

**COMMODORE**

**C 128**

**Część 1**

**WPROWADZENIE**

## S P I S      T R E S C I

1. - Wprowadzenie do Commodora .....	1
2. - Urządzenia peryferyjne .....	13
/sprzęt wizyjny, monitory, stacje dysków, drukarki i inne/	
3. - System C 64 .....	94
4. - System C 128 .....	41
5. - System CP/M .....	125
6. - Grafika w C 128 .....	180
7. - Dźwięk i muzyka w C 128 .....	244

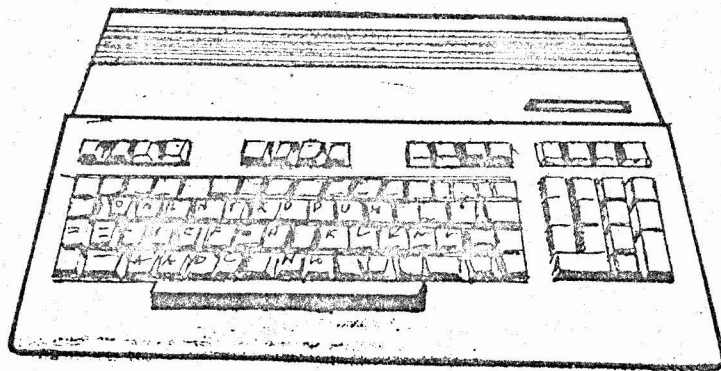


## 1. WPROWADZENIE DO COMMODORE'A 128

W rozdziale tym dowiesz się:

- o czym jest ta książka
- o cechach charakterystycznych Commodore'a 128
- o trzech trybach pracy Commodore'a 128
- dlaczego warto jest nabyć Commodore'a 128

Komputer osobisty Commodore 128 /przedstawiony na fot. poniżej/



Rys.1 - Commodore 128

jest jednym z najbardziej wartościowych komputerów, jakie ukazały się na amerykańskim rynku komputerowym do chwili obecnej /1986 r./. Łączy on niesamowitą wręcz różnorodność cech, a wszystko to w cenie nie przekraczającej 300 \$. Niektóre z tych cech to "jaskółki" w przemyśle komputerowym. Oczywiście, Commodore 128 jest pierwszym ulepszeniem wszechobecnego komputera Commodore 64, którego sprzedano już ponad 3 miliony. Commodore 128 jest pierwszym tanim komputerem osobistym, który oferuje pełne 1/8 megabajta /128 K/ pamięci. Jest to także pierwszy komputer, który łączy dużą rozdzielczość grafiki barwnej Commodore'a 64 z poważnym systemem operacyjnym dla użytkowników profesjonalnych /CP/M/. Commodore 128 jest pierwszym komputerem posiadającym dwa całkiem oddzielne systemy mikroprocesorowe i układy scalone odpowiedzialne za generowanie obrazu, co nadaje mu dwie, zasadniczo odmienne osobowości tj. komputera domowego i małego komputera biurowego. Jest też Commodore 128 pierwszym tanim komputerem posiadającym "inteligentną" stację dysków, zdolną do odczytywania dziesiątek formatów zapisu. Jest to w końcu - co najważniejsze - pierwszy komputer łączący w jednej obudowie to, co składa się na trzy komputery: system C 64, system C 128 i system CP/M.

### CO MOŻNA ZNALEŢĆ W TEJ KSIĄŻCE

Książka ta opisuje cechy Commodore 128, jego możliwości i sposoby ich wykorzystania oraz sugeruje do jakich celów mogą one być przydatne.

W pierwszym rozdziale przedstawiamy ogólny, skrótowy opis Commodore'a 128, wytłumaczymy co doprowadziło do jego powstania i zaprezentujemy jego podstawowe cechy. W dalszych rozdziałach zagłębimy się w szczegóły tych cech i omówimy ich wykorzystanie. Omówimy różne rodzaje urządzeń zewnętrznych /peryferyjnych/, które mogą okazać się Wam potrzebne w obsłudze Commodore'a 128, a następnie dość szczegółowo poznamy trzy dostępne tryby pracy tj. C 64, C 128 i CP/M. Zakończymy

rozdziałami opisującymi dźwiękowe i graficzne możliwości C 128 i sposoby korzystania z nich w Waszych programach.

Pisząc tę książkę nie mieliśmy na celu tworzenia pełnego kursu o komputerach - kursu od podstaw. Nie chcieliśmy też uczyć Was podstaw BASIC'u czy też programowania. Uwagę swoją zogniskowaliśmy na tych cechach Commodore'a 128, które powodują, iż jest on ciekawym i jedynym w swoim rodzaju komputerem.

Od czasu do czasu w książce tej podajemy nazwy producentów sprzętu peryferyjnego oraz oprogramowania, a także organizacji, które mają swój udział w przemyśle związanym z Commodore'em. Adresy tych producentów i organizacji znajdują się na końcu książki.

Jeżeli kupiłeś/aś Commodore'a 128, lub jeśli nosisz się z zamiarem dokonania takiego zakupu, książka ta powinna przynieść odpowiedzi na większość Twoich pytań dotyczących obsługi tego komputera; dowiesz się z niej, czy potrafi on wykonać to, do czego go potrzebujesz.

#### SŁOWO O NAZEWNICTWIE

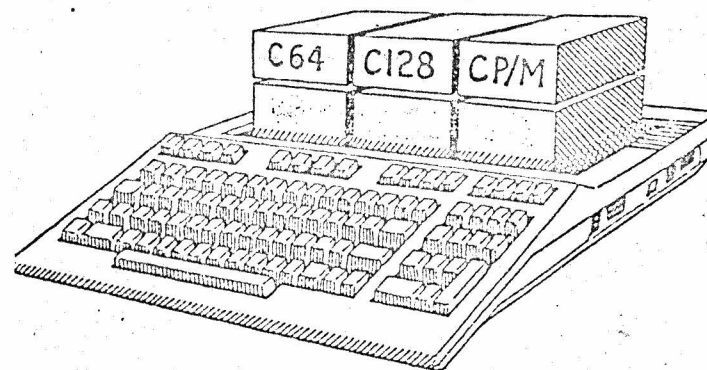
W komputerze, który zawiera w sobie trzy komputery, łatwo o pomyłkę, o który komputer w danej chwili chodzi: o główny komputer, czy o te poszczególne - wewnętrzne. Commodore odnosi się do komputerów wewnętrznych jako do systemów i tak o nich będziemy mówić w tej książce. Mówiąc o komputerze głównym będziemy używać nazwy Commodore 128 lub w skrócie - C 128. Trzy systemy budowane w C 128 to: system C 128, system C 64 i system CP/M. Komputer starszy, na którym Commodore 128 był wzorowany, to Commodore 64, w skrócie: C 64.

#### COMMODORE 128: TRZY KOMPUTERY W JEDNYM

Cechą główną Commodore'a 128, tą, która wyróżnia go spośród prawie wszystkich komputerów kiedykolwiek wyprodukowanych, jest to, że są to właściwie 3 komputery umieszczone

w jednej obudowie. Te trzy komputery mają wspólną klawiaturę i mogą razem korzystać ze sprzętu zewnętrznego, np. z ekranu lub ze stacji dysków.

A zatem: kiedy C 128 stoi u Ciebie na biurku, wygląda jakby stanowił tylko jeden system komputerowy, a przecież pod jego elegancką obudową kryją się trzy oddzielne komputery - jak pokazano, cokolwiek dziwnie, na rysunku.



Rys.2 - Trzy komputery w jednym

Czym są te trzy komputery i dlaczego Commodore zdecydował się umieścić je w jednej obudowie?

Pierwszy komputer ukryty w C 128 to wierna kopia komputera wcześniejszego: C 64. Pojawił się on na rynku w roku 1982 i okazał się być jednym z najpopularniejszych komputerów domowych wszechczasów, o czym świadczy fakt, iż sprzedano już ponad 3 miliony egzemplarzy C 64, a popyt nadal nie maleje. C 64 był udanym modelem, ponieważ oferował m.in. doskonałą barwną grafikę oraz dźwięk, a więc był doskonały

w zastosowaniach w szkolnictwie i dla rozrywki. Użytkownikom, dzięki wbudowanemu językowi BASIC firmy Microsoft łatwo było nauczyć się programowania i - co nie mniej istotne - cena C 64 była konkurencyjna w stosunku do innych mikrokomputerów.

Commodore 64 był udany, a więc - rozrósł się wokół niego przemysł software'owy. Dziś istnieje ponad 6 tysięcy programów do C 64, a każdego dnia powstają nowe. Jak zobaczymy, ta olbrzymia ilość oprogramowania, w dużej mierze przyczyniła się do powstania C 128. C 64 nie był bez wad. Weźmy choćby szybkość z jaką stacja dysków mogła przesyłać dane. Była ona żenująco mała! Użytkownicy, którzy potrzebowali załadować jakiś bardzo długi program lub zbiór danych z dysku przyzwyczaili się, że podczas tego procesu można było spokojnie zaparzyć sobie kawę. Ponadto C 64 mógł wyświetlać tylko 40 znaków w wierszu ekranu. To było dobre dla prostych programów półprofesjonalnych i do zabaw, ale użytkownicy, którzy chcieli korzystać z oprogramowania profesjonalnego potrzebowali ekranu, który wyświetlałby 80 kolumn - co jest standardowe w przemyśle.

Commodore, wobec konkurencji innych krajowych\* komputerów, chciał wprowadzić nowy komputer, który - z zastosowaniem bardziej zaawansowanej technologii - wyeliminowałby niektóre problemy zaistniałe w C 64. Jednakże, ze względu na ogromną ilość istniejących programów dla C 64, firma nie chciała całkowicie nowej maszyny, która nie byłaby w stanie pracować na istniejących programach. Miliony ludzi, którzy zainwestowali w programy przeznaczone dla C 64, nie patrzyłoby przychylnie na komputer, który miał być - w założeniu - ulepszonym C 64, do którego nie pasowałyby poprzednie programy.

Trudno jest jednak ulepszyć komputer nie zmieniając go zarazem. Jeżeli Commodore dokonałby znacznych zmian w C 64, nowy komputer nie mógłby pracować na starych programach do C 64. Wielu producentów zmagało się już z podobnym problemem. Decydowali się w końcu na jakieś rozwiązanie, godzili zach-

---

\* dotyczy USA

wawczość z innowacją. Commodore także zdecydował się wypić to piwo, którego sobie nawarzył. W jednej maszynie połączył i stary C 64 i nowy komputer C 128, który ma być podobny do C 64, ale lepszy od niego.

Jak zaznaczyliśmy wcześniej, te dwa komputery w jednym, nazywamy systemami: system C 64, który zezwala Commodore'owi 128 na zrównanie się z Commodore'em 64 i system C 128, który stanowi całkiem nowy komputer, z licznymi ulepszeniami i usprawnieniami. Do tego dodany jest jeszcze trzeci system: CP/M, który czyni Commodore'a 128 poważnym komputerem profesjonalnym. Poniżej opiszemy pokrótce te trzy systemy, a w następnych rozdziałach przeprowadzimy szczegółową ich analizę.

Trudno jest ulepszyć komputer nie zmieniając go jednocześnie. Za chwilę zobaczysz dlaczego Commodore 128 jest równocześnie i innym i lepszym komputerem.

#### SYSTEM C 64

Jak już zaznaczyliśmy, Commodore 64 posiadał wspaniałą grafikę barwną i dźwięk; w rzeczywistości były one najbardziej rozbudowane w porównaniu z innymi komputerami domowymi wyprodukowanymi w kraju\*. Grafika barwna wykorzystwała SPRITES - wyszukane, inteligentnie pomyślane jednostki graficzne, które można było zaprogramować w odpowiedni sposób tak, aby przedstawiały obiekty takie, jak samochody wyścigowe lub statki kosmiczne. Można im było rozkazać, aby poruszały się po ekranie tak jak byśmy chcieli i czyniły to automatycznie, bez zbędnego zwalniania programu. Zdolność generowania dźwięku nadała C 64 niesłychanej żywotności jak na maszynę tak tanią. C 64 posiadał zasadniczo 3 generatory plus dużą różnorodność sposobów kształtowania dźwięku, co mogło dawać niemal dowolny efekt dźwiękowy, od naśladowania instrumentów muzycznych począwszy, na głosie ludzkim skończywszy.

---

\* dotyczy USA

Wszystkie te cechy zostały zachowane w systemie C 64 Commodore'a 128. W istocie, system C 64 Commodore'a 128 jest niemalże identyczny z Commodore'em 64. Będzie on pracował na tych samych programach. Ekran jest ten sam, obsługa klawiatury wygląda identycznie. Stacje dysków i inne urządzenia peryferyjne działają jak przedtem.

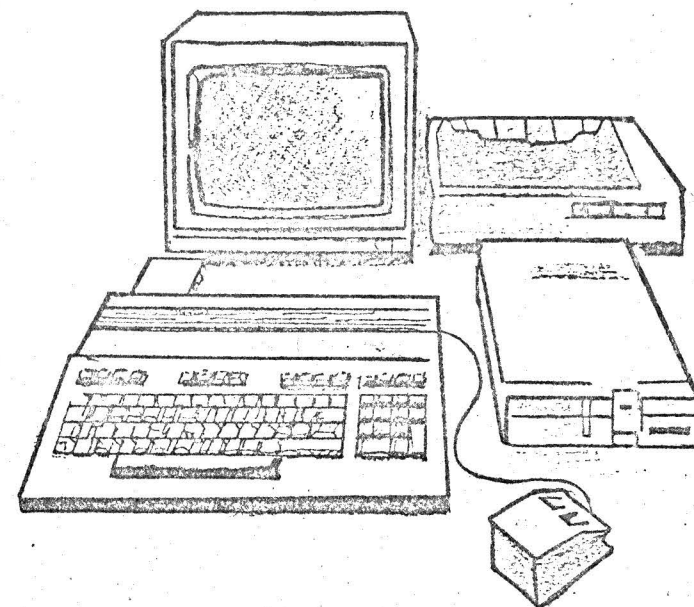
Jeżeli jesteś właścicielem/ka/ Commodore'a 64, uznasz ten system za nieoceniony, albowiem całe Twoje oprogramowanie będzie pracowało w nowej maszynie bez najmniejszych modyfikacji - tak samo, jak pracowało w C 64. Nawet jeżeli nigdy nie posiadałeś/aś/ Commodore'a 64 i tak system C 64 jeszcze Ci się przysłuży, albowiem będziesz mógł/mogła/ kupić i wykorzystać którykolwiek z tysięcy programów napisanych dla C 64. System ten będzie szerzej opisany w rozdziale 4.

#### SYSTEM C 128

System C 128 jest faktycznie nowym komputerem wewnątrz Commodore'a 128. Jakie ulepszenia są tu wprowadzone? Powiedzieliśmy już, że Commodore 64 wymagał dwóch usprawnień: przyspieszenia transmisji danych na dysk i z dysku oraz wprowadzenia 80-kolumnowego trybu pracy ekranu. C 128 eliminuje obydwie problemy: oferuje znacznie szybszy dostęp do danych na dysku, a także wybór za przyciśnięciem klawisza 80-kolumnowego lub 40-kolumnowego ekranu barwnego.

#### Szybkość transmisji danych z dysku

Czas transmisji danych z dysku jest teraz ponad pięć razy krótszy. Zostało to umożliwione dzięki całkiem nowej rodzinie napędów dyskowych, zastosowanych w C 128. Te nowe napędy noszą nazwy: Commodore 1571 i 1572. 1571 - to napęd pojedynczy, pokazany na rysunku wraz z innymi urządzeniami peryferyjnymi.



Rys.3 - System Commodore 128

Napęd 1572 - to urządzenie o podwójnym napędzie. Napędy te zostaną omówione szerzej w rozdziale poświęconym urządzeniom peryferyjnym.

### Tryb 80-kolumnowy

Commodore 128 może pracować w dwóch trybach: 40-kolumnowym lub 80-kolumnowym, wybieranych odpowiednim przyciskiem na klawiaturze. Przy 80-kolumnach można korzystać z profesjonalnego, bardziej naukowego oprogramowania w C 128, co nie było możliwe w C 64. Zauważ, że nie wszystkie monitory będą funkcjonowały w trybie 80-kolumnowym. Więcej na ten temat powiemy w następnym rozdziale.

Innowacje wprowadzone przez system C 128 nie kończą się na 80 kolumnach i na szybszym dostępie do danych na dysku. Oprócz wymienionych zmian C 128 wyposażony jest w znacznie lepszą wersję języka BASIC, posiada większą pojemność pamięci oraz monitor języka maszynowego. Są to tylko najbardziej widoczne ulepszenia.

### Ulepszony BASIC 7.0

Większość komputerów posiada wbudowany język programowania po to, by użytkownicy mogli nauczyć się programowania, aby mogli pisać własne programy, czy też by mogli poznać programy napisane przez innych. Standardowym językiem rezydentnym większości małych komputerów jest BASIC, który jest napewno najłatwiejszym do nauczania językiem programowania. Wersja BASIC'a wbudowana do Commodore'a 64, nazwana BASIC'iem 2.0 służyła na ogół bardzo dobrze, ale miała pewne wady. Np. trudno było wykorzystać wspaniałe możliwości graficzne i dźwiękowe Commodore'a 64. Trudno też było pisać programy w nowoczesnym "strukturalnym" stylu, gdyż BASIC C 64 nie posiadał pewnych rozkazów, które ułatwiałyby tego rodzaju programowanie. Było wprowadzić kilka akcesoriów, takich jak Simons BASIC lub SUPR EXPANDER, które dodawały trochę rozkazów dla ulepszenia BASIC'a, ale nie były one częścią języka standardowego.

Nowa wersja BASIC'a dostępna w systemie C 128, nazwana BASIC'iem 7.0 likwiduje dotychczasowe niedociągnięcia, a ponadto dodaje wiele nowych cech. Daje ona znacznie zwiększoną liczbę rozkazów, w tym takich, które ułatwiają nowoczesne pro-

gramowanie strukturalne, jak: DO... LOOP, BEGIN....END oraz IF THEN ... ELSE. Korzystanie z grafiki jest uproszczone dzięki dodaniu poleceń, które pozwalają Ci rysować okręgi, prostokąty, punkty i linie; wypełniać przestrzenie kolorem, dzielić ekran na część tekstową i część graficzną. SPRITES /barwne obiekty graficzne, które mogą być tak zaprogramowane, aby niezależnie poruszały się na ekranie, tak jak wyścigówki czy rakiety kosmiczne/ definiuje się prostymi rozkazami, które tworzą sprites, poruszają nimi i sprawdzają, czy nie dochodzi do kolizji między nimi. Istnieje nawet wbudowany edytor sprite'ów, który pozwala Ci projektować je na ekranie Commodore'a, a następnie zachować do wykorzystania w programie.

W C 128 są także rozbudowane rozkazy umożliwiające korzystanie z systemu dyskowego /DOS - czyli Dyskowy System Operacyjny/. Jest też cała gama nowych poleceń, ułatwiających cały proces programowania, takich jak AUTO - do automatycznego numerowania linii programu; RENUM - do przenumrowania linii programu, czy TRACE - do analizy działania programu, linia po linii.

Ogólnie rzecz biorąc, nowy BASIC jest rozkoszą dla programisty. Jest to zapewne jeden z najlepszych BASIC'ów, dostępny dla nabywcy komputera, którego cena byłaby porównywalna z ceną Commodore'a 128.

### Rozbudowana pamięć

Jak sama nazwa sugeruje, Commodore 128 ma pamięć 128 K - czyli dwa razy większą od pamięci C 64. Ze względu na różnorodne sprawy techniczne, właściwym efektem zwiększonej pamięci C 128 jest to, że programy w BASIC'u w systemie C 128 mogą być ponad dwa razy dłuższe od tych z systemem C 64, jak się przekonamy w rozdziale poświęconym systemowi C 128. Jeżeli zaś i 128 K pamięci nie wystarcza, można ją powiększyć jeszcze bardziej za pomocą dołączanych modułów pamięci, aż do 512 K.

### Ulepszona klawiatura

Osobna klawiatura numeryczna /klawisze cyfrowe zgrupowane razem tak jak w kalkulatorze/ jest przydatna każdemu, kto ma do czynienia z wieloma danymi cyfrowymi, a C 64 jej nie posiadał. Wielu użytkowników Commodore'a 64 spędzało długie godziny na wprowadzaniu programów języka maszynowego w formie długich ciągów cyfr i oni to często upominali się o pewne ulepszenia w tym zakresie. Commodore 128 posiada klawiaturę numeryczną, wbudowaną do komputera po prawej stronie klawiatury standardowej.

Nowa klawiatura posiada też /poza innymi nowymi klawiszami/ dodatkowe klawisze sterujące kursorem, które ułatwiają manipulację nim. Klawisz HELP którym przyciskając otrzymasz dodatkowe informacje wyjaśniające, w tych programach, które korzystają z tej właściwości i klawisz 40 kolumn - 80 kolumn do przełączania trybów pracy ekranu. O nowej klawiaturze powiemy więcej w rozdziale poświęconym systemowi C 128.

### Wbudowany Monitor Języka Maszynowego

Jeżeli jesteś programistą języka maszynowego lub języka assemblera, lub jeśli po prostu chcesz zbadać różnorakie aspekty techniczne Twojego komputera, docenisz tę nową cechę Commodore'a 128: wbudowany program monitora, który pozwoli Ci asemblować, disasemblować i uruchamiać programy w języku maszynowym lub w języku assemblera, a także przeglądać i modyfikować zawartość pamięci.

### SYSTEM CP/M

Tak, jakby dwa komputery w jednym nie wystarczały, Commodore dodał jeszcze trzeci, całkiem odmienny system: CP/M. CP/M to skrót od: Program Sterujący dla Mikrokomputerów. Jest to DOS /program, który kontroluje pracę dysku/, rozwinięty przez Digital Research, Inc., i dostępny niemalże od naj-

wcześniejszych dni mikrokomputerów, czyli prawie od dziesięciu lat. Dowiesz się więcej o tym, czym w ogóle są systemy operacyjne i o CP/M, w rozdziale 5.

CP/M jest systemem operacyjnym, który usiłuje ujednolicić pisanie programów i ich wykorzystywanie tak, aby program napisany dla CP/M jednego komputera, mógł też pracować w innym komputerze. CP/M był bardzo udany i popularny i dziesiątki tysięcy programów - najczęściej programów profesjonalnych - zostało dla niego napisanych. W ten sposób, czyniąc CP/M dostępnym w C 128, Commodore "podłącza się" do olbrzymich zasobów istniejących programów. Jako że większość tego oprogramowania ma charakter profesjonalny, dodanie CP/M /wraz z 80-kolumnowym ekranem i z szybszymi stacjami dysków/ zamienia Commodore'a 128 w poważną maszynę dla profesjonalistów. Rozdział 5 przytacza kilka bardziej popularnych programów CP/M. Opisuje również olbrzymią ilość bezpłatnego, publicznego oprogramowania dla CP/M.

### I NA KONIEC ...

Jeżeli chciałbyś wejść w posiadanie jednego z najżywczej-  
niejszych komputerów naszych czasów, komputera, który przyj-  
mie tysiące już istniejących programów szkolnych /edukacyj-  
nych/, czy też rozrywkowych lub innych, który może też słu-  
żyć za maszynę profesjonalną, biurową - to Commodore 128  
przedstawia nieocenioną wartość. W rozdziałach, które nastę-  
pią, powiemy Wam więcej o tym komputerze i zbadamy szczegó-  
łowo jego zdolności. Zaczniemy od omówienia urządzeń peryfe-  
ryjnych do C 128: ekrany wyświetlające, stacje dysków dru-  
karki i inne urządzenia zewnętrzne.



## 2. URZĄDZENIA PERYFERYJNE:

SPRZĘT WIZYJNY /MONITORY/, STACJE DYSKÓW, DRUKARKI  
I INNE

~~XX~~

W tym rozdziale dowiesz się:

- co to są urządzenia peryferyjne i do czego służą
- jakie urządzenia peryferyjne są najczęściej używane do Commodore 128
- jakie urządzenia peryferyjne są dostępne do Commodore 128
- jak podłączyć urządzenia peryferyjne do Commodore 128

Gdy kupujesz samochód, zazwyczaj nabywasz także wiele dodatków: np. automatyczną skrzynię biegów, wspomaganie układu kierowania, klimatyzację, barwioną szybę, magnetofon stereo, etc. Jedne z nich są niezbędne dla prowadzenia pojazdu, niektóre zaś stanowią o komforcie, a jeszcze inne, jak np. spoilery są przejawem snobizmu. Analogiczną sytuację mamy przy zakupie komputera. Istnieje wiele dodatków do głównej jednostki, jaką jest komputer, mniej lub bardziej niezbędnych do pracy z komputerem. W komputerowym świecie dodatki te nazywa się urządzeniami peryferyjnymi. Oznacza to, że są usytuowane wokół komputera, na jego "peryferiach". Urządzeniami peryferyjnymi, używanymi do pracy z Commodore 128 są: urządzenia wizyjne /monitory/, stacje dyskó, drukarki i modemy.

Jeśli chcesz widzieć informacje wyprowadzane przez komputer, będziesz potrzebował pewnego rodzaju urządzenia wizyjnego: telewizora lub monitora. Jeśli będziesz chciał przechowywać programy lub dane, w czasie gdy komputer jest wyłączony - będziesz potrzebował magnetofonu kasetowego lub stacji dyskó. Jeśli będziesz chciał drukować programy lub dane na papierze - będziesz potrzebował drukarki. Jeśli będziesz chciał komuni-

kować się za pomocą linii telefonicznej - będziesz potrzebował modemu. Są to jednak urządzenia podstawowe. Poza nimi istnieje wiele innych urządzeń peryferyjnych, tak, że trudno jest się zdecydować, które dodatki zakupić /i na które nas stać/. Podobnie jak przy zakupie samochodu, tak i tu możesz mieć nie lada kłopot z wyborem urządzeń peryferyjnych do Twojego komputera.

W tym rozdziale zamierzamy wprowadzić Cię w świat urządzeń peryferyjnych Commodore 128 objaśniając, do czego służą, przeglądając sprzęt dostępny na rynku i doradzając jego odpowiedni wybór. Następnie wyjaśnimy jak dołączyć urządzenia peryferyjne do Commodore 128. Nie jest to sprawa prosta. Istnieje tyle różnych wtyczek, wejść, portów i połączeń o różnych kształtach i rozmiarach, jeśli chodzi o Commodore 128, że staje się niezbędne objaśnienie, jakiego rodzaju urządzenia podłączyć, gdzie i po co.

Zacznijmy od omówienia tego, co już znajduje się w Commodore 128. Z opisu tego wyłoni się nam potrzeba użycia pewnych urządzeń peryferyjnych.

### CO JUŻ ISTNIEJE W COMMODORE 128

Zaproponowaliśmy różnego rodzaju urządzenia, które mogą okazać się niezbędne dla twego systemu komputerowego; co już posiadamy w komputerze; czego nie potrzebujesz kupować.

#### Klawiatura

Jak możesz zobaczyć na rys.1 w rozdziale pierwszym, Commodore 128 wygląda tak, jak by był samą klawiaturą. Klawiatura ta stanowi urządzenie, za pomocą którego użytkownik wprowadza informację do komputera. Ta ładnie wykonana obudowa plastikowa, która mieści klawiaturę, zawiera także wiele innych urządzeń.



### Układy scalone

Commodore 128 zawiera przede wszystkim cały układ "kości" półprzewodnikowych tzw. chipy /są to układy połączeń wytrawionych w małym kryształu krzemu/, które sterują działaniem komputera. Jako użytkownik prawdopodobnie nie będziesz musiał wiedzieć zbyt wiele o tych układach scalonych. Jeśli nie interesują Cię szczegóły techniczne, możesz opuścić kilka następnych ustępów tej książki.

Sercem Commodore 128 jest mikroprocesor 8502. Układ ten jest bliskim kuzynem układu 6502, którego użyto w Apple II, Atari i innych komputerach. 8502 steruje działaniem Commodore, gdy pracuje on w systemie C64 lub C128. W systemie CP/M sterowanie przejął mikroprocesor Z80A. Dodatkowo /oprócz tych mikroprocesorów/ Commodore 128 posiada specjalne układy scalone do sterowania grafiką, dźwiękiem i do komunikacji z urządzeniami peryferyjnymi.

Grafika jest pod kontrolą układu o nazwie VIC II /"Video Interface Chip"/. W trybie pracy 80-kolumnowej /który omówimy poniżej/, wykorzystany jest układ 8563, który pozwala wyświetlać dwa razy więcej znaków w linii, niż układ VIC II.

Dźwięk jest sterowany przez SID /"Sound Interface Device"/. SID jest to ten sam układ, który został użyty w Commodore 64. Dwa układy odpowiedzialne są za komunikację z urządzeniami peryferyjnymi, które nazywane są CIA's /"Complex Interface Adaptors"/.

### Pamięć

Istnieją także inne układy scalone, które stanowią pamięć Commodore 128. Pamięć podzielona jest na dwie części: pamięć stałą, która wypełniona jest informacjami już przez producenta, o nazwie ROM /Read Only Memory - pamięć tylko do odczytu/ i pamięć do zapisu, która może zawierać programy i dane

tworzone przez użytkownika o nazwie RAM /"Random Access Memory" - pamięć o swobodnym dostępie/. Commodore 128, jak wynika z jego nazwy, zawiera 128k pamięci RAM. /Jeden "K" jest równy 1024 bajtów, a więc C128 zawiera 131072 bajtów/.

Cokolwiek zawarte jest w pamięci RAM, będzie skasowane po odłączeniu komputera od zasilania. A więc nie jest to dobre miejsce na przechowywanie programów lub danych.

Więcej informacji o tym uzyskasz w tej części rozdziału, która dotyczy magnetofonu kasetowego i stacji dysków.

### Połączenia

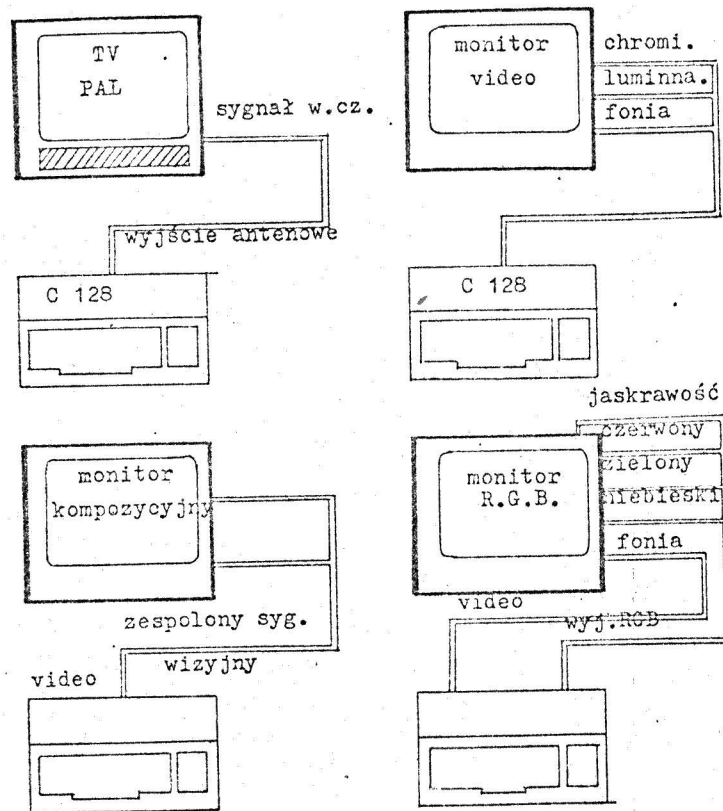
Ostatnią grupę podstawowych elementów wbudowanych w Commodore 128 stanowią różnorodne złącza /gniazda/, które umożliwiają kontakt komputera z urządzeniami peryferyjnymi. Więcej szczegółów na ten temat można znaleźć w ostatniej części tego rozdziału.

W tym rozdziale omówimy najczęściej stosowane urządzenia peryferyjne. Następnie opisana zostanie metoda połączenia ich z Commodore 128.

W następnych rozdziałach omówimy szerzej różnego rodzaju urządzenia, które są szczególnie pomocne w pracy z C64, C128 lub CP/M.

### Urządzenia wizyjne

Urządzenie wizyjne jest jedynym urządzeniem peryferyjnym, które jest absolutnie niezbędne do pracy z Commodore 128. Urządzeniem tym może być zwykły telewizor lub monitor telewizyjny. Co to jest monitor? Jest to urządzenie identyczne jak telewizor, ale nie mogące odbierać audycji telewizyjnych. W zamian za to, obraz monitora jest ostrzejszy, ponieważ sygnały z komputera nie przechodzą przez złożone obwody układów do odbioru programów telewizyjnych. Istnieją różne



Rys.4 - Możliwości podłączenia urządzeń wizyjnych do C128

rodzaje monitorów, a więc wybór odpowiedniego urządzenia nie jest zadaniem łatwym. Zależy to też od tego, do czego chcesz używać komputera, jakiej jakości obrazy chcesz uzyskać i ile pieniędzy możesz na monitor przeznaczyć. Rozpatrzmy możliwe rozwiązania.

#### Telewizor

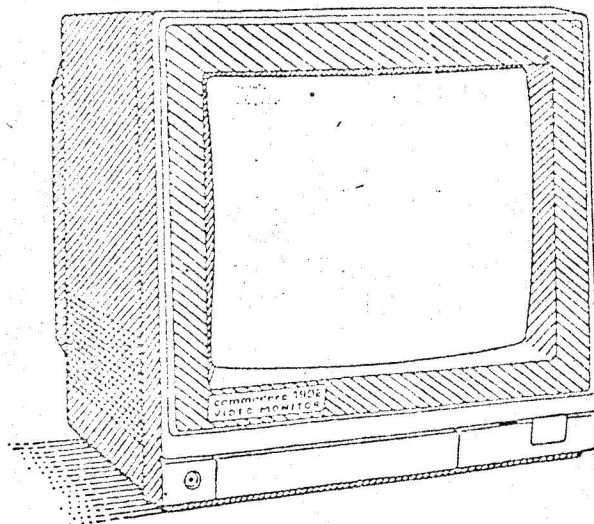
Najtańszym urządzeniem wizyjnym do Commodore 128 jest zwykły telewizor. Jest to najlepsze rozwiązanie, jeśli chcesz używać komputera do gry lub prostych programów. Do tych celów najlepszy jest telewizor kolorowy. Ponieważ prawie każdy posiada telewizor więc sprzęt ten nic nie kosztuje. Natomiast jeśli masz tylko jeden telewizor, a chcesz używać komputera w innym pokoju niż telewizyjny, może zaistnieć wtedy potrzeba zakupu drugiego odbiornika. W takiej sytuacji proponujemy zakup monitora.

W przypadku używania telewizora do pracy z komputerem, niezbędne jest urządzenie przełączające wyjście antenowe z pracy z komputerem na pracę ze zwykłą anteną. Urządzenie takie jest dostarczane przez Commodore razem z C128. Aby urządzenie działało, należy połączyć je przewodem antenowym z C128, a drugim przewodem połączyć urządzenie z telewizorem.

Podstawową wadą telewizora jest niedostateczna ostrość obrazu /ilość szczegółów obrazu jest ograniczona/. W przypadku obrazów, jakie występują w grach lub w prostych programach edukacyjnych, odbiornik telewizyjny jest urządzeniem odpowiednim. Natomiast jeśli chodzi o wyświetlanie tekstów, sprzęt taki jest nieodpowiedni, ponieważ litery i wyrazy zacierają się i czytanie ich staje się męczące.

### Monitor kolorowy

Ponieważ na ekranie telewizora tekst jest trudny do odczytania, używa się różnego rodzaju monitorów. Cechuje je ostrzejszy obraz niż w odbiorniku telewizyjnym. A więc powinienś wybrać monitor, jeśli będziesz używał komputera do



Rys.5 - Monitor kolorowy

redagowania tekstów lub do pracy z programami cyfrowymi, lub jeśli zamierzasz przeznaczyć dużo czasu na pisanie własnych programów.

Najtańszym monitorem do Commodore 128 jest tzw. monitor bezpośredni /direct monitor/, ponieważ sygnały wizyjne nie przechodzą przez obwód dekodera sygnałów programu telewizyjnego /co ma miejsce w normalnym telewizorze/. Tutaj, jeden przewód od komputera jest rozdzielony na 3 części przy wejściu do monitora: jeden odpowiada za głos, a dwa pozostałe za specjalne sygnały obrazu, zwane "luminancja" i "chrominancja". Commodore jest unikalnym mikrokomputerem, jeśli chodzi o użycie takiego monitora. Większość komputerów wykorzystuje urządzenie zwane monitorem złożonym, w którym sygnały "luminancji" i "chrominancji" przekazywane są jednym przewodem. Commodore oferuje b.dobry obraz, ale użytkownik musi nabyć specjalny monitor, produkowany przez Commodore, aby w pełni wykorzystać możliwości systemu.

Commodore produkuje monitor bezpośredni, który zyskał sobie dużą popularność wśród użytkowników tego sprzętu. Jest to 1702 color video monitor. Możesz oczywiście użyć monitora złożonego, chociaż wtedy obraz nie będzie tak ostry. Inne firmy produkują monitory złożone niezbyt drogie /poniżej \$ 200/, jednakże wyższej jakości monitor złożony kosztuje ok. \$ 350.

Chociaż monitor bezpośredni daje ostrzejszy obraz niż telewizor, może on nie być wystarczający dla wszystkich celów. W szczególności będzie wyświetlał znaki w 40 kolumnach, ale już nie w 80. Zobaczmy co to znaczy.

### 40 Kolumn kontra 80 kolumn

Commodore 64 - poprzednik Commodore 128, wyświetla tekst tylko w jednym rozmiarze. Czterdzieści znaków /liter lub cyfr/ mieści się w jednej linii na ekranie, tzn. jeden znak w każdej z 40 kolumn, a więc litery są dość duże i można je odczytać,

nawet gdy są lekko zamazane. Podobny efekt mamy przy wykorzystaniu Commodore 128 w systemie pracy C64. Jednakże w przypadku pisania listów na maszynie do pisania, w linii mieści się więcej niż 40 znaków / blisko 80 /. Także drukarka komputera nie drukuje tekstu 40-kolumnowego.

Wielu użytkowników komputerów /używających komputer do przetwarzania tekstów lub do innych celów/ chce widzieć linię tekstu na ekranie komputera w tej samej postaci, w jakiej zostanie ona wydrukowana. Z tych powodów większość komputerów, używanych profesjonalnie stosuje tryb 80-kolumnowy.

Commodore 128 pozwala Ci na wybór między trybem 40- i 80-kolumnowym. Dokonuje się tego przez naciśnięcie klawisza klawiatury. Jednakże, jeśli wybierzesz tryb 80-kolumnowy, powinieneś posiadać monitor, który ostro wyświetli tak małe znaki. Telewizor nie nadaje się do tego celu. Nawet monitor złożony, nie daje zadowalających efektów. W tym przypadku stosuje się monitor specjalny zwany RGB.

#### Monitor RGB

Monitor ten jest droższy od poprzednio omawianych, lecz daje obraz znacznie wyższej jakości. RGB znaczy Red-Green-Blue Intensity. W monitorze tym, sygnały niosące informacje o 3 kolorach /czerwonym, zielonym, niebieskim/ i intensywności świecenia, przesyłane są czterema oddzielnymi przewodami. Daje to bardzo czysty, kolorowy obraz, ponieważ sygnały te nie są mieszane w komputerze i rozdzielane w monitorze, jak to ma miejsce w przypadku monitora bezpośredniego i złożonego. Ponieważ zespół sygnałów wychodzących z C128 musi być odmienny w przypadku monitora RGB, używa się innego połączenia niż przy monitorach bezpośrednich, złożonych i telewizorach. W rzeczywistości możliwe jest stosowanie dwóch różnych połączeń jednocześnie /pokażemy to pod koniec tego rozdziału/.

Monitory RGB są produkowane przez wiele firm. Rozbieżność cen jest duża - im lepszy obraz, tym wyższa cena. Monitory najwyższej jakości kosztują ok. \$ 750, ale również dobry monitor RGB możesz już nabyć za ok. \$ 400. Rys.2 pokazuje połączenia różnych urządzeń wizyjnych.

Poważnym problemem przy użyciu C128 jest niemożność zastosowania monitora RGB w trybie 40-kolumnowym. Niestety oznacza to, że dla użycia obu trybów pracy, potrzebujesz w rzeczywistości dwóch monitorów: bezpośredniego /lub złożonego/ i RGB. Na szczęście firma Commodore rozwiązała ten problem.

W tym samym czasie, gdy ukazał się Commodore 128, firma zaprezentowała nowy monitor - 1902, pokazany na rys.5. Monitor ten /którego koszt wynosi mniej niż \$ 300/ jest kombinacją kilku monitorów. Możliwe jest użycie go jako monitora złożonego lub jako RGB, przełączając tylko jeden przełącznik na płycie frontowej. Ponieważ przyjmuje on także sygnały złożone /mieszane/, może być użyty jako zwykły telewizor.

#### Monitory monochromatyczne

Jeśli zamierzasz wykorzystać swój komputer do przetwarzania tekstów lub w celach profesjonalnych, najbardziej przydatny będzie monitor monochromatyczny. Tak jak w telewizorze czarno-białym, na monitorze tym nie uzyskuje się barwnego obrazu - jest on zazwyczaj czarny lub czarno-zielony. Monitor ten można nabyć za mniej niż \$ 200, a uzyskuje się na nim ostrzejsze obrazy, niż na monitorze RGB. Jeśli będziesz spędzał dużo czasu patrząc na teksty na ekranie, a nie interesują Cię kolorowe gry - monitor ten będzie najlepszym rozwiązaniem.

Jeśli zamierzasz używać trybu 80-kolumnowego, upewnij się, że wybrany przez Ciebie monitor monochromatyczny ma dostateczną rozdzielczość dla tego trybu pracy - nie wszystkie są odpowiednie.

### Programy na Cartridge'ach\*

Cartridge nie są urządzeniami peryferyjnymi w tym sensie co monitor lub stacja dysków. Gniazdo cartridge'a jest już wbudowane w Commodore 128, a więc nie trzeba kupować żadnych elementów, aby umożliwić użycie cartridge. Głównym celem produkcji cartridge'ów jest umożliwienie /w dogodny sposób/ przedsiębiorstwom wytwarzającym oprogramowanie, rozprzestrzenienie ich produktów. Tylko producent może zapisać program na cartridge'u, więc nie możesz wykorzystać tego urządzenia do magazynowania własnych programów /jak to jest możliwe w magnetofonie lub stacji dysków/.

Cartridge jest to plastikowe "pudełko" wielkości kaset magnetofonowej. Niektórzy producenci oprogramowania upodobali sobie takie rozwiązanie, ponieważ jest ono prawie niezniszczalne i praktycznie nie do skopiowania. Jednakże cartridge są kosztowniejsze niż dyskietki, więc inni producenci ich unikają. Istnieje duża różnorodność oprogramowania w postaci cartridge'ów na Commodore C64, które są także dostępne na Commodore C128 /w trybie C64/. Programy te nie są zbyt drogie: \$ 19 - \$ 40 za gry i do \$ 100 za poważniejszy program. Z pewnością powstanie też wiele programów na cartridge'ach specjalnie na C128. Ważną rzeczą, jeśli chodzi o cartridge, jest konieczność wyłączenia zasilania komputera przed wstawieniem cartridge'a. Jeśli tego nie zrobisz, ryzykujesz uszkodzenie komputera lub cartridge'a.

### Kasety magnetofonowe

Jeśli wyłączysz zasilanie C128, to cała zawartość pamięci RAM zostanie skasowana. Jeśli napisałeś długi program i

---

\* Cartridge - dołączany z zewnątrz moduł zawierający zaprogramowaną pamięć ROM; program zawarty w tej pamięci uruchamia się natychmiast po włączeniu zasilania /przyp.tiur./

chciałbyś go uruchomić bez ponownego wpisywania to musisz wiedzieć, że istnieje taka możliwość. Potrzebujesz takiego urządzenia zapisującego, by programy i inne dane, takie jak listy /pisanie za pomocą programu przetwarzania tekstów/ mogły być długo przechowywane. Urządzenia takie są czasem nazywane "mass storage" /pamięci masowe/, ponieważ mogą przechowywać więcej danych niż pamięć komputera. Do tego celu nie nadają się cartridge /niemożność zapisania własnego programu, a jedynie możliwość odczytywania programów producenta/.

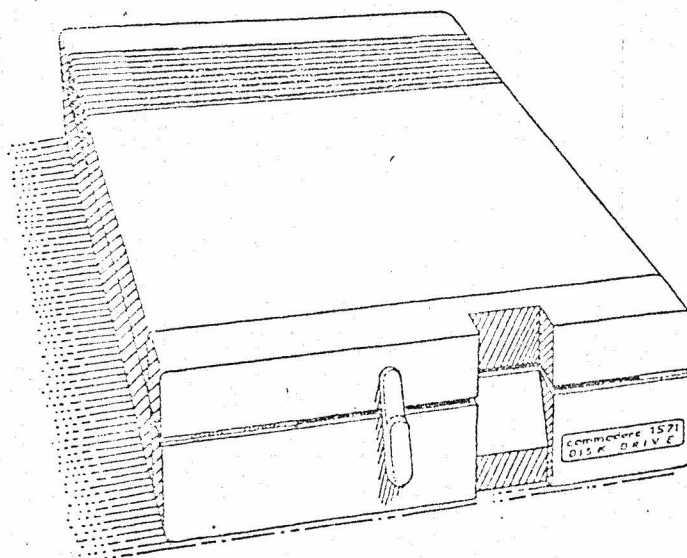
Kasety stanowią tanią formę pamięci masowych do C128. Są one identyczne jak kasety używane do magnetofonów. Do większości mikrokomputerów możliwe jest podłączenie dowolnego magnetofonu kasetowego, lecz C128 wymaga specjalnego magnetofonu o nazwie Datasette, produkowanego tylko przez firmę Commodore. /Datasette zapisuje sygnały cyfrowe dokładniej niż sygnały analogowe, używane w innych magnetofonach/. Jest to najtańszy sposób przechowywania programów i danych, ale jednocześnie bardzo powolny - musisz długo oczekiwać na załadowanie programu do komputera.

Większość użytkowników, którzy zaczynają od magnetofonów, szybko zmienia go na stację dysków, radzimy więc zacząć od stacji dysków! /jeśli stać Cię na to/. Ponad 90% użytkowników C64 wybrało stację dysków, a nie Datasette i prawdopodobnie analogicznie będzie z C128.

### Stacje dysków

Po urządzeniach wizyjnych, stacje dysków są najbardziej popularnymi urządzeniami peryferyjnymi do komputerów osobistych. Podobnie jak magnetofony, stacje dysków realizują 2 podstawowe cele: pozwalają wprowadzać do komputera gotowe /produkowane/ programy oraz przechowywać własne programy i dane, oferują one jednak znacznie więcej możliwości. Cho-

ciaż ich cena nie przekracza \$ 300, jest to już poważny zakup, ale za to otrzymujemy znacznie szybszą i dokładniejszą w działaniu, niż magnetofon kasetowy, pamięć masową urządze-



Rys.6 - Stacje dysków

nia, a więc warte tej ceny. Możliwe jest zastosowanie "starych" stacji /budowanych dla C64/ do C128. Najbardziej popularną stacją dla C64 jest 1541 - stacja jednodyskietkowa. Dwudyskietykowe /mogą pracować jednocześnie 2 dyskietyki/ stacje produkowane są też przez inne firmy.

Wszystkie te urządzenia będą współpracowały także z C128. Nie będą korzystały jednakże z największej zalety C128 tj. zwiększonej szybkości obsługi stacji dysków. Dla C128 Commodore wprowadziło całkiem nowe stycje dysków. 1571 jest jednodyskietkowym modelem pokazanym na rys.6. Stacja ta może pracować w trzech trybach, analogicznie do 3 systemów pracy C128. W trybie C64 jest ona całkowicie kompatybilna ze "stara" stacją 1541 /i pracuje z taką samą prędkością co 1541/. W trybie C128 pracuje 5 razy szybciej /!/, a w trybie CP/M pracuje jeszcze szybciej i jest kompatybilna z większością istniejących urządzeń, pracujących pod CP/M. Dwudyskietykowa wersja tej stacji nosi nazwę 1571.

#### Prędkość transmisji danych

Co ta nowa prędkość pracy stacji oznacza? "Stara" stacja 1541 przesyłała ok. 320 znaków na sekundę. Jeśli miałbyś pięciostronicowy list lub program o, powiedzmy, 15000 znaków, "stara" stacja przesyłałaby je w ciągu 46 sek. /lub nowa w trybie C64/. W trybie C128 nowa stacja przesyła 2000 znaków na sek., więc taki sam materiał będzie przesyłany tylko 7,5 sek. W trybie CP/M proces ten trwa jeszcze krócej. Przy 3500 znaków na sek. transmisja zajmie 4,2 sek. Jeśli kiedykolwiek oczekiwałeś na załadowanie długiego programu do C64, wiesz, że zwiększenie prędkości transmisji nie jest zwykłym snobizmem, ale istotnie ułatwi Ci życie. Im dłuższe piszesz programy /lub używasz/ i im częściej posługujesz się komputerem, tym bardziej docenisz wagę zwiększenia prędkości napędu stacji dysków. Nowa stacja 1571 dodatkowo może przecho-

wać 2 razy więcej informacji na dyskietce: 340k bajtów, podczas gdy "stara" 1541 - 170k bajtów.

Nowa stacja dysków /1571/ pozwala na dziesięciokrotnie nawet szybszą transmisję danych, niż "stara" 1541.

#### Drukarki

Istnieje wiele gier i innych programów na Commodore 128, gdzie nie istnieje potrzeba drukowania danych wyjściowych. Jednakże, jeśli chcesz wykorzystać Twój C128 do przekazywania tekstów lub do innych poważnych programów lub gdy jeśli chcesz pisać naprawdę długie programy w BASIC-u, będziesz potrzebował drukarki aby utrwalić rezultaty swojej pracy na papierze. Rodzaj potrzebnej drukarki zależy od tego jak szybko chcesz drukować teksty, jakiego wyglądu tekstu oczekujesz i ile pieniędzy możesz na to przeznaczyć.

Najtańszą drukarką jest drukarka mozaikowo-igłowa. Jest ona szybka, ale jakość druku nie jest najlepsza. Jeśli masz zamiar prowadzić poważną korespondencję korzystając z komputera, będziesz prawdopodobnie potrzebował drukarki z opcją. Wysoka jakość liter NLQ /letter quality/. Te drukarki są droższe, ale używają mechanizmów podobnych do używanych w maszynach do pisania, a więc druk jest przejrzysty, są one jednak zazwyczaj wolniejsze od poprzednich.

Commodore nie wyprodukowała żadnej specjalnej drukarki do C128, ale istnieje wiele różnych drukarek, które współpracują z C64 i bardzo dobrze będą pracowały z C128. Są to MPS-801, MPS-802 i MPS-803. 801 i 803 są najtańsze. 802 jest bardziej złożoną drukarką, przeznaczoną dla biznesu. Wszystkie te drukarki drukują zarówno tekst, jak i obrazy. Będąc zdolnymi do drukowania obrazów są one użyteczne przy tworzeniu sztuki za pomocą komputera czy też wykresów i grafów.

Inne firmy także produkują drukarki, które mogą być wykorzystane do pracy z C128. Należy jednak zwrócić uwagę, że jeśli drukarka nie została wyprodukowana specjalnie do pracy z C128, to potrzebuje ona układu sprzęgającego /mała skrzynka ze specjalną wtyczką/ z komputerem. Nawet po zastosowaniu tego układu, nie wszystkie drukarki będą drukowały obrazy, a więc upewnij się, że urządzenie robi to czego oczekujesz zanim zabierze je do domu.

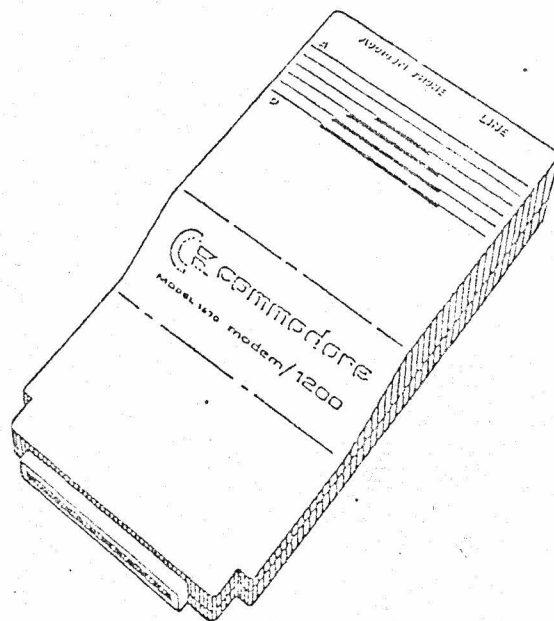
Do wykorzystania w biznesie będziesz pewnie potrzebował drukarki o "wysokiej jakości liter", takiej jak DPS-1101. Drukuje ona teksty o podobnej jakości, jak wysokiej klasy biurowa maszyna do pisania, ale jest wolniejsza od drukarki mozaikowej.

#### Modemy

Jedną z najdynamiczniej rozwijających się dziedzin wykorzystania komputerów domowych jest telekomunikacja. Oznacza to używanie komputera do przesyłania danych za pomocą linii telefonicznych. Do czego Ci to potrzebne? Możesz zechcieć przesłać program koledze z innego miasta. Możesz także być zainteresowany w uzyskaniu dostępu do aktualizowanej bazy danych. Takie serwisy, bazy danych dadzą informacje o ostatnich cenach, prognozach pogody, programach kinowych itp. /dotyczy USA i innych krajów zachodnich/. Pozwalają one także na utrzymanie łączności z grupą ludzi o wspólnych zainteresowaniach, jak np. filatelistyka czy muzyka elektroniczna. Grupy takie nazywać będziemy SIGs /Special Interest Groups/. Istnieje np. SIG sprzętu komputerowego Commodore. Grupy takie oferują darmowe programy, porady lub też stanowią forum dyskusyjne.

Najważniejszą cechą różniącą między sobą modemy, jest szybkość transmisji tzn. szybkość z jaką dany modem jest w





Rys.7 - Modem

stanie wysłać i odebrać pewną porcję informacji. Powolne modemy potrzebują dużo czasu na przesłanie nawet krótkiej informacji, a więc jeśli zamierzasz używać modemu dość często, szybszy model będzie dobrym nabytkiem.

Najwolniejsze modemy mają prędkość przesyłu 300 bodów. Co oznacza 1 bod? Jest to ilość znaków, które może przesłać modem w ciągu 1 sek. podzielona przez 10. A więc modem 300 bodowy może przesłać 30 znaków na sek. /prawdziwa definicja jest nieco inna, ale powyższa daje pojęcie o szybkości pracy modemu/.

Commodore sprzedaje różne modemy: modele 1600 i 1650 - bardzo popularne dotychczas i nowsze - 1660 i 1670, które dopiero wyprodukowano. Trzy pierwsze są modemami 300-bodowymi, a model 1670, pokazany na rys.7 pracuje z prędkością 1200 bodów. Modemy 300-bodowe kosztują od \$ 140 do \$ 200, podczas gdy 1670 osiąga cenę ok. \$ 300.

Istnieje szeroka gama modemów produkowanych przez inne firmy, takie jak Hayes, Human Engineered Software i Anchor Automation. Niektóre z tych modemów zostały skonstruowane specjalnie do C128, ale większość wymaga urządzenia sprzęgającego /interface'u/ o nazwie RS-232, który kosztuje ok. \$ 50 i jest produkowany przez wiele firm /w tym Commodore i MSD Systems/.

Oprócz prędkości, o klasie modemu decydują takie cechy jak: czy modem automatycznie przyjmuje telefony? Czy potrafi automatycznie nakręcać numery telefoniczne? Z urządzeń proponowanych przez Commodore wszystkie modemy, oprócz 1600, posiadają wyżej wymienione cechy, chociaż 1650 i 1660 wymagają specjalnego oprogramowania dla tych celów. Np. jeśli chcesz wykazać numer, to właśnie program utworzy odpowiednie sygnały elektryczne po każdej wybranej przez Ciebie cyfrze. W modelu 1670, modem sam generuje sygnały, program jest tylko potrzebny do określenia wybranego numeru. Model 1670 jest

uniwersalnym urządzeniem. Jest kompatybilny z szeroką gamą oprogramowania telekomunikacyjnego.

Modem powinien być urządzeniem peryferyjnym, które zakupisz jako jedno z pierwszych. Pozwoli Ci ono robić zakupy nie wychodząc z domu, posiadać informacje o ostatnich cenach, prognozach pogody, programach kinowych, elektronicznych biuletynach, a przede wszystkim otrzymasz bezpłatnie oprogramowanie.

#### Joysticki

W przypadku gier i różnych innych programów, będziesz prawdopodobnie potrzebował jednego lub więcej joysticków albo podobnych urządzeń. Joysticki są to małe, ręczne dźwignie, które pozwalają na bezpośrednie przesyłanie informacji typu góra-dół lub lewo-prawo do komputera. Większość joysticków zawiera także przycisk używany do uruchamiania działań lub laserów w grach komputerowych.

Oprócz joysticków istnieje szereg różnorodnych innych urządzeń w mniejszym lub większym stopniu spełniających podobne funkcje co joysticki, takich jak manipulatory kulowe, myszy, tablice dotykowe i wioselka /paddles/. Niektórzy użytkownicy preferują te, lub inne z tych urządzeń. Także niektóre gry i programy wymagają określonych urządzeń tego rodzaju. Jednakże joystick jest najpopularniejszym urządzeniem tego rodzaju i jest to dobry sprzęt na początek dla miłośników gier.

Zwróć uwagę, że nie wszystkie z tych urządzeń mogą być użyte do dowolnych programów. Joysticki i manipulatory kulowe są zazwyczaj kompatybilne, ponieważ obydwa urządzenia przesyłają informacje o kierunku do komputera. Natomiast tablice dotykowe, myszy i wioselka przesyłają informacje o położeniu. Bardzo często urządzenia te pracują ze "wskaźnikiem" pożądanego punktu na ekranie monitora.

#### INNE URZĄDZENIA PERYFERYJNE

Ilość urządzeń peryferyjnych produkowanych do Commodore 64 i Commodore 128 ciągle się powiększa. Istnieją np. syntezatory mowy lub czujniki otwierania drzwi od garażu, zabezpieczenia domu od złodziei lub ognia, lub też urządzenia do sterowania modelem kolejki. Za pomocą odpowiedniego sprzętu możesz wykorzystać swojego C128 do wykonywania rysunków technicznych poprzez rysowanie na ekranie lub też tworzenia muzyki za pomocą klawiatury muzycznej.

Nie będziemy tutaj opisywać tych urządzeń. To, co napisaliśmy o najbardziej popularnych urządzeniach peryferyjnych, powinno Ci wystarczyć dla wybrania odpowiednich urządzeń do Twojego C128.

#### Podłączanie urządzeń peryferyjnych do C128

Wszystkie urządzenia peryferyjne łączy się z C128 przez przyłączenie ich do gniazd, znajdujących się z boku lub tyłu Twojego C128. Liczba tych gniazd /portów/ jest duża i różnią się one między sobą rozmiarami i kształtem. Łatwo jest się pomylić i starać się podłączyć wtyczkę w niewłaściwe miejsce, a więc ta część rozdziału poświęcona będzie opisowi połączeń i wyjaśnieniu, które urządzenie podłączyć do których gniazd.

Pierwsza grupa gniazd, którą omówimy, znajduje się po prawej stronie Commodore 128, jak to pokazano na rys.8.

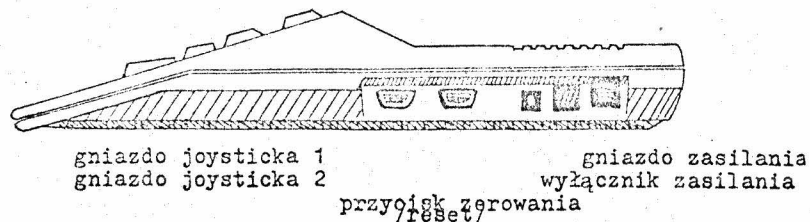
#### Badź ostrożny

Podłączanie urządzeń peryferyjnych do C128 jest tak proste, jak podłączanie elementów zestawu stereofonicznego - zazwyczaj istnieje tylko jeden sposób podłączenia wtyczki. Ponieważ niektóre gniazda C128 mają różne napięcia wyprowadzone na zewnątrz, dobrze jest wyłączyć zasilanie przed podłączeniem lub odłączeniem urządzenia peryferyjnego. Nigdy nie

wolno podłączać lub odłączać pamięci ROM na cartridge'ach oraz urządzeń peryferyjnych przyłączonych do portu użytkowego bez uprzedniego wyłączenia zasilania.

#### Zasilanie i zerowanie /Reset/

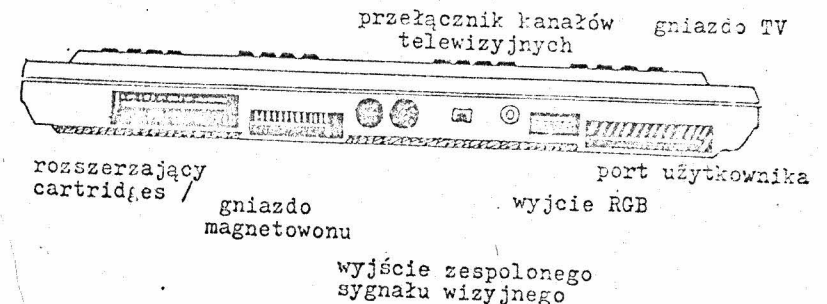
Na prawym boku C128, najbliższej tylnej ścianki, znajduje się gniazdo zasilania. Tu właśnie odłączony jest przewód zasilający. Źródłem zasilania jest mała skrzyneczka /zasilacz/ o długości ok. 6 cali, która transformuje napięcie sieci na niższe, używane w C128. Obok gniazda zasilania znajduje się wyłącznik zasilania. Zwróć uwagę, że mimo wyłączenia komputera, stabilizator nadal pracuje. Możesz więc wyłączyć zasilacz z sieci, jeśli nie będziesz używał komputera przez pewien czas. Przedłuż to "życie" zasilacza. Możesz także zakupić rozgałę-



Rys.8 - Prawy bok Commodore 128

ziacz z wyłącznikiem. Jest to użyteczne, ponieważ pozwoli podłączyć komputer oraz wszystkie peryferia do rozgałęziacza, przez co cały sprzęt będzie wyłączany i włączany w tym samym momencie. Obok wyłącznika zasilania znajduje się przycisk zerowania /reset/. Pozwala on wyzerować pamięć komputera bez wyłączania i powtórnego załączania. Możesz go użyć wtedy, gdy program "zawiesi się" i komputer nie reaguje na rozkazy z klawiatury. Także niektóre programy zabezpieczone przed kopiowaniem wymagają wyzerowania komputera przed załadowaniem ich. Możesz to zrobić, oczywiście, przez wyłączenie i powtarzanie włączenie komputera, ale jest to szkodliwe dla elektrycznych elementów komputera, a więc lepiej jest użyć przycisku zerowania.

#### Gniazda /porty/ sterownicze



Rys.9 - Tylna ściana komputera

Gniazda sterownicze używane są do joysticków lub podobnych urządzeń. Podłączanie takich urządzeń jest bardzo proste: przyłącza się je bezpośrednio do gniazd sterowniczych. Bądź jednak ostrożny, jeśli korzystasz tylko z jednego joysticka. Musisz uważać na to, aby podłączyć go do odpowiedniego gniazda. Oprogramowanie produkowane przez Commodore wymaga zazwyczaj podłączenia joysticka do portu 1, chociaż inne firmy mogą stosować port 2. Reszta gniazd znajduje się na tylnej ścianie C128.

