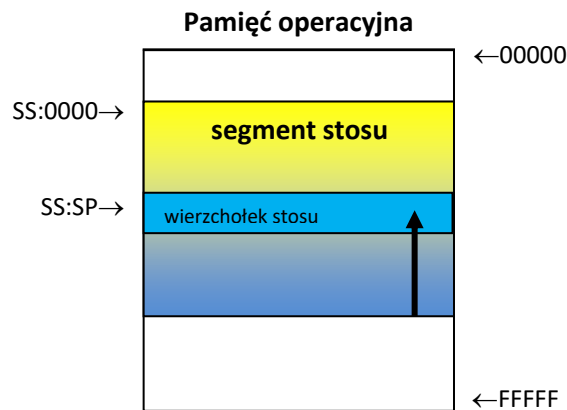


Operacje na stosie

Stos jest obszarem pamięci o dostępie LIFO (Last Input – First Output). Adresowany jest niejawnie przez rejestr segmentowy **SS** oraz wskaźnik wierzchołka stosu **SP**. Używany jest do przechowywania stanu procesora podczas wykonywania procedury i obsługi przerwań oraz do przekazywania parametrów do procedury.



Na stosie można złożyć dane oraz z niego pobrać.

PUSH *src* - instrukcja składowania na stosie działa następująco

- odejmuje **2** od zawartości rejestru **SP**
- zapisuje zawartość ***src*** na stosie pod adresem **SS:SP**.

POP *dst* - instrukcja pobierania ze stosu działa następująco

- pobiera ze stosu wartość spod adresu **SS:SP** i zapisuje do ***dst***.
- dodaje **2** do rejestru **SP**.

src oraz **dst** mogą określać rejestr 16-bitowy lub komórkę pamięci (na stosie umieszczane są wartości 16-bitowe).

Dla programów typu program.com rejestr segmentowy stosu **SS** ustawiany jest na taką samą wartość jak rejestr segmentu kodu **CS** zaś zawartość **SP** przyjmuje wartość **0FFFFh**. Pod adresem **SS: 0FFFFh** przechowywany jest **adres powrotu z programu** po instrukcji **ret**.

Można obserwować zawartość stosu oraz zachowanie wskaźnika stosu naciskając przycisk **[Stack]** w oknie emulatora. Pozycja wierzchołka stosu wskazywana jest przez symbol **<**.

Przykłady

ORG 100h

MOV AX, 1234h

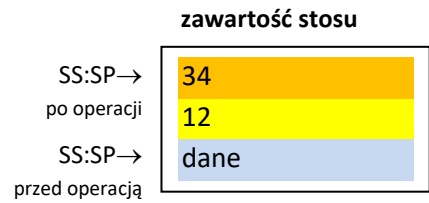
PUSH AX ; zapisanie zawartości AX na stosie

MOV AX, 5678h ; zmiana zawartości AX

POP AX ; odtworzenie oryginalnej zawartości AX

RET

END



ORG 100h

MOV AX, 1234h ; zapisanie wartości 1234h do AX.

MOV BX, 5678h ; zapisanie wartości 5678h do BX

PUSH AX ; zapisanie zawartości AX na stosie

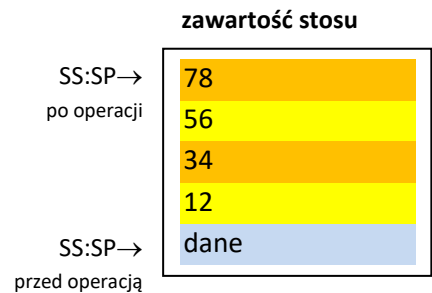
PUSH BX ; zapisanie zawartości BX na stosie

POP AX ; zapisanie do AX oryginalnej zawartości BX

POP BX ; zapisanie do BX oryginalnej zawartości AX – zamiana miejscami

RET

END



Procedury

Procedura jest fragmentem kodu wywoływanego z programu w celu realizacji określonego zadania. Zastosowanie procedur sprawia, że program uzyskuje określoną strukturę i jest łatwiejszy do zrozumienia. Wykorzystanie procedur umożliwia powtórne użycie kodu (wielokrotne wykorzystanie procedur w różnych programach). Użycie procedur zmniejsza wielkość pamięci zajmowanej przez program.

Po zakończeniu procedury program z reguły wraca do miejsca wywołania (następnej instrukcji). W tym celu przechowuje się na stosie adres powrotu (ślad).

Składnia procedury jest następująca:

```
name PROC  
  
    ; kod procedury ...  
  
RET  
name ENDP
```

name – nazwa procedury.

Instrukcja **RET** kończy wykonanie procedury i nakazuje powrót do programu wywołującego.

PROC oraz **ENDP** są dyrektywami kompilatora i nie generują żadnego kodu. Kompilator po prostu zapamiętuje adres procedury.

Instrukcja **CALL** służy do wywołania procedury.

