

1. (2 pkt.) Podaj sekwencję rozkazów umieszczającą liczbę 14,3125 na wierzchołku stosu rejestrów koprocesora arytmetycznego bez wykorzystywania segmentu danych statycznych.

```
push dword ptr 1h3125  
fld dword ptr [esp]  
add esp,4  
push dword ptr 10000  
fild dword ptr [esp]  
add esp,4  
fdiv
```

statycznych.

2. (2 pkt.) Co oznacza określenie *wyrównanie naturalne* w odniesieniu do położenia danych w pamięci? Jakie wady ma program, w którym dane nie są wyrównane naturalnie?

Wyrównanie naturalne to na przykład kiedy adres danej jest podzielny przez jej długość.  
Np. liczba 4-bajtowa została zapisana w pamięci pod adresem 456. Program, w którym dane nie są wyrównywane naturalnie posiada konsekwencje wydajnościowe. Ulokowanie danych z wyrównaniem naturalnym pozwala na uzyskanie szybszego dostępu do nich.

3. (2 pkt.) W którym programie, w którym dane nie są wyrównane naturalnie?

3. (2 pkt.) Podaj metodę wyznaczania przez procesor wartości znacznika OF przy rozkazach odejmowania?

Suma modulo dwóch najstarszych pożyczek

$$\begin{array}{r} al = \begin{smallmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 2 \\ \times & 0 & 0 & 0 & 0 & b & 0 & 0 \end{smallmatrix} \\ bl = 0 0 0 0 0 0 0 1 \\ \hline 0 1 1 1 1 1 1 1 \end{array} \quad 0 \oplus 1 = 1$$

4. (3 pkt) Podaj równoważny kod w języku C dla następującego fragmentu programu w asemblerze w trybie 32-bitowym

```
mov edi, offset dana
mov ecx, 10
mov al, '0'
cld
repne scasb
mov k, edi
```

```
int * edi = & dana;
int ecx = 10;
char al = '0';
int k = edi[0];
for (int i=0; i<ecx; i++)
    if (edi[i] == al)
        k = edi[i+1];
    break;
```

mov k, edi

5. (2 pkt.) Jakie działania należy podjąć aby ewentualne dzielenie przez zero w obliczeniach wykonywanych przez koprocesor arytmetyczny nie powodowało wyjątku?

Należy zamaskować ten wyjątek przy pomocy rozkazu fldcw (fold control word), którego typowa wartość wynosi 037Fh - samo maskowanie dzielenie przez zero to trzeba podać jedynkę na bicie o indeksie 2.

- koprocesor arytmetyczny nie powodował wyjątku.
6. (2 pkt.) Podać przyczynę stosowania oddzielnej pamięci podrzecznej L1 dla rozkazów i danych w procesorach Intel Core i7. Jak nazywa się ten rodzaj architektury?
7. (2 pkt.) Podać rozmiar pamięci podrzecznej (rozmiar

Stosowanie oddzielnej pamięci podrzecznej L1 dla rozkazów i danych to przykład architektury Harvardzkiej. Zabieg ten jest stosowany, aby przyspieszyć pracę procesora, który w tym samym momencie może wczytywać rozkaz oraz wykonywać operacje na danych bez niepotrzebnego czekania, aż któraś operacja się skończy.

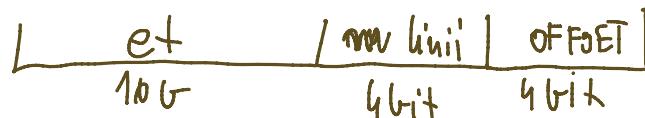
6. (2 pkt.) Podać przyczynę stosowania oddzielnej pamięci podrzecznej L1 dla rozkazów i danych w procesorach Intel Core i7. Jak nazywa się ten rodzaj architektury?
7. (2 pkt.) Podać rozmiar pamięci podrzecznej (rozmiar obszaru danych tej pamięci) o adresowaniu bezpośrednim, czterokanałowej, jeśli przyjęto adresowanie 24-bitowe i następujące rozmiary: etykiet - 16 bitów, offset 4 bitów.
8. (2 pkt.) Wyjaśnij jaki jest wpływ obsługi przerwań na nowoczesnych procesorach.

$$4 \text{-kanalny } 2^2$$

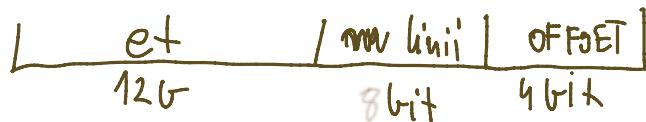
$$\text{adres} - 24 \text{ bity}$$

$$\text{etykietka} - 16 \text{ bitów}$$

$$\text{offset} - 4 \text{ bity}$$



$$2^4 = 16 \text{ B} \quad 256 \cdot 4 = 1024 \text{ B} = 1 \text{ kB}$$



1. Pamięć dwukanałowa podrzeczna, o adresowaniu bezpośredniem, jeśli adresowanie 24bitowe, etykietka 12bitów, offset 4bity

8 bitów- nr linii

$2^8 = 256$  wierszy

$2^4 = 16$  ilość kolumn

$x/2^4=2^8$

$x \cdot 2^4 \cdot 12 = 4\text{kB} = 2^2 \cdot 2^3 \cdot 10 \text{ b}$

pamięć będzie wynosiła  $2^2 \cdot 4\text{kB} = 8\text{kB}$

$$2^4 = 16 \text{ B} \quad 256 \cdot 16 \cdot 2 = 8192 = 8 \text{ kB}$$

$$2^8 = 256$$

- następujące rozmiary: etykietka  
8. (2 pkt.) Wyjaśnij jaki jest wpływ obsługi przerwań na zużycie energii w nowoczesnych procesorach.

nowystawiania wyjątku nr 6

8. (2 pkt) Wyjaśnij, czym różni się wykorzystanie energii w nowoczesnych procesorach od wykorzystania energii w procesorach z lat 80. i 90. XX wieku?
9. (2 pkt.) Wyjaśnij przyczynę powstawania wyjątku nr 6 (divide by zero). Wyjaśnij różnicę pomiędzy ścisłe i luźno określonymi kodami operacji dzielenia.

kod niepowiązany z żadną instrukcją

INVALID OPCODE

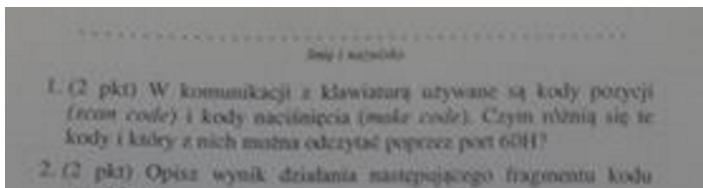
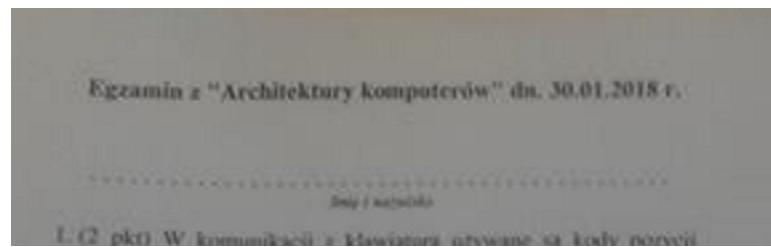
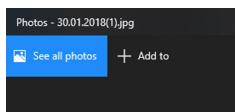
- powiązanymi systemami HW/LD.
- 11.(2 pkt) Wyjaśnij różnicę pomiędzy interfejsem ABI a API?

w szczególności ABI definiuje standard wywoływanego (ang. calling convention), który określa sposób wywoływanego funkcji, przekazywania jej argumentów, zwracania obliczonej wartości, podaje wykaz rejestrów, których zawartości powinny być zachowane, itp.; m.in. standard ten określa czy parametry przekazywane są przez rejestyry czy przez stos, a jeśli przez stos to w jakiej kolejności są ładowane, czy dopuszcza się zmienną liczbę argumentów, itd.

API - pewien interfejs określający sposób porozumiewania się programu z programem lub z systemem operacyjnym

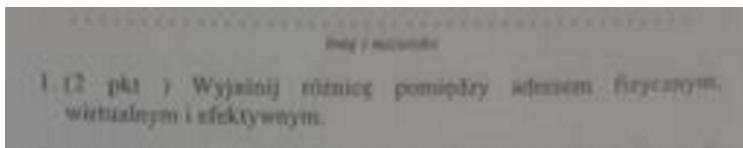
- API?
12. (2 pkt) Na czym polega różnica między przesunięciem logicznym a przesunięciem arytmetycznym?

Przesunięcie logiczne to SHL i SHR, w momencie kiedy przesuwamy w lewo to na najmniej znaczącym bicie pojawia się zero, a jak w prawo to na najbardziej znaczącym bicie też pojawia się zero. Z kolei przy przesunięciu arytmetycznym są to rozkazy SAL i SAR, w przypadku SAL działa on tak samo jak SHL czyli na najmniej znaczących bitach pojawiają się zera. Z kolei SAR w przypadku liczb ujemnych na najstarszych bitach umieści 1, a w przypadku liczb dodatnich 0. Dlatego przy pomocy SAR możemy dzielić przez  $2^k$  liczby w kodzie U2, gdzie k to ilość przesuniętych bitów.



Kod pozycji jednoznacznie określa klawisze, ale nie są identyczne z kodami pozycji. Kody naciśnięcia nie

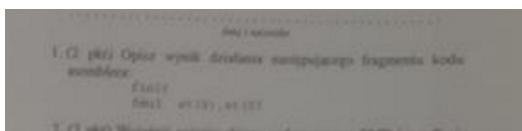
występują na poziomie programowania. Z portu 60H można odczytać kod pozycji. Niektóre kody pozycji mają te same wartości dla różnych klawiszy np. lewy i prawy ctrl dlatego stosuje się dodatkowe bajty E0H i E1H



Adres fizyczny - wskazuje jednoznacznie na komórkę pamięci.

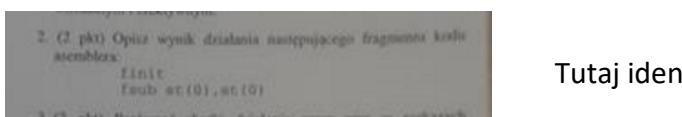
Adres wirtualny - iluzja pamięci fizycznej, jest to pamięć operacyjna komputera z perspektywy programisty.