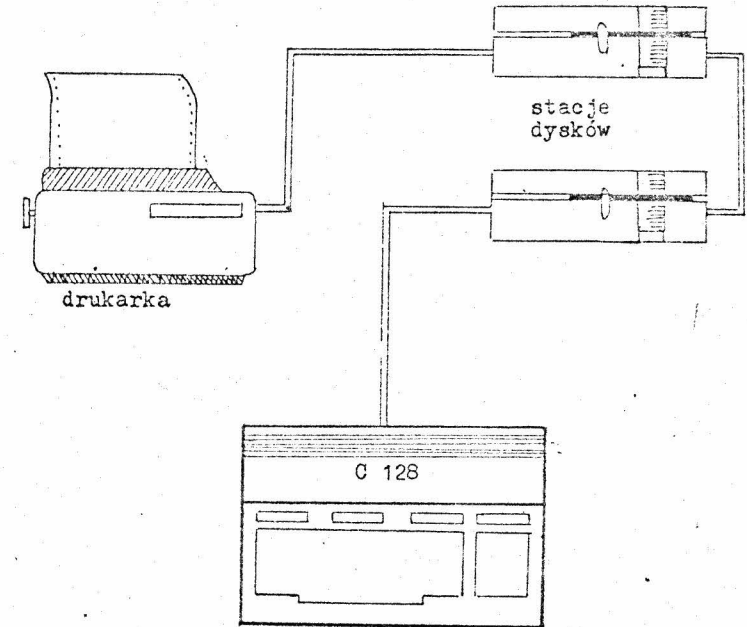
#### Gniazdo szyny szeregowej

Gniazdo to używane jest do różnego rodzaju urządzeń peryferyjnych, najczęściej stacji dysków lub drukarek. W technice szeregowy oznacza, że każdy znak lub bajt informacji jest przesyłany w oddzielnych bitach, jeden za drugim. Ponieważ każdy bajt składa się z 8 bitów, ten sposób przesyłania informacji jest teoretycznie 8 razy wolniejszy niż przesyłanie całego bajtu tak, jak ma to miejsce w wyjściu równoległym. Jednakże przesyłanie wszystkich bitów w jednej chwili wymaga ośmiu oddzielnych wyjść, a więc technika szeregowy jest tańsza.

Słowo "szyna" oznacza pewnego rodzaju ścieżkę, po której przesyłane są dane w komputerze. Szyna przypomina autostradę. Dane wysyłane są w dół szyny lub do różnych punktów przeznaczenia. Tak jak miejscowości położone wzdłuż autostrady, tak i urządzenia peryferyjne mogą zajmować różne miejsca wzdłuż szyny. Przyjrzyjmy się temu układowi bardziej szczegółowo.

#### Łańcuch szeregowy /nazwa ang. - daisy - draining/

Zaletą szyny szeregowy jest możliwość przyłączenia do niej więcej niż jednego urządzenia peryferyjnego. Jak to jest możliwe? Dzięki połączeniu w łańcuch szeregowy. Oznacza to, że pierwsze urządzenie jest podłączone bezpośrednio do szyny szo-



Rys.10 - Szeregowe łączenie urządzeń

regowej komputera. Następnie drugie urządzenie podłączone jest nie do komputera, ale do pierwszego urządzenia peryferyjnego. Trzecie urządzenie podłącza się do drugiego i tak dalej.

W ten sposób można podłączyć 5 urządzeń peryferyjnych. Pokazane jest to na rys.10.

Może pojawić się pytanie: jeśli wszystkie urządzenia podłączane są do jednego kabla, to w jaki sposób komputer wybiera urządzenie, z którym chce się komunikować?

Commodore 128 współpracuje z "inteligentnymi" urządzeniami peryferyjnymi, może więc komunikować się przez łańcuch szeregowy. Połączenia są prostsze niż w innych komputerach i rozszerzyć system jest znacznie łatwiej.

#### "Inteligentne" urządzenia peryferyjne

Aby zrozumieć jak działa szyna szeregową, powinienś wiedzieć, że C128 może używać specjalnego rodzaju urządzenia peryferyjnego, zwanego "inteligentnym" urządzeniem peryferyjnym. Pod tym względem Commodore jest niezwykle w świecie komputerowym. Większość systemów komputerowych może używać tylko "niemych" urządzeń peryferyjnych. Komputer szczegółowo określa, co mają robić, a urządzenia te informują komputer o wszystkich drobnych czynnościach, które wykonują i nie mogą podejmować decyzji bez pomocy komputera. Natomiast C64 i C128 mogą używać urządzeń, które wykonują wiele zadań samodzielnie. Stacja dysków np. może ustalać porządek na dyskietce, "wie" jakie programy znajdują się na dyskietce oraz może czytać i zapisywać programy na dyskietkę bez jakiegokolwiek pomocy komputera.

"Inteligencja" ta pozwala komputerowi komunikować się z poszczególnymi urządzeniami, nawet jeśli jest ich kilka, podłączonych do jednej szyny szeregową. Za każdym razem, gdy komputer przesyła informację do urządzenia /jak np. rozkaz do stacji dysków, aby przesłać zbiór do komputera/ wysyła także informację

o numerze urządzenia. Każde urządzenie ma swój numer i kiedy informacja pojawia się na szynie, urządzenie sprawdza numer. Jeśli stacja dysków posiada numer 8, a informacja jest dla urządzenia nr 4 /drukarka/, stacja dysków ignoruje ją, a drukarka przyjmie informację i wykona zawarte w niej polecenie - prawdopodobnie wydrukowanie jakiegoś zbioru.

Teraz wiesz dlaczego tylko niektóre urządzenia /te, budowane specjalnie dla Commodore/ będą współpracowały z C128. Muszą to być "inteligentne" urządzenia peryferyjne, zdolne do interpretowania specjalnych wiadomości wysyłanych z komputera.

#### Sieci i obsługa zbiorów

Innym przeznaczeniem wyjścia szeregowego jest podłączanie wielu komputerów C128, C64 razem tak, aby korzystały ze wspólnych urządzeń, jak np. stacja dysków i drukarka. Jest to system popularny w szkołach, gdzie ekonomiczniej jest udostępnić studentom tylko klawiaturę, a zakupić stację dysków i drukarkę, które będą obsługiwały całą klasę. W takim systemie komputer jest połączony - przez szynę szeregową - ze skrzynką służącą do obsługi zbiorów, która jest z kolei połączona z drukarką i stacją dysków.

#### Port użytkownika /User Port/

Port użytkownika znajduje się po prawej stronie tylnej ścianki C128. Najczęściej używany jest do podłączenia modemów, ale można tam podłączyć także wiele innych urządzeń, jak np. syntezator mowy, sterownik makiety kolejowej itp. Gniazdo to jest podobne do gniazda RS-232 /które jest standardem dla wielu komputerów/, ale są pewne różnice. Standardowy sprzęt RS-232 nie pasuje więc do tego gniazda. Modemy i inne urządze-

nia, które używają tego gniazda muszą być specjalnie modyfikowane do pracy ze sprzętem Commodore. Gniazdo to jest identyczne tak w C128 jak C64, a więc sprzęt, który pracował z C64 będzie pracował z C128. Aby przyłączyć modem Commodore'a do tego gniazda nie potrzebujesz kabla - przyłączasz go bezpośrednio do gniazda. Oczywiście, musisz także podłączyć modem do telefonu /zgodnie z instrukcją modemu/. Możliwe jest użycie sprzętu w standardzie RS-232, ale potrzebne jest tu specjalne urządzenie adaptacyjne do pracy z C128.

#### Wyjście TV i przełącznik kanałów

To małe, okrągłe wyjście jest używane do przyłączania zwykłego telewizora. Przewód łączy komputer z wejściem antenowym telewizora. Przełącznik kanałów powinien być nastawiony na kanał 3 lub 4, w zależności od tego, który z tych kanałów nie jest wykorzystywany w Twojej okolicy.

#### Wyjście złożone i bezpośrednie video

Jest to jedno z dwóch dużych, okrągłych wyjść /bardziej z lewej/ z tyłu C128. Ten rodzaj wyjścia nazywa się wyjściem DIN. Przewód wychodzący z niego dzieli się na kilka wtyczek typu RCA - jacks /te, które używa się do łączenia sprzętu hi-fi/. Jeśli używasz bezpośredniego monitora kolorowego, wtyczki te podłącza się do 3 oddzielnych gniazd z tyłu monitora, oznaczonych chroma, luma i audio. Jeśli używasz monitora złożonego, są tylko 2 wtyczki podłączane do monitora: sygnału video i audio.

#### Wyjście RGBI Video

Jest to małe, podłużne wyjście typu D. Służy do bezpośredniego połączenia monitora RGB przy użyciu 4 różnych wtyczek: czerwonej, zielonej, niebieskiej i audio.

#### Wyjście rozszerzenia pamięci /lub cartridge/

Jest to wyjście łatwe do użycia, ale jego wpływ na komputer jest złożony. Najczęściej używa się tego wyjścia dla programów zapisanych na cartridge'ach. Robi się to w prosty sposób: wyłącza się zasilanie komputera, podłącza cartridge, włącza zasilanie i gra lub program, który jest na cartridge'u przejmie sterowanie komputerem.

Oprócz gier możesz użyć tego wyjścia do zmiany języka rezydującego w komputerze np. z BASIC-a na FORTH. Jeśli znasz się trochę na komputerach, zainteresuje Cię informacja, że prawie każdy wewnętrzny sygnał C64 i C128 jest dostępny poprzez to złącze, zawierające szynę danych i szynę adresową. A więc, w zależności od tego, co jest przyłączone do tego wyjścia może to przejąć sterowanie komputerem, zamieniając go w urządzenie o innych cechach. Jest nawet możliwe podłączenie innego mikroprocesora, co spowoduje, że komputer nabierze nowych własności. C64 korzysta z tej możliwości, aby nabrać cech komputera pracującego w systemie CP/M /w C128 taka możliwość już rezyduje/.

#### Wyjście magnetofonu kasetowego

Jest to wyjście typu Molex. Przewód z Commodore Datasette lub innego magnetofonu zaprojektowanego do pracy z C128 jest podłączany do tego wyjścia.

Teraz, gdy już wiesz, jakich urządzeń peryferyjnych możesz użyć do C128 i jak je podłączyć, powinieneś poznać trzy systemy w jakich może pracować komputer osobisty Commodore 128: system C128, C64 i CP/M. Przy omawianiu tych trybów pracy powiemy, jakie urządzenia peryferyjne są najbardziej przydatne dla każdego z systemów.

### 3. SYSTEM C 128

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

W tym rozdziale dowiesz się:

- co to jest system C128
- co oferuje system C128
- jakiego sprzętu będziesz potrzebował do pracy w systemie C128

Poznasz:

- nowe właściwości języka BASIC 7.0
- system obsługi dyskietek /DOS/ w C128
- dodatkowe źródła informacji

System C128 jest jednym z trzech systemów operacyjnych dostępnych w komputerze osobistym Commodore 128. Głównie tym systemem operacyjnym różni się C128 od Commodore C64.

W pierwszej części tego rozdziału opiszemy najbardziej interesujące właściwości i co bardziej niezwykle cechy oraz omówimy różne możliwe zastosowania systemu C128.

Dalsza część rozdziału wyjaśnia, jakich dodatkowych elementów /urządzeń/ będziesz potrzebował do pracy w tym systemie. Następnie streścimy możliwości, jakie daje nowa wersja BASIC-a i nowy system obsługi dyskietek DOS.

Na koniec wskażemy źródła, z których będziesz mógł otrzymać dodatkowe informacje o tym systemie.

#### Co to jest system C128?

Jak sobie przypominasz ze wstępu, firma Commodore miała trzy cele, gdyby budowała nowy komputer Commodore 128. Przede

wszystkim nowy C128 powinien być całkowicie kompatybilny z C64 /system C64 w Commodore 128/, ale także powinien posiadać na tyle większe możliwości w porównaniu z C64, aby przyciągnąć uwagę klientów /system C128/. Nowy C128 powinien również współpracować z programami pisanyymi na komputery pracujące w systemie CP/M /system CP/M w C128/. Aby zaspokoić te wymagania Commodore musiał wbudować te trzy systemy operacyjne w C128. W pierwszej chwili taka sztuka wydaje się bardzo trudna do zrealizowania, ze względu na różne ich właściwości, jak to miało miejsce w przypadku systemów C64, C128 i CP/M. Wbudowanie 3 komputerów w jeden multisystem komputerowy, byłoby niemożliwe przy użyciu tradycyjnych metod projektowania. Zamiast tego firma porównała różnorakie elementy software'u i hardware'u, składające się na C64, C128 i CP/M, aby ustalić elementy wspólne. Następnie zbudowano C128 używając zespołów złożonych z tych elementów i korzystając z funkcji "organizatora" /inteligentnego sterownika pamięci i urządzeń/ do zbudowania układów składających się na komputer o cechach C64, C128 i CP/M. Rys.11 pokazuje taki układ różnych zespołów użytych w C128.

Możesz zauważyć, że niektóre zespoły Commodore 128 nie są wykorzystane w systemie C128. Celem selekcji było zapewnienie najlepszego układu właściwości C128. Szczegółowe informacje o tym, jak zorganizowane zostały te zespoły dla C64 i CP/M podamy w rozdziałach 4 i 5.

#### Co oferuje system C128

Commodore 128 pracujący w systemie C128 stanowi rozszerzenie Commodore 64. Ten tryb pracy oferuje wszystkie właściwości C64 z licznymi tylko jego wadami. System C128 posiada te same możliwości graficzne i dźwiękowe oraz wejścia /wyjścia, łącznie z wyprowadzoną szyną systemową, uniwersalnym programwalnym portem użytkownika i dwoma wewnętrznymi



zegarami czasu rzeczywistego. Może on nawet współpracować ze wszystkimi urządzeniami zewnętrznymi C64 takimi, jak monitor kolorowy 1702 i stacja dysków 1541. Z drugiej strony system C128 zwalnia użytkownika z pracy w nieco prymitywnym i ograniczonym BASIC-u 2.0 /w C64/ i równie przestarzałym systemie DOS.

A więc jakie zalety posiada C128? Po pierwsze- system C128 daje dostęp do prawie dwukrotnie większej pamięci niż C64. Pozwala także na pracę w 80-kolumnowym trybie tekstowym i graficznym, za pomocą ulepszonej, 96-klawiszowej klawiatury oraz zapewnia szybszą obsługę dysków w systemie z jednym lub dwoma napędami dysków.

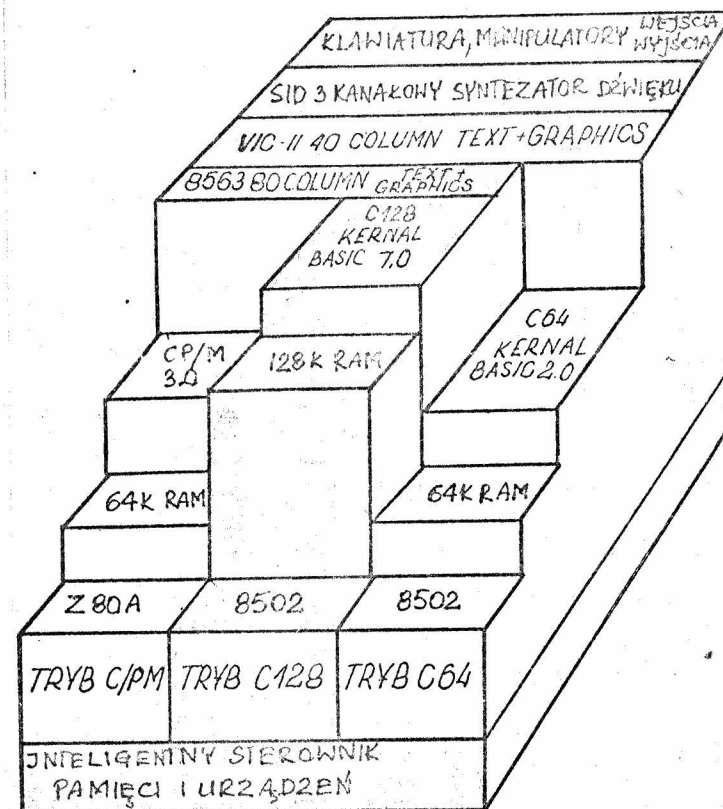
Opisane zalety systemu C128 są konsekwencją modyfikacji hardware'u komputera. Jeśli chodzi o stronę programową firma Commodore zmodyfikowała wewnętrzny system operacyjny /tzn. rezydentne rozkazy operacyjne/, dodając kilka istotnych ulepszeń do edytora ekranowego, wbudowała bardzo silny program monitora języka maszynowego oraz BASIC 7.0 - najsilniejszy ze wszystkich używanych przez komputery Commodore.

Na koniec, kiedy pracujesz w systemie C128 z nowymi wieloformatowymi stacjami dysków 1571 lub 1572, możesz korzystać z obustronnych dyskietek zamiast jednostronnych /2 x więcej miejsca na dyskietce/. Zauważalnie wzrasta też szybkość wymiany informacji między stacją dysków a komputerem.

Przyjrzyjmy się tym własnościom bardziej szczegółowo.

#### Zastosowania C128

Commodore 128 jest wyjątkowo "silnym" komputerem, o właściwościach znacznie poszerzających możliwości zastosowań w stosunku do C64. Pomimo dobrej grafiki i dźwięku C64 posiada wiele słabości, które ograniczają zakres jego zastosowań raczej do celów wymagających symulacji wzrokowej lub słuchowej, niż do



Rys.11 - Budowa systemu C128

celów wymagających szybkości i uniwersalności obliczeń. W konsekwencji głównym zastosowaniem C64 okazała się zabawa i edukacja. C128 zawiera 128K pamięci, możliwość wyboru ekranu 40- lub 80-kolumnowego i bardziej przydatny dla biznesu system pamięci dyskiety oraz wiele innych zalet co powoduje, że może być on użyty do wszystkich celów, do jakich może być użyty komputer osobisty.

Dzięki unikalnym właściwościom C128 znajduje zastosowanie w domu, szkole, biurze a nawet w przemyśle. Dla potwierdzenia tego podajemy przykłady praktycznych zastosowań C128:

- praca biurowa;
- prowadzenie interesów /przedsiębiorstw/;
- projektowanie wspomagane komputerem;
- sterowanie urządzeniami;
- projektowanie oprogramowania komputerowego;
- nauka wspomagana komputerem;
- przetwarzanie danych dla potrzeb prywatnych;
- zabawa.

#### Zalety C128

Aby zdać sobie sprawę dlaczego C128 posiada możliwości tak różnorodnych zastosowań, powinienś zrozumieć, w jaki sposób jego nowe właściwości wpływają na ten bardzo dobry komputer osobisty. Tym właśnie zajmiemy się w następnej części rozdziału.

#### Większa pamięć

System C128 oferuje 128K pamięci użytkowej tzn. 2 razy tyle co C64. Zawiera on także miejsce na poszerzony ROM - based software package. ROM - bsp jest to oprogramowanie stale przechowywane w specjalnym układzie scalonym pamięci typu ROM, będącej w dyspozycji C128. Ten rodzaj oprogramowania jest do-

łączany do pamięci C128 tak jak cartridge - przez wejście rozszerzające C128 lub jak układ ROM - wbudowany wewnątrz C128.

Dla wielu użytkowników C128 poszerzona pamięć Commodore 128 oznacza tylko jedno: możliwość uruchamiania większych i bardziej wyszukanych programów, jak sztuczna inteligencja /AI/, systemy ekspertowe, duże programy kalkulatorów na tabelkach /płacht/ i doskonalsze edytory tekstowe. Istotnym jest to, że wiele programów o zastosowaniu komercyjnym nie produkowanych przez Commodore mogło by być dostępne na C128 tylko wtedy, gdyby komputer ten posiadał odpowiednio dużą pamięć. Jeszcze kilka lat temu 64K pamięci było "aż nadto", dzisiaj najpopularniejsze oprogramowanie użytkowe przeznaczone jest do pracy na systemach, zawierających 128K lub 256K pamięci. C128 ze swoim systemem operacyjnym zapisanym w pamięci ROM i unikalnym, inteligentnym sterownikiem pamięci, udostępnia całkowicie 128K pamięci RAM dla programów użytkowych. Natomiast większość innych komputerów osobistych używających wbudowanego dyskowego systemu operacyjnego /DOS/ /jak CP/M czy MSDOS/ na dużą część dostępnej pamięci RAM zajęta przez swój DOS. Ponieważ w C128 ten problem nie istnieje, dla użytkownika dostępne jest całe 128K pamięci.

Pamięć C128 może być powiększona nawet o 384K przez dołączenie dodatkowej pamięci RAM. Pamięć ta jest przyłączana do C128 za pomocą złącza rozszerzającego w ten sam sposób, w jaki podłącza się cartridge. Żeby nie było niejasności - te dodatkowe 384K pamięci nie stanowi dodatku do już istniejącej pamięci RAM. Commodore używa jej do symulacji dyskiety /RAM-disc/.

RAM-disc jest to sekcja pamięci RAM używana tak, jakby była przyłączona dodatkowa stacja dysków. Programy i dane mogą być przesyłane między standardową pamięcią RAM symulo-

waną dyskietką tak, jakby była przyłączona stacja. Główną zaletą takiego układu jest szybkość - programy i zbiory mogą być przesyłane między obszarami pamięci w czasie rzędu części sekundy. Jest to istotnie duża prędkość przesyłu, nawet w porównaniu z prędkością przesyłu w bardzo szybkich stacjach dysków. Pamięć typu RAM-disc jest szczególnie użyteczna tam, gdzie w trakcie normalnej pracy wymagane jest ciągłe czytanie i zapisywanie informacji na dyskietkę. Oczywiście, kiedy skończysz używać dany program, wszystkie dane z RAM-disc muszą zostać przepisane na dyskietkę, gdyż inaczej zostaną stracone. Jeśli program ma być dostępny z obszaru pamięci RAM-disc, musi on najpierw zostać tam przepisany z dyskietki.

#### Poszerzona klawiatura

Commodore 128 posiada nową klawiaturę. Klawiatura ta pokazana na rys.12 posiada o 26 klawiszy więcej, niż klawiatura C64. Można w niej wyodrębnić część, która jest identyczna z oryginalną klawiaturą C64. Sekcja ta zawiera analogiczne klawisze tekstowe i graficzne, a także standardowe klawisze sterujące C64: Shift, Control, Run/Stop, CB, INS/DEL, Clear/Home i Restore.

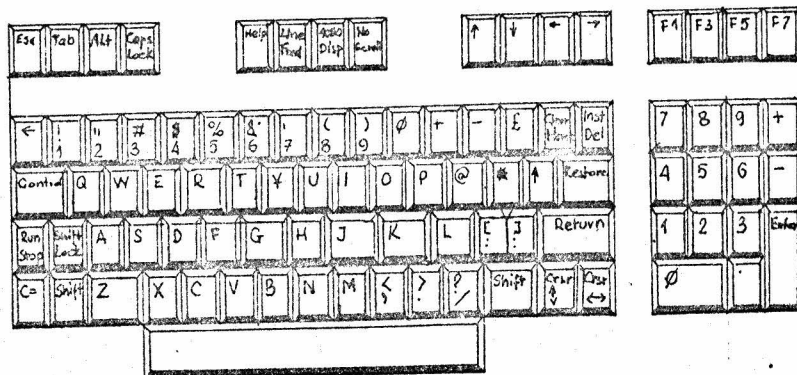
C128 posiada ponadto osobny 14-klawiszowy blok numeryczny, znajdujący się po prawej stronie podstawowej klawiatury. Jest on bardzo pomocny przy dłuższym używaniu komputera, jako kalkulatora. Ten zespół klawiszy posiada oprócz klawiszy numerycznych, własny plus, minus, kropkę dziesiętną i klawisz

Enter dla większej wygody. Powyżej głównej klawiatury znajduje się rząd 16 klawiszy, podzielonych na 4 grupy. Grupy te zawierają: klawisze funkcyjne, takie jak w C64, nowy zestaw klawiszy przesuwania kursora i 8 klawiszy o specjalnym przeznaczeniu. 4 klawisze funkcyjne analogiczne do klawiszy funkcyjnych C64 znajdują się powyżej klawiszy numerycznych. Każdy z tych klawiszy /funkcyjnych/ ma przypisane 2 różne funkcje

np. F1/F2, F3/F4 etc. czyli tak, jakby dostępne było 8 klawiszy. Klawisze te są bardzo ważne, ponieważ nie są one na stałe zdefiniowane - jak pozostałe. Oznacza to, że ich definicja /zarówno prosta, jak i złożona/ może być zmieniana przez program. Jest to duży plus, gdyż można używać tych klawiszy jako sterujących przy programach użytkowych. Klawiatura C128 zawiera dwa ważne klawisze: ESC i ALT /znajdują się one w grupie położonej najbardziej na lewo/. Klawisze ESC i ALT są popularne w małych komputerach, używanych w biznesie od wielu lat, a więc naturalnym jest, że znalazły się w C128, który może być używany w tej dziedzinie. Przeznaczeniem tych klawiszy jest zwiększenie możliwości klawiatury, co uzyskuje się przez kombinację tych klawiszy. Dzięki temu C128 staje się komputerem atrakcyjniejszym, szczególnie dla tych, którzy przyzwyczaili się już do pracy na tradycyjnych klawiaturach.

Klawisz Help jest przeznaczony szczególnie dla tych użytkowników C128, którzy planują pisać własne programy w BASIC-u. Kiedy C128 pracuje w BASIC-u, naciśnięcie klawisza Help powoduje wysłanie sygnału "pomóż mi - co zrobiłem źle?" do interpretera BASIC-a. Użycie tego klawisza w momencie, gdy program w BASIC-u zatrzymał się z powodu jakiegoś błędu, spowoduje wyświetlenie przez komputer informacji o miejscu w programie, w którym mógł wystąpić błąd. Oczywiście klawisz Help tak jak wszystkie klawisze klawiatury C128 jest redefiniowalny.

Pozostałe nowe klawisze, opisane w Tabl.3-1 służą do wywołania szeregu różnych funkcji przydatnych przy pracy z C128.



Rys.12 - Klawiatura C128

#### 40- i 80-kolumnowy tryb tekstowy oraz graficzny

Inno, nową właściwością systemu C128, jest zdolność do wyświetlania zarówno 80 jak i 40 znaków w linii. Jest to możliwe dzięki drugiemu procesorowi wizyjnemu wewnątrz C128. Procesor 8563 może zostać włączony w miejsce standardowego układu 40-kolumnowego przez naciśnięcie klawisza

40/80 Disp przed włączeniem zasilania C128 lub wyzerowaniem komputera przyciskiem RESET, gdy komputer jest włączony do sieci. Po wykonaniu tych czynności komputer automatycznie przechodzi do trybu 80-kolumnowego. Oczywiście, do użycia tego trybu potrzebujesz monitora zdolnego do wyświetlania 80 kolumn,

podłączonego do wyjścia RGBI w C128. Powrót do tego 40-kolumnowego wymaga także wyłączenia i włączenia zasilania lub naciśnięcia przycisku RESET. Możliwa jest też programowa zmiana trybu pracy. Dzięki temu programy komercyjne mogą upewnić się, czy C128 jest w odpowiednim trybie tekstowym,

Korzyści wynikające z nowego 80-kolumnowego trybu tekstowego są wielorakie. Naturalnie masz 80 znaków w 25 liniach, a więc możesz uzyskać jednocześnie do 2000 znaków na ekranie i to w kolorze. Co więcej, z powodu tego, że tryb tekstowy ma charakter mapy bitowej, C128 może wyświetlać wszelkie kombinacje dostępnych znaków graficznych i tekstowych /co jest możliwe w trybie 40-kolumnowym/, a nawet mieszać znaki tekstowe i graficzne z rysunkami w wysokiej rozdzielczości. Zestaw znaków zdefiniowanych w C128 /tzn. kształty znaków, które pojawiają się na ekranie po naciśnięciu jednego z klawiszy tekstowych lub graficznych/ może być zmieniany na zupełnie nowy zestaw znaków, nawet nowy język. Możliwe jest także, poprzez niewielkie zmiany wewnętrzne, otrzymanie zestawu znaków o zmniejszonych rozmiarach i uzyskanie obrazu o 132 kolumnach i 25 liniach.

Pamiętaj, że Commodore 128 ma dostęp do obu video procesorów /40- i 80-kolumnowego/ i w trybie pracy C128 może wykorzystać ten fakt. W C128 można wybierać żądany tryb ekranowy nie tylko przez naciśnięcie klawisza 40/80 Disp, ale może on być również ustalany programowo. W ten sposób możliwe jest wyprowadzenie danych na ekran w dwojaki sposób z jednego programu. Oba obrazy nie mogą być uaktualniane w tym samym czasie, ale można to zrobić wybierając jeden z dwóch. Ponieważ 40- i 80-kolumnowe wyjścia pochodzą z oddzielnych procesorów ekranowych, ekrany te mogą pokazywać różne informacje i nawet być w zupełnie różnych trybach graficznych i tekstowych. Aby wykorzystać tę zaletę dualnego obrazu ekranowego, C128 musi mieć oba wyjścia video podłączone do właściwych monitorów. Opiszemy to nieco później.

Można zwiększyć szybkość operacji wykonywanych przez C128 przez użycie tylko trybu 8 $\phi$ -kolumnowego. W tym celu należy użyć programu, który wyłączy układ odpowiedzialny za tryb 4 $\phi$ -kolumnowy. Operacje na danych w takim układzie wykonywane są prawie z dwukrotnie większą prędkością niż w układzie z procesorem ekranowym 4 $\phi$ -kolumnowym. Zwiększenie mocy obliczeniowej ma duże znaczenie, gdy komputer ma wykonywać operacje typu przetwarzania danych lub obliczeń numerycznych, typowych dla wykorzystania komputera w interesach, przemyśle i nauce.

Tablica 3-1 - Nowe klawisze Commodore 128

Klawisz	Definicja lub funkcja
Esc	Używany wraz z jednym lub więcej klawiszem tekstowym, aby wygenerować właściwą sekwencję kodu /używane w C128 w trybie emulatora CP/M/.
Tab	Używany do przesunięcia kursora do następnej pozycji tabulacji.
Alt	Używany jednocześnie z niektórymi klawiszami tekstowymi dla wygenerowania specjalnego kodu ALT.
Caps Lock	Używany w trybie 8 $\phi$ -kolumnowym, aby umożliwić niezależny wybór dużych i małych liter i dwóch zestawów graficznych.
Help	Używany w BASIC-u 7. $\phi$ dla zlokalizowania błędów w czasie programowania. Może być także używany jako klawisz "Help" /"Pomoocy"/ w niektórych typach programów.
Line Feed	Przesuwa o jedną linię bez generacji powrotu karetki /CR/.

Klawisz	Definicja lub funkcja
4 $\phi$ /8 $\phi$ Disp	Przełącza obraz z 4 $\phi$ na 8 $\phi$ -kolumnowy
No Scroll	Zabrania przesuwania ekranu przez zabezpieczenie kursora przed przekroczeniem 25 linii.  Przesuwa kursor o jedną linię do góry.  Przesunięcia kursora.
Klawisze funkcyjne	Klawisze definiowalne /patrz rozdział o BASIC 7. $\phi$ /
14-klawiszowy blok numeryczny	Blok klawiszy zgrupowanych, podobnie jak w kalkulatorze z oddzielnymi klawiszami dodawania, odejmowania, kropki i Enter .

#### Ulepszony BASIC

Język zwany BASIC-iem 7. $\phi$  dostępny w systemie C128 jest najbardziej rozbudowanym BASIC-iem oferowanym do tej pory przez firmę Commodore. Język ten zawiera 14 $\phi$  rozkazów, operacji i funkcji - 2 $\phi$  rozkazów, 35 operacji i 13 funkcji więcej niż w wersji BASIC'a 2. $\phi$  na C64.

BASIC 7. $\phi$  daje Ci większe możliwości sterowania: przebiegiem programu, testowaniem, wartościami logicznymi, funkcjami i procedurami matematycznymi, operacjami tekstowymi, danymi wyjściowymi, obliczeniami, magazynowaniem informacji, konstruowaniem i operacjami na zbiorach oraz wszystkimi dostępnymi operacjami graficznymi i dźwiękowymi. Dodatkowo

BASIC 7.0 dostarcza pełnego zestawu rozkazów, które upraszczają obsługę stacji dysków i przeglądania zbiorów. Mimo swoich ulepszeń, BASIC 7.0 pozostaje wciąż kompatybilny /w dół/ z wcześniejszymi wersjami BASIC-a 2.0 i 4.0. Oznacza to, że programy napisane w starszej wersji BASIC-a będą pracowały przy interpreterze 7.0 /ale nie na odwrót/. Ta kompatybilność odnosi się jednak tylko do programów pisanych jakby na papierze. Dyskowe wersje programów BASIC-a są przechowywane w zupełnie inny sposób i nie są kompatybilne.

BASIC 7.0 korzysta niemalże w pełni ze 128K pamięci RAM C128 /122365 bajtów/ i to w taki sposób, aby uniknąć wszelkich niedogodności, które pojawiały się w pracy większych programów na Commodore 64. /Programy na C64 często zatrzymują się na kilka sekund, kiedy BASIC 2.0 stara się zrobić miejsce w pamięci na więcej operacji/. 64K pamięci Commodore 128 przeznaczone jest do przechowywania instrukcji programu, podczas gdy wszystkie zmienne generowane przez program znajdują się w drugiej części /też 64K/ RAM. Przy tak dużej pamięci, możliwe jest podzielenie jej w taki sposób, że wiele programów może się znaleźć w pamięci w tej samej chwili.

Inną zaletą BASIC-a 7.0 jest wstępne zdefiniowanie na osiemnastu klawiszach funkcyjnych najczęściej używanych rozkazów BASIC-a. Oczywiście, mogą one być później przededefiniowane za pomocą rozkazów BASIC-a dla wykonywania innych operacji. zgodnie z Twoim własnym wyborem.

W dalszej części przejrzymy różne rozkazy i funkcje zawarte w BASIC-u 7.0.

#### Ulepszony edytor ekranowy

Komputery Commodore zawsze znane były z uniwersalności wbudowanych edytorów ekranowych. Te pełnoekranowe edytory działają jak przejściowo pracujące okna, w których możesz pisać i

zestawiać swoje rozkazy w BASIC-u. Dzięki nim możesz przesuwować kursor w jakiejkolwiek miejsce na ekranie, wstawiać i kasować dowolne ilości znaków i oczyścić ekran w całości.

Edytory ekranowe Commodore'a posiadają także dość interesujące właściwości. Np.: możesz wylistować grupę linii programu z pamięci na ekran, a następnie użyć kursora edytora ekranowego i dowolnych klawiszy, aby zmienić cokolwiek w linii zapisanej w BASIC-u, chociaż żadna zmiana nie zostanie wprowadzona do pamięci, dopóki nie naciśniesz klawisza Return. Powielanie linii jest bardzo proste: wystarczy wpisać nowy numer linii na miejsce starego i nacisnąć Return. Edytor ekranowy posiada także specjalny tryb pracy, który pozwala mieszać teksty, grafiki, a nawet działanie kursora w Twoim programie.

Firma Commodore dodała kilka nowych cech edytorowi ekranowemu C128 w stosunku do C64. Np.: linie programów mogą mieć długość 160 znaków /80 w C64/. Stało się to możliwe przez zastosowanie przemysłnego auto-przewijacza razem z edytorem ekranowym, który znajduje początek linii w BASIC-u na ekranie. W trybie 80-kolumnowym zdanie w BASIC-u może mieć długość dwóch linii, a w trybie 40-kolumnowym - 4 linii. Dwie inne zalety tego edytora ekranowego to: wstawianie i kasowanie linii oraz możliwość kontroli brzegu /marginesu/ 4 sposobami.

Wbudowany /rezydujący/ program monitora dla osób programujących w języku assemblera

Commodore 128 posiada wbudowany program monitora języka maszynowego. Ten program użytkowy /narzędziowy/ był kiedyś wbudowany na wszystkie PET Commodore'a, ale został później usunięty z VIC-20 i C64 w związku z ograniczeniami pamięci. Monitor /tak będziemy nazywali ten program/ jest ab-



solutnie podstawowym narzędziem dla każdego zainteresowanego pisaniem, testowaniem i modyfikowaniem programów napisanych w języku maszynowym 8502. Używany jest przez profesjonalnych programistów, a także właścicieli C128, którzy pragną pisać krótkie i szybkie programy w języku maszynowym, aby skrócić czas wykonywania niewralgicznych punktów programów pisanych w BASIC-u. Monitor jest także doskonałym narzędziem do nauki dla studentów, którzy chcą zrozumieć jak działa program w języku maszynowym /większość komercyjnych programów, takich jak np. BASIC 2.0, Logo i FORTH napisana jest w języku maszynowym/. Program monitora w C128 jest ulepszoną wersją wcześniejszych programów, wykorzystywanych przez Commodore. Podobna wersja była dostępna dla użytkowników C64 jako część Commodore'owskiego "Assembler Development Package".

Tablica 3-2 omawia rozkazy monitora wbudowane w C128. Więcej szczegółów o tym jak działa program monitora, można znaleźć w "Commodore 128 Users Guide" i "Commodore 128 System Guide".

Tablica 3-2 - Opis rozkazów monitora C128

Rozkaz	Nazwa rozkazu	Znaczenie rozkazu
A	ASSEMBLE	Asemlacja linii kodu maszynowego 6502
C	COMPARE	Porównanie dwóch obszarów pamięci i podanie różnic
D	DISASSEMBLE	Deasemblacja linii kodu maszynowego 6502
F	FILL	Wypełnienie obszaru pamięci podanym bajtem

Rozkaz	Nazwa rozkazu	Znaczenie rozkazu
G	GO	Rozpoczyna realizację programu maszynowego od określonego adresu
H	HUNT	Wyszukuje w pamięci wszystkie miejsca, w których występuje podany ciąg bajtów
L	LOAD	Ładuje zbiór z taśmy lub dysku
M	MEMORY	Wyświetla wartości hexadecymalne i ASCII odpowiadające danym lokacjom pamięci
R	REGISTERS	Wyświetla zawartość rejestrów 8502
S	SAVE	Zapisuje zawartość pamięci na taśmę lub dysk
T	TRANSFER	Przesyła blok instrukcji z jednej sekcji pamięci do drugiej
V	VERIFY	Porównuje zawartość pamięci z zawartością dyskietki lub taśmy
E	EXIT	Wyjście z programu MONITORA C128
.	/kropka/	Asemlacja linii programu 6502
>	/większy niż/	Modyfikacja pamięci
:	/średnik/	Modyfikacja zawartości rejestrów 8502
@	/małpa, embrion/	Wyświetla statwę dysku



### Szybszy i bardziej elegancki w obsłudze system dyskowy

Ostatnim bardzo istotnym ulepszeniem C128 jest doskonalszy system obsługi dyskietek. C128 nie tylko może współpracować z jednostronną stacją dysków 1541 od Commodore'a C64, ale także z "super inteligentnym" dwustronnymi systemami dyskowymi - stacjami dysków 1571 /pojedynczy napęd/ i 1572 /podwójny napęd/.

C128 może współpracować z jedną lub kilkoma stacjami typu 1541, ponieważ są one całkowicie kompatybilne. Jednakże w takim układzie nie są wykorzystane główne zalety C128 tj. większe możliwości magazynowania danych na jednej dyskietce i większa prędkość wymiany informacji między stacją a komputerem. Zaleta, jaką jest łatwiejsza obsługa systemu dyskowego, wynika z działania BASIC-a 7.0, a nie stacji dysków, a więc występuje też przy obsłudze stacji 1541.

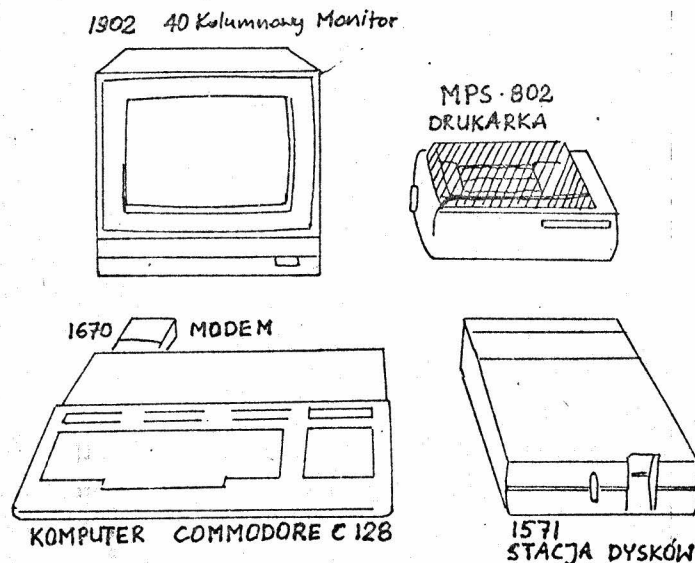
Prawdziwe polepszenie możliwości magazynowania informacji ma jednak miejsce, gdy C128 używany jest ze stacjami 1571 i 1572. Gdy użyte jest jedno z tych urządzeń, C128 ma bezpośredni dostęp do dwukrotnie większej ilości informacji na jednej dyskietce, niż w przypadku 1541. Np.: stacja 1541 mogła przechowywać 170K danych na dyskietce, a 1571 lub 1572 - 340K. Oczywiście, całkowita pojemność wzrasta do 680K, jeżeli umieszczisz dwie dyskietki w stacji 1572. Oprócz tego nowe stacje są także szybkie. Stacja 1541 z typową prędkością transmisji danych około 320 znaków na sekundę, jest jednak bardzo wolna i niestety nie staje się szybsza przez podłączenie do C128. Z drugiej strony, nowe stacje użyte w systemie C128 przesyłają 5200 znaków/sek. Tak duża prędkość transmisji czyni C128 maszyną użyteczną przy najbardziej nawet intensywnym korzystaniu z dyskietek w interesach. Commodore zaopatrzył C128 w łatwy do obsługi zestaw rozkazów DOS. Rozkazy DOS są takie same, jak użyte w starszych modelach firmy Commodore komputerach PET i CBM /BASIC 4.0/.

Instrukcje DOS umożliwiają realizację szeregu operacji dyskowych takich jak odczytywanie katalogu programów na dyskietce /directory/, formatowanie czystych dyskietek /przy użyciu rozkazu HEADER/, kopiowanie, nazywanie oraz przemieszczanie zbiorów. Przy wykorzystaniu stacji 1572 istnieje nawet instrukcja kopiowania jednej dyskietki na drugą /BACKUP/. Rozkazy te są bardzo ważne dla użytkowników C128 i dlatego omówimy je w dalszej części tego rozdziału.

### Jakiego sprzętu potrzebujesz do pracy w systemie C128

Jak już dowiedziałeś się z poprzednich rozdziałów Commodore 128 z całą swoją siłą i możliwościami jest po prostu liczącą /przetwarzającą/ częścią systemu. Nadal potrzebujesz takich urządzeń jak monitor do komunikowania się z maszyną, czy urządzenia o charakterze pamięci masowej do przechowywania programów i danych. Podstawowym elementem układu pracy C128 jest też drukarka /do dokumentowania ważniejszych obliczeń wykonywanych przez jednostkę centralną/ oraz modem /do połączenia C128 z innymi komputerami przez sieć telefoniczną/.

Ponieważ C128 jest kompatybilny z urządzeniami peryferyjnymi C64, masz bardzo szerokie możliwości ich wyboru, tak starych, jak i nowych. Jeśli już posiadasz kompletny zestaw urządzeń peryferyjnych C64 możesz je w prosty sposób połączyć z C128. Oczywiście, nie będziesz mógł korzystać z podstawowych zalet C128 takich, jak tryb pracy 80-kolumnowy czy zwiększone możliwości przechowywania informacji oraz prędkość wymiany informacji, ale będziesz mógł wykonać wszystkie czynności, jakie można wykonać dysponując takim sprzętem.



Rys.13 - Zalecany zestaw urządzeń do pracy w systemie C128

Najlepszym sposobem wykorzystania wszystkich możliwości C128 jest zastosowanie sprzętu skonstruowanego specjalnie dla niego. Rys.13 przedstawia zalecany przez nas zespół urządzeń.

Jeśli Twoje wymagania komputerowe nie są tak wysokie, np. jeśli po prostu chcesz eksperymentować z BASIC 7.0 i 40-kolumnowymi grafikami, może Ci wystarczyć telewizor kolorowy jako monitor i Datasette jako pamięć masowa.

Teraz, gdy już poznałeś główne cechy C128, które stanowią o jego wyższości nad C64 przyjrzymy się bardziej szczegółowo dwóm z nich: BASIC-owi 7.0 i nowemu DOS.

### WSZYSTKO O BASIC-u 7.0

W tym rozdziale opiszemy nowy BASIC 7.0, wbudowany w C128. Rozdział ten podzielony jest na 3 części. Pierwsza część omawia główne cechy BASIC-a 7.0, dając Ci ogólne pojęcie o tym języku. Druga część wymienia i objaśnia rozkazy i instrukcje wspólne dla systemu C64 /BASIC 2.0/ i systemu C128 /BASIC 7.0/ Commodore 128. W ten sposób poznasz instrukcje niezbędne do pisania programów, które pracowałyby w systemie C64 i C128. Trzecia część wymienia i omawia instrukcje charakterystyczne tylko dla systemu C128. Jeśli interesują Cię tylko usprawnienia zawarte w BASIC-u 7.0, przeczytaj tylko trzecią część tego rozdziału, ale jeśli chcesz programować w BASIC-u 7.0 powinieneś przeczytać też pierwszą część. Rozdziały 6 i 7 omawiają instrukcje dotyczące grafiki i dźwięku BASIC-a 7.0.

### Ważniejsze informacje o BASIC-u 7.0

Przyjrzymy się głównym cechom nowego BASIC-a 7.0 C128.

#### Tryby pracy

Kiedy używasz trybu pracy C64 wykorzystujesz BASIC 2.0, ten sam język, który zawarty jest w Commodore 64. Programy

będą więc pisane dokładnie w ten sam sposób, jak na C64. Gdy pracujesz w trybie C128 - BASIC 2.0 zastąpiony zostaje BASIC'iem 7.0 i stosujesz zupełnie inne metody programowania.

#### Zmodyfikowany Microsoft BASIC

BASIC Commodore 128 stanowi zmodyfikowaną wersję BASIC-a firmy Microsoft, zwanego w świecie urządzeń pracujących w systemie CP/M pod nazwą MBASIC /patrz opis MBASIC w rozdziale 5, system CP/M/. Nowy BASIC 7.0 bazuje na instrukcjach ze starego BASIC-a 2.0. Dodano do tego instrukcje i rozkazy z Commodore PET/CBM BASIC 4.0 /PET/CBM - rodzina komputerów profesjonalnych przeznaczonych do stosowania w interesach/. Dodatkowo dołączono szereg nowych instrukcji do sterowania dźwiękiem, grafiką, "oknami", spritami /figurki animacyjne/ i urządzeniami peryferyjnymi. BASIC firmy Microsoft był pierwszym BASIC-iem na mikroprocesory 8080, Z80, 6502 i zastosowany został do szeregu późniejszych mikrokomputerów. BASIC 7.0 zawiera wiele nowych instrukcji, które mają usunąć szereg słabości BASIC-a 2.0 głównie w sferze sterowania programem. Dodano także nowe instrukcje graficzne, dźwiękowe, edycji ekranowej, obsługi urządzeń WE/WY oraz klawiatury.

#### Edytor ekranowy

Nowy BASIC 7.0 pozwala na bezpośrednią współpracę /za pomocą klawiszy kierunkowych kursora/ z edytorem ekranowym. Mechanizm jest podobny jak w przypadku BASIC-owskiego edytora IBM PC, gdzie dla wybrania i edycji linii programu, przesuwa się kursor do linii, którą chce się zmienić, wpisuje poprawną informację i naciska Return dla zarejestrowania zmiany. Przy wykorzystaniu trybu 40-kolumnowego, BASIC 7.0 pozwala na wykorzystanie tych samych znaków, co C64. a więc nadal nie możesz korzystać jednocześnie z dużych i małych liter oraz dwóch zbiorów symboli graficznych.

Z drugiej strony, jeśli BASIC 7.0 pracuje w trybie 80-kolumnowym możesz mieszać wszelkie dostępne znaki tekstowe i graficzne. Także wtedy, kiedy wprowadzasz program w BASIC-u 7.0, umieszczanie spacji między słowami kluczowymi, zmiennymi i operatorami - jest dozwolone. Wprowadzanie spacji zwiększa przejrzystość instrukcji, a specja zajmują bardzo mało miejsca w pamięci, więc mogą być używane często.

#### Instrukcje graficzne oraz PRINT

Wszelkich znaków, kodów kontrolnych, barwy i znaków sterujących kursorem, można używać również wewnątrz cudzysłówów. Oznacza to, że możesz użyć każdego klawisza klawiatury bezpośrednio w rozkazie PRINT. A więc możesz korzystać wielokrotnie z klawiszy kierowania kursorem, symboli graficznych i kolorów w instrukcji PRINT i w ten sposób tworzyć złożone obrazy na ekranie.

#### Programowalne klawisze funkcyjne i pozostałe

Istnieją 4 klawisze funkcyjne i każdy z nich może realizować 2 funkcje. Klawisze te są programowalne z poziomu BASIC-a, a więc mogą być wykorzystane przez Twój program do dowolnych celów. Kiedy uruchamiasz BASIC klawisze zdefiniowane tak, jak to podano w tabl.3-3. F1, F3, F5 i F7 realizują swoje funkcje przez bezpośrednie naciśnięcie klawisza. Używając klawisza Shift uzyskujesz dostęp do klawiszy F2, F4, F6 i F8. Programowanie tych klawiszy odbywa się za pomocą rozkazu KEY, po którym następują inne instrukcje lub znaki BASIC-a. Np.:

KEY 7, "GRAPHIC 0" + CHR\$ /13/  
wykona rozkaz GRAPHIC 0 po naciśnięciu klawisza F7.

BASIC reaguje także na klawisze Help, No Scroll, Caps Lock, Alt, 40/80 Disp, Tab i Line Feed. Każdy

z tych klawiszy posiada swoją pierwotną definicję, ale może ona być zmieniana na poziomie BASIC-a.

#### Pamięć i BASIC 7.0

BASIC 7.0 posiada dostęp do 128K pamięci RAM C128. Te 128K podzielone jest na dwie sekcje. Dolna 64K przeznaczona jest do przechowywania programów w BASIC-u. Górna 64K wykorzystywana jest jako obszar pracy. W tym obszarze BASIC przechowuje zmienne, tablice i znaki. Interpreter BASIC-a zajmuje 48K pamięci ROM. Ponieważ tablice, znaki i zmienne znajdują się w oddzielnym obszarze pamięci, układ taki pozwala tekstowi programu w BASIC-u wykorzystać całe 64K, bez względu na wielkość tablic, wektorów i znaków. Umożliwia to uruchamianie programów zajmujących więcej pamięci niż w C64, a więc dłuższe i bardziej wymyślne programy są dla Ciebie dostępne.

Tablica 3-3 - Pierwotnie zdefiniowana treść klawiszy funkcyjnych BASIC-a 7.0 /w systemie C128/

Klawisz	Słowo kluczowe BASIC-a	C e l
F1	GRAPHIC	Ustawienie żadanego trybu graficznego
F2	DLOAD"	Załadowanie do komputera zbioru o podanej nazwie
F3	DIRECTORY	Daje wykaz zbiorów na dyskietce
F4	SCNCLR	Wyczyszczenie ekranu
F5	DSAVE"	Załadowanie na dyskietkę zbioru o podanej nazwie
F6	RUN	Uruchamia program zawarty w pamięci lub na dyskietce

Klawisz	Słowo kluczowe BASIC-a	C e l
F7	UST	Wyświetla listę programu zawartego w pamięci
F8	MONITOR	Uruchamia program monitora języka maszynowego

#### Zmienne i ograniczenia BASIC-a 7.0

BASIC 7.0 pozwala korzystać ze zmiennych trojakiego rodzaju: normalnych numerycznych /zmiennoprzecinkowych/, całkowitych i tekstowych. Zmienne numeryczne uznawane są przez BASIC za domyślne tzn. jeśli jakiejś zmiennej nie zdefiniujesz, to będzie ona traktowana jako numeryczna. Zmienne tego rodzaju mogą składać się z max. 9 cyfr i mogą reprezentować liczby z przedziału od  $10^{-38}$  do  $10^{38}$ .

W BASIC-u 7.0 zmienne całkowite /liczby bez części ułamkowej/ zawierają się w przedziale - 32768 do + 32767 i są wyróżniane w programie przez dodanie znaku % na końcu nazwy zmiennej.

Zmienne tekstowe, używane do przechowywania danych tekstowych i graficznych mogą zawierać do 255 znaków i są rozpoznawalne dzięki symbolowi \$ /dolar/ dodanemu do nazwy zmiennej. Nazwy zmiennych w BASIC-u 7.0 mogą mieć taką samą długość jak same zmienne, ale tylko 2 pierwsze znaki są rozpoznawalne. Innymi słowy np. nazwy zmiennych UNCOLA i UNLIMITED są dozwolone, ale przez BASIC będą traktowane jako nazwy tej samej zmiennej. Pierwszy znak musi być literą, a

drugi może być literą lub cyfrą. Symbole graficzne nie mogą być użyte w nazwie zmiennej. Oto kilka prawidłowych zmiennych, zapisanych w BASIC-u 7.0

FOOBAR      zmienna o formacie zmiennoprzecinkowym  
NAME\$      zmienna tekstowa  
P123%      zmienna całkowita

Tablice i wektory definiowane są w BASIC-u przez instrukcję DIM. Ilość wymiarów tablic jest ograniczona tylko przez długość linii programu. Do nazw tablic odnoszą się inne zasady niż do nazw zmiennych. Oprócz słów kluczowych BASIC-a, w BASIC-u 7.0 osiem nazw zmiennych jest zarezerwowanych i nie może być użytych w Twoim programie. Są one wymienione w Tabl. 3-4. Commodore 128 ma wbudowaną funkcję zegara 24-godzinnego, którego stan jest zmieniany co 1/60 sekundy. Ustawiania i odczytywania stanu zegara dokonuje się, korzystając z zarezerwowanej zmiennej o nazwie TI%. Zegar ten nie jest oddzielnym zegarem, pracującym w czasie rzeczywistym, lecz programowo realizowana jego symulacja /"jiffy", clock/ analogicznie do użytego w C64. Jest on bardzo użyteczny przy programowaniu, kiedy potrzeba odmierzyć krótki, dokładny odcinek czasu. Podczas wykonywania pewnych operacji /np. w przypadku użycia stacji dysków/ odliczanie czasu zostaje wstrzymane. Ponieważ nie ma zasilania baterijnego, zegar musi być ustawiany każdorazowo po załączeniu komputera.

Dzięki nowej instrukcji TRAP możliwe jest wykorzystanie zmiennych, opisujących błędy, które wystąpiły w programie do bardziej "twórczego" przetworzenia lub identyfikacji tychże błędów, niż to normalnie oferuje BASIC.

Zmienna DS związana z kanałem obsługi stacji dysków pozwoli Ci ustalić przyczynę migotania czerwonej lampki na stacji dysków.

## Inne szczegóły

BASIC 7.0 oferuje wszystkie normalne operatory matematyczne + - \* / 1/, a także operatory logiczne AND, OR i NOT. Długość linii w BASIC-u 7.0 jest ograniczona do 160 znaków /wliczając w to spacje/. A więc linia w BASIC-u zajmuje 2 linie na ekranie 80-kolumnowym i 4 linie na ekranie 40-kolumnowym.

## Przegląd instrukcji BASIC-a wspólnych dla typów C128 i C64

W tej części dokonamy krótkiego przeglądu instrukcji i rozkazów, które są wspólne dla BASIC-a 2.0 i 7.0. Instrukcje BASIC DOS są zawarte w oddzielnej części tego rozdziału.

## Instrukcje przypisania i podstawiania

Oba dialekty BASIC 2.0 i 7.0 oferują instrukcje wymazywania zmiennych, definiowania własnych funkcji i określania wymiarów tablic /wektorów/. Są one pokazane w Tabl. 3-5.

Tablica 3-4 - Zarezerwowane nazwy zmiennych w BASIC-u 7.0

Zmienne	Zastosowanie
ST	Zmienna stanu: mówi o powodzeniu operacji wejścia/wyjścia
TI	Przechowuje wartość 24-godzinnego zegara
TI%	Ustawia wartość początkową tego zegara zgodnie z podanym czasem
DS	Odczytuje kanał rozkazu stacji dysków, zwraca status
EL	Linia, w której pojawił się ostatni błąd

Zmienne	Zastosowanie
ER	Przechowuje kod ostatniego /tzn. najbliższego momentowi odczytu stanu zmiennej/ błędu, jaki pojawił się w czasie pracy programu
ERR\$	Zawiera komunikat odpowiadający błędowi, którego kod jest zawarty w zmiennej ER

#### Instrukcje programowania

Jak pokazuje Tabl.3-6 obydwa BASIC-i zawierają instrukcje, które pozwalają manipulować programem w pamięci.

#### Pętle

Instrukcje sterujące przebiegiem programu i tworzeniem pętli, wypisane są w Tabl.3-7. Standardowa pętla FOR ... NEXT pozwala na wykonywanie fragmentu programu ustaloną ilość razy. Wyrażenie STEP umożliwia podanie wartości kroku, o jaki będzie się zmieniać zmienna sterująca po każdym obiegu pętli. GOSUB wykorzystywany jest do wywoływania podprogramów. Wyrażenie RETURN przekazuje wykonywanie programu do linii następującej po GOSUB. ONGOSUB jest ulepszoną wersją GOSUB, która pozwala na indeksowane adresowanie podprogramów.

Tablica 3-5 - Instrukcje przypisania i podstawiania

Instrukcja	Działanie
CLR	Usuwanie wszystkich zmiennych i tablic
DEF	Definiowanie własnych funkcji
DIM	Deklarowanie wymiarów tablicy
LET	Instrukcja podstawiania /opcjonalna/

Tablica 3-6 - Instrukcje programowania

Instrukcja	Działanie
LIST	Listuje program z pamięci na ekran
REM	Pozwala na umieszczenie komentarza do linii
STOP	Program zatrzyma się i wydrukuje numer linii
NEW	Usuwa z pamięci aktualny program i wszystkie zmienne
RUN	Uruchamia program lub plik o podanej nazwie
CONT	Uruchamia zatrzymany program

#### Instrukcje sterowania programem

Wiele użytecznych instrukcji sterujących programem, zawartych jest w obu BASIC-ach. Pokazane są one w Tabl.3-8. Instrukcja WAIT powoduje zatrzymanie programu i oczekiwanie na zmianę wybranego bitu w określonym miejscu pamięci, a następnie wznowienie wykonywania programu. Używana jest głównie do kontrolowania stanu bitów w rejestrach wejścia/wyjścia takich, jak np. rejestry sprite'ów.

#### Sterowanie komputerem i pamięcią

Jak to pokazano w Tabl.3-9 istnieje kilka użytecznych instrukcji, umożliwiających kontrolowanie pamięci C128, takich jak PEEK /która odczytuje zawartość miejsca pamięci o podanym adresie/ czy POKE /wpisuje liczbę całkowitą z przedziału 0-255 do określonego miejsca w pamięci/. Instrukcja SYS wywołuje podprogram /w języku maszynowym/ o podanym adresie. Może ona także załadować /ale nie musi/ argumenty x, y i s do kumulatora, rejestrów x, y, a także na stos



mimo procesora 85 $\phi$ 2. Zakres adresów dostępnych bezpośrednio dla instrukcji SYS, leży w przedziale  $\phi$ -65535, dostęp do drugiego banku pamięci - pośrednio. USR to inny rozkaz, pozwalający na skorzystanie z podprogramu w języku maszynowym. W przeciwieństwie do SYS wymaga załadowania adresu początkowego tego podprogramu do dwóch specjalnych komórek pamięci /o adresach 1281 i 1282/. USR może także przekazywać parametry /wartości zmiennych/ z BASIC-a do podprogramu maszynowego i z podprogramu do BASIC-a. SYS nie pozwala na tego typu operacje.

Tablica 3-7 - Instrukcje pętli i skoków

Instrukcja	Zastosowanie
FOR...NEXT STEP	Wykonuje ciąg instrukcji wymienioną ilość razy
GO SUB /RETURN	Używane do skoku i powrotu z podprogramem
GOTO	Wykonanie skoku do linii o podanym numerze
ONGOSUB	Indeksowany GOSUB
ONGOTO	Indeksowany GOTO
RETURN	Powrót z podprogramu

Tablica 3-8 - Instrukcje sterowania programem

Instrukcja	Zastosowanie
END	Zakończenie wykonywania programu
RESUME	Wznowienie wykonywania programu do określonej linii
WAIT	Czekanie na zmianę określonego miejsca w pamięci

### Sterowanie operacjami wejścia/wyjścia i danymi

Oba BASIC-i 7. $\phi$  i 2. $\phi$ , jak to pokazano w Tab.-3-1 $\phi$ , dysponują szeregiem instrukcji, służących do wprowadzania i wyprowadzania informacji, a także przechowywania informacji i danych wewnątrz programu.

Instrukcje INPUT i GET służą przypisywaniu wartości, wprowadzonych za pomocą klawiatury zmiennym /tekstowym lub numerycznym/ w programie. Klawisz Return musi być zawsze przyciśnięty, aby zakończyć wprowadzanie informacji za pomocą instrukcji INPUT. Warunek ten nie musi być spełniony w przypadku instrukcji GET /używanej tylko do zmiennych tekstowych/, ponieważ użycie tego rozkazu powoduje automatyczne odczytywanie wszystkich znaków wprowadzonych za pomocą klawiatury i podstawianie pod zmienną. Instrukcje INPUT # i GET # są odmianami INPUT i GET, służącymi do wczytywania informacji z urządzeń wejściowych /które magazynują informacje w postaci zbiorów/ takich, jak Datasette, stacja dysków, modem'a nawet klawiatura.

Tablica 3-9 - Sterowanie komputerem/pamięcią

Instrukcja	Działanie
PEEK	Odczytuje zawartość danego miejsca pamięci
POKE	Wpisuje daną wartość w określone miejsce pamięci
SY.	Wywołuje podprogram w jęz.maszynowym i załadowuje rejestry
USR	Wywołuje podprogram w jęz.maszynowym i przesyła parametry



Tablica 3-10 - Instrukcje sterowania operacjami WE/WY  
i danymi

Instrukcja	C e l
GET	Pozwala wprowadzać znaki z klawiatury bez użycia Return
GET #	Wprowadzanie pojedynczego znaku z dowolnego urządzenia wejściowego
INPUT	Wprowadza ciąg znaków /liter, cyfr/ z klawiatury
INPUT #	Wprowadza ciąg znaków z dowolnego urządzenia wejściowego
PRINT	Wyświetla ciąg znaków na ekranie
PRINT #	Wyprowadza ciąg znaków na dowolne urządzenie wyjściowe
DATA	Zawiera listę cyfr i znaków graficznych
READ	Przypisuje znaki zawarte w instrukcji DATA zmiennym
RESTORE	Zeruje wskaźnik DATA/READ

PRINT jest instrukcją używaną do wyprowadzania informacji na monitor. PRINT # służy do wyprowadzania informacji na jakiekolwiek urządzenie wyjściowe.

CMD jest instrukcją pozwalającą na określenie z jakiego urządzenia wyjściowego chcemy korzystać. DATA jest używana do przechowywania listy stałych, cyfr lub liter wewnątrz programu w formacie ułatwiającym ich wykorzystanie w programie.

READ używa się do podstawienia stałych, zawartych w instrukcji DATA pod zmienne.