CALL name

Oto przykład zdefiniowania i użycia procedury:

```
ORG 100h  
MOV AX, 1  
  
CALL m1  
  
MOV AX, 2  
  
CALL m1  
  
RET          ; powrót do system operacyjnego.  
  
m1 PROC  
    ADD AX, 5  
    RET      ; powrót do programu wywołującego.  
m1 ENDP  
END
```

Istnieją różne **sposoby przekazywania parametrów i wyniku** działania pomiędzy programem wywołującym a funkcją. Najprostszym jest **przekazywanie przez rejestry**.

Przykład:

W następującym przykładzie procedura otrzymuje dwa parametry przez rejestry **AL** oraz **BL**, mnoży je i wynik zwraca przez rejestr **AX**:

```
ORG 100h
MOV AH, 0
MOV AL, 1
MOV BL, 2

CALL m2
CALL m2
CALL m2
CALL m2

RET          ; powrót do system operacyjnego.

m2 PROC
    MUL BL    ; AX = AL * BL.
    RET      ; powrót do program wywołującego.
m2 ENDP

END
```

W powyższym przykładzie obliczona jest wartość **2** do potęgi **4**.

Przykład:

W następnym przykładzie używana jest procedura do wyświetlenia komunikatu *Hello World*:

ORG 100h

LEA SI, komunikat ; pobranie adresu zmiennej komunikat do SI.

CALL drukuj

RET ; powrót do systemu operacyjnego.

; =====

; procedura wyświetla napis zakończony zerem (znacznik końca)

; adres łańcucha znaków przekazywany jest przez rejestr SI:

drukuj PROC

next_char:

CMP b.[SI], 0 ; sprawdzenie czy osiągnięto znacznik końca (0)

JE stop ; jeśli tak to koniec procedury

MOV AL, [SI] ; pobranie następnego kodu ASCII .

MOV AH, 0Eh ; **funkcja systemowa**

INT 10h ; wyświetla znak zapisany w AL.

INC SI

JMP next_char ; przejście do następnego znaku.

stop:

RET ; powrót do program wywołującego.

drukuj ENDP

; =====

komunikat DB 'Hello World!', 0 ; tablica znaków zakończona znacznikiem końca (0).

END

Prefiks "**b.**" przed [SI] oznacza, że porównujemy bajty; porównując wartości 16-bitowe używamy prefiksu "**w.**" . Jeśli porównywana jest zawartość rejestru – prefiksy te nie są potrzebne, gdyż kompilator zna rozmiary rejestrów.

Przykład definicji i użycia procedury:

outhex PROC

;specyfikacja funkcji

; wejście: BH zawiera bajt

; wyjście: pisz szesnastkową wartość bajtu

;zapamiętanie stanu procesora (rejestrów wykorzystywanych przez funkcję)

push AX

push BX

push CX

push DX

;operacje realizowane przez funkcję

mov DL, BH

mov CL, 4

shr DL, CL ; analiza 4 bitów mniej znaczących

mov CX, 2 ; pętla 2 razy

AGAIN:

cmp DL, 9

ja BIG

add DL, 30h

jmp PRINT

BIG: add DL, 37h

PRINT: mov AH, 2

int 21h ; wyświetlenie znaku

mov DL, BH

and DL, 0Fh ; wyzeruj starsze 4 bity

loop AGAIN

;odtworzenie stanu procesora (w odwrotnej kolejności!!!)

pop DX

pop CX

pop BX

pop AX

;powrót do program wywołującego

ret

outhex ENDP

Użycie procedury

mov BH, *zmienna* ; podaj argument dla OUTHEX

call outhex

Przykład procedury komunikującej się z programem głównym za pośrednictwem stosu:

```
N EQU 4
org 100h
; sumowanie kilku liczb przekazywanych przez stos
push a
push b
push c
push d
push N          ; liczba sumowanych elementów
call suma

add SP,10        ; przesunięcie wskaźnika stosu o obszar zajęty przez parametry

ret

suma PROC
    mov BP, SP    ; ustawienie rejestru bazowego dla dostępu do parametrów
    push CX
    mov CX, [BP+2]
    mov AX, 0
    mov SI, 4
    ets: add AX, [BP+SI]
        add SI, 2
    loop ets
    pop CX
    ret           ; wynik zwracany przez AX
suma ENDP

a dw 3
b dw 2
c dw 7
d dw 4
end
```

Narysować obraz stosu w momencie wywołania procedury i prześledzić operacje na stosie podczas pracy programu.

Wykorzystanie stosu do komunikacji z procedurą.

