





ZPRAVODAJ

ATARI
Klub Praha

příloha III

ING. TOMÁŠ RYŠAVÝ

Mikroprocesory 65C02 a R6520 (PIA)

Vydává 487. ZO Svazarmu —
ATARI KLUB v Praze 4.

Šéfredaktor a vedoucí redakční rady
JUDr. Jan Hlaváček.

Zástupce šéfredaktora ing. Stanislav
Borský.

Obálku navrhl RNDr. J. Tamchyna.

Adresa redakce:

487. ZO Svazarmu - ATARI KLUB Praha
REDAKCE

poštovní přihrádka 51
100 00 Praha 10

Řídí redakční rada: V. Bílek, ing. J. Bis-
kup, RNDr. J. Bok, CSc., ing. S. Borský,
ing. V. Friedrich, ing. O. Hanuš, RNDr.
L. Hejna, CSc., Z. Lazar, prom. fyz.,
CSc., ing. M. Vavrda.

Otisk povolen se souhlasem redakce
při zachování autorských práv a s uve-
dením pramene. Rukopisy nevyžáda-
né redakcí se nevracejí. Za původnost
a věcnou správnost ručí autor.

Vychází šestkrát ročně. Neprodejně.
Členům klubu distribuováno zdarma.
Nepravidelné přílohy na objednávku
jsou kompenzovány zvláštním klubo-
vým příspěvkem.

Rozsah čísla 21 stran. Neprošlo
jazykovou úpravou.

Tiskne Ústř. kulturní dům železni-
čářů, Praha 2, nám. Míru 9

Do tisku předáno 11/88

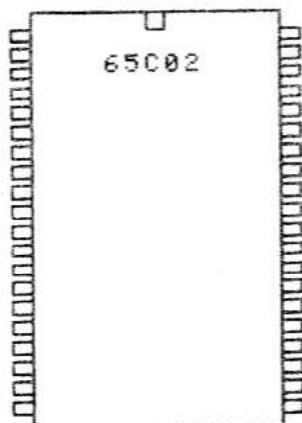
Vydávání schváleno OV Svazarmu
Praha 4 a OŠK ONV Praha 4.

Evidenční číslo ÚVTEI 86 042.

© ATARI KLUB Praha, 1988

μ P 65C02

CMOS 8-Bit Microprocessor Family



OBSAH:

- popis μ P 65C02
- adresovaci mody
- instrukcni soubor
- popis signalu
- programovaci model
- popis R6520 (PIA)

POPIS P 65C02

Mikroprocesor 65C02, který je vyráběn několika světovými výrobci polovodivů např. fy. Rockwell int., Synertek atd. je osmibitová paralelní procesorová jednotka, která je svou velkou výkoností předurčena pro řízení malých výpočetních systémů a periférií. Je vyrobena CMOS technologií s vodivostním kanálem typu N. Tato verze známého procesoru 6502, je díky vylepšené technologii a architektuře značně výkonnější než původní procesor 6502. Výkonnost je ještě zvětšena 27 novými instrukcemi a dvěma novými adresovacími módy. Tri-stavový 16-bitový adresový bus dovoluje adresovat paměťový prostor 64k bytů a umožňuje přímý přístup do paměti (DMA).

Základní vlastnosti :

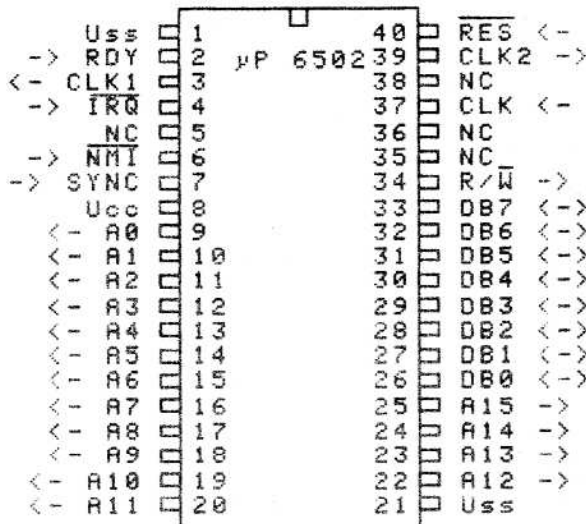
- * 8-bitové paralelní zpracování dat
- * Přímý adresovatelný paměťový prostor 64k bytů
- * CMOS technologie s vodivostním kanálem typu N
- * Jednoduché 5V napájení
- * Spotřeba 8mA při 4MHz
- * Desítková nebo dvojková aritmetika
- * 56 instrukcí z 6502 + 27 nových instrukcí
- * 13 adresovacích módů + 2 nové
- * Rozsáhlá indexovací schopnost
- * Programovatelný ukazatel zásobníku
- * Proměnná délka zásobníku
- * Možnost prerušení
- * Možnost DMA
- * Architektura pipeline
- * Hodinový kmitočet 1-4 MHz
- * Rozsáhlý soubor podpurných obvodů

ORGANIZACE PAMĚTI 6502

IRQH	FFFFH
IRQL	FFFEH
RESH	FFFDH
RESL	FFFCH
NMIH	FFFBH
NMIL	FFFAH
Volná oblast	FFF9H
	0200H
ZASOBNÍK	01FFH
	0100H
Volná oblast	00FFH
	0000H

PIN CONFIGURATION

µP 65C02



POPIS SIGNALU :

Hodiny (CLK1, CLK2)

Procesor 6502 vyžaduje dvoufázové synchronní hodiny pracující s úrovněmi Ucc.

Adresová sběrnice (A0-A15)

Vstupy jsou kompatibilní s TTL, připojit je možno jedno standardní hradlo TTL (30 pF).

