O B S A H

CIA – Complex Inteface Adapter 6526

1.0 Pár slov úvodem

CIA 6526, patřící do rodiny obvodů 65xx slouží jako programovatelný vstupně/výstupní obvod. Jeho srovnání např. s obvodem 8255 nebo jinými U/V obvody řady 65XX, jako je např. 6522 neobstojí, CIA je mnohem univerzálnější, je opravdu komplexní.
To bylo zřejmě důvodem, proč konstruktéři C-64 sáhli při řešení U/V obvodů počítače po tomto obvodu.
Jeho výhody a bezkonkurenční postavení mezi osmibitovými U/V obvody Vám objasní následující stránky.

2.0 Technické údaje

Ke své činnosti je CIA 6526 vybaven:

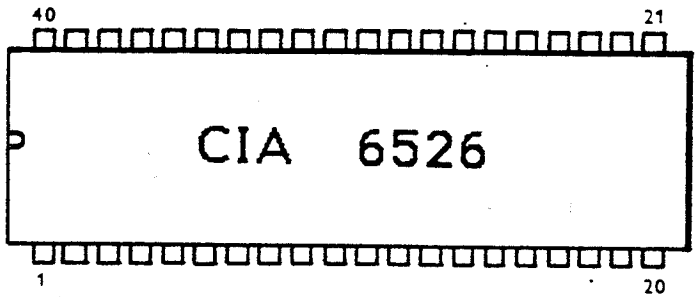
- 16 jednotlivými, nezávisle programovatelnými vstupně/výstupními linkami,
- 8 nebo 16 bitovým HANDSHAKE pro vstup nebo výstup,
- 2 nezávislými, za sebou zapojitelnými 16 bitovými intervalovými časovači (timery),
- 24 hodinovými (AM/PM) hodinami s programovatelným časem signalizace,
- 8 bitovým posuvným registrem pro sériový vstup/výstup.

Blokové schema objasňující funkci obvodu CIA 6526 najdete na obr.2

2.1 Zapojení obvodu

1 GND; zem
2–9 U/V port A; 8 bitů obousměrně.
10–17 U/V port B; 8 bitů obousměrně.
18 PC – CONTROL PORT; jeho výstup signalizuje použitelnost dat na portu B nebo na obou portech.
19 TOD – TIME OF DAY; jen vstup 50/60 Hz. Řídí hodiny reálného času.
20 +5V Ucc; napájecí napětí.
21 IRQ – INTERRUPT REQUEST; jen výstup. Bude LOW při souhlasném nastavení bitu v ICR s příchodem náležející události.
22 R/W – READ/WRITE; jen vstup. Při LOW vstup dat ze sběrnice, při HIGH výstup dat na sběrnici.
23 CS – CHIP SELECT; jen vstup. LOW = datová sběrnice platná, HIGH = datová sběrnice uzavřená (Tri-state).

24 FLAG; jen vstup, význam jako PC.
25 O2; systémový takt 2. Jen vstup. Všechny akce na datové sběrnici probíhají jen při O2 = HIGH.
26-33 DB7 - DB0 DATA BUS; datová sběrnice obousměrná. Rozhraní k procesoru.
34 RES - RESET; jen vstup. LOW = zpětné nastavení CIA na výchozí stav.
35-38 RS3 - RS0 - REGISTER SELECT; jen vstup. Slouží k výběru jednoho z 16ti registrů CIA. Platné jen při CS = LOW.
39 SP - SERIAL PORT; sériový port obousměrný. Slouží k vstupu/výstupu posuvných registrů.
40 CNT - COUNT; obousměrný. Ustup/výstup taktu posuvných registrů nebo řídící vstup pro intervalový časovač.



Obr.1 Zapojení pouzdra CIA

2. Popis registrů CIA 6526

CIA má 16 registrů, které mohou být osloveny jako paměťová místa. Commodore 64 má hned dva tyto obvody. První je dosažitelný na adresách $DC00 - $DC0F, druhý na adresách $DD00 - $DD0F. V dalším textu je uveden krátký popis těchto 16 řídících registrů.

REG. 0 - Port Register A
Přístup R/W
Obsah tohoto registru odpovídá stavu pinů PA0 - 7.
REG. 1 - Port Register B
Přístup R/W
Obsah tohoto registru odpovídá stavu pinů PB0 - 7.

REG. 2 - Směrový registr A
Přístup R/W
Tímto registrem se dá každý z 8 vývodů portu A nastavit jako vstup nebo výstup. K tomu musí být odpovídající bit směrového registru nastaven na LOW pro vstup a HIGH pro výstup.

REG. 3 - Směrový registr B
Přístup R/W
Tímto registrem se dá každý z 8 vývodů portu B nastavit jako vstup nebo výstup. K tomu musí být odpovídající bit směrového registru nastaven na LOW pro vstup a HIGH pro výstup.

REG. 4 - Timer A LOW-byte
Přístup R/W
Při čtení dává obsah tohoto registru okamžitý stav čítače A (LOW-byte).
Prostřednictvím příkazu k zápisu se může do tohoto registru zapsat LOW-byte hodnoty, od níž má čítač počítat k nule.

REG. 5 - Timer A HIGH-byte
Přístup R/W
Při čtení dává obsah tohoto registru okamžitý stav čítače A (HIGH-byte).
Prostřednictvím příkazu k zápisu se může do tohoto registru zapsat HIGH-byte hodnoty, od níž má čítač počítat k nule.

