**Lab 4**

**Obliczanie wyznacznika macierzy**

1. Należy stworzyć program obliczający **wyznacznik macierzy 4x4**.

2. Wejściem programu jest macierz 4x4 (podajemy 16 liczb, wiersz po wierszu), wyjściem programu ma być wyznacznik (liczba).

3. Całość należy wykonać korzystając z jednostki **SSE** (bazując na umiejętnościach z lab 4).

# **Algorytm**

Dana jest macierz M 4x4:

[a0 a1 a2 a3]  
[b0 b1 b2 b3]  
[c0 c1 c2 c3]  
[d0 d1 d2 d3]

Wyznacznik liczymy z rozwinięcia Laplace'a:

detM = a0\*b1\*c2\*d3 - a0\*b1\*c3\*d2 - a0\*b2\*c1\*d3 +  
 a0\*b2\*c3\*d1 + a0\*b3\*c1\*d2 - a0\*b3\*c2\*d1  
 - a1\*b0\*c2\*d3 + a1\*b0\*c3\*d2 + a1\*b2\*c0\*d3 -  
 a1\*b2\*c3\*d0 - a1\*b3\*c0\*d2 + a1\*b3\*c2\*d0  
 + a2\*b0\*c1\*d3 - a2\*b0\*c3\*d1 - a2\*b1\*c0\*d3 +  
 a2\*b1\*c3\*d0 + a2\*b3\*c0\*d1 - a2\*b3\*c1\*d0  
 - a3\*b0\*c1\*d2 + a3\*b0\*c2\*d1 + a3\*b1\*c0\*d2 -  
 a3\*b1\*c2\*d0 - a3\*b2\*c0\*d1 + a3\*b2\*c1\*d0

Po wyciągnięciu przed nawiasy otrzymujemy:

detM = (a0\*b1-a1\*b0)\*(c2\*d3-c3\*d2) +  
 (a2\*b0-a0\*b2)\*(c1\*d3-c3\*d1) +  
 (a0\*b3-a3\*b0)\*(c1\*d2-c2\*d1) +  
 (a1\*b2-a2\*b1)\*(c0\*d3-c3\*d0) +  
 (a2\*b3-a3\*b2)\*(c0\*d1-c1\*d0) +  
 (a3\*b1-a1\*b3)\*(c0\*d2-c2\*d0)

W wyjściowym wzorze należało wykonać 72 mnożenia i 23 dodawania, po uproszczeniu liczby te zredukowały się odpowiednio do 24 i 17.

# **Wektoryzacja**

Oto „fragment” powyższego równania, który doskonale nadaje się do wektoryzacji:

(a0\*b1-a1\*b0) [x0]  
(a2\*b0-a0\*b2) = [x1]  
(a0\*b3-a3\*b0) [x2]  
(a1\*b2-a2\*b1) [x3]  
 A' B' A”B”

Jeśli dane są wektory A = [*a*0, *a*1, *a*2, *a*3] oraz B = [*b*0, *b*1, *b*2, *b*3], to wyrażenie można opisać jako: X = A' ⋅ B' − A” ⋅ B”, gdzie *A'*, *B'*, itd. to wektory *A*, *B* z odpowiednio przestawionymi elementami. Ponieważ indeksy w *A'* i *B”* oraz *A”* i *B'* są takie same, wystarczą zatem dwa argumenty definiujące rozmieszczenie elementów.

macro.mac:

; makro oblicza wyrażenie wektorowe A'B'-A”B”  
  
; %1 - wektor A  
; %2 - wektor B  
; %3 - stała dla rozkazu shufps która wygeneruje wektory A' i B”  
; %4 - stała dla rozkazu shufps która wygeneruje wektory B' i A”  
; %5, %6 - rejestry robocze -- niszczone  
; %7 - rejestr wynikowy  
  
%macro calc 7  
 movaps %7, %1  
 movaps %6, %2  
  
 shufps %7, %3 ; A'  
 shufps %6, %4 ; B'  
 mulps %7, %6 ; %7 := A'B'  
  
 movaps %5, %1  
 movaps %6, %2  
  
 shufps %5, %4 ; A”  
 shufps %6, %3 ; B”  
 mulps %6, %5 ; %6 := A”B”  
  
 subps %7, %6 ; %7 := A'B' - A”B”  
%endmacro

Czynność opcjonalna:

Wektor A” można otrzymać z wektora A', przez co jeden z rozkazów movaps staje się zbędny.

Do generowania stałej dla rozkazu shufps można utworzyć makrodefinicję, dzięki czemu kod stanie się przejrzystszy; może ona wyglądać tak:

macro.mac:

; xmm = |d|c|b|a|  
%define sh(d,b,c,a) (((d&3) << 6) | ((c&3) << 4) | ((b&3) << 2) | ((a&3) << 0))

Żeby maksymalnie wykorzystać moc obliczeniową należy obliczać wyznaczniki dwóch macierzy jednocześnie. Ramką zostały otoczone wyrażenia obliczane przez makro calc.