Datová sběrnice (DB0-DB7)

Osmibitová datová sběrnice je obousměrná a slouží k přesunu dat mezi procesorem, pamětí a periferiemi. Tristavový výstup je sloučen s TTL a umožňuje připojit jedno hradlo TTL (30 pF).

Reset (RES)

Tento vstup je využíván pro uvedení procesoru do počátečního stavu. Při úrovni L na tomto vstupu je zakázána výměna dat. Jestliže je detekována vzestupná hrana, je zahájeno resetování. Po systémové inicializaci která trvá šest hodinových cyklů je nastaven příznak prerušení a mikroprocesor uloží do PC obsah adresy FFFC a FFFD. Toto je počáteční umístění programového řízení. Jestliže napětí Ucc dosáhne 4.75V při zapínání musí být zajisteno přidržení úrovně L po dobu nejméně dva hodinové cykly. Po této době je zajistena platnost signálu R/W.

Zadost o preruseni (IRQ)

Tento vstup slucitelny s TTL provadi zadost periferie o preruseni. Mikroprocesor provede dokonceni prave provedene instrukce, po dokonceni bude testovat stav priznaku preruseni v priznakovem registru. Jestliže priznak neni nastaven, mikroprocesor provede preruseni. Programovy citac (PC) a priznakove registr (S) jsou ulozeny do zasobnikove pameti. Dale bude nastaven priznak preruseni, aby nemohlo dojít k dalsimu preruseni. Po ukonceni teto casti bude do programoveho citace (PC) ulozen obsah adres FFFE a FFFF, a podle tohoto obsahu bude predano rizeni.

Cteni/Zapis (R/W)

Tento signal je generovan procesorem, ridi pohyb dat na datove sbornici. Signal je ve stavu H, pouze kdyz procesor uklada data do pameti nebo na periferie je ve stavu L.

ADRESOVACI MODY :

- a. Accumulator addressing - Tato forma adresovani je representovana jednobytovou instrukci, zahrnujici operace ve stradaci.
- b. Immediate adresing - U teto forme je operand obsazen v druhem bytu instrukce, bez potreby dalsiho adresovani.
- c. Absolute addressing - U absolutnim adresovani druhy byte instrukce specifikuje nizsi cast efektivni adresy a treti byte vyssich 8 bitu adresy. Timto modem adresovani je primum pristup k 64k bytum adresniho prostoru (max. FFFFH).
- d. Zero page addressing - tyto instrukce dovoluji pro kratsi operacni kody a casove zpracovani pouze vykonat druhy byte instrukce a prijmut nulovou vyssi cast adresy. Opatrne vyuzivani nulte stranky ma za nasledek velke zvyšení efektivity programu.
- e. Indexed zero page addressing - (X,Y indexovani) Tato forma adresovani se vyuziva ve spojeni s X,Y index registry a je odvolavana jako 'Absolute,X' a 'Absolute,Y'. Efektivni adresa je vytvorena pridaním obsahu X a Y k adrese obsazene ve druhem a tretim bytu instrukce. Tento typ indexovani umoznuje jakokoliv umistovani a modifikovani mnohonasobnych poli v pameti.
- f. Implied addressing - U 'obsahovem' adresovani, adresa obsahujici operand je zahrnuta v operacnim kodu instrukce.
- g. Relative addressing - Relativni adresovani je pouzito pouze s vetvicimi instrukcemi. Druhy byte instrukce se stava operandem, který je jako 'offset' pridan k obsahu nizsich osmi bitu programoveho citace, kdyz citac je nastaven na dalsi instrukci. Rozsah posunutí je -128 az +127 bytu z dalsi instrukce.

- h. Indexed indirect addressing - V indexovanem nepřímém adresování, označovaném jako [Indirect,X] je druhý byte instrukce přidán k obsahu index registru X, bez přenosu. Výsledek tohoto součtu ukazuje na obsah paměti v nulté stránce, kde je uloženo nízších 8 bitů efektivní adresy. Vyšších 8 bitů efektivní adresy je uloženo na vedlejší bunce paměti v nulté stránce. Obe části efektivní adresy musí být uloženy v nulté stránce.
- i. Indirect indexed addressing - V nepřímém indexovaném adresování, označovaném [Indirect],Y, ukazuje druhý byte instrukce na paměťové místo v nulté stránce. Obsah tohoto místa je přidán k obsahu Y index registru. Výsledkem je nízších 8 bitů efektivní adresy. Přenos z tohoto součtu je přidán k obsahu dalšího paměťového v nulté stránce. Výsledkem je vyšších 8 bitů efektivní adresy.
- j. Absolute indirect - Druhý byte instrukce obsahuje nízších 8 bitů paměti. Vyšších 8 bitů je obsazeno v třetím byte instrukce. Obsah plně specifikované bunky paměti ukazuje na nízších 8 bitů efektivní adresy. Další bunka paměti obsahuje vyšší část efektivní adresy.
- k. Indexed absolute indirect - je druh adresování, kdy druhý a třetí byte instrukce je přičten k obsahu index registru X. Výsledkem je 16-ti bitová adresa, která ukazuje na nízších osm bitů efektivní adresy. Vedlejší paměťové místo obsahuje vyšších osm bitů efektivní adresy.
- l. Indirect - V nepřímém adresování ukazuje druhý byte instrukce na paměťové umístění v nulté stránce, jehož obsahem je nízších osm bitů efektivní adresy. Vedlejší paměťové místo obsahuje vyšších osm bitů efektivní adresy.

