

## Procesory:

1. Dwa rdzenie pojawiły się w – **pentium 4**
2. Głównym konstruktorem 8086 był – **Stephen Morse**
3. Ile etapów przetwarzania rozkazów występuje w intel pentium III – **12**
4. Ile instrukcji na słowach może wykonywać jedna instrukcja mmx – **4**
5. Ile operacji na bajtach może wykonywać jedna instrukcja mmx – **8**
6. Ile rejestrów indeksowych występuje w EM64T w procesorze intel – **2**
7. Ile rejestrów ogólnego przeznaczenia dołożono w trybie EMT64T – **4 (tyle było)/bez zmian**
8. Ile rejestrów segmentowych występuje w trymie M64T procesorów intel – **6**
9. Ile rejestrów XMM posiadają w trybie EMT64T proc. Intel – **nie ma takich rejestrów**
10. Ile rejestrów XMM posiadają w trybie EM64T proc intel – **16**
11. Instrukcje SSE wprowadzono w - **pentium III**
12. Instrukcje SSE2 wprowadzono w – **pentium IV**
13. Instrukcje AVX wprowadzono w – **sandy bridge**
14. Jednostka zarządzania pamięcią intel wprowadził w – **80386**
15. Koprocesor matematyczny intel wprowadził w – **80486DX**
16. Procesor mógł współpracować z koprocesorem w – **8088 lub 80486DX**
17. Maksymalna częstotliwość taktowania procesora 4004 to – **740KHz**
18. Pierwszy procesor marki intel został wprowadzony w - **1971r.**
19. Pierwszy 32bitowy procesor linii x86 to – **80386**
20. Procesor 8086 powstał w – **1978r.**
21. Procesor 8086 posiada – **29tys tranzystorów**
22. Procesor F14 CAD/C to procesor – **20bitowy**
23. Procesor Core 2 posiada współczynnik IPC równy – **3.5**
24. Technologia EM64T wprowadzono w – **Pentium 4**
25. Stronicowanie pamięci zostało wprowadzone w – **80386**
26. W którym procesorze można policzyć adres instrukcji  $CS*16+IP$  – **8086**
27. Tryb chroniony wprowadzono w - **80286**
28. W trybie 32bitowym procesorze rejestr wskaźnika instrukcji koprocesora jest – **40bitowy**

## Instrukcje :

1. Dodawanie zawartości do dwóch rejestrów i stałej – lea
2. By wykryć że pierwszy argument U2 porównania jest mniejszy lub równy wykonamy – jng
3. By wykryć że pierwszy argument U2 porównania jest większy wykonamy – jnle
4. Instrukcja niezminiająca flagi to: xchg
5. Instrukcja inc/dec zmienia flagi – OSZAP
6. Instrukcja zamieniająca liczbe bez znaku na podwójne słowo – movzx
7. Instrukcja zamieniająca liczbe ze znakiem bajt na podwójne słowo – movsx
8. Przesunięcie armetyczne lewo/prawo – sal/sar
9. Przesuniecie logiczne lewo/prawo – shl/shr
10. Przeszukiwanie bitów wstecz/w przód – bsr/bsf
11. Wpisanie wartości do dwóch rejestrów – LDS
12. Szyfrowanie danych – xlatb
13. Do odwołania się się do zmiennych lokalnych stosuje się rejestr – EBP
14. Współczesne procesory i7 zbudowane sa z około – ponad 731mln

## Instrukcje na łańcuchach:

1. MOVS/MOVSb/MOVSW/MOVSd/MOVSQ  
Prześlij - łańcuch/bajtów/słów/podwójnych słów/poczwórnych słów
2. CMPS/CMPSb/CMPSQ/CMPSd/CMPSQ  
Porównaj - łańcuch/bajtów/słów/podwójnych słów/poczwórnych słów
3. SCAS/SCASb/SCASW/SCASd/SCASQ  
Przeszukiwanie(skanuj) - łańcuch/bajtów/słów/podwójnych słów/poczwórnych słów
4. LODS/LODSb/LODSW/LODSd/LODSQ  
Ładuj - łańcuch/bajtów/słów/podwójnych słów/poczwórnych słów
5. STOS/STOSb/STOSW/STOSd/STOSQ  
Zapamiętaj - łańcuch/bajtów/słów/podwójnych słów/poczwórnych słów
6. Powtarzaj:

REP – dopóki ECX nie jest zerem

REPE/REPZ – dopóki equal/zero

REPNE/REPNZ – dopóki not equal/not zero

## Rodzaje adresowania:

1. Rejestrowy – Mov edx, ebx
2. Prosty natychmiastowy – mov edi, offset tabela
3. Bezpośredni – mov edi, tabela
4. Pośredni – rejestrowy – mov edi, [ebx] / mov [edi], edx
5. Pośredni – bazowy – mov edi, [ebx+tabela]
6. Pośredni – indeksowy – mov edi, [esi\*4+tabela]
7. Pośredni – bazowo-indeksowo – mov edi, [ebx+eax\*4]