;Program procSuma.asm

org 100h

mov AH, 09H ;wyświetlenie komunikatu „Podaj dwie liczby jednocyfrowe”

lea DX, komunikat

int 21H ;dos interrupt

;wczytywanie wartości z klawiatury

mov AH, 0AH ;buforowane wejście z klawiatury

lea DX, listaPar

int 21H

;pobierz dwie liczby całkowite i przekaż na stos

mov AL, poleOper ;pobierz 1 znak z poleOper

mov AH, 00H

push AX ; 1 argument na stos

mov BL, poleOper +1 ; pobierz 2 znak z poleOper

mov BH, 00H

push BX ; 2 argument na stos

call oblicz ;odłożenie adresu powrotu na stosie i przejście do procedury

;miejsce powrotu po wykonaniu procedury

;zapisanie sumy do zmiennej wynik

mov wynik+8, AH

mov wynik+9, AL

;wyświetlenie wyniku

mov AH, 09H ;wyświetlenie napisu

lea DX, wynik

int 21H

ret ;---- Wykorzystanie stosu w programie →→→→

oblicz PROC

push BP ;wykorzystany będzie rejestr BP

mov BP, SP ;BP wskazuje na wierzchołek stosu

mov AX, [BP+4] ;pobranie pierwszego operandu

add AX, [BP+6] ;dodanie drugiego argumentu do AX

aaa ;korekcja dziesiętna dla kodów ASCII

or AX, 3030H ;zamiana na kod ASCII wynik zwracany przez AX

pop BP ;odtworzenie BP

ret 4 ;pobranie IP z dodaniem 4 do SP tak by wskazywał początkową lokalizację

oblicz ENDP

;

listaPar LABEL BYTE

maxNapis DB 5

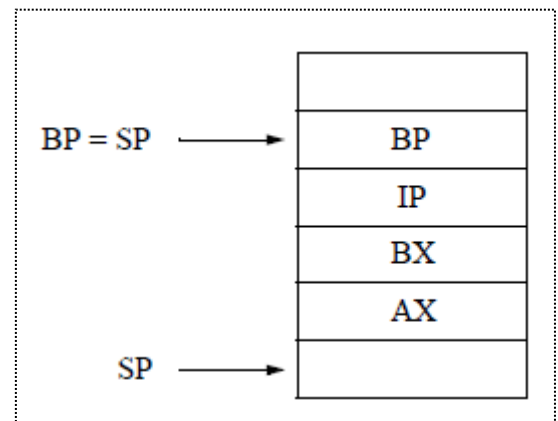
liczbaZn DB ?

poleOper DB 5 DUP(?)

komunikat DB 0DH, 0AH, "Podaj dwie liczby jednocyfrowe: \$"

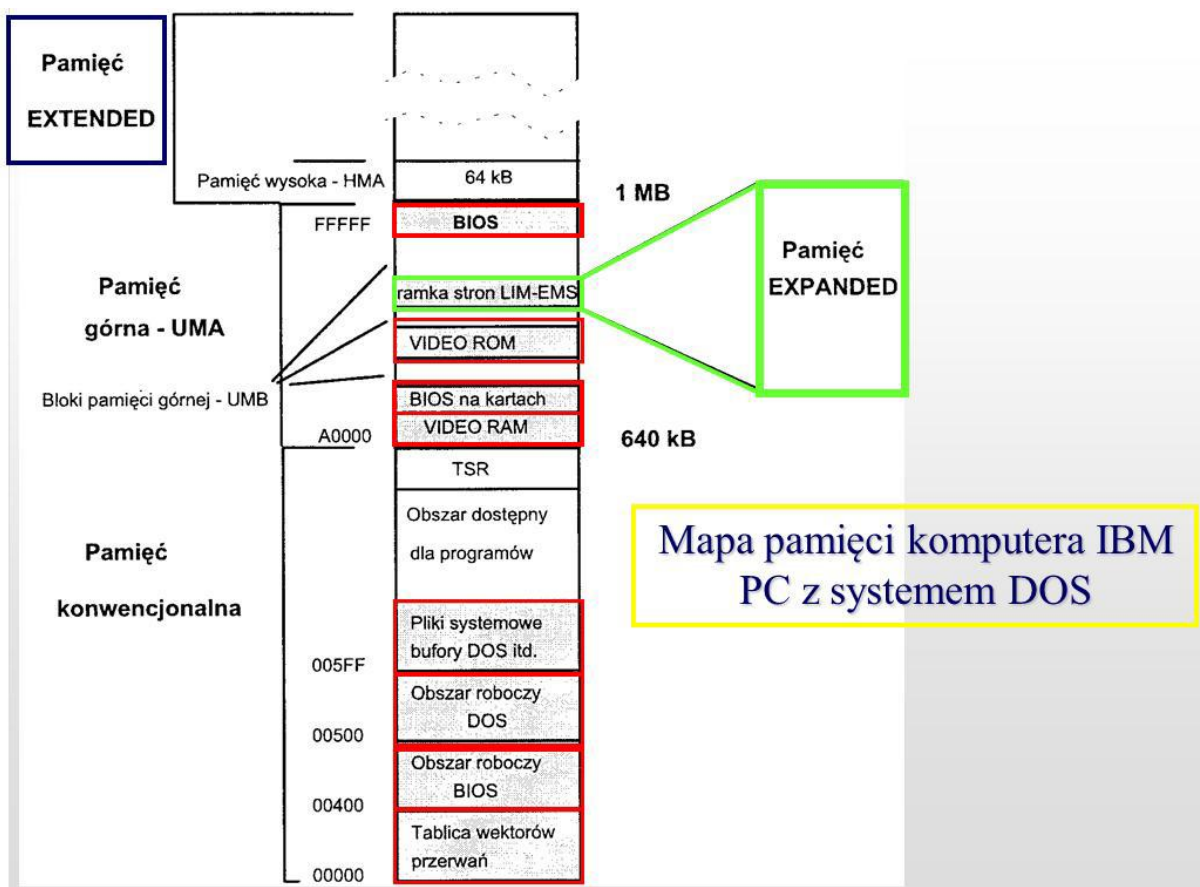
wynik DB 0DH, 0AH, "Suma = \$"