POZN.: Adresovací módy k. a l. jsou platné pouze pro 65C02

INSTRUKČNÍ SOUBOR - Abecedně

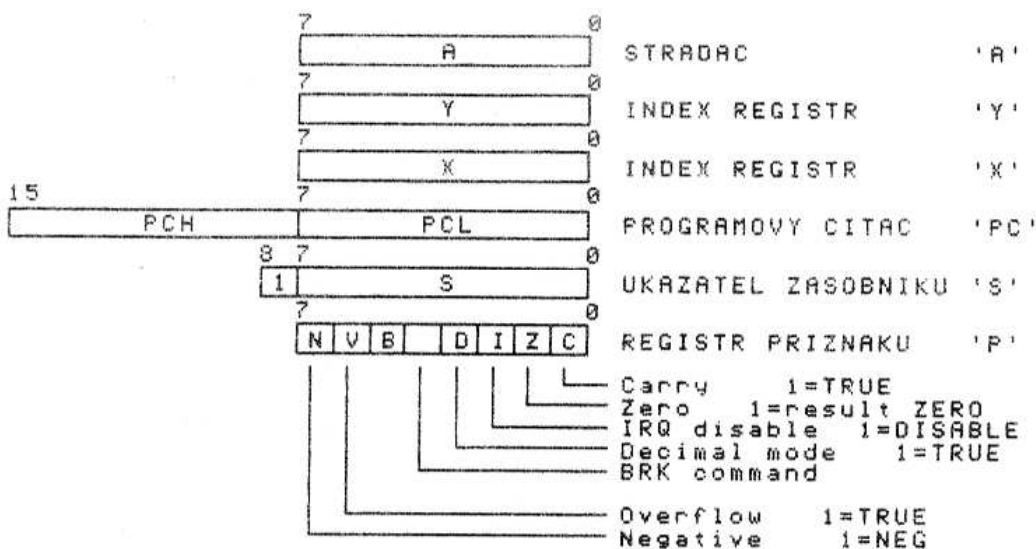
ADC	Součet operandů se strádacem a CY
AND	Logické 'AND' obsahu paměti a strádace
ASL	Posun vlevo o 1 bit (obsahu paměti nebo strádace)
BCC	Podmíněný skok při CY=0
BCS	Podmíněný skok při CY=1
BEQ	Podmíněný skok při Z=1
BIT	Test bitu paměti a strádace
BMI	Podmíněný skok při N=1
BNE	Podmíněný skok při Z=0
BPL	Podmíněný skok při N=0
BRA *	Relativní okamžitý skok
BRK	Instrukce BREAK
BVC	Podmíněný skok při V=0
BVS	Podmíněný skok při V=1

CLC	Nulovani priznaku CY (Carry)
CLD	Nulovani priznaku D (Decimal mode)
CLI	Nulovani priznaku I (Interrupt)
CLV	Nulovani priznaku V (Overflow)
CMP	Porovnani obsahu pameti se stradacem
CPX	Porovnani obsahu pameti se Index reg.X
CPY	Porovnani obsahu pameti se Index reg.Y
DEA	* Zmenseni obsahu stradace o 1
DEC	Zmenseni obsahu pameti o 1
DEX	Zmenseni obsahu Index registru X o 1
DEY	Zmenseni obsahu Index registru Y o 1
EOR	'Exclusive-OR' obsahu pameti a stradace
INA	* Zvyseni obsahu stradace o 1
INC	Zvyseni obsahu pameti o 1
INX	Zvyseni obsahu Index registru X o 1
INY	Zvyseni obsahu index registru Y o 1
JMP	Nepodmineny skok
JSR	Skok do podprogramu
LDA	Naplneni stradace z pameti
LDX	Naplneni Index registru X z pameti
LDY	Naplneni Index registru Y z pameti
LSR	Posun vpravo o 1 bit (obsahu pameti nebo stradace)
NOP	Prazdna operace
ORA	Logicke 'OR' obsahu pameti se stradacem
PHA	Ulozeni stradace do zasobniku
PHP	Ulozeni registru priznaku do stradace
PHX	* Ulozeni registru X do zasobniku
PHY	* Ulozeni registru Y do zasobniku
PLA	Vyjmuti obsahu vrcholu zasobniku do stradace
PLP	Vyjmuti obsahu vrcholu zasobniku do registru priznaku
PLX	* Vyjmuti obsahu vrcholu zasobniku do registru X
PLY	* Vyjmuti obsahu vrcholu zasobniku do registru Y
ROL	Rotace vlevo o 1 bit (obsahu pameti nebo stradace)
ROR	Rotace vpravo o 1 bit (obsahu pameti nebo stradace)
RTI	Navrat z preruseni
RTS	Navrat z podprogramu
SBC	Odecteni obsahu pameti od stradace s vypujcenim
SEC	Nastaveni CY na 1
SEI	Nastaveni D na 1
STA	Ulozeni obsahu stradace do pameti
STZ	* Ulozeni nuloveho obsahu do pameti
STX	Ulozeni obsahu Index registru X do pameti
STY	Ulozeni obsahu Index registru Y do pameti

TAX Prenos stradace do Index registru X
 TAY Prenos stradace do Index registru Y
 TRB * Testovani a nastavovani bitu v pameti se stradacem
 TSX Prenos ukazatele zasobniku do stradace
 TXA Prenos Index registru X do stradace
 TXS Prenos Index registru X do ukazatele zasobniku
 TYA Prenos Index registru Y do stradace