REG. 6 - Timer B LOW-byte
Přístup R/W
Při čtení dává obsah tohoto registru okamžitý stav čítače B (LOW-byte).
Prostřednictvím příkazu k zápisu se může do tohoto registru zapsat LOW-byte hodnoty, od níž má čítač počítat k nule.

REG. 7 - Timer B HIGH-byte
Přístup R/W
Při čtení dává obsah tohoto registru okamžitý stav čítače B (HIGH-byte).
Prostřednictvím příkazu k zápisu se může do tohoto registru zapsat HIGH-byte hodnoty, od níž má čítač počítat k nule.

REG. 8 TOD 10THS - Time of Day; reálný čas - desetiny sekundy
Přístup R/W
Při čtení dávají bity 0 - 3 okamžitý stav desetin sekundy hodin reálného času v BCD formátu.
Bity 4 - 7 jsou vždy LOW.
při zápisu do registru (dle předvolby v registru 15-CRB).
Je-li bit 7 CRB = LOW, zapisují se desetiny sekundy v BCD formátu. Bity 4 - 7 musí být LOW.
Je-li bit 7 CRB = HIGH, zapisuje se do signalizačního registru čas signalizace, resp. jeho desetiny sekundy ve formátu BCD. Bity 4-7 musí být LOW.

REG. 9 TOD SEC - Time of Day; reálný čas - sekundy
Přístup R/W
Při čtení tohoto registru dostanete sekundy reálného času ve formátu BCD. Přitom bity 0-3 reprezentují jednotky a bity 4-7 desítky sekund.
Při zápisu do tohoto registru můžete analogicky k registru 8 buď nastavit reálný čas nebo zapsat do signalizačního registru čas signalizace (sekundy). Počet sekund musí být vyjádřen ve formátu BCD.

REG. 10 TOD MIN - Time og Day; reálný čas - minuty
Registr 10 je organizován a obsluhován stejně jako registr 9 s tím rozdílem, že se jedná o minuty.

Reg. 11 TOD HR - Time of Day; reálný čas - hodiny
Přístup R/W
Při čtení dává tento registr aktuální stav hodin reálného času. Bity 0-3 přitom reprezentují jednotky. Hodiny počítají ve dvanáctihodinovém cyklu. Proto pro desítky hodin stačí bit 4. Bit 7 slouží jako FLAG pro americké nastavení času AM - dopoledne (bit 7 = LOW) a PM - odpoledne (bit 7 = HIGH).
Při zápisu do tohoto registru je postup analogický ostatním registrům reálného času.

REG. 12 SDR - Serial Data Register; sériový posuvný datový registr
Přístup R/W
Do tohoto registru se zapisují data, která se posunovala bit po bitu přes sériový port SP. Při čtení se stejná data vysílají ven přes pin SP.

REG. 13 ICR - Interrupt Control Register; registr pro řízení přerušení.
Přístup R/W
Při čtení (INT DATA) mají jednotlivé bity následující význam.
Bit 0 = HIGH - podtečení časovače A.
Bit 1 = HIGH - podtečení časovače B.
Bit 2 = HIGH - shoda reálného času a času ze signalizačního registru.
Bit 3 = HIGH - SDR plný při přijímání dat nebo prázdný při vysílání dat.
bit 4 = HIGH - impulz na pinu FLAG.
bity 5-6 jsou vždy 0
bit 7 signalizuje nastavení nejméně jednoho bitu v registrech INT MASK a INT DATA.
Pozor: při čtení tohoto registru se data vymažou!
Při zápisu (INT MASK):
bit 7 = HIGH - každý jednotlivý bit nastavuje odpovídající bit masky . Ostatní zůstávají nedotčeny.
bit 7 = LOW - každý jednotlivý bit maže odpovídající bit masky (nastavuje je na LOW). Ostatní bity zůstávají nedotčeny.

REG. 14 CRA - Control Register A; řídící registr A
Přístup R/W
Bit 0 = HIGH - časovač A - start.
bit 0 = LOW - časovač A - stop.
bit 1 = HIGH - podtečení časovače A bude signalizováno na pinu PB6.
bit 2 = HIGH - každé podtečení časovače A překlopí PB6 do druhého stavu.
bit 2 = LOW - každé podtečení časovače A vyrobí na PB6 HIGH impuls s délkou odpovídající jednomu systémovému taktu.
bit 3 = HIGH - časovač A počítá jen jednou od výchozí hodnoty k nule a potom se zastaví.
bit 3 = LOW - časovač A počítá stále dokola od nastavené hodnoty k nule.
bit 4 = HIGH - je nezbytné zapsat do časovače A novou startovní hodnotu. Tento bit funguje jako STROBE. Při každém nezbytném zápisu musí být znovu nastaven.
bit 5 - tento bit určuje zdroj časování časovače.
bit 5 = HIGH - časovač počítá vzestupné hrany na CNT - pinu.
bit 5 = LOW - časovač počítá systémové takty.
bit 6 = HIGH - SP je výstup.
bit 6 = LOW - SP je vstup.
bit 7 = HIGH - k časování je využit kmitočet 50 Hz.
Bit 7 = LOW - k časování je využit kmitočet 60 Hz.