END



Przeznaczenie i działanie BIOSu

BIOS (ang. *Basic Input/Output System* – podstawowy system wejścia-wyjścia) – zapisany w pamięci stałej, zestaw podstawowych procedur pośredniczących pomiędzy systemem operacyjnym, a sprzętem. Posiada on własną pamięć konfiguracji, w której znajdują się informacje dotyczące daty, czasu oraz danych na temat wszystkich urządzeń zainstalowanych w komputerze. Jest to program zapisany w pamięci ROM płyty głównej oraz innych kart rozszerzeń takich jak np. karta graficzna. BIOS składa się przede wszystkim z wykonywalnego kodu zapisanego w kości pamięci umieszczonej na płycie głównej. Program BIOSu wykonuje się przy każdym uruchomieniu lub zrestartowaniu komputera i dlatego pamięć ta nie może się zerować po odcięciu zasilania. Z tego powodu nośnikiem, na którym zapisujemy ten kod, są kości pamięci typu FLASH ROM (aktualnie EEPROM - Electrically Erasable Programmable Read-Only Memory), a kiedyś były to po prostu kości ROM. Oprócz kodu, częścią BIOSu jest również CMOS (Complementary Metal Oxide Semiconductor). CMOS jest niewielkim fragmentem pamięci, w którym są zapisane ustawienia BIOSu i programu SETUP. Kiedy wyłączone jest zasilanie, pamięć ta jest podtrzymywana za pomocą baterijki umieszczonej na płycie głównej. Podstawowym zadaniem BIOSu jest załadowanie systemu operacyjnego „wyższego rzędu“ (np. DOS, Windows, Linux) i przekazanie mu informacji o zainstalowanym w komputerze sprzęcie.



BIOS wykonuje szereg różnych zadań.

- **przeprowadzanie po włączeniu komputera testów systemu** (tzw. test POST - Power On Self Test) - sprawdzenie poprawności funkcjonowania podstawowych urządzeń systemu i danych zapisanych w CMOSie.

Po restarcie systemu, niezależnie od jego przyczyny, procesor rozpoczyna wykonywanie instrukcji od adresu pamięci FFFF:0000h. Pod adresem tym znajduje się kod instrukcji skoku do procedury POST.

Jego zadaniem jest przetestowanie oraz inicjalizacja podstawowych układów płyty głównej.

Kolejno testowane są:

- procesor
- zawartość pamięci ROM (poprawność odczytu)
- pamięć RAM
- testowanie i inicjowanie układów programowalnych płyty głównej
- inicjowane są struktury systemowe – tablica wektorów przerwań
- testowane są standardowe urządzenia peryferyjne – klawiatura, stacje dysków, karta grafiki
- ostatnim etapem jest sprawdzenie obecności BIOS-u na kartach

- **załadowanie BIOSów urządzeń** - zanim BIOS wykona swój kod, uruchamia znalezione BIOSy innych urządzeń zainstalowanych w komputerze.

- **zabezpieczenie komputera** - większość BIOSów ma możliwość zabezpieczenia dostępu do obszaru MBR (Master Boot Record) dysków twardych, żeby nie dostał tam się wirus. Można też zabezpieczyć dostęp do komputera za pomocą hasła, o które system pyta się po uruchomieniu komputera.

- **programowanie układów programowalnych** - zainicjowanie niektórych (sprzętowych) komponentów systemu i umożliwienie im dalszej pracy. Takim komponentem jest np. sterownik DMA.

- **przydział zasobów systemowych** (funkcja dostępna na płytach z technologią PCI (Peripheral Component Interconnect) i PNP (Plug&Play)) - BIOS przydziela zasoby, takie, jak adresy I/O (Input/Output), numery przerwań IRQ (Interrupt Request Line) i kanały DMA (Direct Memory Access), zainstalowanym urządzeniom.

- **umożliwienie użytkownikowi konfiguracji sprzętu** - użytkownik komputera ma możliwość podczas uruchamiania komputera wejść do programu SETUP i zmienić niektóre ustawienia sprzętu.

- **załadowanie systemu operacyjnego** - tzw. IPL (Initial Program Load).

- **zarządzanie energią** (power management) - po pewnym okresie nieaktywności, BIOS wyłącza niektóre komponenty systemu w celu minimalizacji zużycia energii.

