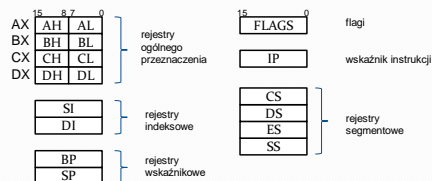


Architektura procesora

Procesor 8086 - rejestry



(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

2

Rejestry

- **AX (ang. Accumulator)** - jest wykorzystywany głównie do operacji arytmetycznych i logicznych.
- **BX (ang. Base Registers)** - rejestr bazowy, głównie wykorzystywany przy adresowaniu pamięci.
- **CX (ang. Counter Registers)** - rejestr często wykorzystywany jako licznik, np. przy instrukcji LOOP.
- **DX (ang. Data Register)** - rejestr danych, wykorzystywany przy operacjach mnożenia i dzielenia, a także do wysyłania i odbierania danych z portów.

(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

3

Rejestry c.d.

- **SI (ang. Source Index)** - rejestr indeksujący pamięć, wskazuje obszar z którego przesyłane są dane. W połączeniu z DS tworzy adres logiczny DS:SI
- **DI (ang. Destination Index)** - rejestr indeksujący pamięć, wskazuje obszar, do którego przesyłane są dane. W połączeniu z ES, tworzy adres logiczny ES:DI
- **BP (ang. Base Pointer)** - rejestr stosowany do adresowania pamięci.
- **SP (ang. Stack Pointer)** - wskaźnik stosu.

(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

4

Rejestry c.d.

- **IP (ang. Instruction Pointer)** - zawiera adres aktualnie wykonywanej instrukcji, może być modyfikowany przez rozkazy sterujące pracą programu.
- **FLAGS** - rejestr znaczników.

(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

5

Rejestry c.d. - segmentowe

- **CS (ang. Code Segment)** - rejestr informujący o segmencie aktualnie wykonywanego rozkazu. Razem z IP tworzy adres logiczny CS:IP kolejnej instrukcji.
- **DS (ang. Data Segment)** - rejestr informujący o segmencie z danymi.
- **ES (ang. Extra Segment)** - rejestr informujący o segmencie dodatkowym np. przy operacjach przesyłania łańcuchów.
- **SS (ang. Stack Segment)** - rejestr informujący o segmencie stosu.

(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

6

Adres w trybie rzeczywistym

powstaje w wyniku sumowania położenia segmentu i przesunięcia w nim.

(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

7

Architektura 8086

(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

8

IA32- rejestry

(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

9

Rejestr flag

Rejestr flag w architekturze Intel x86			
Bit	Skróć/wartość	Opis	KVP
0	CF	flaga przeniesienia (carry)	S
1	IF	zarezerwowany	S
2	PF	flaga parzystości (parity)	S
4	AF	flaga wywołania (adjust)	S
6	ZF	flaga zera (zero)	S
7	SF	flaga znaku (sign)	S
8	TF	flaga umożliwiająca krokowe wykonanie (trap)	X
9	IF	flaga zezwolenia na przerwanie (interrupt enable)	X
10	DF	flaga kierunku (direction)	C
11	OF	flaga przepełnienia (overflow)	S
12, 13	IOPL	poziom uprawnień we/wy (I/O privilege level, od x86)	X
14	NT	nested task flag (od x86)	X
16	RF	flaga wznowienia (resume, od x86)	X
17	VM	flaga trybu Virtual 8086 (od x86)	X
18	AC	alignment check (od x86SX)	X
19	VIF	Virtual interrupt flag (od Pentium)	X
20	VIP	Virtual interrupt pending (od Pentium)	X
21	ID	identification (od Pentium)	X
31, 30, 22-31	0	zarezerwowany	

S: Znacznik stanu
C: Znacznik kontrolny
X: Znacznik systemowy

(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

10

Koprocesor

stos rejestrów

(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

11

Rejestry MMX

Działają na nich instrukcje całkowitoliczbowe SIMD
Wykorzystują rejestry koprocesora

(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

12

Rejestry XMM

Działają na nich instrukcje zmiennoprzecinkowe SIMD

xmm0

xmm1

xmm2

xmm3

xmm4

xmm5

xmm6

xmm7

127

0

rejestr kontrolny

MXCSR

(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

13

EM64T- rejestry

RAX

RBX

RCX

RDX

63

32,31

16,15

8,7

0

EAX

16,15

8,7

0

AH

AL

BH

BL

CH

CL

DH

DL

rejestry
ogólnego
przeznaczenia

RSI

RDI

SI

DI

rejestry
indeksowe

RBP

RSP

BP

SP

rejestry
wskaźnikowe

R8

...