Adres efektywny - adres lokacji pamięci (wirtualnej?) zawierającej argument, na którym zostanie wykonana operacja.

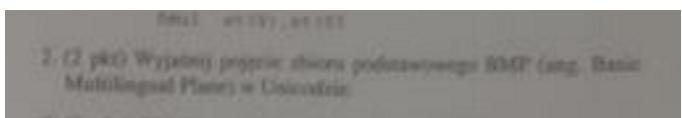


finit    wynik to IND  
fmul st(0), st(1)

Indeterminate Number - Nieokreślony numer, bo nie wiadomo co po finit było na koprocessorze dlatego nie wiadomo co będzie po wykonaniu tej operacji.

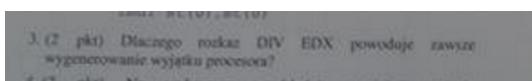


Tutaj identycznie



Wyjaśnij pojęcie zbioru podstawowego BMP w unicodzie.

Jest to większość powszechnie używanych znaków przyporządkowana punktom kodowym od 0 do 65535. Wartości punktów kodowych ze zbioru BMP dają się przedstawić w postaci 16-bitowych liczb binarnych.



Generuje wyjątek, ponieważ dzielenie przechowuje wynik w rejestrze 32 bitowym EAX, więc jeżeli podzieli się liczbę EDX:EAX przez EDX to dostajemy  $(EDX * 0x100000000 + EAX)$  co jest zawsze większe od  $0x100000000$  i nie zmieści się do rejestrów 32 bitowych.

Test 41 (01.06.10)

3. (2 pkt) Porównaj skutki działania przez zero w rozkazach wykonywanych przez procesor i koprocesor arytmetyczny.

4. (2 pkt) Na podstawie rozkładu wyników podaj:

Podczas dzielenia przez 0 w rozkazie wykonanym przez procesor wystąpi wyrzucenie wyjątku i przerwanie programu. Ta sama operacja wykonywana przez koprocesor nie przerwie działania programu, ale tylko wpisze jako wynik operacji wartość specjalną INF.

3. (2 pkt) Dlaczego wykonanie działania 1.0/4.0 przez koprocesor arytmetyczny nie prowadzi nigdy do wygenerowania wyjątku, natomiast dzielenie 1.0/3.0 może prowadzić do takiego wyjątku?

4. (2 pkt) Podaj wynik

Dlaczego dzielenie 1.0/4.0 nie powoduje wyjątku a 1.0/3.0 powoduje, bo jest to nieskończone rozwinięcie i nie jesteśmy w stanie podać dokładnej wartości, za ten wyjątek odpowiada precision flag.

arytmetyczny nie prowadzi nigdy do wygenerowania wyjątku, natomiast dzielenie 1.0/3.0 może prowadzić do takiego wyjątku?

4. (1 pkt) Podaj wartości znaczników zadanego, przekształcenia i rozszerzenia rozkazu:

addl 10, 11

movl 12, 13

5. (3 pkt) W pewnym kontekście użyciwa jest pamięć podlegająca o pojedynczości 1024 bajtów. W pamięci nie ma wolnego jest

CF 0      OF 0      ZF 1

subl dl, al  
movl dl, 1



Bo mov nie zmienia flag

4. (2 pkt) Na wybranym przykładzie wyjaśnij pojęcie heterogeniczności architektury systemu komputerowego.

5. (2 pkt) Podaj równoważny fragment kodu w zasobniku bez użycia rozkazów skoku (przyjąć, że 32-bitowa zmieniona k jest w rejestrze eax, zaś m w ebx)

if (k1<-5.0) m=2.0; else m=3.0;

6. (2 pkt) Podaj równoważny kod w języku C dla następującego

$$k = \alpha x \quad m = \beta x$$

cmp eax, -5

je dalej

mov ebx, 3

jmp koniec

dalej: mov ebx, 2

koniec:

5. (2 pkt) Podaj równoważny fragment kodu w assemblerze bez użycia rozkazów skoku (przyjąć, że 32-bitowa zmenna x jest w rejestrze eax, zaś y w ebx)

```
if (x<=-5) y=2; else y=3;
```

$$x = \text{eax} \quad y = \text{ebx}$$

```
cmp eax, -5
```

```
jbe dalej
```

```
mov ebx, 3
```

```
jmp koniec
```

dalej: mov ebx, 2

koniec:

6. (2 pkt) Podaj równoważny kod w języku C dla następującego fragmentu programu w assemblerze w trybie 32-bitowym

```
dana dd 3  
mov ecx, 32  
etik:  
    cmp dana, 0  
    ja koniec  
    inc dana  
    loop etik  
koniec:
```

```
long long dana = 3;  
int ecx = 32;  
for (int i = ecx; i >= 0; i--)  
{  
    if (dana == 0)  
        goto koniec;  
  
    dana++;  
}
```

koniec

```
if (x<=-5) y=2; else y=3;
```

6. (2 pkt) Podaj równoważny kod w języku C dla następującego fragmentu programu w assemblerze w trybie 32-bitowym

```
dana dd -3.0  
fld1  
fcomi dana  
jne etyk  
mov dana, 0C0000000h  
etyk:
```

7. (2 pkt.) Dlaczego stosowanie układów DMA podnosi wydajność?

float dana = -3.0;

if (dana != 1)

goto etyk;

dana = 0x0C000000;

etylk:

Koniec.

7. (2 pkt.) Podaj sposób wyznaczania wartości znaczników CF i OF przez procesor w trakcie operacji arytmetycznych.

8. (2 pkt.) Wyjaśnij znaczenie buforu BTB. Jaki rodzaj pamięci jest

CF - jeżeli wystąpiło połączenie mu nie będzie żadnego znaczącego bitu przy odejmowaniu, albo przeniesienie z małego buforu znaczącego bitu.

OF - suma % 2 dwóch ostatnich małych przeniesień

7. (2 pkt.) Dlaczego stosowanie układów DMA podnosi wydajność przetwarzania?

DMA - Direct Memory Access służy do przesyłania danych z pamięci do pamięci lub urządzeń I/O bez użycia procesora. Jedyne co potrzebuje to impuls z procesora prowadzący do rozpoczęcia przesyłu danych, kiedy skończy sygnalizuje to w sposób przerwania sprzętowego. Podnosi wydajność, bo w czasie kiedy procesor przesyłał by dane to nic by nie mógł robić, a tak może wykonywać kolejne rozkazy w tym czasie.

Podaj przykład jej fizycznej realizacji.

7. (2 pkt.) Opisz wynik działania następującego fragmentu kodu assemblej:

```
xor ebx, ebx  
sub ebx, 1  
etk:  
lea ebx, [ebx+8*ebx]  
jnc etk
```

Xor ebx, ebx

Sub ebx, 1

Etk:

Lea ebx, [ebx+8\*ebx]

Jnc etk

Lea wyznaczy wartość 9\*FFFFFFF,  
Carry flag jest ustawiony z operacji sub  
dlatego pętla wykona się tylko raz.

8. (2 pkt.) Wyjaśnij znaczenie bufora BTB. Jaki rodzaj pamięci jest w nim wykorzystywany?  
 9. (2 pkt.) Przedstaw budowę komórkę pamięci Flash

Bufor taki przechowuje historię 256 skoków warunkowych i na jego podstawie procesor przewiduje zachowanie się danej instrukcji skoku. Pozycja skoku zawiera informacje:

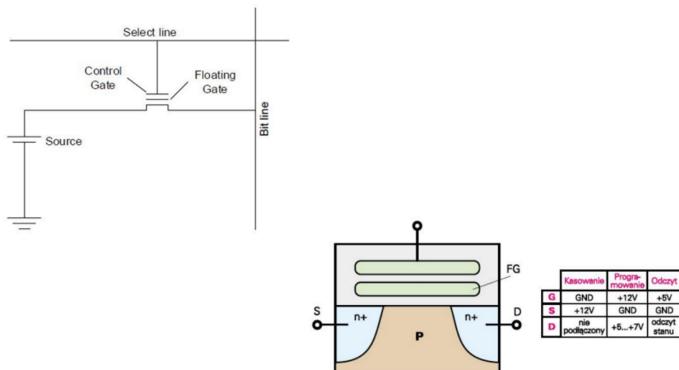
bit ważności - czy zawarte informacje są aktualne,  
 bity historii - jak często realizowane były rozgałęzienia,  
 adres źródłowy rozkazu rozgałęzienia,  
 adres docelowy rozgałęzienia.

Chyba SRAM?

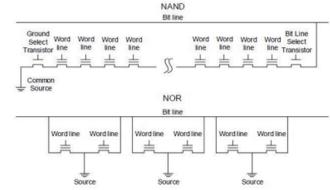
- w nim wykorzystywany?  
 9. (2 pkt.) Przedstaw budowę komórkę pamięci Flash.  
 10. (2 pkt.) Wysokość maksymalna pojemności pamięci flash

## Budowa komórka pamięci flash

## Właściwości pamięci flash

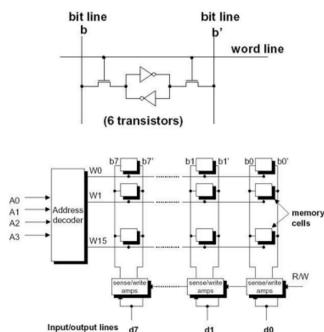


- Pływająca bramka (FG) stanowi kondensator o bardzo małej pojemności i upływności – pułapka potencjału.
- Bramki NAND łączy się szeregowo, bramki NOR równolegle.
- Pozwala to na dostęp do pojedynczego bitu w NOR i blokowy w NAND
- NAND w porównaniu do pamięci NOR mają krótsze czasy zapisu, odczytu, większą gęstość zapisu i co za tym idzie niższy koszt na MB pojemności. Wytrzymałość liczby cykli zapisu jest 10 razy większa
- Czas zapisu i odczytu 70-120 ns
- Kasowania sektora 0,7 s (64kB)



- w nim wykorzystywany?  
 9. (2 pkt.) Przedstaw budowę komórkę pamięci SRAM.