- **pośredniczenie pomiędzy zainstalowanym sprzętem a systemem operacyjnym** - BIOS jest platformą stanowiącą interfejs do zainstalowanego sprzętu, umożliwiającą komunikację pomiędzy programami a sprzętem (głównie dzięki mechanizmowi przerwań).
- **inicjalizacja pracy systemu** (instrukcje pobierane podczas startu pracy procesora, programowanie układów programowalnych, takich jak sterowniki przerwań czy DMA, wpisanie wartości początkowych do struktur systemowych w pamięci, na przykład inicjacja tablicy wektorów przerwań)
- **zapewnienie w postaci programów obsługi przerwań (programowych bądź sprzętowych), procedur obsługi (sterowników) podstawowych, standardowych urządzeń systemu.**
- **niwelacja z punktu widzenia systemu operacyjnego, różnic konstrukcyjnych płyt głównych pochodzących od różnych producentów** (HAL – warstwa abstrakcji sprzętowej).

Warstwa abstrakcji sprzętowej (ang. Hardware Abstraction Layer – HAL) – sterownik urządzenia dla płyty głównej. Stanowi ogniwo pośredniczące między sprzętem a jądrem systemu operacyjnego. Odseparowuje konkretną architekturę systemu komputerowego od oprogramowania użytkowego. Dla programisty jest sposobem komunikacji ze sprzętem przez udostępniane funkcje biblioteczne i sterowniki.

BIOS – podstawowe procedury wejścia / wyjścia

Dotyczą działania określonych procedur przerwań, których przyczyną jest wykonanie określonego rozkazu. Procedury umieszczone są w BIOS-ie (pamięć ROM) jako podstawowe procedury wejścia / wyjścia. Dotyczą tylko wybranych przerwań od 0 do 1Fh.

Przykładowe przerwania BIOS – u:

- INT10h – obsługa ekranu
- INT13h – obsługa dysków twardych
- INT17h – obsługa drukarki

Część urządzeń może nie być obsługiwanych przez BIOS. Są to urządzenia które najczęściej posiadają własne sterowniki lub własny BIOS.

Przykładowe funkcje systemowe (BIOS, DOS) do obsługi urządzeń wejścia-wyjścia

Wyświetlanie znaku na ekranie

```
mov DL, 'A'
mov AH, 2
int 21h
```

Czytanie znaku z klawiatury do AL.

```
mov AH, 1          // lub mov AH, 7 gdy czytamy bez echa
int 21h
```

Czytanie napisu z klawiatury

```
mov DX, offset bufor
mov AH, 0Ah
int 21h
```

...

```
bufor DB 10, ?, 10 dup ('-')
```

gdzie 10 oznacza wielkość bufora na napis, zaś w miejscu ? będzie umieszczona liczba znaków przeczytanych. Aby wyświetlić odczytany napis używamy funkcji 9 i adresu DS:DX+2

Wyświetlenie napisu na ekranie

(napis w tablicy msg musi być zakończony znakiem \$)

```
mov DX, offset msg      // lub lea DX, msg
mov AH, 9
int 21h
.....
msg DB "Witaj $"
```

Przykład wczytania napisu i wyświetlenie go na ekranie

```
org 100h
    mov dx, offset buffer
    mov ah, 0Ah
    int 21h          ; wczytanie napisu do bufora

    xor bx, bx       ; zerowanie rejestru bx
    mov bl, buffer[1]
    mov buffer[bx+2], '$' ; dodanie znaku '$' na końcu napisu
    mov dx, offset buffer + 2
    mov ah, 9
    int 21h          ; wyświetlenie napisu na ekranie
ret
    buffer db 10, ?, 10 dup('-')
```

Oczekiwanie na wciśnięcie klawisza – pobranie znaku bez echa

mov AH, 0

int 16h

Po wykonaniu

AH = kod przeglądania BIOS

AL = kod znaku ASCII

Sprawdzenie czy w buforze klawiatury jest znak (bez pobierania znaku)