REG. 15 - CRB - Control Register B; řídící registr B
Přístup R/W
Bity 0-4 - tyto bity mají stejný význam jako v CRA, ale vztaženy na časovač B a PB7.
Bity 5-6 - tyto bity určují zdroj časování pro časovač B.
00 = časovač počítá systémový takt.
01 = časovač počítá vzestupné hrany na CNT.
10 = časovač počítá podtečení časovače A
11 = časovač počítá podtečení časovače A když CNT = HIGH.
Bit 7 = HIGH - nastavení signalizace (shoda času v registrech reálného času a nastaveného v signalizačních registrech).

3.0 Vstupně-výstupní porty

Porty A a B sestávají každý z 8 bitového datového registru (DR) a jednoho registru pro směr dat (DDR). Pokud je v DDR jeden bit nastaven HIGH, potom pracuje odpovídající bit v PR jako výstup. Pokud je bit v DDR nastaven na LOW, je bit v PR definován jako vstup.
Během jednoho přístupu ke čtení dat vrací PR okamžitý stav odpovídajících pinů (PA0-7 a PB0-7) a to jak při vstupu tak při výstupu dat. Proto mohou převzít PB6 a PB7 ještě výstupní funkce pro oba časovače.

Přenos dat mezi CIA a na PA/PB připojený vnější svět může být realizován ve sdíleném provozu za pomoci pinů PC a FLAG.
PC bude po dobu jednoho systémového taktu LOW, pokud dojde k přístupu za účelem čtení nebo zápisu na PRB. Tento signál může také ukazovat využitelnost dat na PB, případně přijetí dat z PB.
FLAG je vstup řízený sestupnou hranou ipulsu, který může být spojen s PC nebo jiným CIA. Sestupná hrana na vstupu FLAG nastavuje také FLAG - Interrupt Bit.

Sériový datový port SDR je synchronní 8 bitový posuvný registr. CRA bit 6 určuje, zda pracuje v módu vstup nebo výstup dat.
V módu vstup dat budou data na SP se stoupající hranou signálu na CNT přenesena do registru pro zápis. Po 8 CNT impulsech bude obsah posuvného registru přenesen do SDR a nastaví se SP-bit v ICR.
Ve výstupním módu funguje časovač A jako generátor rychlosti v Baudech. Data z SDR budou do SP přesunována s rychlostí odpovídající polovině frekvence podtečení časovače A. Teoretická nejvyšší rychlost přenosu tedy odpovídá 1/4 systémového taktu.
Přenos dat začíná, když jsou data zapsána do SDR, předem nastavený časovač A běží a je v nepřetržitém módu (CRA bit 0 = HIGH a bit 3 = LOW). Takt vydělený časovačem se objeví na CNT. Data budou přenesena z SDR do posuvného registru a potom budou se sestupnou hranou na CNT vyslána z SP ven.
Po 8 impulzech na CNT bude vyvoláno SP-Interrupt. Pokud bude před tímto jevem registr SDR naplněn novými daty, budou nyní automaticky přetažena do posuvného registru a vysláno ven. V tomto případě se neobjeví žádné přerušení.
Data z SDR budou vyslána ven s MSB jako prvním. Vstupující data musí mít stejné uspořádání (princip FIFO).

4.0 Časovač

Každý z obou intervalových časovačů sestává z 16 bitového čítače (Read Only) a jedné 16 bitové mezipaměti (Write Only).
Data, která jsou do časovače zapisována, uloží se do mezipaměti, zatím co data z něj čtená ukazují okamžitý stav čítače.
Oba časovače mohou být použity buď nezávisle na sobě nebo za sebou. Různé druhy provozu umožňují vyrábět různé časové prodlevy, různě délky impulsů a řetězce impulsů. Při využití vstupu CNT mohou časovače počítat vnější impulsy nebo měřit frekvenci.
Každý časovač má přidělený řídicí registr CRA/CRB, který dovoluje tyto funkce:
START/STOP (bit 0)
Tento bit dovoluje kdykoliv časovač spustit nebo zastavit.
PB ON/OFF (bit 1)
Pomocí tohoto bitu bude signalizováno podtečení časovače na portu B (PB6 pro časovač A a PB7 pro časovač B). Tato funkce má přednost před

směrem dat nastaveným v DDRB.
TOGGLE/PULSE (bit 2)
Pomocí tohoto bitu se určuje, jaký impuls se při podtečení časovače na PB objeví. Buď bude při každém podtečení výstup na PB6(7) překlopen do druhé úrovně, nebo bude vyslán kladný puls s dobou trvání jeden takt.
ONE-SHOT/ CONTINUOUS (bit 3)
V provozu ONE-SHOT počítá časovač od hodnoty uložené do mezipaměti k nule, nastaví bit ICR, uloží do časovače znovu hodnotu z mezipaměti a potom se zastaví.
V módu CONTINUOUS běží popsaný cyklus bez bez přerušení stále dokola.
FORCE LOAD (bit 4)
Tento bit umožňuje časovač v kteroukoliv chvíli znovu naplnit bez ohledu na to, zda běží či ne.
INPUT MODE (bit 5 CRA, bit 5-6 CRB)
tyto bity umožňují volbu taktu, kterým bude časovač řízen.
Časovač A může být zásobován buď systémovým taktem, nebo taktem generovaným na CNT.
Časovač B může být kromě toho řízen impulsy podtečení z časovače A a to buď nezávisle nebo v závislosti na CNT=HIGH.