## Flagi:

1. CF – flaga przeniesienia
2. 1 – zarezerwowany
3. PF – flaga parzystości
4. AF – flaga wyrównania
5. ZF – flaga zera
6. SF – flaga znaku
7. TP – flaga umożliwiająca krokowe wykonanie
8. IF – flaga zezwolenia na przerwanie
9. DF - flaga kierunku
10. Wysyłanie flag na stos PUSHF/PUSHFD/PUSHFQ
11. Pobieranie flag ze stosu POPF/POPFQ/POPFQ
12. Zmieniają przynajmniej jedną flagę: STC C, CLC C, CMC C, STD D, CLD D, STI I, CLI I, FCOM\* (KAŻDA ODMIANA)

## Inne:

IA32=8 rejestrów

MMX=8 rejestrów=XMM

EM64T=16 rejestrów=EM64T-XMM

AVX(ymm)=16 rejestrów

AVX512(zmm)=16 rejestrów

## Rejestry segmentowe:

LDS/LES/LFS/LGS/LSS – załadowanie pełnego wskaźnika z użyciem DS/ES/FS/GS/SS

CS – rejestr informujący o segmencie aktualnie wykonywanego rozkazu

DS – rejestr informujący o segmencie z danymi

ES – rejestr informujący o segmencie dodatkowym przy op. przesyłania łańcuchów

SS – **Szare Szeregi xD!** – rejestr informujący o segmencie stosu

## Kolos wykład - 1 termin:

1. W którym procesorze firmy intel pojawił się po raz pierwszy tryb chroniony – **80286**

2. Parametry aktualne przechowuje się w segmencie – **SS**

3. Indeksowy tryb adresowania występuje w instrukcji – **mov [rax\*8+tablica], rdx**

4. Która instrukcja nie jest poprawna? – **fstp st(1)**

5. Która instrukcja nie zmienia flagi P? – **paddd**

6. Po wykonaniu instrukcji – **0f8f65432h**

```
mov ecx 10023004h
```

```
mov eax 87654321h
```

```
sar eax, cl
```

```
bts eax, 23
```

