/XE zajišťujeme tyto služby:

- vydávání disketového zpravodaje FLOP
- kazetová verze zpravodaje FLOP
- tisknutý zpravodaj OSČASNÍK
- informační BULLETIN FLOPU
- úpravy počítačů, datasetů, doplňky
- poštovní zásilková služba publikací a doplňků
- ediční a vydavatelská činnost pro kluby i jednotlivce
- konzultace, poradenská činnost

Ze starších publikací a zpravodajů pořídíme na požádání kopie.



Mikroprocesor 6502

POPIS INSTRUKCÍ

Vydal:

FLOP ROŽNOV 1992

Odpovědný redaktor: Lýdie Kolaříková

Rozsah: 28 stran

Tisk:

FLOP ROŽNOV 1992

Náklad:

500 ks

Přetisk pouze se svolením vydavatele.