5.0 Hodiny reálného času

Hodiny reálného času (TOD - Time of Day) jspou 24 hodinové (AM/PM) s rozlišením 1/10 sekundy.
Setávají ze 4 registrů:
desetiny (1/10) sekundy,
sekundy,
minuty,
hodiny.
Bit AM/PM je MSB bit v hodinovém registru.
Každý registr je organizován ve formátu BCD, takže přečtené hodnoty je možno be velkého přepočítávání ihned použít.
Jako takt pro hodiny slouží 50/60 Hz signál (programovatelný pomocí bitu 7 CRA) na pinu TOD.
Kromě toho je zde ještě signální registr, pomocí kterého můžeme v libovolný čas vyvolat přerušení. Tento signální registr leží na stejné adrese jako registr TOD. Proto je přístup řízen pomocí bitu 7 CRB. Signální registr slouží pouze k zápisu. Každý pokus o přístup ke čtení dává vždy stav registru TOD bez ohledu na nastavení bitu 7 CRB.
Aby byl hodinový čas správně nastaven a čten, musí se dodržovat následující posloupnost:
Když je hodinový registr naplněn, hodiny se automaticky zastaví. Teprve když následuje zápis do registru 1/10 sekundy, běží čas dále. Takto se dají hodiny odstartovat opravdu v libovolný čas.
Protože během čtení celého času může dojít k přenosu dat do již přečteného registru, je při čtení hodinového registru celý čas přenesen do

mezipaměti. Mezipaměť se uvolní teprve tehdy, když je přečten i registr 1/10 sekundy.
Pokud má být čten jen jeden registr, je nutno postupovat tak, že po přečtení požadovaného registru nebo registrů se musí nakonec přečíst ještě registr 1/10 sekundy.

5.1 Správný čas pomocí triku

Dlouhodobá přesnost hodin TI$, řízených systémem, je z hlediska potřeby systému dostatečná. Počítá s maximální chybou 1/2 hod. za den. Pro ty, kterým záleží na zcela přesném čase se nabízí v CIA zabudované systémové hodiny. Ty získávají takt ze sítě, která má obvykle dlouhodobě vysokou přesnost. (To však zatím neplatí pro CSFR).

6.0 Programy pro CIA

Abychom Vám ovládání systémových hodin ulehčili, nabízíme Vám dva krátké programy v Basicu, sloužící k nastavení a čtení hodin reálného času.

6.1 Program pro zápis času do CIA

Ještě poznámku k nastavování hodin:
hodnota pro registr 1/10 sekundy je vždy nastavována na nulu.

```
10 REM C=56328: REM základní adresa hodin v CIA 1
20 REM C=56584 pro hodiny v CIA 2
30 POKE C+7, PEEK(C+7) AND 127
35 POKE C+6, PEEK(C+6) OR 128
40 INPUT "ZAPIS CAS VE FORMATU HHMMSS"; A$
50 IF LEN(A$) <> 6 then 40
60 H=VAL(LEFT$(A$,2))
70 M=VAL(MID$(A$,3,2))
80 S=VAL(RIGHT$(A$,2))
90 IF H>23 THEN 40
100 IF H>11 THEN H=H+68
110 POKE C+3, 16*INT(H/10)+H-INT(H/10)*10
120 IF M>59 THEN 40
130 POKE C+2, 16*INT(M/10)+M-INT(M/10)*10
140 IF S>59 THEN 40
150 POKE C+1, 16*INT(S/10)+S-INT(S/10)*10
160 POKE C,0
```

6.2 Program pro čtení hodin CIA

Čtení hodinového času umožňuje následující program.

```
10 C=56328
20 PRINT "SHFT/CLR"
30 H=PEEK(C+3): M=PEEK(C+2): S=PEEK(C+1): T=PEEK(C)
40 FL = 1
50 IF H>32 then H=H-128: FL=0
60 H=INT(H/16)*10+H-INT(H/16)*16: ON FL GOTO 80
65 IF H=12 THEN 85
70 H=H+12
80 IF H=12 THEN H=0
85 M=INT(M/16)+M-INT(M/16)*16
90 S-INT(S/16)+S-INT(S/16)*16
100 T$=STR$(T)
110 H$=STR$(H): IF LEN(H$) = 2 THEN H$=" 0"+RIGHT$(H$,1)
120 M$=STR$(M): IF LEN(M$) = 2 THEN M$=" 0"+RIGHT$(M$,1)
130 S$=STR$(s): IF LEN(M$) = 2 THEN S$=" 0"+RIGHT$(S$,1)
140 PRINT"HOME"
150 PRINT RIGHT$(H$,2)":"RIGHT$(M$,2)":"RIGHT$(S$,2);
160 PRINT RIGHT$(T$,1)
170 GOTO 30
```

Po stisknutí RUN/STOP-RESTORE se musí čas znovu nastavit, protože všechny hodnoty v registrech CIA se nastaví na počáteční hodnoty. Naneštěstí se to týká i bitu pro 50/60 Hz. Hodiny se pak budou silně opožďovat.

7.0 Zvláštnosti využití CIA v C-64

Když budete využívat CIA zabudované v C-64, musíte respektovat skutečnost, že CIA v počítači mají jisté funkce, které jsou pro práci počítače nezbytné. Zvláště to platí pro využívání přerušení, na základě kterých operační systém počítače vyvolává jisté strojové rutiny. Pokud možno také neměňte masku v ICR.

CIA 1 - základní adresa $D000 (56320)

REG. 0 (PRA)

Bit 0-7 V normálním provoze slouží k výběru řadku klávesnicove matice. Dále jsou některé bity spojeny s CONTROL PORTem 1, který slouží k připojení pákového ovladače nebo ovladače PADDLE.