### Pamięć statyczna SRAM



- Przerutnik dwustanowy

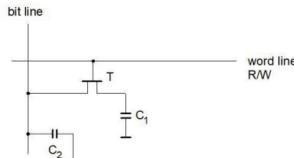
#### Zalety

- Brak potrzeby odświeżania zawartości
- Krótki czas dostępu ~10 ns (5-10 cykli)
- Krótszy cykl odczytu
- Wysoka odporność na zakłócenia

#### Wady

- Duży pobór mocy
- Mały stopień scalenia

### Pamięć dynamiczna DRAM



- Tranzystor i kondensator

#### Zalety

- Niski pobór mocy
- Duży stopień scalenia
- Niski koszt wykonania

#### Wady

- Czas dostępu ~60 ns
- konieczność odświeżania zawartości
- dłuższy cykl odczytu
- Niska odporność na zakłócenia

Flash.

10. (2 pkt.) Wyjaśnij różnicę pomiędzy interfejsem ABI a API?

w szczególności ABI definiuje standard wywoływania (ang. calling convention), który określa sposób wywoływania funkcji, przekazywania jej argumentów, zwracania obliczonej wartości, podaje wykaz rejestrów, których zawartości powinny być zachowane, itp.; m.in. standard ten określa czy parametry przekazywane są przez rejestyry czy przez stos, a jeśli przez stos to w jakiej kolejności są ładowane, czy dopuszcza się zmienną liczbę argumentów, itd.

API - to interfejs określający porozumiewanie się programu z systemem lub programem.

Liczba pkt.

11. (2 pkt.) Omówić główne zadania jednostki sterującej w procesorze biorąc pod uwagę informacje dostarczane na jej wejście i informacje generowane na wyjściu.

Wykonywanie rozkazu przez procesor rozpoczyna się pobraniem rozkazu z pamięci, po czym identyfikowany jest jego kod - na tej podstawie jednostka sterująca w procesorze wydaje odpowiednie dyspozycje dla jednostki arytmetyczno-logicznej, rejestrów i innych podzespołów procesora, kierując odpowiednio przepływem i przetwarzaniem danych. Na wyjściu są mikrorozkazy do innych układów procesora jakie rozkazy mają wykonać.

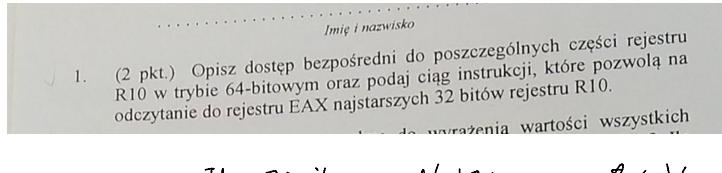
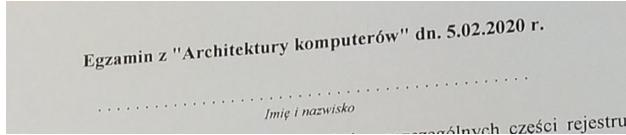
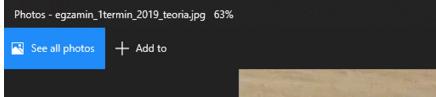
12. (2 pkt.) Jaki wpływ na wykonywanie niektórych rozkazów ma przedrostek (dodatkowy bajt) REP (kod F3H)?

Jeżeli damy jakąś wartość do ecx to przedrostek rep przy niektórych rozkazach powoduje powtarzanie tego rozkazu, po każdym powtórzeniu zawartość ecx jest zmniejszana o 1 az do momentu jak ecx = 0 wtedy jest kończone

11. (2 pkt.) Omów technikę statycznego szeregowania rozkazów.

12. (2 pkt.) Podaj definicję atrybutu CL przy dostępie do pamięci RAM.

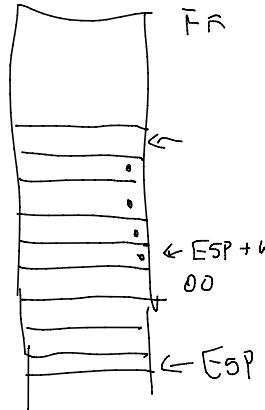
**CL - CAS LATENCY** - czas pomiędzy sygnałem cas a pojawieniem się pierwszych danych do odczytu



16bitów

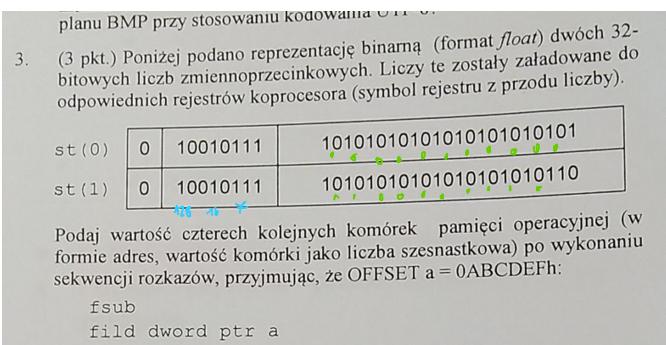
r10      r10d      r10w      r10b

push r10  
mov eax,[esp+4]



odczytanie do rejestru EAX najstarszych 32 bitów  
(1 pkt.) Ile bitów jest potrzebne do wyrażenia wartości wszystkich punktów kodowych z planu podstawowego BMP w Unicodzie? Ile bajtów potrzebujemy na wyrażenie maksymalnej wartości punktu z planu BMP przy stosowaniu kodowania UTF-8?

16 bitów, a w UTF-8 3 bajty



$$s+(1) - s+(0)$$

138  
144  
151

$$151 - 128 = 24$$

$$s+(1) = s = 0$$

M =

$$\begin{aligned} &1.10101010101010101010101010 \xrightarrow{\text{shift left}} 0^2 \cdot 2^{24} \\ &1.10101010101010101010101010 \xrightarrow{\text{shift left}} 1 \cdot 2^{24} \\ &\underline{0.000000000000000000000001} \cdot 2^{24} \end{aligned}$$

$$1 \cdot 2 = 10 = 2$$

$$s = 0$$

$$M = 128$$

$$M = 00..00$$

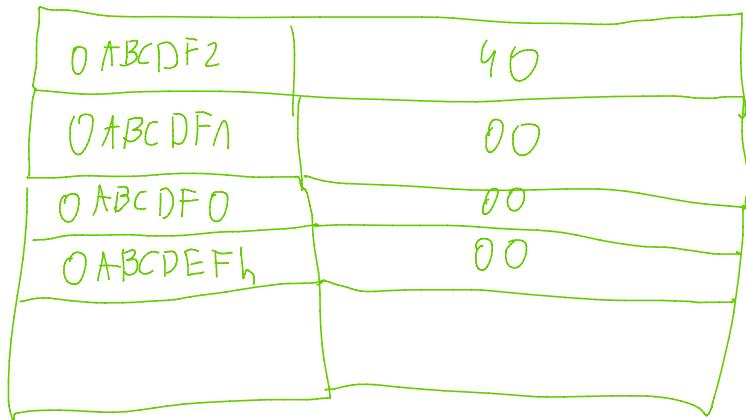
$$0100\ 0000\ \dots\ 0000$$

Ad = 128

Ad = 00..00

0100 0000 ... 0000

4000 0000



4. (2 pkt.) W wyniku wykonania poniższego fragmentu programu do rejestru EAX została wpisana liczba 0CAFEBABEH. Podać liczbę, która została wpisana do rejestru EDX (w podanym przykładzie rozkazy MOV zajmują po 5 bajtów).

```
alfa:  
mov      eax, OFFSET alfa  
mov      edx, OFFSET $-5
```

\*  
0CAFEBABEh      mov eax,OFFSET alfa

0CAFEBABE 1011 1110  
0101  
-----  
1100 0011

0CAFEBAC3h

0CAFEBABEh      mov eax,OFFSET alfa

0CAFEBAC3h      mov edx,OFFSET \$-5

```
mov      edx, OFFSET $-5
```

5. (2 pkt.) W trybie 16-bitowym, w pamięci pod adresem fizycznym 12345h znajduje się nowa procedura obsługi przerwania klawiatury (nr 9, IRQL). Podaj sekwencję rozkazów asemblera, która umieści (nadpisze) w tablicy wektorów przerwań właściwe wartości.

6. (2 pkt.) Jakie informacje są przesyłane pomiędzy mikrokontrolerem

```
mov ax, 0  
Mov ds, ax  
cli  
mov ds:[36], word ptr 45h
```

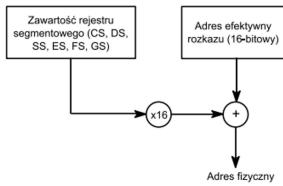
OFFSET  
SEG      12345h

SEG:OFFSET

<pre> cli mov ds:[36], word ptr 45h mov ds:[38], word ptr 1230h sti </pre>	OFFSET SEG 1230 1234	12345 h
--	-------------------------------	---------

jest na rysunku.

W praktyce programowania (w trybie rzeczywistym) zazwyczaj nie podaje się 20-bitowego adresu fizycznego rozkazu (instrukcji) lub danej, ale adres ten wyraża się w postaci dwóch liczb szesnastkowych rozdzielonych dwukropkiem: pierwsza z tych liczb wskazuje początek segmentu, w którym znajduje omawiany obiekt, druga zaś określa przesunięcie wewnątrz segmentu, tj. odległość obiektu od początku segmentu (wyrażona w bajtach). Przykładowo, 32-bitowy programowy licznik czasu BIOS umieszczony jest w lokacji pamięci o adresie 40H:6CH, tzn. w lokacji pamięci o adresie fizycznym  $40H \times 16 + 6CH = 46CH$ . W odniesieniu do podanego schematu adresowanego mówimy, że adres lokacji pamięci został wyrażony w postaci *segment : offset*.



6. (2 pkt.) Jakie informacje są przesyłane pomiędzy mikrokontrolerem klawiatury a sterownikiem (kontrolerem) klawiatury (np. 8041) znajdującym się na płycie głównej, a jakie informacje są lokowane przez sterownik klawiatury w izolowanej przestrzeni wejścia-wyjścia?