## Funkcje

Funkcje pozwalają na wykonywanie operacji na liczbach, literach, urządzeniach wyjściowych i pamięci. Tabl.3-11 pokazuje, że zarówno BASIC 7.0 jak i 2.0 zawierają szeroką gamę takich funkcji. Funkcje te służą np. do uzyskania wartości SIN lub COS, przypisania wartości liczbowej zmiennej tekstowej i na odwrót, etc. Informują również czy np. przycisk strzału /fire/ joysticka został naciśnięty.

Funkcje kończą opis instrukcji i słów kluczowych wspólnych dla BASIC-a 2.0 i 7.0. W następnej części omówimy instrukcje BASIC-a 7.0, które stanowią o jego wyższości nad wersją 2.0.

Tablica 3-11 - Funkcje

Instrukcje	C e l
FRE	Informuje o wielkości wolnej pamięci
ASC	Zwraca wartość pierwszego znaku dołączonego ciągu w kodzie ASCII
CHR\$	Zamienia liczbę na odpowiadający jej w kodzie ASCII znak
INSTR	Wyszukuje i zwraca grupę znaków w zmiennej tekstowej /ciągu/
LEFT\$	Wybiera N znaków od lewej strony zmiennej tekstowej
LEN	Oblicza długość ciągu znaków
MID\$	Wybiera N środkowych znaków zmiennej tekstowej
ABS	Moduł /wartość bezwzględna liczby/

Instrukcje	C e l
ATN	Arctangens /argument typu zmiennoprzecinkowego
COS	Cosinus
DEC	Przypisuje wartość dziesiętną zmiennej hexadecimalnej
EXP	Oznacza stałą e do potęgi X
FN	Pozwala na wywołanie własnej funkcji oraz przekazanie parametrów, dla których ma zostać obliczona
HEX\$	Przypisuje wartości dziesiętne, wartość hexadecimalną i znajduje jej odpowiednik tekstowy
INT	Obcina część ułamkową liczby zmiennoprzecinkowej
LOG	Oblicza wartość logarytmu naturalnego zmiennej X

#### Ulepszone instrukcje BASIC-a 7.0

Będziesz mógł ocenić prawdziwą wartość BASIC-a 7.0, kiedy zapoznasz się z nowymi instrukcjami tego języka. Opiszemy tu wszystkie nowe rozkazy, oprócz nowych instrukcji DOS, graficznych i dźwiękowych, którymi zajmiemy się później. Rozszerzeniu uległy rozkazy grafiki bitowej, sterowania sprit'ami i instrukcje kontrolujące 3 kanały dźwięku. Dodano też nowe instrukcje pętli, które pozwalają na pisanie bardziej złożonych programów oraz instrukcję okna, która pozwala wydzielić z obszaru roboczego monitora prostokątny wycinek, w którym będzie drukowany tekst.

#### Nowe instrukcje graficzne /powstałe w oparciu o grafikę bitową w systemie C128/

W Tabl.3-12 wymieniono nowe instrukcje, pozwalające rysować na ekranie o charakterze mapy bitowej /tzn. każdemu punktowi na ekranie odpowiada jeden lub grupa bitów w pamięci/. Masz możliwość rysowania linii, prostokątów, wielokątów i okręgów oraz wypełniania ich dowolnym kolorem. Możesz zapamiętać pewien obszar ekranu, jako ciąg znaków /zmienną tekstową/ w programie /większość BASIC-ów czyni to za pomocą tablicy/ i błyskawicznie odtworzyć go w dowolnym miejscu na ekranie.

Są też instrukcje pozwalające odczytać barwę danego pixela /punktu/ oraz zmieniające podziałkę ekranu na układ współrzędnych 1024 na 1024 punkty. Instrukcje te oraz ich działanie opisane zostały w rozdziale 6 - Grafiki.

Tablica 3-12 - Instrukcje graficzne BASIC-a 7.0

Instrukcja	Zastosowanie
BOX	Rysuje prostokąt
CHAR	Wyświetla znaki w dowolnym miejscu ekranu
CIRCLE	Rysuje okręgi, owale, prostokąty i wielokąty
COLOR	Wybiera barwę ramki, tła i znaków /pierwszego planu/
DRAW	Rysuje linie i zaznacza punkty
GRAPHIC	Wybiera jeden z 6 trybów graficznych
LOCATE	Ustawia kursor graficzny w punkcie o podanych współrzędnych
PAINT	Zamalowywuje wnętrza figury ograniczonej ciągłą linią

Instrukcja	Zastosowanie
SCALE	Zmienia skalę ekranu na 1024 x 1024 punkty
SCNCLR	Czyści ekran i nadaje kolor tła
SSHAPE	Zapamiętuje pewien obszar ekranu w zmiennej tekstowej
GSHAPE	Odtwarza fragment obrazu ze zmiennej tekstowej utworzonej instrukcją SSHAPE
WINDOW	Rysuje okno o podanych wymiarach
RCLR	Podaje kolor źródłowy
RDOT	Podaje kolor pixela /tło, pierwszy plan/
RWINDOW	Odczytuje wymiary okna
RGR	Podaje w jakim trybie znajduje się ekran

#### Nowe instrukcje sterowania sprite'ami

Jednymi z głównych zalet BASIC-a 7.0 są jego instrukcje sterowania sprite'ami /patrz Tabl.3-13/. Sprite'y, opisane szczegółowo w rozdziale 6, są to programowalne obiekty graficzne, używane często w grach i do potrzeb symulacyjnych. Ponieważ sprite'y "żyją" na niezależnych planszach bitowych, nie mogą się nawzajem wymazywać. Fakt, iż mogą poruszać się automatycznie dzięki układowi VIC II powoduje, że wydaje się być dość sensownym wykorzystanie ich w programach. Niestety, w C64 sprite'y musiały być sterowane za pomocą instrukcji POKE i PEEK, co znacznie utrudniało ich wykorzystanie. Nowy BASIC 7.0 zawiera zestaw zrozumiałych i prostych instrukcji, które pozwalają na łatwe manipulowanie sprite'ami. Możesz na przykład zorientować się dzięki nim, kiedy sprite'y nakładają się

na siebie na ekranie, albo kiedy przemieszczają się na tle wyświetlanego znaku.

Tablica 3-13 - Instrukcje sterowania sprite'ami

Instrukcja	Zastosowanie
BUMP	Rejestruje w co uderzył który sprite /1-8/
COLLISION	Powoduje skok GOSUB, kiedy sprite'y zderzą się lub sprite uderzy w wyświetlany znak
MOVSDR	Automatycznie przesuwa sprite'a w miejsce o podanych współrzędnych
SPRCOLOR	Ustawia dodatkowe kolory wszystkich sprite'ów w trybie wielokolorowym
SPRDEF	Powoduje włączenie edytora sprite'ów
SPRITE	Włącza sprite'a, koloruje go, zwiększa rozmiary, ustawia priorytet i tryb
SPRSAV	Przesyła ciąg znaków na miejsce sprite'a lub na odwrót
RSPCOLOR	Sprawdza, jakie dodatkowe kolory ostatnio zdefiniowano instr.SPRCOLOR
RSPDOS	Sprawdza prędkość i pozycje sprite'a
RSPRITE	Podaje atrybuty sprite'a

#### Instrukcje sterowanie dźwiękiem

Commodore 64 znany zawsze był ze swojego SID /3-kanałowy generator dźwięku/ i zaskakujących dźwięków, jakie zdolny był wygenerować: sonety Bacha, piękne symfonie oraz inne efekty dźwiękowe. BASIC 7.0 zaopatrzony został w in-

strukcje, które ułatwiają tworzenie takich efektów. Tabl.3-14 wymienia 6 instrukcji kontrolujących dźwięk. Zostały one szczegółowo opisane w rozdziale 7.

Zwróć uwagę, że instrukcje te bazują na 3-kanalowym syntezatorze muzycznym. Możesz tworzyć własne obwiednie dźwięku dla każdego kanału. Łatwo jest uzyskać gitarę, bęben czy saksofon, grające jednocześnie.

Instrukcja FILTER pozwala na dołączenie filtra częstotliwościowego do wyjścia układu SID w celu uzyskiwania dziwnych efektów dźwiękowych takich jak np. modulacja kołowa.

#### Instrukcje sterowania programem

Tabl.3-15 zawiera nowe instrukcje sterujące programem. Masz tu pętlę warunkową w postaci DO UNTIL i DO WHILE, a więc możesz napisać coś takiego:

```
100 DO
110 INPUT "CZY LUBISZ SWOJ KOMPUTER?"; ANS$
120 LOOP UNTIL ANS$ = "TAK"
130 PRINT "DZIEKUJE"
lub
100 DO UNTIL ANS$ = "QUIT"
.
. /instrukcje/
.
200 INPUT "AGAIN OR QUIT"; ANS$
210 LOOP
```

Tablica 3-14 - Instrukcje sterowania dźwiękiem

Instrukcja	Działanie
ENVELOPE	Kreuje parametry obwiedni instrumentu: ADSR, barwa dźwięku, szerokość impulsu, kształt fali etc.

Instrukcja	Działanie
FILTER	Określa rodzaj filtrów dołączanych do wyjść generatorów: dolnoprzepustowe, pasmowoprzepustowe i /lub/ górnoprzepustowe
PLAY	Określa utwór /nut/, jaki ma być odtworzony przez symulowany instrument
SOUND	Efekty dźwiękowe: dowolna częstotliwość z pełną kontrolą jej zmian w czasie
TEMPO	Ustawia tempo gry dla wszystkich trzech generatorów
VOL	Ustawia poziom głośności dla instrukcji SOUND

Możesz także powiedzieć DO WHILE /dosł. "rób gdy"/ jakieś wyrażenie jest prawdziwe bądź fałszywe.

Instrukcja EXIT pozwala na wyjście z pętli DO LOOP kiedy spełniony jest jakiś określony warunek. Instrukcja ELSE została dodana do instrukcji warunkowej IF THEN, aby umożliwić bardziej jasny i elegancki zapis.

Instrukcje BEGIN/BEND umożliwiają umieszczenie w instrukcji warunku instrukcji złożonej tzn. składającej się z kilku linii /która wykonywana jest, jeśli warunek jest spełniony/.

Przykład:

```
100 INPUT A
110 IF A < 100 THEN BEGIN
120 PRINT "LICZBA JEST MNIEJSZA OD 100"
130 BEND: ELSE: "LICZBA JEST WIEKSZA OD 100"
```

### PRINT USING, PUTDEF i GETKEY

C128 pozwala na użycie instrukcji PRINT USING co oznacza, że możesz decydować w jaki sposób liczby zostaną wyświetlone na ekranie. Była to jedna z instrukcji, której najbardziej brakowało użytkownikom BASIC-a 2.0. Możesz ustalić format wyświetlanej cyfry tzn. położenie kropki dziesiętnej. Analogicznie instrukcją PUTDEF pozwala na zmianę znaków definiujących format wyświetlanych liczb, na dowolne znaki występujące na klawiaturze /tzn. znakiem takim może być np. zamiast \$ - itp./.

Instrukcja GETKEY jest analogiczna do GET /która czeka na wciśnięcie klawisza/. Instrukcja GETKEY tym różni się od instrukcji GET, że nie musi być umieszczana w pętli, gdyż sama z siebie czeka na wciśnięcie klawisza.

Tablica 3-15 - Zaawansowane instrukcje sterowania programem

Instrukcja	C e l
DO/UNTIL/LOOP	Instrukcja pętli DO/UNTIL
DO/WHILE/LOOP	Instrukcja pętli DO/WHILE
EXIT	Natychmiastowe wyjście z pętli DO
IF THEN BEGIN/BEND	Umożliwia zastosowanie instrukcji złożonej z kilku linii po THEN
IF THEN ELSE	Pozwala na wykonanie instrukcji w zależności od wartości logicznej wyrażenia /innej dla prawdy, innej dla fałszu/

### Instrukcje pomocnicze

Ostatecznie, nowy BASIC 7.0 zawiera popularne instrukcje BASIC -a Microsoft, pomagające pisać programy w BASIC-u. Są one wymienione w Tabl.3-16.

Zwróć szczególną uwagę na instrukcję TRAP, która pozwala na "przechwycenie" błędów użytkownika i ich obróbkę, a także wyłączenie normalnego procesu sprawdzania poprawności składni BASIC-a.

Tablica 3-16 - Instrukcje wspomagające programowanie

Instrukcja	Efekt
AUTO	Włącza automatyczną numerację linii
RENUMBER	Przenumerowuje linie programu
DELETE	Kasuje grupę linii
HELP	Wyświetla numer linii, w której wystąpił błąd
TRAP	Skacze do podprogramu obsługi błędów
TRON, TROFF	Tryb śledzenia przebiegu programu: wyświetlane są numery aktualnie wykonywanych linii

### Co to jest DOS i jak z niego korzystać?

Stacje dysków Commodore są to pamięci masowe o swobodnym dostępie. Termin "swobodny" jedynie w przybliżeniu odnosi się do metody przechowywania programów i ich danych /użycie zbiorów i rekordów/. Metoda magazynowania informacji na dyskietkach ma niewątpliwą przerwę nad wykorzystaniem

do tego celu kasety magnetofonowej, ale jest ona osiągana kosztem większej złożoności stacji dysków. W przypadku magnetofonu, Ty jesteś odpowiedzialny za uruchomienie i zatrzymanie go oraz ustawienie kasety. Czynności te nie występują w przypadku użycia stacji dysków. Na przykład, jednostronna dyskietka może przechować setki programów i zbiorów danych z tysiącami rekordów. Te programy i zbiory pogrupowane są w małe bloki, które rozmieszczone są na dyskietce. Byłoby śmieszne sterować stacją dysków w taki sposób, jak kaseta magnetofonową. Tego rodzaju operacje /sterowanie magazynowaniem i prowadzenie ewidencji programów/ lepiej zostawić komputerowi. Kiedy używasz stacji dysków Commodore do pracy ze swoim C128, to tak jakbyś używał drugiego komputera z własnym programem o nazwie System Obsługi Dysków /DOS/. Jego zadaniem jest tylko udostępnienie Ci wszystkich możliwości, jakie daje metoda przechowywania danych na dyskietce.

#### Czym jest DOS?

Termin DOS<sup>\*</sup> oznacza wspólną nazwę dla grupy programów użytkowych, przeznaczonych do obsługi dysku. Programy te wykonują takie operacje, jak przygotowanie dyskietki dla przyszłych operacji magazynowania informacji, odczytywanie aktualnej zawartości dyskietki, kopiowanie danych z jednej dyskietki na drugą, kasowanie starych danych w celu przygotowania miejsca na nowe. Niektóre programy zawarte w DOS realizują operacje, których przeznaczenie jest mniej oczywiste. Te specjalne operacje DOS wynikają z wymagań stawianych przez C128. Np. jeśli program służący do prowadzenia korespondencji chce "zobaczyć" wykaz nazw i adresów już znajdujących się na dyskietce, musi wysłać kilka rozkazów do DOS stacji dysków. DOS; w odpowiedzi, używa swoich instrukcji dla zlokalizowania

---

\* DOS - Disc Operating System = dyskowy system operacyjny

zbiorów na dyskietce i wysyła żądane informacje do programu głównego. Tego typu operacje DOS są rzadko znane i wykorzystywane przez użytkowników C128, którzy korzystają przede wszystkim z gotowych programów, ale są bardzo ważne dla tych, którzy zamierzają sami je pisać. Commodore DOS zawiera szereg instrukcji, które są wykorzystywane przede wszystkim wewnątrz innych programów pisanych na C128. Jednakże najbardziej znane są te instrukcje DOS, z których /przynajmniej czasami/ korzysta każdy użytkownik stacji dysków.

#### Gdzie znajduje się Commodore DOS?

Stacje dysków Commodore'a są urządzeniami "inteligentnymi". Przypisuje im się takie określenie, ponieważ posiadają własne, wewnętrzne komputery, pracujące pod kontrolą programu DOS, który umożliwia wykonywanie rutynowych operacji dyskowych.

Co więcej, ponieważ DOS jest przechowywany w pamięci wbudowanej do stacji dysków, stacja może sama wykonywać wszelkie operacje DOS. Większość komputerów musi szczegółowo kontrolować pracę swoich stacji dysków, C128 musi jedynie wysłać pojedynczy rozkaz szyną szeregową do stacji dysków Commodore. Komputer stacji dysków porównuje ten rozkaz z listą instrukcji zawartych w programie DOS, wybiera odpowiednią funkcję i żądana operacja DOS zostaje wykonana. Ponieważ stacja dysków Commodore ma możliwość wykonania wszystkich operacji DOS, C128 potrzebuje tylko wysłać odpowiednią instrukcję do stacji i może zająć się innymi czynnościami.

DOS Commodore'a jest przechowywany w pamięci ROM stacji dysków. Stacja dysków posiada swój własny komputer, z mikroprocesorem, pamięciami RAM i ROM oraz układami WE/WY. Największą korzyścią wynikającą z przechowywania DOS w pamięci ROM stacji dysków /w porównaniu z wykorzystywaniem do tego

samemu celu dyskietki/ jest fakt, iż natychmiast po włączeniu stacji do sieci, jest on gotowy do działania. Inną korzyścią jest to, że obszar pamięci niezbędny do przechowywania DOS jest wydzielony z pamięci wewnętrznego komputera stacji dysków, a nie samego C128. Oznacza to, że więcej pamięci C128 jest dostępne dla programów. I ostatecznie, magazynowanie DOS w ROM-ie eliminuje potrzebę przechowywania go na Twoich dyskietkach, dzięki czemu całe dyskietki mogą być wykorzystane do przechowywania programów i danych.

#### Czy wszystkie programy DOS są takie same?

Commodore posiada trzy różne stacje dysków, które możesz przyłączyć do C128 - 1541 /przeznaczona do C64/, 1571 i 1572 /przeznaczone do pracy z komputerem Commodore/. Wszystkie te stacje posiadają wbudowany DOS. DOS stacji 1541 nieznacznie różni się od DOS stacji 1571 i 1572, ale jest z nim w znacznym stopniu kompatybilny. Sposób programowania stacji przez użytkownika w obu przypadkach jest taki sam, a tylko ich możliwości są różne.

Wiele różnic między starym i nowym DOS spowodowanych jest ograniczonymi możliwościami wolniejszej szyny szeregowej C64 i faktem, że stacja 1541 miała bezpośredni dostęp tylko do jednej strony dyskietki. Nowe stacje są precyzyjnymi urządzeniami potrafiącymi wykorzystać możliwości szybszej szyny szeregowej C128. Pozwalają one obsługiwać obie strony dyskietki jednocześnie. Mimo różnic, nowe stacje mogą czytać i zapisywać informacje na dowolnych dyskietkach, pochodzących ze stacji 1541 /używana jest tylko jedna strona dyskietki/. Dokonują tego przez zmianę "osobowości" swojego DOS na DOS 1541.

#### Jakie zadania może wykonać dla Ciebie DOS?

Ponieważ DOS jest przechowywany w stacji dysków Commodore, jego uruchomienie następuje natychmiast po włączeniu zasilania stacji. Następnie, jeśli chcesz skorzystać z usług DOS, musisz mu przesłać odpowiedni rozkaz. Na szczęście, C128 posiada kompletny zestaw łatwych do użycia instrukcji i rozkazów DOS, które są osiągalne poprzez BASIC 7.0.

Zwróć uwagę, że wiele z tych instrukcji jest podobnych do siebie. To częściowe podobieństwo było konieczne dla zapewnienia kompatybilności ze strukturą DOS użytą w BASIC-u 2.0 /C64/. Oto lista instrukcji i rozkazów nowego DOS:

APPEND	CONCAT	HEADER
BACKUP	DOPY	LOAD
BLOAD	DCLEAR	OPEN
BOOT	DCLOSE	RECORD
BSAVE	DIRECTORY	RENAME
CATALOG	DLOAD	SAVE
CLOSE	DSAVE	SCRATCH
COLLECT	DVERIFY	VERIFY
	DOPEN	

Omówmy teraz instrukcje i rozkazy jeden po drugim. To, co nazywamy rozkazem można wprowadzać bezpośrednio z klawiatury, choć można także wykorzystać w programie. Natomiast tego, co nazywamy instrukcją, można użyć tylko w programie, czyli będzie wykorzystywał głównie przez użytkowników C128, którzy zamierzają pisać swoje własne programy w BASIC-u 7.0. Opis, który tu zamieścimy ma za zadanie dać Ci pojęcie o funkcjach, które mogą spełniać te instrukcje i rozkazy.

Więcej szczegółów na ten temat możesz znaleźć w książce "C128 Users Guide" /"Przewodnik Użytkownika Komputera C128"/.



# APPEND

Rozkaz ten jest specjalnym rozkazem "otwierania zbioru", który używany jest w celu dodania nowych danych do końca już istniejącego zbioru.

# BACKUP

Rozkaz ten przekopiuje zbiory z jednej dyskietki na drugą. Może on być użyty tylko w systemie z dwoma napędami, jak np. 1572 i nie da się wykorzystać do programów zabezpieczonych przed kopiowaniem.

# BLOAD

Rozkaz ten używany jest do ładowania zbiorów typu binarnego z dyskietek do komputera. C128 posiada dwa rozkazy o podobnym charakterze: BLOAD i DLOAD. Ten ostatni jest używany do zbiorów takich jak programy w BASIC-u, które zawsze znajdują się w tym samym miejscu w pamięci. Jednakże, nie wszystkie programy są w BASIC-u, a więc nie muszą być konieczne przypisane do jednego miejsca pamięci. Zbiory binarne mogą znajdować się w dowolnym miejscu w pamięci. W czasie ładowania tych zbiorów do komputera musi być określony adres w pamięci komputera, gdzie umieszczony zostanie ładowany zbiór. Adres ten jest automatycznie przypisywany do danego zbioru przez rozkaz BSAVE, który umieści zbiór na dyskietce. BLOAD umieszcza dany zbiór binarny pod wymienionym adresem w pamięci komputera.

# BOOT

Jest to zupełnie nowy rozkaz DOS, używany do ładowania i uruchamiania zbiorów binarnych. Zbiory te są automatycznie uruchamiane po załadowaniu /oczywiście dotyczy to wyłącznie

zbiorów przeznaczonych do uruchomienia, np. zawierających program w jęz.maszynowym/. Podobnie jak BLOAD ładowane zbiory nie są przypisywane do stałego miejsca pamięci.

# BSAVE

Jest to odmiana rozkazu DSAVE. Stosuje się go do zbiorów binarnych. Rozkaz ten pozwala na załadowanie segmentu pamięci C128 na dyskietkę. Zbiory kreowane w ten sposób nie mogą być załadowane z powrotem do komputera za pomocą rozkazu DLOAD, gdyż zostaną automatycznie przeniesione do obszaru pamięci przeznaczonego do BASIC-a. Załadowania można dokonać rozkazem BLOAD.

# CATALOG

Rozkaz ten pozwala przejrzeć zawartość skorowidza zbiorów, zawartych na dyskietce. Sposób działania tego rozkazu jest inny niż w C64 /LOAD "p", 8:LIST/. Metoda wykorzystana w C64 nie działała dobrze w przypadku długich skorowidzów, ponieważ nie można było zatrzymać ich "przewijania" w czasie wyświetlania. Poza tym w metodzie tej skorowidz był ładowany do pamięci jako zbiór BASIC-a, kasując tym samym program, aktualnie znajdujący się w pamięci. Rozkaz CATALOG likwiduje obie te wady: wyświetlanie informacji na ekranie może być kontrolowane, a skorowidz ładowany jest bezpośrednio do pamięci ekranu, nie niszcząc tym samym programów w BASIC-u.

# CLOSE

Instrukcja ta zamyka zbiory poprzednio otwarte przez DOPEN lub OPEN. CLOSE używa się tylko w programach i można stosować do zbiorów wszelkiego typu łącznie ze zbiorami wysyłanymi do drukarki czy modemu.

### COLLECT

Instrukcja ta pozwala odzyskać obszar na dyskietce, na którym znajdują się niewłaściwie zamknięte zbiory.

### CONCAT

Pozwala na połączenie dwóch zbiorów danych w jeden.

### COPY

Rozkaz używany do kopiowania zbiorów. Może być wykorzystany do kopiowania z jednej dyskietki na drugą /co wymaga stacji dysków z dwoma napędami/ lub też do tworzenia kopii danych zbiorów na tej samej dyskietce, lecz pod inną nazwą.

### DCLEAR

Instrukcja ta kasuje zawartość otwartych kanałów do zbiorów na dyskietkach. Jest ona wykorzystywana przez programistów do zabezpieczenia się przed pozostałościami starych danych w kanale po poprzednich operacjach na zbiorach /patrz DCLOSE/.

### DCLOSE

Używana jest do zamykania jednego lub wszystkich aktualnie otwartych kanałów do stacji dysków. Terminem "kanał" określa się specjalną ścieżkę komunikacyjną między C128 a wewnętrznym komputerem stacji dysków. W czasie normalnej pracy wiele kanałów może być otwartych /każdy do innego zbioru/, ale ilość kanałów otwartych w tym samym czasie jest ograniczona.

Instrukcja DCLOSE jest wykorzystywana przez program do zamykania kanałów już niepotrzebnych /patrz DOPEN/.

### DIRECTORY

Wyświetla skorowidz dyskietki na ekranie /to samo co CATALOG/.

### DOPEN

Instrukcja używana jest do otwierania kanału komunikacji ze stacją dysków. Jest to dyskietkowa wersja instrukcji OPEN.

### DSAVE

Rozkaz DSAVE zapisuje na dyskietce program z pamięci BASIC-a. Jest to dyskietkowa wersja rozkazu SAVE /patrz BSAVE/.

### DVERIFY

Rozkaz ten porównuje program w pamięci z programem, znajdującym się na dyskietce /patrz również VERIFY/.

### HEADER

Rozkaz ten przygotowuje nową dyskietkę do jej pierwszych operacji czytania i zapisywania zbiorów. Dzieli on czystą dyskietkę na bloki oraz tworzy tablicę lokacyjną i czysty blok ze skorowidzem dyskietki. Nadaje także dyskietce specjalny tytuł i kod ID. Jest to rozkaz o charakterze destrukcyjnym, który całkowicie niszczy wszelkie zbiory, które wcześniej znalazły się na dyskietce.

### LOAD

LOAD używane jest do ładowania zbioru z Dataset lub stacji dysków. Odmiany tego rozkazu mogą być używane do

określonych zbiorów /programy w BASIC-u/ i do zbiorów typu binarnego /nierelokowalnych programów w jęz.maszynowym i danych/ - patrz SAVE.

#### RECORD

Ta instrukcja jest używana w programach do znajdowania dowolnego bajtu danego rekordu w określonym zbiorze.

#### RENAME

Rozkaz ten używany jest do zmiany nazwy zbioru na dyskietce. Zwróć uwagę, że dwa zbiory na jednej dyskietce nie mogą posiadać tej samej nazwy.

#### SAVE

Ta wersja rozkazu DSAVE może być użyta do zapisywania zbiorów na jakichkolwiek urządzeniach zewnętrznych Commodore, takie jak Dataset, ekran, modem, drukarka lub stacja dysków. Może być jednak użyte tylko do programów znajdujących się w standardowej pamięci BASIC-a. Programy typu binarnego mogą być zapisywane tylko przy wykorzystaniu pomocniczego programu monitora jęz.maszynowego wbudowanego w C128 /patrz LOAD/.

#### SCRATCH

Ten rozkaz używany jest do niszczenia lub usuwania zbioru lub grupy zbiorów z dyskietki. Zaleca się usuwanie istniejącego zbioru przed zapisaniem nowego pod tą samą nazwą. System DOS Commodore'a nie pozwoli Ci zapisać zbioru, jeśli istnieje już zbiór o tej samej nazwie.

#### VERIFY

Tę wersję DVERIFY stosuje się zarówno do zbiorów umieszczonych na Dataset, jak i tych ze stacji dysków.

#### Czy DOS można zmienić?

System DOS, który omawialiśmy do tej pory jest jednym z dwóch dostępnych systemów DOS w Commodore 128. Ten szczególny DOS znajduje się w stacji dysków Commodore i jest najłatwiej osiągalny. Jest to też jedyny DOS jaki możesz wykorzystać, gdy komputer pracuje w systemie C128 lub C64. Jednakże ten wbudowany DOS może być zastąpiony innym systemem DOS zwanym CP/M. Jak, dlaczego i kiedy trzeba użyć tego systemu - wyjaśnione zostanie w rozdziale 5.

#### Gdzie możesz znaleźć więcej informacji o C128?

Bez względu na to jak prosty w obsłudze jest komputer i jak nieskomplikowane są jego operacje, jego właściciele zawsze będą zabiegać o dostęp do bardziej szczegółowej dokumentacji, specjalnych opisów obsługi i wykorzystania komputera i wielu innych publikacji z tym związanych. Na szczęście istnieje wiele źródeł, z których mogą czerpać informacje tego typu użytkownicy C128.

#### Książki, które powinieneś mieć

Jeśli dotychczasowy rynek Commodore 64 można traktować jako wyznacznik ilości książek o C128, które zostaną napisane, to będziesz miał z czego wybierać. Pozycje te podejmą takie tematy, jak programowanie w BASIC-u C128, pierwsze kroki z C128, wewnątrz C128, zastosowania C128, użycie grafiki i dźwięku C128, biznes i C128, programowanie w jęz.maszynowym. Dodatkowo pojawiają się też książki zawierające ciekawe rozwiązania problemów informatycznych i programy. Wy-

starczy rzucić okiem na pobliską księgarnię, aby przekonać się, że ilość publikacji o C128 wzrasta w szybkim tempie\*.  
/Aktualnie w Polsce pojawił się na giełdach "The C128 User's Guide"/.

#### Czasopisma dla użytkowników komputerów Commodore

Właściciele C128 mają do wyboru szeroki wachlarz czasopism, traktujących o komputerach. Wiele z nich przeznaczonych jest specjalnie dla użytkowników sprzętu Commodore'a. Różnią się one od siebie jeśli chodzi o formę i poruszane problemy, tak więc możesz wybrać jeden lub 2 tytuły, które zaspokoją Twoje potrzeby.

Oto częściowa lista publikacji przeznaczonych przede wszystkim dla użytkowników sprzętu Commodore /początkujących i bardziej zaawansowanych/.

- 1/ "Commodore Power Play" wydawane przez Contemporary Marketing Inc.
- 2/ "Commodore Microcomputers" - " -
- 3/ "COMPUTE!'s Gazette" wydawane przez COMPUTE! Publications Inc.
- 4/ "COMPUTE!" - " -
- 5/ "RUN" CW Communications/Peterborough Inc.
- 6/ "The Transactor" Transactor Publishing Inc.

#### Grupa użytkowników Commodore

Jedną z najlepszych i najłatwiejszych form pomocy dla nowych właścicieli C128 jest lokalna grupa użytkowników Commodore. Istnieje ponad 600 takich organizacji w 16 krajach i 50 stanach. Np. w samej Californii istnieje 66 takich klubów. Kompletne listy takich grup z adresami są systematycznie publi-

\* Nie dotyczy PRL

owane w takich czasopismach jak "Commodore Microcomputers" i "RUN".

Grupa użytkowników komputera stanowi z założenia klub otwarty dla wszystkich o podobnych zainteresowaniach. Typowymi członkami takiej grupy będą użytkownicy nowego typu komputera. Ich celem będzie pomoc w udzielaniu informacji o nowym sprzęcie. Wielu sponsorów takich grup prowadzi kursy wykorzystania sprzętu i programowania oraz gromadzi ogólnodostępne oprogramowanie. Niektóre grupy użytkowników sprzętu Commodore prowadzą elektroniczne banki informacji /system komputerowy, z którym można się skontaktować za pomocą modemu akustycznego. Takie elektroniczne banki informacji spełniają wiele podobnych funkcji jak same kluby komputerowe tzn. np. forum dyskusyjne i bibliotek oprogramowania.

#### Sieć informacyjna Commodore'a

Sieć informacyjna Commodore'a jest dostępna dla wszystkich zainteresowanych komputerami Commodore, którzy mają dostęp do CompuServe /elektroniczny system bazy danych/. W przeciwieństwie do innych elektronicznych banków informacji, sieć ta jest prowadzona przez firmę Commodore. Sieć informacyjna Commodore jest aktualnie podzielona na wiele mniejszych specjalistycznych grup zainteresowań /SIG/, które nieznacznie różnią się od siebie. Każda z nich posiada własną sekcję pomocy, bazę danych i bibliotekę oprogramowania. Więcej informacji o grupach SIG Commodore'a można uzyskać kontaktując się z systemem CompuServe\* za pomocą modemu /za pomocą rozkazu GO-PCS/.

Szczegółowe informacje o CompuServe można uzyskać w każdym sklepie komputerowym\*.

\* Dotyczy USA

### Informacje o oprogramowaniu

Następną publikacją, która zainteresuje właścicieli C128 jest "Software Encyclopedia" opublikowana i systematycznie uaktualniana przez Commodore Business Machines.

Encyklopedia ta zawiera setki opisów różnych programów na komputery Commodore, włącznie z programami Commodore 128.

### 4. SYSTEM C 64

=====

W rozdziale tym dowiesz się:

- co to jest system C64
- co system ten ma do zaoferowania
- jakiego sprzętu będziesz potrzebował do pracy w tym systemie
- o popularnych programach na C64
- czym różni się BASIC 2.0 od BASIC-a 7.0
- jak wygląda struktura DOS w systemie C64

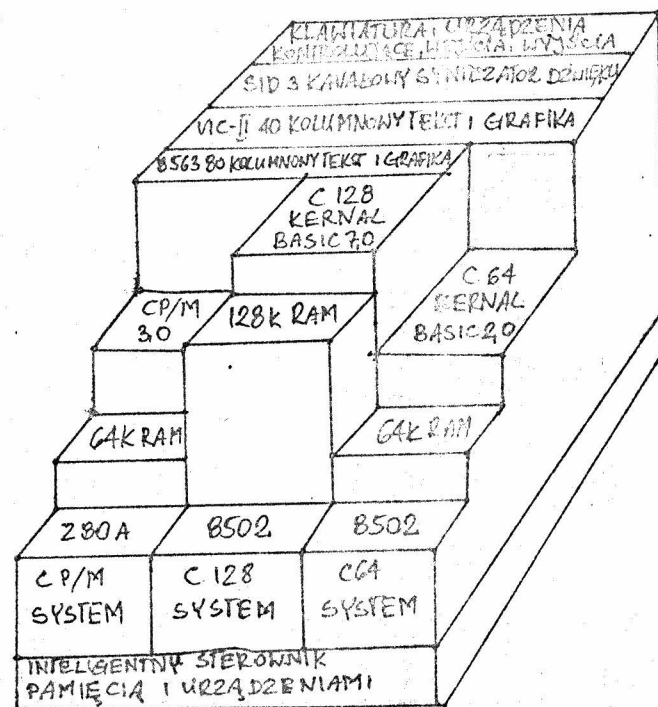
Drugim trybem pracy Commodore 128 jest tryb C64. W tym rozdziale wyjaśnimy na czym on polega, dlaczego jest tak ważny dla Ciebie, jakich urządzeń zewnętrznych będziesz potrzebował oraz w jaki sposób znaleźć się w tym trybie pracy. Następnie opiszemy szereg interesujących programów, które pozwalają na szerokie wykorzystanie Twojego sprzętu.

W ostatniej części tego rozdziału opiszemy zmiany, jakie pojawiają się w Commodore 128, gdy zostanie przełączony na tryb C64.

Nasze omówienie systemu /trybu/ C64 zakończymy krótkim przeglądem zmian jakie powoduje tryb C64 w systemie DOS. Zaczniemy od definicji trybu C64.

#### Co to jest tryb C64?

W ostatnim rozdziale wyjaśniliśmy, że Commodore 128 zbudowany jest z elementów składających się na trzy różne komputery. Jeden z zespołów układów tworzy tryb C128, natomiast drugi - C64. Elementy składające się na system C64 pokazano na rys. 14. Rozwiązują one wewnętrzny układ hard-



Rys.14 - Struktura systemu C64 w Commodore 128

ware i software Commodore'a 128, tworząc wierną kopię Commodore'a 64. A więc jeśli używasz Commodore 128 w trybie C64 otrzymujesz w rzeczywistości Commodore 64. Naturalnie sprawność pracy w tym trybie zależy od Twojej znajomości Commodore 64. Jeśli nie bardzo znasz komputer C64 - oto jego krótkie omówienie.

#### Historia Commodore 64

Określenie między rokiem 1979 a 1982 przynosi pierwszy duży sukces firmy Commodore na rynku komputerów osobistych - była to era komputera osobistego VIC-20.

VIC był to mały, prosty komputer o klawiaturze podobnej do maszyny do pisania, z wbudowanym BASIC-iem, kolorową grafiką, przyjemnymi efektami dźwiękowymi i pamięcią RAM 6K. Był to zresztą najtańszy do tej pory komputer. Cena VIC-20 nie była przypadkowa: miał on stanowić konkurencję dla video-gier, takich jak urządzenie ATARI 2600. Strategia ta okazała się słuszną i w roku 1981 ponad milion ludzi posiadało już ten komputer. Jednakże, ponieważ technologia produkcji tego komputera nie stała na najwyższym poziomie, firma Commodore doszła do wniosku, że aby pozostać numerem 1 na rynku komputerowym, musi stworzyć nowy, lepszy komputer do końca 1982 r.

Okres od 1979 do 1981 był także szczytem szaleństwa na punkcie domowych video-gier. To właśnie w tym okresie związana z Commodore firma MOS Technology rozpoczęła opracowywanie serii super-silnych scalonych układów wizyjnych i dźwiękowych. W 1981 r. zakończyła ona projektowanie dwóch mikroukładów specjalizowanych o dużym stopniu scalenia: układu interfejsu dźwiękowego 6581 /SID - Sound Interface Device/ oraz kontrolera wizyjnego 6567 VICII /Video Interface Chip II/.

Obie te kości razem były zdolne stworzyć naprawdę niezwykłą grafikę i dźwięk. Firma Commodore początkowo zamierzała wykorzystać oba te układy jako "kamień węgielny" video-gier nowej generacji. Fenomenalny sukces mikrokomputera VIC-20 skłonił jednak firmę do użycia ich jako centralnego elementu nowego komputera osobistego o pamięci 64K bajtów. Nikt do tego momentu nie zaoferował komputera osobistego o takiej pamięci. Tak narodziła się koncepcja stworzenia Commodore 64.

Kiedy Commodore 64 pojawił się na rynku komputerowym w 1982 r. posiadał wszystkie niezbędne cechy, aby konkurować z popularnymi już komputerami ATARI 800, TI 99/4 i Apple II. Podobnie jak VIC-20, C64 wyróżniał się spośród konkurentów niską ceną jak na swoje możliwości. Z końcem 1985 r. ponad 2 miliony ludzi posiadało już C64 i istniało już ponad 6 tysięcy programów na ten komputer. Tylko jeden, inny komputer został sprzedany w tylu egzemplarzach - był to VIC-20. Jeśli chodzi o wielkość oprogramowania - tylko Apple II i CP/M mogą w tym względzie konkurować z C64.

#### Założenia budowy C64

Firma Commodore postawiła sobie wiele różnych wymagań konstruując komputer C64 i one zdecydowały o ostatecznym kształcie i powodzeniu C64. Commodore tak dobrała cechy C64, aby był on jak najbardziej atrakcyjny na rynkach komputerów domowych i szkolnych. Do tego momentu firma odnotowała sukcesy na obu rynkach - VIC 20 królował jako komputer domowy, a PET - szkolny. Commodore postanowił połączyć najwarteściowsze cechy VIC i PET, konstruując komputer mogący odnosić sukcesy na obu rynkach. Ale nowy C64 powinien także zawierać cały szereg nowych możliwości, aby przyciągnąć klientów. Także urządzenia peryferyjne i oprogramowanie do tego komputera nie powinno być drogie. Oznaczało to, że firma musiała wykorzystać ostatnie osiągnięcia w dziedzinie inżynierii sprzętowej i oprogramowania, aby C64 kosztował mniej niż sprzęt konkurentów i działał lepiej.

Oto lista cech, które Commodore 64 przejął od VIC-20 i PET:

- Architektura bazująca na mikroprocesorze 6502
- System operacyjny "Kernal"
- BASIC 2.0
- Klawiatura podobna do maszyny do pisania
- 40-kolumnowy ekran tekstowy
- Grafika PET
- Interfejs do magnetofonu kasetowego
- Szyna szeregową dla stacji dysków i drukarek
- Port użytkownika dla urządzeń peryferyjnych typu modem i syntezytor dźwiękowy

Oto lista nowych cech C64:

- Kolorowa grafika
- Grafika wykorzystująca "sprites"
- Trzykanałowy, 8-oktawowy syntezytor dźwięku
- Możliwość dostosowania do pracy pod systemem CP/M
- Ulepszone gniazdo cartridge'a
- Bezpośrednie wyjścia wizji i dźwięku wysokiej jakości
- Modulator RF
- 64K pamięci RAM z większymi możliwościami gospodarowania nią
- 2 gniazda sterownicze dla joysticków, pióro świetlne etc.
- Nowa rodzina niedrogich urządzeń peryferyjnych - monitor kolorowy, stacja dysków, drukarki i modemy.

Jak możesz się zorientować na podstawie powyższych list, Commodore zaopatrzyła C64 w zestaw istotnych cech. Firma wprowadziła komputer na rynek w 1982 r. /cena \$ 595/. W roku 1983 cena spadła do \$ 300, a w połowie roku 1985 - do \$ 150.



Jakich urządzeń zewnętrznych potrzebujesz do pracy  
w trybie C64

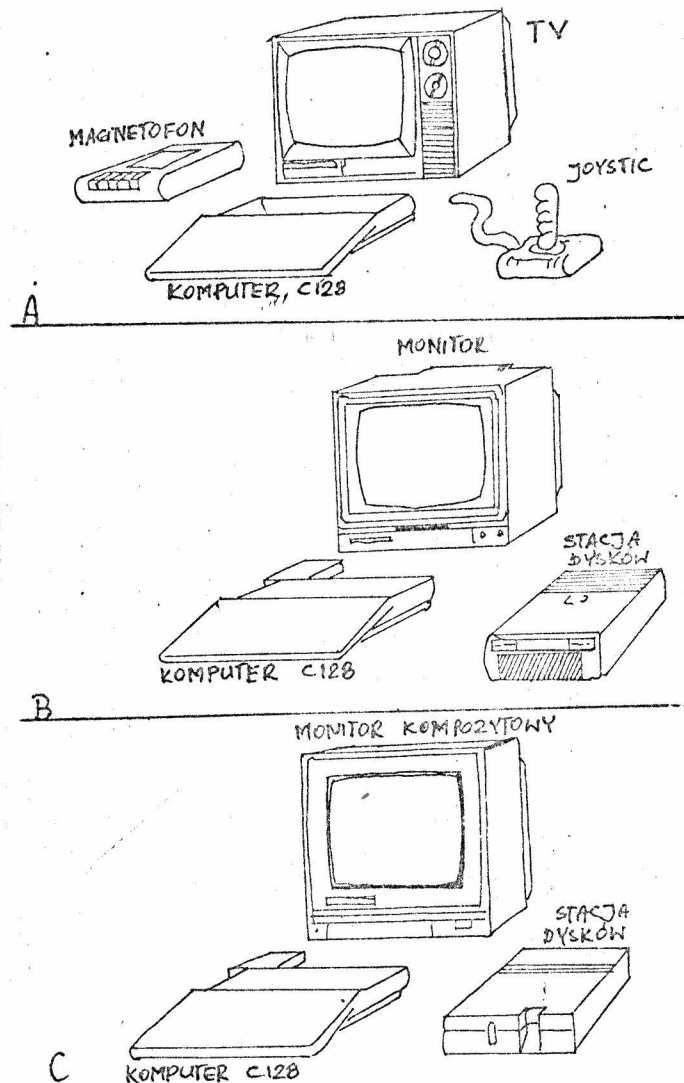
Do pracy w trybie C64 będziesz potrzebował przynajmniej monitora /lub telewizora/ mogącego pracować w trybie 40-kolumnowym. Rodzaj monitora, który wybierzesz zależy od tego, co będziesz chciał robić, gdy Twój C128 będzie pracował w trybie C64. W trybie tym możesz wybierać z takiego samego zestawu akcesoriów, jak właściciele C64. Możesz także wykorzystać urządzenia peryferyjne nowego Commodore'a, jak monitor kolorowy 1902 oraz stacje dysków 1571 i 1572 /skonstruowane specjalnie dla C128 i opisane w rozdziale 2/, ponieważ one również mają możliwość pracy w trybie C64.

Naturalnie rodzaj sprzętu jakiego będziesz potrzebował zależeć będzie od użytego oprogramowania. Jeżeli będziesz chciał wykorzystać tryb C64 dla uruchomienia gier zapisanych w cartridge'ach i programów edukacyjnych /stworzonych dla C64/ będziesz potrzebował tylko monitora lub telewizora i ewentualnie joysticka lub zestawu wiosełek /paddles/. Jeśli natomiast będziesz chciał korzystać z programów C64 na kasetach, będziesz potrzebował Commodore Datasette. Jeśli będziesz chciał użyć bardziej rozbudowanego oprogramowania, opartego na dyskietkach, będziesz potrzebował jednej lub więcej stacji dysków. To, czy będziesz potrzebował innych urządzeń, takich jak np. drukarka czy modem, należy od potrzeb Twoich programów na C64. Np. jeśli planujesz wykorzystać swój komputer do przetwarzania tekstów, będziesz potrzebował drukarki, lecz jeśli interesują Cię programy o charakterze gier - nie będzie Ci ona potrzebna. Modem z kolei będziesz potrzebował, jeśli zamierzasz korzystać za pomocą komputera z ogólnie dostępnego systemu komputerowego, takiego jak CompuServe lub The Source\*.

Jak więc widzisz, nie jest łatwo dobrać urządzenia zewnętrzne dla C128, pracującego w trybie C64. Trzy z możliwych

\* W USA.

konfiguracji sprzętowych pokazano na rys.15.



Rys.15 - Trzy przykładowe konfiguracje sprzętu  
do pracy z komputerem C128 w trybie C64

Pierwszy z wymienionych przykładów stanowi prawdziwie minimalny system przeznaczony do programowania lub reklamy. W tym systemie telewizor zastępuje monitor, Datasette stanowi pamięć masową, a pojedynczy joystick - urządzenie wejściowe. Drugi przykład ilustruje, jak właściciel C64, posiadający pełny sprzęt C64 może go wykorzystać, zamieniając C64 na C128. Commodore 1702 użyty jest jako monitor, stacja dysków 1541 jako pamięć masowa, a MPS-801 stanowi drukarka systemu. I wreszcie ostatni przykład pokazuje wykorzystanie do pracy w trybie C64 nowych urządzeń C128 takich jak: stacja dysków 1571, monitor 1902, drukarka MPS-82 i modem 1670.

Naturalnie możliwe są inne konfiguracje niż te wymienione, ale zawsze pamiętaj o wybieraniu urządzeń zewnętrznych w zależności od potrzeb oprogramowania Twojego C128, pracującego w trybie C64. /Skorzystaj z dołączonych do oprogramowania instrukcji, zawierających zazwyczaj, oprócz samego opisu programu, także wykaz niezbędnego sprzętu/. Jeśli nadal nie jesteś zdecydowany jakie urządzenia peryferyjne wybrać, wróć do rozdziału 2.

#### Jak i kiedy możesz przejść do trybu C64?

Aby załadować program napisany na komputer C64, musisz najpierw przełączyć komputer na tryb pracy C64. Są dwie metody przejścia do tego trybu. Jeśli komputer pracuje w trybie C128 można użyć rozkazu "GO 64". Druga metoda polega na włączeniu zasilania C128 po uprzednim umieszczeniu w odpowiednim gnieździe dowolnego cartridge'a z programem napisanym na C64. W obu przypadkach, jeśli komputer został przełączony na tryb C64, nie ma możliwości programowego powrotu do trybu C128 /Commodore 64 nie posiada rozkazu GO 128, a więc aby spełnić wymóg całkowitej kompatybilności, nie posiada go również C128 w trybie C64/. Jedynym sposobem powrotu do trybu C128 jest wyłączenie i włączenie zasilania komputera.

#### Co można zrobić w trybie C64?

Jedną z głównych przesłanek użycia trybu C64 jest istnienie dużej liczby wysokiej klasy programów na komputer Commodore 64, które mogą zostać użyte bez jakichkolwiek modyfikacji w trybie C64. Istnieją różnorodne programy do wykorzystania w interesach, domu czy szkole.

W tej części zamierzamy omówić różne rodzaje oprogramowania dostępnego na Commodore C64. Podzielimy ten świat oprogramowania na 5 kategorii: oprogramowanie osobiste, zabawa, rozwój oprogramowania, edukacja oraz osobna grupa programów łączących zabawę z edukacją /"edutainment"/.

Prosimy zwrócić uwagę, że niemożliwe jest wymienienie wszystkich programów, składających się na te 5 kategorii. Gdybyśmy chcieli to zrobić, musieli byśmy napisać osobną książkę /i to bardzo grubą/. A więc ta część tego rozdziału nie będzie przewodnikiem po oprogramowaniu, ale raczej "rzutem oka" na rodzaje dostępnych programów, z krótkim opisem popularnych programów reprezentujących każdą z 5 kategorii lub programów o interesujących i niezwykłych cechach, jako przykładów dostępnego oprogramowania. Istnieje już bardzo bogate oprogramowanie C64, a cały czas tworzone są nowe programy, tak więc powinieneś sam zorientować się w temacie przed zdecydowaniem się na zakup programów.

Nie będziemy podawać cen omawianych tu programów, a to dlatego, że zmieniają się one bardzo szybko i zależą od dostawcy. Generalnie jednak oprogramowanie osobiste i narzędziowe kosztuje od \$ 50 do \$ 150, podczas gdy gry i oprogramowanie rozrywkowe oraz edukacyjne - od \$ 20 do \$ 50. Ceny te są stosunkowo niższe niż analogiczne oprogramowanie przeznaczane na bardziej kosztowne komputery.

### Oprogramowanie osobiste

Przez oprogramowanie osobiste rozumiemy programy, które pomagają zrealizować różnorodne zadania w domu i pracy. Zalicza się do nich: programy przetwarzania tekstów, kalkulatory<sup>\*</sup> na tabelkach /płachty/, bazy danych, programy finansowe i telekomunikacyjne.

### Programy do przetwarzania tekstów

Przetwarzanie tekstów jest najpopularniejszym /spośród tych "poważnych"/ zastosowaniem komputerów osobistych, takich jak Commodore 64 i Commodore 128. Programy tego typu pozwalają na pisanie dokumentów /tzn. wszystkiego tego co można pisać na maszynie do pisania - od pamiętników do powieści/ na ekranie komputera. Ekran może być w nich "oknem" pokazującym fragment znacznie obszerniejszych dokumentów. Możesz dowolnie modyfikować tekst przez dodawanie lub kasowanie słów lub zdań. Gdy osiągnie on już żadaną postać, możesz go wydrukować za pomocą drukarki.

Programy obróbki tekstów zawierają zazwyczaj rozkazy dodawania i kasowania poszczególnych liter, zdań i paragrafów, przepisywania dokumentów na dyskietkę i na odwrot, ustawiania marginesów i akapitów, automatycznego numerowania stron i drukowania tekstów za pomocą drukarki. Większość programów do przetwarzania tekstów posiada wiele innych możliwości, takich jak wyszukiwanie danego słowa czy zdania w danym tekście, podkreślanie słów i zdań oraz projektowanie szaty graficznej gotowego tekstu w dowolny sposób. Większość z tych programów używa automatycznego przewijania wyrazu. Oznacza to, że kiedy dojdiesz do końca linii, program automatycznie przesunął kursor do nowej linii, a więc możesz kontynuować pisanie bez potrzeby używania Return lub innego klawisza.

Bardzo użyteczną cechą wielu takich programów jest możliwość łączenia części jednego tekstu z drugim. Jest to często używane w przypadku listów, gdy jeden dokument zawiera listę nazwisk i adresów, a drugi zawiera listy, które należy wysłać osobom wymienionym na liście. Istnieje wiele bardzo dobrych programów do obróbki tekstów na Commodore 64. Bank Street Writer znany z łatwości obsługi, szczególnie popularny jest wśród młodzieży szkolnej, która rzadko używa bardziej złożonych procesorów słownych. Inny program tego typu, Speed Script rozpowszechniany przez COMPUTE! Publications, Inc. ma większe możliwości niż BSW. Jest to wspaniały program, szczególnie przy niezbyt częstym korzystaniu z tego typu oprogramowania, a to ze względu na jego cenę - równą jednemu czasopiśmie COMPUTE!. Jeśli jednak zapragnąbyś silniejszego programu, to masz obszerną listę profesjonalnych produktów, z której możesz dokonać wyboru. Oto tylko 2 przykłady: Easy Script /firmy Commodore/ oraz WordPro 3 Plus/64. Przyjrzyjmy się jednemu z tych najbardziej dopracowanych i kosztownych procesorów tekstowych. Word Pro 3 Plus/64 /firmy Profesjonal Software/ posiada 23 rozkazy i 47 funkcji sterujących, które wykonują dużą liczbę podstawowych operacji przetwarzania. Oprócz tych podstawowych operacji wykonywanych także przez mniejsze programy, jak Speedscript, Word Pro, może wykonywać takie działania jak drukowanie części i super dużych tekstów, dodawanie kolumn liczbowych do tekstu oraz używanie pisma tłustego. Word Pro posiada także specjalny słownik /umożliwiający sprawdzenie pisowni wyrazów/<sup>\*</sup> - bardzo przydatną cechę, którą posiadają wszystkie lepsze programy przetwarzania tekstów. Word Pro jest złożonym programem, ale jak większość programów wysokiej jakości posiada dobrze opracowany opis obsługi. Na dyskietce umieszczone są także przykładowe dokumenty ułatwiające poznanie możliwości programu.

---

<sup>\*</sup> Oczywiście w języku angielskim

## Programy kalkulacyjne /Spreadsheet - płachta/

Pierwszy program tego typu, Visi Calc, przyczynił się walcnie do wykorzystania komputerów osobistych w interesach. Dzięki temu programowi sprzedano miliony komputerów dla amerykańskiego biznesu i zmieniono poglądy ludzi na komputery osobiste. Te same możliwości posiada C64.

Co to jest program spreadsheet? Jest to elektroniczny odpowiednik tabel używanych w księgowości: kartka papieru z danymi liczbowymi na dole i graficznym ich przedstawieniem prowadzącym się do góry, ale elektroniczne płachty automatycznie wykonują wszystkie działania, których potrzebujesz. Są one wykorzystywane do różnych celów. Np. możesz za ich pomocą przedstawić graficznie ilość sprzedawanych towarów w różnych latach. Na dole każdej kolumny może być pokazana suma sprzedanych produktów w danym roku, a na osi poziomej /ponad numerem roku/ procentowy wzrost lub spadek w sprzedaży tego produktu. Raz skonstruowany wzór płachty może być wykorzystywany wielokrotnie do różnych celów /ewentualnie trochę modyfikowany w zależności od potrzeby/.

Microsoft MultiPlan jest jednym z wielu programów kalkulacyjnych przeznaczonych dla Commodore 64. Jest on o tyle atrakcyjny, że można go użyć na takich komputerach jak IBM PC i Apple Macintosh. Wersja na C64 posiada wiele tych samych możliwości. Multiplan posiada także obszerny i przystępny opis dla początkujących.

Calc Result /firmy Handic Software/ jest nieco ambitniejszym programem. Jest łatwy w obsłudze i zawiera szereg profesjonalnych cech. Istnieje on w 2 wersjach: Calc Result Easy daje Ci możliwość wykorzystania 1000 komórek, podczas gdy Calc Result Advanced- 2000 oraz wiele dodatkowych możliwości. Obie wersje są bardzo łatwe w obsłudze. PractiCalc /firmy Computer Software Associates/ jest nieco mniejszym i prostszym programem tego typu. Główną jego zaletą jest to, że będąc bardzo

prostym programem pozostawia dużo wolnego miejsca w pamięci dla przechowywania aktualnych płacht. Jest to poza tym jeden z najtańszych programów tego typu, polecanym szczególnie początkującym.

## Bazy danych

Programy bazy danych /takie jak "The Manager" f-my Commodore/ znajdują szerokie zastosowanie wśród użytkowników C64. Iza danych jest to jakby elektroniczny segregator, który zapewniasz własnymi informacjami. Mogą to być dane o codziennych wydatkach, kolekcjach filatelistycznych, ulubione przepisy, wycinki książek lub czasopism, adresy rodziny, znajomych, klientów. Poza tym możesz użyć komputera do sortowania, wyszukiwania i uaktualniania tych danych. W niektórych przypadkach bazy danych, podobnie jak płachty kalkulacyjne, wymagają pewnej pracy początkowej. Ponadto musisz określić formę w jakiej mają być prezentowane informacje. Na szczęście takie programy jak np. Manager zawierają kilka wzorów form oraz zestaw prostych instrukcji do ich projektowania.

## Finanse osobiste

Ponieważ Commodore 64 był przede wszystkim przeznaczony na rynek komputerów domowych, nie powinno nikogo dziwić, że istnieje bogate oprogramowanie związane z prywatnymi finansami. Przykładami takich właśnie programów są Personal Accountant i Home Accountant /firmy Continental Software/. Pozwalają one na wykonanie takich operacji jak obliczenie bilansu książeczki czekowej, wypisywanie czeków, prowadzenie budżetu domowego i uzyskiwanie bieżących informacji o stanie finansowym. Tablice, zestawienia i przystępne wykresy graficzne używane są tutaj do ukazania zmian podatków i dokonywania porównań. Możliwe jest również pobranie danych finansowych prze-

chowywanych przez te programy i przesłanie ich do programu porządkującego podatki w końcu roku.

#### Programy telekomunikacyjne

Setki tysięcy właścicieli Commodore 64 używają swoich C64 do komunikowania się z innymi komputerami za pomocą modemów i oprogramowania tychże modemów. Programy "telekomunikacyjne", jak je się powszechnie nazywa, istnieją w bardzo wielu odmianach, ale w gruncie rzeczy są one bardzo podobne do siebie, jeśli chodzi o wykonywane operacje. Główną ich funkcją jest sterowanie wysyłaniem i odbieraniem informacji od modemu /który stanowi jedynie fizyczne połączenie C64 z linią telefoniczną/.

Najprostsze programy komunikacyjne zmieniają Twój C64 w końcówkę do przekazywania informacji. Urządzenie takie spełnia 2 funkcje: wysyła wprowadzone z klawiatury dane do modemu oraz wyświetla na ekranie dane z niego otrzymane. Kończąc tak /tzn. C64 w takim trybie pracy/ nie umożliwia przechowywania przychodzących informacji, ani redagowania wiadomości przed ich wysłaniem, a więc ich wykorzystanie jest w zasadzie ograniczone do sporadycznych łączności. Ten rodzaj oprogramowania telekomunikacyjnego zazwyczaj otrzymuje się "gratis" przy zakupie modemu np. 1660 lub 1670 firmy Commodore.

Jeśli Twoje potrzeby w tej dziedzinie są nieco większe, to będziesz potrzebował programów typu Super Term firmy Midwest Micro, Inc., lub Smart 64 Terminal Plus 3 firmy Micro Technics Solutions Corp., które posiadają większe możliwości telekomunikacyjne: magazynowanie przychodzących danych oraz możliwość zredagowania informacji przed wysłaniem. Pozwalają także na drukowanie przychodzących danych. Programy tego typu mogą także automatycznie wybierać numer telefoniczny, przysyłać zbiory binarne korzystając ze specjalnego protokołu zabezpieczającego informację przed zniszczeniem podczas pro-

cesu przesyłania. Podsumowując, programy telekomunikacyjne tego typu posiadają duże możliwości, ale są trudne w użyciu.

Jeżeli Twoje potrzeby telekomunikacyjne ograniczają się do korzystania z CompuServe, to będziesz potrzebował programu VIDTEX. Program ten wykonuje wszystkie opisane poprzednio operacje, a dodatkowo steruje interfejsem, łączącym CompuServe z Tobą. Interfejs ma jeden główny cel - ułatwić komunikację z CompuServe, a realizuje go wieloma metodami. Np. za pomocą VIDTEX Twój C64 automatycznie rozumie specjalny kod ekranowy, który CompuServe wysyła, aby wyczyścić ekran i przesunąć kursor. Dzięki temu Twój ekran jest "czystszy" i łatwiej na nim odczytywać informacje. Te kody ekranowe pozwalają na szybsze przesyłanie informacji z CompuServe, dzięki czemu zaoszczędzisz trochę pieniędzy. VIDTEX pozwala także na łatwiejsze przesyłanie zbiorów w kodzie ASCII i zbiorów binarnych. Program ten ma też wiele innych cech, które ułatwiają współpracę z CompuServe.

#### Programy edukacyjne

Drugą, najobszerniejszą grupę oprogramowania C64 po grach i zabawach, stanowi oprogramowanie edukacyjne. W grupie tej można wydzielić kilka podgrup jak np. przedszkolna, dzieci w wieku 5-9 lat, 9-13 lat, 13-17 lat i dorośli. Programów tych używa się do nauki przeróżnych przedmiotów szkolnych, od nauki podstaw języka począwszy /ręczenniki, czasowniki i przymiotniki/, do geografii, chemii i matematyki.

Ponad sto programów edukacyjnych już znajduje się w wypożyczalniach publicznych, a jeszcze większa ilość może być udostępniana bezpłatnie przez większość większych klubów użytkowników Commodore lub dostawców tych komputerów. Wydruki sepek programów tego typu można również znaleźć w czasopiśmie

przeznaczonych dla użytkowników Commodore, jak np. "RUN", "COMPUTE!"s Gazette", "Commodore Power Play" czy "Commodore Microcomputers". Istnieje też wiele książek, zawierających programy edukacyjne na C64. Jednakże wysokiej klasy oprogramowanie tego typu trzeba kupować. Przykładami mogą być programy SPELLIT!, MATCH, BLASTER!, WORD ATTACK! i SPEED READER firmy Davidson Associates. Istnieją także programy przeznaczone dla wyższych uczelni, jak THE PERFECT SCORE firmy Mindscape lub specjalne programy przygotowujące kandydatów na studia wyższe. Można także nabyć programy do samokształcenia, jak np. Timeworks' Evelyn Wood Speed Reddling, który służy powiększaniu biegłości w czytaniu i zdolności zapamiętywania.

#### Gry edukacyjne

Gry edukacyjne /edutainment/ jest to nowe określenie, które oznacza całkowicie nową kategorię oprogramowania C64. Zawiera ona programy zbudowane tak, aby nauczać danego przedmiotu jednocześnie bawiąc, słowem - edukacja i rozrywka. Wiele programów spełnia te warunki tak dobrze, że trudno jest się czasem zorientować, że właśnie korzystamy z takiego programu - zbyt dobrze się nim bawimy. Gry edukacyjne przeznaczone są dla wszystkich grup wiekowych. Wspaniałym programem dla przedшкоłaków jest Icorning With Leeper firmy Sierra On-line. Program ten służy rozwijaniu podstawowych umiejętności takich, jak koordynacja kończyn i wzroku, rozpoznawania liter, liczb, identyfikacji przedmiotów, liczenia, rozwiązywania problemów i wrażliwości artystycznej. Dla nastolatków - program Type Attack firmy Serious Software. Ten niezwykły program uczy pisania na klawiaturze w trakcie quizu, w którym liczy się czas odpowiedzi. Nowym programem w omawianej grupie jest Trivia Fever firmy PSI. Ten nowy produkt oferuje tysiące wyzywająco prostych pytań z siedmiu dziedzin na 3 poziomach trudności. Gry edukacyjne nie zatrzymują się tutaj, bowiem istnieją programy, których przeznaczeniem jest uczenie wielu rozmaitych

rzeczy, takich jak np. przetwarzanie tekstów, programowanie czy posługiwanie się płachtami kalkulacyjnymi.