Bit 0-4 Ovládač 0 v pořadí:nahoru, dolů,vlevo, vpravo, tlačítko.

Bit 6-7 Výběr PADDLE A/B. Pouze jeden z bitů může být HIGH.

REG. 1 (PRB)
Bit 0-7 V normálním provozu probíhá zde výběr sloupce klávesnicové matice, pokud byla stisknuta klávesa.
Bit 0-4 Stejná funkce jako u REG 0, ale pro CONTROL PORT 2 (ovládač 1).

REG. 13 (ICR)
Bit 4 - vstup dat z kazetového portu.

CIA 2 - základní adresa $DD00 (56576)

REG. 0 (PRA)
Bit 0-1 VA 14-15 nejvyšší adresové bity video RAM.
Bit 2 TXD - jen ve spojení s modulem RS232, jinak volný.
Bit 3 ATN - výstup sériové sběrnice.
Bit 4 CLOCK - výstup sériové sběrnice.
Bit 5 DATA - výstup sériové sběrnice.
Bit 6 CLOCK - výstup sériové sběrnice.
Bit 7 DATA - výstup sériové sběrnice.

REG. 1 (PRB)
Bit 0-7 Obvykle volné. Při použití modulu RS232 získávají následující význam.
Bit 0 RXD (Recieve Data)
Bit 1 RTS (Request to Send)
Bit 2 DTR (Data Terminal Ready)
Bit 3 RI (Ring Indicator)
Bit 4 DCD (Data Carrier Detect)
Bit 6 CTS (Clear to Send)
Bit 7 DSR (Data Set Ready)

REG. 13 (ICR)
Bit 4 RXD - jen při provozu RS232, jinak volný.

8.0 Využití Joysticku

K ovládání Joysticku je nutno pamatovat na to, že využívá stejné bity jako klávesnice. Po dobu práce s Joystickem musí být proto klávesnice odpojena. Nelze proto po odpojení klávesnice používat například příkaz INPUT.
V Basicu jsou příkazy pro ovládání Joysticku následující:
POKE 56322,224 - čtení Joysticku, odpojení klávesnice.
Vlastní čtení probíhá příkazem **PEEK(56320)** u portu 2. Pokud chceme číst Joystick v portu 1, musíme změnit adresu příkazu PEEK na 56321.
Z programu pro Joystick se vrátíme do programu využívajího klávesnici příkazem **POKE 56322,255.**
Celý program pro ovládání Joysticku v Basicu vypadá například takto:

```
10      POKE 56322,224
20      J=PEEK(56320)
30      IF (J AND 1)=0 THEN PRINT"NAHORU"
40      IF (J AND 2)=0 THEN PRINT"DOLU"
50      IF (J AND 4)=0 THEN PRINT"VLEVO"
60      IF (J AND 8)=0 THEN PRINT"VPRAVO"
70      IF (J AND 16)=0 THEN PRINT"TLACITKO"
80      GOTO 20
```

Commotronic 6/91

Následující 2 kapitoly objasňují méně zkušeným programátorům využití příkazů POKE a PEEK, logických funkcí OR a AND při tvorbě bitových masek používaných při řízení UP. Objasněna je také funkce a programování USER PORTu C64 v Basicu V2 .

9.0 PEEK, POKE & CO

Krátce si připomeneme, k čemu slouží uvedené příkazy. Příkaz POKE bude první, na který se zaměříme.
Jeho syntaxe je následující:
POKE adresa, hodnota
Příkazem se mění obsah jednotlivých paměťových míst počítače.
Pojmy "adresa" a "hodnota" jsou vyhrazeny pro číselnou hodnotu nebo pro proměnnou. Adresa představuje fyzický, procesorem adresovatelný prostor paměti. Pro oslovení libovolného paměťového místa využívá procesor 6510 celkem 16 adresových vodičů. Jimi může adresovat 2↑16, t.j. 65536 paměťových míst. Každé paměťové místo má délku 8 bitů, t.j. 1 bajt a může mít 256 různých stavů. Tím je určena velikost "hodnoty". Hodnota může být jakékoliv číslo mezi 0 a 255.
Příkazem **POKE 49152,255** se do paměťového místa 49152 ($C000) zapíše hodnota 255. Předchozí obsah se přepíše a ztratí.
Aby se naopak určil obsah určitého místa v paměti, použije se příkaz PEEK.
hodnota = PEEK(adresa)
V tomto případě představuje výraz "hodnota" proměnnou, jež je k dispozici na paměťovém místě zadaném adresou.
Při každém pokusu o čtení, kdy je zadána adresa vyšší než 65535 nebo nižší než 0 zareaguje počítač chybovým hlášením. Další příkazy o nichž se chceme krátce zmínit, jsou příkazy pro logické operace **AND a OR.**
Tyto příkazy umožňují cíleně ovlivňovat jednotlivé bity uvnitř jednoho bajtu.
Malým příkladem si působení těchto příkazů objasníme.
K tomu si představme 8 relé, jimiž budeme spínat různé spotřebiče. Těchto 8 relé je dosažitelných na adrese 56576+1. Přitom řídíme každé relé jedním bitem adresové sběrnice tak, že úroveň HIGH-1 relé sepne a úroveň LOW-0 rozepne.
Relé si označíme 0 až 7. Označení odpovídá jednotlivým bitům datové sběrnice.

bity datové sběrnice	**7 6 5 4 3 2 1 0**
relé	**7 6 5 4 3 2 1 0**

Pokud chceme, aby přitáhlo relé 4, obdržíme následující masku:

bity datové sběrnice	**7 6 5 4 3 2 1 0**
nastavená úroveň	**0 0 0 1 0 0 0 0**

Tuto masku musíme nyní převést na dekadické číslo. Výsledek je 2↑4 = 16. Příkazem POKE 56577,16 necháme relé 4 přitáhnout.