7. (2 pkt.) Porównać skutki dzielenia przez zero w rozkazach

Po naciśnięciu lub zwolnieniu dowolnego klawisza mikrokontroler (mikroprocesor) klawiatury formuje kod naciśnięcia (ang. make code) lub kod zwolnienia (ang. break code) klawisza — kod ten zostaje przesłany do układów płyty głównej komputera. Kod zwolnienia zawiera kod naciśnięcia poprzedzony bajtem F0H.

Odtworzony kod pozycji (lub kod pozycji zwiększonego o 128) zostaje udostępniony w porcie 60H – kod pozycji naciśniętego (lub zwolnionego) klawisza można więc odczytać za pomocą rozkazu in al, 60H . Jednocześnie generowany jest sygnał przerwania poprzez linię IRQ 1, z którą skojarzony jest wektor przerwania nr 9. Zarówno mikrokontroler na płycie głównej jak też mikrokontroler klawiatury mogą być programowane za pomocą rozkazów wysyłanych przez główny procesor poprzez porty 60H i 64H. Istnieje kilkanaście rozkazów, które mogą być przesyłane do klawiatury, m.in. w ten sposób można określić parametry tzw. autorepetycji, czyli samoczynnego powtarzania wysyłania kodu odpowiadającego przytrzymywaniu dłużej klawiszowi. Schemat blokowy sterownika klawiatury zainstalowanego na płycie głównej pokazany jest na rysunku.

Mikrokontroler wysyła make code i break code czyli kod naciśnięcia i kod zwolnienia do sterownika klawiatury, który zamienia te kody na kod pozycji scan code. W porcie lokowane są właśnie kody pozycji (scan code).

sterownik klawiatury w izolowanej przestrzeni wejścia-wyjścia?

7. (2 pkt.) Porównać skutki dzielenia przez zero w rozkazach wykonywanych przez procesor i koprocesor arytmetyczny.

Dzielenie przez 0 na koprocesorze wykaże wartość specjalną INF (i możliwe że wyrzuci program w zależności od ZM), z kolei dzielenie przez 0 w procesorze

wyrzuci wyjątek i przerwie program.

wykonywany przez procesor i koprocesor arytmetyczny.

8. (2 pkt.) W jakich okolicznościach w kodzie programu stosowany jest przedrostek REPNE?  
9. (2 pkt.) Podaj definicję czasu cyklu (*cycle time*) w dostępie do pamięci.

Na pewno w SCASIE i w CMPSIE

Przedrostek REPNE.

9. (2 pkt.) Podaj definicję czasu cyklu (*cycle time*) w dostępie do pamięci.

Cycle time - określa najkrótszy czas pomiędzy kolejnymi odwołaniami do pamięci.

10. (3 pkt) W pewnym komputerze używana jest pamięć podręczna czterokanałowa o całkowitej pojemności obszaru danych 16 kB ( $=2^{14}$ ) bajtów. W pamięci tej stosowane jest odwzorowanie bezpośrednie z 16-bajtowymi blokami. Algorytm wymiany bloku w kanale to LRU. Wyznaczyć współczynnik trafienia dla podanego niżej kodu (podać pełne obliczenia), przy następujących założeniach:
- zmienna x ma adres 0;
  - pamięć podręczna jest inicjalnie pusta;
  - dostęp do pamięci odbywa się tylko przy dostępie do zmiennej x, pozostałe zmienne zapamiętane są w rejestrach;
  - w kompilatorze nie jest stosowana optymalizacja.
- ```
char x[8192];
for (int i = 0; i < 4096; i++)
{
    x[i+4096] += x[i];
```

11. (2 pkt.) Omówić rolę bufora rozgałęzień (BTB/Branch Target Buffer) w mikroarchitekturach procesorów.

12. (2 pkt.) Wyjaśnić...

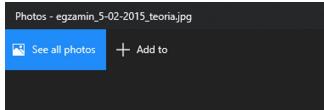
Bufor taki przechowuje historię 256 skoków warunkowych i na jego podstawie procesor przewiduje zachowanie się danej instrukcji skoku. Pozycja skoku zawiera informacje:

bit ważności - czy zawarte informacje są aktualne,  
bity historii - jak często realizowane były rozgałęzienia,  
adres źródłowy rozkazu rozgałęzienia,  
adres docelowy rozgałęzienia.

12. (2 pkt.) Wyjaśnić, na czym polega zjawisko hazardu zasobów w procesorze. Opisz wybraną metodę, która pozwala na jego zminimalizowanie w procesorach Intel Core.

4. hazardы :

- a. **konflikt zasobów** - gdy dwie instrukcje wymagają dostępu do tych samych zasobów procesora (przykładowe rozwiązanie to oddzielna pamięć cache dla danych i dla rozkazów, odczytywanie danych z pamięci z wyprzedzeniem i wprowadzenie ukrytych rejestrów gdy występuje konflikt rejestrów),
- b. **konflikt danych** - jeśli dwie kolejne instrukcje wykonują operacje na tych samych rejestrach, może zaistnieć sytuacja gdy jedna instrukcja potrzebuje wartości rejestrów, który jest wynikiem obliczeń poprzedniej instrukcji i nie jest jeszcze dostępny (przykładowe rozwiązanie to wstrzymywanie potoku, tzw. *pipeline stall*, który właściwie jest wstrzymaniem potoku, aż wynik obliczeń poprzedniej instrukcji będzie gotowy),
- c. **konflikt sterowania** - każdy skok w programie powoduje zakłócenie kolejki instrukcji, występujących kolejno w pamięci. Poznanie następnej instrukcji do wykonania po skoku warunkowym jest możliwe dopiero po pewnej liczbie cykli zegara - w tym czasie procesor może rozpoczęć wykonywanie kolejnych instrukcji i unieważnić je w zależności od tego czy skok rzeczywiście się wykona czy nie (przykładowe rozwiązanie to przewidywanie statyczne lub dynamiczne skoków).



1. (1 pkt.) Do organizacji pętli rozkazowych używane są rozkazy skoku, które zazwyczaj ... zawartość wskaźnika instrukcji EIP.

2. (1 pkt.) Czym różni się rozkaz CALL od rozkazu INT ?

Call to jak wywołujesz podprogram, a int to jak przerwanie

Call - ret

Int - iret

Int - EIP, CS, FLAGI

CALL - ślad

2. (1 pkt.) Czym różni się rozkaz CALL od rozkazu INT ?

3. (3 pkt.) Podać zawartość czterech bajtów pamięci (w układzie adres: wartość), które zostały zapisane w wyniku wykonania poniższych rozkazów

|         |                                                   |
|---------|---------------------------------------------------|
| 46      | 7E45BBH                                           |
| 50      | 7E45BAH                                           |
| 00      | 7E45B9H                                           |
| 00      | 7E45B8H                                           |
| 7E45B7H | jeśli wiadomo, że na wierzchołku stosu            |
| 7E45B6H | rejestrów koprocesora arytmetycznego              |
| 7E45B5H | znajdowała się liczba o wartości dziesiętnej +4.5 |

4. (1 pkt.) W jaki sposób można zapisać na stosie daną 8-bitową

1001

$1.001 \cdot 2^2$

0 100 0000 1001 . . . 0000

4 090 00 00

Szeregowanie statyczne (odpowiednia kompilacja usuwa 70% zależności)

Szeregowanie dynamiczne (sprzętowe) – prekursor Out of Order

Technika notowania (scoreboarding)

4. (1 pkt.) W jaki sposób można zapisać na stosie daną 8-bitową znajdującą się w rejestrze CL ?

movsx eax, cl

dec esp

and ecx, 000000FFh

push eax

mov [esp], cl

push ecx

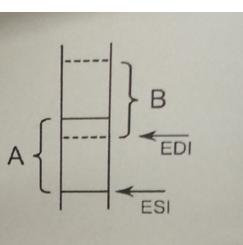
5. (2 pkt.) Jaką rolę w trakcie asemblacji programu pełni licznik lokacji?

\$ Org \$ - 5

Licznik lokacji to rejestr programowy definiowany przez asembler. Określa on adres komórki pamięci operacyjnej, do której zostanie przesłany aktualnie tłumaczony rozkaz lub dana. Po załadowaniu rozkazu lub danej, licznik lokacji zostaje zwiększony o ilość bajtów zajmowanych przez ten rozkaz lub daną. Przed rozpoczęciem tłumaczenia pierwsza linia licznika lokacji jest równy zero.

lokacji?

6. (3 pkt.) W programie używane są dwa 50-bajtowe obszary pamięci A i B, których początek wskazują, odpowiednio, rejesty ESI i EDI (zob. rys.). Napisać fragment programu w asemblerze, który korzystając z rozkazu MOVSBL prześle zawartość obszaru A do obszaru B. Wskazówka: rozkaz CLD zera znacznik DF, a rozkaz STD wpisuje 1 do tego znacznika.



STD

add edi, 40

add esi, 40

mov ecx, 50

rep movsb

7. (2 pkt.) Podać numer linii i wartość etykiet, które należy sprawdzić w pamięci podrzędnej o adresowaniu bezpośrednim przy wykonaniu następującej sekwencji rozkazów

mov esi, 10195FC0h  
mov al, [esi]

Zakłada się adresowanie 32-bitowe, rozmiar pamięci podrzędnej 256 kB, rozmiar linii 32 bajtów



256 kB

$$2^{10} \cdot 2^8 = 2^{18}$$

$$32 - 14 - 5 = 13$$

$$\underline{256 \text{ kB}} \quad 2^{10} \cdot 2^8 = 2^{18}$$

$$32 - 14 - 5 = 13$$

$$\text{linia } 32B = 2^5$$

$$\frac{2^{32}}{2^{18}} = 2^{14}$$

$$\frac{2^{18}}{2^9} = 2^{13}$$

0001 0000 0001 101 0101 1111 110 0000

etylatku

unid

oparte na VPU?

9. (3 pkt.) Wyjaśnić różnicę między statycznym i dynamicznym sposobem przewidywania wykonania rozgałęzienia w procesorach?

Do 75% statyczne ma tylko statycznie zakodowane, że dane rozkazy zawsze skaczą, a niektóre nie zawsze.