#### Programy rozrywkowe

Jest to najobszerniejsza kategoria oprogramowania dostępnego na C64 /ok. 60-70% programów napisanych na C64/. W kategorii tej można wyróżnić szereg mniejszych grup: proste gry /tzw. arcade games/ stanowią najliczniejszą spośród tych grup. Drugą pod względem liczebności tworzą gry przygodowe /adventure/. Poza tym można jeszcze wyróżnić gry symulacyjne oraz gry zaliczane do obszaru artystycznej i muzycznej rekreacji.

Wymienione tu kategorie reprezentowane są przez bogaty asortyment programów. Oczywiście większość z nich jest napisana przez profesjonalistów i rozprowadzana w postaci nagranych kaset lub dyskietek, ale wiele również można znaleźć w książkach, czasopiśmie oraz otrzymać od lokalnej grupy użytkowników Commodore'a. Nie jesteśmy oczywiście w stanie opisać i wymienić wszystkich programów tego typu, ale postaramy się krótko wymienić najpopularniejsze.

#### Proste gry

Takie gry jak Pitstop II, Beach Head i Impossible Mission służą pobudzeniu Twojego wzroku, słuchu czy dotyku w fascynujący i czarujący sposób. Pitstop II na przykład umieszcza Cię za kierownicą samochodu wyścigowego i pozwala Ci wziąć udział w różnych międzynarodowych wyścigach. W grze tej jesteś kierowcą obserwującym wietrzną i krętą trasę przez szybę samochodu.

Beach Head jest grą wojenną, w której masz wiele zadań do wykonania - takich, jak odparcie ataku kamikaze - przed wysadzeniem swoich komandosów na silnie ufortyfikowanej



plaży, bronionej przez nieprzyjaciela i próbą jej zdobycia.

W Impossible Mission przeciwnikiem jest zły naukowiec. Twoim zadaniem jest odpieranie ataków robotów oraz omijanie pułapek czekających w podziemnym labiryncie, pełnym tuneli i pokojów, a także gromadzenie jak największej ilości wskazówek, aby móc unieruchomić diabelskie urządzenie naukowca.

Te proste gry mają zazwyczaj nieskomplikowane reguły oraz prosty scenariusz z wyraźnie określonym wzorcem przeszkód i nagród. Kluczowym elementem tych programów jest współzawodnictwo. Zdobywanie punktów, przemijanie czasu i gromadzenie goli służy pokazaniu poczynionych postępów. Możesz współzawodniczyć z tym samym przeciwnikiem co wczoraj lub z kimś zupełnie innym. Gry te ćwiczą wzrok, słuch, dłoń i umysł, wspaniale wyrabiają zręczność, bystrość i koncentrację.

#### Gry przygodowe

Natura tych gier jest zupełnie odmienna od poprzednich. W przeciwieństwie do nieskomplikowanych, wielokrotnie powtarzających się elementów w poprzednich grach, gry przygodowe rozwijają odwagę i wyobraźnię. Wyrabiają umiejętność rozwiązywania różnorodnych problemów i zagadek. Gry te nie wymagają szybkości, ich celem nie jest wyrabianie zręczności. Są one raczej grami umysłowymi, których celem jest ćwiczenie umiejętności logicznego myślenia i dedukcji w zupełnie inny sposób, niż robią to gry opisane poprzednio. Dlatego też wymagają one koncentracji i dużej ilości czasu. Scenariusz jest bardzo istotną częścią tych gier. Przede wszystkim musi on być łatwy do zrozumienia, ale musi posiadać również uwikłaną wewnętrzną strukturę z dużą ilością możliwych zdarzeń. Przebieg gry zależy od poczyniań gracza. Takie gry przygodowe, jak ZORK I, ZORK II, czy ZORK III zabierają Cię do kompletnie wymyślnego świata science-fiction, gdzie musisz walczyć z różnego rodzaju stworami oraz poszukiwać ukrytych skarbów.

Istnieją także gry, które przenoszą Cię głęboko w świat science fiction, gdzie musisz nauczyć się przetrwać w odległych zakątkach przestrzeni kosmicznej /"SUSPENDED"/. Powstały również dwie gry przygodowe, bazujące na opowiadaniach Agaty Christie. Dzięki nim zamieniasz się w prawdziwego detektywa /DEAD-LINE/ i /EYE WITNESS/. Wszystkie wymienione tu programy są dziełem firmy Inforcom.

#### Programy symulacyjne

Następną kategorią gier są gry symulacyjne. Programy te wykorzystują w pełni możliwości obliczeniowe, graficzne i dźwiękowe C64 do tworzenia bardzo złożonych gier symulujących rzeczywistość - jak np. symulacja lotu samolotem. Nie jest to jednak to samo co w omawianych wcześniej prostych grach zręcznościowych, gdzie sterując małym samolotkiem możesz unosić się, opadać, skręcać i lądować za pomocą prostych ruchów joysticka, ale symulacja prawdziwego lotu, gdy siedzisz w kokpicie otoczony przez realistycznie wyglądające przyrządy. Przed sobą i po bokach widzisz niebo oraz ląd. Podczas lotu Twojej maszyny widok zmienia się w zależności od manewrów wykonywanych przez samolot. Istnieje tylko jedna metoda, aby wygrać w takiej grze: trzeba posiadać pewną wiedzę o kierowaniu prawdziwym samolotem.

Istnieje wiele gier tego typu na Commodore 64. Flight Simulator firmy subLOGIC Corporation oraz Solo Flight firmy MicroProse Software symulują lot samolotem. Subwar 64 firmy Clockwork Computers, Inc. imituje obraz widziany z wieżyczki kontrolnej łodzi podwodnej. W programie tym możesz pełnić różnorakie funkcje. Możesz być nawigatorem, oficerem dyżurnym lub oficerem kierującym atakiem torpedowym. Każda z tych postaci wymaga posiadania innych prądyspozycji.



### Artystyczna i muzyczna rekreacja

Ostatnią grupę programów rozrywkowych na C64 stanowią programy umożliwiające spokojny, aktywny relaks /bez współzawodnictwa/. Pozwalają one wykorzystać C64 do rozwijania swojego talentu. Możesz malować ludzi lub krajobrazy za pomocą takich programów jak Koalapad, możesz tworzyć i animować grafikę komputerową za pomocą programu Sprite Maker, słuchać muzyki komputerowej, tworzyć melodie i komponować większe utwory za pomocą takich programów jak MusiCalc i Kawasaki Synthesizer. Oczywiście są to tylko przykłady wybrane spośród dziesiątków istniejących programów artystycznych i muzycznych, napisanych na C64. Jeśli Twoje cechy charakteru skłaniają Cię ku tego rodzaju formom rozrywki, to będziesz miał możliwość korzystania z wielu niezwykłych programów tego typu.

### Rozszerzenia programowe

Istnieje wiele programów na Commodore 64, które pomogą Ci przekształcać, pisać i testować własne programy. Niektóre z nich dodają do wbudowanego BASIC-a 2.0 kilka rozkazów, podczas gdy inne, np. potrafią całkowicie zastąpić BASIC nowym językiem programowania. Wiele z tych programów tworzy zupełnie nowe systemy operacyjne, inne są programami pomocniczymi, przeznaczonymi do wykonywania pojedynczych /lecz niezwykle przydatnych/ funkcji, a jeszcze inne stanowią dodatki do istniejącego systemu. Kilka spośród nich ma na celu zredukowanie wysiłku wkładanego w wykonywanie niektórych nudnych, lecz niezbędnych operacji związanych z programowaniem i obsługą komputera.

### Rozszerzenia systemowe

Jeśli życzylibyś sobie użyć innego systemu niż BASIC 2.0 to istnieje bogate oprogramowanie, które Ci to umożliwi. Dla tych, którzy zamierzają pisać wyjątkowo szybkie programy, istnieją różnorodne programy umożliwiające pracę w języku assemblera. Np. Commodore 64 posiada własny pakiet, przeznaczonych do tego celu programów o nazwie Assembler 64. Zawiera on pełny makroassembler, program edytora, program ładujący i 2 programy monitora języka maszynowego. Jeśli chciałbyś, zachowując korzyści jakie daje programowanie w języku assemblera, mieć dostęp do szeregu ułatwień charakterystycznych dla języków programowania wyższego rzędu, to prawdopodobnie najlepszy okaże się interpreter języka Forth taki, jak Super Forth 64 firmy Parsec Research. Ta implementacja języka Forth zawiera pełną arytmetykę zmiennoprzecinkową i pozwala na uruchomienie programów wynikowych, zupełnie niezależnie od systemu /programy mogą być uruchamiane bez FORTH/. Jednakże, jeśli nie zależy Ci na szybkości i prostocie jaką daje Forth, to możesz skorzystać z bardziej złożonych języków oprogramowania, jak C, Pascal czy Fortran. Dobre implementacje wymienionych języków produkuje firma Abacus Software. Możesz wybrać także, jako narzędzia pracy, języki Logo czy Pilot. Obydwa są dostępne na Commodore 64.

### Programy pomocnicze

Istnieje wiele programów na C64, których zadaniem jest pomóc w procesie pisania programów. Np. mogą to być specjalne edytory ekranowe jak Screen-Graphics - 64 firmy Abacus Software, który pozwala na tworzenie obrazów tekstowych i graficznych za pomocą rysika ukazującego się na ekranie. Istnieją edytory czcionek, umożliwiające predefiniowanie standardowego zestawu znaków, takie jak np. Ultra Font + firmy COMPUTE! Publications, Inc. Istnieją też edytory

sprite'ów, które można wykorzystać do tworzenia, animacji, przechowywania informacji o sprite'ach w formie, odpowiedniej do wykorzystania w programie. Istnieją też programy monitora języka maszynowego analogiczne do tego, który dostępny jest w trybie pracy C128. Można je wykorzystać do pisania własnych, niedużych programów w języku maszynowym lub tworzyć krótkie i szybkie podprogramy dostępne z poziomu języka BASIC.

Przykładami omawianych programów monitora są Micromon i Supermon, które są dostępne bezpłatnie /wersja Micromon publikowana była w COMPUTE!'s First Book of Commodore 64/.

#### Programy rozszerzające BASIC

Programy te stanowią następną kategorię oprogramowania na C64. Ich celem jest zwiększenie "siły" igły BASIC-a 2.0. Dokonuje się tego przez dodanie szeregu nowych rozkazów graficznych, dźwiękowych i operowania sprite'ami oraz przez usprawnianie przebiegu operacji wykonywanych pod kontrolą starszego DOS BASIC-a 2.0. Użytkownik C64 ma do dyspozycji wiele różnorodnych pakietów realizujących te cele. Firma Skyles Electric Works oferuje cartridge pod nazwą VIC Tree, który dodaje 40 nowych rozkazów do BASIC-a 2.0 - w tym ulepszone rozkazy DOS i rozkazy pomocnicze, jak AUTO, RENUMBER czy TRACE. Jeszcze większe możliwości daje Simon's BASIC firmy COMMODORE. Natomiast cartridge tej samej firmy - Super Expander daje te same możliwości graficzne i dźwiękowe co BASIC 7.0.

Następną ofertę w tej dziedzinie stanowią kompilatory BASIC-a. Kompilatory, takie jak BLITZ! firmy Skyles Electric Works czy BASIC Compiler-64 firmy Abacus Software umożliwiają jednokrotną translację programów w BASIC-u na język maszynowy. Skompilowane programy, choć fizycznie znacznie cięższe niż przed kompilacją, działają tak szybko, jak analogiczne

programy pisane w PASCAL-u, językach PILOT czy LOGO.

#### Programy ułatwiające programowanie

Ostatnią kategorię oprogramowania na C64 stanowią programy usuwające pewne uciążliwości towarzyszące programowaniu w BASIC-u czy Asemblerze 6502. Programy te nie zmieniają ani nie modyfikują końcowych wersji pisanych programów, ale po prostu ułatwiają ich pisanie.

Dwa najbardziej popularne programy tego rodzaju na C64 to COMPUTE!'s Automatic Proofreader do programów w BASIC-u oraz COMPUTE!'s HIX do programów w języku maszynowym, opracowane w celu ułatwienia użytkownikom C64 wprowadzania programów publikowanych w czasopiśmie COMPUTE! Publication. Programy te umożliwiają użytkownikom C64 korzystanie z setek przetestowanych programów w BASIC-u i w języku maszynowym, nawet bez jakiegokolwiek znajomości tych języków programowania. Programy ułatwiające programowanie podobne do tych dwóch, znaleźć można w innych publikacjach Commodore'a. Innym programem tego typu, dostępnym w COMPUTE! Publications, jest Meta BASIC. Program ten wykonuje podobne funkcje jak pakiety rozszerzające BASIC /opisane poprzednio/, ale w przeciwieństwie do nich, Meta BASIC nie dodaje żadnych operacji do zestawu instrukcji BASIC-a. A więc programy pisane przy pomocy Meta BASIC mogą być uruchamiane bez tego programu. Meta BASIC dodaje jedynie rozkazy, ułatwiające pracę takie jak: AUTO, RENUM, TRACE, DUMP, CAT, ERR, DELETE, MERGE i LLIST.

#### Co możesz robić w trybie C64?

Pozdriak 3 opisywał wiele cech Commodore 128 zwłaszcza tych, stanowiących rozszerzenia w stosunku do cech, które

rymi dysponuje Commodore 64. Jednakże C128 użyty w trybie C64 nie korzysta z tych rozszerzeń. Po pierwsze, dostępne jest tylko 64 K pamięci RAM, ponieważ C64 potrafi "zagospodarować" tylko te 64 K /pozostałe 64 K pamięci jest w trybie C64 wyłączane/. Jakakolwiek zewnętrzna RAM - np. symulująca dyskietkę - pozostanie nieznana, a więc nieużyteczna. Pamięć ROM zostaje także okrojona w tym trybie pracy. System operacyjny C128, z wbudowanym programem monitora w języku maszynowym i ulepszonym edytorem ekranowym, zastąpiony zostaje przez system operacyjny używany w C64 /bez monitora i wspianego edytora ekranowego/. Załączenie tego trybu zmienia także konfigurację pamięci i sprzętu, co między innymi zmienia możliwości WE/WY. Na przykład użytkownik nie może już korzystać z bloku 14 klawiszy funkcyjnych, czy też jakichkolwiek innych, właściwych tylko trybowi C128. Zmiana WE/WY eliminuje także 80-kolumnowy tryb ekranowy. I ostatnia, istotna zmiana zarówno w układach WE/WY C128, jak i wewnętrznym systemie operacyjnym, uniemożliwiają korzystanie z większych możliwości przechowywania informacji i szybszej ich obsługi przez nowsze stacje dysków 1571 i 1572.

#### Czym różni się BASIC w trybie C64?

Jedną z konsekwencji zmiany trybu Commodore 128 na tryb C64 jest zastąpienie BASIC-a 7.0 BASIC-iem 2.0, tym samym, który jest użyty w Commodore 64. Dlatego też komputer w tym trybie nie rozumie już nowych rozkazów, instrukcji i funkcji właściwych tylko BASIC-owi 7.0. Dostępne rozkazy wymienione zostały w tabl.4-1.

#### Kompatybilność BASIC-a 2.0 i 7.0

Jak pewnie zorientowałeś się z tabl.4-1 BASIC 2.0 nie posiada specjalnych rozkazów graficznych, dźwiękowych i operowania sprite'ami, a także rozkazów, takich jak PRINT USING,

AUTO czy WHILE ... DO. Istnieje duża liczba podobnych operacji dostępnych w BASIC-u 7.0, do których nie masz dostępu w trybie C64. Jeśli zamierzasz pisać programy w BASIC-u w tym trybie, lub modyfikować już napisane, powinieneś zaznać się z różnicami między obydwoa BASIC-ami. Istnieje możliwość pisania programów tak, aby były kompatybilne z obydwoa trybami i obydwoa BASIC-ami, ale do tego potrzebna jest znajomość obu BASIC-ów. Na szczęście, jak pewnie zauważyłeś, BASIC 2.0 jest podzbiorem BASIC-a 7.0. Szczegółowego porównania tych dwóch BASIC-ów dokonaliśmy w rozdziale 3.

#### O systemie DOS w trybie C64

Ostatnią zauważalną różnicę w pracy COMMODORE'A 128 w trybie C128 i w trybie C64 stanowi struktura DOS. Przede wszystkim nowe rozkazy BASIC 7.0 DOS są już niedostępne w trybie C64. Mogą być wykonywane te same operacje dyskowe, ale musisz w tym celu zapoznać się z zestawem rozkazów DOS używanych przez Commodore 64.

Tabl.4-1 - Wykaz rozkazów i funkcji BASIC-a 2.0

Systemowe	Podstawiania	Funkcje Tekstowe	Funkcje Numeryczne	Inne Funkcje	WE/WY	Przebieg programu
NEW	LET	ASC	ABS	PRE	GET	GOTO
CLR	DIM	CHR\$	ATN	POS	GET	ON...GOTO
CMD	DEF FN	LEFT\$	COS	SPC	INPUT	GOSUB
LOAD	DATA	LEN	EXP	TAB	INPUT	ON...GOSUB
SAVE	READ	MIDS	FNxx	DEF FN	PRINT	RERUTN
VERIFY	RESTORE	RIGHT\$	INT	PEEK	PRINT	FOR...NEXT..STEP
OPEN	POKE	STR\$	LOG		WAIT	IF...THEN
CLOSE		VAL	PEEK			SYS
LUN			RND			USR
STOP			SGN			
CONT			SIN			
END			SQR			
LIST			TAN			
REM						

### Różnice w rozkazach

Jednowyrazowe rozkazy, takie jak HEADER, CATALOG, RENAME czy SCRATCH zastąpione zostały krótkimi zdaniami. Np. dla uformowania nowej dyskietki musisz najpierw otworzyć kanał operowania na zbiorach do stacji dysków za pomocą instrukcji:

OPEN 1,8,15

Następnie musisz użyć innego rozkazu BASIC-a 2.0 do przesłania rozkazu DOS kanałem:

PRINT 1, "NEW: nazwa dyskietki , id "

W tym momencie możesz przesyłać więcej rozkazów DOS bez potrzeby powtórznego otwierania kanału. Przykłady różnych rozkazów DOS oraz ich znaczenie pokazane zostały w Tabl.4-2.

Tabl.4-2 - Przykładowe instrukcje DOS w BASIC-u 2.0

Zdanie w BASIC-u 2.0	Znaczenie DOS
"NEW: nazwa dyskietki , id "	Uformowanie nowej dyskietki
"SCRATCH: nazwa zbioru "	Kasowanie zbioru
"RENAME: nowa nazwa = stara nazwa "	Zmiana nazwy zbioru
"COPY: przeznaczenie = źródło "	Kopiowanie zbioru
"VALIDATE"	Odzyskiwanie niewykorzystanej przestrzeni na dyskietce
"INITIALIZE"	Inicjalizuje nową tablicę lokacyjną dyskietki /disk allocation table/

Po przesłaniu ostatniego rozkazu DOS powinienś zamknąć ten kanał rozkazem:

CLOSE 1.

W ten sposób można przysyłać większość rozkazów DOS. Jednakże istnieją dwie operacje wykonywane przez BASIC 2.0 DOS, które mają dość skomplikowaną budowę. Są to: czytanie skorowidza dyskietki i czytanie kanału stacji dysków, którym przesyłane są informacje o zaistniałych błędach.

W BASIC-u 2.0 skorowidz dyskietki musi być załadowany do pamięci tak, jakby był zbiorem w BASIC-u. Załadowany raz, może być przeglądany /listowany/ z poziomu BASIC-a. Oto zestaw instrukcji, które realizują to zadanie:

LOAD "g", 8

LIST

Ponieważ BASIC traktuje skorowidz jako program, zostaje on umieszczony w pamięci BASIC-a, niszcząc znajdujące się tam już programy. A więc nie możesz sprawdzać skorowidza dyskietki bezpośrednio przed zapisaniem programu na nią, aby zobaczyć czy dana nazwa zbioru już się tam znajduje. Jest to często przykra niedogodność. Poza tym przeglądanie skorowidza może być nieco kłopotliwe, szczególnie gdy jest on długi, a nie ma sposobu łatwego wyświetlenia pojedynczej strony informacji w danym momencie.

Jeśli coś działa nieprawidłowo i stacja dysków nie jest w stanie wykonać żądania DOS, należy odczytać zawartość kanału błędów, aby określić rodzaj błędu. Np. możesz przypadkowo użyć niesformatowanej dyskietki w stacji dysków lub starać się zapisać na dyskietce zbiór o nazwie, która już istnieje w skorowidzu. Są to typowe błędy operatora. Oczywiście, kanałem tym wysłane będą też informacje o poważniejszych problemach, jak wystąpienie błędu przy czytaniu poszczególnych ścieżek /wynikające np. ze zużycia dyskietki/. Niestety, Commodore 128 nie wyświetla automatycznie wiadomości o błędach.

dzie natury dyskietkowej, a nawet nie sprawdza czy błąd taki się pojawił.

Tabl.4-3 - Porównanie różnych operacji systemu DOS komputera C128

DOS Wedge	BASIC 2.0	BASIC 7.0
§	LOAD "§" : LIST	DIRECTORY
nazwa zbioru	LOAD " nazwa zbioru ", 8	DLOAD " nazwa zbioru "
% nazwa zbioru	LOAD " nazwa zbioru ", 8,1	BLOAD " nazwa zbioru "
nazwa zbioru	SAVE " nazwa zbioru ", 8	DSAVE " nazwa zbioru "
nazwa dyskietki , id	PRINT 1, "N: nazwa dyskietki, id "	MEADER " nazwa dyskietki , id"
S: nazwa zbioru	PRINT 1, "S: nazwa zbioru	SCRATCH " nazwa zbioru "
V	PRINT 1, "V"	COLLECT
	100 OPEN 1,8,15	? DS§
	110 INPUT 1,ER,ER§, T,S	
	120 PRINT ER;ER§;T;S	
	130 CLOSE 1	

Trzeba mu zlecić wykonanie tej czynności. Zarówno w trybie C128, jak i C64 wymagane jest ręczne sprawdzanie błędów, ale metody odczytywania stanu dyskietki są zupełnie różne. W BASIC-u 7.0 wystarczy tylko napisać:

? DS §

Natomiast w przypadku BASIC-a 2.0 jest to bardziej skomplikowane, mianowicie:

100 OPEN 1,8,15

110 INPUT 1, ER, ER§, T, S

120 PRINT ER; " "; ER§; " "; T; " "; S

130 CLOSE 1

RUN

Jak więc widzisz, konieczny jest mały program, aby uzyskać informacje o błędzie w stacji dysków.

#### Usprawnienia 2.0 DOS

Ostatni przykład świadczy o tym, jak skomplikowane są czasem operacje BASIC 2.0 DOS. Nie załamuj się jednak. Jeśli będziesz musiał wykonywać operacje DOS w trybie C64, nie będziesz mógł użyć BASIC-a 7.0, ale będziesz mógł skorzystać z C64 DOS Wedge, podobnie jak dwa miliony użytkowników Commodore 64. C64 DOS Wedge jest to krótki program pomocniczy, który firma Commodore stworzyła, aby ułatwić niektóre operacje BASIC-a 2.0 DOS. Program ten jest ładowany w nieużywane miejsce pamięci C64, a wiele operacji połączonych systemów BASIC-a 2.0/Wedge DOS jest łatwiejszych do zapamiętania i wymaga mniej pisania niż ich równoważnik w BASIC-u 7.0. C64 DOS Wedge jest dostarczany bezpłatnie z każdą stacją dysków 1541. Wiele różnych książek i czasopism zawiera opisy poszczególnych operacji tego programu. Wymienić tu można np. "COMPUTE!'s First Book of the Commodore 64".

Krótki przegląd instrukcji Wedge można znaleźć w Tabl. 4-3.

Więcej szczegółów o operacjach systemu DOS w trybie C64 można znaleźć w dokumentacji dostarczanej razem ze stacją dysków oraz w "The Commodore 128 System Guide".

## 5. TRYB CP/M

XXXXXXXXXXXXXXXXXXXX

W rozdziale tym dowiesz się:

- co to jest CP/M i dlaczego możesz chcieć go używać
- jakie wyposażenie potrzebne jest do uruchomienia CP/M na C128
- jakie oprogramowanie istnieje w świecie CP/M
- gdzie możesz znaleźć oprogramowanie dla Twojego C128
- jaka jest organizacja CP/M
- jakie są rozkazy CP/M i co one robią
- gdzie możesz uzyskać więcej informacji o CP/M

Jeśli nigdy wcześniej nie spotkałeś się z terminem "CP/M", to rozdział ten przeznaczony jest właśnie dla Ciebie. Rozdział ten będzie pożyteczny także, jeśli jesteś zaznajomiony z CP/M, pracującym na innym niż C128 komputerze, a chciałbyś poznać istniejące różnice. Zwrot CP/M oznacza Control Program for Microcomputers /Program Sterujący dla Mikrokomputerów/. Nie mówi Ci to wiele, ale daje dobre pojęcie tego, czym jest CP/M: jest to sposób sterowania programami. CP/M jest specjalnym rodzajem dyskowego systemu operacyjnego. Dyskowy system operacyjny jest rodzajem programu zarządzającego, który używany jest do wykonywania operacji, takich jak uruchamianie programów, wykonywanie kopii programów, wymazywanie zbiorów, pomiar pojemności dysku itp.

W części tej chcemy przedstawić Ci czym jest CP/M, jakie są jego własności i zalety, co jest potrzebne do używania CP/M, jakie istnieją najważniejsze i najbardziej użyteczne programy dla CP/M i gdzie je znaleźć - i w końcu nieco informacji o rozkazach CP/M i ich działaniu. Będziemy chcieli także przedstawić Ci gdzie należy szukać bardziej szczegółowych informacji o CP/M.

### Co to jest CP/M i czym jest tryb CP/M

Jeśli jesteś zaznajomiony z Commodore 64 i stacją dysku 1541 wiesz, że możesz użyć rozkazów BASIC-u do uzyskania dostępu do programów zmagazynowanych na dyskietce włożonej do napędu.

Jeśli czytałeś rozdział tej książki o nowym trybie C128 wiesz, że Commodore posiada więcej rozkazów BASIC-a służących do współpracy z dyskiem. Używasz tych rozkazów do wyszukiwania zbiorów i programów na dyskietce, znajdującej się w napędzie, do kasowania programów lub tworzenia ich kopii na innych dyskietkach, a nawet do uruchamiania programów w innych językach niż BASIC. Na nieszczęście obsługa ta związana z zarządzaniem dyskiem, zawarta jest na poziomie innego języka i posiada bardziej złożoną składnię niż sobie życzymy. Uniemożliwia nam to przejęcie całkowitego sterowania zbiorami i programami na dysku.

### Co to jest Dyskowy System Operacyjny?

CP/M jest w pełni wartościowym systemem operacyjnym, który daje całkowitą kontrolę nad dyskiem. Pierwszym celem stosowania dyskowego systemu operacyjnego jest uproszczenie obsługi zbiorów i programów. Jest to dokonywane w CP/M przez wprowadzenie zbioru rozkazów, które są bardzo proste w użyciu i łatwe do zapamiętania. Ponieważ CP/M nie jest częścią innego języka, tak jak rozkazy dyskowe C64 lub BASIC 7.0, korzystanie z niego jest znacznie prostsze. Dyskowy system operacyjny, taki jak CP/M spełnia obecnie rolę "matki" lub programu zarządzającego. Jest to innymi słowy program, którego używasz do sterowania innymi programami - rodzaj misarza, który przeprowadza operacje na programach, znajdujących się na dysku. Po co więc jeszcze jeden DOS? Firma Commodore zastosowała CP/M nie z tego powodu, że dostarcza on więcej lub lepsze rozkazy DOS, ale dlatego, że daje on możliwość

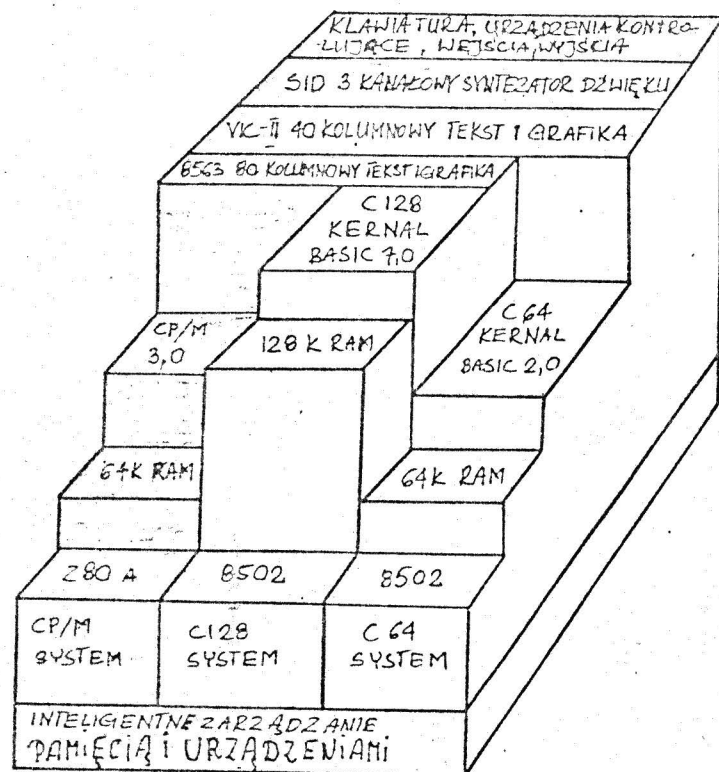
korzystania z ogromnego oprogramowania, które będzie mogło działać na C128. CP/M posiada unikalną pozycję w świecie mikrokomputerów. Był to pierwszy system operacyjny, zaprojektowany dla mikrokomputerów /CP/M wyszedł około 1975 r. - trzy lata po wprowadzeniu pierwszego mikrokomputera: Altair 8080/. Spowodował on, że wiele różnych mikrokomputerów może wykorzystywać ten sam program. Program napisany do pracy pod CP/M na jednym komputerze, będzie działał na innym komputerze, jeżeli komputer ten również pracuje pod systemem CP/M. Obecnie istnieją dziesiątki firm, które wytwarzają komputery z CP/M i istnieje olbrzymi wybór oprogramowania.

### Tryb CP/M: Trzecia Osobowość

W przeciwieństwie do systemu DOS C128, który rezyduje w pamięci ROM i może być uruchomiony bezpośrednio po włączeniu komputera, system CP/M umieszczony jest na dyskietce jako program i musi być najpierw załadowany do pamięci komputera, zanim cokolwiek będzie mógł wykonać. Rysunek pokazuje jak to jest robione.

Różnica ta jest istotna ponieważ musisz nauczyć się "wciągać" CP/M przed rozpoczęciem pracy z nim. Ponadto tryby C64 i C128 używają nowego mikroprocesora 8502. Z drugiej strony CP/M wymaga procesora Z80A. Commodore wbudował ten mikroprocesor do C128. Tak więc Z80 i CP/M zmieniają C128, nadając mu całkowicie odmienną osobowość, którą Commodore nazywa trybem CP/M. Rysunek pokazuje także różne układy lokalne, które tryb CP/M może wykorzystywać.





rys. 16 - Struktura systemowa Commodora c 128 w CP/M

#### Trochę z historii CP/M

CP/M został zaprojektowany przez Gary Kildalla /obecnie bardzo bogatego człowieka/, pracującego ówczesnie dla firmy Intel - olbrzimego przedsiębiorstwa wytwarzającego układy mikroprocesorowe. Gary zaprojektował pakiet /zestaw/ podprogramów dla małego, 4-bitowego mikroprocesora 4004. Podprogramy te mogły być używane przez inne programy i upraszczały one pracę

innych programistów Intela. Następnie zaawansowana technologia doprowadziła do zaprojektowania silniejszych układów mikroprocesorowych takich, jak Intel 8008 i 8080 oraz kompatybilnego z nim Z80. Kildall zaprojektował dalsze programy dla tych mikroprocesorów. Jego głównym celem było stworzenie środowiska, które pozwalałoby na magazynowanie i uruchamianie programów znajdujących się na dysku. Ponieważ nikt nie wierzył, że mikrokomputery staną się kiedykolwiek bardziej popularne, lub że będą kiedykolwiek używane w tak drogich urządzeniach, jak np. stacje dysków, Intel dał Kildall'owi pozwolenie na sprzedaż CP/M we własnym zakresie. Założył on spółkę nazwaną Digital Research, która zresztą do dzisiejszego dnia stale się rozwija.

CP/M był pierwszym dyskowym systemem operacyjnym dla mikrokomputerów i obecnie posiada ponad 1,5 mln użytkowników

Należy zauważyć, że do chwili powstania CP/M nie było żadnego systemu operacyjnego dla mikrokomputerów. Tak więc CP/M został szybko przejęty przez większość użytkowników i przedsiębiorstw wytwarzających komputery oparte na mikroprocesorach 8080 i Z80. Ponieważ CP/M nie posiadał żadnych konkurentów, szybko stał się standardem w przemyśle. Ponieważ każdy ówczesny komputer miał układ 8080 lub kompatybilny z Z80, CP/M sprawił, że jeden program mógł pracować na wielu różnych komputerach.

Większość systemów opartych na CP/M zawierała co najmniej klawiaturę i monitor lub terminal, jeden lub dwa napędy dysków oraz 48K lub 64K pamięci. Komputery nie były zaprojektowane tak, aby była zapewniona kompatybilność między nimi i każdy z nich posiadał ekran o innych wymiarach, dyski o różnych pojemnościach i różne typy klawiatur. Jednak CP/M - wielki kompensator - niwelował te różnice.

### Czego dostarcza CP/M

Obecnie istnieje wiele mikrokomputerów pracujących pod CP/M. Większość z nich posiada umiarkowane ceny /od 1000 - 2000 zł/ i jest nabywana przez małe i średnie firmy. Jednakże, ponieważ mikrokomputery pracujące pod CP/M przedstawiały tak cenną wartość, została sprzedana duża ich ilość i powstała ogromna ilość oprogramowania dla tych maszyn. Do chwili pojawienia się C128 nie było taniego komputera domowego pracującego pod systemem CP/M.

### Niewiarygodna ilość oprogramowania pracującego pod CP/M

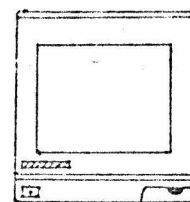
CP/M daje dostęp do ogromnego świata oprogramowania - świata, w którym istnieją dosłownie tysiące programów dla Twojego C128. Niektóre z tych programów są bardzo dobre. Wiele z nich jest bezpłatnych, jeśli tylko wiesz jak je uzyskać.

Aby dać Ci wyobrażenie jak wielka jest ilość dostępnego oprogramowania należy podać, że eksperci twierdzą, iż istnieje ok. 10 000 programów pracujących pod CP/M. Programy te zajęłyby ok. 3000 dyskieciek CP/M dla C128. Większość z tego oprogramowania jest do zastosowań profesjonalnych i w biznesie. Znajdziesz tu doskonałe programy przetwarzania tekstów, placht kalkulacyjnych, pakietów finansowych, zarządzania bazą danych, języków różnych rodzajów, programów użytkowych i wiele innych. W istocie C128 pracujący pod CP/M jest okazją dla małych zakładów /firm i przedsiębiorstw/. Cena C128 ze stacją dysków 1571 jest konkurencyjna w porównaniu z IBM PC. Oprogramowanie jest w pełni przetestowane i wolne od błędów.

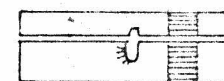
Dostępna jest taka ilość oprogramowania, że możesz łatwo znaleźć programy szczególnie interesujące dla Ciebie. Na przykład możesz znaleźć programy obliczeniowe, napisane specjalnie dla praktyki prawniczej, bazy danych dla zbieraczy znaczków, lub inżynierskie plachty kalkulacyjne dla metalurgów. Istnieje także olbrzymia ilość bezpłatnego oprogramowania. Dalej opiszemy to szczegółowo.

### Jakiego rodzaju sprzętu potrzebujesz do uruchomienia CP/M

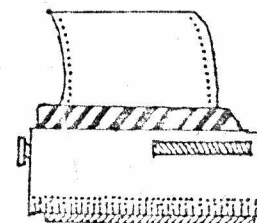
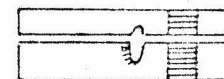
W celu zastosowania CP/M na C128 będziesz potrzebował użyć pojedynczej stacji dysków 1571 lub podwójnej 1572, 40 lub 80 kolumnowego monitora i oczywiście zbioru dyskie-tek z systemem CP/M. Możesz także nabyć drukarkę MPS-802 lub DPS-1101 oraz modem 1660 lub 1670 /patrz rys. /.



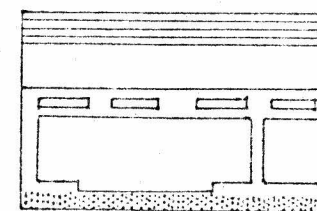
1902  
80 KOLUMNOWY  
MONITOR



1572 PODWÓJNA  
STACJA DYSKÓW



MPS-802  
DRUKARKA



COMMDORE C 128  
KOMPUTER

rys.17. Zalecana konfiguracja w CP/M

### Stacje dysków

C64 używa stacji 1541. Stacje te są wolne, ponieważ użyto w nich metody szeregowej do przesyłania danych z dysku do komputera. Nowe stacje 1571 i 1572 są o wiele szybsze niż 1541. Stacje stare przysyłały dane z prędkością 320 znaków na sekundę. Nowe w trybie CP/M mogą przysyłać dane z prędkością 3500 znaków na sekundę, co daje szybkość ponad 10 razy większą. Nowe napędy zawiera inteligentny mikroprocesor, który daje C128 przewagę nad innymi komputerami CP/M.

W pierwszych latach istnienia CP/M całe oprogramowanie było rozpowszechniane na 8-calowych dyskietkach i był używany tylko jeden format magazynowania danych na dyskietce. Gdy napędy dyskietek 5 1/4 cala stały się bardziej popularne, fabryki zaczęły stosować różne techniki formatowania. Obecnie stosuje się około kilkunastu formatów, co stwarza problem uzyskania kompatybilności. Wprawdzie nie istnieje taki program, który nie może pracować pod inną wersją maszyny CP/M, ale powstaje tu problem wprowadzenia programu do komputera, gdyż stacja dysku nie może zrozumieć sposobu zapisu na dyskietce. Stacja 1571 jest wystarczająco sprytna, aby automatycznie wykryć rodzaj formatu stosowanego dla CP/M. Oznacza to, że niezależnie od tego, co będzie uruchamiał Twój komputer, stacja 1571 jest zdolna to odczytać. Wybór pomiędzy stacją z pojedynczym napędem lub podwójnym jest sprawą wygody i kosztu. Pojedynczy napęd pozwala uruchamiać programy bezpośrednio. Jednakże wytwarzanie kopii programów i zbiorów przy użyciu pojedynczego napędu jest nużące, jeśli musi być często wykonywane. Podwójny napęd stacji 1572 pozwala łatwo kopiować zbiory z jednej dyskietki na drugą.

### Wyświetlanie

Są dwa sposoby wyświetlania tekstu w trybie CP/M. Najlepszym i najbardziej kosztownym sposobem jest nabycie nowego 80-kolumnowego kolorowego monitora 1902 firmy Commodore. Monitor ten posiada piękne kolory i może współpracować z komputerami C64, C16, Plus/4, C128. Ekran ma rozmiary 640 na 200 pixeli /najmniejszych elementów obrazu punktów/ i może wyświetlać paletę 16 kolorów. Nowy układ scalony video w C128, który jest często używany także w IBM PC, steruje tym ekranem w trybie 80-kolumnowym.

Zaletą trybu 80-kolumnowego jest to, że znaki są przechowywane i wyświetlane z pamięci RAM. Gdy komputer jest włączany po raz pierwszy do sieci, zbiór czcionek /znaków/ przechowywany w pamięci ROM przemieszczany jest do pamięci RAM. Gdy CP/M chce wyświetlić znak dokonuje tego biorąc go z tego obszaru pamięci. Ponieważ znaki przechowywane są w pamięci RAM, możliwa jest ich modyfikacja i tworzenie nowych znaków przez użytkownika, stosownie do potrzeb jego programów. Oznacza to, także, że możesz miksować grafikę i tekst na tym samym ekranie przy pracy z CP/M.

Możesz także używać 80-kolumnowego monitora monochromatycznego o wysokiej rozdzielczości. Wyświetla on tylko jeden kolor. Na kolorowym monitorze znaki tekstowe są rozciągnięte nieco w kierunku poziomym, ale są one jeszcze odczytywalne. Kolorowy monitor Commodore'a zawiera wbudowany głośnik, regulatory siły głosu, koloru, odcienia, jasności i kontrastu.

Drugim sposobem wyświetlania tekstu w CP/M jest tryb 40-kolumnowy. Tryb ten korzysta ze standardowego układu wyświetlającego VICII stosowanego w C64 i pozwala używać standardowego 40-kolumnowego kolorowego monitora 1702 firmy Commodore.

Monitor 1702 jest tańszy od nowego monitora 1902, ale nie potrafi wyświetlać 80 kolumn w linii, czego oczekują programy CP/M. Zamiast tego, pierwszych 40 kolumn jest wyświetlanych normalnie, natomiast specjalny klawisz klawiatury C128 pozwala na przesunięcie pozostałych kolumn tak, aby stały się widoczne. Czytanie programów CP/M w ten sposób, wymaga stałego napięcia uwagi przy obserwowaniu ekranu i jest zalecane tylko w przypadku, gdy będą tworzone programy, wymagające tylko 40 kolumn. Innym usprawiedliwieniem może być pisanie na CP/M gier, które używają koloru i dźwięku C128. Subtelna przewaga użycia 40-kolumnowego trybu polega na tym, że prędkość zegara systemowego jest zwiększona dwukrotnie z 1 do 2 MHz.

#### Dyskiety CP/M

Będziesz potrzebował zbioru dyskielek z CP/M dla swojego C128. Obecnie CP/M jest rozprowadzany w pakietach, składających się z 2 dyskielek. Jedna zawiera specjalne programy instalacyjne CP/M, a druga zawiera programy użytkowe. Wyjaśnimy to bardziej szczegółowo później.

#### Ważne programy dla posiadaczy CP/M

W 1976 r. było 100 programów, które pracowały pod CP/M. W 1983 r. było ich ponad 10 000. Przy tak wielkim oprogramowaniu CP/M użytkownicy CP/M na C128 stają wobec poważnego problemu wyboru. W części tej pomożemy wybrać najlepsze programy z szerokiego i obfitego asortymentu programów CP/M. Skupimy się przy tym na programach, które według naszego odczucia, są najlepsze także ze względu na cenę. Przy czytaniu tej książki zostaniesz poinformowany o kilku najbardziej popularnych programach i jednocześnie dowiesz się jakie jest ich działanie.

Popularne oprogramowanie CP/M zawiera programy przetwarzania tekstów, stosowane do pisania czegokolwiek - począwszy od krótkich listów, a skończywszy na długich rozdziałach książek. Większość ekspertów zgadza się z tym, że prawdziwie potężne programy przetwarzania tekstu były przyczyną rozpowszechnienia się oprogramowania CP/M na rynku. W końcu każdy potrzebuje napisać coś na komputerze. Istniejące narzędzia przetwarzania pozwalają na automatyczne sprawdzenie liter w Twoim dokumencie, budowanie tablicy skojarzeń, tworzenie skorowidzów, lub wstawianie imion w miejsce określonych liter.

Najbardziej popularnymi programami używanymi w biznesie, są programy typu "spreadsheets" /płachty/. Programy te pozwalają na manipulowanie liczbami i rysunkami w dużej macierzy, składającej się z rzędów i kolumn. Kilka najlepszych programów tego typu było zaprojektowanych po raz pierwszy dla CP/M.

Programy zarządzające bazą danych pozwalają na manipulowanie dużą ilością informacji, która może być zdefiniowana jako zbiór pól danych takich jak imiona, adresy, numery telefonów itp. Bazy danych są używane do: obsługi klientów, określanie kwot do zapłaty, prowadzenie inwentaryzacji, analizowanie danych, drukowania sprawozdań itp.

Dla CP/M stworzono także bardzo dobre oprogramowanie dla celów finansowych. Są to programy do obsługi kont, ksiąg obsługi /saldowania/, książeczek czekowych, sporządzania bilansów, zestawiania wpływów i programy wpływów podatkowych. Tysiące przedsiębiorstw prowadzi swoje rachunki przy użyciu CP/M. Odkryjesz także duży asortyment oprogramowania komunikacyjnego, pracującego pod CP/M. Programy te pozwolą Ci na wysyłanie zbiorów i programów przez linie telefoniczne, stosowanie zdalnej obsługi baz danych takich jak: The Source i ComupServe. CP/M zawiera słynny program telekomunikacyjny

zwany Modem 7 - jest on ogólnodostępny. Istnieje także duża ilość języków programowania, pracujących pod CP/M. Najbardziej popularnym jest Microsoft BASIC /zwany MBASIC/, ale znajdziesz także takie języki, jak: Pascal, język C, Forth, COBOL, Lisp, Prolog i inne. Nie oczekuj jednak wielu gier dla CP/M. Wprawdzie istnieją dobre gry tekstowe, ale Kildall zaprojektował CP/M do pracy ze zwykłymi terminalami i klawiaturami. Dlatego też standard CP/M nie dostarcza grafiki, dźwięku i klawiszy funkcyjnych, odpowiednich dla gier, a większość programów CP/M używa trybu 80-kolumnowego. Jednakże już wkrótce zobaczymy, że jest możliwe aby programy napisane pod CP/M C128 korzystały z grafiki, okien i efektów dźwiękowych C128. Możemy oczekiwać takich rozwiązań w ciągu najbliższych paru lat.

#### Niezgodność formatów zapisu danych na dysku

Jak wykazaliśmy uprzednio istnieje mnogość oprogramowania sprawdzonego w działaniu dla CP/M, które może być użyte bezpośrednio na Twoim C128. Jedynym wymaganiem jest to, aby było ono dostępne na dyskietkach 5 1/4 cala.

Jednym ze szczególnie drażliwych problemów, związanych z rozpowszechnianiem oprogramowania CP/M jest format zapisu danych na dysku. Początkowo CP/M tworzony był na dyskietkach 8 calowych jednostronnych o pojedynczej gęstości zapisu i stosowany przy tym jeden format zapisu zwany IBM System 34. Obecnie istnieje kilka nowych sposobów zapisu dyskietek, obejmujących dyskietki dwustronne o podwójnej gęstości zapisu. W chwili gdy wprowadzono dyskietki 5 1/4 cala, niektóre firmy komputerowe, jak KayPro, Morrow, Osborne, nieznacznie zmieniły format zapisu, odpowiednio do poszczególnych rozwiązań sprzętowych. W wyniku tego powstało niedopasowanie pomiędzy stacjami dysków różnych komputerów CP/M. Commodore omija ten problem niekompatybilności przez zastosowanie inteligent-

nego napędu peryferyjnego. Stacje 1571 i 1572 są zdolne odczytać dyskietkę sformatowaną według dowolnego z formatów CP/M. Gdy włożysz dyskietkę do 1571, stacja próbuje określić typ formatu zapisu, a gdy już tego dokona, odczytanie danych nie stanowi problemu. CP/M na C128 posiada także zdolność formatowania dyskietek według jednego z kilku formatów CP/M. Tak więc możesz napisać program, który będzie mógł być odczytany na innym komputerze CP/M. Teraz, gdy wiesz już nieco o formatach zapisu, zobaczmy z jakiego oprogramowania możesz korzystać.

#### Programy przetwarzania tekstów /Word processing/

Programy te należą do pierwszych programów jakie są nabywane przez większość użytkowników. Word processor jest to program edycyjny, który działa podobnie jak maszyna do pisania, ale posiada znacznie więcej możliwości od niej. Oprócz tego, że pozwala on Ci na natychmiastowe wstawianie liter, słów i zdań w dowolne miejsce przygotowywanego przez Ciebie dokumentu, może on także automatycznie poprawić cały dokument w ciągu kilku sekund, wyrównywać marginesy, zmieniać wielkość odstępów itd.

#### WordStar

WordStar był pierwszym programem edycyjnym dla CP/M typu "to co widzisz jest tym co otrzymasz". Oznacza to, że takie operacje formowania jak podwójne spacje, justowanie, ustawianie marginesu, są widoczne na ekranie zanim tekst zostanie wydrukowany. Tekst zostanie wydrukowany w takiej postaci, w jakiej jest widoczny na ekranie. Starsze edytory tekstu podzielone były na dwie części: edytor do wprowadzania tekstu i urządzenie formatujące/drukujące do wydrukowania tekstu. Nie można było zobaczyć jak będzie wyglądał wydruk, zanim nie została wydrukowana kopia. Pro-

gram ED.COM wchodzący w skład oprogramowania CP/M dla C128 jest "starszym krewnym" WordStar'a.

WordStar tak jak wiele innych programów, musi być "zainstalowany" na konkretnym systemie komputerowym z CP/M, na którym będzie pracował. Proces ten jest niezbędny, ponieważ każdy system może mieć ekran o różnych rozmiarach, różne klawiatury. Gdy instalujesz program CP/M musi mu po pierwsze określić charakterystyki swojego ekranu i drukarki. Następnie program używa wbudowanej tablicy do wprowadzenia do programu zmian, niezbędnych do jego poprawnej pracy na sprzęcie posiadanym przez użytkownika.

WordStar używa potężnego programu "instalacyjnego", który dostarcza bardzo wyczerpującego menu dotyczącego drukarek i terminali, które mogą z nim pracować. Mimo, że WordStar jest jednym ze starszych programów i bardziej dojrzałe programy znajdują się już na rynku CP/M, to posiada on złożony zbiór rozkazów, które musisz zapamiętać aby pracować na nim efektywnie. Dobrze zaprojektowany zbiór "help" sprawia, że jest to bardzo łatwe. Dokumentacja do WordStar'a jest maszynowa, ale istnieje oprócz niej wiele książek, które pozwalają w sposób prosty zapoznać się z nim. Dostępna dla WordStar'a opcja "mailmerge" pozwala Ci na drukowanie kopii tych samych listów do różnych ludzi, których adresy znajdują się w zbiorze baz danych. Możesz także wprowadzać dodatkowe informacje do każdego listu.

#### Perfect Writer

Perfect Writer jest częścią rodziny Perfect Software, w której skład wchodzi Perfect Calc i Perfect File. Tak jak Word Star, Perfect Writer jest edytorem całoe ekranowym, ale dodatkowo pozwala na edycję ekranu podzielonego na części. Oznacza to, że możesz podzielić ekran na dwa poziome okna i

przeglądać różne części tego samego dokumentu. Pozwala Ci to czytać jedną część podczas modyfikacji drugiej. W rzeczywistości Perfect Writer pozwala na otworenie 7 zbiorów w tym samym czasie i dokonywanie jednoczesnej ich edycji. Inną zaletą zastosowania podziału na okna jest to, że możesz ciąć i łączyć dwa dokumenty jednocześnie znajdujące się na ekranie. W WordStar musisz przepisać tekst, który ma być "doklejony" np. na dyskietkę, następnie skasować bieżący dokument, załadować ponownie nowy dokument i wczytać uprzednio przepisany zbiór tekstowy. Wbudowane rozkazy, które ustalają odstępy, marginesy, nagłówki i zakończenia umieszczane są w tekście w postaci specjalnych symboli. Rozkazy te powodują formowanie całego zbioru. Ale istnieją także mechanizmy formowania tekstu, dzięki którym każdy paragraf może być oddzielnie przeformowany. Perfect Writer jest nieco przyjemniejszy w użyciu niż WordStar, ponieważ większość jego rozkazów wywołuje się dwoma klawiszami /Escape i Control/. Perfect Writer posiada także doskonały program instalacyjny. Istnieje wersja Perfect Writer dla C128.

#### Programy zarządzające bazami danych /Database Managers/

Database managers są programami używanymi do manipulowania danymi, występującymi w postaci rekordów. Rekordy można sobie wyobrazić jako pojedyncze linie informacji, które zawierają np. nazwiska, adresy, wielkość kont, ilość dostępnych elementów itd. Database manager jest programem, który pozwala Ci wprowadzać dane, ustalać jak będą przechowywane, szukać określonych danych, drukować skrócone sprawozdania. Najczęstszymi użytkownikami database managers są przedstawiciele biznesu, którzy najczęściej mają do czynienia z informacjami dotyczącymi klientów, które te informacje mogą łatwo być przechowane w bazie danych.



dBASEII

Prawdopodobnie najbardziej popularnym programem bazy danych dla CP/M jest dBASEII firmy Ashton-Tate. Znany jest jako relatywna /względna/ baza danych, co oznacza, że jakkolwiek nowa informacja może być dołączona do już istniejących. Np. nazwisko może poprzedzać adres i numer telefonu, lub numer telefonu może poprzedzać adres i nazwisko. Aktualne zastosowania dBASEII wymagają umiejętności technicznych przy programowaniu, ale żadne techniczne umiejętności nie są potrzebne przy użyciu programu końcowego. dBASEII zawiera język programowania, którego używasz do tworzenia bazy danych. Początkowo jest on trudny do nauczenia, ale opanowany pozwoli Ci stworzyć złożone bazy danych. Istnieje wiele dobrych książek do nauki programowania dBASEII.

Spreadsheets /płachty kalkulacyjne, kalkulatory na tebkach/

Spreadsheet jest prostym elektronicznym rejestrem, służącym do analizy typu "co będzie gdy". Daje możliwość testowania różnych wariantów poprzez manipulowanie liczbowymi danymi wprowadzonymi w postaci rzędów i kolumn. Możesz wprowadzać oprócz liczb także formuły /wzory/, które będą działały na danych, znajdujących się w rzędach i kolumnach. Typowe pytania, na które spreadsheet odpowie mogą być następujące: "Jak inwestować jeśli stopa procentowa wynosi 15%?"; "Jaki jest efekt wytwarzania większej ilości skarpet purpurowych niż zielonych?"; Jednym z głównych zastosowań spreadsheet jest modelowanie i przewidywanie finansowe. Najważniejsze cechy charakteryzujące spreadsheet to rozmiar płachty roboczej, szybkość obliczeń, zdolność do tworzenia sprawozdań, współpraca z innymi programami i łatwość wprowadzania danych.

MultiPlan

MultiPlan jest jednym z pierwszych spreadsheet dla CP/M. Jest zorganizowany na bazie płachty roboczej /worksheet/, która przedstawiona jest na ekranie w postaci rzędów i kolumn, tak jak w rejestrach księgowych.

MultiPlan używa ekranu jako okna, w którym widoczny jest fragment dużo większej płachty roboczej. Na ekranie widocznych jest 7 wierszy i 19 kolumn, a cały obszar składa się z 63 kolumn i 255 wierszy. Masz możliwość przesuwania "okna" w dowolne miejsce płachty oraz wprowadzania danych do dowolnej komórki płachty roboczej. Możesz także wprowadzać formuły /wzory/, według których automatycznie zostaną dokonane obliczenia na danych zawartych w wierszach i kolumnach. Przeliczenie jest prawie natychmiastowe. Sposób w jaki pracuje MultiPlan jest podobny do pracy procesora tekstu, z wyjątkiem używania klawiszy kursora do poruszania się po rzędach i kolumnach. Możesz skasować duże obszary płachty za pomocą przyciśnięcia kilku klawiszy. Możesz także wprowadzać tekst np. nagłówki opisujące zawartość pól. Istnieje w MultiPlanie rozbudowany system pomocy, który pozwala Ci "skoczyć w program" bez obawy. Istnieje też kilka dobrych książek do nauki MultiPlan'u. Jedynym niedostatkim MultiPlan'u jest to, że nie posiada on prostego sposobu otrzymywania reprezentacji graficznej wyników. Możemy jednak znaleźć taką wersję MultiPlan'u, w której w połączeniu z C128 i jego doskonałym ekranem graficznym, będzie to możliwe.

SuperCalc 2

Innym doskonałym programem typu spreadsheet jest SuperCalc 2. Jest on podobny do MultiPlan'u, lecz jego płachta robocza posiada nieco więcej rzędów i kolumn. SuperCalc ma



interesujący system przetwarzania wsadowego, który pozwala Ci na umieszczanie szeregu jego rozkazów na płachcie i wykonywanie ich. Na przykład jeśli chcesz otrzymać wydruk raportu, który będzie używał różnych części płachty roboczej, rozkazy w zbiorze mogą określać obszary tych części niezależnie. Jest to szczególnie dogodnie dla ekspertów, którzy chcą zaprojektować złożone pakiety, które będą używane przez nowicjuszy. Supercalc używa także pewnego formatu składowania danych /specyficznego/ dla spreadsheet SDI /dla Super Data Exchange/. Może on przechowywać dane w postaci ograniczonej przecinkami, co powoduje, że są one dostępne dla innych języków programowania CP/M takich jak Microsoft BASIC, CBASIC i Pascal /patrz rozdział o językach programowania CP/M/.

#### Oprogramowanie finansowe

Programy finansowe są to pakiety CP/M, które pozwalają firmom prowadzić księgowość na komputerze. Większość pakietów finansowych oferuje ogólny program księgowania oraz system listy płac. Całkowity system zawiera programy do obsługi odbioru, płatności, inwentaryzacji i kosztów pracy. Oprogramowanie finansowe jest trudne do zdefiniowania i istnieje wiele pakietów należących do tej kategorii, znacznie różniących się między sobą. Nabywając taki system dla swojego C128 powinienś zwrócić uwagę na to, jak elastyczny jest dany program i jak elastyczne jest sporządzanie sprawozdań. Powinieneś także sprawdzić, jakie jest współdziałanie pomiędzy pakietami, tzn. czy różne części systemu, takie jak książki czekowe i generatory zestawienia wpływów, mogą porozumiewać się ze sobą. Możesz także być zainteresowany sprawą zabezpieczenia i uzyskać pewność, że istnieje możliwość ochrony programu przed dostępem osób nieupoważnionych /np. przez hasło/, ponieważ tylko np. niektórzy pracownicy mogą wypisywać чеки.

#### Software Pittnes Program

Jest to rzetelny, "oparty o ekran" siedmioczęściowy system rachunkowości. Oferuje on programy obejmujące ogólne rejestry, obsługę odbioru, wyliczanie płatności, porządek sprzedaży, inwentaryzacje, listy płac i koszty zatrudnienia. Wprowadzane ekrany są zdumiewająco wszechstronne. Rozkazy edycyjne podobne do rozkazów WordStar'a pozwalają Ci łatwo poruszać się po ekranach programu, wprowadzać i modyfikować dane znajdujące się na nich. Program może przystosować się do Twojego sposobu prowadzenia rachunkowości. Wszystkie dane o programie są przechowywane w tablicach i mogą być modyfikowane. Istnieje wiele typów funkcji sprawozdawczych, umożliwiających rewizje ksiąg, porównania i analizy. Sprawozdanie dla klientów z uwzględnieniem formatu jaki sobie życzą, jest również możliwe. System ma mnóstwo zalet. System płac obejmuje zasiłki chorobowe, urlopy, FICA, stanowe i lokalne podatki, a nawet nadgodziny.

Program pracuje pod kontrolą specjalnego interpretera - BASIC Interpreter firmy Control C. Program ten musi być nabyty osobno. System zasługuje w pełni na swó nazwę: jest wszechstronny, łatwy do nauczenia i posiada dobrą dokumentację. Szczególnie cenną cechą dla użytkowników C128 jest to, że program jest zdolny zapewnić pracę 24 użytkownikom, pracującym na tym samym zbiorze.

#### Komunikacja /łączość/

Ostatnim szaleństwem są sieci komputerowe i komunikacja między systemami. Łączenie w sieci i komunikacja pozwala na dostęp do oddalonych komputerów najczęściej za pośrednictwem linii telefonicznych. Może to być wykorzystane do wysyłania elektronicznej poczty MCI lub skopiowania

zbiorów lub programów z innego komputera. Poczta elektroniczna jest metodą, z pomocą której możesz pisać i przysyłać do komputera podłączonego do towarzystwa poczty elektronicznej listy. List taki może być wydrukowany i doręczony. Wysyłanie i odbieranie programów jest najczęstszą formą komunikacji. Możesz także użyć komputerowej komunikacji do wywołania oddalonych komputerów i przeszukiwać je, włączając się po fantastycznych bazach danych, które zawierają. Możesz mieć również dostęp do ton bezpłatnego oprogramowania CP/M. Aby rozpocząć komunikację potrzebujesz modemu i programu komunikacyjnego. Modem koduje i dekoduje informacje z komputera na postać standardowych tonów. Jeśli komputer, z którym jesteś połączony posiada modem, to oba komputery mogą się porozumiewać i wymienić informacje między sobą. Gdy odbierasz informację z innego komputera, potrzebujesz otworzyć zbiór na dysku i przejąć ją. Twoje oprogramowanie komunikacyjne musi także być w stanie zaemulować rodzaj popularnego terminala. Jest to konieczne, gdyż większość służb informacyjnych dokonuje połączenia zgodnie z protokołem dostosowanym do określonego rodzaju terminala. Protokół oznacza, że jest prowadzona specjalna kontrola poprawności przekazu pomiędzy nadajnikiem a odbiornikiem, która zapewnia wiarygodność transmisji. Protokół jest niezbędny z powodu możliwości występowania w liniach telefonicznych ładunków elektrostatycznych lub szumów, które mogą spowodować zniekształcenie nadawanych i odbieranych informacji. Stosowanie protokołu gwarantuje, że znak wysłany, będzie prawidłowo odebrany. Najbardziej popularnym protokołem jest protokół XMODEM, który zaprojektowany został przez Warda Christensena. XMODEM jest częścią innego programu zwanego Modem 7, zaprojektowanego przez Warda i opisanego w dalszej części rozdziału, zatytułowanej "Bezpłatne oprogramowanie". Większość połączeń między komputerami odbywa się bez pomocy protokołu. Wysłany

znak jest powtarzany przez komputer odbierający informację. Tak więc możesz zobaczyć go na swoim ekranie /tryb ten nazywany jest full duplex/. W trybie tym nie jest jednak wykonywane sprawdzanie identyczności znaków wysłanego i odebranego. Tryb ten zwany jest trybem bezprotokołowym. Tryb jakiego użyjesz zależy od rodzaju przesyłanego zbioru. Zbiór programy taki jak COM jest utworzony najczęściej w postaci czystego kodu dla mikroprocesora Z80. Wysyłanie takiego zbioru wymaga zastosowania protokołu. Natomiast możesz nie mieć żadnych obaw w czasie przesyłania notatki /listu/ za pośrednictwem poczty elektronicznej.

### Crosstalk

Mimo tego, że Crosstalk produkowany przez Microsoft nie był pierwszym programem komunikacyjnym na rynku, szybko stał się jednym z najbardziej popularnych. Crosstalk jest zaprojektowany głównie do pracy z modemem Hayes Smart modem ponieważ modem ten jest w pewnym stopniu "inteligentny", co jest wymagane przez program Crosstalk. Może on automatycznie wybierać numer zarówno w systemie impulsowym, jak i częstotliwościowym i posiada możliwość wyboru jednego z kilku protokołów. Crosstalk przechwytuje przychodzące dane i umieszcza je w pamięci, pozwala Ci na ich edycję i zapisanie na dysku, gdy są już gotowe do zapamiętania. Tak jak większość dobrych programów komunikacyjnych, Crosstalk pozwala Ci ustalić ważne parametry takie, jak numer telefonu, szybkość przesyłu informacji, liczbę bitów danych, liczbę bitów stopu, parzystość, pełny lub pół-dupleks. Tym co jest szczególnie miłe jest to, że ustalasz wszystkie te parametry na wydzielonym fragmencie ekranu, tak że stan ich jest zawsze znany. Ponieważ każdy komputer, z którym chcesz się porozumieć może wymagać różnych parametrów, Crosstalk pozwala Ci na zapamiętanie tych parametrów w postaci zbioru o nazwie np. SOURCE.

XTK. Gdy później napiszesz LO SOURCE.XTK, to Crosstalk załaduje te parametry.