Pokud by měla přitáhnout také relé 6 a 2, potom bude potřebný příkaz POKE 56577,2↑6+2↑2 nebo POKE 56577,64+4.
Nyní přejdeme k logickým operacím. V předchozím příkladě jsme zapnuli relé 6 a 2. Chceme, aby navíc přitáhlo relé 7. K tomu se hodí funkce OR, která se dá Basicem lehce interpretovat.
Tedy:

bity datové sběrnice	7 6 5 4 3 2 1 0	
původní stav relé	0 1 0 0 0 1 0 0	= 68
+ relé	1 1 0 0 0 0 0 0	= 128
výsledek	1 1 0 0 0 1 0 0	= 196

Pokud se operací OR nastaví v jiném operandu bit, potom se nastaví tento bit i ve výsledku.
Pozorný čtenář by nyní mohl namítnout, že výsledku lze dosáhnout pouhým sečtením. V tomto případě je to pravda, důležité je však to, že pokud ještě jednou využijeme operaci OR s hodnotou 128, výsledek se nezmění.Pokud bychom však přičetli 128, dostali bychom výsledek 324. Tato hodnota by se příkazem POKE nedala zapsat, vedla by k chybovému hlášení (ILLEGAL QUANTITY ERROR). U hodnot nižších jak 128 by součet vedl ke zcela jinému výsledku a přitáhla by zcela jiná relé, než jsme chtěli.
V Basicu vypadá příkaz s logickou operací OR následovně:
POKE 56577,PEEK(56577)OR 128
Příkazem PEEK jsme si zapsali aktuální stav relé, potom jsme příkazem OR nastavili bit 7 a příkazem POKE zapsali novou hodnotu. To nám umožňuje zapínat nezávisle všechna relé.
K tomu, aby se připojené přístroje daly zase nezávisle vypnout, se však tento příkaz nehodí. Dokáže relé pouze zapínat.
K vypínání relé je vhodný jiný logický příkaz - AND.
Relé 2,6 a 7 z předchozího příkladu jsou přitažena. Chceme vypnout relé 2 bez ovlivňování ostatních.
Tedy:

bit datové sběrnice	7 6 5 4 3 2 1 0	
sepnutá relé	1 1 0 0 0 1 0 0	= 196
AND	1 1 1 1 1 0 1 1	= 251
výsledek	1 1 0 0 0 0 0 0	= 192

Na všechna místa, která jsme nechtěli ovlivnit, jsme zapsali 1, jen bit 2 má hodnotu 0
V Basicu vypadá programový řádek pro tuto operaci na USER PORTu následovně:
POKE 56 577, PEEK(56577) AND 251
Také zde jsme příkazem PEEK přečetli aktuální stav relé, potom jsme příkazem AND smazali bit 2 a výsledek zapsali zpět do paměti.
Zbývá ještě vysvětlit, proč bity na místech 5 až 3, 1 a 0 v druhé hodnotě, zvané taky maska, jsou 1. Není vždy přesně známo, které bity

v bajtu jsou nastaveny na 1. Všechny bity, které mají zůstat zachovány mají v odpovídající bitu v masce hodnotu 1.

Následující 2 kapitoly objasňují méně zkušeným programátorům funkci a programování USER PORTu C64 a dále na příkazech pro ovládání USER PORTu v Basicu V2 využití logických funkcí OR a AND při tvorbě bitových masek používaných při řízení UP.

10.0 USER PORT

Důležité styková rozhraní počítače C-64, které je plánováno pro rozšíření funkcí počítače uživatelem je USER PORT. Tento vývod se nachází na počítači vzadu vlevo. Při bližším pohledu se nejedná o nic jiného, než část základní desky počítače, na které je výřezem v pouzdru počítače přístupno 24 kontaktů.
Připojením různých přístrojů na USER PORT se mění C-64 na centrální jednotku širokého použití. Jednoduchým způsobem může měřit, regulovat, řídit, vysílat a přijímat data. Přes USER PORT komunikuje C-64 se svým okolím.

Za svou mnohostrannost děkuje USER PORT z největší části inteligentnímu integrovanému obvodu CIA 6526. Název CIA - "Complex Interface Adapter", což si můžeme volně přeložit jako všestranný styková obvod, plně vystihuje charakter obvodu.
C-64 má zabudovány dva tyto obvody.
Jeden z obvodů - CIA1 je oslovitelný v adresovém prostoru od $DC00. Slouží ke komunikaci počítače s klávesnicí, CONTROL PORTy a k výrobě systémového přerušení přes vývod procesoru IRQ. Druhý a pro nás zajímavější obvod CIA2 je dosažitelný od adresy $DD00 (56576) a má nejrůznější funkce. Odpovídá za výměnu dat přes sériové rozhraní, řídí polohu video RAM v paměti a obsluhuje USER PORT. Protože USER PORT může být využit jako rozhraní RS232, obdržely jednotlivé vývody zvláštní funkce.