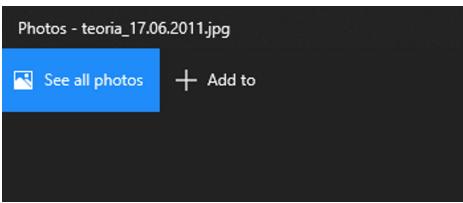
Dynamiczne - oparte na historii wykonanych skoków. Rejestruje się wystąpienie lub niewystąpienie skoku za pomocą *bitu skoku*. Jeśli skok poprzednio wystąpił zakłada się, że wystąpi ponownie. W takim podejściu w przypadku pętli błąd zdarza się dwukrotnie. Lepsze rezultaty uzyskuje się stosując więcej bitów historii lub implementując zaawansowane tablice historii skoków, na podstawie których procesor wybiera bardziej prawdopodobną ścieżkę wykonania programu (np. *Branch Target Buffer*). Istnieją także inne sposoby takie jak : dwa bufory typu FIFO, technika zwielokrotniania strumieni czy bufor pętli.

10. (3 pkt.) Wyjaśnić wpływ występowania przerwań sprzętowych na wydajność IPC procesora z przetwarzaniem potokowym i dużej głębokości potoku.

Działa to tak, że im większy potok to niby powinno być okej, ale tak na prawdę jak przyjdzie co do czego to jak wystąpi dużo przerwań, które psują to przetwarzanie potokowe to wyjdzie gorzej niż jakby ta głębokość potoku była mniejsza.

11. (3 pkt.) Wyjaśnić koncepcję wyrównywania adresów danych w pamięci operacyjnej i wpływ na wydajność nowoczesnych procesorów.

Wyrównywanie adresów to koncepcja polegająca na tym, że dane są kodowane na adresach, które są podzielne przez kolejne potęgi dwójki. Na przykład podczas pobierania danych do pamięci podręcznej trzeba czasem 2 razy załadować zamiast raz. :D



Egzamin z "Architektury komputerów" dn. 17.06.2011 r.

1. (2 pkt.) Podać zawartość rejestru EIP po wykonaniu poniższej

1. (2 pkt.) Podać zawartość rejestru EIP po wykonaniu poniższej sekwencji rozkazów

```
mov    eax, 789
xchg  eax, [esp]
ret
```

2. (2 pkt.) W komórkach

$$EIP = 789$$

1. (1 pkt.) Podać sekwencję rozkazów (zawierającą rozkaz RET), której wykonanie spowoduje załadowanie do wskaźnika instrukcji EIP liczby 456.

push dword ptr 456  
ret

Jmp i sprawdza

1. (2 pkt.) Podany niżej rozkaz JMP typu pośredniego kopiuje zawartość wierzchołka stosu do rejestru EIP

```
JMP    dword PTR [esp]
```

Wysiadź, co innego różni się podany rozkaz od rozkazu RET

met popuże ze stosu  
a jmp poprosta skacze

ret

|  |    |    |
|--|----|----|
|  | 03 | 02 |
|--|----|----|

00430F76H 00430F75H 00430F74H

2. (2 pkt.) W komórkach pamięci operacyjnej o adresach 0x00430F76 i 0x00430F75 została zapisana liczba 515 w postaci 16-bitowej liczby binarnej. Podaj zawartości tych komórek w postaci binarnej przy założeniu, że w komputerze stosowana jest konwencja mniejsze wyżej (ang. big endian). W komputerze stosowana jest pamięć o organizacji bajtowej.

515  
1 0 0 0 0 0 0 1 1  
512 256 128 64 32 16 8 4 2 1

0000 0010 0000 0011  
02 03

3. (1 pkt.) Wyjaśnić znaczenie terminu ramka stosu.

Ramka stosu - pojęcie ramki stosu zostało wprowadzone, aby w łatwy sposób operować stosem w procedurach. Wskaźnik ramki stosu to EBP. W taki sposób postugując się rejestrem EBP, w stły sposób jesteśmy w stanie odwoływać się do parametrów funkcji, czy wydzielić sobie miejsce na zmienne lokalne i się do nich odnosić.

bajtowej.

3. (1 pkt.) W jaki sposób w kodzie assemblerowym rezerwuje się obszary danych statycznych i dynamicznych.

Dane statyczne rezerwujemy w obszarze .data albo na przykład poprzez zarezerwowanie pamięci na te zmienną przy pomocy funkcji malloc, natomiast dane dynamiczne są tworzone na stosie podczas wywoływania podprogramu i po wywołaniu tego podprogramu już nie mamy do nich dostępu.

czyli wywoływanie.

3. (1 pkt.) Wyjaśnić znaczenie terminu *interfejs ABI*.

Interfejs ABI, opisuje sposób wywoywania podprogramów. W jaki sposób przekazujemy parametry. Jak działają podane standardy. Na przykład czy ładujemy parametry przez stos czy przez rejesty. Jeżeli przez stos to od prawej czy od lewej. Czy sami sprzątamy po tych parametrach czy robi to podprogram. Jakie rejesty musimy zachować w trakcie tego podprogramu.

3. (1 pkt.) Wyjaśnić znaczenie terminu *ramka stosu*.

4. (1 pkt.) W jaki sposób sygnalizowany jest nadmiar w trakcie wykonywania rozkazu DIV lub IDIV?

Wyswietlono pytanie o różnicę między rozkazami FLD i FILO?

W postaci wyrzucenia wyjątku procesora i przerwania programu.

obszary danych statycznych i dynamicznych.

4. (2 pkt.) W jaki sposób procesor wyznacza wartości wpisywane do znaczników CF i OF w trakcie wykonywania dodawania.

5. (2 pkt.) W jaki sposób procesor rzuca się zaokrąglenie w kierunku zera

Czy była pożyczka, i xor 2 najstarszych przeniesien

3. (1 pkt.) Wyjaśnić znaczenie terminu *interfejs ABI*.

4. (2 pkt.) Czas wykonywania operacji dzielenia liczb całkowitych przez  $2^k$  można znacznie skrócić, jeśli do tego celu zostanie wykorzystany rozkaz *SAR*.

5. (2 pkt.) W jakich okolicznościach konieczny jest modyfikator

wykonywania rozkazu DIV lub IDIV?

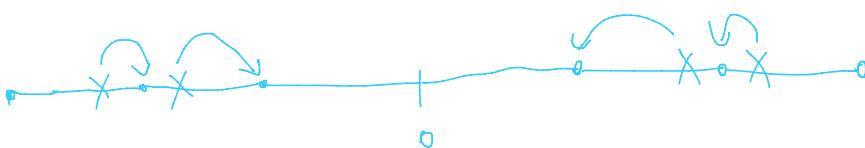
5. (1 pkt.) Na czym polega różnica między rozkazami FLD i FILO?

INT - FILD

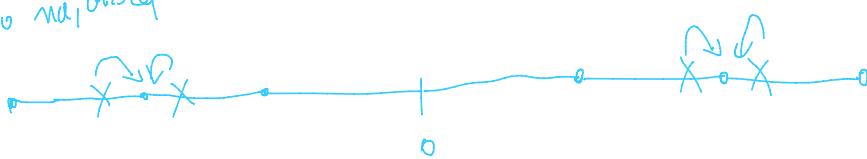
## FLOAT - FLD

4. (2 pkt.) W jaki sposób procesor wyznacza do znaczników CF i OF w trakcie wykonywania dodawania.
5. (2 pkt.) Wyjaśnić czym różni się zaokrąglenie w kierunku zera od zaokrąglenia do liczby najbliższej stosowane w koprocesorze arytmetycznym.

w kierunku 0



do najbliższej



5. (1 pkt.) W jakich okolicznościach koprocesor arytmetyczny generuje wyjątek niedozwolona operacja?

Wyjątek niedozwolona operacja powstaje, gdy niemożliwe jest żadne inne działanie, np. próba obliczenia pierwiastka z liczby ujemnej, próba użycia pustego rejestru stosu.

Jeśli bit IE jest zamaskowany (tj. IM = 1), to wynikiem operacji jest nieliczba – NaN (ang. Not a Number).

5. (1 pkt.) Na czym polega rozszczepianie?
6. (3 pkt.) Omówić sposób odwzorowania zawartości pamięci ekranu w trybie graficznym 13H na postać obrazu wyświetlanego na ekranie.

Sterownik VGA oferuje prosty tryb graficzny oznaczony numerem **13h**, w którym raster ma wymiaru 320x200 px, przy 256 kolorach. W tym trybie pamięć ekranu zawierająca 64000B, umieszczona jest w pamięci począwszy od adresu fizycznego **0A0000h**; kolejny bajty w tym obszarze opisują kolory pikseli wg standardowej palety VGA (paletę można zmieniać).

arytmetycznym.

6. (3 pkt.) Omówić sposób odwzorowania zawartości pamięci ekranu (w trybie tekstowym) na postać tekstu wyświetlanego na ekranie.

Znaki wyświetlane na ekranie w trybie tekstowym są odwzorowaniem obszaru pamięci od określonego adresu fizycznego (**0B8000h**). Każdy znak wyświetlany na ekranie opisany jest na dwóch bajtach : bajt parzysty zawiera kod ASCII znaku, a bajt nieparzysty atrybut, czyli sposób wyświetlania znaku (kolor tła,

kolor litery i ew. miganie). Pamięć ekranu składa się z 4000 bajtów czyli 80 kolumn po 25 linii.

generuje wyjątek niedozwolona operacja?

6. (3 pkt.) Podaj zasady przesyłania danych z klawiatury do komputera na poziomie przesłań sygnałów elektrycznych.

7. (2 pkt.) Znajdź dwie zmienionych cech, wybrać dokładnie

Mikrokontroler klawiatury przesyła kod wcisnięcia lub kod zwolnienia kod ten zostaje przesłany do układów płyty głównej komputera do sterownika klawiatury. Kod zwolnienia klawisza zawiera kod naciśnięcia poprzedzony bajtem F0H. Sygnały przesyłane z klawiatury docierają do układu na płycie głównej komputera (mikrokontroler, np. 8042 lub 8741 lub 8742), który odtwarza oryginalny kod pozycji (w przypadku zwolnienia klawisza: kod pozycji +128) — odtworzony kod pozycji zostaje udostępniony w porcie 60H.

8. (3 pkt.) N- etapowy potok rozkazów może potencjalnie zwiększyć przepustowość n-krotnie. Dlaczego rzadko tak jest w praktyce?