POZN.: instrukce oznacene * jsou platne pouze pro
 verzi mikroprocesoru 65C02

PROGRAMOVACI MODEL



MNEM.	OP	NUBDIZC	A#ZBIRX@	T	UZNAM	POZN.
ADC	s: 6D	**---**	XxX	X	:4: Soucet s carry	:A=A+s+CY %:
AND	s: 2D	*---*	XxX	X	:4: Logicke AND	:A=A&s %:
ASL	d: 0E	*---**	xx		:6: Aritmeticky posun vlevo	:d=(CY,d,0)<-:
ASLA	0A	*---**	X		:2: Aritmeticky posun vlevo	:A=(CY,d,0)<-:
BCC	a: 90	-----		X	:2: Vetveni jestlize CY=0	:If CY=0 %:
BCS	a: 80	-----		X	:2: Vetveni jestlize CY=1	:If CY=1 %:
BEQ	a: F0	-----		X	:2: Vetveni pri rovnosti	:If Z=1 %:
BIT	s: 2C	**---*	**		:4: Testovani bitu	:A&s :
BMI	a: 30	-----		X	:2: Vetveni pri minus	:If N=1 %:
BNE	a: D0	-----		X	:2: Vetveni pri nerovnosti	:If Z=0 %:
BPL	a: 10	-----		X	:2: Vetveni pri plus	:If N=0 %:
BRK	00	--+1--		X	:7: BREAK (-[S]=(PC+2))	:PC=(FFFEH) BREAK :
BUC	a: 50	-----		X	:2: Vetveni pri nepreteceni	:If U=0 %:
BUS	a: 70	-----		X	:2: Vetveni pri preteceni	:If U=1 %:
CLC	: 18	-----0		X	:2: Clear carry	:CY=0 :
CLD	: D8	---0---		X	:2: Clear decimal mode	:D=0 :
CLI	: 58	---0---		X	:2: Clear Int.disable	:I=0 :
CLV	: B8	-0-----		X	:2: Clear overflow flag	:U=0 :
CMP	s: CD	*---**	XxX	X	:4: Porovnani	:A-s :
CPX	s: EC	*---**	X**		:4: Porovnani s index registrem	:X-s :
CPY	s: CC	*---**	X**		:4: Porovnani s index registrem	:Y-s :
DEC	d: CE	*---*	xx		:6: Decrement	:d=d-1 :
DEX	: CA	*---*		X	:2: Decrement index registru	:X=X-1 :
DEY	: 88	*---*		X	:2: Decrement index registru	:Y=Y-1 :
EOR	s: 4D	*---*	UuU	U	:4: Logicke exclusiv OR	:A=AxS :
INC	d: EE	*---*	xx		:6: Increment	:d=d+1 :
INX	: E8	*---*		X	:2: Increment index registru	:X=X+1 :
INY	: C8	*---*		X	:2: Increment index registru	:Y=Y+1 :
JMP	s: 4C	-----	*	X	:3: Skok	: :
JSR	s: 20	-----	*		:6: Skok na podprogram	:-[S]=PC+2 !:
LDA	s: AD	*---*	XxX	X	:4: Load Accumulator	:A=s %:
LDX	s: AE	*---*	Xyy		:4: Load index registr	:X=s %:
LDY	s: AC	*---*	Xxx		:4: Load index registr	:Y=s %:
LSR	d: 4E	0---**	xx		:6: Logicky posun vpravo	:d=->(0,d,CY):
LSRA	: 4A	0---**	X		:2: Logicky posun vpravo	:A=->(0,A,CY):
NOP	: EA	-----		X	:2: No operation	: :
ORA	s: 0D	*---*	XxX	X	:4: Logicke inclusiv OR	:A=AvS :
PHA	: 48	-----		X	:3: Uloz accumulator	:-[S]=A :
PHP	: 08	-----		X	:3: Uloz stavovy registr	:-[S]=P :
PLA	: 68	-----		X	:4: Vyber accumulator	:A=[S]+ :
PLP	: 28	*****		X	:4: Vyber stavovy registr	:p=[S]+ :
ROL	d: 2E	*---**	xx		:6: Rotace vlevo	:d=(CY,d)<- :
ROLA	: 2A	*---**	X		:2: Rotace accumulatoru vlevo	:A=(CY,A)<- :
ROR	d: 6E	*---**	xx		:6: Rotace vpravo	:d=->(CY,d) :
RORA	: 6A	*---**	X		:2: Rotace accumulatoru vpravo	:A=->(CY,d) :
RTI	: 40	*****		X	:6: Navrat z preruseni	:<PC,P>=[S]+ :
RTS	: 60	-----		X	:6: Navrat z podprogramu	:PC=(S)+1 :
SBC	s: ED	*---**	XxX	X	:4: Odecitani s carry	:A=A-s-CY %:
SEC	: 38	-----1		X	:2: Nastaveni carry flagu	:CY=1 :
SED	: FB	---1---		X	:2: Nastaveni desitkoveho modu	:D=1 :
SEI	: 78	---1---		X	:2: Nastaveni zakazani preruseni	:I=1 :
STA	d: 8D	-----	xX	X	:4: Store accumulator	:d=A :
STX	d: 8E	-----	x*		:4: Store index registr	:d=X :
STY	d: 8C	-----	y*		:4: Store index registr	:d=Y :