10.1 I/O porty

Zabývejme se blíže obvodem CIA 6526. Jeho různé funkce je možno řídit 16 adresovými registry. Principiálně se dají volat jako místa v paměti. Chceme-li tedy, můžeme je buď číst pomocí příkazu **PEEK(adresa)**, nebo do nich zapisovat příkazem **POKE adresa, hodnota.** Port představuje v přeneseném slova smyslu spojení nebo bránu. Jak je vidět z obsazení konektoru USER PORTu, 8 bitů portu CIA2 je vyvedeno k volnému použití. Port PA je zcela vyčleněn pro sériový přenos dat. Rozhodující pro stav 8 vývodů portu jsou registry 1 a 3. Každý vývod může být programován jako vstup anebo výstup. Je tedy možné poslat na jednotlivé vývody - piny logické úrovně nebo naopak vně přivedená napětí logických úrovní mohou být čtena počítačem. Inteligentní obvod CIA reaguje přitom jako TTL hradla.

10.2 Směrový registr

Přiřazení vstupů a výstupů jednotlivým bitům se děje pomocí registru směru toku dat. Pro USER PORT je to registr číslo 3. V něm je každému bitu přiřazen k ovládání určitý bit portu B. Bit 0 řídí PB0, bit 1 řídí PB1, atd., až bit 7 řídí PB7. Je-li tento bit HIGH (1), chová se odpovídající pin jako výstup, LOW (0) úroveň bitu odpovídá vstupu.
Adresa báze CIA2 je dekadicky 56576. K tomu, abychom oslovili řídící směrový registr, přičteme jeho číslo k adrese báze - základní adresy. Příkaz, který komplexně nastaví port B jako výstupy bude vypadat následovně:

POKE 56576+3,255

Číslo 255 je nejvyšší číslo, zobrazitelné 8 bity (1111 1111 bin. je 255 dekadicky).

10.3 Datový registr

Zcela analogicky jsou jednotlivým pinům - vývodům přiřazeny bity datového registru. Datový registr má číslo 1. Dostaneme se do něj přičtením 1 k adrese báze CIA2. Každému bitu nastavenému v tomto registru na úroveň HIGH, logická jednotka, odpovídá na výstupu napětí blízké +5V. Každému bitu nastavenému na úroveň LOW, logická nula, odpovídá na výstupu napětí blízké 0 Volt. Chceme-li například na vývodu PB0 nastavit úroveň HIGH, vypadá příkaz následovně:

POKE 56576+1,1

Samozřejmě je nutno nejdřív definovat všechny piny PB jako výstupy. Pokud se vymazáním všech bitů směrového registru č.3 příkazem

POKE 56576+3,0

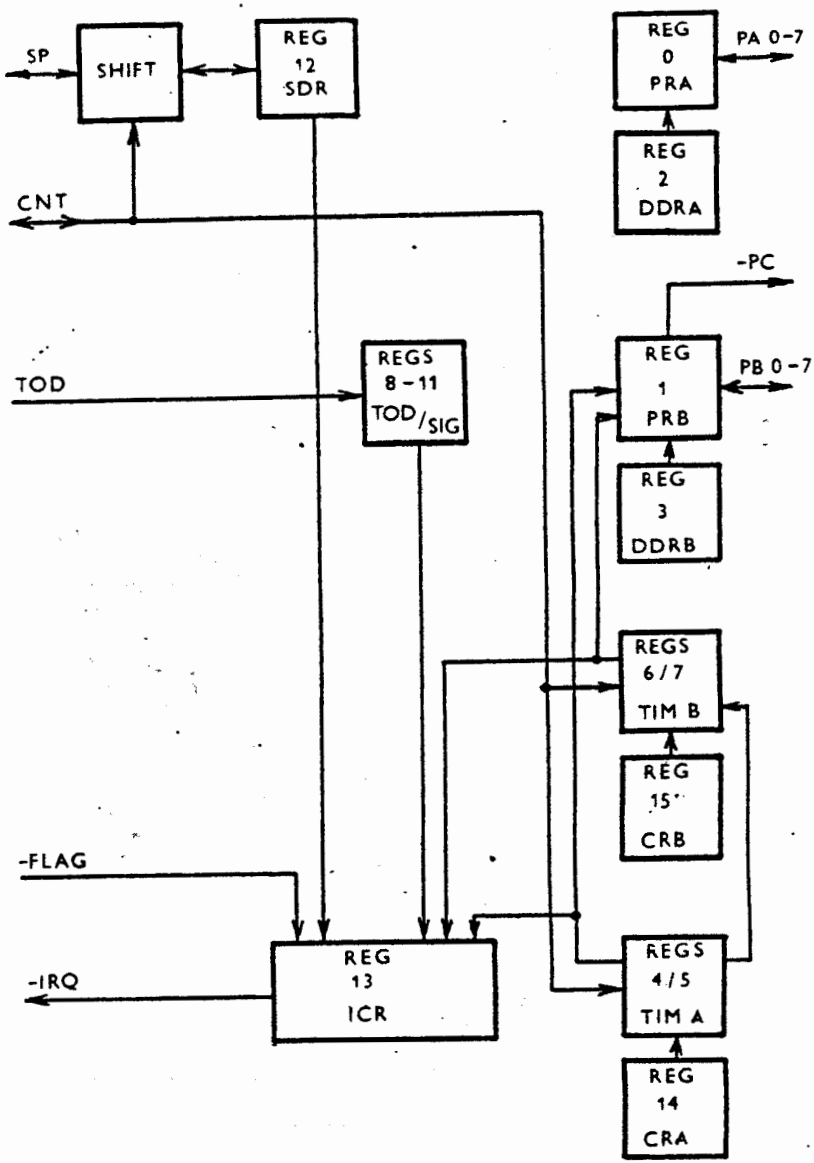
nastaví všechny piny jako vstupy, je možno zjistit jejich stav - logickou úroveň na nich pomocí příkazu