9. (2 pkt.) Porównaj tradycyjny model hierarchii układów pamięci

Programy rzadko przystosowane są do takie przetwarzania. Każda pętla warunkowa powoduje nieoczekiwana zmianę kolejności wykonywania rozkazów, przez co potok musi być wyczyszczony i ładowany od początku (straty wydajności można zmniejszyć np. poprzez tablice historii skoków). Często wynik jednej operacji jest argumentem drugiej, przez co występują przestoje w wykonywaniu programu. Poszczególne instrukcje znajdujące się w potokach mogą powodować między sobą konflikty dostępu do pamięci, przez co trzeba zaczekać na zakończenie dostępu przez jedną instrukcję aby inna mogła odczytać i zapisać dane. Oprócz tego w systemie często występują przerwania sprzętowe, a każdorazowe wywołanie podprogramu obsługi przerwania wymusza wyczyszczenie potoku, a po zakończeniu przerwania trzeba by wypełnić go od nowa. Wszystkie te czynniki przy rosnącej ilości etapów potoku nawarstwiają różnicę się coraz bardziej, spowalniając i niwelując korzyści z równoległego przetwarzania instrukcji.

praktyce?

9. (2 pkt.) W tradycyjnym modelu hierarchii układów pamięci pamięć podręczna L2 jest pamięć off-chip. Ten stan rzeczy w nowoczesnych architekturach uległ zmianie i obecnie L2 jest pamięcią on-chip. Wyjaśnij przyczynę tego zjawiska. Na czym polega zatem różnica między pamięcią L1 i L2?

10. (2 pkt.) Porównaj procesory RISC i CISC pod kątem

przychodzących kosztów produkcji i konserwacji.

Która maszyna jest szybsza i dlaczego?

9. (2 pkt.) Podaj definicję czasu dostępu do pamięci.

Czas dostępu (access time) - mierzony od momentu kiedy procesor zażąda odczytu komórki do momentu kiedy dane będą dostępne

polega zatem różnica mówiąca o mniejszej ilości mikrooperacji.

10. (2 pkt.) Porównaj procesory RISC i CISC pod kątem minimalizacji odwołań do pamięci operacyjnej.

W tym celu wporawdza się w mikroarchitekturach

Procesory RISCowe dużo rzadziej odwołują się do pamięci niż procesory CISCome. Mają one więcej rejestrów ogólnego przeznaczenia i to na nich wykonują większość rozkazów. W procesorach tego typu do pamięci mają dostęp tylko dwa rozkazy - **load** i **store**. W przeciwnieństwie do procesorów CISCome, w których zdecydowanie więcej rozkazów operuje na pamięci, np. **mov**, **add**, itd.

systemach wieloprocesorowych. Przedstaw jedną zaletę i jedną wadę każdej z organizacji.

10. (2 pkt.) Czy czas trwania cyku zegarowego wpływa na przepustowość pamięci SDRAM? Odpowiedź uzasadnij.

9. (2 pkt.) Podaj definicję czasu dostępu do pamięci.

10 (2 pkt.) Podaj i objasnij podstawową różnicę w działaniu pamięci SDRAM a DRAM.

11. (2 pkt.) Omów główne zadania jednostki sterującej w

Tak, ponieważ SDRAM jest pamięcią synchroniczną (Synchronous Dynamic RAM). SDRAM dlatego, że jest pamięcią synchroniczną musi być odświeżana zgodnie z cyklem zegara gdy nie jest aktualnie używana. Pamięć DRAM jest odświeżana w dowolnym momencie "zatrzymując" na chwilę pracę procesora, przez co czas procesora jest marnowany.

pamięci SDRAM a DRAM.

11. (2 pkt.) Omów główne zadania jednostki sterującej w procesorze biorąc pod uwagę informacje dostarczane na jej wejście i informacje generowane na wyjściu.

Wykonywanie rozkazu przez procesor rozpoczyna się pobraniem rozkazu z pamięci, po czym identyfikowany jest jego kod - na tej podstawie jednostka sterująca w procesorze wydaje odpowiednie dyspozycje dla jednostki arytmetyczno-logicznej, rejestrów i innych podzespołów procesora, kierując odpowiednio przepływem i przetwarzaniem danych. Na wyjściu są mikrorozkazy do innych układów procesora jakie rozkazy mają wykonać.

zmianę nazwy rejestrów (register renaming).

12. (3 pkt.) Wymień hipotetyczny ciąg mikrooperacji dla instrukcji POP.

- a. Pobranie zawartości rejestru ESP na magistralę adresową,
- b. wysłanie żądania odczytu z pamięci,
- c. zapisanie zawartości magistrali danych do rejestru wskazanego jako operand rozkazu **pop**,
- d. przesłanie zawartości rejestru ESP na magistralę danych,
- e. przesłanie zawartości magistralny danych do sumatora i zwiększenie wartości sumatora o 4,
- f. przesłanie zawartości sumatora na magistralę danych i następnie do rejestru ESP.

klasztowym?

12. (3 pkt.) Wymień hipotetyczny ciąg mikrooperacji dla instrukcji PUSH.

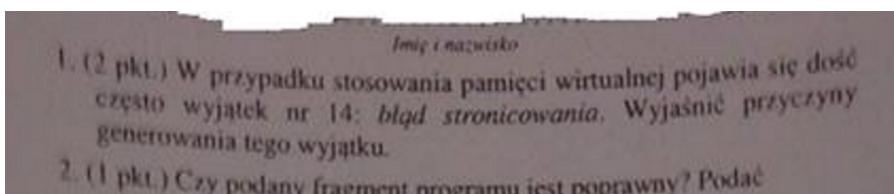
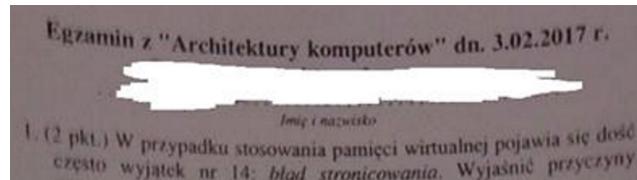
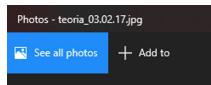
**Zapisanie do pamięci wartości z rejestru, odjęcie 4 od ESP.**

- a. Pobranie zawartości rejestru źródłowego na magistralę adresową,
- b. wysłanie żądania zapisu do pamięci,
- c. zapisanie zawartości magistrali danych do pamięci,
- d. przesłanie zawartości rejestru ESP na magistralę danych,
- e. przesłanie zawartości magistrali danych do sumatora i zwiększenie zawartości sumatora o (-4),
- f. przesłanie zawartości sumatora na magistralę danych i następnie do rejestru ESP.

wejście i informacje generowane dla wyjścia  
12. (3 pkt.) Wymień hipotetyczny ciąg mikrooperacji dla instrukcji  
LOOP.

Zmniejszenie o 1 ecx, skok gdy jnz do etykiety.

- A) Pobranie zawartości rejestru podanego przez operand
- roku dec do magistrali danych
- B) Wysłanie zawartości do sumatora
- C) Zwiększenie wartości sumatora o (-1)
- D) Wynik znajduje się na magistrali adresowej więc należy go wysłać z powrotem do rejestru podanego przez operand
- roku inc
- E) Porównanie wartości rejestru z 0
- F) Skok, gdy różny od 0



generowany, jeśli odwołanie dotyczy strony aktualnie nieobecnej w pamięci operacyjnej.

generowania tego wyjątku.  
2. (1 pkt.) Czy podany fragment programu jest poprawny? Podać uzasadnienie.  
PUSH 11223344h  
POP AL

Ten fragment kodu nie jest poprawny bo nie można użyć rejestru 8 bitowego do

instrukcji pop.

POP AL

3. (2 pkt.) W programie wystąpił wyjątek nadmiaru przy dzieleniu. Czy istnieje możliwość kontynuacji tego programu?

Nie, tzn można samemu sobie napisać obsługę wyjątku, że np. wyświetla się kwiatak podczas takiego dzielenia a nie wywala programu, ale to ciężkie

4. (2 pkt.) Określić możliwość realizacji 4-kanalowej pamięci podręcznej o odwzorowaniu tylko asocjacyjnym.

5. (2 pkt.) W programie wystąpił wyjątek nadmiaru przy dzieleniu. Czy istnieje możliwość kontynuacji tego programu?

5. (1 pkt.) Zadaniem poniższej sekwencji rozkazów jest wpisanie liczby -1 na wierzcholek stosu rejestrów koprocessora arytmetycznego. Uzupełnić brakujący rozkaz.

push dword PTR -1

- - - - -

add esp, 4

6. (2 pkt.) Poniżej podano sekwencję dwóch rozkazów. Napisać

push dword ptr [esp]

add esp, 4

6. (2 pkt.) Poniżej podano sekwencję dwóch rozkazów. Napisać równoważny ciąg rozkazów, w którym nie wystąpi rozkaz sete.

cmp bx, si

sete al

7. (2 pkt.) W jakim celu w najnowszych generacjach procesorów Core

cmp bx, si

jne dalej

mov al, 1

dalej:

7. (2 pkt.) W jakim celu w najnowszych generacjach procesorów Core wprowadzono rodzinę rozkazów ADX lub BMI2?

wprowadzono rodzinę rozkazów ADX lub BMI2.