mov AH, 1

int 16h

Po wykonaniu

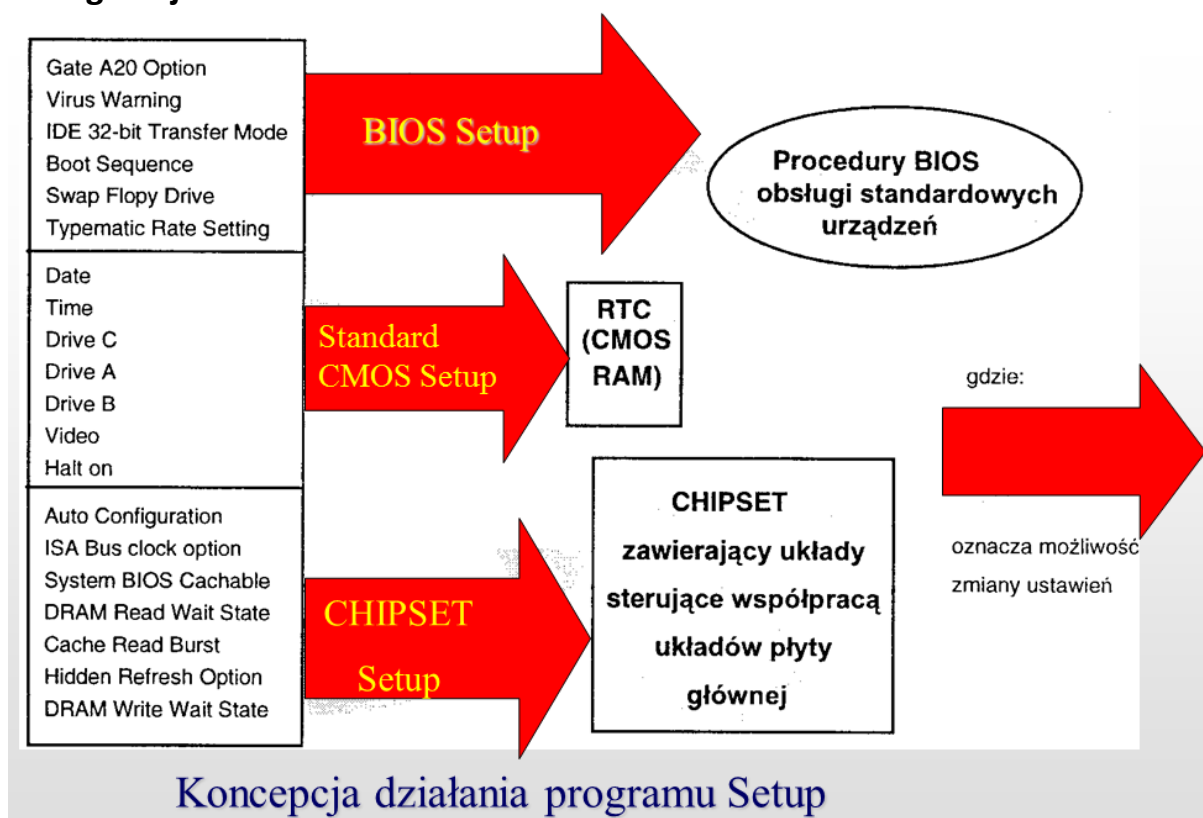
ZF = 1 jeśli klawisz nie był wciśnięty

ZF = 0 jeśli klawisz nie wciśnięty

AH = kod przeglądania BIOS

AL = kod znaku ASCII

Konfiguracja BIOS-u



Aby dostać się do programu konfiguracyjnego, trzeba zaraz po włączeniu peceta nacisnąć określony klawisz, zazwyczaj [Del], [F1], [F2], [F10] lub kombinacje [Ctrl Alt Esc].

Na ekranie startowym powinien się pojawić stosowny komunikat. Parametry BIOS-u pozwalają zmieniać różne ustawienia, począwszy od najprostszych, takich jak tryb transmisji portu szeregowego, skończywszy na bardzo interesujących.

Przy zmianach ustawień BIOS-u zachować należy ostrożność i rozważę.

Np. ustawienie zbyt wysokiego napięcia zasilającego procesor, a także zmiany w geometrii twardego dysku mogą mieć opłakane skutki.

Jednocześnie w niejednym komputerze drzemią niespodziewane zapasy wydajności, które można wykorzystać tylko za pomocą BIOS-u. W ten sposób zyskasz większą wydajność, nie inwestując ani grosza.

BIOS – Setup – podstawowe ustawienia

- Standard CMOS Setup – dotyczy parametrów zapisanych w pamięci konfiguracyjnej zegara czasu rzeczywistego
- BIOS Features Setup – zmienia własności procedur BIOS-u wywoływanych przerwami sprzętowymi bądź programowymi
- CHIPSET Features Setup – zmienia sposób pracy układów zawartych w Chipsecie, sterujących pracą układów płyty głównej
- PCI Configuration Setup – ustawia opcje dotyczące sposobu pracy magistrali PCI
- Power Management Setup – zarządza oszczędzaniem mocy

Standard CMOS Setup Menu

Zawiera proste parametry konfiguracyjne BIOS-u:

- Date - ustawiamy datę (miesiąc, dzień, rok).
- Time - ustawiamy aktualny czas (godzina, minuta, sekunda).
- mamy też możliwość ręcznych ustawień parametrów dysków twardych podłączonych do komputera .

BIOS Features Setup Menu

Mamy tu możliwość zwiększenia wydajności komputera oraz dokonania zmian niektórych opcji według własnego upodobania (np. kolejności przeszukiwania napędów w poszukiwaniu systemu operacyjnego). Przedstawię tylko standardowe funkcje występujące w większości BIOS-ów.