R15

R8W

R8D

(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

14

EM64T- rejestry

RFLAGS

63

31

EFLAGS

0

flagi

RIP

EIP

IP

wskaźnik instrukcji

CS

DS

ES

SS

FS

GS

rejestry
segmentowe

(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

15

EM64T- rejestry XMM

Działają na nich instrukcje zmiennoprzecinkowe SIMD

xmm0

xmm1

xmm2

xmm3

xmm4

xmm5

xmm6

xmm7

xmm8

xmm9

xmm10

xmm11

xmm12

xmm13

xmm14

xmm15

127

0

rejestr kontrolny

MXCSR

(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

16

AVX- Advanced Vector eXtensions

Rejestry ymm - działają na nich instrukcje zmiennoprzecinkowe SIMD - AVX

ymm0

ymm1

ymm2

ymm3

ymm4

ymm5

ymm6

ymm7

ymm8

ymm9

ymm10

ymm11

ymm12

ymm13

ymm14

ymm15

255

127

xmm0

0

(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

17

AVX512- Advanced Vector eXtensions

Rejestry zmm – tylko w wybranych procesorach

zmm0

zmm1

zmm2

zmm3

zmm4

zmm5

zmm6

zmm7

zmm8

zmm9

zmm10

zmm11

zmm12

zmm13

zmm14

zmm15

511

255

ymm0

127

xmm0

0

zmm31

(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

18

32-bits MOXSR Register

(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

2

(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

21

(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

2

4 bity = 1 cyfra BCD

(C) IISL d.KIK PCz 2019

Programowanie niskopoziomowe

22

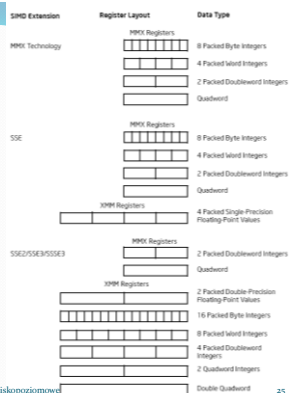
(C) IISL d.KIK PCz 2019

Programowanie niskopoziomowe

2

SIMD

Rejestry i typy danych



(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

25

Kilka instrukcji

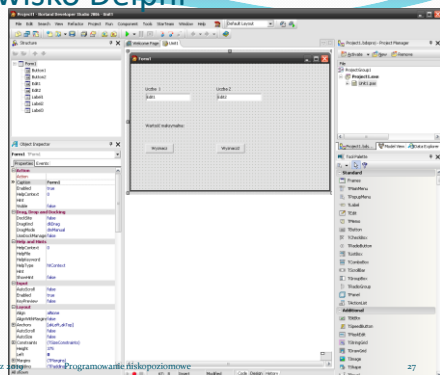
		add	eax,edx
mov	eax,edx	sub	rax,rbx
mov	eax,[ebx]	mul	ecx ;edx:eax=eax*ecx
mov	[rdx],rax	inc	ecx
mov	eax,zmienna	dec	rcx
push	bp	cmp	eax,ecx
pop	eax		
call	podprogram	jz	etykieta
ret		jnz	etykieta
		jc	etykieta
		jnc	etykieta

(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

20

Środowisko Delphi



(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

27

Użycie rejestrów

Rejestr	Windows 32	Windows 64
do użycia	EAX, ECX, EDX, ST(0)-ST(7), Ko-K7, xMM0-xMM7,	RAX, RCX, RDX, R8-R11, ST(0)-ST(7), Ko-K7, xMM0-xMM5, xMM16-xMM31
do zabezpieczenia	EBX, ESI, EDI, EBP	RBX, RSI, RDI, RBP, R12-R15, xMM6-xMM15
parametry funkcji	cdecl, stdcall, pascal, Gnu C: na stosie, fastcall Microsoft/Gnu : ecx, edx fastcall Borland eax, ecx, edx thical Microsoft ecx	RCX, RDX, R8, R9 lub xMM0-xMM3 reszta na stosie.
zwracające wartość funkcji	EAX, EDX, ST(0)	RAX, xMM0

$$\underline{\mathbf{x} = \mathbf{X}, \mathbf{Y} \text{ lub } \mathbf{Z}}$$

(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

28

Przykład

```
function TForm1.MyMax(x,y:integer):integer;
asm
    mov eax,x
    cmp eax,y
    jnc @@exit
    mov eax,y
@@exit:
    // mov result,eax
end;
```

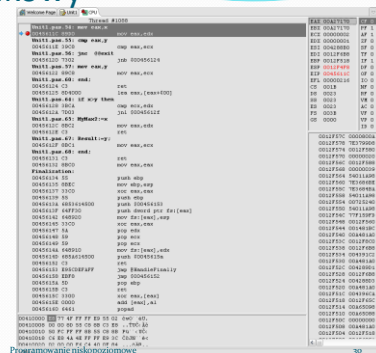
```
function TForm1.MyMax2(x,y:integer):integer;
begin
  if x>y then
    MyMax2:=x
  else
    Result:=y;
end;
```

(C) IISL d.KIK PCz 2019

Programowanie niskopoziomowe

20

Kod wynikowy



(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