8. (3 pkt.) Przedstaw równoważny fragment kodu bez użycia instrukcji operacji blokowych

mov ecx, 120h

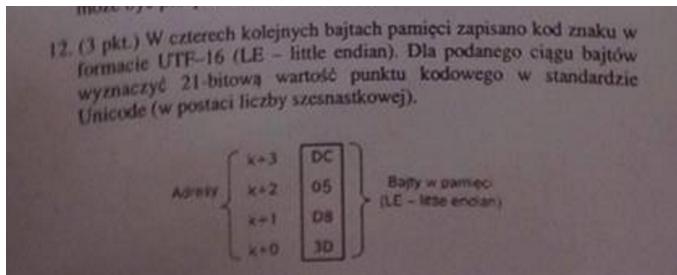
std

repne movsb

9. (2 pkt.) Który z niżej wymienionych sposobów adresowania jest

mie wykona się

repne movsb  
 9. (2 pkt.) Który z niżej wymienionych sposobów adresowania jest niedostępny w architekturze x86 w trybie 32-bitowym – adresowanie natychmiastowe, bezpośrednie, rejestrowe, indeksowe, pośrednie dla danych, pośrednie dla skoków, z autoinkrementacją i autodekrementacją.  
 10. (2 pkt.) W jakim celu wprowadza się w mikroarchitekturach zmianę



110110 xxxxxxxxxx | 110111 yyyy/yyyy

$$(1F408)_{16} - (10000)_{16} = (0F408)_{16} = \\ = (0000111101\ 000001000)_2$$

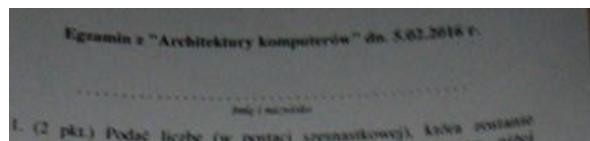
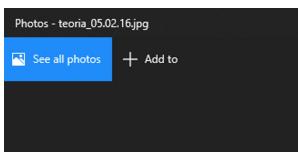
DC 05 D8 3D

1101 10po 0011 1101 1101 1100 0000 0101

0000 1111 0100 0000 0101

OF408h

1F408h



Imp i morskie

1. (2 pkt.) Podać liczbę (w postaci szesnastkowej), która powstanie wprowadzona do rejestru DI po wykonaniu podanego niskiego fragmentu programu:

```
linie dw 242, 424, 37, -3, 1223, 8, 0
      - 61 23 15 67 83 -
      mov es1, OFFSET linie + 6
      mov ebx, 0
      mov di, [esi][ebx]
```

2. (1 pkt.) Jaki wpływ na wykonywanie niektórych rozkazów ma

|           |
|-----------|
| 0000 6011 |
| 1111 1111 |
| 1111 1111 |
| 1111 1101 |

-3.

0000 0011  
 1111 1101  
 0 0 1 1  
 1014 812 256 118 64 32 16 8 4 2 1

0000 0011

-3

mov di, [esi][ebx]  
 2. (1 pkt.) Jaki wpływ na wykonywanie niektórych rozkazów ma przedrostek (dodatkowy bajt) REP (kod F3H)?  
 3. (2 pkt.) Wyjaśnij na przykładzie zastosowanie rozkazu ADC w

PRZYROSTEK TEN WYSTĘPUJE PRZY INSTRUKCJACH BLOKOWYCH. POLEGA ON NA POWTÓRZENIU ROZKAZU ECX RAZY.

przedrostek (dodatkowy bajt) REP (kod F3H)  
 3. (2 pkt.) Wyjaśnij na przykładzie zastosowanie rozkazu ADC w operacjach dodawania liczb w podwójnej precyzyji.  
 4. (1 pkt.) Wyjaśnij na przykładzie zastosowanie rozkazu ADC w

dodawanie 1 do edx : eax

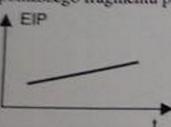
add eax, 1  
 adc edx, 0

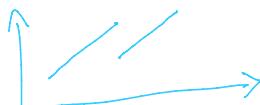
operacjach dodawania liczb w podwójnej precyzyji.  
 4. (1 pkt.) W jaki sposób procesor, w trakcie wykonywania programu, odróżnia rozkazy wykonujące działania na operandach 16- i 32-bitowych?

66h

5. (3 pkt.) W trakcie obserwacji działania fragmentu programu, w którym nie występują rozkazy skoku, zauważono, że zmiany wskaźnika instrukcji (licznika rozkazów) EIP przebiegają w sposób pokazany na rysunku. Naszkicować analogiczny wykres dla poniższego fragmentu programu:

▲ EIP

|                                                                                     |                                                                |
|-------------------------------------------------------------------------------------|----------------------------------------------------------------|
|  | ptl: mov ecx, 1<br>add bh, [esi]<br>dec esi<br>loop ptl<br>nop |
|-------------------------------------------------------------------------------------|----------------------------------------------------------------|



6. (2 pkt.) Wyjaśnij, dlaczego asembler traktuje rozkaz add [esi+edi], 1 jako błędny?  
 7. (1 pkt.) Czym różni się instrukcje koprocessora arytmetycznego: FST

nic wiadomo jaki rozmiar trzeba ptr

add [esi+edi], 1 jako błędny?  
 7. (1 pkt.) Czym różni się instrukcje koprocessora arytmetycznego: FST i FIST?  
 8. (3 pkt.) Poniższy rysunek przedstawia pojedyncza linię z pamięci

FIST - int      FST - float

8. (3 pkt.) Poniższy rysunek prezentuje pojedynczą linię z pamięci podrzędnej 1-kanalowej o adresowaniu bezpośrednim z systemu o adresowaniu 32-bitowym. Podać liczbę linii tej pamięci podrzędnej. Szary obszar na rysunku reprezentuje 12-bitowe pole etykiet. Liczby pokazane na rysunku są przykładowymi zawartościami bajtów pamięci w zapisie szesnastkowym.

01 02 03 04 05 06 07 08 01 02 03 04 05 06 07 08

$2^4 - \text{OFFSET}$   
4 bits

$$20 - 4 = 16$$

$2^{16}$  - liczba linii

64 kB - rozmiar linii

9. (2 pkt.) Zakładając, że pamięci DDR2 i DDR3 są taktowane z taką samą częstotliwością i mają identyczne czasy opóźnień CL, wskazać która z nich oferuje większą przepustowość pamięci (bandwidth). Odpowiedź uzasadnić.

tego nic michisny

Odpowiedź uzasadnić.

10. (2 pkt.) Wymienić obszary zastosowań architektury harwardzkiej w procesorach Intel Core.

11. (3 pkt.) Omówić technikę statycznego przewidywania wykonania rozgałęzień w procesorze.

w LA oddzielne mu domki i rozgałęzy

11. (3 pkt.) Omówić technikę statycznego przewidywania wykonania rozgałęzień w procesorze.

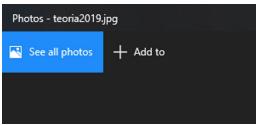
Statyczne - na sztawno określa się te rozgałęzienia

Skłoczy, a te mle do 75% ochnie

rozgałęzień w procesorze.

12. (3 pkt.) Dlaczego w niektórych rzeczywistych systemach komputerowych następuje spadek wydajności przy rozbudowie systemu o nowe jednostki przetwarzające?

Przy wielu jednostkach przetwarzających mogą następować konflikty przez co wykonywanie może być dłuższe niż przy mniejszej ilości jednostek przetwarzających.



a 1. (1 pkt.) Wyjaśni dlaczego assembler traktuje rozkaz  
or [8\*esi],-3  
jako błędny?

(1 pkt.) Wyjaśni dlaczego assembler traktuje rozkaz  
and [2\*edi],-1  
jako błędny?

Potwierdza ptr, bo nie wiadomo jak dalej się to daleć.

jako błędny:  
x (2 pkt.) Jakiej wartości należy spodziewać się w rejestrze ECX po wykonaniu następującego fragmentu kodu. Odpowiedź uzasadnij.  
mov ecx, 3003h  
egz: dec cl  
jnc egz

Nie skończona pętla, bo dec nie zmienia CF

\* 2. (2 pkt.) Przyjmując, że opodec operacji skoku warunkowego jest równy 73h, podaj kod rozkazu jae \$-8 (jako liczby szesnastkowe).

1010

0110

F3 F6h

x 3. (2 pkt.) Wymień hipotetyczny ciąg mikrooperacji dla instrukcji POP.

- Pobranie zawartości rejestru ESP na magistralę adresową,
- wysłanie żądania odczytu z pamięci,
- zapisanie zawartości magistrali danych do rejestru wskazanego jako operand rozkazu `pop`,
- przesłanie zawartości rejestru ESP na magistralę danych,
- przesłanie zawartości magistralny danych do sumatora i zwiększenie wartości sumatora o 4,
- przesłanie zawartości sumatora na magistralę danych i następnie do rejestru ESP.

x 3. (2 pkt.) Ponizej podano reprezentację binarną dwóch 32-bitowych liczb binarnych zmiennoprzecinkowych (format `float`) x i y. Ile wynosi iloczyn x\*y tych liczb? Wynik podać w postaci liczby dziesiętnej. Wskazówka: w formacie 32-bitowym wykładnik jest przesunięty o 127.

|   |   |          |                                  |
|---|---|----------|----------------------------------|
| X | 1 | 01000000 | 00000000000000000000000000000000 |
| Y | 1 | 11000000 | 00000000000000000000000000000000 |

$$s=1 \quad w=64 \quad Mw=63$$

$$s=1 \quad w=128+64 \quad w=192$$

$$1.0 \cdot 2^{65} \quad 1.0 \cdot 2^2 = 100$$

$$1.0 \cdot 2^{-63} \quad \text{wynik} = 4$$

5. (2 pkt.) W programie używany jest 8192-bajtowy ( $=2^{13}$ ) obszar pamięci. Jego początek wskazuje rejestr EDI. Napisać fragment programu w assemblerze 32-bitowym, który korzystając z rozkazu STOSW wypełni ten obszar znakami o punkcie kodowym U+01C2 w kodowaniu UTF-16.  
Wskazówka: rozkaz CLD zeruje znacznik DF, a rozkaz STD wpisuje 1 do tego znacznika.

110110 xxxxxxxxxx

110111 yyyyyyyyyy

$$(1F408)_{16} - (10000)_{16} = (0F408)_{16} = \\ = (0000111101\ 0000001000)_2$$

Mov ax, 01C2h

clD

Mov ECX, 8152

Nep STOSW

4. (3 pkt) Podaj fragment programu w assemblerze 32-bitowym, który zrealizuje te same działania (na poziomie instrukcji, a nie logicznym!) co poniższy kod w języku C.

```
int procedura ()
{
    return 0;
}

int (*wsk_funkcja)();

void main ()
{
    wsk_funkcja = procedura;
    wsk_funkcja();
}
```

5. (2 pkt) Wśród technik dynamicznego przydziału pamięci dla procesu wykorzystuje się alokacje zmiennych dynamicznych na stosie. W jaki sposób w programie assemblerowym decydujemy o przydzielanej wielkości stosu?