- IDE HDD Block Mode - włączenie transferu blokowego powoduje, że komputer wczytuje z dysku do pamięci bloki po kilka sektorów naraz, co zwiększa wydajność twardego dysku. Trzeba jednak uważać, gdyż starsze modele dysków nie radzą sobie z tą opcją.
- Video BIOS Shadow - włączenie tej opcji powoduje skopiowanie BIOS-u karty graficznej z pamięci typu ROM do szybkiej pamięci RAM, co pozwala na szybszy dostęp i ma wpływ na działanie programów i systemów graficznych .
- CPU Level 1 Cache - pozwala włączyć pamięć cache pierwszego poziomu.
- CPU Level 2 Cache - pozwala włączyć pamięć cache drugiego poziomu.
- Quick Power On Self Test - jeżeli chcemy, by proces uruchamiania komputera przebiegał nieco szybciej, warto uaktywnić tą opcję.
- Boot Up Floppy Seek - jeżeli nie mamy stacji dyskietek w komputerze, należy tę opcję koniecznie wyłączyć, w przeciwnym wypadku otrzymamy komunikat o błędzie.
- Boot Up NumLock Status - jeżeli włączymy tę opcję (On) komputer uruchomi się z klawiaturą numeryczną działającą w trybie numerycznym, wyłączenie tej opcji (Off) spowoduje, że standardowo klawiatura numeryczna będzie ustawiona w trybie sterowania kursorami.
- Security Option - jeśli wybierzemy "System", komputer będzie prosił o podanie hasła podczas każdego uruchamiania. Wybranie opcji "Setup" będzie wymuszało na użytkowniku podanie hasła tylko w momencie dostępu do ustawień BIOS-u, opcja ta będzie jednak uaktywniona tylko wtedy, gdy wprowadzimy hasło.
- Floppy Disk Access Control - za pomocą tego parametru ustawiamy atrybuty stacji dyskietek, możemy zezwolić użytkownikom na zapis i odczyt (R/W) lub tylko odczyt (Read Only).

Advanced Chipset Features , Chipset Features Setup

To menu zawiera głównie opcje dotyczące parametrów buforów pamięci.

Advanced Chipset Features

- DRAM CAS Latency - ustawiamy czas, po jakim odczytana z pamięci informacja jest dostępna (im mniejsza wartość, tym mamy wydajniejszy system).
- DRAM Idle Timer - pozwala ustawić czas zamknięcia otwartej strony pamięci SDRAM, mniejsza wartość - wydajniejszy system.
- DRAM RAS Precharge Time - podajemy, co ile cykli ma być odświeżana pamięć SDRAM; im mniej, tym lepiej.
- System BIOS Cacheable - powoduje buforowanie BIOS-u i co za tym idzie, zwiększenie prędkości systemu, który odczytuje informacje BIOS-u z bufora (szybciej), a nie bezpośrednio z pamięci FlashROM (wolniej).

Chipset Features Setup

- Parallel Port Mode - pozwala na wybór trybu pracy (ECP lub EPP lub oba naraz) portu równoległego w naszym komputerze. Najlepiej wybrać oba naraz, dzięki temu uzyskamy pewność, że będą poprawnie funkcjonować różnego typu urządzenia podłączone do portu równoległego.
- IDE Ultra DMA Mode - jeżeli dysk twardy zainstalowany w naszym komputerze działa w trybie UDMA, należy uaktywnić tę opcję.

PNP and PCI Setup

PNP/ PCI Configuration Setup

Konfiguracja slotów do kart rozszerzeń.

- PNP OS Installed - pozwala określić, czy używamy systemu operacyjnego wspomagającego obsługę urządzeń o charakterze Plug and Play.
- Slot IRQ - możemy dla poszczególnych slotów przydzielić konkretne przerwania lub zlecić to systemowi (Auto).
- IRQ Used By ISA - pozwala określić, czy dane przerwanie jest wykorzystywane przez kartę rozszerzeń typu ISA.
- DMA Used By ISA - określamy, czy konkretny kanał DMA jest używany przez karty ISA.
- VGA BIOS Sequence - jeżeli nasz komputer ma dwie karty graficzne (jedną opartą na złączu PCI, drugą AGP), możemy za pomocą tej opcji określić priorytety dla tych kart.

Integrated Peripherals

Opcje udostępniania przez tę kategorię programu Setup umożliwiają wpływ na pracę wbudowanych w płytę główną interfejsów, dzięki którym możemy rozbudować nasz komputer o kolejne urządzenia peryferyjne.

OnChip IDE Channel 0/1 – ta opcja uaktywnia poszczególne kanały (pierwszy – primary oraz drugi secondary) interfejsu IDE wbudowanego w płytę główną.

IDE Prefetch Mode – „Pobrania wstępne”- tak nazywa się opcja, którą tutaj możemy włączyć. Dzięki temu kontroler uzyska szybszy dostęp do dysków.

USB Controller – domyślne ustawienie to „Enable”. Włącza i wyłącza kontroler USB, znajdujący się na płycie głównej. Jeżeli nie używamy żadnego urządzenia typu USB możemy włączyć ustawienie „Disabled”.

USB Mouse Support – opcja ta pozwala na określenie sposobu kontroli nad podłączoną muszką USB. Kontrola ta może odbywać się przez BIOS bądź system operacyjny.