Oprócz posiadania dobrze zaprojektowanego trybu emulacji terminala, Crosstalk posiada tryb przesyłania zbiorów. Komunikujące się komputery mogą być zupełnie różne i pracować w odmiennych konfiguracjach tzn. możesz uzyskać połączenie między C128 a Osborne. Jeśli tylko oba te komputery pracują pod programem Crosstalk, to przesłanie zbioru jest łatwe. Ludzie używający programu Crosstalk do przesyłania "upakowanych" zbiorów typu COM, docenią tryb nie wymagający nadzoru /unattended mode/. Pozwala on programowi Crosstalk wywoływać nie pilnowane komputery pracujące także przy wykorzystaniu Crosstalk, sprawdzać zbiory i przysyłać je w razie potrzeby. Przesłanie może odbywać się zarówno przy pomocy protokołu XMODEM w którym wszystkie znaki są sprawdzane w czasie wysyłania, jak i w trybie bezprotokołowym, w którym nie istnieje kontrola poprawności danych. Crosstalk jest "kuloodporny" i prawie niemożliwe jest zniszczenie go. Rozkazy help w trybie on-line dają Ci opis każdego rozkazu bez potrzeby zwracania się do podręcznika. Crosstalk dostępny jest na wielu komputerach włącznie z IBM PC, przenośnych komputerach CP/M takich jak Osborne, Morrow i Kay Pro.

### Języki

Jeśli nie jesteś zadowolony z zastosowań CP/M, lub chcesz tworzyć lepiej sprzedawalne programy, musisz wiedzieć jak używać języka komputera. Język programowania jest tym, co jest używane do tworzenia czegoś - co ostatecznie stanie się "instrukcjami" Twojego układu scalonego Z80. Na przykład, CP/M, WordStar i MultiPlan były napisane w języku niskiego poziomu, zwanym assemblerem. Z drugiej strony Software Fitness Program był napisany w BASIC'u - języku programowania wyższego rzędu. Istnieje wiele różnych typów języków komputero-

wych pracujących pod CP/M. Ponieważ świat CP/M był w początkowych latach tworzony przez hobbistów komputerowych tzw. hackers, zrodziło się wiele języków eksperymentalnych z których kilka stało się produktami komercyjnymi. Istnieją dziesiątki różnic pomiędzy językami, niemożliwych do przedstawienia w tej książce, ze względu na brak miejsca. Ogólnie mówiąc najbardziej popularnym obecnie jest język BASIC, szczególnie dlatego, że jest łatwy do nauczania i stosowania. Ponadto jest dostępny w swych różnych wersjach na prawie wszystkich komputerach oraz jest językiem konwersacyjnym. Następnymi pod względem popularności są: Pascal i trochę nowszy język C. Pascal jest językiem poważniejszym, wymagającym większego doświadczenia, ale w zamian bardziej elastycznym /i czasami szybszym/ niż BASIC. Pascal używany jest tradycyjnie w szkołach do nauki programowania wyższego stopnia. Język C powoli osiąga pozycję Pascala, ponieważ jest szybszy od niego i dostarcza bardziej zwartego kodu. Ponadto na tym języku oparty jest znany system operacyjny UNIX. Wiele profesjonalnych programów powstało także w oparciu o język C.

### Microsoft BASIC

Microsoft BASIC posiada fantastyczną historię w świecie CP/M i przemysłu komputerów osobistych w ogóle. Bill Gates obecnie pracujący w jednym z największych software'owych wydawnictw w przemyśle i jego partner John Allen stworzyli w 1975 roku Microsoft BASIC, aby uruchomić pierwszy, prawdziwy mikrokomputer Altair 8080. W tych dawnych czasach nie było jeszcze dyskietek, więc Gates i Allan uruchamiali BASIC przy użyciu czytnika taśmy papierowej i programu wprowadzającego dane /loader/. Gdy powstał CP/M Gates i jego partner szybko stworzyli wersję BASIC'a opartą na dysku zwaną MBASIC. Program natychmiast przyjął się i od tej pory

Microsoft BASIC stał się faktycznie standardowym językiem programowania dla CP/M.

Microsoft BASIC jest dostępny w wersji interpretera i kompilatora. Interpreter, bardziej popularny, przyjmuje rozkazy BASIC'a w chwili napisania ich i wykonuje programy natychmiastowo. Kompilator BASIC'a czyta instrukcje, które są utworzone w zbiorze tekstowym przy użyciu edytora i zwykle po kilku minutach podaje Ci informację, jeżeli występują jakieś błędy. Jednakże kompilator wytwarza o wiele szybszy kod. Najczęściej programiści próbują przygotować program z wykorzystaniem przyjemniejszego w obsłudze interpretera, a następnie używają kompilatora do przyspieszenia wykonywania programu. Używanie Microsoft BASIC pod CP/M jest bardzo proste. W BASIC'u każda instrukcja zajmuje jedną linię i jest poprzedzona numerem linii. Możesz zatem kontrolować zawartość programu przy pomocy edytora linii. Możesz usunąć linię przez napisanie numeru linii bez podawania instrukcji, lub możesz wstawić nową linię poprzez podanie nowego numeru i instrukcji. Możesz uzyskać listing całego programu lub jednej określonej linii. Microsoft BASIC posiada wbudowany edytor, ale możesz także tworzyć program korzystając z jakiegokolwiek popularnego programu obróbki tekstów, lub nawet edytora ED.COM. W MBASIC posiadasz zbiór wbudowanych funkcji takich jak np. SIN, LOG i EXP. Struktury sterowania nie są tak rozbudowane jak w Pascalu i C - zawierają WHILE-WEND, IF-THEN-ELSE i FOR-NEXT.

W MBASIC możesz łączyć programy zbyt duże, aby w całości zmieściły się w pamięci, jak również używać zwykłych i sekwencyjnych plików. Funkcja PRINT USING jest przeznaczona do wyprowadzania liczb dziesiętnych w uporządkowanych kolumnach. Sprawdzanie błędów jest dobre i możesz sam napisać własne podprogramy obsługi błędów użytkownika. MBASIC daje dostęp do pamięci za pośrednictwem rozkazów POKE i PEEK.

Instrukcje CALL i USR pozwalają Ci uruchamiać programy w języku maszynowym Z80 z poziomu języka MBASIC. Podręcznik do Microsoft BASIC jest bardzo dobry, a ponadto istnieje wiele książek o programowaniu w tym języku.

Microsoft BASIC nie jest tak silny jak BASIC 7.0 na C128, ale jest kompatybilny z wieloma innymi maszynami. Na przykład MBASIC i BASICA dołączane do każdego IBM PC są prawie identyczne z MBASIC dla CP/M. Ponieważ 80-kolumnowy kolorowy procesor wizyjny wbudowany w C128 posiada takie same właściwości, jak karta koloru w IBM PC, jest prawdopodobne, że wiele programów napisanych na IBM PC w BASIC'u pracujących z kolorem, będzie można łatwo przerobić tak, aby działały na C128.

#### PASCAL/MT+

Zwolennicy bardziej wyrafinowanego języka Pascal zapewne docenią wartość języka PASCAL/MT+ firmy Digital Research. Pascal jest językiem kompilacyjnym /choć pojawiają się ostatnio również interpretery Pascala/, najbardziej znanym ze swojej "strukturalności". Pascal jest bardziej dopasowany do oczekiwań zawodowych programistów niż język BASIC /choć najnowsze wersje BASIC'a z możliwościami programowania strukturalnego wzorowanego na PASCAL'u zaczynają powoli tracić swój charakter bardziej zabawowy niż profesjonalny/.

Język Pascal/MT+ jest jednym z najszybszych i najbardziej wydajnych języków typu Pascal na rynku CP/M. Jest całkowicie kompatybilny z popularnym językiem USCO Pascal, ale jest szybszy i wytwarza bardziej upakowany kod /mniej bajtów na program/. Język pozwala Ci tworzyć dużą liczbę programowych nakładek /overlays/, które mogą być pobrane z dysku i umieszczone w określonym miejscu pamięci. Pozwa-

la to programistom Pascala na pisanie o wiele dłuższych programów niż wydawałoby się to możliwe. Przetwarzanie matematyczne dostępne w tym języku jest bardzo silne. Liczby mogą mieć długość do 32 bitów, a arytmetyka BCD pozwala na użycie 18 cyfr - co jest konieczne w przypadku rozliczeń finansowych. /Błędy zaokrąglania są prawdziwym problemem w większości języków używanych w interesach/.

Istnieje wiele dodatkowych pomocniczych programów użytkowych dostarczonych wraz z językiem Pascal/MP+ zawierających kompletny system uruchamiania /debugging system/, disassembler, linker /program łączący/, służący do użycia PASCALA/MP+ z programami w języku maszynowym. Jednym z najbardziej korzystnych dodatków jest pakiet szybkiego programowania. Jest to pewien typ edytora podobnego do WordStar'a, który posiada możliwość automatycznego formowania i wbudowany wyszukiwacz błędów syntaktycznych /składniowych/.

Jeśli jesteś dobry w Pascal'u wychyci on prawie wszystkie Twoje błędy. Następnie Twój zbiór przekazany zostaje do szybkiego kompilatora, który pomija sprawdzanie składni. Końcowa wersja programu Pascala pracuje z taką samą szybkością, ale dzięki temu pakietowi uzyskaliśmy przyspieszenie programowania.

#### BDS-C

Język C jest językiem programowania, którego popularność stale wzrasta. Język C został stworzony przez Kernighana i Ritchie z Bell Labs dla pomocy przy opracowywaniu systemu operacyjnego zwanego UNIX. Pascal nie był dobry do tego celu, ponieważ nie wytwarzał wystarczająco upakowanego kodu, był zbyt wolny i nie posiadał prostego połączenia z jednostką centralną procesora /CPU/. Mimo tego, że język C nie był przez długi czas dostępny na CP/M, jego popularność stale wzrastała z powodu jego przewagi nad innymi językami stu-

żącymi do projektowania skomplikowanych programów. Język C znajduje się w połowie drogi pomiędzy Pascallem a językami assemblerowymi. Wygląda on bardziej tajemniczo niż Pascal i jest zazwyczaj o wiele szybszy od niego. Język C powstał dlatego, że system operacyjny jest bardzo złożoną częścią oprogramowania i musi być szybki w działaniu. Pisanie systemu w języku maszynowym może wydawać się najlepszym rozwiązaniem, ale gdy program się rozrasta, język assemblera staje się nieporęczny i trudny do sterowania. Język C jest językiem strukturalnym tak jak Pascal. Niektóre doskonałe programy dla CP/M były napisane przy użyciu języka BDS-C m.in. edytor tekstu Mince i word processor PeachText.

Język BDS-C był pierwszą wersją języka C, która pojawiła się na rynku CP/M. W ciągu lat rozwijał się, aż stał się wolnym od błędów programem, preferowanym przez wielu użytkowników. Nie jest to tak pełny język, jak opisany przez Kernighana i Ritchie język The C Programming Language /Prentice - Hall, 1978/, ale zawiera duży podzbiór ważnych własności. Jedną z nich jest to, że program używa bardzo mało pamięci nawet przy największych programach. Programy matematyczne zmienoprzecinkowe nie są kompatybilne z podobnymi na Unix C. Szybkość kompilacji programów napisanych w BDS-C jest zdumiewająco duża w porównaniu z językiem takim jak Pascal. Czasami program jest kompilowany w ciągu sekund, podczas gdy w Pascalu ten sam proces może zająć kilka minut. Kompilator BDS-C jest łatwy w użyciu ponieważ nie wymaga wielu złożonych operacji łączenia przy uruchamianiu. Skomplikujesz program, następnie połączysz i już jest gotowy do działania. Jedynym słabym punktem BDS-C jest to, że tworzone relokowalne zbiory przez kompilator nie są kompatybilne z assemblerem Microsoft tak, że łączenie z językiem maszynowym jest utrudnione.

### Programy użytkowe

Programy użytkowe pomagają Ci pracować w środowisku CP/M. Są one także programami, które znajdują zastosowania np. jako kalendarze i notesy, lub dostarczają narzędzi, które sprawiają, że programowanie pod CP/M jest łatwiejsze.

### Power!

Power! z Computing! w San Francisco jest programem przygotowanym przez Digital Research. Jest to zbiór programów użytkowych, które przejmują obowiązki w sytuacji, gdzie PIP, TYPE, ERASE i RENAME obawiają się działać.

Aby uruchomić Power! podajesz jego nazwę i program sam instaluje się wewnątrz CP/M, stając się Twoim "niewolnikiem". Najbardziej użytecznym rozkazem jest rozkaz COPY. Gdy wykonasz rozkaz DIR w programie POWER! przed każdą nazwą zbioru zostanie wyświetlony numer. Jeśli teraz chcesz skopiować wybrane zbiory wystarczy, że podasz: COPY 1-3,7,13 i wtedy zostaną skopiowane zbiory o numerach 1-3,7 i 13. Rozkaz ERASE pracuje w podobny sposób. Istnieje także możliwość sprawdzenia czy istnieją jakiegokolwiek złe sektory na dyskietce i odtworzenia zbiorów, które przypadkowo skasowałeś. Power! posiada wygodne w użyciu menu, łatwe do kontrolowania przez początkujących. Istnieje także dobry system uruchomieniowy dla programistów, który pozwala Ci czytać sektory z dysku lub zawartość pamięci.

### Rozrywka

Programy rozrywkowe są czymś czego brakuje z powodu natury sprzężenia z użytkownikiem. Powodem tego jest fakt, że aby zachować kompatybilność programu z każdym komputerem pracującym pod CP/M nie może robić nic oprócz umieszczania znaków tekstowych w różnych miejscach ekranu. Standardowy CP/M

nie ma możliwości generowania dźwięku i koloru. Istnieje natomiast mnóstwo gier tekstowych, takich jak Zork i Adventure. Istnieją miliony gier napisanych w MBASIC takich jak Tic-Tac-Toe, Animals, Guess the Number i Blackjack, ale nie spodziewamy się po nich wiele. Najlepsze gry są zazwyczaj pisane w języku maszynowym lub C.

### Przygoda /Adventure/

Adventure z firmy Adventure International jest grą, w której przeszukujesz jaskinię pełną dziwnych magicznych postaci. Komunikujesz się z grą za pośrednictwem uproszczonego języka angielskiego. Celem jest znalezienie jednego z kilku skarbów - musisz przy tym nie dać się zabić. Wpisujesz do Adventure polecenie lub pytanie np. "Kill dragon" /zabij smoka/, lub "pick up magic lantern" /podnieś magiczną laternię/. Natomiast Adventure udzieli Ci dłuższych odpowiedzi np. "jesteś w dużej jaskini i ściany pokryte są galaretką - nie możesz wspiąć się po nich". /Jak dowiesz się później, jedyną drogą wyjścia jest przegryzienie się przez ścianę/. W grach Adventure możesz zobaczyć spis rzeczy, które zebrałeś, obejrzeć otoczenie, zbierać materiały itd. Ludzie bywają tak zachwyceni grami typu Adventure, że chęć wygrania staje się ich obsesją.

### Bezpłatne oprogramowanie dla CP/M

Wcześniej wspominaliśmy, że istnieje bezpłatne oprogramowanie dla CP/M - np. XMODEM. Nie wspomnieliśmy o tym, że niektóre programy bywają bardzo dobre, że uzyskanie ich nie jest trudne. Oprogramowanie typu public domain /ogólnie dostępne/ stanowią programy, które są dostępne dla każdego kto tego sobie życzy. Są one w rzeczywistości własnością publiczną.



Ponieważ CP/M był pierwszym systemem operacyjnym dla mikrokomputerów wielka liczba hobbistów nabywała komputery CP/M i rozpoczęli oni pisać programy sami. Pamiętaj, że zanim nastąpiła rewolucja mikrokomputerowa programiści i hobbysci nie mogli używać komputerów, gdyż tylko duże przedsiębiorstwa mogły sobie pozwolić na ich kupno. Od chwili ukazania się tanich mikrokomputerów CP/M, dosłownie tysiące hobbystów powiedziało: "Teraz mogę napisać program, o którym zawsze marzyłem" i przystąpiło do pracy. Od tej chwili powstawały programy, z których większość była słaba, ale niektóre były prawdziwymi perłami.

Ile jest bezpłatnego oprogramowania? Wielkość tego oprogramowania jest mierzona w tomach. Tom jest to pojedyncza dyskietka zawierająca od 10 do 20 programów i 250 000 znaków. Jeden z katalogów składa się z 250 tomów bezpłatnego oprogramowania! Oprogramowanie to można otrzymać w klubach użytkowników komputerów w różnych częściach kraju. Wysyłają one tomy biorąc tylko zwrot kosztów kopiowania. Drugim znaczącym źródłem są systemy "electronic bulletin board" /elektroniczny biuletyn/ zwane RCPMS /Remote CP/M computer systems/. Oparte są one o komputery CP/M pracujące w systemie on-line posiadające duże twarde dyski zawierające setki bezpłatnych programów CP/M. Programy te dostępne są za pośrednictwem modemu i mogą być "wciągnięte" do Twojego komputera za pomocą specjalnego bezpłatnego oprogramowania komunikacyjnego\*.

Jakie dobre bezpłatne oprogramowanie jest dostępne

Najbardziej popularnymi programami są narzędzia programowe i programy użytkowe. Istnieją programy do odzyskiwania uszkodzonych dysków i odtwarzania skasowanych zbiorów, a także dziesiątki programów komunikacyjnych. Przejrzymy najbar-

\* - nie w KDL

dziej popularne programy, które mogą być przydatne użytkownikowi CP/M 3.0 na C128.

#### Modem 7

Modem 7 /rozprowadzany także razem z programem zwanym XMODEM/ jest najbardziej popularnym programem, który posiada dosłownie setki wersji. Program ten pomaga w uzyskiwaniu innych, bezpłatnych programów od RCPM's. Specjalny tryb sprawdzania powoduje, że nie występuje żaden błąd podczas operacji przesyłania. Bez takiego sprawdzania byłoby prawie pewne, że program nie działałby właściwie. Ponieważ modemy dla CP/M nie są standaryzowane, Modem 7 musi być dopasowywany do każdego komputera CP/M, z którym będzie współpracował. Szczęśliwie różne wersje programu Modem 7 były dopasowywane do większości maszyn CP/M i modemów na rynku, tak że nie zajmie Ci dużo czasu znalezienie wersji Modem 7 odpowiedniej dla C128. W uznaniu znaczenia i wszechobecności programu Modem 7 większość komunikacyjnych programów na rynku posługuje się takim samym protokołem transmisji jak Modem 7.

#### Kompresja /SQ/ i Ekspansja /USQ/

Program SQ używany jest do kompresji zbiorów, tzn. zmniejszanie ich rozmiarów tak, aby mogły być przesłane przez modem w krótszym czasie. USQ są programami, które przywracają zbiorom normalną postać.

#### Programy obsługi dysku /Dysk Utility - DU/

DU pozwala na: inteligentną naprawę dysku w celu odzyskania skasowanych zbiorów, izolowanie uszkodzonych sektorów oraz dokonywanie "przeszczepów" danych z dysku, który



wydać się być całkowicie zniszczony. DU jest nieco złożonym programem narzędziowym, ale szczególnie popularnym, gdyż pozwala programiście CP/M stać się chirurgiem. DU jest także popularnym przy usuwaniu zabezpieczenia dysku i analizowaniu formatu zapisu danych.

#### FindBad i UNERA

FindBad lokalizuje uszkodzone sektory na dysku i izoluje je tak, że w przyszłości możesz używać dysku tak, jakby nie był uszkodzony. UNERA pomaga Ci odzyskiwać zbiory, które przypadkowo wymazałeś z dysku. Nawet gdy skasowałeś zbiór komend: ERASE w rzeczywistości istnieje on jeszcze na dysku - tylko jego nazwa jest usunięta ze skorowidza dyskietki.

#### Dan's Information Management System /DIMS/

Jest to pełny system zarządzania bazą danych rywalizujący z dBASEII. Program jest systemem programów Microsoft BASIC powiązanych ze sobą i zaprojektowanych w ten sposób, że mogą one być używane przez każdego. Napisany został przez Dana Dugana i nadaje się dla przeprowadzania operacji pocztowych średniej skali, spisów alfabetycznych, rejestrów i podobnych zastosowań. Rekordy mogą mieć długość do 256 znaków i do 30 pól danych, a rozmiar zbioru ograniczony jest jedynie pojemnością dysków.

#### Inne, dobre, ogólnodostępne oprogramowanie

CBASIC 2 Accounting System firmy Osborne /Mc Graw-Hill zawiera kompletny system obliczeniowy przeznaczony dla małych firm wraz z odpowiednią literaturą, dostarczaną przez wydawcę Osborne/Mc Graw-Hill. Istnieje ponadto wiele języków włącznie z kompilatorem języka BASIC-E z Lawrence Livermore Laboratory,

BDS-C, kilka wersji języka Forth, PILOT i nawet kompilator Pascala. Możesz także znaleźć gry takie jak oryginalny program StarTrek i inne gry video. Istnieją także assembly, programy ustawiania zbiorów, programy zwiększające szybkość działania systemu, programy szybkiego kopiowania i wiele innych.

#### Jak otrzymać bezpłatne oprogramowanie?

Są dwie metody wymiany bezpłatnego oprogramowania: pierwsza za pośrednictwem dysku przez grupy użytkowników, a druga poprzez połączenie komputerów linią telefoniczną z systemem głównym - bulletin boards.

#### Grupy użytkowników CP/M

Grupa użytkowników CP/M jest to grupa ludzi, zajmująca się katalogowaniem i wymianianiem informacji o CP/M bez czerpania z tego korzyści materialnych. Grupy użytkowników spotykają się zwykle dwa razy w miesiącu i są rozrzucone po całym kraju. Możesz dowiedzieć się czegoś o grupach, znajdujących się na Twoim obszarze poprzez czytanie magazynów, wyszczególnionych na końcu rozdziału, lub poprzez czytanie prasy lokalnej\*. Grupy użytkowników oferują zwykle oprogramowanie każdemu, nawet osobom nie będącym członkami, a kilka największych grup kopiuje i wysyła oprogramowanie na dyskach przy pobraniu symbolicznej opłaty.

Jedną z największych grup oferujących ogólnodostępne oprogramowanie CP/M odpowiednie dla C128 jest CP/MUG i składająca się z grup użytkowników CP/M /CP/M User Group/. Grupy te działają przy poparciu Lifeboat Associates in New York. CP/MUG dostarczają 5 1/4 calowych dyskietek o pojedynczej gęstości w formacie Kaypro w cenie 18 \$ za dyskietkę.

\* w USA

Inną dużą grupą użytkowników jest First Osborne Group /FOG/ z miasta Daly w Kalifornii. Rozpowszechnia ona oprogramowanie według formatów: Osborne, Kaypro II, Morrow, Compu-Pro, PCM Micromate i Zorba. Posiada ona ponad 15 000 członków na całym świecie. Przystąpienie do grupy kosztuje 24 \$.

#### Bulletin Boards

Drugim sposobem otrzymania bezpłatnego oprogramowania jest telefoniczne połączenie się z systemem bulletin board i załadowanie programu do swojego komputera. W całym kraju\* istnieją tysiące takich systemów konserwowanych przez pewnych pracowitych i życzliwych ludzi, którzy czerpią zadowolenie z tego, że ludzie korzystają z ich systemów CP/M. Większość z tych systemów posiada ostatnio wydane oprogramowanie grup użytkowników takich jak np SIG/M, FOG i C/MUG. Możesz zapoznać się z zawartością tych programów przed załadowaniem poprzez przeczytanie najpierw specjalnego zbioru z opisem programów. Systemy RCP/MS działają jako punkty rozprowadzania oprogramowania grup SIG/M i FOG, a ich numery telefonów są rozpowszechniane przez te grupy. W Stanach Zjednoczonych istnieje ponad 25 systemów FOG RCP/MS. Kontakt z tymi grupami /adres i telefon/ podany jest w Dodatku A. Innym źródłem bezpłatnego oprogramowania CP/M są CompuServe i The Source.

#### Struktura CP/M: układ, rozkazy i programy użytkowe

Witamy w CP/M 3.0.

CP/M pojawił się w kilku wersjach. Po raz pierwszy ukazał się jako wersja 1.3 i krótko po tym jako wersja 1.4. W 1979 r. wprowadzono wersję CP/M 2.2. Jak się ocenia na początku 1983 r. istniało ponad 1,5 mln kopii CP/M.

\* USA.

W 1983 r. Digital Research opracował nową wersję CP/M zwaną CP/M 3.0. W ostatniej chwili zmieniono nazwę na CP/M PLUS, jednak lepiej przyjęła się nazwa CP/M 3.0. Wersja ta jest rozszerzoną wersją 2.2 z pewnymi nowymi rozkazami i małą różnicą: jest zdolny do pracy z pamięciami o pojemności większej niż 64 K. Powstały dwie wersje CP/M 3.0: wersja dla pracy z bankami pamięci /banked version/ dla dużych pamięci i wersja zwykła dla pamięci o pojemności do 16 K. Wersja z przełączanymi bankami pozwala na pracę dwóch programów jednocześnie: każdy pracuje w swoim obszarze pamięci. Z powodów technicznych C128 używa zwykłej wersji CP/M 3.0 i nie używa w sposób bezpośredni dodatkowej pamięci w komputerze. Jednakże jest możliwe przy użyciu subtelnych metod uzyskanie dostępu do dodatkowych obszarów pamięci.

Niektóre programy dla wersji 1.3 i 1.4 znajdują się jeszcze w użyciu. Powinienesz jednak wiedzieć, że programy pracujące pod wersją CP/M 1.3 i 1.4 nie będą pracowały pod wersją 2.2 lub 3.0. Również programy pracujące pod wersją 2.2 i 3.0 nie będą pracowały pod wersjami 1.3 i 1.4. Jednakże większość programów napisanych dla wersji 2.2 będzie bez zmian pracowało pod wersją 3.0.

#### Rozpoczęcie pracy w CP/M

W przeciwieństwie do BASIC'a 7.0, który jest zainstalowany w pamięci ROM wewnątrz C128, program CP/M jest przechowywany na dysku i musi być wczytany przed każdym użyciem. Proces ten zwany jest booting /wciąganie/. Na dysku znajdują się dwa programy, które są ładowane w chwili rozpoczęcia pracy: CPM.SYS i CCP.COM. Kiedy zakończy się proces wciągania, CP/M jest już w pełni "zainstalowany" w pamięci i będzie pracował aż do chwili wyłączenia komputera.

Jeśli posiadasz 80-kolumnowy monitor, to po załadowaniu programu CP/M na ekranie pojawi się komunikat:

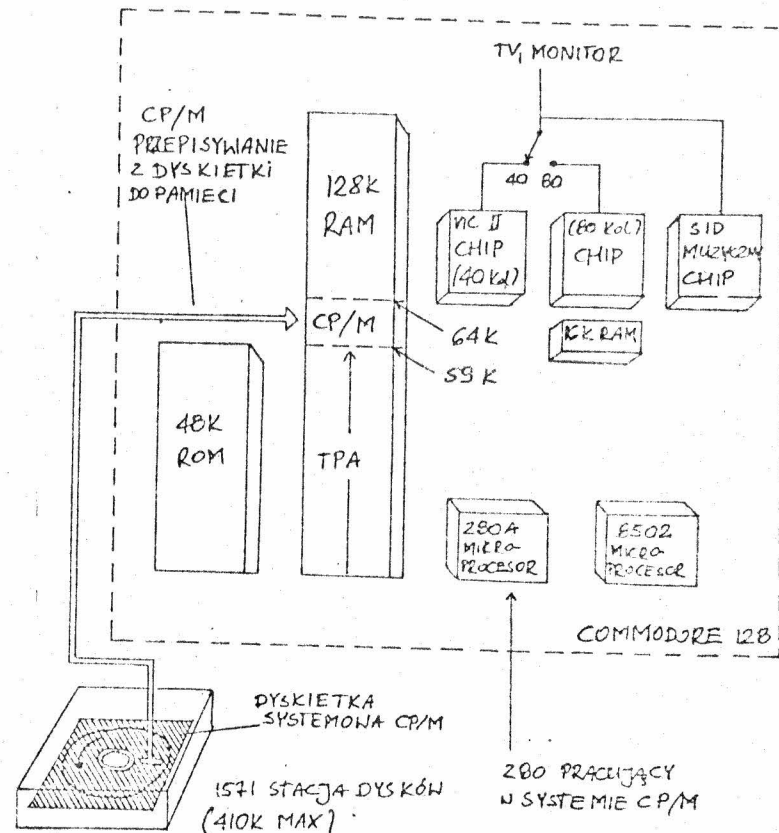
CP/M 3.0 On the Commodore 128 13 March 85  
80 column display

A>\_

Znak A>ukazuje się na dole ekranu. Jest to znak zachęty CP/M. /prompt/. Znakiem zachęty może być symbol lub ciąg znaków, które wyświetla system operacyjny gdy oczekuje na wprowadzenie przez Ciebie rozkazów. Oznacza to "Czekam, abyś coś napisał". Litera A informuje, że komputer jest logicznie połączony z napędem A. Napęd A jest to napęd, w którym umieszcza się dyskietkę z CP/M w celu załadowania systemu. Jeśli posiadasz dwa napędy podłączone do C128, to drugi napęd oznaczony jest literą B. Oznaczenie to pozwala Ci przełączać dyski i wykorzystywać programy znajdujące się na dyskietkach w różnych napędach. Symbol > /większe niż/ jest sposobem wskazania przez CP/M, w którym miejscu masz wprowadzić dane wejściowe. Za znakiem tym znajduje się migocący kursor. Kursor ten wskazuje, w którym miejscu na ekranie pojawi się znak, który wpiszesz w danej chwili.

#### Tak więc gdzie jest CP/M?

Informacja o tym, w którym miejscu pamięci jest umieszczony CP/M nie jest tak istotna w codziennej pracy. Jednakże jest to ważne w chwili, gdy mamy zamiar nabyć oprogramowanie, które nie było specjalnie zaprojektowane dla C128. Powodem tego jest to, że różne programy potrzebują pamięci o różnej wielkości, aby mogły działać. Jeśli masz pewność, że zrozumiałeś jak duży obszar wolnej pamięci oferuje C128 dla CP/M, to możesz mieć również pewność, że wybrane przez Ciebie programy będą działały.



rys. 18 Wykorzystanie mikroprocesora Z 80A w trybie CP/M

Część CP/M zajmuje wewnątrz pamięci C128 obszar zaczynający się od komórki o adresie 59000 a kończący się na komórce 64000. Poniżej znajduje się duży obszar pamięci, do którego są ładowane programy CP/M. Obszar ten zwany jest TPA /obszar przejściowy programów i obszar, w który ładowane są uruchomione pod CP/M programy. Dopóki program CP/M, który chcesz uruchamiać nie potrzebuje więcej niż 59 K pamięci, może on pracować pod CP/M 3.0.

### Zbiory w CP/M

CP/M pozwala na dostęp i manipulacje informacją przechowywaną w zbiorach na dysku. Zbiorem jest jakakolwiek informacja zgrupowana w niezależną jednostkę, posiadającą odrębną nazwę. Zbiorem może być pojedynczy program, pojedyncza litera, lub cała książka. Minimalna długość zbioru wynosi 1 znak, maksymalna jest ograniczona pojemnością dysku. Nie istnieją żadne podstawowe zasady, mówiące z czego ma się składać zbiór. Ty sam decydujesz o zawartości zbioru w momencie jego tworzenia. Natomiast istnieją zasady odróżniania jednego zbioru od drugiego. Zasadą tą jest nadawanie każdemu zbiorowi charakterystycznej i niepowtarzalnej nazwy. W CP/M wszystkie zbiory są opisane poprzez nazwę zawierającą od jednego do ośmiu znaków, zwaną nazwą zbioru /filename/. W CP/M nie może być dwóch takich zbiorów, które posiadają dokładnie taką samą nazwę. Ponadto dopuszczalne są tylko duże litery. Występuje także ograniczenie mówiące o tym, że tylko duże litery i symbole mogą być używane w nazwie zbioru.

Znakami, których nie możesz użyć są:

< > . ; : = ? \* [ ] ( ) /

oraz znak tabulacji. Poniżej porano kilka prawidłowych nazw:

CHAP-12

PIP

BASKET

PROGRAMS

A347689

ME@

BIG-FILE

RT+XT

CP/M zawiera liczne rozkazy, które pozwalają Ci na operacje na tych zbiorach. Zwykle musisz określić nazwę zbioru, na którym chcesz dokonać jakiejś operacji. Aby ułatwić to, CP/M używa specjalnych znaków, zwanych wildcards /nazwy zmienne/, które pozwalają Ci pracować na większej ilości zbiorów jednocześnie, a także na operacje na grupach zbiorów.

### Typy zbiorów

Gdy korzystasz z CP/M użyteczna jest możliwość oznakowania zbiorów w sposób, który pozwoli Ci wiedzieć do czego będą one używane. Na przykład zbiór tekstowy i zbiór programu są różne, ale możesz nie zorientować się o tym na podstawie ich nazw. CP/M pozwala Ci określić typ zbioru poprzez zastosowanie trzy-znakowego rozszerzenia dodawanego z prawej strony nazwy zbioru. Wykaz rozszerzeń podany jest w Tabelicy 5-1. Typ zbioru oddzielony jest od nazwy zbioru kropką - jest to zresztą przyczyna niemożności użycia kropki jako elementu nazwy zbioru.

Jeśli możesz zidentyfikować typ zbioru /filetype/, możesz łatwo określić z jakim rodzajem zbioru współpracujesz. Jest to przede wszystkim ważne, aby móc rozróżniać dwa rodzaje zbiorów: zbiory danych i zbiory programu.

Zbiór danych jest to zwykle zbiór, zawierający teksty, liczby i zestawy znaków. Typowym zbiorem danych jest dokument tworzony przez procesor tekstu. W CP/M zbiór taki po-

siada zwykle rozszerzenie TXT dla tekstów i DOC dla dokumentów. Zawartość zbiorów danych może zwykle być sprawdzona za pomocą rozkazu TYPE /jak? ... zobaczymy wkrótce/.

Zbiór programu zawiera instrukcje i kody w konkretnym języku komputera. Na przykład zbiór o rozszerzeniu .COM nazywany jest zbiorem rozkazowym i zawiera kody, które będą interpretowane przez Z80 w Twoim C128. Program napisany przez Ciebie w języku Microsoft BASIC może posiadać rozszerzenie BAS. Tablica 5-1 przedstawia kilkanaście najczęściej spotykanych typów zbiorów dla CP/M.

Książki wyszczególnione na końcu tego rozdziału, podobnie jak podręcznik C128 wyjaśniają te problemy bardziej szczegółowo. Należy zaznaczyć, że nie musisz stosować wyłącznie rozszerzeń wymienionych w tabeli 5-1. Możesz tworzyć i używać nazwy które sprawiają, że typ zbioru jest dla Ciebie oczywisty.

Tabela 5-1 - Typy zbiorów CP/M 3.0

Rozszerzenie	T y p
.ASM	Plik źródłowy w języku asemblera .
.BAK	Plik - kopia /backup/
.BAS	Program źródłowy w języku BASIC
.COM	Program w języku maszynowym 8080 lub Z80
.DAT	Plik danych
.DOC	Dokument
.FOR	Program w FORTRAN-ie
.HEX	Plik heksadecymalny utworzony przez programy MAC lub ASM
.INT	Pośredni plik roboczy /intermediate work file/
.LIB	Plik biblioteczny, używany przez ASM, MAC lub RMAC

Rozszerzenie	T y p
.PAS	Program źródłowy w PASCAL-u
.OVL	Plik nakładkowy, używany przez inny program
.OVR	- " -
.PL1	Program źródłowy w PL/1
.PRN	Drukowalny plik utworzony przez ASM, MAC lub RMAC
.REL	Relokowany plik utworzony przez RMAC
.SUB	Plik źródłowy wykorzystywany przez SUBMIT
.SYM	Tablica znaków /symboli/ utworzona przez ASM, MAC lub RMAC
.SYS	Plik systemowy CP/M /używany przez CP/M/
.TXT	Plik tekstowy
.XRF	Plik-odsylacz /cross-reference file/
.\$\$\$	Plik przejściowy, tymczasowy /temporary file/

Poniżej przedstawiono kilka przykładów pełnych nazw zbiorów wraz z rozszerzeniem:

MYPROG, BAS /Program w języku BASIC/  
 JOE-3-14, LFT /List do Joe napisany 3.14/  
 PIP, COM /Program CP/M do kopiowania zbiorów/  
 CHAP2, BAK /Kopia zbioru tekstowego CHAP2 TXT/  
 PERFECT, OVL /Zbiór-nakładka dla programu Perfect Writer/  
 A347689 PES /Mazny wybór dla nazwy/

#### Rozkazy wbudowane systemu CP/M

Rozkazy są instrukcjami, które mówią CP/M co ma robić. Egzekwujemy te rozkazy przez napisanie ich nazwy po znaku gotowości A > i naciśnięcie klawisza Return. Są dwa rodzaje rozkazów w CP/M i istnieje między nimi subtelna różnica.

Są to "rozkazy wbudowane" i "rozkazy przejściowe". Krótkie programy dla rozkazów wbudowanych znajdują się zawsze w pamięci od chwili, gdy CP/M rozpoczyna pracę i są one częścią CP/M. Programy dla rozkazów przejściowych nie są częścią CP/M i nie są automatycznie ładowane od chwili wystartowania CP/M. Gdy wprowadzasz rozkaz przejściowy, CP/M odwołuje się do dysku, poszukuje programu odpowiadającego temu rozkazowi, ładuje go do pamięci i wykonuje. Rozkazy wbudowane wykonywane są szybciej, ponieważ znajdują się one już w pamięci. Natomiast przejściowe są wolniejsze, gdyż są to programy, które muszą być po pierwsze zlokalizowane na dysku, a następnie załadowane do pamięci. Jednakże przejściowe mogą zajmować znacznie więcej miejsca niż wbudowane i w wyniku tego mogą wykonywać więcej. Przejściowe określane są często jako rozkazy "narzędziowe", ponieważ zwykle pomagają Ci wykonywać wiele użytecznych operacji na zbiorach i dysku.

W przypadku CP/M 3.0 różnica pomiędzy wbudowanymi i przejściowymi rozkazami jest jeszcze bardziej zamazana. Cztery rozkazy wbudowane posiadają przejściowe "rozszerzenie", które jest wykonywane jeśli potrzebujesz skorzystać z dodatkowych możliwości.

W tablicy 5-2 przedstawiono sześć rozkazów wbudowanych. Jako użytkownik CP/M powinieneś wiedzieć czy rozkaz jest wbudowany, czy przejściowy, ponieważ w przypadku przejściowego, odpowiadający mu program musi znajdować się na dysku. Jeśli podasz nazwę rozkazu przejściowego, który nie znajduje się na dysku, otrzymasz wiadomość: File not found /zbiór nie znaleziony/, co oznacza, że CP/M nie może znaleźć zbioru o podanej nazwie. Tak więc rozkazy wbudowane są bardziej wygodne w użyciu niż przejściowe.

Spśród sześciu wbudowanych rozkazów, dwa nie posiadają rozszerzeń przejściowych - DIRSYS i USER. Jeśli przyjrzyj

się swojej dyskietce systemowej CP/M, znajdziesz takie programy jak DIR.COM, ERASE.COM, RENAME.COM i TYPE.COM. Są to rozszerzenia wbudowanych komend, wykonywane wówczas, gdy rozkazu dodajemy parametry "niezrozumiałe" dla jego wbudowanej wersji. Parametry te zwane są "opcjami" /options/ lub parametrami opcjonalnymi i są wyróżnione przez umieszczenie dodatkowej informacji w nawiasach kwadratowych, bezpośrednio po nazwie rozkazu.

Na przykład, aby otrzymać prosty listing zbiorów znajdujących się na dysku, możesz użyć wbudowanego rozkazu DIR, ale aby uzyskać listing z podaniem rozmiarów zbiorów, używasz DIR SIZE, który to automatycznie przywołuje przejściową wersję DIR. Wszystkie opcje w CP/M 3.0 umieszczane są w nawiasach kwadratowych.

Tabela 5-2 - Rozkazy wbudowane systemem CP/M 3.0

Rozkaz	Wykonywana funkcja
DIR	Wyświetla skorowidz dysku
DIRSYS	Wyświetla skorowidz zbiorów systemowych
ERASE	Wymazuje /kasuje/ zbiór dyskowy
RENAME	Zmienia nazwę zbioru dyskowego
TYPE	Wyświetla zbiór tekstowy na ekranie
USER	Zmienia obszar użytkownika

Nowe zasady postępowania dla CP/M 3.0 są następujące: jeśli podamy rozkaz RENAME, ERASE lub TYPE bez podania nazwy zbioru, wykonywana jest wtedy wersja przejściowa i jesteś proszony o dalsze informacje. Na przykład jeśli wprowadzisz rozkaz TYPE bez podania jakiegokolwiek nazwy, przywoływane jest wersja przejściowa /odczytywana z dysku i wykonywana/ i zostajesz poproszony o podanie nazwy zbioru "Enter filename": /Wprowadź nazwę zbioru/. Jeśli rozkazy DIR ERASE lub TYPE



są wprowadzone wraz z nazwą zbioru i parametrami opcjonalnymi, również będzie wykonywana wersja przejściowa tych rozkazów. Masz rację, jeśli wydaje Ci się to nadmiernie skomplikowane. Przed opracowaniem CP/M 3.0 rozkazy przejściowe i wbudowane nie zachodziły na siebie. Powodem podstawowym z jakiego nakładają się na siebie w CP/M 3.0 jest umożliwienie rozkazom wbudowanym uzyskanie dodatkowych własności bez zmieniania ich nazwy.

Inne rozkazy wbudowane CP/M zmieniają numer logicznie przyłączonego do komputera napędu dyskowego /default disk drive/ - wskazany literą przed znakiem . Jeśli obecnie przyłączonym dyskiem jest dysk A i wprowadzisz oznaczenie nowego dysku z następującym po nim dwukropkiem /tzn, gdy napiszesz B:/ CP/M dokona zmiany dysku aktualnie przyłączonego /logicznie!/ na dysk B.

Teraz, gdy zrozumiałeś różnicę pomiędzy rozkazami wbudowanymi i przejściowymi, zbadajmy bardziej szczegółowo rozkazy wbudowane.

#### DIR i DIRSYS

Rozkaz DIR /wymawiany di, aj, a:/ używany jest do wyświetlania na ekranie nazw zbiorów, znajdujących się na dyskietce. Zbiory mogą znajdować się zarówno na dyskietce w napędzie A jak i w B. Wbudowana wersja rozkazu DIR w sposób prosty wyświetla nazwy jednego lub wszystkich zbiorów. Z drugiej strony wersja przejściowa pozwala na wprowadzenie osiemnastu różnych opcji i włącznie z możliwością uzyskania informacji: o rozmiarze każdego zbioru w bajtach, o łącznej długości obszaru zajmowanego na dyskietce przez wszystkie zbiory oraz z możliwością uzyskania uporządkowanego spisu nazw zbiorów. Możesz także z pomocą DIR.COM uzyskać informację o atrybutach zbiorów - to znaczy, jakiego są one typu - tylko odczytywalne

/read only/, zbiory systemowe, zbiory typu odczyt/zapis /read/write/ lub zbiory skorowidzowe.

#### ERASE, RENAME i TYPE

Rozkaz ERASE używany jest wtedy, gdy chcesz usunąć zbiór z dyskietki CP/M. Gdy zbiór zostanie usunięty, nie możesz go odzyskać z powrotem chyba, że posiadasz specjalne programy do odzyskiwania zbiorów - patrz poprzedni rozdział/. Musisz być więc bardzo ostrożny w czasie używania tego rozkazu. Rozkaz RENAME używany jest do zmiany nazw dowolnych zbiorów. Na przykład możesz chcieć przemianować zbiór CHAPONE.TXT na CHAPTWO.TXT. RENAME dokona tego bez zmieniania zawartości zbioru. Gdy teraz użyjesz DIR, zobaczysz zbiór o nazwie CHAPTWO w miejscu zbioru CHAPONE.

TYPE jest wbudowanym rozkazem CP/M, służącym do przeglądania zawartości zbiorów tekstowych lub zbiorów danych. Nie powinien używać TYPE do przeglądania zawartości zbiorów typów innych niż tekstowy, takich jak np. zbiór COM. Może to spowodować wysłanie nieprawidłowych kodów znaków na ekran, mogących odłączyć logicznie ekran i będzie wymagane ponowne załadowanie systemu CP/M.

#### USER

Każdy dysk CP/M może być podzielony na 16 regionów lub obszarów, zwanych obszarami użytkowników i numerowanymi od 0 do 15. Każdy obszar może być przeznaczony do przechowywania zbiorów dla różnych użytkowników dysku lub różnych zakresów tematycznych. Podział na obszary użytkowników jest najczęściej wykorzystywany na dysku twardym, na którym praca może być rozdzielona na kilku ludzi. Ponieważ prawdopodobnie nie będziesz miał wielu użytkowników swojego C128, rozkaz USER będzie raczej okazjonalnie używany do



magazynowania różnych kategorii zbiorów. Gdy uruchomisz CP/M po raz pierwszy, system automatycznie wybierze obszar 0. Możesz zmienić go na inny, przez podanie po komendzie USER numeru obszaru. Zbiory, które znajdują się w jednym obszarze nie są dostępne z innych obszarów, do chwili gdy zostaną przekopiowane do tego obszaru. Wszystkie rozkazy wbudowane są dostępne z obszaru użytkowego. Gdy włączysz nowy obszar użytkowy, CP/M zawiadomi Cię o tym nowym symbolem. Na przykład podanie USER 9 zmieni obszar użytkowy na 9, a znak zachęty przybierze postać: 9A> . Korzystając z obszarów użytkowników należy mieć na uwadze, że łatwo jest zapomnieć o położeniu zbioru w obszarach, co zabiera dużo czasu przy późniejszych próbach jego odszukania.

#### Programy przejściowe i użytkowe CP/M.

Istnieje 6 ważnych programów przejściowych CP/M 3.0, które są używane dość często i które przedstawiono w tablicy 5-3.

#### Pomoc CP/M i HELP

Jedną z nowych cech CP/M 3.0, która jest niedostępna we wcześniejszych wersjach CP/M, jest istnienie programu przejściowego HELP. HELP.COM używany jest do wyświetlania na ekranie informacji o różnych rozkazach w CP/M. Używa on bardzo dużego zbioru roboczego zwanego HELP.HLP, który zawiera większość używanych tekstów. Użycie polega na podaniu HELP przed słowem /tematem/, którego ma dotyczyć informacja np. HELP PIP. Na ekranie zostanie wtedy przedstawiony związany opis tego co robi rozkaz /PIP/. HELP nie może być użyte w czasie, gdy jest wykonywany program przez komputer. Jeśli masz dobrą książkę o CP/M, możesz zrezygnować z używania HELP i zaoszczędzić dość duży obszar na dysku, kasując HELP.HLP.

#### Przygotowywanie dyskietek CP/M: FORMAT, COPYSYS

Informacja jest magazynowana na powierzchni dyskietki CP/M jako ciąg bloków /zwanych sektorami/. Są stosowane różne sposoby umieszczania tych sektorów na dysku, dlatego też każdy dysk musi być przygotowany do odbioru informacji według wymaganego formatu. Przygotowanie takie nazywamy "formatowaniem". Musi być ono wykonane zanim czysta, wyjęta prosto z pudełka dyskietka, będzie użyta do magazynowania zbiorów. Rozkaz przejściowy służący do formatowania dysku CP/M nazywa się FORMAT. Dysk, na którym przeprowadzono operację formatowania znajduje się w stanie gotowym do odczytania, lecz nie zawiera żadnej informacji. Operacja DIR dokonana na czystym dysku spowoduje podanie wiadomości: no files /brak zbiorów/. Ponieważ posiadamy inteligentną stację dysków, wersja rozkazu FORMAT używana w C128 oczekuje na podanie typu komputera CP/M, z którym ma współpracować formatowana dyskietka. Od chwili gdy dysk CP/M zostanie sformatowany, jest on gotowy do przyjmowania zbiorów. Dysk taki jednak nie może być użyty do "wciągania" lub startowania CP/M ponieważ nie posiada jeszcze żadnego programu CP/M. W CP/M 3.0 program CP/M tak zwany "system" przechowywany jest częściowo na kilku niewidocznych sektorach dysku, a częściowo w postaci dwóch widzialnych zbiorów: CPM, SYS i CCP.COM. Gdy C128 jest uruchamiany po raz pierwszy sprawdza czy dysk CP/M znajduje się w napędzie. Jeśli tak, to zostaje załadowany do systemu CP/M z niewidocznych sektorów do pamięci. Sterowanie przekazywane jest do programu systemowego w pamięci, co powoduje wczytanie do pamięci zbiorów CPM, SYS i CCP.COM. Zbiory te zawierają większość kodu niezbędnego do skompletowania systemu CP/M.

Program COPYSYS używany jest do sformatowania dysku i skopiowania systemu CP/M, znajdującego się w niewidocznych sektorach z jednej dyskietki na drugą.

Tabela 5-3 - Najczęściej używane rozkazy przejściowe

Rozkaz	Wykonywane funkcje
HELP	Dokonuje wyświetlenia informacji o rozkazach CP/M
FORMAT	Używany do formatowania czystej dyskietki
COPYSYS	Tworzy nowy dysk z systemem CP/M
PIP	Używany do kopiowania zbiorów
SHOW	Podaje dane o zbiorach
ED	Standardowy program edycyjny /redagujący/ CP/M

Kopiowanie zbiorów

Gdy już sformatowałeś dysk i zainstalowałeś system CP/M za pomocą COPYSYS jesteś gotowy do kopiowania zbiorów na dysk przy użyciu PIP. PIP jest skrótem od nazwy Peripheral Interchange Program /program wymiany pomiędzy peryferiami/ i pozwala on CP/M na przemieszczanie zbiorów nie tylko pomiędzy dyskami, ale także pomiędzy dyskiem a drukarką, lub pomiędzy dyskiem a ekranem. W CP/M stacje dysku, drukarka i ekran są urządzeniami "peryferyjnymi", które mogą być dostępne bezpośrednio z CP/M.

Najważniejszą funkcją CP/M jest kopiowanie zbiorów z dysku na dysk. Istnieje duża liczba opcji, które można stosować w PIP. PIP może kopiować pojedyncze zbiory lub grupy zbiorów zmieniając ich nazwę podczas kopiowania, drukując grupy zbiorów, wyprowadzać część zbioru, łączyć kilka zbiorów w jeden zbiór itd.

Określanie rozmiaru zbioru: SHOW

Zanim będziesz mógł przekopiować zbiory z jednego dysku na drugi musisz mieć pewność, że jest wystarczająca ilość

miejsca na dysku docelowym /destination disk/ tzn. tym, na który będą kopiowane zbiory. Rozkaz SHOW używany jest do uzyskania informacji o wolnych obszarach dysku. SHOW używany jest także do określenia ilości zbiorów w skorowidzu. CP/M ogranicza maksymalną liczbę zbiorów, które mogą znajdować się na dysku do 64. Trzecią cechą SHOW, wywoływaną za pomocą opcji USER jest ustalenie liczby aktywnych obszarów użytkowych i liczby zbiorów zawartej w każdym obszarze. Inna opcja SHOW zwana DRIVE podaje opis cech charakterystycznych dysku, takich jak całkowita pojemność, obszar skorowidza i rozmiar bloku.

Edytor systemowy: ED

ED jest bardzo prostym programem edycyjnym, zwanym "edytorem linii", który jest dostarczany na każdej dyskietce CP/M. ED nie jest użyteczny do poważnego przetwarzania tekstów, lecz raczej do wprowadzania krótkich zbiorów tekstowych, takich jak procedury skatalogowane lub krótkie wiadomości. Jakkolwiek poważne przetwarzanie tekstów wymaga takich programów, jak Perfect Writer lub WordStar.

Rzadziej używane rozkazy przejściowe CP/M

Istnieje wiele rozkazów przejściowych w CP/M 3.0, które są rzadziej używane niż opisane poprzednio. Są one przedstawione w Tablicy 5-4. Zbadamy je w dalszej części.

DATE i FKEYS

Rozkaz DATE używany jest do ustawiania lub wyświetlania wewnętrznego czasu i daty. C128 przechowuje te informacje w zegarze czasu rzeczywistego. Gdy użyjesz DATE bez opcji, zostanie wyświetlony czas i data. Z opcją nastąpi ustawienie daty i czasu.

FKEYS używany jest do nadawania przez użytkownika C128 klawiszom funkcyjnym funkcji odpowiadających całym rozkazom CP/M. Na przykład możesz zdefiniować klawisz F1 w taki sposób, że gdy zostanie naciśnięty wygeneruje funkcję DIRW.COM FULL, EXCLUDE i RETURN. Spowoduje to listing wszystkich zbiorów, które nie są typu COM i oszczędzi nam przyciskania 26 klawiszy.

#### Rozkaz DEVICE i logiczne/fizyczne urządzenia

CP/M został zaprojektowany do pracy z różnymi rodzajami jednostek peryferyjnych lub "urządzeń". W C128 urządzeniami peryferyjnymi są klawiatura, drukarka, ekran, modem. CP/M posiada także coś co jest zwane "urządzeniami logicznymi". Urządzenia logiczne reprezentuje szczególną funkcję spełnianą przez Twój komputer, natomiast urządzenie fizyczne, jest to rzeczywiste wybrane przez Ciebie urządzenie, służące do realizacji tej funkcji. Przenosimy nasz wybór do C128 przez użycie rozkazu przejściowego DEVICE.

Tabela 5-4 - Rzadziej używane rozkazy przejściowe CP/M

Rozkaz	Wykonywana funkcja
DATE	Wyświetlanie i zmiana czasu i daty
DEVICE	Wyświetlanie i zmiana przydziału peryferii
FKEYS	Przydział rozkazów kluczem funkcyjnym C128
GET	Wprowadza informację ze zbioru zamiast z klawiatury
INITDIR	Przygotowuje dysk do oznaczenia czasu i daty dla zbiorów
PUT	Przesyła dane wyjściowe do zbioru dyskowego zamiast na ekran lub drukarkę
SET	Zmienia atrybuty zbioru, nadaje etykiety, ustala hasło

Rozkaz	Wykonywana funkcja
SET DEF	Ustala parametry "ścieżki" odpowiedzialnej za sposób poszukiwania zbiorów na dyskietkach
SUBMIT	Wykonuje rozkazy umieszczone w zbiorze dyskowym

Wybór taki czyni możliwym przypisanie różnym fizycznym urządzeniom różnych logicznych wartości, z którymi CP/M będzie się komunikował. Może to być przydatne gdy próbujesz sterować kilkoma różnymi urządzeniami podłączonymi do C128.

Rozkaz DEVICE używany jest zarówno do ustawiania przydziałów jak i do wyświetlania ich aktualnego stanu.

#### GET, PUT i INITDIR

Rozkaz GET pozwala Ci pobierać dane wejściowe ze źródła innego niż klawiatura - w tym przypadku ze zbioru dyskowego. Rozkaz GET jest przydatny przy automatycznym uruchamianiu programów, lub części programów, które wymagają każdorazowo takich samych danych wejściowych. Aby użyć GET, powinieneś utworzyć zbiór za pomocą edytora lub programu typu word processor, który zawiera informację, którą chcesz uzupełnić następny program. Tak więc musisz napisać GET FILL - nazwa i typ. Następne uruchomienie programu, który podałeś w GET spowoduje przesłanie informacji. Program Twój będzie wykonywany, ale będzie automatycznie zasilany zawartością zbioru, który określiłeś za pomocą GET.PUT posiada przeciwnie działanie w stosunku do GET, zamiast pobierania informacji ze zbioru PUT zapisuje dane wyjściowe programu do zbioru w celu późniejszego użycia. Na przykład chcesz skorzystać z danych wyjściowych kilku programów jako zbiorów teksto-

wych przeznaczonych dla programu przetwarzania tekstu w celu dokonania ich edycji. CP/M 3.0 posiada zdolność oznaczania czasu i daty zapisania zbiorów. Oznacza to, że w czasie gdy tworzysz lub zmieniasz zbiór w CP/M 3.0, a oznaczanie czasu i daty jest odblokowane, zbiór będzie zachowany wraz z datą i godziną jego zapisania. Czas ten i data mogą być wyświetlone w skorowidzu z pomocą DIR i SHOW. Aby odblokować funkcje ustawiania czasu i daty należy wprowadzić rozkaz INITDIR bezpośrednio po sformatowaniu dysku, zanim jakikolwiek zbiór zostanie na niego skopiowany.

#### Hasła, SET, SETDEF i SUBMIT

SET używane jest w kilku przypadkach. Najczęściej używa się SET przy zmianie atrybutów zbiorów dyskowych typu zapis/odczyt /write/read/. Zbiór może być typu R/O - tylko odczyt /read only/, R/W - odczyt/zapis /read/write/, SYS co oznacza zbiór systemowy lub DIR - zbiór skorowidzowy /jest to normalny atrybut nadany wszystkim zbiorom/. Z pomocą SET możesz chronić jakikolwiek zbiór lub grupę zbiorów przed modyfikacją. Możesz nawet chronić zbiór przed czytaniem czy kopiowaniem przez określenie hasła, które musi zostać podane, aby uzyskać dostęp do zbioru.

Hasła są nazwami przypisanymi do indywidualnych zbiorów. Użytkownik nie może skopiować, wykasować lub odczytać zbioru chronionego hasłem, dopóki nie wpisze tego hasła. SET pozwala ponadto na włączenie funkcji ustawiania czasu i daty.

Początkujący użytkownicy CP/M mieli często problemy w momencie, gdy byli zainstalowani do drugiego dysku /B/ i podali rozkaz, który znajduje się na dyskietce w napędzie A. Rozkaz SETDEF oprócz innych rzeczy pozwala Ci ustawić "ścieżkę poszukiwania" odpowiednio dla danego rozkazu CP/M. Tak więc możesz powiedzieć CP/M, aby najpierw "spojrzał" na dysk

w napędzie A, a dopiero później B przy próbie odszukania danego programu.

Sekwencja rozkazów, która normalnie wprowadzana jest z klawiatury może być umieszczona w zbiorze dyskowym i wywołwana przez rozkaz przejściowy SUBMIT. CP/M będzie wykonywał ten zbiór tak, jakby te rozkazy przychodziły z klawiatury. Kiedy lista rozkazów jest wyczerpana, sterowanie wraca do klawiatury. Taki tryb pracy nazywany jest przetwarzaniem wsadowym. Zwykle będziesz używał SUBMIT w przypadku, gdy będziesz chciał otrzymać częściej używany zbiór rozkazów - np. gdy będziesz chciał skopiować 10 różnych zbiorów na 10 różnych dyskietkach. Zbiór submit będzie zawierał 10 nazw zbiorów i operacje PIP i będziesz musiał podać zbiór rozkazów SUBMIT tylko 10 razy zamiast 1000. Możesz nawet przygotować taki zbiór, który automatycznie będzie żądał od Ciebie wprowadzenia nowej dyskietki.

#### Programy narzędziowe dla programistów CP/M

Istnieje kilka programów narzędziowych używanych przede wszystkim przez programistów języków assemblera. Jeśli nie masz zamiaru pisać programów w języku assemblera, nie musisz zwracać sobie tym głowy. Tablica 5-5 przedstawia te programy.

Rozkaz	Wykonywana funkcja
DUMP	Wyświetla hexadecymalną zawartość zbiorów CDM na ekranie
HEXCOM	Generuje wykonywalny zbiór COM ze zbioru Intel HEX
LINK	Tworzy wykonywalny zbiór COM z relokowanych modułów
MAC	Makroassembler CP/M

Rozkaz	Wykonywana funkcja
RMAC	Makroassembler relokowalny CP/M
SAVE	Zapamiętuje część pamięci jako zbiór dyskowy
XREF	Tworzy listing sprawozdawczy z tablic MAC/RMAC

Rozkazy MAC i RMAC są programami zwanymi assemblerami. Assembler zamienia program wpisany przez programistę na grupę specjalnych kodów, które należą do zbioru instrukcji maszynowych układu Z80 w C128. Oryginalny zbiór źródłowy jest napisany z pomocą word processor'a w języku zwanym językiem assemblera. MAC jest rozwinięciem starszego assemblera CP/M zwanego ASM i posiada możliwość tworzenia instrukcji "makro". Makroinstrukcja jest to skondensowana linia programu, która reprezentuje niejednokrotnie wiele instrukcji programu. RMAC jest assemblerem relokowalnym co oznacza, że pozwala programiście na pisanie kodu, który może być umieszczony w dowolnym miejscu pamięci bez uszczerbku dla jego pracy. MAC wytwarza kod absolutny nie relokowalny.

HEXCOM, LINK i XREF są programami narzędziowymi programów MAC i RMAC. SAVE jest rozkazem przejściowym, który bierze zawartość TPA i zachowuje ją w postaci zbioru. Jedynie programiści piszący programy w językach assemblerowych będą normalnie korzystali z SAVE.

#### Gdzie możesz nauczyć się czegoś więcej o CP/M?

Istnieje wiele sposobów znalezienia dalszych informacji o CP/M. Istnieje wiele dobrych książek, które szczegółowo wyjaśniają jak należy wykorzystać system operacyjny CP/M. Są też książki, które pokazują jak pisać programy, które mają być uruchamiane pod CP/M, a także takie, które przedstawiają wewnętrzną strukturę CP/M. Istnieją nawet książki, które

opisują najlepsze oprogramowanie dla CP/M. Poniżej podano tytuły kilku użytecznych książek:

CP/M Primer, Second Edition by Stephen Murtha i Mitchell Waite /Indianapolis: Howard W.Sams & Co., Inc., 1983/. Proste wprowadzenie do CP/M i jego programów użytkowych. Uniklane w swej przystępności dla początkujących.

CP/M User Guide, Osborne/Mc Graw-Hill, Third Edition, by Thom Hogan /Berkeley: Osborne McGraw-Hill, 1984/. Jest to książka bardzo tej szczegółowa i wyczerpująca niż CP/M Primer, ale także bardziej złożona. Zawiera opis wszystkich wersji CP/M włącznie z CP/M 3.0, używanym przez C128.

The CP/M Plus Handbook, by Alan R. Miller /Berkeley: Sybex Computer Books, 1984/. Książka ta jest całkowicie poświęcona wersji CP/M 3.0, która z wyjątkiem kilku szczegółów, jest kompatybilna z CP/M używanym na C128. Nie jest tak wyczerpująca jak CP/M User Guide.

Soul of CP/M, by Mitchell Waite and Robert Lafore /Indianapolis: Howard W.Sams & Co., Inc., 1983/. Jest to bardzo proste wprowadzenie do struktury wewnętrznej i programowania CP/M. Pokazuje jak pisać programy w języku assemblera, korzystające z wbudowanych wywołań systemowych.

CP/M Bible, by Mitchell Waite and John Angermeyer /Indianapolis: Howard W. Sams & Co., Inc., 1984/. Książka ta zawiera spis wszystkich rozkazów CP/M w zunifikowanym, przystępnym formacie.

Innym źródłem informacji o CP/M są magazyny. W nich znajdziesz zarówno nazwy i adresy przedsiębiorstw, które wysyłają sprzęt dla Twojego komputera, jak i handlowe oprogramowanie oraz artykuły o użyciu programów CP/M.

Poniżej podano kilka lepszych magazynów:

Byte, Peterborough, NH. Byte często przeprowadza dyskusje i przeglądy dotyczące hardware i software /oprzyrządowanie i oprogramowanie/.

InfoWorld, Menlo Park, CA. Magazyn ten posiada słągą historię promowania nowości w przemyśle mikro! Każdego miesiąca dostarcza kilku artykułów i ocenia wszystkie produkty. Jego ochrona CP/M stała się nikła wraz ze wzrostem znaczenia IBM PC.

Gazety są także dobrym źródłem wiadomości o CP/M.