I II  
 +-------------+ +-------------+  
detM1 = |(a0\*b1-a1\*b0)|\*|(c2\*d3-c3\*d2)| +  
 |(a2\*b0-a0\*b2)|\*|(c1\*d3-c3\*d1)| + => xmm3 4  
 |(a0\*b3-a3\*b0)|\*|(c1\*d2-c2\*d1)| +  
 |(a1\*b2-a2\*b1)|\*|(c0\*d3-c3\*d0)| +  
 +-------------+ +-------------+  
 III IV  
 +-------------+ +-------------+  
 |(a2\*b3-a3\*b2)|\*|(c0\*d1-c1\*d0)| +  
 |(a3\*b1-a1\*b3)|\*|(c0\*d2-c2\*d0)|  
 => xmm4 5  
detM2 = |(e2\*f3-e3\*f2)|\*|(g0\*h1-g1\*h0)| +  
 |(e3\*f1-e1\*f3)|\*|(g0\*h2-g2\*h0)| +  
 +-------------+ +-------------+  
 V VI  
 +-------------+ +-------------+  
 |(e0\*f1-e1\*f0)|\*|(g2\*h3-g3\*h2)| +  
 |(e2\*f0-e0\*f2)|\*|(g1\*h3-g3\*h1)| + => xmm5 7  
 |(e0\*f3-e3\*f0)|\*|(g1\*h2-g2\*h1)| +  
 |(e1\*f2-e2\*f1)|\*|(g0\*h3-g3\*h0)|  
 +-------------+ +-------------+

sse\_det4x4.asm:

; esi+0x00 -> a0 a1 a2 a3 = A  
; esi+0x10 -> b0 b1 b1 b3 = B  
; esi+0x20 -> c0 c1 c2 c3 = C  
; esi+0x30 -> d0 d1 d2 d3 = D  
  
; edi+0x00 -> e0 e1 e2 e3 = E  
; edi+0x10 -> f0 f1 f2 f3 = F  
; edi+0x20 -> g0 g1 g2 g3 = G  
; edi+0x30 -> h0 h1 h2 h3 = H  
  
; rejestry robocze: xmm6, xmm7  
  
; xmm3 = wektor I  
 movups xmm0, [esi+0x00] ; A  
 movups xmm1, [esi+0x10] ; B  
 calc xmm0, xmm1, sh(1,0,2,0), sh(2,3,0,1), xmm6, xmm7, xmm3  
  
; xmm4 = wektor II  
 movups xmm0, [esi+0x20] ; C  
 movups xmm1, [esi+0x30] ; D  
 calc xmm0, xmm1, sh(0,1,1,2), sh(3,2,3,3), xmm6, xmm7, xmm5  
  
; xmm3 \*= xmm4 (wektor I \* wektor II)  
 mulps xmm3, xmm4  
  
; xmm5 = wektor V  
 movups xmm0, [edi+0x00] ; E  
 movups xmm1, [edi+0x10] ; F  
 calc xmm0, xmm1, sh(1,0,2,0), sh(2,3,0,1), xmm6, xmm7, xmm5  
  
; xmm4 = wektor VI  
 movups xmm0, [edi+0x20] ; G  
 movups xmm1, [edi+0x30] ; H  
 calc xmm0, xmm1, sh(0,1,1,2), sh(3,2,3,3), xmm6, xmm7, xmm4  
  
; xmm5 \*= xmm4 (wektor V \* wektor VI)  
 mulps xmm3, xmm4  
  
; xmm2 = wektor III  
 movups xmm0, [esi+0x00] ; A-E  
 shufps xmm0, [edi+0x00], sh(3,2,3,2)  
 movups xmm1, [esi+0x10] ; B-F  
 shufps xmm1, [edi+0x10], sh(1,3,1,3)  
 calc xmm0, xmm1, sh(3,2,1,0), sh(3,2,1,0), xmm6, xmm7, xmm2  
  
; xmm4 = wektor IV  
 movups xmm0, [esi+0x20] ; C-G  
 shufps xmm0, [edi+0x20], sh(0,0,0,0)  
 movups xmm1, [esi+0x30] ; D-H  
 shufps xmm1, [edi+0x30], sh(2,1,2,1)  
 calc xmm0, xmm1, sh(3,2,1,0), sh(3,2,1,0), xmm6, xmm7, xmm4  
  
; xmm4 \*= xmm2 (wektor III \* wektor IV)  
 mulps xmm4, xmm2  
  
; teraz sytuacja przedstawia się następująco:  
;  
; | mm3 | mm4 | mm5 |  
; |M11 M12 M13 M14|M15 M16|M21 M22|M23 M24 M25 M26|  
;  
; detM1 = M11+...+M16, detM2 = M21+...+M26  
  
  
 movaps xmm0, xmm5  
  
 shufps xmm0, xmm3, sh(1,0,3,2) ; xmm0 = |M13 M14|M23 M24|  
 shufps xmm5, xmm3, sh(3,2,1,0) ; xmm5 = |M11 M12|M25 M26|  
 ; xmm4 = |M15 M16|M21 M22|  
  
 addps xmm0, xmm4  
 addps xmm0, xmm5 ; xmm0 = |M1' M1”|M2' M2”|  
  
 movaps xmm1, xmm0  
 shufps xmm1, xmm1, sh(2,3,0,1) ; xmm1 = |M1” M1' |M2” M2' |  
 addps xmm0, xmm1 ; xmm0 = |detM1 detM1|detM2 detM2|