IDE HDD Block Mode – Tutaj uaktywniamy transfer blokowy z i do twardego dysku. Polega to na zdolności dysku do zapisu i odczytu wielu sektorów jednocześnie.

Onboard Legacy Audio – jest to funkcja, która uaktywnia kodek audio, który jest wbudowany w płytę główną. Jest to kodek karty dźwiękowej zintegrowanej z płytą w trybie zgodności z SoundBlater.

Power Menagment Setup

Opcje znajdujące się w tym menu umożliwiają redukcję poboru mocy przez nasz komputer.

- Power Menagment – mamy tu możliwość wybrania kilku z góry określonych, typów ustawień (Max Saving, Min Saving), wyłączenia jakichkolwiek funkcji oszczędzania energii (Disable) lub zdefiniowania ich samemu (User Define).

- Suspend Mode – określamy czas, po którym komputer przechodzi w stan czuwania (Suspend) lub możemy tę funkcję wyłączyć (Disable).

- Thermal Monitor - pokazuje aktualną temperaturę otoczenia procesora

- Voltage Monitor - pokazuje napięcia podawane na procesor

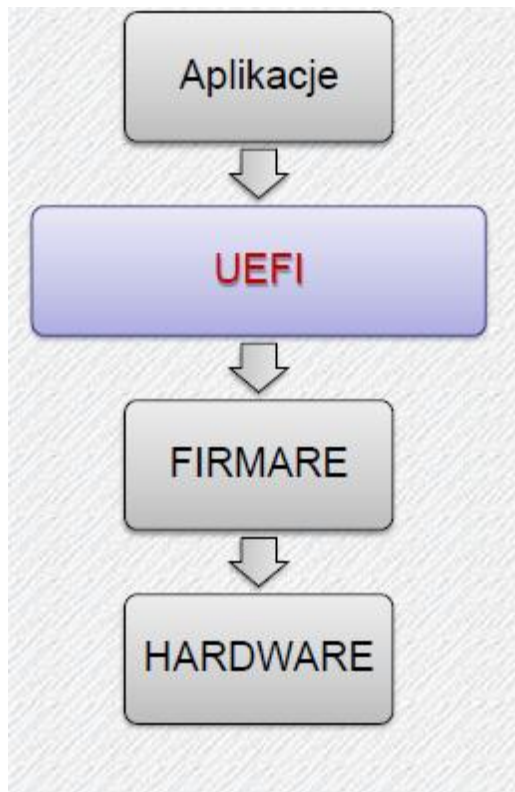
Load BIOS Default

Dzięki tym opcjom możemy powrócić do standardowych ustawień Setupu BIOS-u. Jest to jednak wyjście ostateczne, bo jak ostrzegają sami producenci płyt głównych - ustawienia te nie są optymalne i dalekie od ustawień dla konkretnych konfiguracji.

- Supervisor Password - pozwala ustawić hasło zabezpieczające zarówno system, jak i dostęp do Setupu BIOS-u.

- **User Password** - pozwala ustawić hasło zabezpieczające system. Uwaga! Jeżeli za pomocą jednej z dwóch powyższych opcji ustawimy hasło dostępu do systemu i następnie je zapomnimy, jedynym wyjściem będzie skasowanie wszystkich ustawień BIOS-u za pomocą zwarcia odpowiednich wyprowadzeń na płycie głównej (proces ten jest opisany w każdej instrukcji do płyty głównej).
- **IDE Autodetection** - pozwala przeprowadzić automatyczne wykrywanie dysków twardych, w jakie wyposażony jest komputer.
- **Save & Exit Setup** - wybierając tę opcję, decydujemy się na zapisanie wszystkich dokonanych przez nas zmian w pamięci CMOS i wyjście z Setupu BIOS-u.
- **Exit Without Saving** - opuszczamy program konfiguracyjny BIOS-u z pominięciem dokonanych przez nas zmian.

BIOS/UEFI



Adres	Nazwa układu
000 – 01F	Sterownik DMA nr 1
020 – 03F	Sterownik przerwań nr 1
040 – 05F	Generatory programowalne
060 – 06F	Sterownik klawiatury
070 – 07F	Zegar czasu rzeczywistego
080 – 08F	Rejestr stron DMA
0A0 – 0BF	Sterownik przerwań nr 2
0C0 – 0DF	Sterownik DMA nr 2
0F1	Reset koprocatora arytmetycznego
0F8 – 0FF	Koprocator arytmetyczny
1F0 – 1F8	Sterownik dysków twardych
200 – 207	Game-port
278 – 27F	Port równoległy nr 2
2F8 – 2FF	Port szeregowy nr 2
378 – 37F	Port równoległy nr 1
3B4 – 3BA	Sterownik VGA (mono)
3C0 – 3DA	Karta VGA
3F0 – 3F7	Sterownik dysków elastycznych
3F8 – 3FF	Port szeregowy nr 1

Przestrzeń adresowa
układów wejścia / wyjścia