7. Która z instrukcji neguje flage CF? – **CMC**

8. Która z instrukcji umożliwia wpisanie wartości do dwóch rejestrów? – **LDS**

9. Która z instrukcji pozwala na wypełnienie tekstu znakami? – **stosb**

10. Prefix LOCK nie odnosi się do instrukcji – **lea**

11. Instrukcje SSE intel wprowadził po raz pierwszy w procesorze – **pentium III**

12. Ile rejestrów YMM używają w trybie x86/x64 procesory intel? – **8/16**

13. Która instrukcja dla liczb ze znakiem zmienia słowo na podwójne słowo? – **cwde**

14. Która z instrukcji dla liczb bez znaku przesyła dla warunku większości? – **cmovnb**

15. Do zmiany kolejności słów w rejestrze MMX służy instrukcja – **pshufw**

| Tryby adresowania -                 |   |  |                                     |
|-------------------------------------|---|--|-------------------------------------|
| 1) Rejestrowy                       | push ebx<br>mov edx,ebx<br>inc ecx<br>dec r9  |  |                                     |
| 2)Prosty-Natychmiastowy             | mov al, 5;<br>mov r10d,32;<br>mov edi, ;,offset tabela<br>jnz petla;                | Argumentem instrukcji jest wartość (zawiera się w kodzie rozkazu):   |                                     |
| 3)Bezpośredni                       | mov al, [1234ec5fh]<br>mov edi, tabela;pobiera pierwszy element<br>mov zmienna, rdx | Argumentem instrukcji jest adres w pamięci (wskaźnik):   |                                     |
| 4)Pośredni - Rejestrowy             | mov al, [rcx]<br>mov edi, [ebx]<br>mov [edi], edx                                   | Argumentem instrukcji jest rejestr – wskaźnik:   |                                     |
| 5)Pośredni - Bazowy                 | mov al, [ebx+5]<br>mov edi, [ebx+tablica]<br>mov [rbp+8], rdx                       | Argumentem instrukcji jest wskaźnik:   |                                     |
| 6)Pośredni - Indeksowy              | mov al, [esi]<br>mov edi, [esi*4+tablica]<br>mov [rdi*8+tablica], rdx               | Argumentem instrukcji jest rejestr – wskaźnik:   |                                     |
| 7)Pośredni – bazowo-indeksowy       | mov al, [ebx+esi+3]<br>mov edi, [ebx+eax*4]<br>mov [rbp+rdi*8+tablica], rdx         | Argumentem instrukcji jest wskaźnik:   |                                     |
| Operacje na flagach                 |   | Operacje na bitach   |                                     |
| STC                                 | Ustawienie CF   | BT   | Testowanie bitu                     |
| CLC                                 | Zerowanie CF  | BTS  | Testowanie bitu z ustawianiem       |
| CMC                                 | Zanegowanie CF  | BTR  | Testowanie bitu z zerowaniem        |
| CLD                                 | Zerowanie DF – flagi kierunku   | BTC  | Testowanie bitu z negacją           |
| STD                                 | Ustawienie DF   | TEST   | Porównanie logiczne                 |
| LAHF                                | Przesłanie flag do rejestru AH  | BSF  | Przeszukiwanie bitów w przód        |
| SAHF                                | Przesłanie rejestru AH do flag  | BSR  | Przeszukiwanie bitów wstecz         |
| PUSHF                               | Wysłanie flag na stos   | LZCNT  | Zlicza zerowe bity od najstarszego  |
| /PUSHFD/<br>PUSHFQ                  |   | TZCNT  | Zlicza zerowe bity od najmłodszego  |
| POPF/<br>POPFD/<br>POPQ             | Pobranie flag ze stosu  | BEXTR  | Wycina ciąg bitów                   |
| STI                                 | Ustawienie IF – flagi przerwań  | BLSI   | Kopiuje najmłodszy ustawiony bit    |
| CLI                                 | Zerowanie IF  | BLSR   | Zeruje najmłodszy ustawiony bit     |
| Operacje na łańcuchach              |   | BLSMSK   | Tworzy maskę do bitu=0              |
| MOVS/MOVSb/MOVSW/MOVSd/MOVSQ        | Prześlij łańcuch/bajtów/słów/podwójnych słów/poczwórnych słów                       | BZHI   | Zeruje starsze bity                 |
| CMPS/CMPSb/CMPSw/CMPSd/CMPSQ        | Porównaj łańcuchy/bajtów/słów/podwójnych słów/poczwórnych słów                      | IA32= 8 Rejestrów<br>MMX= 8 Rejestrów=XMM<br>EM64T=16 rejestrów=EM64T-XMM<br>AVX(ymm)=16 rejestrów<br>AVX512(zmm)=16 rejestrów |                                     |
| SCAS/SCASb/SCASw/SCASd/SCASQ        | Skanuj łańcuch/bajtów/słów/podwójnych słów/poczwórnych słów                         | Instrukcje przesunięć i rotacji  |                                     |
| LDS/LDSb/LDSw/LDSd/LDSQ             | Ładuj łańcuch/bajtów/słów/podwójnych słów/poczwórnych słów                          | SAR  | przesunięcie arytmetyczne w prawo   |
| STOS/STOSb/STOSw/STOSd/STOSQ        | Zapamiętaj łańcuch/bajtów/słów/podwójnych słów/poczwórnych słów                     | SHR  | przesunięcie logiczne w prawo       |
| REP                                 | Powtarzaj dopóki ECX nie jest zerem   | SAL  | przesunięcie arytmetyczne w lewo    |
| REPE/REPZ                           | Powtarzaj dopóki equal/zero   | SHL  | przesunięcie logiczne w lewo        |
| REPNE/REPNZ                         | Powtarzaj dopóki not equal/not zero   | SARX   | przesunięcie arytmetyczne w prawo   |
| Operacje na rejestrach segmentowych |   | SHRX   | przesunięcie logiczne w prawo       |
| LDS                                 | Załadowanie pełnego wskaźnika z użyciem DS  | SHLX   | przesunięcie logiczne w lewo        |
| LES                                 | Załadowanie pełnego wskaźnika z użyciem ES  | SHRD   | przesunięcie w prawo double         |
| LFS                                 | Załadowanie pełnego wskaźnika z użyciem FS  | SHLD   | przesunięcie w lewo double          |
| LGS                                 | Załadowanie pełnego wskaźnika z użyciem GS  | ROR  | rotacja w prawo                     |
| LSS                                 | Załadowanie pełnego wskaźnika z użyciem SS  | ROL  | rotacja w lewo                      |
|                                     |   | RCR  | rotacja w prawo przez przeniesienie |
|                                     |   | RCL  | rotacja w lewo przez przeniesienie  |
|                                     |   | RORX   | rotacja w prawo                     |

|    |    |                                       |   |
|----|----|---------------------------------------|---|
| 0  | CF | flaga przeniesienia                   | S |
| 1  | 1  | zarezerwowany                         |   |
| 2  | PF | flaga parzystości                     | S |
| 4  | AF | flaga wyrównania                      | S |
| 6  | ZF | flaga zera                            | S |
| 7  | SF | flaga znaku                           | S |
| 8  | TP | flaga umożliwiająca krokowe wykonanie | X |
| 9  | IF | flaga zezwolenia na przerwanie        |   |
|    |    | X                                     |   |
| 10 | DF | flaga kierunku                        | C |

Rejestry c.d. - segmentowe  
CS - rejestr informujący o segmentacie aktualnie wykonywanego rozkazu. Razem z IP tworzy adres logiczny CS:IP kolejnej instrukcji.  
DS - rejestr informujący o segmentacie z danymi.  
ES - rejestr informujący o segmentacie dodatkowym np. przy operacjach przesyłania łańcuchów.+FS+CS  
SS - rejestr informujący o segmentacie stosu.