MNEM.	OP	NUBDIZC	A#ZBIRX	T	UYZNAM
TAX	AA	*---*	X	:2	Premísteni accumulatoru do X: X=A
TAY	AB	*---*	X	:2	Premísteni accumulatoru do Y: Y=A
TSX	BA	*---*	X	:2	Premísteni ukaz. zas. do X : X=S
TXA	8A	*---*	X	:2	Premísteni index reg. do A : A=X
TXS	9A	-----	X	:2	Premísteni X do ukaz.zas. S : S=X
TYA	9B	*---*	X	:2	Premísteni index reg. do A : A=Y
	XX			X	Hexadec. oper. kod / počet cyklu
	-				Flag neovlivnen
	*				Flag ovlivnen
	0				Flag reset
	1				Flag nastaven
	+				Flag nastaven se stackem
N	N				Priznak zaporu (bit 7)
U	U				Priznak pretečení (bit 6)
B	B				Indicator BREAK (bit 4)
D	D				Rizení desítkového modu (bit 3)
I	I				Rizení povolení prerušeni (bit 2)
Z	Z				Priznak nuly (bit 1)
C	C				Priznak prenosu (bit 0)

POZN.: Doplněk pro 65C02

Rozšířeny instrukční soubor:

Mnem.	Op.kod[HEX]	Mnem.	Op.kod[HEX]
BRA	80	DEA	3A
INA	1A	PHX	DA
PHY	5A	PLX	FA
PLY	7A	STZ	9C (Absolute)
STZ	9E (Absolute,X)	STZ	64 (Zero Page)
STZ	74 (Zero Page,X)	TSB	1C (Absolute)
TSB	14 (Zero Page)	TSB	0C (Absolute,X)
TSB	04 (Zero Page,X)		

Nové adresovací módy:

Mnem.	Op.kod[HEX]	
JMP	7C	Skok (Indirect absolute,X)
ADC	72	Součet paměti se adresací a CY (Indirect)

Mnem. Op.kod[HEX]

AND	32	'AND' pameti se stradacem (Indirect)
CMP	D2	Porovnaní obsahu pameti a stradace (Indirect)
EOR	52	'Exclusiv OR' pameti se stradacem (Indirect)
LDA	82	Uložení obsahu pameti do stradace (Indirect)
ORA	12	'OR' pameti a stradace (Indirect)
SBC	F2	Odečtení pameti od stradace s CY (Indirect)
STA	92	Uložení obsahu stradace do pameti (Indirect)

MNEMONICKY	A#ZBIRKA	POPIS	OP.KOD	s
	X	Platne pro vsechny mody		
	*	Platny ne-indexni mod		
	x	X/ne-indexni mod platny		
	y	Y/ne-indexni mod platny		
		Prictení XXH k op.kodu	+XXH	
		Odectení XXH od op.kodu	-XXH	
		Prictení X v cyklu		+X
		Odectení x v cyklu		-X
A	A	Accumulator		
#n	#	Primo	-04H	-2
<n	*	Nulta stranka (Zero page)	-08H	-1
n	*	Nulta stranka (DIRECT mod)	-08H	-1
n,X	x	Nulta stranka indexovana (X)	+08H	+0
n,Y	y	Nulta stranka indexovana (Y)	+08H	+0
>nn	*	Absolutne	+00H	+0
nn	*	Absolutne (Extend mod)	+00H	+0
nn,X	x	Absolutne indexovane (X)	+10H	+0
nn,Y	y	Absolutne indexovane (Y)	+00H	+0
LDX nn,Y	y	"_"	+10H	+0
	I	Implicitne		
a	R	Relativne (PC=PC+1+offset)		+2
[nn,X]	x	Neprime indexovani (X)	-0CH	+2
[nn],Y	y	Neprime indexovani (Y)	+04H	+1
[nn]	R	Absolutne neprimo	+20H	+2
BYTE n(.....)		Byte(y) (8-bit)		
BYTE 'string'(.....)		Byte textoveho(ch) retezce(u)		
DIRECT		Mod adresovani nulte stranky		
EXTEND		Absolutni adresovaci mod		
RMB nn(.....)		Reservni byte pameti (Reserve memory bytes)		
WORD nn(.....)		Slovo(a) (16-bit)		
A		Accumulator (8-bit)		
P		Stavovy registr (status) (8-bit)		
PC		Programovy citac (Program counter) (16-bit)		
S		Ukazatel zasob.pameti (9-bit MSB=1)		
X		Index registr X (8-bit)		
Y		Index registr Y (8-bit)		
a		Relativni adresa (-128 az +127)		
d		Urceni		
n		8-bit vyraz (0 - 255)		
nn		16-bit vyraz (0 - 65535)		
s		Zdrojovy		
string		Retez ASCII znaku		

MNEMONICKY	POPIS
+	: Aritmeticky soucet
-	: Aritmeticky rozdil
*	: Aritmeticke nasobeni
/	: Aritmeticke deleni
&	: Logicke AND
	: Logicke NOT
v	: Logicke inclusiv OR
x	: Logicke exclusiv OR
<-	: Rotace vlevo
->	: Rotace vpravo
[]	: Neprime adresovani
[]+	: Neprime adresovani, auto-increment
-[]	: Auto-decrement, neprime adresovani
< >	: Kombinace operandu
\$: Obsah programoveho citace
%	: s=s+1 jestliže presahne hranice stranky
!	: PC -efektivni adresa zdroje
-->	: Ustup na pin
<--	: Vystup z pinu
<-->	: Ustup/vystup pinu
0000H - 00FFH	: PAGE 0 (viz.adresovani nulte stranky)
0100H - 01FFH	: PAGE 1 (umisteni stacku, 01FFH =start)
XX00H - XXFFH	: PAGE n (kde n=XXH)
FFFAH - FFFBH	: Nemaskovatelné preruseni (NMI vector)
FFFCH - FFFDH	: Reset (RESET vector)
FFFEH - FFFFH	: Zadost o preruseni (IRQ vector)
FFFEH - FFFFH	: Vector vyrazu BREAK (viz.BRK)

R 6520 (PIA)

Peripheral Interface Adapter

Zakladni vlastnosti :