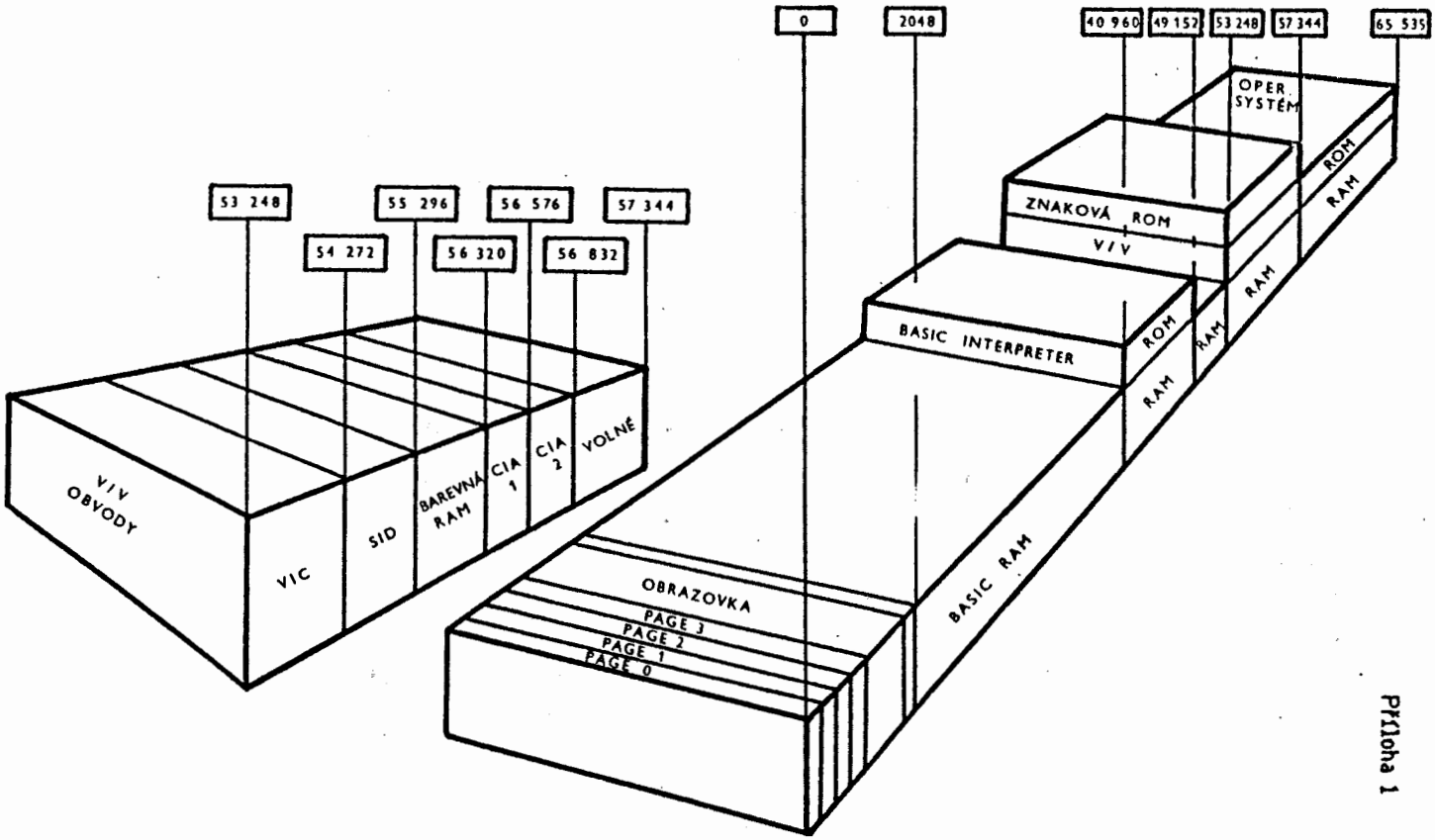
PEEK(56576+1)

Hodnota (stav) se čte v datovém registru.
Minimální program, který by kontroloval stav PB na USER PORTu, by v Basicu vypadal následovně:

```
10 POKE 56576+3,0
20 PRINT PEEK(56576+1)
```

Pokud tento program odstartujeme, aniž by na UP bylo cokoliv připojeno, obdržíme jako odpověď 255. Z toho vyplývá, že všechny piny PB jsou ve stavu HIGH.


SP
SHIFT
REG
12
SDR
CNT
REG
0
PRA
PA 0-7
REG
2
DDRA
-PC
TOD
REGS
8-11
TOD/SIG
REG
1
PRB
PB 0-7
REG
3
DDRB
REGS
6/7
TIM B
REG
15
CRB
-FLAG
-IRQ
REG
13
ICR
REGS
4/5
TIM A
REG
14
CRA



Mapa paměti C - 64

Příloha 1