Digital Research News są wydawane przez twórców CP/M i opisują ostatnie wydawnictwa i zastosowanie programów.

Life lines jest to magazyn/gazeta wytwarzany przez Lifeboat Associates, który dostarcza wielu pożytecznych porad dotyczących CP/M włącznie z opisami najnowszego ogólnodostępnego oprogramowania.

## 6. GRAFIKA W C-128

~~~~~

W tym rozdziale dowiesz się:

- jakie typy grafiki oferuje Commodore 128
- o historii grafiki w C-128
- jak korzystać z nowych trybów graficznych C-128
- jak działa grafika typu pamięć-ekran - jest to jeden z dwóch najbardziej rozpowszechnionych sposobów przekazywania informacji dla procesora wizyjnego, polegający na tym, że informacja o kolorze każdego pojedynczego punktu ekranu /pixela/ jest zawarta w jednym lub kilku bitach pamięci RAM komputera /przyp.tłum./  
/bit mapped graphic/
- co oferuje tryb 80-kolumnowy jeśli chodzi o grafikę
- jak tworzyć i wykorzystywać SPRITE'y

W rozdziale tym przyjrzymy się wyjątkowej grafice jaką oferuje komputer Commodore 128. Na początek zrobimy krótki przegląd grafiki w systemie C-64. Większa część tego rozdziału poświęcona jest jednak nowej grafice systemu C-128, jako że system CP/M był zaprojektowany z myślą o terminalach pracujących głównie z tekstami, a więc opartych na kodzie ASCII. Nie posiada on rozkazów, które służyłyby do manipulacji grafiką video, nie będziemy więc go omawiać w niniejszym rozdziale. Istnieje jednak możliwość tworzenia grafiki w systemie CP/M za pośrednictwem układu scalonego 8563 generującego obraz przy zastosowaniu kompilatorów innych języków pracujących pod CP/M takich jak BASIC lub C.



## OGÓLNE O GRAFICE C-128

Rozdział ten nie ma zadania wyczerpująco przedstawić technik programowania grafiki dla C-128, a raczej dać ogólne pojęcie o cechach grafiki C-128. Pokróćce wyjaśnić w jaki sposób C-128 wykorzystuje VIC II i nowy układ scalony 8563 dla generowania obrazu na ekranie; ma wreszcie ujawnić, jakie są aktualnie dostępne operacje graficzne w języku BASIC 7.0.

### Co my rozumiemy przez grafikę?

W książce tej mówiąc o grafice - mówimy o wyświetlaniu obrazków w miejsce liter czy cyfr. Obrazem może być cokolwiek co potrafisz przedstawić na ekranie Commodore'a 128, może to być rakieta, robot, jakiś wzór artystyczny, pewne ustawienie odpowiednio zaprojektowanych postaci, bryły geometryczne itd.

### Układy scalone VIC II /6566/ i 8563, odpowiedzialne za generowanie obrazu

Zanim zabrnijemy za daleko w naszej nauce grafiki, musimy przyjąć jedną rzecz do wiadomości. To, co czyni grafikę Commodore'a wyjątkową - w porównaniu z innymi tanimi komputerami domowymi, to sposób, w jaki znaki graficzne powstają i są kontrolowane wewnątrz komputera. W większości komputerów wykorzystywane są popularne układy kontrolowania grafiki/tanie/. Jednakże Commodore 64 i 128 posiadają skomplikowany układ scalony nazywany VIC II. Układ scalony VIC II wywodzi się jeszcze z układu VIC I stosowanego w starym komputerze Commodore VIC 20 i jest właściwie całym mikroprocesorem i video-procesorem. Oficjalnie znany jest jako układ scalony interfejsu odpowiedzialnego za generowanie obrazu, gdyż przekształca on sygnały komputerowe w sygnały monitorowe lub telewizyjne. To właśnie ten szybki i sprytny układ scalony VIC II dodaje wyrazistości graficznej jaką oferuje C-64 i C-128. Commodore 128 po-

siada nie tylko VIC II, lecz również nowy układ scalony, odpowiedzialny za generowanie obrazu 80-kolumnowego - 8563. Układ ten udostępnia C-128 grafiką tego typu, jakiego wymagają użytkownicy profesjonalni i 80-kolumnowy barwny tekst. Gdy będziesz czytać ten rozdział, zachowaj w pamięci, że wszystko, co robisz z BASIC'iem ostatecznie sprowadza się do wysyłania instrukcji do układów scalonych VIC II lub do 8563 i że to właśnie te układy sprawiają, że grafika działa tak dobrze.

### Jakiego typu grafiki oferuje nam Commodore 128?

System C-128 oferuje te same tryby graficzne, co jego poprzednik C-64, a ponadto - co istotne - kilka nowych. Istnieje sześć różnych trybów wyświetlania i mamy do dyspozycji szesnaście kolorów. Mamy nowy tryb 80-kolumnowy. Mamy ponadto osiem programowalnych ruchomych sprite'ów. Dostęp do tych trybów, jak się wkrótce przekonamy, został znacznie ułatwiony.

Tak jak Commodore 64 również Commodore 128 posiada 40-kolumnowy tryb tekstowy, znaki graficzne programowalne w czterech kolorach. Ma taką samą grafikę i ten sam zestaw znaków tekstowych. C-128 ma również rozdzielczość 320 x 200 punktów w trybie graficznym. Ten nowy tryb graficzny w Commodore 128 jest znacznie łatwiejszy i prostszy w użyciu z poziomu BASIC'a.

### Nowy 80-kolumnowy tryb graficzny

Commodore 128 oferuje nowy tryb graficzny, jakiego nie ma w Commodore 64, tj. tryb 80-kolumnowy. Jak zaznaczyliśmy we wstępie i w rozdziale poświęconym CP/M, tryb 80-kolumnowy tworzony jest przez video-procesor 8563. Układ ten ma cechy niemalże identyczne z zestawem układów odpowiedzialnych

za kolor stosowanych w karcie graficznej w IBM/PC tj. rozdzielczość 640 x 200 punktów i 16 kolorów. Ten układ scalony umożliwia wprowadzenie dwukrotnie większej rozdzielczości poziomej w porównaniu z grafiką Commodore 64 /640 punktów vs 320/ i dwa razy więcej znaków /80 vs 40/. Układ ten wymaga 80-kolumnowego monitora i nie będzie współpracował z normalnym odbiornikiem telewizyjnym.

Tryb 80-kolumnowy oferuje ponadto dynamiczny zbiór znaków co oznacza, że kształty znaków ukazujących się na monitorze przechowywane są w pamięci RAM. To z kolei oznacza, że możesz sam tworzyć własne znaki, a nawet grafikę. W BASIC'u 7.0 nie ma rozkazów graficznych, które kontrolowałyby układ 8563, a zatem będziesz musiał posłużyć się poleceniami BASIC'a: PEEK i POKE aby bezpośrednio manipulować rejestrami układu 8563. Może to uczynić Twoje programy bardziej skomplikowanymi, lecz pozwala za pomocą BASIC'u wykonywać grafikę profesjonalną. Z 80-kolumnowym układem scalonym i z jego dużą rozdzielczością możesz śmiało oczekiwać, że ujrzysz bardzo szczegółowe programy graficzne, a zwłaszcza programy profesjonalne o wysokiej jakości.

#### Nowa dodatkowa pamięć

C-128 ma pojemność pamięci dwukrotnie większą od C-64. Ta dodatkowa pamięć jest bardzo potrzebna w grafice, wielkość pamięci ma dla programisty decydujące znaczenie: decyduje o wielkości i o stopniu złożoności programu, jaki może być napisany. Problem polega na tym, że informacja graficzna jaką widzisz na ekranie Commodore'a 128, w rzeczywistości jest przedstawiana w pamięci komputera jako seria bajtów. Gdy chcemy rysować bardziej skomplikowane i coraz bardziej szczegółowe obrazy na ekranie C-128, zaczynamy wykorzystywać coraz to większą część pamięci RAM. Dla przykładu: w grach stosowana jest popularna technika przechowywania w pamięci kilku różnych obrazów graficznych naraz. Następnie sprawia-

jąc, że VIC II szybko je zamienia, uzyskujemy efekt animacji /jak to ma miejsce w grach zręcznościowych/. Ponieważ jednak każdy z tych obrazów pochłania pewną ilość pamięci, ostatecznie dochodzimy do granicy wyznaczającej ile danych /obrazów/ może być przechowywanych w pamięci.

Dodatkowe 64 kB pamięci Commodore'a 128 oznacza, że programy mogą mieć bardziej skomplikowaną szatę graficzną, wymagającą większej pamięci. Pamięć może przechowywać więcej scan graficznych, a sama grafika może posiadać większą głębię, być bardziej realistyczna. Będziesz mógł zobaczyć tu gry o wiele bogatsze od jakichkolwiek innych, z różnymi wariantami i urozmaiceńiami. To samo będzie się odnosiło do użytkowników definiowanych zbiorów znaków i SPRITE'ów, które opiszemy niebawem.

#### Nowe rozkazy graficzne w BASIC'u

Zasadnicza zmiana potencjalnych możliwości nowego systemu C-128 musi mieć wpływ na to, jaki jest dostęp do grafiki dla osoby programującej w BASIC'u oraz jakie są możliwości manipulowania tą grafiką.

W Commodore 64 programista zmuszony był korzystać z rozkazów PEEK i POKE gdy chciał manipulować grafiką /krótko mówiąc rozkaz POKE służył zmianie zawartości komórki pamięci, a rozkaz PEEK odczytaniu tej zawartości/. Nanoszenie jednego punktu przy użyciu tych haseł jest procesem co najmniej karkołomnym. Po pierwsze, musisz stwierdzić gdzie w pamięci zlokalizowany będzie bit odpowiedzialny za ów punkt /zobaczmy później, że bity i punkty mają wiele wspólnego/. Następnie musisz zastosować skomplikowaną kombinację poleceń PEEK i POKE, aby zmienić ten pojedynczy bit, nie zmieniając przy tym bitów sąsiadujących z nim. Rozkaz POKE miał też inne zastosowanie: zmieniał on sposób, w jaki VIC II traktował przy-

dzielony mu obszar pamięci. Pewne adresy w VIC II zezwalałyby na zmianę miejsc, gdzie układ poszukiwał pamięci ekranu i sposobu, w jaki interpretował tę pamięć w programach do C-64 często widywało się hasła tego rodzaju /prosta zmiana trybu na wielobarwny/:

POKE 5327#, PEEK /5327#/ OR 16

W ten sposób, bez odpowiedniego zrozumienia układu scalonego VIC II i bez wiedzy o tym, iż adres 5327# dotyczy rejestru graficznego w układzie scalonym, nikt - z wyjątkiem ekspertów - nie był w stanie w pełni wykorzystać możliwości graficznych. Programiści najczęściej nie byli odpowiednio uświadomieni! Moglibyśmy wprowadzić przyczołczyć długą opowieść o tym, dlaczego w Commodore zawarto takie rozwiązania, ale ograniczymy się do podania tylko jednej przyczyny: chodziło o minimalizację ceny Commodore'a 64. Dla usprawnienia grafiki można było dodać kilka rozkazów do BASIC'a w Commodore 64, ale pochłaniałoby to więcej pamięci i wymagało odpowiednio wyższej ceny.

Commodore 128 zmienił to wszystko. Teraz mamy 128 K RAM i sporą grupę rozkazów w BASIC'u do obsługi grafiki. W pozostałej części tego rozdziału opiszemy te rozkazy oraz wykazemy, jak znacznie wpłynęły one na wzrost efektywności programów. Nadal można stosować polecenia PEEK i POKE; w niektórych przypadkach ich użycie będzie uzasadnione. Ogólnie rzecz biorąc, nowe rozkazy graficzne BASIC'u znacznie upraszczają pracę programisty grafiki. Te właśnie rozkazy wraz z poleceniami nowego BASIC'u 7.0 stworzą nową generację fascynujących gier i programów szkolnych i profesjonalnych, które korzystają z grafiki.

#### Skąd wzięły się nowe rozkazy BASIC?

Większość usprawnień technologicznych bierze swój początek z wcześniejszych projektów i w tym względzie rozkazy,

które składają się na nowy tryb C-128 nie stanowią wyjątku. W roku 1983 Commodore zaprezentował cartridge\*, nazwany Super Expander 64. Podłączony do odpowiedniego gniazda Commodore'a 64, cartridge ten dodaje cały zbiór nowych rozkazów graficznych i dźwiękowych oraz klawiszy funkcyjnych do istniejącego już, rezydentnego języka BASIC C-64. Wprowadzenie rozkazów te znacznie ulepszyły BASIC, lecz SUPER EXPANDER nie zdobył wielkiej popularności na rynku.

Co zrobił Commodore w nowym systemie C-128? Wziął rozkazy z SUPER EXPANDER'a 64 i umieścił je w ROM języka BASIC 7.0, czyniąc je w ten sposób standardowymi. Każdy, kto kupi C-128 otrzymuje te rozszerzone polecenia graficzne. Teraz możemy oczekiwać, iż liczne rzesze programistów i pisma poświęcone analizom systemu C-128 zrewolucjonizują zastosowanie tych rozkazów.

Teraz, gdy przedstawiłem już nieco historii związanej z grafiką w Commodore 128, przyjrzyjmy się bliżej i dokładniej dostępnym trybom. Zwróćmy uwagę, co one oferują. Pozostała część tego rozdziału podzielona jest na trzy części: grafikę znakową /blokową/, grafikę maskowaną bitami oraz grafikę sprite'ów.

#### OGÓLNIŃ O TRYBACH C-128

Commodore 128 posiada sześć różnych trybów wyświetlania. Poniżej podamy krótką charakterystykę każdego z tych trybów.

---

\* cartridge - płytka zawierająca pamięć ROM, w której znajduje się np. program, rozszerzenie języka BASIC itp. Istotną cechą cartridge'u jest to, że po jego włączeniu w odpowiednie gniazdo komputera i włączeniu tegoż do sieci, program zapisany na cartridge'u jest natychmiast wykonywany /nie jest wymagane wczytywanie z taśmy lub dysku/.

### Tryb graficzny blokowy ze standardowym 40-kolumnowym tekstem

Tryb ten jest identyczny z trybem tekstowym w Commodore 64. Dopuszcza on maksymalnie 1000 znaków tekstowych lub graficznych na ekranie, przy formacie 25 wierszy po 40 znaków w wierszu. W trybie tym każdy znak zdefiniowany jest w macierzy 8 x 8 punktów /pixeli/ i może mieć dowolny kolor /z 16 możliwości/. Jest to jeden z najpopularniejszych trybów graficznych, gdyż obszary punktowe 8 x 8 są łatwe do kontrolowania. Niebawem będziemy mieli dużo do powiedzenia na temat tego trybu.

### Standardowy tryb bitowy /bit mapped graphics/

O standardowym trybie bitowym często mówi się, że jest to tryb o dużej rozdzielczości. Ma on na celu uzyskanie najwyższego stopnia precyzji i szczegółowości. W trybie tym ekran traktowany jest jak układ w kształcie prostokąta, z 320 punktami w linii poziomej i 200 w linii pionowej. W trybie 40-kolumnowego tekstu nie mamy bezpośredniego dostępu do poszczególnych punktów i pod tym względem tryb bitowy góruje nad nim. Jeśli idzie o kolor, możesz wybrać jakiegokolwiek dwie barwy z 16 możliwych dla obszaru 8 x 8 odpowiadającego pozycji znaku w trybie tekstowym 40-kolumnowym. Jedna z nich będzie tłem, druga pójdzie na pierwszy plan. W przeszłości dostęp do trybu bitowego na C-64 wymagał skomplikowanego zastosowania PEEK'ów i POKE'ów. Obecnie dostęp do tego trybu jest prosty i logiczny - wymaga tylko rozkazów BASIC 7.0. W rzeczywistości, jak się wkrótce przekonamy, znaczna część rozkazów graficznych nowego BASIC'u 7.0 jest właśnie specjalnie przeznaczona do operowania trybem bitowym.

### Standardowy tryb bitowy z oknem tekstowym /podzielony ekran/

Tryb podzielonego ekranu jest swoistą mieszanką trybu bitowego i standardowego, 40-kolumnowego tekstu. Kiedy zażą-

dasz tego właśnie trybu od BASIC'a, górne 75% ekranu będzie w trybie standardowej grafiki bitowej, a dolne 25% - w standardowym trybie 40-kolumnowego tekstu. W ten sposób otrzymasz 5 linii tekstu - od linii 19 począwszy w dół - do 24. Daje to pionową rozdzielczość trybu bitowego, rzędu 160 pikseli<sup>\*</sup> /200 linii - 5, co 8 punktów na linię/. Możesz zmienić linię, od której rozpoczyna się tryb tekstowy, przez dodanie parametru s /zawartego w przedziale od 0 do 24/. Tryb podzielonego ekranu jest pożyteczny, gdy chcesz wydrukować informację o zdarzeniach przy dużej rozdzielczości. Technika ta znajduje częste zastosowanie w grach, gdzie na górną część ekranu o dużej rozdzielczości przekaże obraz przestrzeni kosmicznej widzianej ze statku kosmicznego, a w dolnej części ekranu znajdują się komunikaty - w formie tekstu - ze statku macierzystego. Tak podzielony ekran można też wykorzystać w grafice profesjonalnej, umieszczając np. na jednej części wykresy, a na drugiej przetworzony dokument /tekst/.

### Wielobarwny tryb bitowy /multicolor/

W trybie wielobarwnym rezygnuje się z maksymalnej rozdzielczości poziomej na rzecz zwiększonych zdolności kolorystycznych. Rozdzielczość poziomu ekranu zmniejszona jest o połowę /do 160 punktów/, podczas gdy rozdzielczość pionowa pozostaje niezmienną /200 punktów/. Każdy punkt jest dwukrotnie szerszy od standardowego punktu bitowego - przez co zamiast małej kropki przyjmujemy wydłużony prostokąt. Teraz jednak możemy uzyskać 4 kolory w każdym z obszarów punktowych 8 x 8. Te cztery kolory to: kolor pierwszego planu, kolor tła oraz dwa nowe, nazwane multicolor 1 i multicolor 2. Wielobarwny tryb bitowy można też uzyskać w trybie podzielonego ekranu. Tryb wielobarwny nie znajduje większego zastosowania w oprogramowaniu Commodore'a, ponieważ duże

\* piksel - najmniejszy wyróżniony element ekranu /kropka, punkt/

rozmiary pikseli powodują, że obraz wygląda dość surowo, jak-by był niewykończony.

#### Tryb tekstowy 80-kolumnowy

W trybie 80-kolumnowym wykorzystujemy układ scalony 8563; możemy uzyskać 80 znaków na linii, co przy 25 liniach daje w sumie 2000 znaków w trybie tekstowym. Każdy ze znaków może mieć jedną z 16 barw. Układ scalony 8563 jest właściwie układem bitowym z dynamicznym zbiorem znaków. Commodore nie podaje jednak bezpośredniego sposobu na dotarcie do poszczególnych bitów obszaru /np. przez rozkazy BASIC'a/. Niewątpliwie w niejednej książce znajdziemy sposoby na dotarcie do tego trybu graficznego poprzez PEEK i POKE, a najprawdopodobniej pojawi się jeszcze odpowiednie rozwinięcie BASIC'a, które umożliwi dojście bezpośrednie. Byłoby to bardzo pożądane, ponieważ rozdzielczość graficzna układu 8563 wynosi 640 x 200 punktów, to jest dwukrotnie więcej niż oferuje VIC II. Układ ten ma w istocie taką samą rozdzielczość koloru i grafiki, jak IBM PC, a więc jeśli tylko BASIC zostanie rozwinięty w odpowiedni sposób, będzie można przenieść szereg rozwiązań z IBM PC na C 128.

#### Tryb graficzny znakowy /blokowy/

Zanim nauczymy Was nowych rozkazów graficznych i BASIC'a 7.0 działających na ekranie bitowym, pożytecznym i przydatnym może się okazać wytłumaczenie jednej rzeczy: jak tryb 40-kolumnowego tekstu jest wykorzystywany do grafiki blokowej. Tryb ten zapewnia wiele efektywnych możliwości i jakakolwiek analiza nie byłaby pełna bez jego omówienia. Ponieważ istnieje tryb podzielonego ekranu, "mieszanka" grafiki blokowej i bitowej jest możliwa.

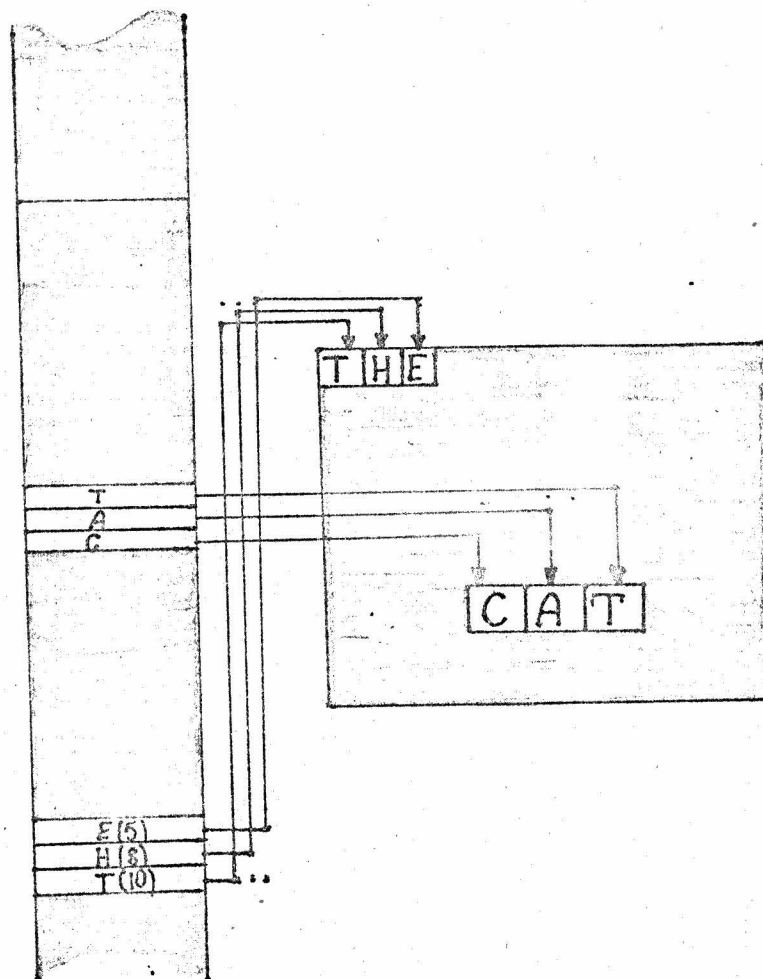
Grafika blokowa /lub znakowa/ osiągalna jest przy wykorzystaniu C-128 w trybie 40-kolumnowego tekstu. Gdy tylko włą-

czamy Commodore'a, zaczyna on pracować właśnie w tym trybie, a później i tak można do niego powrócić przy pomocy rozkazu GRAPHIC 0. Wprawdzie tryb 40-kolumnowy może się wydawać najmniej atrakcyjnym w wykorzystywaniu grafiki, lecz właśnie ten tryb jest najpopularniejszy w istniejącym dziś rozwiniętym oprogramowaniu profesjonalnym. Jest to tryb, w którym napisano większość programów dla Commodore'a 64.

W trybie 40-kolumnowym można zastosować szereg technik w tworzeniu grafiki. Wszystkie te techniki wykorzystują grafikę znaków: użycie specjalnych obrazów punktowych 8 x 8 w tworzeniu obrazów większych. Gdy korzystamy z BASIC'a, najprostszym sposobem na wykonanie grafiki w tym trybie jest umieszczenie kombinacji znaków graficznych klawiatury wewnątrz rozkazów PRINT. Tak jak i C-64, C-128 posiada pełny zestaw znaków graficznych widocznych na każdym klawiszu. W rzeczywistości istnieją cztery zbiory znaków graficznych, z których można skorzystać. Jeżeli naciśniesz klawisze Shift lub C= i odpowiednie klawisze graficzne, odpowiednie znaki graficzne pojawią się na ekranie. Możesz zaprojektować jakiś przedmiot, lub kształt czy kontur, posługując się klawiszami graficznymi. Aby zaprojektować przedmiot na ekranie, posługiwać się będziesz klawiaturą oraz klawiszami sterującymi kursorem. Przedmiot ten zostanie następnie "uchwycony" rozkazem PRINT. Dla przykładu, aby przedstawić obraz samochodu możesz posłużyć się rozkazami BASIC-a:

```
100 PRINT "      "
110 PRINT " |      "
120 PRINT " |      "
130 PRINT " 0    0 "
```

Kiedy uruchomisz program, obraz podobny do samochodu pojawia się na ekranie. Możesz napisać taki program, który przesyłałby Twój samochód \* innych miejscach na ekranie tak, że będzie się wydawało, że samochód "jedzie" na ekranie.



Rys. 19 - Kody ekranowe w pamięci odpowiadają wyświetlanym znakom

Ze 128 różnymi znakami do wyboru, tryb ten jest bardzo silny. Kolory można ustalać używając klawisza koloru Commodore'a wraz z klawiszami graficznymi. Przyciśniesz odpowiedni klawisz koloru, który zechcesz i następny znak graficzny wprowadzony do rozkazu PRINT ukazuje się już w tym kolorze. Przyjrzyjmy się teraz bliżej temu trybowi.

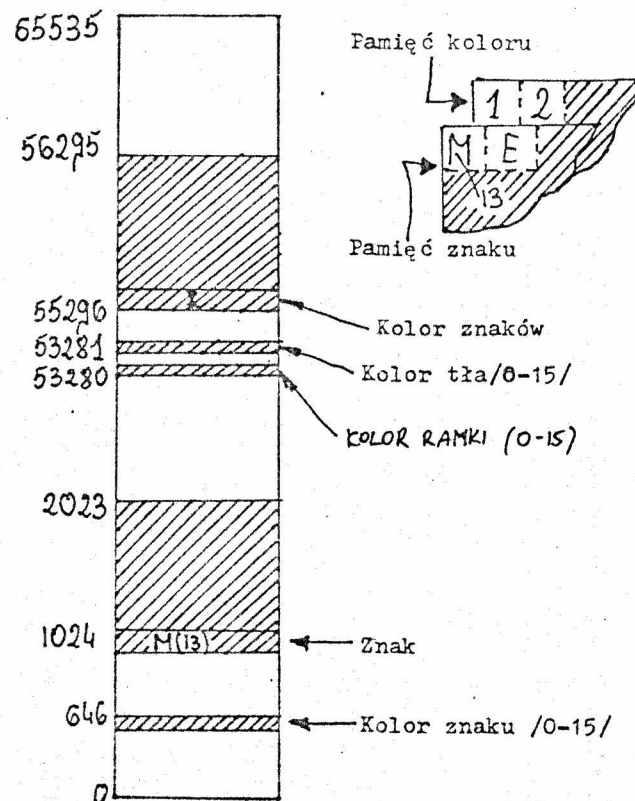
W Commodore 128 wydzielony jest pewien obszar pamięci na przechowywanie znaków i barw wyświetlanych w tym trybie. Znaki, jakie widzisz na ekranie przechowywane są w pamięci w szczególny sposób - jak pokazano na rysunku.

Każdy z możliwych znaków na ekranie jest zdefiniowany w macierzy 8 x 8 punktów i jest reprezentowany przez swój własny numer kodu przechowywany w pamięci. Może to być numer od 0 do 255. Dla przykładu - litera M jest reprezentowana przez kod 13, a znak graficzny piłeczki - przez 61. Układ scalony VIC II zawiera obwody, które wielokrotnie odwołują się do specjalnego obszaru pamięci ekranu, pobierając stamtąd po jednym kodzie znaku, przekształcają go w odpowiednią literę lub znak graficzny formatu 8 x 8, a następnie wyświetlają gotowy już obraz w odpowiednim miejscu na ekranie.

A w jaki sposób nadajemy barwę znakom graficznym?

Na rys. widzimy, że grafice podporządkowane są dwa obszary pamięci. Jeden - to pamięć ekranu, przechowujący body znaków. Drugi - to pamięć koloru, przechowujący barwy każdego bloku /8 x 8 pixel/ ekranu. Pamięć koloru może pomieścić 16 znaków, lecz zamiast kodów znaków zawiera ona numery kolorów od 9 do 15 /tylko niższe 4 bity każdej lokacji są wykorzystywane/. Każda komórka pamięci koloru odpowiada komórce pamięci w pamięci ekranu. Tak więc, kiedy zmieniasz kod koloru w pamięci koloru, w rzeczywistości zmieniasz kolor pewnego znaku na ekranie. Na rysunku tym widać również komórki pamięci stosowane do kontroli kolorów ramki, znaków





Rys. 20 - obszary pamięci, ekranu i koloru

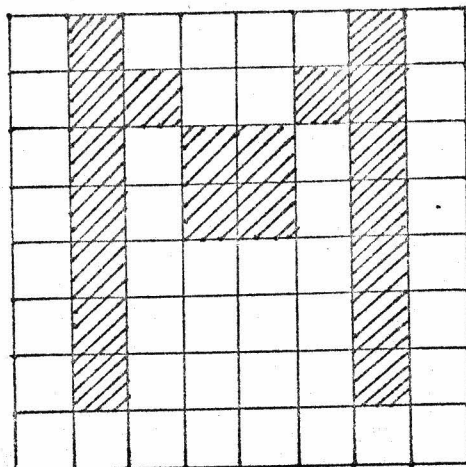
i koloru tła. Informacja o adresach tych komórek przyda Ci się, jeżeli zamierzasz coś wpisywać do nich instrukcją POKE. Komórki pamięci tekstu i koloru zorganizowane są w takim samym formacie, a więc możesz sobie wyobrazić, iż te dwa obszary pamięci są jak dwie "płaszczyzny" nakładające się na siebie. To także jest pokazane na rys.

Zauważ, że choć na ekranie 40 x 25 jest tylko 1000 znaków, istnieje 1024 komórek pamięci przeznaczonych dla tekstu. Dodatkowe 24 zanki wykorzystywane są do przechowywania specjalnych danych systemowych przez C128.

Jest jeszcze trzeci obszar pamięci, obszar krytyczny, jeśli wziąć pod uwagę maksymalizację użytkowania trybu 40-kolumnowego tekstu. Obszar ten przechowuje zbiór definicji znaków. Gdy układ scalony VIC II pobierze kod z naku z pamięci ekranu, wykorzystuje go do "podpatrzenia" konkretnego wzorca układu punktów tworzących dany znak przechowywanego w obszarze definicji znaków. Wzorec ten mówi układowi scalonemu VICII w jaki sposób ma narysować na ekranie dany znak. Dla przykładu wzorec dla litery M przedstawiony jest na rys.

Obszary ciemne odpowiadają punktom, które będą miały kolor przerwszoplanowy, a obszary jasne - punktom w kolorze tła. Tak jak i z pamięcią ekranu, można sprawdzić, aby pamięć zbioru definicji znaków zawierała takie zestawy wzorów, jakie zapragniesz. Zbiór może zawierać znaki narodowe lub graficzne, które, gdy zbierzesz je razem, mogą stworzyć obrazy o dużej szczegółowości. Dostępnych jest kilka popularnych edytorów znaków dla C64, ułatwiających tworzenie własnych zbiorów znaków.

Znaki z klawiatury C-128 nie są odpowiednio elastyczne na to, aby umożliwić tworzenie prawdziwie szczegółowych scen. Najciekawsze programy korzystające z trybu 40-kolumnowego tekstu wykorzystują zbiory wcześniej zaprogramowanych znaków.



Rys. 21 - wzorzec litery "M" w matrycy 8 x 8

W praktyce można jednocześnie mieć w pamięci różne zbiory definicji znaków i sprawiać, aby VIC II przełączać się między nimi. Ponadto umiejscowienie pamięci ekranu /część pamięci, która przechowuje kody znaków, które komputer ma umieścić na ekranie/ w trybie 40-kolumnowym nie jest stałe, jak to ma miejsce w większości komputerów. Można sprawić, aby układ scalony VICII nagle "przeniósł się" do innego obszaru pamięci i aby go traktował jak nową pamięć ekranu. Pamięć ta zajmuje jedynie 1 k bajtów RAM-u, a więc można mieć wiele uprzednio zdefiniowanych ekranów 40-kolumnowych "czekających" w pamięci. Jest to wykorzystywane w wielu grach, w których użytkownik ma do czynienia z różnymi "pomieszczeniami" lub "grotami", które należy zbadać. Faza wstępna programu tworzy ekrany, a następnie układ VIC II kierowany jest na początek odpowiedniego obszaru pamięci.

Oprócz kontrolowania wydruku na ekranie znaków w trybie 40-kolumnowym rozkazami PRINT, możemy też bezpośrednio wpisać je /instr.POKE/ do pamięci ekranu. Wykorzystanie tego sposobu nie jest proste, a przesuwanie pamięci ekranu i użycie własnych zdefiniowanych znaków, wymaga dogłębnej znajomości rejestrów układu VIC II. Jeśli chcesz to bliżej poznać, przejrzyj bibliografię na końcu tego rozdziału, aby znaleźć odpowiednią literaturę.

Tryb 40-kolumnowego tekstu jest szczególnie przydatny w grach i w programach edukacyjnych, a także w prezentacjach tekstu czy menu. Szczególnie jeden typ gier /gry symulacyjne/ wykorzystuje przeddefiniowany zbiór znaków i dużą bazę danych, zawierającą ściśle rozmieszczone kody tych znaków, aby zasymulować lot nad pewnym obszarem lądu. Z "nowych" znaków tworzone są drzewa, jeziora, linie kolejowe, domy, fabryki, stanowiska artyleryjskie. Z naszego punktu widzenia - z bombowca widzimy cień samolotu na ziemi. Opcja układu scalonego VIC II nazwana "smooth scrolling" umożliwia przesuwanie informacji na ekranie o jeden pixel /punkt/ w linii pio-

nowej lub poziomej. Jest to wykorzystane, aby sprawić, że lecimy nad prawdziwym obszarem ładu. W rzeczywistości różne kody znaków są na przemian wpisywane bądź usuwane z pamięci ekranu.

### Tryb bitowy

Teraz, gdy rozumiesz już tryb grafiki blokowej, łatwiej Ci będzie zrozumieć tryb bitowy. Jest to tryb umożliwiający korzystanie wielu nowych rozkazów graficznych, zawartych w BASIC-u 7.0 /nowe rozkazy dotyczące sprite'ów działają zarówno w trybie bitowym jak i tekstowym/. Trybu bitowego używa się przy wykreślaniu rozmaitych wykresów, rysowaniu linii, kontrolowaniu poszczególnych punktów, wyświetlanych na ekranie. W trybie tym każdemu punktowi /zw. pixel/ przypisany jest jeden bit w pamięci. Jeżeli bit ten jest jedynką, punkt do którego jest on przypisany jest "włączony". Jeżeli natomiast ma wartość "zero" punkt jest "wyłączony". "Włączony" punkt pojawia się na ekranie w kolorze pierwszoplanowym. "Wyłączony" punkt ma kolor tła.

Główną wadą trybu bitowego jest to, że zajmuje on znaczną ilość pamięci, a udostępnia tylko jeden kolor plus tło /lub 4 kolory, jeżeli korzystasz z wielobarwnego trybu bitowego, który jednak oferuje tylko 160 punktów w linii poziomej/ dla obszaru 8 x 8 punktów. Dla przykładu, w C128 w trybie bitowym rozdzielczość wynosi 320 i 200, co znaczy, że jest 64000 punktów, z których każdy wymaga 1 bitu pamięci. W jednym bajcie jest 8 bitów, czyli do zapamiętania całego ekranu potrzeba 8000 bajtów. Przypomnijmy, że w trybie 40-kolumnowego tekstu potrzeba było tylko 1000 bajtów na przechowanie informacji z całego ekranu. Użycie trybu bitowego najlepiej jest ograniczać do programów, w których szybkość lub animacja nie stanowią wymogów krytycznych np. do projektowania, CAD/SAM, profesjonalnych wykresów itp. Tryb bitowy

najlepiej sprawdza się przy rysowaniu bardzo szczegółowych obiektów lub skomplikowanych wykresów matematycznych.

### Ustalanie trybu graficznego: rozkaz GRAPHIC

Comodore 128 rozpoczyna pracę w trybie 40-kolumnowego tekstu. Aby "przestawić" go na tryby: bitowy lub bitowy wielokolorowy o dużej rozdzielczości lub aby przełączyć na tryby rozkazu 80-kolumnowe, korzystamy z rozkazu GRAPHIC. Zobaczmy teraz jak to działa. Składnia rozkazu GRAPHIC wygląda następująco: GRAPHIC mode, c.s. "Mode" jest liczbą całkowitą od 0 do 5, określającą odpowiedni tryb, jak to pokazano w tabeli 6-1.

Tabela 6-1 - Tryby graficzne

| Parametr "mode" | O p i s                                   |
|-----------------|-------------------------------------------|
| 0               | Standardowy 40-kolumnowy tryb tekstowy    |
| 1               | Standardowy tryb bitowy                   |
| 2               | Standardowy tryb bitowy z oknem tekstowym |
| 3               | Tryb wielokolorowy /multicolor/           |
| 4               | Tryb wielokolorowy z oknem tekstowym      |
| 5               | 80-kolumnowy tryb tekstowy                |

Parametr C znaczy - oczyść lub nie oczyszczaj ekranu po przełączeniu trybu. Jeżeli C = 1 znaczy to - oczyść, jeżeli C = 0 - nie oczyszczaj. Parametr s ustala linię, na której ma się zaczynać okno tekstowe w trybie podzielonego ekranu. W przypadku nie podania tego parametru jako początek okna tekstowego zostaje przyjęta linia 19, dając 5 linii tekstu w dolnej części ekranu.

Popatrz na przykłady:

GRAPHIC 0,1 : REM 40 kolumnowy tekst, czyszczenie ekranu

GRAPHIC 2,0,15 : REM podzielony ekran, bez oczyszczenia, okno tekstowe od 15 linii

Pierwsze polecenie przełącza nas na tryb 40-kolumnowego tekstu i oczyszcza ekran. Rozkaz drugi przełącza na ekran podzielony nie oczyszczając go, dając 10 linii tekstu /od linii 15 począwszy/ i tryb bitowy z rozdzielczością 320 x 120 punktów.

Aby przejść na tryb bitowy /320 x 200/ i oczyścić ekran, teraz wprowadzić by trzeba do naszego programu rozkaz

GRAPHIC 1,1

Gdy użyty zostanie rozkaz włączający tryb bitowy /GRAPHIC 1-4/ rezerwuje on pewien obszar pamięci na przechowywanie informacji z ekranu. Rozkaz GRAPHIC CLR kasuje ten obszar i przywraca go pamięci /to znaczy, że obszar ten staje się gotowy do zapamiętania kodu programu w BASIC-u/. Jednakże jeżeli przełączysz komputer z powrotem na tryb bitowy i jeśli Twój program za bardzo się rozrósł, możesz otrzymać informację o błędzie. Teraz kiedy już z pomocą rozkazu GRAPHIC wybraliśmy tryb bitowy, zobaczmy jak wybierany jest kolor.

#### Wybieranie kolorów: COLOR

Aby wybrać kolor dla pierwszego planu, tła i ramki na ekranie posługujemy się poleceniem COLOR. Składnia w tym przypadku wygląda następująco: COLOR źródło, kolor gdzie "źródło" - to ta partia ekranu, którą chcesz zabarwić /pierwszy plan, tło lub ramka dla konkretnego trybu/, a "kolor" to odpowiedni kod koloru /1-16/ dla tego koloru, który zechcesz przyisać onemu "źródłu".

Tabela 6-2 podaje kolory i ich kody stosowane w Commodore 128.

Tabela 6-2 - kody kolorów dla rozkazu COLOR

| Kod koloru | K o l o r         |
|------------|-------------------|
| 1          | czarny            |
| 2          | biały             |
| 3          | czerwony          |
| 4          | cyjan             |
| 5          | purpurowy         |
| 6          | zielony           |
| 7          | niebieski         |
| 8          | żółty             |
| 9          | pomarańczowy      |
| 10         | brązowy           |
| 11         | żółto-zielony     |
| 12         | różowy            |
| 13         | niebiesko-zielony |
| 14         | jasno-niebieski   |
| 15         | ciemno-niebieski  |
| 16         | jasnozielony      |

Tabela 6-3 wymienia sześć źródeł kolorów i podaje odpowiadające im kody stosowane w rozkazie COLOR. Trzeba pamiętać na tym, że różnym trybom możesz przypisać różne kolory.

Poniżej mamy kilka przykładów na to, jak stosować rozkaz COLOR:

Tabela 6-3 - kody źródeł kolorów dla rozkazu COLOR

| Kod | Dotyczy                                            |
|-----|----------------------------------------------------|
| 0   | Tło w trybie 40-kolumnowego tekstu                 |
| 1   | Pierwszy plan, tryb bitowy                         |
| 2   | Dodatkowy kolor nr 1 w trybie wielokolorowym       |
| 3   | " " nr 2 " "                                       |
| 4   | Ramka we wszystkich trybach                        |
| 5   | Kolor znaków w trybie 40 lub 80 kolumnowego tekstu |
| 6   | Tło w trybie 80-kolumnowego tekstu                 |

10 COLOR 0,15 : REM tło ciemnoniebieskie

20 COLOR 1,5 : REM pierwszy plan w trybie bitowym purpurowy

30 COLOR 4,11 : REM ramka żółto-zielona

W przykładzie tym zmieniamy wstępnie ustalone przez Commodore'a kolory tła, granicy i znaków. Linia pierwsza ustala tło tekstu i grafiki na kolor ciemnoniebieski. Linia druga ustala kolor pierwszego planu dla trybu bitowego na purpurowy, a linia ostatnia ustala kolor ramki dla wyświetlonego tekstu i trybu bitowego na żółto-zielony.

#### Inne rozkazy zorientowane na kolor

Rozkaz SCNCLE używany jest do oczyszczenia ekranu w jakimkolwiek trybie. Dla przykładu, SCNCLE oczyszcza tryb 40-kolumnowego tekstu, a SCNCLE 2 oczyszcza standardowy tryb bitowy /przez "oczyszczanie" rozumiemy sprawdzenie wszystkiego do koloru tła/. Jeżeli Twój program potrzebuje dowiedzieć się, jaki kolor był przypisany odpowiedniemu "źródłu" np. barwa pierwszego planu w trybie bitowym, możesz wykorzystać funkcję RCLR/N/. N oznacza liczbę reprezentującą tryb źródłowy, który

chcesz sprawdzić. Funkcja ta "oddaje" numer koloru między 1 a 16. Podobną nową funkcją jest RGR, która po prostu zwraca bieżący numer trybu - od 0 do 5.

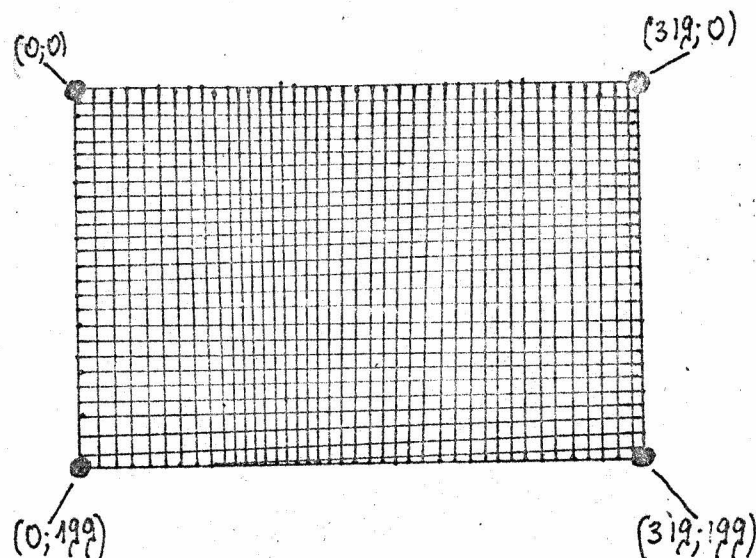
#### Układ współrzędnych dla ekranu w trybie bitowym

Istnieje kilka poleceń, które pomogą Ci narysować obrazki na ekranie w standardowym trybie graficznym /bitowym/. Odpowiednie rozkazy urozniwiają nanoszenie punktów, rysowanie linii, prostokątów, okręgów i wielokątów.

Rys. przedstawia układ współrzędnych dla trybu bitowego. Jak widać lewy górny róg to pozycja wyjściowa /0,0/, a prawy dolny - to maksymalna wartość w układzie współrzędnych /319,199/. Na tym obrazie każdy punkt ma wyznaczoną parę współrzędnych x i y. Dla przykładu, punkt w środku obrazu znajduje się w pozycji /160,100/. Próba naniesienia punktu, którego wartości współrzędnych przekraczają granice ekranu spowoduje, że komputer zatrzyma się i poda komunikat o błędzie składniowym. Dla matematyków numerowanie osi może wydawać się odwrócone "do góry nogami" /0 na górze, maksimum na dole/ ale to jest typowy dla komputerów osobistych sposób "potraktowania" ekranu.

#### Lokowanie kursora graficznego: rozkaz LOCATE

Zanim pokażemy Wam jak rysować na ekranie dużej rozdzielczości /w trybie bitowym/ chcielibyśmy, abyście zrozumieli coś, co nazywamy kursorem graficznym w skrócie PC /Pixel Cursor/. Pixel odpowiada położeniu punktu na ekranie. Kursor graficzny podobny jest do migającego kursora C-128 w trybie tekstowym, który pokazuje gdzie pojawi się następny znak. Kursor graficzny wskazuje gdzie umieszczony będzie następny punkt, ale nie ujrzymy tym razem błyskającego /migającego/ punktu. W rozkazach BASIC-u, kiedy opcjonalne współrzędne nie są podane, w ich miejsce zostają podstawione współrzęd-



Rys. 22 - układ współrzędnych trybu bitowego

ne kursora graficznego /PC/. Na przykład, jeżeli użyjemy rozkazu CIRCLE /który wkrótce poznamy/ bez podania współrzędnych środka rysowanego okręgu, środkiem tym stanie się automatycznie aktualna pozycja kursora graficznego /PC/. Jak możemy ustalić pozycję PC? Rozkaz LOCATE służy do przeniesienia PC do punktu o podanych współrzędnych. Składnia rozkazu LOCATE jest następująca:

LOCATE współrzędna x, współrzędna y

Poniżej mamy przykład polecenia przenoszącego PC na środek ekranu:

`10 LOCATE 160, 160`

Zwróć uwagę, że nie zobaczysz po wykonaniu tego polecenia, dopiero później, gdy spróbujesz wykreślić coś bez podawania współrzędnych początku, kreślenie rozpocznie się od punktu, do którego przeniosłeś kursor graficzny /PC/. Rozkaz ten może być również użyty do kontrolowania pozycji drukowania tekstu na ekranie w trybie bitowym, jak to zobaczymy, gdy nauczymy się korzystać z rozkazu CHAR. Teraz jednak zobaczymy jak możesz narysować na ekranie graficznym.

#### Rysowanie linii i punktów: rozkaz DRAW

Wiesz już jak wybierać różne tryby graficzne, ustalać kolory różnych obszarów i przenosić kursor graficzny, nauczmy się więc teraz jak w prosty sposób można kreślić punkty i linie. BASIC 7.0 Commodore'a 128 wykorzystuje rozkaz kreślący linie o formacie takim, jaki możesz znaleźć również w Apple II BASIC. Rozkazem tym jest DRAW, a jego składnia wygląda następująco:

`/3/ DRAW źródło koloru, a1, b1, TO a2, b2, ...`

gdzie "źródło koloru" może przyjmować wartość 1-3, określającą z jakiego źródła będzie pobrana informacja o kolorze określonego punktu linii. Dla przykładu, jeżeli "źródło ko-



loru" ma wartość 1 oznacza to, że kreślenie będzie wykonywane kolorem pierwszego planu /foreground/ standardowego trybu bitowego; dwójka oznacza dodatkowy kolor nr 1 w trybie wielokolorowym /multicolor 1/, trójka natomiast dodatkowy kolor nr 2 w tym samym trybie /multicolor 2/. Wartości  $a_1$  i  $b_1$  reprezentują współrzędne  $x, y$  początku kreślonego odcinka, a  $a_2$  i  $b_2$  współrzędne jego końca. Tak więc polecenie:

100 DRAW 1,10,10 TO 50,50

spowoduje narysowanie linii rozpoczynającej się w punkcie o współrzędnych 10,10 i kończącej w punkcie 50,50 w kolorze pierwszego planu dla standardowego trybu bitowego. Wielokropki na końcu opisu składni rozkazu DRAW mówią nam, że można kontynuować podawanie współrzędnych, do których mają być kreślone następne linie, a tym samym tworzyć skomplikowane figury geometryczne. Na przykład rozkazy:

100 DRAW 1,100,100 TO 160,40 TO 220,100 TO 100,

100 : REM trójkąt

110 DRAW 1,0,0 TO 0,199 TO 319,199 TO 319,0 TO 0,

0 : REM ramka

narysują na środku ekranu trójkąt, a następnie ramkę wokół krawędzi ekranu.

Można również użyć rozkazu DRAW do "wykreślenia" pojedynczego punktu, opuszczając po prostu słowo TO i następujące po nim współrzędne końca. Pomyśl o tym jako o linii o długości jednego punktu /pixela/. Tak więc rozkazy

30 FOR X = 1 TO 319

40 Y = /X~/ "do potęgi" / 2 / / 490 : REM wykres krzywej  $y = x^2$

50 DRAW 1,X,Y : REM zwróć uwagę na brak TO

60 NEXT X

spowodują narysowanie wykresu funkcji  $y = x^2$  /oczywiście możesz skorzystać z polecenia DRAW dla dowolnego równania/.

## Kreślenie wypełnionych i pustych prostokątów: rozkaz BOX

Rozkaz BOX umożliwia Ci wykreślenie konturu prostokąta dowolnej szerokości i wysokości i w dowolnym miejscu na ekranie w trybie bitowym. Co więcej, możesz polecić, aby rysowany kontur został wypełniony aktualnym kolorem źródła, dając zamalowany prostokąt. Efektywność rozkazu BOX została zwiększona poprzez dodanie możliwości określenia kąta o jaki ma być obrócony dany prostokąt względem swojego środka zanim zostanie wyświetlony. Cecha ta jest przydatna przy nanoszeniu współrzędnych kolorów /plotting in color coordinates/ dla "grafiki żółwia" /np. LOGO/, symulacji i gier. Składnia rozkazu BOX jest pokazana na rys.

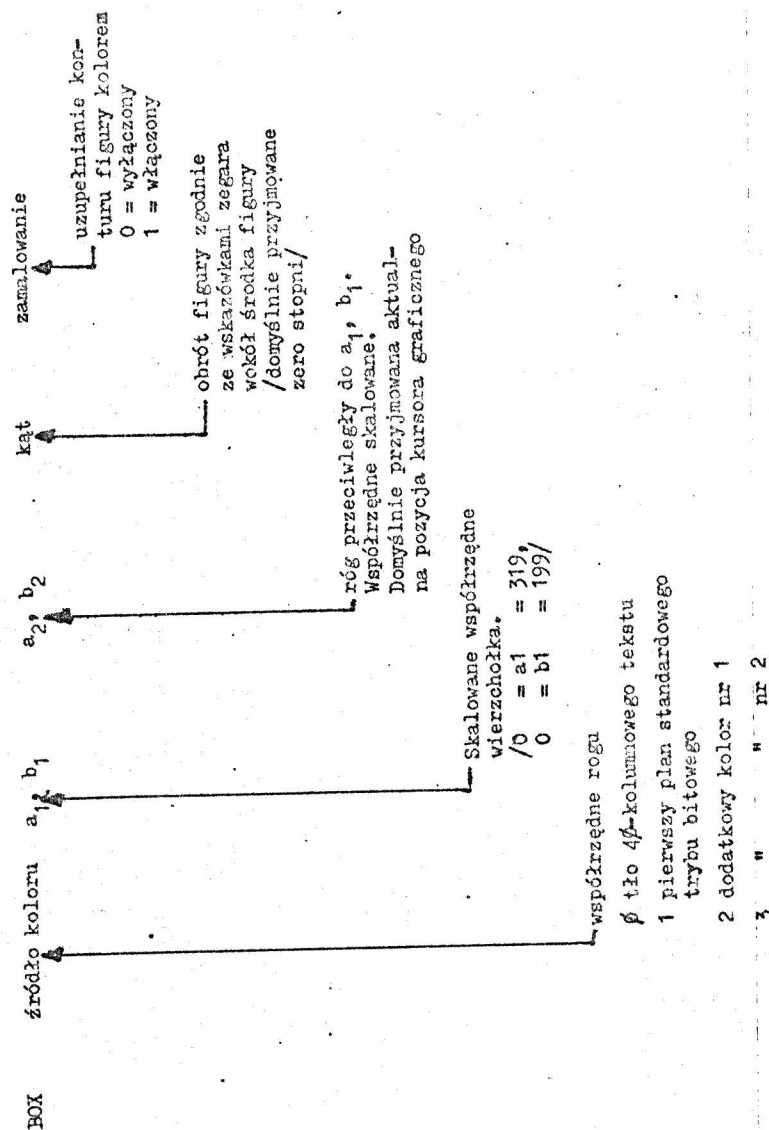
Typowy rozkaz BOX wygląda następująco:

100 BOX 1,10,10,60,60

Pierwszy parametr "źródła koloru" określa z którego ekranu będzie brana informacja o kolorze rysowanego prostokąta; może on przyjmować wartości 0-3. Źródłem tym normalnie jest standardowy tryb bitowy /1/. Jest on również przyjmowany domyślnie w przypadku nie podania parametru "źródło". Źródłami koloru mogą być również: dodatkowy kolor nr 1 /2/, dodatkowy kolor nr 2 /3/ w trybie wielokolorowym/, lub kolor dla w standardowym trybie tekstowym 40-kolumnowym /0/. Następnymi parametrami są: współrzędne lewego, górnego rogu prostokąta / $a_1, b_1$ /, współrzędne prawego, dolnego rogu / $a_2, b_2$ /. W podanym przykładzie rysowany jest prostokąt od współrzędnej 10,10 do 60,60. Prostokąt jest rysowany w standardowym trybie bitowym, ponieważ wybrane zostało źródło koloru 1. Można pominąć parametr wpisując przecinek, który normalnie po nim występuje. Na przykład:

100 BOX,10,10,16,60

używa domyślnej, lub wcześniej ustalonej wartości źródła koloru, która nie jest podana w podanym przykładzie, lecz



Rys.23 - Składnia rozkazu BOX

jest reprezentowana przez przecinek.

Parametr, który następuje po współrzędnych przeciwległego rogu nazywany jest kątem. Kontroluje on o ile stopni prostokąt ma być obrócony przed jego wyświetleniem. Na przykład rozkaz:

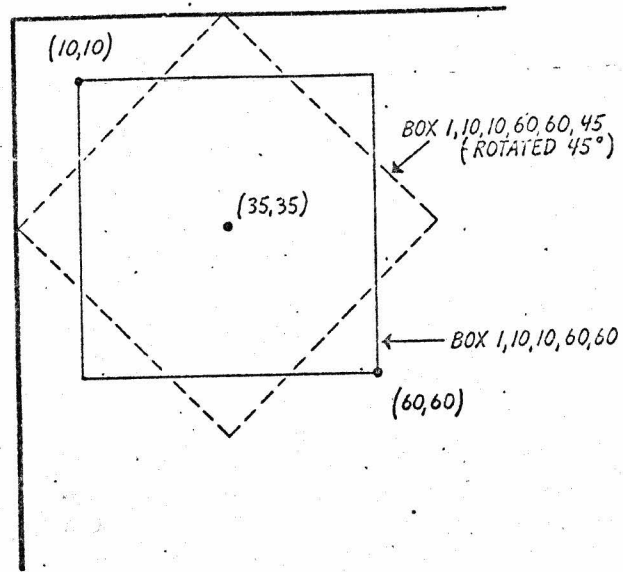
11 $\phi$  BOX,1 $\phi$ ,1 $\phi$ ,6 $\phi$ ,6 $\phi$ ,45,1

wyrysuje prostokąt na ekranie o środku w /35,35/, obrócony o 45° - jak pokazano na rys.

Ostatni parametr decyduje o wypełnieniu wnętrza rysowanego prostokąta. Jeżeli ma on wartość "1", narysowany prostokąt zostaje zamalowany, jeżeli nie jest podany, lub ma wartość " $\phi$ " pojawi się tylko kontur prostokąta. Zamalowywanie jest wykonywane aktualnym kolorem "pierwszego planu". Jeżeli chcesz mieć prostokąt o różnych kolorach konturu i wypełnienia, musisz najpierw wykreślić wypełniony prostokąt w jednym kolorze, następnie zmienić kolor na taki, w jakim chcesz mieć jego kontur, a później wykreślić powtórnie w tym samym miejscu identyczny prostokąt, lecz wypełniony /ostatni parametr -  $\phi$ /.

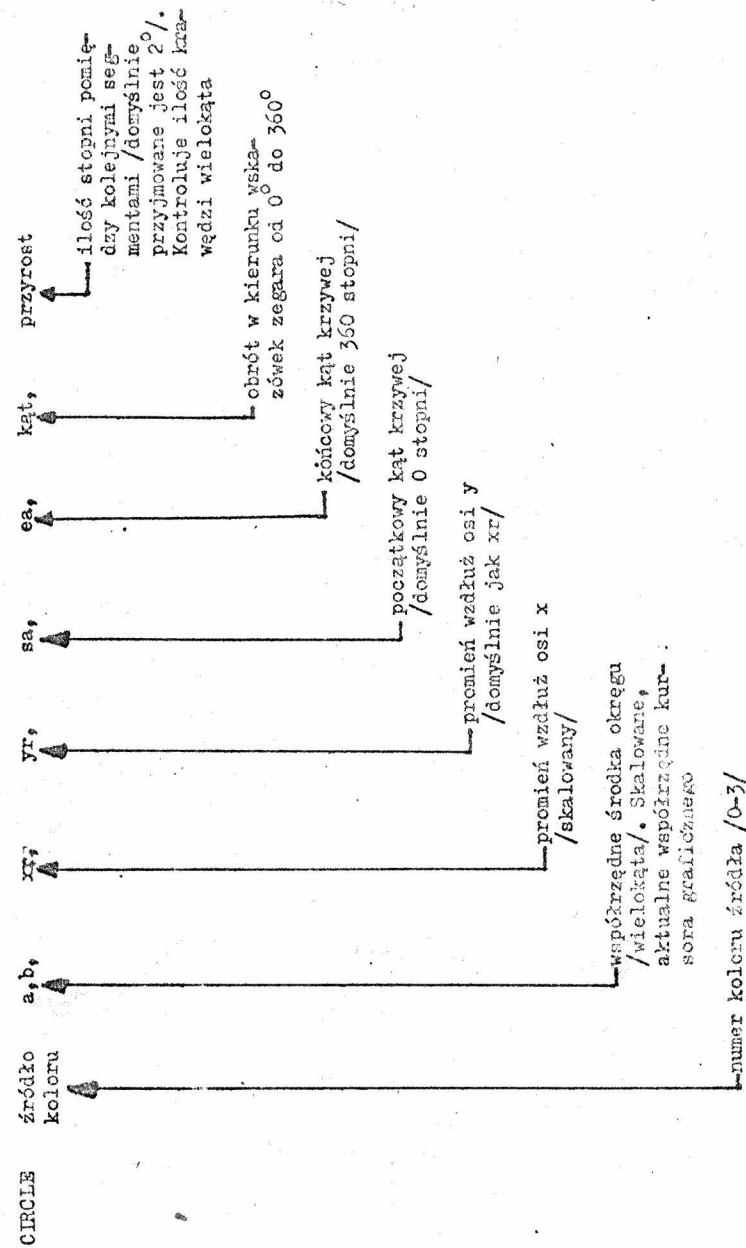
Okręgi, elipsy, trójkąty, wielokąty i łuki. Rozkaz: CIRCLE

Wspomnieliśmy już wcześniej, że można wykreślać na ekranie okręgi, nie mówiliśmy Ci jednak, że przy pomocy rozkazu CIRCLE można kreślić także różnego rodzaju regularne figury, począwszy od okręgów do elips, łuków i wielokątów takich, jak np. ośmiokąty. Składnia rozkazu CIRCLE jest pokazana na rys.



GRAPHICS ON THE C128 149

Rys. 24 - BOX - wykorzystanie parametru "kąt" = 45°



Rys. 25 -

Pierwszy parametr ustala numer źródła koloru, dwa następne określają środek figury. Jeżeli są pominięte jako środek figury zostaje przyjęta aktualna pozycja kursora graficznego.

Dwa kolejne - to promień X i promień Y. Chociaż podawanie dwóch różnych promieni może wydawać się dziwne, jest to jednak niezbędne przy rysowaniu elips. Jeśli pominiemy promień Y, za jego wartość zostanie przyjęty promień X. Jednakże nie otrzymamy wówczas doskonałego okręgu, gdyż osie X i Y są inaczej wyskalowane. W poniższym przykładzie rysowana jest elipsa o środku w punkcie 16φ, 1φφ

1φ CIRCLE, 16φ, 1φφ, 65, 1φ

Aby uzyskać okrąg trzeba odpowiednio dobrać promień Y tak, aby skompensować różnicę w skalowaniu osi. To polecenie spowoduje narysowanie "idealnego" okręgu:

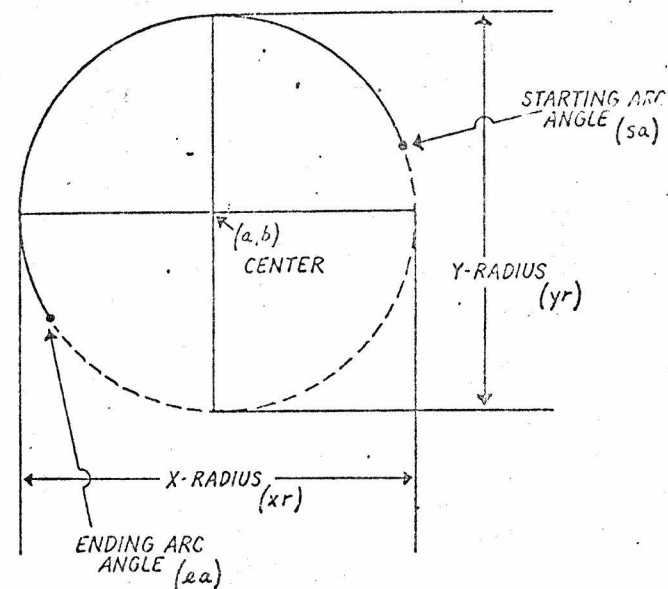
2φ CIRCLE, 16φ, 1φφ, 65, 5φ

Widzimy, że promień Y /5φ/, jest około 0,77 razy mniejszy od promienia X /65/. Parametry następujące po wartościach promieni stosowane są do wykreślenia fragmentu obwodu. Parametr "sa" określa kąt początkowy, a "ea" kąt końcowy tego fragmentu. Polecenie:

1φφ CIRCLE, 16φ, 1φφ, 65, 5φ, 9φ, 27φ

spowoduje narysowanie łuku rozpoczynającego się w miejscu wyznaczonym przez kąt 90° /godz.3/ i kończącego się na 270° /godz.9/, wyglądającego trochę jak uśmiech. Rysunek pola-  
zuje przykład znaczenia poszczególnych parametrów rozkazu CIRCLE.

Przedostatni parametr rozkazu CIRCLE, podobnie jak to ma miejsce w instrukcji BOX, określa kąt, o jaki będzie obrócona figura przed jej zobrazowaniem na ekranie. Kąt obrotu jest podawany w stopniach, a sam obrót odbywa się zgodnie z



Rys. 26 - przykład parametrów rozkazu CIRCLE

kierunkiem ruchu wskazówek zegara wokół środka figury. Parametr obrotu umożliwia nam np. obracanie w prosty sposób elipsy i łuków. Najbardziej interesujący jest jednak ostatni parametr /inc./. Decyduje on o tym, co ile stopni będzie rysowany kolejny "segment" figury. W praktyce okrąg jest rysowany również jako wielokąt, jednakże kąt pomiędzy segmentami wynosi tylko 2°. Segmenty łączą się na obwodzie okręgu, ponieważ jednak są one bardzo małe, powstały wielokąt wygląda jak okrąg. Jeżeli zadeklarujemy jawnie parametr /inc./ podając wartość kąta np. 120°, rozkaz CIRCLE zinterpretuje to jako kąt pomiędzy segmentami, a także wywnioskuje, że będą 3 takie segmenty

$$\frac{360^\circ}{120^\circ} = 3$$

tworzące trójkąt. Poniżej mamy kilka przykładów użycia parametru /inc./:

100 CIRCLE,60,40,20,18,,,45:REM ośmiokąt

110 CIRCLE,260,40,20,18,,,90:REM kwadrat

#### Wypełnianie i malowanie, rozkaz: PAINT

Zapewne zauważyłeś, że w instrukcji CIRCLE nie można dodać parametru decydującego o wypełnieniu wnętrza figury, jako to ma miejsce w rozkazie BOX. Rozkaz PAINT umożliwia Ci wypełnienie dowolnej zamkniętej figury kolorem. Kolor wypełniający figurę musi być taki sam jak kolor konturu lub dowolny kolor /nie będący tłem/ w trybie wielobarwnym, w którym jest więcej niż jeden kolor pierwszoplanowy. Składnia rozkazu PAINT wygląda następująco:

PAINT źródło koloru, a, b, tryb

Źródło koloru jest cyfrą od 0 do 3 i reprezentuje ekran, na którym chcesz malować. Parametry a i b, to współrzędne miejsca, od którego chcesz zacząć malować; jeżeli je puszczysz, zo-

staną przyjętymi współrzędne kursora graficznego /PC/. Parametr trybu - jeżeli = 0 - określa obszar zdefiniowany przez źródło koloru, a jeżeli 1 - określa obszar zdefiniowany przez jakiekolwiek źródło, nie będące tłem. Jeżeli pozycja o współrzędnych /a,b/ jest już zamalowana. Commodore 128 nie zamaluje obszaru wyznaczonego przez obrzeża Twojej figury. Musisz wybrać nowy punkt - wewnątrz figury, od którego zaczniesz się zamalowywanie.

Poniżej mamy przykład zastosowania hasła PAINT. Rysujemy najpierw okrąg, a następnie wypełniamy go kolorem, pierwszego planu

100 CIRCLE,160,100,65,50:REM okrąg środkiem w punkcie /160;100/

200 PAINT,160,100:REM wypełnienie okręgu kolorem pierwszego planu

#### Łączenie tekstu z grafiką: CHAR

Gdy tłumaczyliśmy różne tryby graficzne Commodore'a 128 zaznaczyliśmy, że dwa tryby tekstowe /40-kolumnowy i 80-kolumnowy/ mogą być łączone z trybami grafiki bitowej jedynie przy użyciu podzielonego ekranu. Jeżeli pracujesz w trybie bitowym i drukujesz jakiś znak, będzie on wysłany do ekranu trybu tekstowego i nie ujrzysz go dopóty, dopóki nie będziesz miał /a/ podzielonego ekranu. Na a co będzie, jeżeli zechcesz podpisać swoje obrazki, figury czy wykresy przy nich samych? BASIC 7.0 Commodore'a 128 zawiera rozkaz CHAR służący do rysowania znaków na ekranie bitowym. Składnia tego rozkazu wygląda następująco:

CHAR źródło koloru, x, y, łańcuch, negatyw

Źródłem koloru jest tryb ekranu, na którym chcesz umieścić swój znak, x jest kolumną znaku /0-39/, y jest wierszem znaku /0-24/. Zauważ, że CHAR nie zaczyna się na współrzędnych

pixel'a, lecz jego początek wyznaczony jest wierszami i kolumnami tekstu. Dlatego też, aby narysować - dla przykładu - jakiś tytuł we wnętrzu okręgu, najpierw będziesz musiał/a/ nanieść ów tytuł, a potem dopiero narysować okrąg i przesunąć go - pixel po pixel'u- aż osiągniesz to, czego chcesz.

Parametr "łańcuch" to tekst, jaki chcesz wyświetlić. Jeżeli będzie on za długi i nie zmieści się w linii, wówczas pozostała jego część zostanie wydrukowana od początku następnej linii. Parametr "negatyw" zmienia pole znaku na jego negatyw /jeżeli ma wartość 1/, jeżeli jest zerem pozostanie bez zmian. Ponieważ CHAR drukuje łańcuch znaków, kprzystając bezpośrednio z ROM-u Commodore'a 128, w trybie graficznym nie oferuje on takich możliwości jak PRINT w trybie tekstowym. W rozkazie CHAR nie można umieścić znaków kontrolnych lub graficznych, które działałyby na ekranie w trybie bitowym. Można jednak stosować CHAR również w trybie tekstowym i wówczas obie te grupy znaków mogą być wykorzystane w normalny sposób.

Poniżej mamy rozkazy, powodujące wykonanie następujących czynności: rysowanie okręgu i umieszczanie w jego wnętrzu tytułu: "Okrąg".

#### Nieco informacji i punktach i trybach: RDOT i RCLR

Jeżeli korzystasz z BASIC'a 7.0 Commodore'a 128, jedną z rzeczy, które będziesz często wykonywał jest przesuwanie obiektów, a następnie ustalanie, czy znalazły się one pod innymi obiektami lub też czy ich dotyczą. Czasem będziesz chciał/a/ wiedzieć, gdzie znajduje się PC bez ciągłego programowego śledzenia jego ruchów. Funkcją umożliwiającą ustalenie pozycji PC i źródła barwy dla danego punktu jest RDOT. Jej składnia wygląda następująco:

RDOT /N/

Znaczenie parametru N dla ustalonego położenia kursora graficznego /PC/ jest następujące:

dla N = 0 funkcja zwraca wartość współrzędnej x kursora,  
dla N = 1 współrzędną y, a  
dla N = 2 źródło koloru dla danego punktu /tzn. numer reprezentujący tryb, którego używamy/.

Funkcja RCLR stosowana jest do określenia koloru przypisanego źródłu koloru N, gdzie N przyjmuje wartości pomiędzy 0 a 6, jak to pokazano wcześniej w tabeli 6-3. Funkcja RCLR zwraca kod koloru źródła /od 1 do 16/ zgodnie z tabelą 6-2. Na przykład, jeżeli ustalasz kolor pierwszego planu w trybie graficznym na 6, to po wykonaniu programu:

10 X = RCLR /2/

20 PRINT X

X będzie miało wartość 6. RCLR i RDOT stosujemy, aby ustalić dokładny kolor punktu dla konkretnego źródła koloru. Funkcja RGR podaje aktualny tryb graficzny w postaci liczby zawartej w przedziale od 0 do 6.

#### Zmiana wymiarów obrazów: SCALE

Istnieje rozkaz w słowniczku Commodore'a 128, który może okazać się bardzo przydatny. Zezwala Ci on na zmianę normalnej skali na ekranie 320 x 200 punktów na inny format: 1024 x 1024! Zanim jednak oszalejesz na samą myśl o tym, że masz dostęp do każdego punktu na tak olbrzymim ekranie, uważaj!: na ekranie Twoim nadal jest tylko 320 x 200 pixel'i. Każdy z nich jednak tym razem nie reprezentuje już jednostkowej odległości. Teraz każda kropka w linii poziomej jest odpowiednikiem 1024/320 lub - 3.2 jednostek, a każda kropka w linii pionowej - 5.12 jednostek.



Nowa, większa macierz ekranu stworzona przez SCALE jest użyteczna przy adaptowaniu programów graficznych, które napisane są dla innych komputerów. Dziesiątki monitorów profesjonalnych wysokiej rozdzielczości ma ekran o wymiarach 1024 x 1024 punkty i istnieją setki programów profesjonalnych napisanych przy założeniu takiej właśnie rozdzielczości ekranu. Korzystając z polecenia SCALE łatwiej jest zaadaptować te programy na C128, bez konieczności przeliczania wszystkich współrzędnych.

Pamiętaj, że jeżeli zaprojektowałeś/aś program, który rysuje przedmioty tylko na płaszczyźnie określonej przez współrzędne 320 x 200, jeżeli zapomnisz skorygować współrzędne i przełączysz komputer na tryb 1024 x 1024, program Twój będzie wykonywał wszystkie operacje graficzne jedynie na małym obszarze w lewym górnym rogu ekranu 1024 x 1024.

Efekt działania rozkazu SCALE jest ponadto ujednolicenie współrzędnych figur i przedmiotów rysowanych we wszystkich trybach graficznych. Normalny tryb wielobarwny to 160 x 200, podczas gdy standardowy tryb graficzny to 320 x 200 punktów. Po wyskalowaniu ekranu /rozkaz SCALE 1/ obydwa te tryby obejmują zakres 1024 x 1024 punktów, a odległości między punktami o takich samych współrzędnych są w obu jednakowe.

#### Zapamiętywanie kształtów i rysowanie przy użyciu ciągów znaków

Przypuśćmy, że chcesz narysować szczegółowo jakiś obiekt na ekranie, a następnie pragniesz przenieść go z miejsca na miejsce. Przypuśćmy, że przedmiot ten rysowany był przy pomocy dziesiątek rozkazów graficznych, takich jak DRAW i BOX i wyrysowanie go zajęło wiele czasu. Rozkazy jakimi się posłużyłeś/aś możesz zawrzeć w podprogramie i wywoływać go za każdym razem podając nowe współrzędne. Byłoby to jednak bardzo czasochłonne. BASIC 7.0 ma możliwość zapamiętania kształtu

graficznego z ekranu w zmiennej tekstowej, a następnie przeniesienia go ponownie na ekran w wybrane przez Ciebie miejsce /i to prawie natychmiastowo!/.

#### Zapamiętanie obrazu: SSHAPE

Składnia rozkazu, który zapamiętuje prostokątny obszar ekranu bitowego w zmiennej tekstowej /łańcuchowej; string variable/ BASIC'a wygląda następująco:

SSHape nazwa zmiennej, x1, y1, x2, y2

gdzie: "nazwa zmiennej" jest nazwą zmiennej tekstowej BASIC'a, które przechowuje dane. Współrzędne x1, y1 wyznaczają lewy górny róg prostokąta otaczającego obiekt, który ma być zapamiętany, a x2, y2 to współrzędne prawego dolnego rogu tego prostokąta. W ten sposób, aby zachować obraz przedmiotu, który znajduje się w prostokącie 10,10 - 33,31 w zmiennej F\$, powinieneś wydać polecenie:

100 SSHape F\$,10,10/,33,31

Ponieważ zmienne tekstowe BASIC ograniczane są do 255 znaków, rozmiar obszaru, jaki możemy zapamiętać jest również ograniczony. Możesz skorzystać z podanych niżej wzorów do wyliczenia długości ciągu znaków potrzebnego do zapamiętania danego obszaru zakładając, że rogi tego obszaru są znane. Dla standardowego trybu bitowego:

$$\text{długość} = \text{INT} // \text{ABS} / x1 - x2 / + 1 / / 8 + .99 / * \text{ABS} / y1 - y2 / + 1 / + 4$$

Dla trybu wielobarwnego wzór wygląda następująco:

$$\text{długość} = \text{INT} // \text{ABS} / x1 - x2 / + 1 / / 4 + .99 / * \text{ABS} / y1 - y2 / + 1 / + 4$$

Kształt, który zapamiętujesz, jest przepisywany linia po linii do zmiennej. Ostatnie cztery bajty zmiennej będą zawierały długości wierszy i kolumn wykorzystane przez GSHAPE, jak zaraz to zobaczymy.

## Odtwarzanie obrazu: GSHAPE

Rozkaz GSHAPE działa dokładnie odwrotnie niż SSHAPE. Umieszcza on zawartość zmiennej tekstowej na ekranie w trybie bitowym. Składnia tego rozkazu jest następująca:

GSHAPE zmienna, a, b, tryb

W rozkazie tym parametr "zmienna" oznacza nazwę zmiennej tekstowej, zawierającej dane przeniesione z ekranu, a i b są współrzędnymi miejsca, w którym znajdzie się lewy górny róg przedmiotu, który ma być odtworzony. Jeżeli współrzędne a i b nie zostaną podane w ich miejsce zostaną przyjęte aktualne współrzędne kursora graficznego /PC/. I tak, zakładając, że nasza figura jest przechowywana w zmiennej o nazwie F\$, mamy tu kilka przykładów:

1\$ GSHAPE F\$: REM obiekt wykreślony na pozycji PC

11\$ GSHAPE F\$,16\$,1\$\$: REM obiekt zapamiętany w zmiennej F\$ wykreślony na pozycji 16\$, 1\$\$

### Tryby graficzne

GSHAPE może znacznie więcej niż tylko umieszczanie zapamiętanego obrazu graficznego na ekranie. Parametr "tryb" kontroluje sposób, w jaki punkty obiektu graficznego łączą się z punktami, będącymi już na ekranie. Parametr "tryb" przyjmuje 5 różnych wartości pokazanych w tabeli 6-4.

Tabela 6-4 - Tryby graficzne dla GSHAPE

| Parametr | Znaczenie                                                       |
|----------|-----------------------------------------------------------------|
| \$       | TAK JAK JEST    kopiuje bez zmian obiekt zapamiętany w zmiennej |
| 1        | NEGATYW        zamienia punkty tła i pierwszego planu           |

| Parametr | Znaczenie                                                             |
|----------|-----------------------------------------------------------------------|
| 2        | LUB            logiczne LUB z punktami tła                            |
| 3        | I              logiczne I z punktami tła                              |
| 4        | XOR            logiczne Exclusive-OR /wyłącznie - LUB/ z punktami tła |

Zbadajmy te tryby bardziej szczegółowo.

### Tryb wymieniania lub kopiowania

Tryb "TAK JAK JEST" całkiem przesłania kształt znajdujący się w tle: tak, że każdy bit z tła wymieniony zostaje na bit z pierwszego planu. Dlatego też, niezależnie od tego, co jest narysowane na ekranie w trybie bitowym, kiedy odtwarzamy obiekt przy użyciu GSHAPE w trybie kopiowania, to co znajdzie się pod spodem zostanie całkowicie zamazane.

### N e g a t y w

Tryb "NEGATYW" rysuje dany kształt dokładnie tak, jak tryb kopiowania "TAK JAK JEST", ale zamienia wszystkie kolory pierwszego planu na kolory tła i vice versa. To znaczy, że jeżeli masz czerwony znak na czarnym tle i skorzystasz z trybu "NEGATYW", znak zmieni się na czarny, a tło stanie się czerwone.

### Tryb "LUB"

Tryb ten "dodaje" punkty przenoszonego obrazu do punktów rysunku tła tak, że jeśli którykolwiek z tych punktów /lub obydwa/ ma kolor pierwszego planu, obraz na ekranie będzie również miał kolor pierwszego planu. Tylko wtedy, gdy obydwa punkty są w kolorze tła, obraz na ekranie będzie w kolorze tła.

O trybie tym mówi się, że w nim "pierwszy plan wygrywa" albowiem wystarczy, że kolor pierwszego planu wystąpi na jednej z płaszczyzn /obraz lub ekran bitowy/ aby kolor ten pojawił się na obrazie. "Logika" tego trybu przedstawiona jest w tabeli 6-5.

#### Tryb "AND"

Tryb AND logicznie dodaje odtwarzany kształt do obrazu na ekranie, skutecznie usuwając wszelkie punkty, które nie są wspólne dla obu rysunków. Reguła jest taka, że tylko wtedy, gdy zarówno obraz jak i punkty ekranu bitowego są w kolorze pierwszego planu, obraz końcowy na ekranie będzie także w tym kolorze. Jakakolwiek inna kombinacja daje w efekcie kolor tła. Efekt jest tak, że tylko te części dwóch kształtów, które rozkładają się, są widoczne. Tabela 6-6 przedstawia logikę tego układu.

#### Tryb "XOR"

Tryb XOR jest bardzo pożyteczny. Wykonuje on funkcję wyłączenia -LUB na punktach obiektu i ekranu - tak, że jedynie te punkty obrazu i ekranu w trybie bitowym, które mają różne kolory /jeden - pierwszego planu, drugi - tła/ dadzą punkt w kolorze pierwszego planu. Tabela 6-7 przedstawia logikę trybu XOR.

Tryb XOR jest często stosowany, gdy chcesz narysować jakiś przedmiot na danym tle, a nie chcesz zmieniać tła. Trik polega na tym, że rysujesz swój obiekt przy użyciu GSHAPE w trybie XOR, a następnie rysujesz ponownie, nie zmieniając współrzędnych. Za drugim razem oryginalne tło będzie całkowicie odtworzone. Następnie przenosisz się na nowe miejsce i powtarzasz ten proces. Pamiętaj, że inne tryby nie zezwalają na takie "niedestrukcyjne" rysowanie.

Poniższy program podaje przykład użycia SSHAPE i GSHAPE. Rysujemy okrąg na ekranie, zapamiętujemy go w zmiennej tekstowej za pomocą SSHAPE, oczyszczamy ekran, drukujemy linię tekstu, następnie przemieszczamy nasz obiekt wzdłuż ekranu z pomocą GSHAPE w trybie XOR. Tekst nie zostanie uszkodzony.