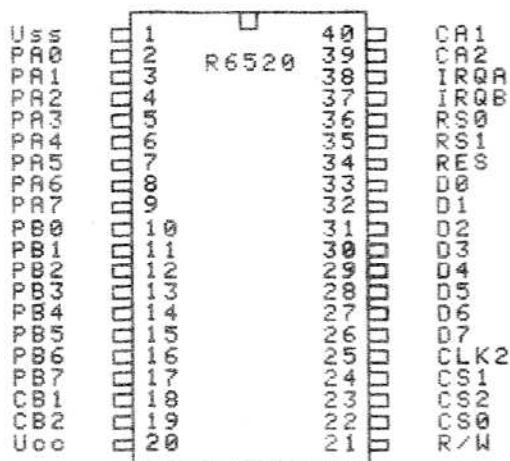
- * Prima nahrada IO MC6820
- * Jednoduché 5V napajeni
- * Dva 8-bitové Bi-directional I/O porty se samostatným primým řízením dat
- * Port A kompatibilní s technologií CMOS
- * Automatické řízení přenosu dat
- * Možnost programovatelného prerušení
- * Automatická inicializace po zapnutí
- * Verze s hodinovým kmitočtem 1 a 2Mhz

POPIS R6520 :

R 6520 je speciální IO patří do řady R65xx, který byl navržen k řízení výměny dat mezi mikropočítačovým systémem a periferiemi. Řízení periferií se uskutečňuje přes dva 8-bitové I/O porty. Každá I/O linka může být programována jako vstupní, nebo výstupní. S každým I/O portem souvisí ještě dva řídicí vodiče, které se uplatňují při přenosu dat jako řízení z, nebo na periferií.

PIN CONFIGURATION

R6520 (PIA)



POPIS SIGNALU :

Reset (RES)

Tento signal je pouzivan pro nastaveni PIA. Signal s urovni L na vstupu RES zpusobi vynulovani vseh vnitrnich registru.

Hodiny (CLK)

Signal CLK je systemovy hodinovy signal, který je vyuzivan pro synchronizaci prenosu dat mezi 65xx a 6520.

Cteni/Zapis (R/W)

Signal R/W je generovan procesorem a je vyuzit pro rizeni prenosu dat mezi procesorem a PIA. Urovni H na vstupu R/W povoluje procesor cteni dat z PIA, urovni L je povolen zapis dat do PIA.

Interrupt Request (IRQA, IRQB)

IRQA, IRQB jsou prerusovaci signaly generovane PIA pro porty A a B. Tyto signaly jsou aktivni v urovni L a umoznuji prerusovacim zpusobem informovat procesor o stavu periferie. Tyto signaly se s vyhodou vyuzivaji pri pripojeni nekolika PIA na signal IRQ procesoru.

Data Bus (D0-D7)

Techto osm linek je vyuzivano pro prenos datove informace mezi PIA a procesorem. Tyto signaly jsou normalne ve vysoke impedanci, krome stavu cteni.

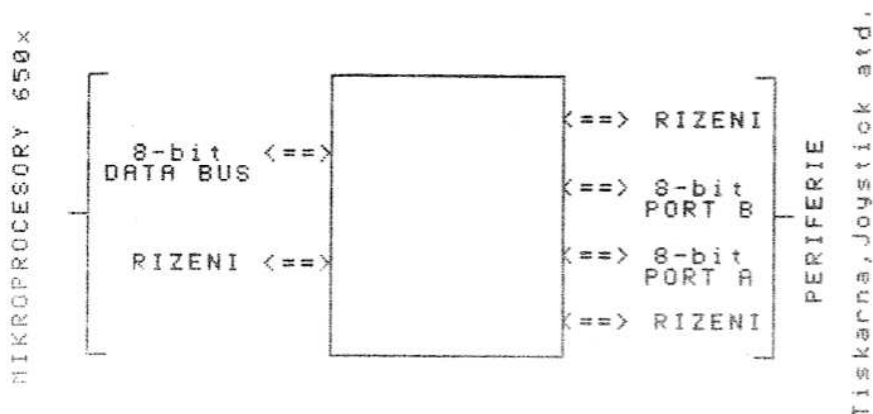
Chip select (CS0, CS1, CS2)

Tyto signaly slouzi pro vyber obvodu. PIA je vybrán, když CS0 a CS1 jsou na urovni H a CS2 na L. Tyto tri signaly jsou normalne pripojeny na adresovou sbernici primo, nebo pres dekoder.

Registr select (RS0, RS1)

Signaly RS0 a RS1 slouzi pro vyber vnitrnich registru PIA.

BLOKOVY DIAGRAM



VNITRNI ARCHITEKTURA

6520 je vnitřně organizován do dvou nezávislých sekcí, nazývaných jako 'sekce A' a 'sekce B'. Každá sekce se skládá z řídicího registru (Control Register - CRA, CRB), datového příkazového registru (Data Direction Register - DDRA, DDRB), výstupního registru (Output Register - ORA, ORB), řízení příznaku přerušeni a vyrovnávacích pamětí (buffer) potřebných pro řízení vnitřní sběrnice.

Datový vstupní registr (DIR) :

Když procesor zapisuje data do 6520, data která se objevují během druhé fáze hodinových pulsu na datové sběrnici jsou zachytávána do datového vstupního registru a sestupnou hranou hodinového pulsu jsou uložena do jednoho ze šesti vnitřních registrů. Toto zaručí, že data na výstupních linkách portu budou mít hladký přechod z úrovně L do H a opačně a že napětí bude konstantní až do doby další změny.