Wskazówka: Przydzielenie dla zmiennej skoku warunkowego jest równa

Pedydujemy w prologu zaraz po ustawieniu rejestrów namki stosu.

stosu?

6. (2 pkt.) Przyjmując, że opcode operacji skoku warunkowego jest równy 72h, podaj kod rozkazu jc \$-6 (jako liczby szesnastkowe).

7. (2 pkt.) Wymień hipotetyczny ciąg mikrooperacji dla instrukcji FCOMI.

1000  
0111

72 F8h

Wskazówka: rozkaz CLD zeruje znacznik DF, a rozkaz STD wypisuje 1 do tego znacznika.

6. (2 pkt.) Opisz rolę kontrolera klawiatury (np. 8042).

7. (3 pkt) Podaj fragment programu w assemblerze 32-bitowym, który dociera do niego kody kliknięcia i kody zwolnienia. Kontroler odtwarza kod pozycji i udostępnia go w porcie 60h

Docierają do niego kody kliknięcia i kody zwolnienia. Kontroler odtwarza kod pozycji i udostępnia go w porcie 60h

7. (2 pkt.) Wymień hipotetyczny ciąg mikrooperacji dla instrukcji FCOMI.

```
FCOM    ST(1)      ; porównanie ST(0) i ST(1)
FSTSW   AX         ; zapamiętanie rej. stanu koproc. w AX
SAHF    AH         ; przepisanie AH do rejestru znaczników
```

Począwszy od procesora Pentium Pro dostępny jest także rozkaz FCOMI, który wpisuje wynik porównania od razu do rejestru znaczników procesora. Stan znaczników procesora (ZF, PF, CF) po wykonaniu rozkazu FCOMI podano w poniższej tabeli. Warto porównać zawartość poniższej tabeli z opisem działania rozkazu CMP, który używany jest porównywania liczb stałoprzecinkowych (dw. 2, str. 12).

|                | ZF | PF | CF |
|----------------|----|----|----|
| ST(0)>x        | 0  | 0  | 0  |
| ST(0)<x        | 0  | 0  | 1  |
| ST(0)=x        | 1  | 0  | 0  |
| niedefiniowane | 1  | 1  | 1  |

7. (3 pkt.) Podaj fragment programu w asemblerze 32-bitowym, który zrealizuje te same działania (na poziomie instrukcji, a nie logicznym!) co poniższy kod w języku C.

```
char function ()
{
    return 0x30;
}

char (*fun_ptr)();

void main ()
{
    fun_ptr = function;
    fun_ptr();
}
```

8. (2 pkt.) W programie używane są dwa 4096-bajtowe obszary pamięci A i B, których początek wskazują, odpowiednio, rejesty ESI i EDI. Napisać fragment programu w asemblerze 32-bitowym, który korzystając z rozkazu CMPSB porówna zgodność obszarów A i B.

Jeśli obszary są identyczne znacznik CF należy ustawić na 1. W przeciwnym wypadku wartość CF ma być równa 0. Wskazówka: rozkaz CLD zeruje znacznik DF, a rozkaz STD wpisuje 1 do tego znacznika.

cl  
mov ecx, 4096  
repne cmpsb  
cmp ecx, 0  
jne dalej  
stc  
jmp koniec  
dalej: cl  
koniec

CLD zeruje znacznik DF, a rozkaz STD wpisuje 1 do tego znacznika.

(2 pkt.) Wyjaśnić zasady działania układów DMA w komputerze.

Układ DMA służy do przesyłania danych między pamięcią, a pamięcią lub urządzeniami I/O bez użycia procesora. Działa on w taki sposób, że sterownik DMA dostaje sygnał od procesora, aby rozpocząć przesyłanie. Po skończonym przesyle sygnalizuje w postaci przerwania sprzętowego. W ten sposób oszczędza się dużo czasu, ponieważ procesor w trakcie pracy DMA może wykonywać inne zadania.

9. (2 pkt.) Wyjaśnić zasady działania układów DMA w komputerze.

10. (2 pkt.) W jaki sposób długość cyklu zegarowego wpływa na przepustowość pamięci SDRAM? Odpowiedź uzasadnij.

W komputerze używana jest pamięć nadregiczna

Wpływa na pamięć SDRAM, ponieważ jest to synchroniczna pamięć. Musi być odświeżana zgodnie z cyklem zegara gdy nie jest aktualnie używana.

10. (2 pkt) W jaki sposób długość cyku zegarowego wpływa na przenośność?

Zadaniem jednostki sterującej procesora jest sterowanie pracą jednostek obliczeniowych. Na wejściu otrzymuje ona kod rozkazu, który musi odkodować zlokalizować argumenty itd. Na wyjściu przekazuje ciąg mikrooperacji zrozumiałych dla jednostek obliczeniowych. Jednostka układowa jest duży skomplikowany układ kombinacyjny.

Tutaj również generowane są mikroimpulsy do podzespołów procesora. Jest to duży i skomplikowany układ kombinacyjny, na którego wejście doprowadzane są sygnały z dekodera sygnałów, licznika sekwencji i rejestru stanu procesora. Licznik sekwencji w trakcie wykonywania jednego rozkazu przyjmuje wartości od zera do  $k$ . Do każdej wartości licznika sekwencji generowany jest inny impuls sterujący. Ten drugi sposób jest na ogół trochę szybszy, ale projektowanie jest bardziej skomplikowane i mniej podatne na zmiany projektowane.

11. (2 pkt) W jaki sposób długość cyku zegarowego wpływa na przepustowość pamięci DRAM (nie SDRAM!)? Odpowiedź uzasadnij.

Długość cyku zegara nie ma żadnego wpływu na przepustowość pamięci DRAM. Przepustowość jest niezmienna i zależy od czasu odświeżania, który jest narzucony/niezależny od zegara.

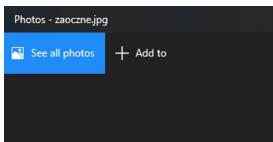
12. (2 pkt) Przedstaw budowę jednostki sterującej mikroprogramowej i opisz zasady jej działania.

Sterowanie mikroprogramowe, gdzie wykonanie każdej instrukcji mikroprogramu powoduje wywołanie jednego lub kilku impulsów do podzespołów procesora. Sterowanie zrealizowane jest za pomocą wewnętrznego mikroprocesora z własnym wskaźnikiem instrukcji, wykonującego mikroprogram. Taki program składa się z szeregu mikrorozkazów, odpowiedzialnych za pojedyncze operacje sterujące przemieszczaniem informacji między podzespołami i rejestrami procesora. Pozwala to na łatwe tworzenie nowych wersji procesorów o bardziej rozbudowanej liście rozkazów przy niezmienionych rozmiarach układów oraz na względnie łatwe usuwanie błędów.

12. (2 pkt) Jakiej wartości należy spodziewać się w rejestrze EDX po wykonaniu następującego fragmentu kodu. Odpowiedź uzasadnij.

|      |        |
|------|--------|
| ako: | inc dx |
|      | jc ako |

Nieokončená pętla, bo inc má změnu CF



Egzamin z „Architektury komputerów”  
dn. 30.01.2016 r.

1. Jaką rolę w trakcie obsługi przerwania sprzętowego pełni tablica deskryptorów przerwań?

Uzgadnia się tablicy deskryptorów aby dostarczyć się do podprogramów systemowych. W kolejnych wersjach systemy adresy ulegają zmianom dlatego aby programista nie musiał tego pamiętać używa się tablicy deskryptorów z stałymi indeksami.

2. Omówić dwa podstawowe sposoby kodowania liczb staloprzecinkowych ze znakiem.

$V_2$  i  $Z_mak - moduł$

Kod  $V_2$  jest to kod uzupełnieniowy do 2. Aby z kodu NKB uzyskać liczbę  $V_2$  ma minusie mniej zanegować tę liczbę i dodać 1. np.  $-5$

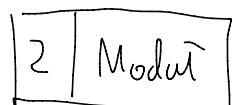
NKB.  $5 : 0101$

$V_2 : 1011 - 5$

$$\begin{array}{r} 1011 \\ -8 \\ \hline +2+1 \end{array}$$

Mniejsza część liczby na - reszta na +

$Z - M$  znak - moduł



$Z = 1$  to minus

$Z = 0$  to plus

moduł jest w NKB

## Moduł jest w NWD

3. Wyjaśnić w jaki sposób rozkazy sterujące (skoki) mogą (opcjonalnie) zmieniać naturalny porządek wykonywania programu. ✓

Wykonując rozkaz skoku możemy przeskoczyć kilka instrukcji albo np. się cofnąć.

4. Czy przetwarzanie potokowe zmniejsza czas wykonywania rozkazu? Odpowiedź uzasadnić. ✓

Przetwarzanie potokowe nie zmniejsza czasu wykonywania pojedyńczego rozkazu, natomiast może zmniejszyć czas wykonywania serii rozkazów z tego powodu, że mogą działać jak na "taśmie produkcyjnej", ponieważ mają te same elementy do wykonania może być wykonywane niezależnie kilka rozkazów.

5. W pewnym programie w assemblerze używana jest zmienna, która przyjmuje wartości całkowite z przedziału  $<-1200, +1200>$ . Jaki typ danych należy wybrać dla tej zmiennej? ✓

## word

7. Jaką rolę w trakcie wykonywania programu pełni wskaźnik instrukcji (licznik rozkazów)?
- |          |           |
|----------|-----------|
| 00000000 | 00430F76H |
| 00000001 | 00430F75H |
8. W pamięci operacyjnej pewnego komputera została zapisana liczba 256 w postaci 16-bitowej liczby binarnej (zob. rysunek). Określić, czy w komputerze stosowana jest konwencja mniejsze niżżej (ang. little endian) czy mniejsze wyżej (ang. big endian).

Po pobraniu rozkazu z pamięci i zdekomponowania rozkazu

EIP wskazuje adres kolejnej instrukcji do wykonania w programie,

Big endian

9. Dlaczego procesor w trakcie wykonywania rozkazu dodawania ADD ustawia jednocześnie dwa znaczniki nadmiaru CF i OF? ✓

żeby programista mógł to zinterpretować dla swoich celów tj. czy chce mu liczbach ze znakiem czy bez znaku bo ta operacja działa na obu typach liczb OF - z znakiem CF - bez znaku