```

90 GRAPHIC 2,1
100 CIRCLE 1,10,8,10,8: REM okrąg o środku /10,8/
110 SSHAPE A0,0,0,20,16: REM zapamiętanie okręgu w
                           ciągu
115 SC:CLR
120 PRINT CHAR 1,0,0, "AAAAAAAAAAAAAAAAAAAAAAAAAA"
130 FOR X = 0 TO 295 STEP 2: REM pętla zmieniająca współ-
                           rzędną x
140 GSHAPE A0,X,0,4: REM odtworzenie okręgu w
                           trybie XOR
150 FOR D = 1 TO 50: NEXT D: REM opóźnienie
160 GSHAPE A0,X,0,4: REM skasowanie okręgu w try-
                           bie XOR
170 NEXT X: REM następna pozycja

```

Tabela 6-5 - Logika dla GSHAPE w trybie LUB

| Obiekt | Ekran | Efekt końcowy na ekranie |
|--------|-------|--------------------------|
| PP*    | PP    | PP                       |
| PP     | tło   | PP                       |
| tło    | PP    | PP                       |
| tło    | tło   | tło                      |

\* pp = pierwszy plan

Tabela 6-6 - Logika dla GSHAPE w trybie I

| Obiekt | Ekran | Efekt końcowy na ekranie |
|--------|-------|--------------------------|
| PP     | PP    | PP                       |
| PP     | tło   | tło                      |
| tło    | PP    | tło                      |
| tło    | tło   | tło                      |

Tabela 6-7 - Logika dla GSHAPE w trybie XOR

| Obiekt | Ekran | Efekt końcowy na ekranie |
|--------|-------|--------------------------|
| PP     | PP    | tło                      |
| PP     | tło   | PP                       |
| tło    | PP    | PP                       |
| tło    | tło   | tło                      |

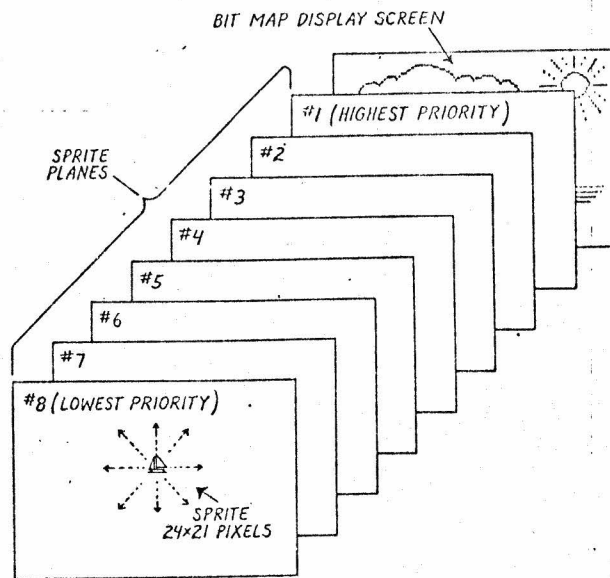
### Grafika sprite

Dowiedzieliśmy się już, że przy przesuwaniu obiektów graficznych na obrazie w trybie bitowym należy uważnie obserwować, w jaki sposób punkty obiektu łączą się z punktami tła. Stosując tryb XOR można uniknąć zmieniania czy wymazywania punktów tła. Pisząc programy, które przesuwają i rysują obiekty, wkrótce napotkasz nowy problem: zbyt mała prędkość. Gdy wielkość animowanego obiektu rośnie, coraz więcej czasu pochłania narysowanie go na ekranie. Maksymalna prędkość obiektu maleje, gdyż trzeba dłużej czekać aż zostanie on narysowany. Co więcej, każda pozycja obiektu musi być pieczołowicie obliczana przez program: to oznacza większą ilość pętli, zmiany współrzędnych pozycji obiektu, testowanie czy wskazują

one na odpowiedni punkt itd. Trudno wyobrazić sobie śledzenie kilku obiektów graficznych jednocześnie w danym programie. Ślad mógłbyś wiedzieć, na przykład, że jakiś obiekt zderzył się z innym obiektem lub z krawędzią pola gry. Niestety, wymagałoby to niezwykle skomplikowanego programu w BASIC'u. Byłby on tak powolny, że nie miałby zastosowania w animacji, symulacji, czy dzisiejszych grach zręcznościowych.

Zarówno Commodore 128 jak i C64 pozwalają na animację i poruszanie obiektami poprzez mechanizm programu zwany "sprite", który jest niestety trudno dostępny w BASIC'u C64. "Sprite" to programowo "inteligentne" obiekty graficzne, które mogą być programowane przy użyciu BASIC'u 7.0 /oraz BASIC'u 2.0 z C64 jeśli masz ochotę korzystać z POKE i PEEK/. Sprite zajmuje prostokątny obszar o szerokości 24 punktów /pixeli/ i wysokości 21 punktów. W Commodore 128 istnieje możliwość zdefiniowania ośmiu takich sprite'ów. Można je tworzyć za pomocą specjalnego edytora, wbudowanego w BASIC lub za pomocą rozkazów DATA i POKE. Dwie cechy decydują o tym, że sprite'y są niezwykle ważnym i użytecznym narzędziem przy programowaniu:

1. Każdy sprite istnieje i porusza się na swej własnej, odrębnej od innych płaszczyźnie ekranu. Poszczególne sprite'y mogą mieć różne kolory i kształty, Płaszczyzny sprite'ów są całkowicie niezależne od siebie i od informacji zawartej na ekranie w trybie bitowym. Podczas gdy sprite będzie przesuwany po ekranie, nie będzie on wymazywał ani w żaden inny sposób wpływał na punkty standardowego tła. Nie będzie też wpływał na punkty w pozostałych płaszczyznach. Sprawia to, że program nie musi trąszyć się o to, jaki sprite nałożył się na jaki sprite, ani na jakie tło. Gdyby tak nie było, program musiałby być bardziej skomplikowany.



Rys. 27 - Sprite'y istnieją na niezależnych od siebie płaszczyznach

2. Sprite'y mogą być automatycznie przesuwane przez układ VIC II do dowolnego punktu na ekranie - podaje się jedynie ostateczne współrzędne sprite'a i przesuwa się on gładko do wyznaczonej pozycji. Można również nadać sprite'owi "kurs" /dowolny kąt 0 - 360 stopni/ oraz prędkość /0 - 15/ i będzie się on przesuwał automatycznie, zgodnie z instrukcją, pojawiając się po przeciwnej stronie ekranu gdy przekroczy jego granicę /tak, jak gdyby odpowiadające sobie brzegi ekranu /tzn. Lewy - Prawy, Góra - Dół/ zostały sklejone ze sobą/. Będzie się tak działo dopóki sprite nie otrzyma rozkazu, aby się zatrzymać. Taki sposób kontrolowania obiektów graficznych stwarza doskonałe warunki do animacji, nie wymagając jednocześnie zły:

\* przyp.tłum.

wiele od programisty. Jest to również przyczyną istnienia tak dobrych programów dla C64.

Sprite'y mogą być również łączone ze sobą, tworząc duże, barwne obrazy. Połączenie czterech sprite'ów tworzy obiekt o szerokości 96 punktów i wysokości 84 punktów, zajmujący 30 - 40% całego ekranu.

Każdemu sprite'owi można przypisać odpowiedni "priorytet" wyświetlania, co sprawia, że można on się przesunąć za lub przed obrazem tła. Pozwala to na uzyskanie efektów trójwymiarowych w programach. Można na przykład narysować ciężarówkę poruszającą się za rzędem drzew i przed rzędem domów.

Program może także z łatwością ustalić czy sprite'y zderzyły się ze sobą bądź z tłem. Takie kolizje powodują przerwanie programu w BASIC'u wówczas można wywołać specjalną procedurę w celu przetworzenia informacji o kolizji. Oznacza to, że napisanie gry, w której będzie się poruszała większa ilość obiektów, bądź w której fragmentach część obiektów będzie wybuchać, nie stanowi większego problemu. Nie zachodzi potrzeba ciągłego sprawdzania współrzędnych obiektów graficznych, jak to ma miejsce w programach nie wykorzystujących sprite'ów.

Sprite'y mogą być powiększane w kierunku x lub y, dając efekt nagłego wzrostu. Użycie tej opcji powoduje, że poszczególne punkty sprite'ów powiększają się dwukrotnie. Nauczmy się teraz, w jaki sposób można tworzyć sprite'y i jak z nich korzystać.

#### Jak powstają sprite'y

Istnieją trzy sposoby tworzenia sprite'ów:

1. Rysowanie i definiowanie obiektów za pomocą wbudowanych rozkazów korzystając z tej metody. Należy narysować żądany obiekt przy użyciu takich rozkazów, jak DRAW, CIRCLE,



BOX i PAINT, a następnie użyć polecenia SSHAPE aby zapamiętać obraz jako ciąg znaków. Zapamiętany obraz jest następnie "zamieniany" na sprite za pomocą rozkazu SPRSAV. Polecenie SPRITE uruchomi obiekt i nada mu barwę, a poruszać nim będzie można poprzez rozkaz MOVSPR. Zapamiętujesz sprite'a na dysku za pomocą komendy BSAVE. /BSAVE pozwala Ci określić pewien obszar pamięci RAM i zapisać zawarte w nim wartości binarne jako plik binarny na dysku. BLOAD odczytuje pliki binarne i ładuje je z powrotem do pamięci RAM/.

2. Użycie trybu definiowania sprite'ów: SPRDEF /tzn. edytora lub projektanta sprite'ów/. Stosując tę metodę rysujesz /za pomocą wbudowanego edytora sprite'ów/ projektowanego sprite'a wprost na ekranie. Możesz kontrolować wygląd obrazu w powiększeniu przesuwając kursor oraz "zapalając" i "gasząc" różne części siatki. Kiedy projektowanie jest zakończone, sprite jest włączany i poruszany za pomocą rozkazów SPRSAV i MOVSPR.

3. Wykorzystanie metody stosowanej w Commodore 64.

Stosując tę metodę, musiałeś rozrysować swojego sprite'a na siatce o wymiarach 24 x 21, zmienić wzór na zbiór liczb, a następnie za pomocą instrukcji POKE wstawić te liczby do wydzielonego obszaru pamięci, przeznaczonego do przechowywania definicji sprite'ów. Nie będziemy omawiać tej metody.

Zarówno metoda 1 jak i 2 są przydatne. Najpierw opiszemy jak używać edytora sprite'ów, gdyż skorzystanie z niego umożliwia otrzymanie poruszającego się po ekranie sprite'a przy minimalnej długości niezbędnego programu.

### Tryb projektowania sprite'ów: SPRDEF

Program wspomagający projektowanie jest wbudowany w Commodore 128, co ułatwia projektowanie kształtu sprite'ów. Program ten jest skonstruowany na wzór edytorów sprite'ów dostępnych dla C64. Aby przejść do trybu projektowania sprite'ów z BASIC'a 7.0, wprowadź rozkaz SPRDEF. Ekran natychmiast się oczyści, a następnie pokaże się siatka sprite'a w lewym górnym rogu ekranu, jak to pokazano na rys. . U dołu ekranu zobaczysz pytanie o numer sprite'u.

Przy pracy w trybie definiowania sprite'ów, normalne wykonywanie BASIC'u 7.0 zostaje zawieszane. Klawiatura zostaje zdefiniowana tak, aby ułatwić kontrolowanie przebiegu definiowania sprite'u. Gdy wprowadzisz numer sprite'u /od 1 do 8/, siatka sprite'u /przestrzeń robocza na ekranie/ zostaje wypełniona powiększonym wizerunkiem sprite'u, podczas gdy on sam zostaje wyświetlony w prawym, górnym rogu ekranu. Przestrzeń robocza zajmuje obszar o szerokości 24 znaków i wysokości 21 znaków tak, że każdy znak odpowiada jednemu punktowi definiowanego sprite'u. Pracując w trybie projektowania, poruszasz się specjalnym kuresem + po przestrzeni roboczej za pomocą klawiszy sterowania kuresem. Wypełniając okienka siatki obiektu w obrębie obszaru roboczego zobaczysz, że zaświecać się będą odpowiednie punkty wyświetlanego sprite'u.

Sprite może być wyświetlany w trybie jednokolorowym lub wielobarwnym /multicolor bit mode/. Możesz nadać sprite'owi którąkolwiek z ośmiu barw pierwszego planu za pośrednictwem klawiatury /jedynie dla wygody - ostateczny kolor sprite'a zostaje określony rozkazem SPRITE/. Można podwoić wielkość sprite'u w kierunku poziomym /i/ lub pionowym, lub też zapamiętać jego wygląd, przytrzymując shift podczas przyciskania Return. Dalsze instrukcje dotyczące



używania edytora zostały zawarte w dokumentacji dołączonej do Commodore'a 128.

### Włączanie i definiowanie sprite'ów: SPRITE

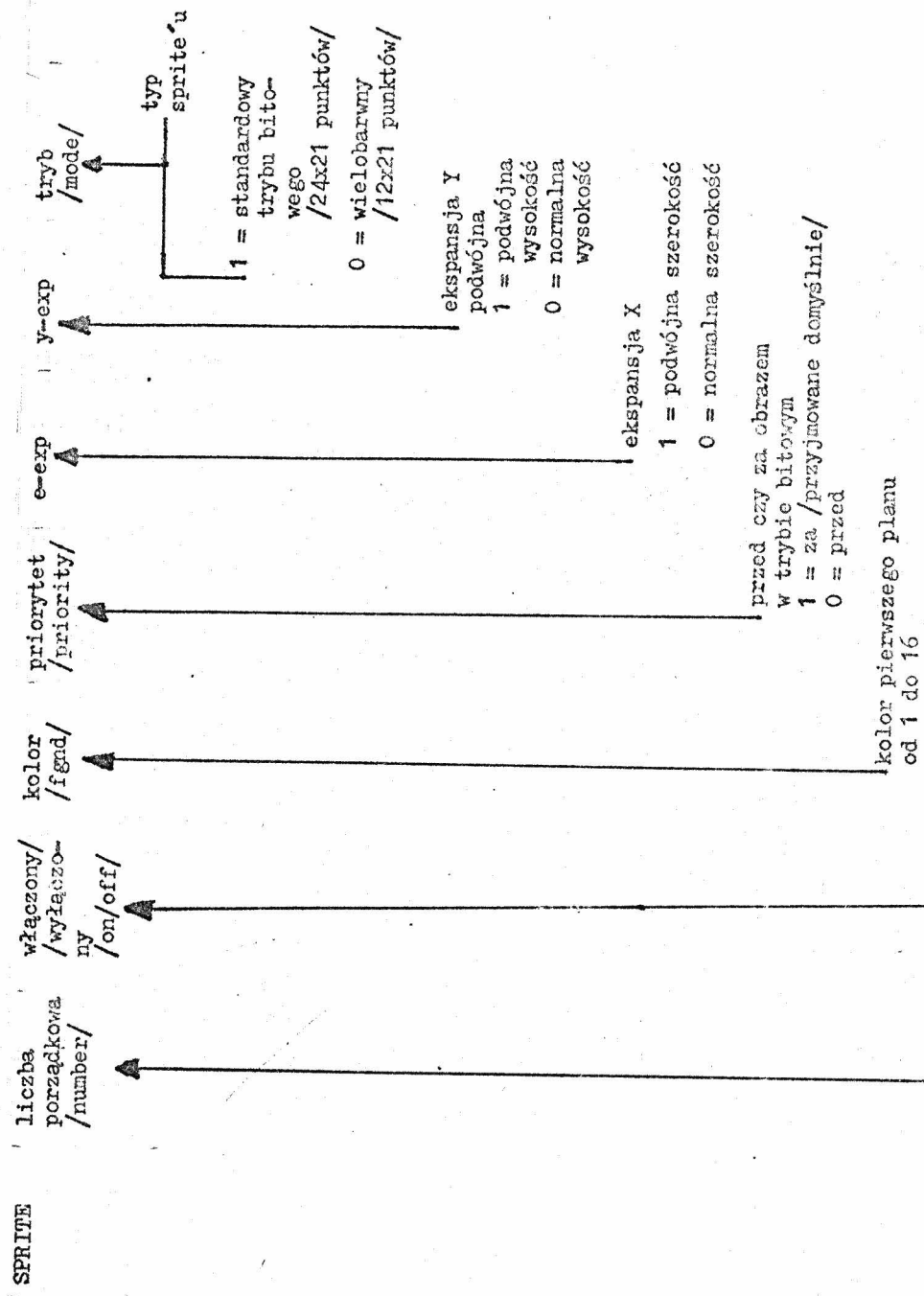
Gdy opuszczasz tryb projektowania sprite'ów, wszystkie zdefiniowane przez Ciebie sprite'y są już zmagazynowane w specjalnie do tego celu przeznaczonym obszarze pamięci. Jednakże na tym etapie sprite'y są bezbarwne i nie są "włączone", tzn. nie są widoczne na ekranie. Rozkaz SPRITE jest odpowiedzialny za ich włączanie i wyłączenie, oraz za nadawanie im koloru. Składnię rozkazu SPRITE ilustruje rys.

```
SPRITE  number  on/off  fgnd  prio-  x-exp  y-exp  mode
         /licz-  /wła-   /ko-   rity   x-exp  y-exp  /tryb/
         ba po-  czony/  lor/   /prio-
         rząd-  /wyłą-   /prio-
         k-     /czony/   rytet/
         wa/                      wa/
```

W tym rozkazie parametr "number" oznacza numer sprite'u /od 1 do 8/, którym chcemy manipulować. Parametr "włączony/wyłączony" /on/off/ określa czy sprite jest włączony czy wyłączony /1 - oznacza, że jest on włączony, 0 - że wyłączony/. Parametr kolor /fgnd/ określa barwę jaką chcesz nadać sprite'owi i może mieć wartość od 1 do 16.

### Priorytet sprite'ów

Parametr "priorytetu" /"priority"/ precyzuje czy dany sprite znajduje się za /1/ czy przed /0/ obiektami na obrazie w trybie bitowym. Priorytet sprite'u jest określany przez jego liczbę porządkową - sprite'y o niższej liczbie porządkowej pojawiają się "przed" sprite'ami o wyższej liczbie. Wynika stąd, że sprite o liczbie porządkowej 1 ma najwyższy priorytet, a sprite numer 8 - najmniejszy. Parametry ekspansji x i y kontrolują długość i szerokość sprite'a. Gdy którykolwiek z tych



parametrów będzie równy 1, sprite będzie dwukrotnie większy wzdłuż odpowiedniej osi /x lub y/.

Parametr trybu określa czy sprite będzie wielobarwny, czy nie. Wielobarwny sprite pojawi się, gdy parametr ten będzie równy 1. Wielobarwne sprite'y mają rozdzielczość 12 punktów na 21 punktów. Rozdzielczość pozioma maleje, ponieważ ilość pojedynczych punktów na ekranie w trybie wielobarwnym /multicolor/ w poziomie maleje do 160, /normalnie 320/. Wielobarwne i standardowe sprite'y mogą być łączone w dowolnym trybie graficznym, ale nie mogą być wyświetlane w trybie tekstowym. Standardowe sprite'y mogą mieć dwie barwy: barwę tła ustaloną za pomocą rozkazu COLOR oraz barwę pierwszego planu ustaloną rozkazem SPRITE. Wielobarwne sprite'y mogą mieć dwie barwy pierwszego planu ustalone za pomocą rozkazu SPRCOL.

Oto kilka przykładów ustalania cech: sprite'ów:

|                        |                                                                                                           |
|------------------------|-----------------------------------------------------------------------------------------------------------|
| 100 SPRITE 1,1,7:      | REM sprite 1, włączony, niebieski                                                                         |
| 110 SPRITE 2,1,5,1:    | REM sprite 2, włączony, purpurowy,<br>z tyłu                                                              |
| 120 SPRITE 8,0,8,,1,1: | REM sprite 8, wyłączony, żółty,<br>z przodu, podwójna szerokość, podwójna wysokość                        |
| 130 SPRITE 8,1,7,,, 1: | REM sprite 8, włączony, niebieski,<br>z przodu, normalna szerokość, normalna wysokość, tryb wielokolorowy |

Dane zawarte w linii 120 wyłączają sprite'a numer 8, zmieniają jego barwę na żółtą, nadają mu wyższy priorytet oraz podwajają wymiary x i y. Zwróć uwagę, że gdy któryś z parametrów sprite'u nie jest podany, za jego wartość zostaje zwykle domyślnie przyjęte 0.

### Animacja sprite'ów: MOVSPR

Teraz, gdy już nauczyliśmy się projektować sprite'y w trybie definiowania, włączać je i wyłączać, nadawać im barwy oraz ustalać ich priorytet, zobaczymy jak je uruchamiać. W celu ustalenia pozycji sprite'u, uruchomienia go i zatrzymania używamy rozkazu MOVSPR. Istnieją trzy typy składni dla tego rozkazu:

MOVSPR numer, x, y: REM bezwzględna pozycja

MOVSPR numer, +/-x, +/-y: REM względna pozycja

MOVSPR numer, x, y: REM kąt i prędkość

W tym przypadku parametr "numer" odpowiada liczbie porządkowej sprite'a /od 1 do 8/, którym chcesz manipulować, natomiast x i y to współrzędne pozycji, jaką chcesz nadać sprite'owi. Gdy wartości x i y zostaną zawarte w rozkazie, sprite automatycznie przesunie się na nową pozycję. Na przykład po rozkazie:

100 MOVSPR 1,285,178

sprite z numerem 1 przesunie się ze swojej dotychczasowej pozycji do prawego dolnego rogu ekranu.

Jeśli parametrom x i y zostaną nadane wartości względne /np. nada im się znaki + lub -, jak pokazano w następnym przykładzie/, sprite przesunie się względem pozycji, jaką ostatnio zajmował /NIE względem kursora graficznego PC/. Na przykład rozkaz:

110 MOVSPR 1, +50, -50

przesunie sprite'a o liczbie porządkowej 1 o 50 jednostek /pixeli/ w prawo i o 50 jednostek do góry. Nie równa się to rozkazowi

110 MOVSPR 1, 50, 50

który przesunie sprite'a do pozycji /50,50/.

Trzeci format rozkazu MOVSPR umożliwia automatyczne uruchamianie i zatrzymywanie sprite'ów. Jest to tak zwany tryb "zółwia", gdzie należy natychmiast zastosować trzeci format instrukcji dla MOVSPR, w którym x i y zostały rozdzielone znakiem . Wartość x określa tu kąt od 0-360° /zgodnie z kierunkiem wskazówek zegara/, pod jakim sprite będzie się poruszał. Wartość y określa stałą prędkość sprite'u, od 0 do 15. Na przykład po wydaniu rozkazu:

100 MOVSPR 2,135, 8 REM sprite 2,135 stopni, prędkość 8  
sprite o liczbie porządkowej 2 zacznie się przesuwać w kierunku prawego dolnego rogu ekranu pod kątem 135° ze względną prędkością 8. Ilustruje to rys.

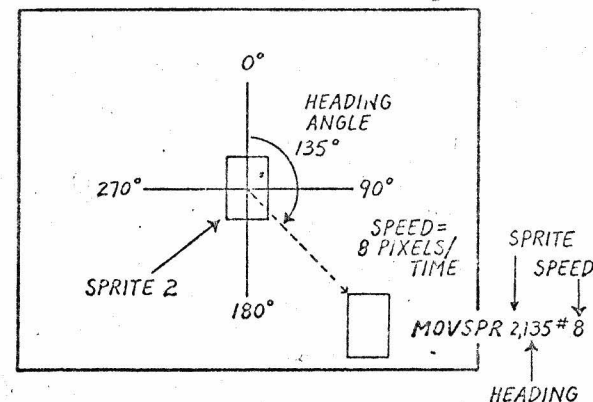
Sprite'y mają tę właściwość, że mogą być uruchamiane jednocześnie lub niezależnie od siebie i będą się wciąż poruszały, jak gdyby owijając się wokół ekranu i przechodząc nad sobą bez zwalniania programu. Gdy sprite zostaje wyłączony za pomocą rozkazu SPRITE, zapamiętuje on swoją ostatnią pozycję. Po ponownym włączeniu, wyrusza on z tej pozycji zgodnie z poprzednią prędkością i kursem, chyba że zostaną one zmienione.

#### Obsługa zderzeń sprite'ów: COLLISION i BUMP

BASIC stosowany w Commodore ma tę korzystną właściwość, że pozwala przewidzieć kiedy sprite'y zderzą się ze sobą bądź z tłem. Instrukcja COLLISION "wyczula" program na zderzenia sprite'ów, a BUMP - informuje, które sprite'y się zderzyły.

#### Wykrywanie zderzeń: COLLISION

Polecenie COLLISION wychwytuje trzy typy "zdarzeń": zderzenia między sprite'ami, zderzenia między sprite'ami a obiektami tła oraz uruchomienie pióra świetlnego. W każdym z tych przypadków, BASIC najpierw dokończy aktualnie wykonywany



Rys. - Automatyczne wyzwalanie ruchu za pomocą MOVSPR

rozkaz, a następnie wykona pierwszą linię podprogramu obsługi zderzeń, której numer jest parametrem instrukcji COLLISION. Na końcu tej procedury należy umieścić rozkaz RETURN. Spowoduje to wznowienie wykonywania programu od linii następującej po tej, która była wykonywana gdy całe zdarzenie miało miejsce.

Zderzenia powodują przerwanie wykonywania programu i odsyłają go do procedury, która sprawuje kontrolę nad zderzeniami. To zjawisko nazywane jest "wychwytywaniem" zdarzeń. Podprogram obsługi zdarzeń może zmienić kierunek sprite'u, gdy

ten uderzy w ścianę na krawędzi ekranu, lub spowodować, że sprite "wybuchnie", zniknie po zderzeniu z innym sprite'em. Rozkaz COLLISION skutecznie "uzbraja" program, by mógł on reagować na takie zderzenia.

Składnia instrukcji COLLISION wygląda następująco:

```
COLLISION      event      line number
               /wydarzenie/ /numer linii/
```

Parametr "wydarzenia" określa w tym przypadku typ zdarzenia jakie ma zostać wychwycone, zgodnie z tabelą 6-8.

Tabela 6-8 - Znaczenie numerów wydarzeń w rozkazie MOVSPR

| Wartość | Wydarzenie do wychwycenia                               |
|---------|---------------------------------------------------------|
| 0       | zderzenie dwóch sprite'ów                               |
| 1       | zderzenie sprite'a z wyświetlonymi na ekranie obiektami |
| 2       | aktywacja pióra świetlnego                              |

Numer linii wskazuje programowi, do którego miejsca ma nastąpić skok, gdy pojawi się przerwanie. Jeśli numer linii nie jest podany w rozkazie, wychwytywanie zdarzeń zostaje uniemożliwione. Zderzenia mają miejsce jedynie w przypadku, gdy część sprite'u pokryje się z częścią innego sprite'u. Mimo, iż punkty poruszającego się sprite'a są niezależne od punktów tła, układ VIC II jest w stanie określić kiedy sprite i obiekty znajdujące się w tle nachodzą na siebie. Tak więc zderzenie między sprite'em a tłem zachodzi wtedy, gdy element sprite'a o barwie pierwszego planu zajmuje tę samą pozycję co punkt o tej samej barwie, wchodzący w skład tła. Wyłączone sprite'y nie powodują zderzeń. Pomimo, iż możesz jednocześnie mieć włączone wykrywanie kilku typów zdarzeń,

tylko jedno zdarzenie naraz może być obsługane. Z tego też powodu pierwszą czynnością, którą powinien wykonać napisany przez Ciebie podprogram obsługi zdarzenia jest wyłączenie możliwości obsługi innych zdarzeń do czasu ukończenia aktualnego podprogramu. Po zakończeniu obsługi zdarzenia i powrocie do głównego programu, każde wydarzenie, które miało miejsce podczas poprzedniego procesu, będzie zapamiętane i będzie powodowało wywołanie kolejnej procedury obsługi zdarzeń. Jeśli procedura ta będzie zwięzła, to nie zostaną przeoczone żadne zdarzenia sprite'ów.

Oto przykład typowego programu wychwytyującego zdarzenia:

```
1000 COLOR rozkaz ustalający kolory
1100 GRAPHIC rozkaz ustalający tryb
/Rozkazy w BASIC'u tworzące właściwy program, rysujące tło itp./
5000 COLLISION 1,1000 : REM uruchomienie wykrywania zderzeń
      sprite'ów
5100 COLLISION 2,2000 : REM uruchomienie wykrywania zderzeń
      sprite'ów z tłem
5200 /Tutaj pętla, w której program czeka na wystąpienie
      zderzenia/
10000 COLLISION 1 : COLLISION 2 : REM wyłączenie wykrywania
      wszelkich zderzeń
10005 /właściwy podprogram obsługi zderzenia sprite-sprite,
      powtórne włączenie wykrywania zderzeń/
15000 RETURN
20000 COLLISION 0 : COLLISION 1 : REM wyłączenie wykrywania
      wszelkich zderzeń
20005 /właściwy podprogram obsługi zderzenia sprite-tło,
      powtórne włączenie wykrywania zderzeń/
25000 RETURN
```

Widzisz więc, że gdy program już ustalił wszystkie parametry początkowe, "przysposabia" się do wychwytywania zderzeń w liniach 500 i 510. Następnie wchodzi on w pętlę, w której powinien poruszać pozostałymi obiektami, grać muzykę, odczytywać położenie joystick'a itd. Jest to najbardziej wolna część programu. Gdy tylko jakiś sprite zderzy się z innym sprite'em lub z tłem, nastąpi skok do podprogramu, zaczynającego się odpowiednio od linii 1000 lub 2000, który obsłuży zderzenie i wróci do pozycji wyjściowej. Na przykład sprite, po zderzeniu z krawędzią otaczającą ekran, może zostać "odbity" po matematycznym obliczeniu kąta dla rozkazu MOVSPR. Zwróć uwagę, że uruchomienie pióra świetlnego może również spowodować wygenerowanie przerwania w programie.

#### Które sprite'y się zderzyły: BUMP

Sama wiadomość, że jakiś sprite zderzył się z innym czy też z tłem może się okazać niewystarczająca. A jeśli więcej sprite'ów zderzyło się ze sobą? Zachodzi więc potrzeba określenia, które sprite'y się zderzyły lub który ze sprite'ów zderzył się z obiektem tła. W tym celu używamy funkcji BUMP.

W przypadku, gdy ma miejsce przerwanie programu wskutek np. zderzenia i program skacze do procedury obsługi, uruchamiamy funkcję BUMP, w celu określenia jakie sprite'y zderzyły się ze sobą bądź z tłem. Składnia BUMP wygląda następująco:

BUMP /typ wydarzenia/

Ponieważ BUMP jest funkcją, jego wynik musi być podstawiony pod zmienną, która będzie mogła być wykorzystana w programie, np.  $R = BUMP /1/$ . Parametr określający typ zderzenia stosowany jest w celu stwierdzenia, który rodzaj zderzenia jest przedmiotem Twojego zainteresowania. Odpowiada on dokładnie typom wydarzeń w rozkazie COLLISION, jak to zostało przedsta-

wione w poprzedniej tabeli. BUMP podaje liczbę z zakresu 0 do 255. Liczbę tę interpretuje się jako ośmiobitową wartość liczbową, gdzie każda pozycja bitu reprezentuje sprite'a, który zderzył się z innym sprite'em lub tłem. Jeśli bit jest włączony /1/ to oznacza, że dany sprite uległ kolizji, gdy wyłączony /0/ - to znaczy, że nie. Przy kilku jednoczesnych zderzeniach między sprite'ami, kilka pozycji bitowych będzie wykazywać aktywność. W tym przypadku należy użyć funkcji RSPOS /omawianej niżej/. Funkcja ta zwraca pozycje x i y oraz prędkość /od 1 do 15/, jeśli sprite nadal się porusza. Oto przykład użycia funkcji BUMP:

1000 A = BUMP /1/ : B = BUMP /1/

Linia ta sprawdza, które ze sprite'ów uległy kolizji między sobą i podstawia wynik pod zmienną A oraz sprawdza, które sprite'y zderzyły się z tłem i podstawia wynik pod zmienną B.

Przedstawiony niżej przykład pokazuje, jak zastosować operator logiczny AND w celu stwierdzenia kiedy sprite numer 2 zderzył się z krawędzią ekranu.

1000 GOSUB 3000 : REM rysowanie wokół ekranu ramki, z którą mają się zderzać sprite'y

1100 COLLISION 1,1000 : REM włączenie wykrywania zderzeń sprite - tło

1200 MOVSPR 1,160,190,45 8 : REM włączenie sprite'a nr 1, ruch po przekątnej

1300 MOVSPR 2,160,190,135 8 : REM włączenie sprite'a nr 2, ruch po przekątnej

2000 GOTO 2000

10000 REM PROCEDURA OBSŁUGI ZDERZEŃ SPRITE'ÓW - USTALENIE KTÓRY SPRITE SIĘ ZDERZYŁ

1010 COLLISION 1 : REM WYŁĄCZENIE PRZERWAŃ OD ZDERZEŃ

1020 IF BUMP /1/ AND 2 THEN GOSUB 2000 : REM CZY SPRITE NR 2?

1030 COLLISION 1,1000 : RETURN : REM PONOWNE WŁĄCZENIE WYKRY-  
WANIA ZDERZEN

2000 REM ZDERZENIE SPRITE'A NR 2

2510 /Działania podejmowane wskutek zderzenia się sprite'a nr 2  
z krawędzią ekranu/

2600 COLLISION 1,1000 : RETURN : REM PONOWNE WŁĄCZENIE WYKRY-  
WANIA ZDERZEN

#### Gdzie znajdują się sprite'y: funkcja RSPPOS

Funkcji RSPPOS używa się w celu określenia pozycji i prę-  
dkości sprite'a. Bez tej funkcji trzeba by było śledzić położe-  
nie sprite'ów i ich prędkość, wprowadzając zmienne do progra-  
mu, co jest teoretycznie możliwe, ale jest to proces dość zło-  
żony. Składnia rozkazu RSPPOS wygląda następująco:

RSPPOS /sprite, dane/

Parametr "sprite" określa numer porządkowy sprite'a, o  
którym żądany informacji, podczas gdy parametr danych okreś-  
la rodzaj informacji jakie zwraca RSPPOS, zgodnie z tabelą  
6-9.

Tabela 6-9 - Znaczenie parametru danych w rozkazie RSPPOS

| Dane | Co zostaje zwrócone             |
|------|---------------------------------|
| 0    | aktualna współrzędna x sprite'a |
| 1    | " " y "                         |
| 2    | prędkość sprite'a               |

RSPPOS określa, w którym z obrazów tła mógł uderzyć  
sprite. Na przykład w grze Breakout sprite byłby kulą, a  
cegły - obrazami. Przy każdym zderzeniu sprite'a z obrazem  
ściany RSPPOS zwraca aktualną pozycję x i y sprite'a, okreś-

lając tym samym, która z cegieł została uderzona. Dana  
cegła mogła by być wtedy wymazywana z ekranu.

#### Dodatkowe uwagi o sprite'ach

Możesz sprawić, by sprite'y stały się niewidzialne,  
nadając im barwy pierwszego planu identyczne z barwami  
podłoża. Sprite będzie się nadal poruszał, jeśli umieś-  
cisz go w automatycznym trybie kursu i prędkości. Techni-  
ka ta jest przydatna w wielu grach. Jednak jeśli wychwy-  
tywanie kolizji nie zostanie zablokowane, sprite zderza-  
jąc się z pozostałymi sprite'ami lub obrazami ekranu bitowego,  
będzie powodował przerwanie w programie. Można też sprawić, że  
program prowadzący sprite'y będzie również obsługiwał niewi-  
dzialne sprite'y.

Mimo, iż istnieje tylko osiem sprite'ów, w dowolnym  
miejscu programu można je powtórnie definiować i wozytywać  
te definicje w trakcie działania programu. W ciągach znaków  
można umieścić obrazy, które mogą być przekazywane sprite'om  
za pośrednictwem SPRSAV. Gra z udziałem ośmiu samochodów  
wścigowych może być z łatwością zamieniona na grę z udziałem  
ośmiu statków kosmicznych.

Wartość prędkości podawana w rozkazie MOVSPR /p-15/  
określa ilość punktów /pixel/ o jaką będzie się przesuwał  
sprite w jednakowych odstępach czasu. Sprite poruszający  
się z dostatecznie dużą prędkością, lub zawierający jedy-  
nie niewielki fragment pierwszego planu może unikać kolizji  
z cienką linią, małym obiektem lub innym małym sprite'em  
przez "przeskakiwanie" /jedynie elementy sprite'ów o barwie  
pierwszego planu reagują ze sobą/.



## O k n a

Gdy już zrozumieliśmy grafikę pola bitowego oraz zasadę działania sprite'ów, popatrzmy na jeszcze jeden nowoczesny atrybut Commodore'a 128 - okna.

Okna są pewnymi obszarami tekstów na ekranie, którymi można oddzielnie manipulować za pośrednictwem BASIC'a lub klawisza Esc. Można zmieniać wielkość okna /tzn. jego wysokość i szerokość/ oraz jego położenie. Gdy już ustawisz okno, wszystko co drukujesz /łącznie z użyciem INPUT i GET/ oraz listing programu w BASIC'u pojawi się w obszarze wyznaczonym przez to okno i nie ma żadnego wpływu na pozostały tekst na ekranie. Rozkazy oczyszczające ekran, powodujące przesuwanie kursora itd. nadal działają, ale jedynie w obszarze okna. Okna mogą być przesuwane do dowolnych wierszy i kolumn. W takim przypadku, lewy górny róg okna staje się nową pozycją macierzystą /0,0/.

### Do czego przydatne są okna?

Za pomocą okien można tak organizować pracę programu, by różne obszary okna były odpowiedzialne za poszczególne funkcje. Na przykład w programach edukacyjnych, opisowe części lekcji mogą być ograniczone do obszaru jednego okna, pytania - do drugiego, a odpowiedzi użytkownika - do trzeciego. Wejścia i wyjścia z poszczególnych okien są niezależne od siebie oraz od tego, co znajduje się poza oknami.

Przesuwanie okna nie powoduje zmian zawartych w nim informacji pod warunkiem, że nie zostanie ono nałożone na inne okno. BASIC 7.0 nie potrafi automatycznie odświeżyć zawartości okna. W tym celu musisz napisać własny program.

Składnia rozkazu okna wygląda następująco:

WINDOW lewy górny róg x, lewy górny róg y, prawy dolny x, y,  
czyszczenie

Minimalne i maksymalne wartości wierszy i kolumn zależą od tego, czy znajdujesz się w trybie 40-kolumnowym, czy 80-kolumnowym. Jeśli opcja czyszczenia okna jest wybrana /ma wartość 1/, okno zostaje oczyszczone zaraz po jego ustawieniu. Na przykład w celu ustawienia okna o szerokości 15 kolumn i wysokości 10 wierszy, zaczynającego się od wiersza 10 i kolumny 40, które ma być oczyszczone, użyjemy rozkazu:

100 WINDOW 10,10,15,20,1

Jedną z wielu zalet używania okien jest to, że ich zawartością można manipulować za pomocą klawisza ESC. Korzystając z tego klawisza można osiągnąć wiele ciekawych efektów przy redagowaniu programów. Przy wyświetlaniu jakiegokolwiek informacji na ekranie w obrębie okna, użytkownik programu może wymazywać linie, wprowadzać informacje i przesuwać zawartość okna, zmieniać jego wymiary, położenie itd. Jest to raczej proces redagowania obrazu, a nie wyrazów. Przy przesuwaniu, tekst który wychodzi poza okno znika. Jest możliwe kontrolowanie ekranu z programu, poprzez wprowadzenie w rozkazie PRINT sekwencji znaków sterujących wywoływanych przy użyciu klawisza ESC. Niektóre funkcje klawisza ESC są łatwe do zapamiętania, gdyż pierwsza litera rozkazu jest pierwszą literą funkcji. Na przykład w celu wprowadzenia linii do okna należy wydrukować;ESC i /insert - umieść\*/ , a dla jej usunięcia - ESC d /delete - usuń\*/.

### Status danego okna oraz obrazu RWINDOW

Żałujemy, że użytkownik pragnie zmienić wymiary okna, zawierającego wyjściowe informacje programu za pomocą klawisza ESC oraz umieścić w nim nowe informacje. Ponieważ zmieniają się wymiary, program musi znać nowe parametry,

\* przyp.tłum.

by odpowiednio przystosować format wyprowadzanych informacji /nie chcemy otrzymać tekstu o szerokości dziesięciu kolumn, skoro okno zostało zaprojektowane na dwadzieścia kolumn/.

Za pomocą funkcji RWINDOW możemy dowiedzieć się, jakiej wielkości jest okno. Składnia tej funkcji wygląda następująco:

RWINDOW /n/

gdzie parametr /n/ odpowiada poszukiwanej wartości. Jeśli  $n = \emptyset$  - podana zostanie ilość linii. Gdy  $n = 1$  uzyskamy informację o ilości rzędów, a gdy  $n = 2$  RWINDOW poda liczbę 40 lub 80, w zależności od tego, który z trybów pracy jest w użyciu.

Przypuśćmy, że okno miało szerokość 20 kolumn, a wydrukowaliśmy tekst o szerokości 40 kolumn. Można program skonstruować tak, aby najpierw zawsze sprawdzał szerokość okna, korzystając z funkcji RWINDOW /1/, a następnie kontrolował ilość wyrazów wpisywanych w jednej linii okna, tak aby nie były one dzielone przy przenoszeniu do następnej linii. Aby tego dokonać, standardowe programy musiałyby sprawdzać długość każdej linii oraz przystosowywać tekst do prawidłowego przenoszenia.

Przeżyłeś długą drogę w tym rozdziale! Teraz, gdy już zapoznałeś się z podstawowymi cechami grafiki na Commodore 128, popatrzmy jak można tworzyć efekty dźwiękowe i muzykę.

## 7. DŹWIĘK I MUZYKA

XXXXXXXXXXXXXXXXXXXX

W tym rozdziale dowiesz się:

- jakie dźwięki może wytwarzać Commodore 128
- jakie urządzenia i oprogramowanie muzyczne są dostępne na rynku
- jakie są zasady generowania dźwięku przez Commodore 128
- jak posługiwać się nowymi rozkazami produkującymi dźwięk w BASIC'u 7.0

Układ scalony generujący dźwięk wbudowany w Commodore'a 128 jest prawdopodobnie najbardziej wyrafinowanym urządzeniem tego typu. Zaprojektowany z myślą o generowaniu wysokiej jakości dźwięku w grach telewizyjnych, układ ten posiada zdumiewające możliwości. Jest w stanie naśladować dźwięki niemalże wszystkich instrumentów muzycznych oraz tworzyć niemal nieograniczone ilości efektów dźwiękowych, łącznie z naśladowaniem ludzkiego głosu.

W tym rozdziale ogólnie omówimy możliwości dźwiękowe Commodore'a 128 oraz opiszemy istniejące oprogramowanie, służące wytwarzaniu dźwięku. Następnie szczegółowo opowiemy, jak powstaje dźwięk w Commodore 128, by wreszcie pokazać, jak uzyskać poszczególne efekty muzyczne oraz jak komponować muzykę przy pomocy BASIC'u 7.0.

### Co można robić z dźwiękiem na Commodore 128?

Najprostszym użyciem możliwości dźwiękowych Commodore'a 128 jest zakupienie profesjonalnego oprogramowania. Stworzone głównie z myślą o muzyce, oprogramowanie to

pozwoli Ci zmienić C128 w wyrafinowany syntezator muzyczny, rywalizujący ze sprzętem profesjonalnym, którego ceny sięgają tysięcy dolarów. Wielu muzyków, kompozytorów i gró rockowych używa już Commodore'a 64 w swej pracy zawodowej, a C128 posiada wszystkie właściwości muzyczne C64.

Być może zechcesz nauczyć się pisać swoje własne programy produkujące dźwięki. Commodore 128 używany w trybie C128, odda nieocenione usługi w tym zakresie. Za pomocą BASIC'a 7.0 kontrolowanie dźwięku staje się procesem o wiele łatwiejszym, niż to było w przypadku wcześniejszej generacji komputerów firmy Commodore. Natychmiastowe sprzężenie zwrotne, jakie zapewnia BASIC sprawia, że możesz eksperymentować z różnymi rozkazami, dopóki nie uzyskaszżądanego efektu.

Oczywiście możesz również pisać programy generujące dźwięk w trybie C64, stosując BASIC 2.0. Commodore 64 /jak też tryb C64 w Commodore 128/ nie zawiera rozkazów w BASIC'u, które tak bardzo ułatwiają programowanie dźwięku w trybie C128. W celu wprowadzenia pewnych liczb do określonych obszarów pamięci /połączonych z układem scalonym SID/, trzeba więc poprzestać na stosowaniu rozkazu POKE w BASIC'u. Oznacza to, że musisz wiedzieć, jakie liczby masz wprowadzić rozkazem POKE oraz w którym miejscu pamięci należy umieścić. Dlatego właśnie programowanie efektów dźwiękowych w trybie C64 jest znacznie trudniejsze niż w trybie C128.

### Układ scalony SID

Commodore 128 posiada specjalny układ scalony - SID /Sound Interface Device - układ interfejsu dźwiękowego/, który panuje nad wszystkimi procesami generowania dźwięku. Układ ten może jednocześnie obsługiwać trzy odrębne kanały dźwięku /zwane głosami/. A więc Commodore 128 może z łatwoś-

\* przyp.tłum.

cią grać muzykę o trzyczęściowej harmonice. Indywidualne brzmienie każdego z tych trzech głosów może być odrębnie modyfikowane tak, by np. jeden z głosów brzmiał jak trąbka, drugi - jak gitara, a trzeci jak pianino. Taki efekt osiąga się modyfikując kształt fali oraz obwiednię dźwięku. Zostanie to szerzej omówione w dalszej części tego rozdziału. Zmieniając obwiednię i kształt fali któregośkolwiek z 3 głosów można również produkować bogatą gamę efektów dźwiękowych - od karabinów maszynowych i spadających bomb, do szelestu liści w jesienne popołudnie.

Gdy już posiadasz umiejętność posługiwania się BASIC'iem do tworzenia dźwięków, możesz do każdego programu dodawać efekty dźwiękowe lub muzykę. Efekty dźwiękowe najczęściej kojarzą się z grami, a Commodore stwarza idealne warunki do tworzenia gier z interesującą grafiką i efektami dźwiękowymi. Ale dźwięk może również być wykorzystywany w innych programach, na przykład programach kalkulacyjnych czy przetwarzania tekstów. Ciche trzaski mogą potwierdzać, że klawisz został przyciśnięty, a brzęczyki mogą sygnalizować wprowadzanie nieprawidłowych danych. Dźwięk generowany za pomocą Commodore'a 128 mógłby mieć zastosowanie nawet w elektronice. Na przykład C128 może generować rozmaite kształty fali, z białym szumem łącznie, w celu testowania sprzętu radiowo-telewizyjnego.

#### Jakiego oprzyrządowania potrzebujesz, jeśli chcesz używać dźwięku?

Mimo, że Commodore 128 posiada wbudowany układ scalony generatora dźwięku SID, sygnał wychodzący z tego układu musi być wzmocniony przez dodatkowe urządzenia zewnętrzne, aby dźwięki były słyszalne. C128 niestety nie posiada ani wbudowanych wzmacniaczy, ani głośnika.

Bardzo często odbiornik telewizyjny lub monitor podłączony do komputera posiada własny wbudowany wzmacniacz i głośnik.

Na przykład nowy, kolorowy monitor 1902 do Commodore'a 128 posiada oba te urządzenia, a ponadto regulator siły głosu. Po podłączeniu wyjścia audio z C128 /z gniazda antenowego lub gniazda monitora/, dźwięki wytwarzane przez C128 stają się słyszalne /zapoznaj się z rysunkami w rozdziale o urządzeniach peryferyjnych, które pokazują wygląd i przeznaczenie poszczególnych gniazd/.

Możesz również podłączyć wyjście komputera do swojego wzmacniacza stereofonicznego /choć nie otrzymasz dźwięku stereo/. Jest to przydatne zwłaszcza przy korzystaniu z monitora bez fonii. Przy odpowiednim programie, po włączeniu wzmacniacza możesz otrzymać oszałamiające efekty: ty /oraz Twoi sąsiedzi - jeśli nie będziesz ostrożny/ przysięgną, że przelatuje nad wami międzygalaktyczny krążownik. Nawet przy małej głośności wzmacniacz stereofoniczny pozwoli na usłyszenie dźwięku lepszej jakości, zarówno w zakresie bardzo wysokich, jak i bardzo niskich tonów, do generacji których zdolny jest Commodore 128.

#### Komercyjny osprzęt i oprogramowanie do generacji dźwięków

Na rynku istnieje cała gama programów muzycznych i dodatkowych urządzeń, jak na przykład klawiatury, dostępne już dla Commodore'a 64. Wszystkie będą mogły być stosowane do Commodore'a 128 w trybie C64, a niektóre także w trybie C128. Producenci rozpoczynają też produkować urządzenia przeznaczone specjalnie dla trybu C128.

W tej części ogólnie omówimy do czego niektóre z tych urządzeń są zdolne. Zamiast omawiania poszczególnych urządzeń, będziemy się raczej starali pokazać, jakie czoły można za ich pomocą nadać komputerowi oraz czego szukać przy zakupie programów muzycznych i sprzętu dla Commodore'a 128.

Przy końcu rozdziału umieszczamy spis firm wytwarzających przystawki i oprogramowanie, umożliwiające wykorzystanie wszystkich cech układu SID.

#### Dodatkowe klawiatury

Istnieje możliwość używania zwykłych klawiszy Commodore'a 128 do komponowania i grania muzyki. Niektóre z dostępnych przystawek wykorzystują tę właściwość. Czasem używany jest tylko jeden rząd klawiszy, udostępniający jedną oktawę. Inne przystawki wykorzystują wszystkie cztery rzędy klawiszy, przypisując każdemu z nich inną oktawę.

Jednak trudno używać komuś, kto przywykł do gry na pianinie lub innym instrumencie klawiszowym, klawiszy drukarki w celu grania muzyki. Rozwiązanie tego problemu niektórzy producenci widzą w klawiaturach podobnych do tych, jakie występują w pianinie, a które można podłączyć do C128. Wybór odpowiedniego typu zależy od tego, jak poważnie traktujesz swoją grę "na" Commodore 128 oraz od tego, jak gruby jest Twój portfel.

Najprostsza i najtańsza z klawiatur nakładana jest wprost na klawiaturę C128 /nie całe 50 dolarów/. Wygląda jak miniaturowa klawiatura pianina, ale przyciśnięcie każdego z klawiszy powoduje jednocześnie przyciśnięcie jednego ze standardowych klawiszy C128. Specjalne oprogramowanie interpretuje te przyciski i przekłada je na notację muzyczną. Nauka gry na takiej klawiaturze nie jest tak łatwa jak nauka gry na prawdziwym pianinie czy klawiaturze syntezatora, a i sama gra na niej trwa dłużej. Jednakże trening czyni mistrza i są wirtuozi nawet tego skromnego typu muzykowania.

Bardziej wyrafinowane są klawiatury o charakterze membranowym, podłączone do portów joysticków w C128. Są one wygodniejsze w użyciu niż klawiatura nakładana, ale daleko im jeszcze do doskonałości klawiatur syntezatorów. Ich koszt waha się

w granicach od 50 do 100 dolarów.

Jeśli stać Cię by wydać 150-300 dolarów, możesz kupić klawiaturę rywalizującą z syntezatorami klasy profesjonalnej. Wiele firm produkuje takie klawiatury. Obejmują one zwykle kilka oktaw.

#### Oprogramowanie muzyczne

Oprogramowanie muzyczne dostępne dla Commodore'a 128 spełnia kilka funkcji. Omówimy te funkcje po kolei, chociaż niektóre programy spełniają po kilka z nich. Niektórzy producenci oferują specyficzne urządzenia, służące pojedynczym celom, po niższych cenach niż urządzenia złożone.

#### Proste instrumenty muzyczne

Commodore 128 może być używany jako instrument muzyczny. Na zwykłej klawiaturze C128 lub na dodatkowej klawiaturze nakładanej, można grać melodię, a C128 generuje dźwięki nut i wysyła je w celu wzmocnienia do odbiornika telewizyjnego, monitora lub wzmacniacza. W tym trybie C128 działa tak, jak zwykły syntezator. Można na nim grać, używając różnych głosów i instrumentów.

Niektóre zestawy oferują sekcję rytmiczną, która gra razem z Tobą, dając Ci możliwość wyboru między rytmem rocka, walca, samby itd. Możesz też zmieniać oktawy, wprowadzać zmiany tempa, vibrato oraz modelować dźwięki, generowane przez C128 tak, aby naśladowały rozmaite instrumenty, wszystko to w trakcie normalnej gry. Niektóre urządzenia pozwalają na zagranie kilku nut lub nut i akordów jednocześnie. Jednak nie więcej niż trzy nuty mogą być zgrane naraz, ponieważ Commodore 128 posiada tylko trzy głosy.

### Urządzenie odtwarzające

Wielu producentów programów posiada przygotowane albumy muzyki, które możesz odtwarzać na Commodore 128 tak, jakby był on adapterem lub magnetofonem. Muzyka zazwyczaj jest nagrana na dyskach i musi być odtwarzana za pośrednictwem programu nadzorującego, który przetwarza informacje zawarte na dyskietkach na muzykę. Prawie każdy gatunek muzyki został nagrany specjalnie dla C128 - rock, jazz, country, muzyka klasyczna, itd.

Możesz potraktować te albumy jako podstawę do własnych eksperymentów. Mogą one być przepisywane za pomocą opisanych niżej programów komponujących. Można wyeliminować jeden z głosów danej kompozycji i samemu zagrać tę część. Można wreszcie zmieniać głosy, instrumentację i rytm, by przekonać się, jakie efekty można samemu stworzyć.

Wiele spośród tych programów odtwarzających pokazuje na ekranie, jakie nuty są w danym momencie wygrywane. Nuty te są zazwyczaj wyświetlane na pięciolinii. W innych programach, muzyce towarzyszy bardziej złożony obraz - od prostych kształtów i kolorów do bogatej animacji. W efekcie oglądasz video całkowicie tworzone przez komputer.

### Urządzenie komponujące

Najbardziej interesującym oprogramowaniem muzycznym dla Commodore'a 128 są programy pozwalające Ci na komponowanie własnej muzyki. Notacja muzyczna jest od razu magazynowana w pamięci, co umożliwia dalsze jej modyfikowanie. Przebieg tego procesu jest różny, w zależności od rodzaju konkretnego programu. Niektóre urządzenia wymagają wpisywania nut poprzez klawiaturę, w innych nuty na pięciolinii są umieszczane na ekranie za pomocą joysticka. Możesz wykorzystywać półtony, bemole oraz nuty o różnej długości - półnuty, ósemki i szesnastki.

Możesz komponować w trzech różnych głosach, a każdemu głosowi przypisać dźwięk innego instrumentu. Po skomponowaniu każdego poszczególnego fragmentu, możesz go przesłuchać aby przekonać się, jak brzmi. Jeśli Ci się nie spodoba, możesz go przeredagować i ponownie przesłuchać itd.

Wiele spośród programów umożliwiających grę i komponowanie muzyki pozwala też dokonać wydruku zapisu nutowego na drukarce. Właściwość ta jest bezcenna zwłaszcza dla zawodowych kompozytorów. Po ukończeniu kompozycji, może ona być wydrukowana i wysłana wprost do menadżera lub do studia. Typowy wydruk będzie miał wszystkie profesjonalne cechy notacji muzycznej. Klucz basowy i wiolinowy, półtony i bemole, nuty o różnej długości, oznaczenia nutowe itd.

### Nauczanie muzyki

Istnieje na rynku kilka programów, które nauczą Cię muzyki lub pokażą Ci, jakie możliwości muzyczne tkwią w układzie SID. Jest to o wiele ciekawszy sposób uczenia się muzyki niż klasyczne metody szkolne.

Pokazaliśmy Ci, co można osiągnąć korzystając z dostępnego na rynku oprogramowania. Jeśli pragniesz sam komponować muzykę, z następnych rozdziałów dowiesz się, jak działa dźwięk w Commodore 128, i jak można wykorzystać BASIC 7.0 do kontrolowania dźwięków.

### Pojęcia związane z dźwiękiem

Jeśli chcesz pisać swoje własne programy tworzące dźwięki, będziesz musiał dowiedzieć się czegoś o dźwięku w ogóle oraz zrozumieć, jak układ SID "myśli" o dźwięku. Taki jest cel tej części podręcznika. W kolejnej części pokażemy Ci, których rozkazów BASIC'u używać w celu uży-



skania opisanych tu efektów. Jeśli już masz wprawę w programowaniu dźwięku na C64, a chcesz jedynie się dowiedzieć czegoś o rozkazach stosowanych do tworzenia dźwięku w BASIC'u 7.0, możesz opuścić tę część i przejść do następnej, zatytułowanej "Dźwięk w BASIC 7.0".

Proste dźwięki muzyczne charakteryzowane są trzema cechami: częstotliwością, natężeniem oraz czasem trwania. Rozpatrzymy teraz każdą z tych cech.

### Częstotliwość

Jedną z najważniejszych cech dźwięku jest jego częstotliwość. Częstotliwość określa wysokość danego dźwięku - flet ma wyższy dźwięk niż kontrabas, to znaczy, że wytwarza dźwięk o wyższej częstotliwości. Częstotliwość mierzona jest w Hertzach /Hz/. Słuch ludzki przeciętnie obejmuje zakres ok. 40-20 000 Hz.

Commodore 128 nie może bezpośrednio wygenerować tak szerokiego zakresu częstotliwości, jego możliwości kończą się przy około 4000 Hz. Jednakże, wytwarzane przez układ SID harmoniczne, sięgają znacznie dalej /pojęcie harmonicznych - wyjaśnimy za chwilę/.

Kiedy programujesz komputer, aby wygenerować jakiś dźwięk, musi podać układowi SID pewną liczbę, określającą częstotliwość dźwięku, który chcesz uzyskać. Liczba ta nie odpowiada bezpośrednio częstotliwości wyrażonej w hercach, lecz jest z nią ściśle związana, jak zobaczymy to w części dotyczącej BASIC'a. Wszystkie dźwięki składają się z fal rozchodzących się w powietrzu. Fale te można porównać do fal oceanicznych z ich wciąż powtarzającymi się szczytami i dolinami. W dźwiękach o wyższej częstotliwości szczyty fal są położone w mniejszej odległości od siebie, przy dźwiękach niższych oddalają się. Wspomnieliśmy o harmonicznych. Czym one są? N.e-

malże każda nuta w muzyce składa się nie tylko z podstawowej częstotliwości danej nuty, lecz również z szeregu innych częstotliwości. Te dodatkowo pojawiające się częstotliwości są wielokrotnościami częstotliwości podstawowej i właśnie one nazywane są harmonicznymi. Pierwsza harmoniczna ma częstotliwość dwukrotnie większą od samej nuty, druga - trzykrotnie itd. Harmoniczne nie są tak głośne jak częstotliwości podstawowe. Te, które harmoniczne są głośniejsze, a które cichsze, jest uzależnione od konkretnej nuty: jej brzmienia i barwy. Możliwość zmiany tych parametrów decyduje o tym, że SID jest w stanie wygenerować dźwięki naśladujące różne instrumenty.

### Natężenie

Natężenie /czasem nazywane "głośnością albo "amplitudą"/ jest najbardziej oczywistą cechą dźwięku obok częstotliwości. Jeśli wyobrazisz sobie dźwięk jako serię fal na morzu, to natężenie będzie wysokością tych fal. Możesz programowo kontrolować natężenie dźwięku wytwarzanego przez Commodore'a 128 tak jakbyś sciszał radio za pomocą gałki. Można osobno regulować natężenie poszczególnych dźwięków. Ale więcej powiemy o tym później.

### Czas trwania

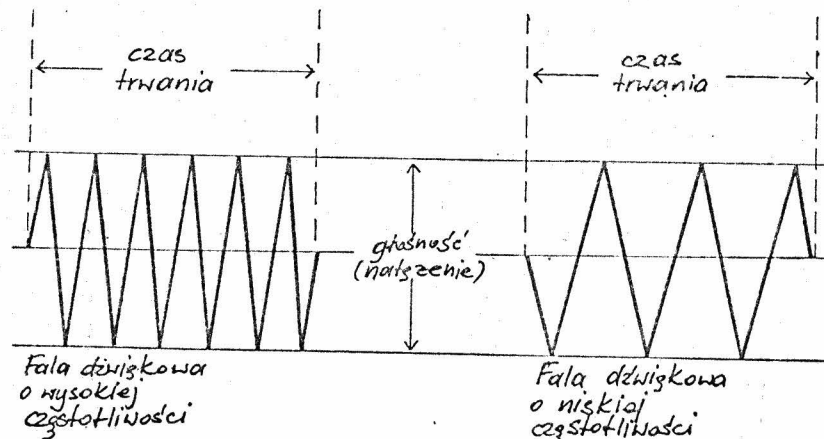
Każdy dźwięk posiada określony czas trwania. Trzask zamkniętej gałki jest np. krótszy niż zawołanie gwizdka pociągu. Gdy będziesz używał C128 do tworzenia dźwięku, będziesz musiał pokazać mu liczbę, określającą czas jego trwania.

W muzykologii uwzględnia się kilka rzeczy przy określaniu czasu trwania. Po pierwsze - tempo. Tempo określa długość taktu. Takt jest podstawową jednostką notacji mu-

zycznej. W każdym takcie mieści się określona ilość nut. Istnieją całe nuty, półnuty /o połowę krótsze od całej nuty/, ćwierćnuty /o połowę krótsze od półnuty/, ósemki i szesnastki. Dokładna ilość nut w danym takcie zależy od oznaczenia takto-  
wego. Jeśli dany utwór ma być grany w takcie na 4/4, to w tak-  
cie będą 4 ćwierćnuty /lub 2 półnuty lub 1 cała nuta/. Jeśli  
melodia jest napisana w takcie na 3/4 /walc/, to w takcie bę-  
dą 3 ćwierćnuty.

A więc jeśli chcesz określić czas trwania danej nuty,  
musisz sprecyzować zarówno tempo, jak i rodzaj nuty. O notacji  
muzycznej powiemy więcej w następnej części, przy omawianiu  
rozkażu PLAY.

Rys. 30 przedstawia częstotliwość, czas trwania, natęże-  
nie kilku typowych fal akustycznych.



Rys.30 - Parametry fali dźwiękowej

Gdybyś podłączył mikrofon do oscyloskopu /urządzenie  
umożliwiające obserwację przebiegów elektrycznych/ i przy-  
tknął go do źródła dźwięku, zobaczyłbyś fale o mniej wię-  
cej takim kształcie. Zazwyczaj jednak fale wytwarzane przez  
instrumenty muzyczne są bardziej złożone.

Użytkownikom wielu komputerów osobistych, ta skromna  
wiedza o dźwięku wystarczy. Komputery te mogą generować je-  
dynie proste dźwięki o różnych wysokościach. Nie są one  
zdolne do bardziej wyrafinowanych działań, niezbędnych do  
wiernego naśladowania różnych instrumentów muzycznych oraz  
innych efektów dźwiękowych.

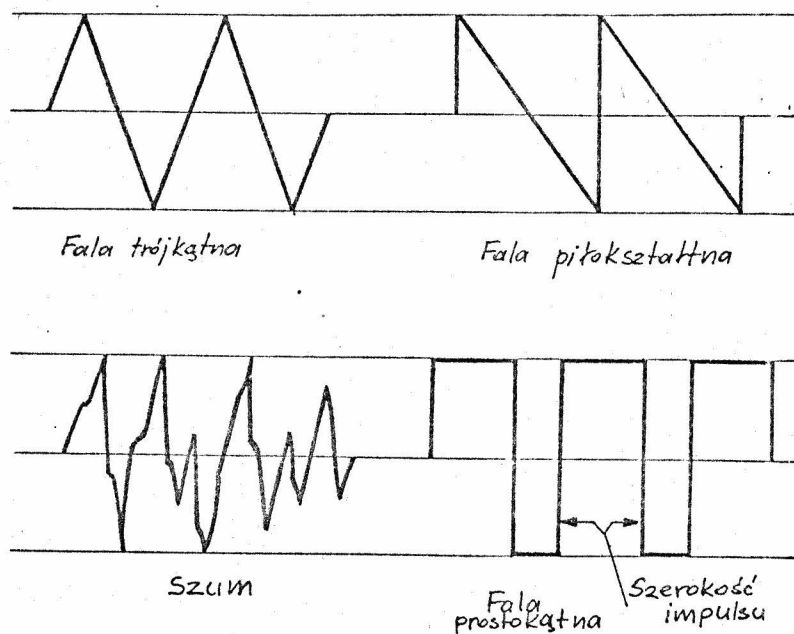
#### Kształty fal

Na rys. 31 przedstawiliśmy jedynie falę trójkątną.  
Jednak układ SID może generować inne rodzaje fal - piło-  
kształtną, prostokątną lub nieregularną /szum/. Wszystkie  
4 rodzaje przedstawia rys.31.

Fale piłokształtne i trójkątne tworzą proste dźwięki  
melodyczne. Dźwięki fali trójkątnej są łagodniejsze, podob-  
ne do tych, jakie wydają instrumenty drewniane. Fala piło-  
kształtna wydaje dźwięki bardziej zbliżone do brzmienia in-  
strumentów blaszanych /różnice w brzmieniu są spowodowane  
różną zawartością harmonicznych w tych dźwiękach/.

Fala szumowa ma zupełnie inne zastosowanie. Produkuje  
głównie dźwięki dudniące i syczące. Stosowana jest do two-  
rzenia takich efektów, jak wybuchy, wystrzały i lawiny.  
Osiąga się je dzięki bardzo nieregularnemu kształtowi fali.  
Tworzy ona tzw. "biały" szum, który jest mieszaniną wielu  
różnych częstotliwości i brzmi, jakby był statyczny. Niskie  
częstotliwości tworzą dźwięki dudniące, podczas gdy wyso-  
kie - dźwięki typu "shhh" i "ssss". Łącząc biały szum z ob-  
wiednią /opisaną poniżej/ lub z filtrami, można stworzyć  
niesamowitą różnorodność efektów dźwiękowych.

Fala prostokątna tworzy dźwięki, które najlepiej imitują pewne instrumenty takie jak np. fortepian. Bardzo ważne jest sprecyzowanie szerokości impulsów. Różne szerokości impulsu /decydujące o tzw. współczynniku wypełniania/ dają różne efekty.



Rys.31 - Ctery różne kształty fal

### Obwiednia

Gdy na jakimś instrumencie muzycznym /np. na skrzypcach/ wygrywana jest pewna nuta, nie zaczyna się ona nagle, trwa z określonym natężeniem przez jakiś czas i nagle się urywa, jak by to sugerował rys. 32. Gdybyś spojrział na oscyloskopie na obraz dźwięku wytworzonego przez prawdziwy instrument muzyczny, zobaczył byś, że dzieje się coś zupełnie innego, bardziej skomplikowanego. Wyobraź sobie, że każda nuta ma swoje własne życie: przechodzi przez kilka etapów. Człowiek np. rodzi się, rośnie, staje się coraz silniejszy, traci część swych sił, ale trwa przez kilka dziesiątków lat aż w końcu umiera. Nuta przechodzi podobny proces, a jego poszczególne etapy noszą nazwy: atak, zanik, podtrzymanie oraz wybrzmiewanie /attack, decay, sustain, release: ADSR/.

Po złożeniu tych czterech części składowych nuty w jedną całość, otrzymamy tak zwaną obwiednię. Rozpatrzmy dokładniej elementy obwiedni.

### A t a k

Dźwięk nuty zagranej na skrzypcach najpierw przez pewien czas wzrasta od zera do pewnego szczytowego natężenia. Ten czas nazywany jest atakiem i może być potraktowany jako etap wzrostu danej nuty. Atak nie trwa zbyt długo /mierzony jest w milisekundach, czyli tysięcznych częściach sekundy/, ale wpływa na ostateczny dźwięk nuty. Na przykład cymbały mają krótszy atak niż skrzypce czy flet.

### Z a n i k

Gdy nuta osiągnęła już swoje maksymalne natężenie, natychmiast zaczyna zanikać. Jest to tak, jakby człowiek zaczął tracić maksymalną sprawność sportową gdy dorośnie - szczytowe zdolności zanikają prawie zaraz po ich osiągnięciu.

### Podtrzymanie

W czasie fazy zaniku, nuta traci część swojej siły, ale nie całą. Spada ona do poziomu trochę niższego od szczytu. Poziom ten nazywa się poziomem podtrzymania. Zwróć uwagę, że podtrzymanie odnosi się do poziomu natężenia, a nie do okresu czasu, jak to miało miejsce w przypadku ataku i zaniku. Czas, w którym nuta utrzymuje się na poziomie podtrzymania uzależniony jest od czasu trwania nuty, jak to wyjaśniono wyżej.

### Wybrzmiewanie

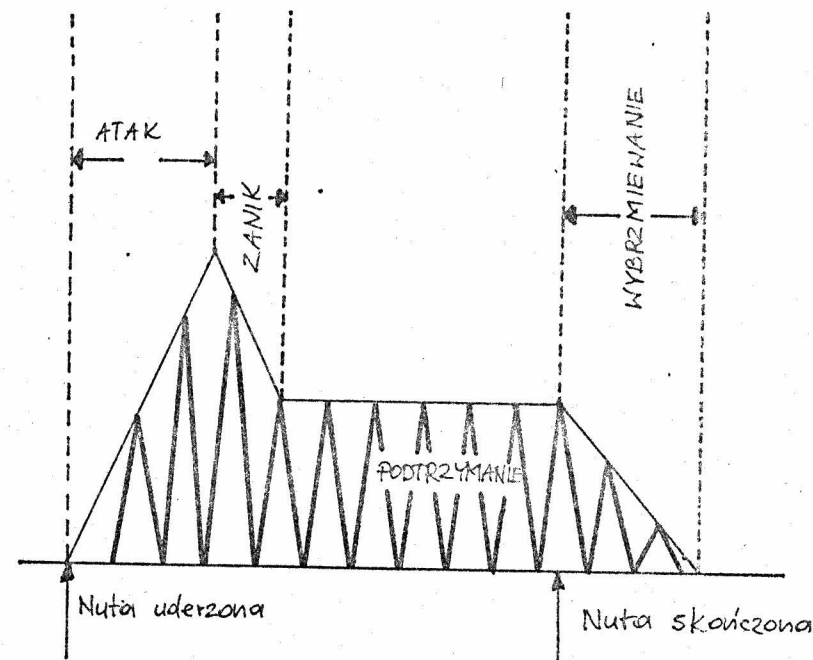
Cdy nuta przebrzmiała przez pewien czas /czas trwania/, powoli zanika, nie urywa się nagle, ale bardziej lub mniej stopniowo zamiera.

### Złożenie wszystkich etapów razem

Rys. 32 przedstawia obwiednię nuty wraz z fazami ataku, zaniku, podtrzymania i wybrzmiewania.

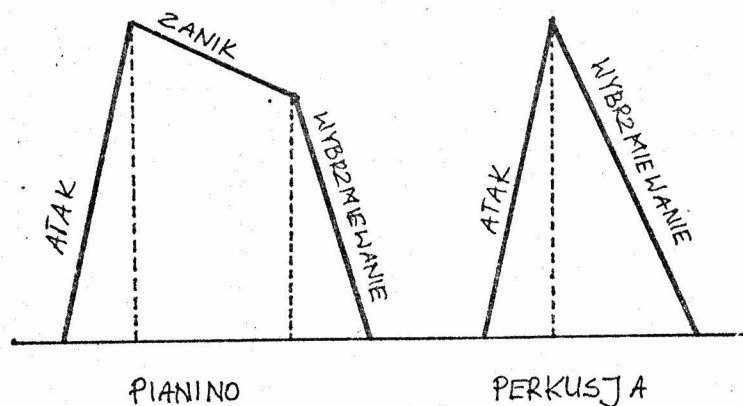
Zwróć uwagę, że obwiednia służy jedynie kształtowaniu istniejących fal akustycznych. Fale akustyczne i obwiednie, to dwie różne rzeczy. Fale akustyczne posiadają swoją własną częstotliwość i czas trwania. Obwiednia określa jedynie, w jaki sposób fale będą narastały i zamierały. Zauważ, że na wykresach fal akustycznych pokazaliśmy górne i dolne ich części, podczas gdy na wykresie obwiedni widać jedynie górne części tych fal. Upraszczając to wykres. Poszczególne fale nadal oscylują po obu stronach linii ciągłej, mimo iż nie zostało to pokazane.

Nie wszystkie instrumenty wykorzystują wszystkie cztery fazy obwiedni. Wykorzystują je instrumenty drewniane /np. klarnety i flety/, smyczkowe /skrzypce/ oraz dęte /trąbki i waltonie/. Ale niektóre instrumenty uderzane /percussion instrument/ w ogóle nie posiadają fazy podtrzymania. W przypadku



Rys. 32 - Obwiednia nuty

fortepianu /instrument uderzany/ zastępuje ją faza zaniku. W efekcie tego dźwięk szybko narasta do wartości maksymalnej, później powoli opada na zboczu zaniku, a następnie w fazie wybrzmiewania osiąga punkt zerowy. Lecz w przypadku bębnow i cymbałów fazy zaniku i wybrzmiewania są połączone tak, że wykres obwiedni nagle się wznosi, a następnie natychmiast opada do zera. Pokazuje to rys.33.



Rys.33 - Obwiednia dla pianina i perkusji

Zwykle dla jednego instrumentu istnieje kilka podobnie brzmiących wykresów obwiedni. Często można pominąć fazę zaniku albo fazę wybrzmiewania i uzyskać mniej więcej ten sam efekt.

W następnej części tego rozdziału, poświęconej BASIC'owi 7.0 nauczysz się jak naśladować dźwięki poszczególnych instrumentów, projektując własne obwiednie, to jest nadając atakowi, zanikowi, podtrzymaniu i wybrzmiewaniu określone wartości. Commodore posiada definicje obwiedni dla niektórych instrumentów wbudowane wprost do BASIC'a. Dla tych instrumentów nie musisz już więc ich projektować.

#### Odchylenie /Sweep/

Inną metodą modyfikowania nut przez układ SID jest wprowadzanie odchylenia. Odchylenie pozwoli Ci rozpocząć nutę od pewnej częstotliwości, a następnie zmieniać częstotliwość w trakcie jej trwania. Należy poinformować SID, czy chcesz by odchylenie działało w górę czy w dół, na jakiej częstotliwości chcesz zakończyć daną nutę oraz jaki ma być krok /tzn. co ile Hz ma się zmieniać częstotliwość zanim osiągnie końcową wartość/. Wybór kroku decyduje o szybkości zmiany.

#### Głosy

Wspomnieliśmy już, że układ SID może produkować trzy odrębne głosy lub kanały. Każdy z tych głosów może być programowany niezależnie od pozostałych. To tak, jakbyś posiadał trzy odrębne układy scalone. Każdemu głosowi można przypisać inną częstotliwość, czas trwania i obwiednię. Zwróć jednak uwagę, że regulator siły głosu w układzie SID oddziałuje na wszystkie głosy jednocześnie. Możesz określić poziom podtrzymania każdego z głosów z osobna, ale ogólny regulator głosu wpływa na wszystkie jednakowo.

## Filtry

Istnieją inne sposoby modyfikowania dźwięków tworzonych przez układ SID poza definiowaniem obwiedni. Jednym z nich jest stosowanie filtrów.

Filtr jest obwodem elektrycznym, który nie przepuszcza pewnych częstotliwości. Filtrami są na przykład regulatory niskich i wysokich tonów w Twoim zestawie stereofonicznym. Gdy ustawisz regulator wysokich tonów na minimum, dźwięki o wyższej częstotliwości są wyciszane, a gdy ustawisz regulator niskich tonów na minimum, wytłumione zostaną dźwięki o niższej częstotliwości. W podobny sposób możesz wykorzystać układ SID do modyfikowania barwy dźwięku.

### Filtry górno- i dolnoprzepustowe

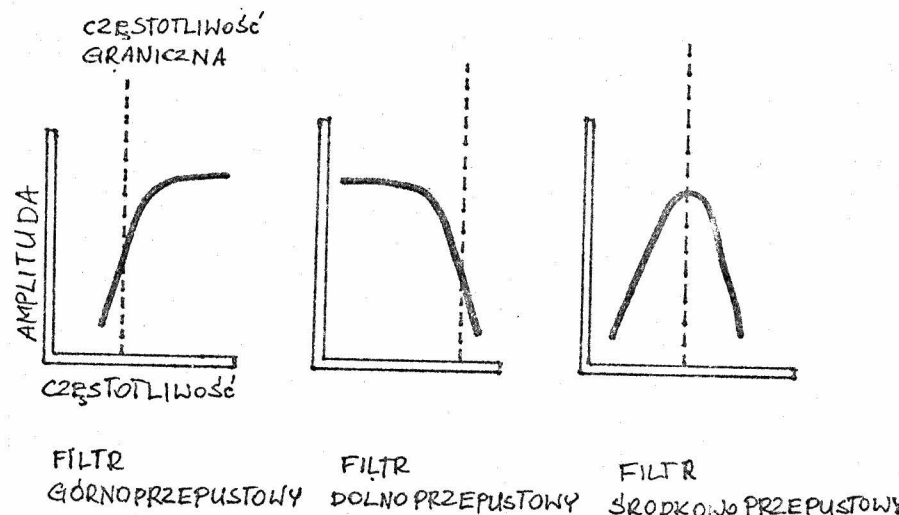
Istnieją trzy typy filtrów wbudowanych do układu SID - górnoprzepustowe, dolnoprzepustowe oraz środkowoprzepustowe. Gdy będziesz używał filtra górnoprzepustowego, zablokujesz dźwięki o niskiej częstotliwości, tak jakbyś ustawił regulator niskich tonów w swoim sprzęcie stereofonicznym na minimum. Powstają w ten sposób metaliczne dźwięki o wysokiej częstotliwości. Filtr dolnoprzepustowy natomiast blokuje wysokie częstotliwości, dając w efekcie głębokie, mocne tony.

### Częstotliwość graniczna

O jakich konkretnie częstotliwościach mówimy opisując filtrowanie wysokich i niskich częstotliwości? Gdy przekreślisz gałkę na swoim zestawie stereo nie wiesz dokładnie jakie częstotliwości zostają wyeliminowane. Układ SID jednak pozwala Ci dokładnie określić punkt odcięcia. Jeśli na przykład używasz filtra górnoprzepustowego i określisz częstotliwość graniczną na 1000 Hz, to wszystkie częstotliwości poniżej 1000 Hz zostaną osłabione lub całkowicie wyeliminowane. Podobnie, jeśli

używasz filtra dolnoprzepustowego z taką samą częstotliwością graniczną, to zostaną wyciszone wszystkie dźwięki o częstotliwości powyżej 1000 Hz.

Obwody elektryczne pełniące funkcje filtrów nie są doskonałe, a więc częstotliwości graniczne nie są dokładne. Częstotliwości, które znajdują się poza pasmem przepuszczania filtru będą do pewnego stopnia słyszalne, ale im dalej dana częstotliwość znajdzie się od częstotliwości granicznej, tym będzie słabsza. Rys. 34 pokazuje jaki efekt wywierają filtry na poziomy dźwięku w pewnym zakresie częstotliwości.



Rys. 34 - Filtry



### Filtr środkowoprzepustowy

SID posiada trzeci rodzaj filtra, zwany filtrem środkowo-przepustowym. Jak sama nazwa wskazuje, tłumi on zarówno wysokie, jak i niskie częstotliwości, przepuszczając tylko te o wartościach środkowych.

Rys.34 pokazuje wpływ jaki wywierają te trzy rodzaje filtrów na amplitudę dźwięków o różnych częstotliwościach.

Łącząc filtr górnoprzepustowy z dolnoprzepustowym otrzymasz filtr, który będzie przeciwieństwem filtra środkowoprzepustowego: będzie on tłumił /a nie przepuszczał/ średnie częstotliwości.

Inną cechą filtrów, którą można manipulować, jest rezonans. Określa on, co będzie się działo z dźwiękiem w pobliżu częstotliwości granicznej. Za pomocą rezonansu można wpływać na ostryść otrzymywanego dźwięku.

Zwróć uwagę na fakt, że filtry wpływają na wszystkie trzy głosy jednocześnie. A więc nie można za pomocą filtru podkreślić niskich tonów w jednym głosie i wysokich w drugim.

### Modulacja kołowa i synchronizacja

Istnieje możliwość łączenia dwóch głosów ze sobą, by uzyskać interesujące efekty muzyczne. W modulacji wyjścia dwóch głosów są dodawane do siebie. Wywołuje to efekt wibrowania dźwięku słyszalnego np. w przypadku dzwonów i gongów. Przy wysynchronizacji natomiast, dwa głosy są dodawane do siebie logicznie za pomocą instrukcji AND. W ten sposób można stworzyć dźwięk, który będzie wznosił się i opadał, podobny np. do dźwięku wydawanego przez komara oraz inne złożone struktury harmoniczne.

Bardzo ogólnie omówiliśmy proces tworzenia dźwięków na Commodore 128. Istnieje mnóstwo innych sposobów generowania dźwięku, ale spójrzmy teraz na specyfikę generowania dźwięków za pośrednictwem BASIC'u.

### Dźwięk a BASIC 7.0

W tej części omówimy rozkazy, których używa BASIC 7.0 by panować nad zdolnościami akustycznymi i muzycznymi układu SID. Jest sześć takich rozkazów: VOL, SOUND, ENVELOPE, PLAY, TEMPO oraz FILTER. Opiszemy każdy z tych rozkazów po kolei.

#### Rozkaz VOL

Jak wspomnieliśmy poprzednio, natężenie wszystkich trzech głosów może być kontrolowane przez jeden rozkaz BASIC'u. Rozkazem tym jest właśnie VOL /volume natężenie/. Jest to jeden z prostszych rozkazów, ponieważ posiada tylko jeden parametr. Oto typowa linia programu w BASIC'u, zawierająca ten rozkaz:

128 VOL 8

Natężenie może przyjmować wartości od 0 do 15, przy czym 0 oznacza brak dźwięku, a 15 jego maksimum.

Ponieważ możesz regulować całkowite natężenie dźwięku za pomocą pokrętła telewizora lub aparatury stereo, być może zechcesz nastawiać natężenie na najwyższy poziom i regulować głośność poszczególnych nut zmianą poziomu podtrzymania. VOL przydatny jest w tworzeniu efektów dźwiękowych oraz włączaniu i wyłączaniu dźwięków. Zwróć jednak uwagę na to, że zanim użyjesz rozkazu SOUND do wytworzenia dźwięku, musisz najpierw użyć rozkazu VOL. Nie wydobędziesz żadnego dźwięku z układu SID, jeśli nie ustawisz VOL na natężenie różne od zerowego.

#### Rozkaz SOUND

Rozkaz SOUND jest podstawą umiejętności akustycznych BASIC'a 7.0. Za jego pomocą możesz ustawić częstotliwość, czas trwania odchylenia /SWEEP/ oraz kształt fali dźwięku. Na ogół nie jest on stosowany do grania muzyki /do tego celu

stosuje się rozkaz PLAY/. SOUND używa się dla otrzymywania efektów dźwiękowych oraz w sytuacjach, kiedy pragniesz bardzo precyzyjnie panować nad częstotliwością i innymi cechami dźwięku przez Ciebie wytwarzanego.

#### Prosta odmiana rozkazu SOUND

W swej najprostszej formie, SOUND może być wykorzystywany jedynie w celu nastawiania częstotliwości oraz czasu trwania dźwięku. Musi też sprecyzować, którego głosu chcesz używać, gdyż każdy dźwięk musi być wytworzony przez jeden z trzech generatorów układu SID. Na przykład linia programu:

100 SOUND 2, 4291, 60

spowoduje dźwięk o częstotliwości średniego C, wytworzony przez generator /głos nr 2/, trwający jedną sekundę. Przyjrzyjmy się trzem parametrom, zawartym w tym poleceniu.

Pierwszy parametr /cyfra 2/ określa głos. Może on mieć wartość od 1 do 3.

Drugi parametr określa częstotliwość. Jego wartość waha się od 0 do 65535. Niestety liczba ta nie odpowiada rzeczywistej częstotliwości. Zależność między liczbą a częstotliwością można wyrazić za pomocą następującego krótkiego wzoru:

wartość parametru = częstotliwość / 0.06097

Zakładając, że wartość parametru częstotliwości waha się od 0 do 65535, wartość właściwej częstotliwości zawiera się w przedziale od 0 do ok. 4000 Hz.

Środkowe C w skali muzycznej ma częstotliwość 262 Hz. Dzieląc tę liczbę przez 0.06097 ustalimy, że wartość parametru określającego tę nutę wynosi 4297.

Tabela 7-1 pokazuje częstotliwości i wartości parametrów dla nut w oktawie rozpoczynającej się od środkowego C.

Tabela 7-1 - Częstotliwości oraz odpowiadające im wartości parametrów dla nut

| Nuta | Częstotliwość właściwa | Parametry dźwięku |
|------|------------------------|-------------------|
| C    | 262                    | 4297              |
| C #  | 277                    | 4543              |
| D    | 294                    | 4822              |
| D #  | 311                    | 5100              |
| E    | 330                    | 5412              |
| F    | 349                    | 5724              |
| F #  | 370                    | 6069              |
| G    | 392                    | 6429              |
| G #  | 415                    | 6807              |
| A    | 440                    | 7217              |
| A #  | 466                    | 7643              |
| B    | 494                    | 8102              |
| C    | 523                    | 8578              |

Wartości parametrów dla nut w następnej oktawie można ustalić przez podwojenie wartości z tabeli 7-1, a dla nut w poprzedniej oktawie, przez podzielenie tych wartości przez 2.

Trzecim parametrem rozkazu SOUND jest czas trwania. Wyrażany jest on w jednostce  $1/60^{\text{a}}$  sekundy. Jeśli więc parametr ten wynosi 60, to oznacza to, że dźwięk będzie trwał jedną sekundę. Wartość 30 da Ci pół sekundy, wartość 120 - dwie sekundy, itd.

\* Dotyczy to tylko komputerów przystosowanych do współpracy z siecią zasilającą o częstotliwości 60 Hz /USA/; zarówno w Polsce jak i większości krajów europejskich obowiązuje standard 50 Hz, w związku z czym opisywana jednostka będzie miała wartość  $1/50$  sekundy /przyp.tłum./

# Ustalanie odchylenia /SWEEP/ oraz kształtu fali

Rozkazu SOUND można używać tak jak to właśnie przedstawiliśmy, tylko z trzema parametrami. Może on jednak również określać odchylenie oraz kształt fali. Rys.35 pokazuje rozkaz SOUND ze wszystkimi parametrami.

Jeśli zostaną podane parametry odchylenia, nuta rozpocznie się od pewnej częstotliwości i przemieści się do innej w czasie swego trwania. Aby wywołać to odchylenie musisz sprecyzować trzy parametry, oprócz głosu, częstotliwości i czasu trwania. Te nowe parametry to: kierunek odchylenia, minimalna częstotliwość odchylenia oraz wielkość kroku odchylenia.

Kierunek odchylenia określa czy chcesz by dźwięk podwyższył swoją częstotliwość w czasie swego trwania /wartość 0/, obniżył ją /wartość 1/, czy też oscylował między wyższą a niższą częstotliwością /wartość 2/.

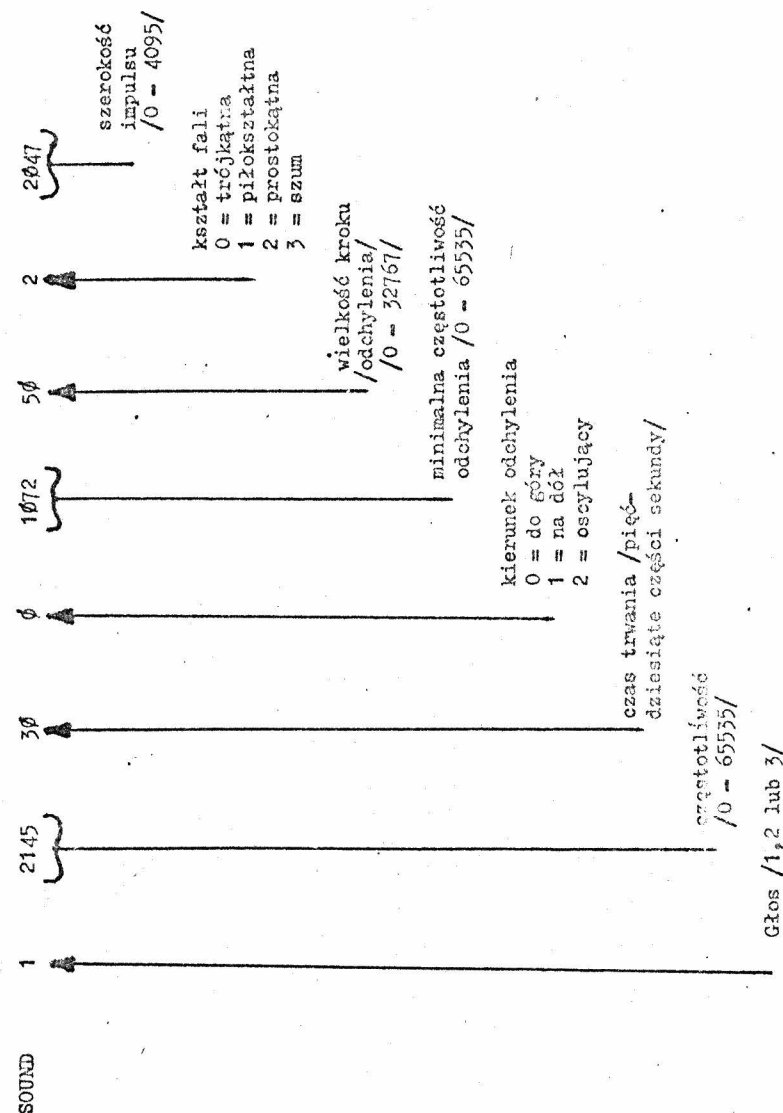
Minimalna częstotliwość odchylenia precyzuje niższą z częstotliwości - dźwięk odchylił się w dół do tej częstotliwości od wartości wyznaczonej przez parametr częstotliwości właściwej albo rozpocznie się od minimalnej częstotliwości i odchylił się do góry /w zależności od tego, jaki kierunek określisz/. Jeśli wybierzesz tryb oscylujący, wysokość dźwięku będzie oscylowała pomiędzy dwiema podanymi wartościami.

Wielkość kroku odchylenia określa jak szybko ono nastąpi. Parametr ten może mieć nawet wartość 32,767, ale zazwyczaj przyjmuje wartości poniżej 100.

Następujący króciutki program spowoduje, że częstotliwość spadnie o jedną oktawę do środkowego C, przy wielkości kroku odchylenia równym 100:

100 VOL 15

110 SOUND 2, 8578, 60, 1, 4291, 100, 0



Rys.35 - Rozkaz SOUND

Za pomocą rozkazu SOUND możesz również wybrać jeden z czterech rodzajów fal wytwarzanych przez układ SID - trójkątne, piłokształtne, prostokątne lub biały szum. Jeśli wybierzesz falę prostokątną, musisz sprecyzować szerokość impulsu. Parametr ten przyjmuje wartości od 0 do 4095. Gdy jego wartość wynosi połowę wartości maksymalnej, tzn. 2048, kształt fali jest kwadratowy, a wytworzony dźwięk jest "najbogatszy". Wartości do niej zbliżone są często wykorzystywane do naśladowania pianina. Inne wartości szerokości impulsu tworzą ostrzejszy, bardziej brzęczący dźwięk.

Przedstawiony niżej program wykorzystuje falę piłokształtną do wytworzenia dźwięku A, powyżej środkowego C. Ponieważ odchylenie w nim nie występuje, wszystkie odnoszące się do niego parametry są równe zeru.

100 VOL 15

110 SOUND 1, 7216, 30, 0, 0, 0, 1, 0

Następny program wytworzy dźwięk podobny do organów za pomocą fali prostokątnej /kwadratowej/

100 VOL 15

110 SOUND 1, 7216, 30, 0, 0, 0, 2, 2047

A oto program, który produkuje krótki dźwięk przypominający wystrzał, wykorzystując do tego celu biały szum. Łącząc ciąg tych dźwięków za pomocą pętli FOR...NEXT wygenerujesz dźwięk przypominający serię strzałów z karabinu maszynowego:

100 VOL 15

110 SOUND 1, 1000, 10, 0, 0, 0, 3, 0

Zauważ, że dźwięk powstaje natychmiast po wydaniu rozkazu SOUND. Będzie on trwał przez czas określony parametrami obwiedni.

### Rozkaz ENVELOPE /Obwiednia/

Teraz, gdy już nauczyliśmy się produkować dźwięki, omówimy sposób tworzenia muzyki na Commodore 128. Podstawowym rozkazem, służącym temu celowi jest PLAY. Zanim jednak zrozumiesz zasadę działania rozkazu PLAY, musisz się nauczyć jak działają pozostałe trzy rozkazy: ENVELOPE, TEMPO oraz FILTER. Omówimy je kolejno.

ENVELOPE definiuje cechy poszczególnych obwiedni. Po zdefiniowaniu obwiedni możesz użyć rozkazu PLAY, aby określić, jakie konkretne nuty mają być zagrane w ramach danej obwiedni. Rozkaz ENVELOPE sam w sobie nie tworzy dźwięku; dostarcza on jedynie definicji, która zostanie później wykorzystana przez rozkaz PLAY. /Aby zrozumieć przeznaczenie rozkazu ENVELOPE musisz znać ogólne zasady działania obwiedni oraz fazy ataku, zaniku, podtrzymania i wybrzmiewania, które zostały omówione w pierwszej części tego rozdziału/.

### Przeddefiniowane instrumenty

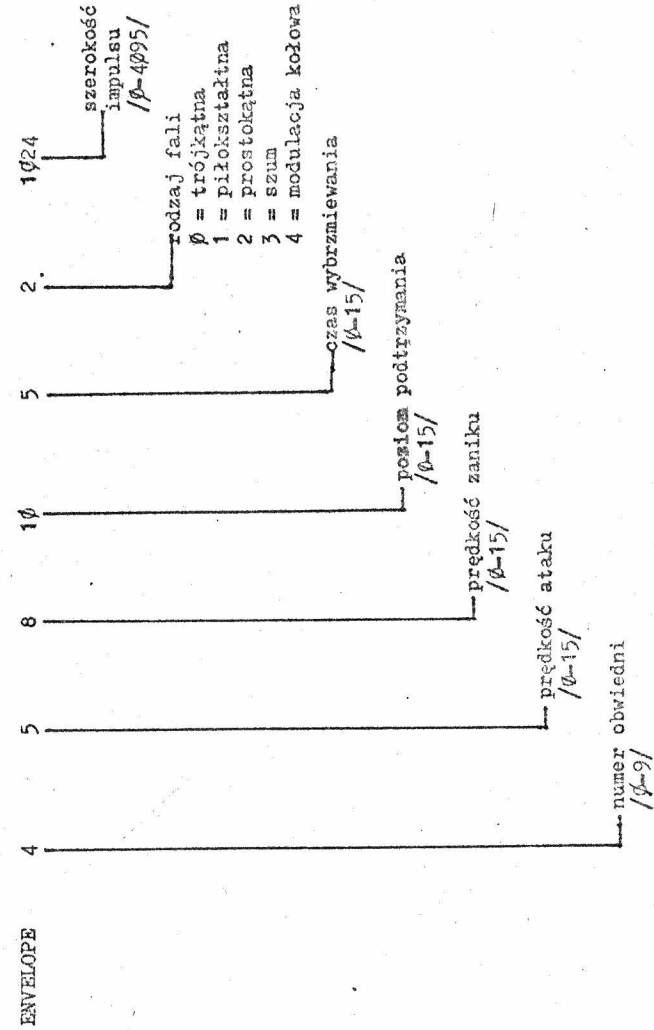
Commodore 128 posiada zapamiętane parametry dla dziesięciu "standardowych" instrumentów /od organów do ksylofonu/. Obwiednie dla tych instrumentów zostały określone z góry i wpisane do pamięci ROM komputera, tak więc nie musisz odzwiercać ich sam. Tabela 7-2 pokazuje wartości ataku, zaniku, podtrzymania i wybrzmiewania oraz kształtów fal dla poszczególnych instrumentów.

Przeznaczeniem rozkazu ENVELOPE jest zmienianie tych przeddefiniowanych obwiedni w celu tworzenia dźwięków innych instrumentów. Należy po prostu wykonać rozkaz ENVELOPE, zawierający numer obwiedni, którą pragniesz zmienić oraz parametry, które chcesz podstawić pod parametry standardowe.

Tabela 7-2 - predefinedowane instrumenty

| Liczba porządkowa | Instrument    | Atak | Zanik | Podtrzymanie | Wybrzmienie | Kształt fali | Szerokość |
|-------------------|---------------|------|-------|--------------|-------------|--------------|-----------|
| 0                 | Portepian     | 0    | 9     | 0            | 0           | 2            | 1536      |
| 1                 | Akordeon      | 12   | 0     | 12           | 0           | 1            |           |
| 2                 | Organy parowe | 0    | 0     | 25           | 0           | 0            |           |
| 3                 | Bębny         | 0    | 5     | 5            | 0           | 3            |           |
| 4                 | Flet          | 9    | 4     | 4            | 0           | 0            |           |
| 5                 | Gitara        | 0    | 9     | 2            | 1           | 1            |           |
| 6                 | Klawikord     | 0    | 9     | 0            | 0           | 2            | 512       |
| 7                 | Organy        | 0    | 9     | 9            | 0           | 2            | 2048      |
| 8                 | Trąbka        | 8    | 9     | 4            | 1           | 2            | 512       |
| 9                 | Ksylofon      | 0    | 9     | 0            | 0           | 0            |           |

Rys. 36 pokazuje różne parametry rozkazu ENVELOPE.



Rys. 36 - Rozkaz ENVELOPE

Pierwszym parametrem w tym rozkazie jest numer obwiedni. Jeśli chcesz zmodyfikować np. standardowy dźwięk fletu, musisz więc nadać temu parametrowi wartość 4.

Następne cztery parametry charakteryzują samą obwiednię. Są to kolejno - atak, zanik, podtrzymanie, wybrzmiewanie /pojęcia te zostały już zdefiniowane w pierwszej części rozdziału/. Wszystkie te parametry mogą posiadać wartość od 0 do 15. Atak, zanik i wybrzmiewanie są charakteryzowane czasem trwania, natomiast podtrzymywanie amplitudą /głośnością/ nuty w fazie podtrzymania. /Właściwy czas trwania tej fazy jest określony przez rozkaz PLAY/.

Czas trwania ataku, zaniku i wybrzmiewania przy określonych wartościach parametrów pokazuje tabela 7-3. /Skrót "ms" oznacza milisekundy, czyli tysięczne części sekundy/.

Ostatnie dwa parametry rozkazu ENVELOPE określają kształt fali oraz, w przypadku fali prostokątnej, szerokość impulsu. Oba są identyczne z parametrami występującymi w rozkazie SOUND, lecz posiadają jedną dodatkową właściwość: oprócz określania czterech typów fal, umożliwiają również określenie modulacji kołowej.

W modulacji kołowej, trzeci głos pełni jak gdyby funkcję obwiedni dla modulowania głosu pierwszego. Spróbuj wybrać bardzo niską częstotliwość dla głosu trzeciego, grając ją jednocześnie normalną nutą w głosie pierwszym. Powinieneś osiągnąć dźwięki wibrujące, podobne do tych, jakie wytwarzają dzwony.

Poniższy rozkaz może posłużyć jako przykład wykorzystywania ENVELOPE. Zmodyfikuje on obwiednię trąbki, wydłużając fazę wybrzmiewania tak, że dźwięk będzie wygasał wolniej oraz podnosząc poziom podtrzymania t 4 do 6.

100 ENVELOPE 8, 8, 9, 6, 3, 2, 512

Tabela 7-3 - Czas trwania ataku, rozkładu i zwolnienia

| Wartość parametru | Czas trwania ataku | Czas trwania rozkładu i zwolnienia |
|-------------------|--------------------|------------------------------------|
| 0                 | 2 ms               | 6 ms                               |
| 1                 | 8 ms               | 24 ms                              |
| 2                 | 16 ms              | 48 ms                              |
| 3                 | 24 ms              | 72 ms                              |
| 4                 | 38 ms              | 114 ms                             |
| 5                 | 56 ms              | 168 ms                             |
| 6                 | 68 ms              | 204 ms                             |
| 7                 | 80 ms              | 240 ms                             |
| 8                 | 100 ms             | 300 ms                             |
| 9                 | 250 ms             | 750 ms                             |
| 10                | 500 ms             | 1.5 s                              |
| 11                | 800 ms             | 2.4 s                              |
| 12                | 1 s                | 3 s                                |
| 13                | 3 s                | 9 s                                |
| 14                | 5 s                | 15 s                               |
| 15                | 8 s                | 24 s                               |

#### Rozkaz TEMPO

Wprowadzenie rozkazu TEMPO można porównać z uruchomieniem metronomu. Rozkaz ten określa długość taktu, a tym samym czas trwania nut, półnut, itd. Oto przykład wprowadzenia tempa o wartości 40:

100 TEMPO 40

Parametr rozkazu TEMPO może przyjmować wartości w zakresie od 0 do 255. Przy wartości 0 nuty będą następowały jedna po drugiej. Wstępnie ustaloną /w momencie włączenia komputera/ wartością tego parametru jest 8. Im większa war



tość parametru TEMPO, tym krótsze nuty. A więc przy wartości 6 uzyskamy wolną muzykę taneczną, podczas gdy wartość 200 może być wprowadzana do gier w celu uzyskiwania bardzo szybkich sekwencji dźwięków.

Zależność między wartościami stosowanymi w rozkazie TEMPO a aktualnym czasem trwania taktu w sekundach, można przedstawić następującym wzorem:

czas trwania /w sekundach/ = 19.22/ustawienie tempa

Jeśli wydasz rozkaz TEMPO 10, takt będzie trwał dwie sekundy. Przy rozkazie TEMPO 80 takt będzie trwał około 1/4 sekundy.

#### Rozkaz FILTER

Omówiliśmy już działanie poszczególnych filtrów - górno-przepustowego, dolnoprzepustowego oraz środkowoprzepustowego. Rozkaz FILTER w BASIC'u 7.0 pozwala na włączenie tych filtrów w układzie SID i przefiltrowanie dźwięku wytwarzanego przez nasz program zanim przedostanie się on na zewnątrz.

Oto typowy przykład rozkazu FILTER:

100 FILTER 1000, 0, 0, 1, 10

Pierwszym parametrem tego rozkazu jest częstotliwość graniczna. Parametr ten może mieć wartość od 0 do 2047. Kolejne trzy parametry funkcjonują jako przełączniki i mogą mieć wartość 0 /wyłączony/ lub 1 /włączony/. Pierwszy z nich jest przełącznikiem filtru dolnoprzepustowego, drugi - środkowoprzepustowego, a trzeci górno-przepustowego. Zwróć uwagę, że można włączyć kilka filtrów jednocześnie. Piątym i ostatnim parametrem jest rezonans. Jego wartość waha się od 0 do 15.

Powyższy rozkaz nastawia więc częstotliwość rezonansową na 1000, wyłącza filtry dolno- i środkowoprzepustowe, włącza filtr górno-przepustowy oraz nadaje rezonansowi wartość 10.

#### Rozkaz PLAY

Rozkaz PLAY posiada zdumiewające możliwości. Pozwala nie tylko grać sekwencje, ale też wybrać obwiednię, grać wieloma głosami naraz oraz określić typ filtra.

#### N u t y

Nuty, które ma zagrać rozkaz PLAY zapisywane są w postaci odpowiadających im liter. Litery te są zapisywane w ciągach.

Rozkaz:

100 PLAY "C D E C"

zagra więc pierwsze cztery nuty melodii "Frère Jacques" w tonacji C. W skład ciągu wchodzi wszystko, co zawarte jest w cudzysłowie po PLAY. Odstęp między nutami nie są właściwie istotne, ale dobrze je stosować dla czytelności.

Możesz również wprowadzać półtony i bemole. Oto te same nuty w tonacji D, z nutą F#:

100 PLAY "D E # F D "

Zwróć uwagę, że znak bemola musi poprzedzać nutę, inaczej niż to jest w przypadku zwykłej notacji muzycznej. Półtony są opisywane znakiem #.

Ponadto możesz również grać z pauzami, tzn. brakiem dźwięku, który trwa tyle samo ile poszczególne nuty. Pauza oznaczana jest literą R i jest traktowana tak jak pozostałe nuty.

#### Taktowanie

Do tej pory mieliśmy do czynienia z całymi nutami. Możesz jednak dowolnie określać rodzaj nuty jaką chcesz uzyskać. Na przykład rozkaz:

100 PLAY "HA QB"

zagra półnutę A oraz ćwierćnutę B. Poniższa lista pokazuje, jakie liczby odpowiadają poszczególnym rodzajom nut:

W = cała nuta  
H = półnutą  
Q = ćwierćnuta  
I = ósemka  
S = szesnastka

Kropka ./ = nuta z kropką /1 1/2 raza dłuższa od zwykłej długości./.

Jeśli użyjesz jednej z tych liter dla określenia wartości jakiejś nuty, wszystkie nuty następujące po niej, będą miały identyczną wartość dopóki jej nie zmienisz, wprowadzając inną literę.

Następujący rozkaz PLAY zagra pierwsze dwa takty melodii "Marysia miała owieczkę małą":

100 PLAY ".QE SD QC D E E HE"

A oto pierwsze cztery takty "Claire de Lune"

100 PLAY "Q C C C D H E D Q C E D D H C"

Wartości nadwane poszczególnym nutom są względne. Oznacza to, że gdy np. określisz jakąś nutę jako ćwierćnutę, nie jest to równoznaczne z określeniem bezwzględnego czasu trwania tej nuty. Aby tego dokonać, musisz użyć rozkazu TEMPO. Jeśli nadasz parametrowi tego rozkazu wartość 20, na przykład, to takt będzie trwał około jednej sekundy, a więc ćwierćnuta będzie trwała przez czwartą część tego okresu czasu, to jest 1/4 sekundy. Jeśli parametr TEMPO wynosił 40, takt będzie trwał pół sekundy, a więc ćwierćnuta będzie trwała jedynie 1/8 sekundy.

Oto przykład programu, który ustawia poziom natężenia na maksimum, przeddefiniowuje obwiednię fortepianu, ustala tempo na 40 i odtwarza nuty jako ćwierćnuty, nadając im czas podtrzymania równy 1/8 sekundy.

100 VOL 15

110 ENVELOPE 0, 2, 10, 10, 0, 2, 2047

120 TEMPO 40

130 PLAY "Q C D E F G A B"

### Kontrola syntezy

Można określać również inne cechy dźwięku tworzonego przez rozkaz PLAY. Cechy te nazywa się łącznie kontrolą syntezy. Za pomocą PLAY możesz zdecydować, którego głosu użyjesz, w której oktawie będziesz grać, z której obwiedni, jaką częstotliwość nadać dźwiękowi oraz czy włączyć lub wyłączyć filtr. Własności te są określane pojedynczymi literami i numerami. Ponieważ żadna z tych liter nie pokrywa się z literami skali muzycznej ani literami oznaczającymi długość nut, BASIC nie myli ich ze sobą.

Tabela 7-4 pokazuje zakres wartości, jakie można nadać każdej z tych cech.

Tabela 7-4 - Oznaczenie literowe rozkazu PLAY

| Symbol | Cecha     | Zakres                        | Wartość domyślna |
|--------|-----------|-------------------------------|------------------|
| V      | Głos      | 1 - 3                         | 1                |
| O      | Oktawa    | 0 - 6                         | 4                |
| T      | Obwiednia | 0 - 9                         | 0                |
| U      | Głośność  | 0 - 15                        | 9                |
| X      | Filtr     | 0 = wyłączony<br>1 = włączony | 0                |

Tak więc kod kontrolny V2 włączy drugi głos, 06 wywoła szóstą oktawę, T7 określi siódmą obwiednię, U4 nada natężeniu wartość 4, X1 włączy filtr. W rozkazie PLAY możesz użyć niektórych lub wszystkich kodów jednocześnie. Jeśli jednak chcesz użyć więcej niż jednego kodu, powinieneś je umieścić w takiej samej kolejności, w jakiej zostały przedstawione w tabeli. Układ SID będzie je wtedy szybciej przetwarzał.

Poniższy program wybierze pierwszy głos, piątą obwiednię, nastawi głośność /natężenie/ na maksimum /15/ oraz zagra wszystkie nuty gamy jako ćwierćnuty:

100 PLAY "V1 T5 U15 Q C D E F G A B C"

#### Zwielokrotnione głosy

Aby grać kilkoma głosami, musisz najpierw określić, jakie nuty ma grać pierwszy głos, następnie, za pośrednictwem kodu kontrolnego V przełączasz się na drugi głos, dla którego również określasz rodzaj nut. Dzięki sprawności BASIC'a nie ma zbyt dużego opóźnienia między pierwszym a drugim głosem. Jeśli jednak chcesz zsynchronizować głosy, powinieneś wprowadzać dłuższe nuty przed sekwencjami krótszych. Jeśli jeden głos będzie grał dwie ćwierćnuty w tym samym czasie, w którym drugi będzie grał jedną półnutę, powinieneś zacząć od półnuty.

Podstawową jednostką czasu w muzyce jest takt. Powinieneś o tym pamiętać przy komponowaniu muzyki, złożonej z wielu części. Rozpisuj kody dla poszczególnych taktów oddzielnie, dając każdemu głosowi po kolei jeden kod. Na przykład w linii

100 PLAY "U1 04 W C V2 Q 06 C 05 6 E C"

grę rozpocznie pierwszy głos pełną nutą C, a następnie drugi głos zacznie grać cztery ćwierćnuty - C, G, E oraz C. Nuta gra-

na przez pierwszy głos będzie w czwartej oktawie. Pierwsza nuta C grana przez drugi głos jest w szóstej oktawie, a pozostałe nuty - w piątej.

Z tą podbudową teoretyczną możesz śmiało rozpocząć tworzenie muzyki na Commodore 128.