Řídicí registry (CRA a CRB) :

Tabulka zobrazuje určení bitů a funkcí řídicích registrů. Řídicí registry povolují mikroprocesoru řízení operací řízeného přerušeni (CA1, CA2, CB1, CB2) a řízení výstupu periférií (CA2, CB2). Bit 2 v každém registru řídí adresování primárních datových registrů (DDRA, DDRB) a výstupních registrů (ORA, ORB). Další dva bity (bit 6 a 7) obstarávají v každém řídicím registru informaci o stavu přerušovacího vstupního vodiče (CA1, CA2, CB1, CB2). Tyto přerušovací příznakové bity (IRQA1, IRQB1) jsou normálně dotazovány procesorem během obsluhy přerušeni a určují zdroj aktivního přerušeni od periférie. Jsou přímo spojeny s přerušovacími vodiči, které řídí přerušeni mikroprocesoru (IRQ, NMI).

Funkce řídicích registrů:

	I	7	I	6	I	5	4	3	I	2	I	1	0	I
CRA	I	IRQA1	I	IRQA2	I	řízení CA2	I	DDRA	I	řízení CA1				
	I	7	I	6	I	5	4	3	I	2	I	1	0	I
CRB	I	IRQB1	I	IRQB2	I	řízení CB2	I	DDRB	I	řízení CB1				

Datové příkazové registry (DDRA, DDRB) :

Datové příkazové registry povolují procesoru programovat každý vodič osmibitového portu jako vstupní, nebo výstupní. Každý bit v DDRA řídí odpovídající vodič portu A a každý bit v DDRB řídí odpovídající vodič portu B. '0' na patřičné bitové pozici programuje odpovídající vodič portu jako vstupní, '1' jako výstupní.

Výstupní registry (ORA, ORB) :

Výstupní registry ukládají výstupní data, která se objeví na I/O portu. Zapisují '0' do odpovídajícího bitu, jestliže vodič je programován jako výstupní a úroveň je (<0.4U). Funkce obou registrů je totožná.

Rizeni preruseni :

Ctyri vodice pro rizeni preruseni od periferie (CA1,CA2,CB1,CB2), jsou zpracovavany prerusovací logikou (A,B). Tato logika prevadi obsah odpovídajícího ridicího registru, provadi aktivni prechod na vystupy preruseni a vykonava operace pro osetreni ctyr prerusovacich vodicu.

Interface buffer(A,B) a buffer datove sbernice (DBB) :

Tyto vyrovnavaci pameti slouzi k uchovani informaci pri rizeni pomalych periferii (tiskarny atd.).

FUNKCNI POPIS

Bit 2 (DDR) v kazdem ridicim registru (CRA a CRB) ridi pristup do datoveho ridicího registru nebo na periferni interface. Jestliže bit 2 je '1' je vybrán vystupni registr (ORA,ORB), jestliže bit 2 je '0' je vybrán (DDRA,DDRB).Ridici bit datoveho ridicího registru spolecne s vodici pro vyber registru (RS0,RS1) provadi vyber ruznych vnitřnich registru, jak je znazorneno v tabulce.

vodic vyberu registru		ridici bit DDRA,DDRB		vyber registru	
RS1	RS0	CRA-2	CRB-2		
0	0	1	-	perif.interface A	
0	0	0	-	DDRA	
0	1	-	-	ridici registr A	
1	0	-	1	perif.interface B	
1	0	-	0	DDRB	
1	1	-	-	ridici registr B	

Pri zapisu dat do DDRA,ORA,DDRB, nebo ORB registru musi byt ve ve vlastnim ridicim registru bit 2 nastaven jako prvni.

Ustupy vyberu registru (RS0,RS1) :

Tyto dva vstupy vyberu registru jsou vyuzivany k vyberu ruznych registru uvnitř 6520. Ustupy jsou pouzivany v kombinaci s vnitřnimi ridicimi registry k vyberu jednotlivych registru, které jsou pristupne mikroprocesoru. Ustupy jsou normalne pripojeny na adresovou sbernici. RS0,RS1 pracuji v kombinaci se vstupy chip-select a dovoluji procesoru adresovat jeden 8-bitovy registr v celem adresovem prostoru. Timto registrem muze byt vnitřni registr (CRA, ORA atd.), nebo jim muze byt I/O port.

Processor muze zapisovat primo do ridicího registru (CRA,CRB), datoveho prikazoveho registru (DDRA,DDRB) a periferního vystupního registru (ORA,ORB). Cist muze primo obsah ridicího registru a datoveho prikazoveho registru. Pristup k perifernímu vystupnímu registru s cilem cist data zpet do procesoru je odlišny a je popsán samostatne.

Ctení I/O portu A :

I/O port A se skládá z 8 vodičů, které mohou být programovány jako vstupní, nebo výstupní. Když jsou programovány jako výstupní, každý bit na výstupu odpovídá příslušnému obsahu periferního výstupního registru, když jsou programovány jako vstupní, budou mít úroveň L, nebo H podle vstupních dat. Registr ORA je bez efektu, jestliže port je programován jako vstupní. Osm vodičů I/O portu A proto obsahuje buď vstupní, nebo výstupní data, podle toho který vodič je programován jako vstupní, nebo výstupní.

Provedením čtecí operace s $RS1=0, RSO=0$ a $(CRA-2)=1$, jsou data přímo přenesena z portu do procesoru. Procesor musí být programován tak, aby byly ovládnuty pouze bity, které jsou důležité pro provedení periferních operací.

Ponevadž procesor vždy čte bity portu A, místo aktuálního registru (ORA), může dojít k rozdílu mezi daty čtenými procesorem a daty uloženými ve výstupním registru. V tomto případě bude čist procesor data z odpovídajícího pinu.

Ctení I/O portu B :

Ctení I/O portu B poskytuje kombinaci vstupních a výstupních dat, podobně jako u portu A.

Výstupy žádosti o prerušení (IRQA, IRQB) :

IRQA, IRQB jsou aktivní v úrovni L a uskutečňují prerušení mikroprocesoru přímo, nebo přes obvody priority prerušení. Výstupy žádosti IRQA a IRQB souvisí přímo s porty A a B.

Každý výstup má dva prerušovací příznakové bity, které mohou způsobit nastavení výstupu na L. Tyto bity jsou bity 6 a 7 ve dvou řídících registrech. Tyto bity uskutečňují spojení mezi prerušovacím signálem od periferie a vstupním prerušením procesoru. Každý příznak má odpovídající bit zakázání prerušení, který umožňuje procesoru povolit, nebo zakázat prerušení pro každý ze čtyř prerušovacích bitů (CA1, CA2, CB1, CB2). Čtyři prerušovací příznaky jsou nastaveny aktivním přechodem signálu na vstupy (CA1, CA2, CB1, CB2).

Rizení IRQA :

Bit 7 řídícího registru je vždy nastaven příchodem kladného signálu na prerušovací vstup CA1. Prerušení od tohoto příznaku může být zakázáno nastavením bitu 0 v řídícím registru (CRA) na log 0. Podobně bit 6 řídícího registru A může být nastaven příchodem kladného signálu na prerušovací vstup CA2. Prerušení od tohoto příznaku může být zakázáno nastavením bitu 3 v řídícím registru na 0.

Oba bity 6 a 7 jsou resetovány čtecí operací procesoru.

Rizení IRQB :

Rizení IRQB je stejné jako u IRQA. Dochází ke stejnému nastavování v řídícím registru jako u IRQA.

SHRNUTÍ :

IRQA na úrovni L jestliže $CRA-7=1$ a $CRA-0=1$ nebo
když $CRA-6=1$ a $CRA-3=1$

IRQB na úrovni L jestliže $CRB-7=1$ a $CRB-0=1$ nebo
když $CRB-6=1$ a $CRB-3=1$

POZN: Kladným přechodem se rozumí změna úrovně z L na H, zaporným změna z H na L.

Porty A a B jsou slucitelne s TTL logikou a mohou byt zatezovany jednim standartnim hradlem TTL (1.6 mA, 30 pF). Port B neni primo slucitelny s logikou CMOS!

TABULKY NASTAVENI :

Rizeni CA1,CB1

CRA (CRB)		prechod		prerusovaci vystup	
bit1	bit0	vstup.sig.		IRQA (IRQB)	
0	0	zaporny		ZAKAZANO - zustava H	
0	1	zaporny		POVOLENO - (*)	
1	0	kladny		ZAKAZANO - zustava H	
1	1	kladny		POVOLENA - (*)	

(*) - na urovni L jestlize bit 7 v CRA (CRB) je nastaven prechodem signalu na CA1 (CB1).

POZN. Bit 7 CRA (CRB) bude nastaven na log 1 prechodem signalu na CA1 (CB1). Toto je nezavisle na stavu bitu 0 v CRA (CRB).

CA2,CB2 Ustupni mody

CRA (CRB)			prechod		prerusovaci vyst.	
bit5	bit4	bit3	vstup.sig.		IRQA (IRQB)	
0	0	0	zaporny		ZAKAZANO -stav H	
0	0	1	zaporny		POVOLENO - (+)	
0	1	0	kladny		ZAKAZANO -stav H	
0	1	1	kladny		POVOLENO - (+)	

(+) - na urovni L kdyz bit 6 v CRA (CRB) je nastaven prechodem signalu na CA1 (CB1).

POZN. Bit 6 CRA (CRB) bude nastaven na log 1 prechodem signalu na CA2 (CB2). Toto je nezavisle na stavu bitu 3 v CRA (CRB).

CB2 Ustupni mody

CRA			Mod		Popis	
bit5	bit4	bit3				
1	0	0	'Handshake'		1.	
			na cteni			
1	0	1	Pulsni vystup		2.	
1	1	0	'Rucni' vystup		CA2 nastaveno na L	
1	1	1	'Rucni' vystup		CA2 nastaveno na H	

- CA2 je nastaven na H aktivnim prechodem vstupniho prerusovaciho signalu CA2 a nastaven na L ctecimi operacimi procesoru. Toto umoznuje kladne rizeni dat prochazejicich z procesoru na periferie.
- CA2 je nastaven na L po dobu jednoho cyklu hodin po cteni dat. tento puls muze byt vyuzit pro rizeni periferie.

CA2 Ustupni mody

I	CRB			I	Mod	I	Popis	I
	bit5	bit4	bit3					
I	1	0	0	I	'Handshake'	I	1.	I
I	I	I	I	I	na zapis	I		I
I	1	0	1	I	Pulsni vystup	I	2.	I
I	1	1	0	I	'Rucni' vystup	I	CB2 nastaveno na L	I
I	1	1	1	I	'Rucni' vystup	I	CB2 nastaveno na H	I

1. CB2 je nastaven na L zapisovaci operaci procesoru a je nastaven na H aktivnim prechodem perusovaciho signalu CB1. Toto umoznuje kladne rizeni dat prochazejicich z mikroprocesoru na periferie.
2. CB2 je nastaven na L po dobu jednoho cyklu hodin po zapisu dat. Tento puls muze byt vyuzit pro rizeni periferie.

Poznamka redakce: Tato prirucka neprosla redakcni ani jazykovou upravou.

Publikované zo súhlasom - vid' Prohlášení představitelů AK Praha.

Igi/2019