

```

.nb302_unroll_loop:
    ;# quad-unroll innerloop here
    mov rdx, [rsp + nb302_innerjjnr] ;# pointer to jjnr[k]

    mov eax, [rdx]
    mov ebx, [rdx + 4]
    mov ecx, [rdx + 8]
    mov edx, [rdx + 12] ;# eax-edx=jnr1-4

    add qword ptr [rsp + nb302_innerjjnr], 16 ;# advance pointer (unrolled 4)

    mov rsi, [rbp + nb302_pos] ;# base of pos[]

    lea rax, [rax + rax*2] ;# replace jnr with j3
    lea rbx, [rbx + rbx*2]
    lea rcx, [rcx + rcx*2] ;# replace jnr with j3
    lea rdx, [rdx + rdx*2]

    ;# move j O coordinates to local temp variables
    movlps xmm0, [rsi + rax*4] ;# jxOa jyOa - -
    movlps xmm1, [rsi + rcx*4] ;# jxOc jyOc - -
    movhps xmm0, [rsi + rbx*4] ;# jxOa jyOa jxOb jyOb
    movhps xmm1, [rsi + rdx*4] ;# jxOc jyOc jxOd jyOd

    movss xmm2, [rsi + rax*4 + 8] ;# jzOa - - -
    movss xmm3, [rsi + rcx*4 + 8] ;# jzOc - - -
    movhps xmm2, [rsi + rbx*4 + 8] ;# jzOa - jzOb -
    movhps xmm3, [rsi + rdx*4 + 8] ;# jzOc - jzOd -

    movd mm0, eax ;# save j3 in mm0-mm3
    movd mm1, ebx
    movd mm2, ecx
    movd mm3, edx

    movaps xmm4, xmm0
    unpcklps xmm0, xmm1 ;# jxOa jxOc jyOa jyOc
    unpckhps xmm4, xmm1 ;# jxOb jxOd jyOb jyOd
    movaps xmm1, xmm0
    unpcklps xmm0, xmm4 ;# x
    unpckhps xmm1, xmm4 ;# y

    shufps xmm2, xmm3, 136 ;# 10001000 => jzOa jzOb jzOc jzOd

    ;# xmm0 = Ox
    ;# xmm1 = Oy
    ;# xmm2 = Oz

    movaps xmm3, xmm0
    movaps xmm4, xmm1
    movaps xmm5, xmm2
    movaps xmm6, xmm0
    movaps xmm7, xmm1

```

```
movaps xmm8, xmm2
```

```
subps xmm0, [rsp + nb302_ixO]  
subps xmm1, [rsp + nb302_izO]  
subps xmm2, [rsp + nb302_ixH1]  
subps xmm3, [rsp + nb302_izH1]  
subps xmm4, [rsp + nb302_ixH2]  
subps xmm5, [rsp + nb302_izH2]  
subps xmm6, [rsp + nb302_ixH2]  
subps xmm7, [rsp + nb302_izH2]  
subps xmm8, [rsp + nb302_izH2]
```

```
movaps [rsp + nb302_dxOO], xmm0  
movaps [rsp + nb302_dyOO], xmm1  
movaps [rsp + nb302_dzOO], xmm2  
mulps xmm0, xmm0  
mulps xmm1, xmm1  
mulps xmm2, xmm2  
movaps [rsp + nb302_dxH1O], xmm3  
movaps [rsp + nb302_dyH1O], xmm4  
movaps [rsp + nb302_dzH1O], xmm5  
mulps xmm3, xmm3  
mulps xmm4, xmm4  
mulps xmm5, xmm5  
movaps [rsp + nb302_dxH2O], xmm6  
movaps [rsp + nb302_dyH2O], xmm7  
movaps [rsp + nb302_dzH2O], xmm8  
mulps xmm6, xmm6  
mulps xmm7, xmm7  
mulps xmm8, xmm8  
addps xmm0, xmm1  
addps xmm0, xmm2  
addps xmm3, xmm4  
addps xmm3, xmm5  
addps xmm6, xmm7  
addps xmm6, xmm8
```

```
    ;# start doing invsqrt for jO atoms  
    rsqrtps xmm1, xmm0  
    rsqrtps xmm4, xmm3  
rsqrtps xmm7, xmm6
```

```
movaps xmm2, xmm1  
movaps xmm5, xmm4  
movaps xmm8, xmm7
```

```
mulps xmm1, xmm1 ;# lu*lu  
mulps xmm4, xmm4 ;# lu*lu  
mulps xmm7, xmm7 ;# lu*lu
```

```
movaps xmm9, [rsp + nb302_three]  
movaps xmm10, xmm9
```

```

movaps xmm11, xmm9

    mulps  xmm1, xmm0 ;# rsq*lu*lu
    mulps  xmm4, xmm3 ;# rsq*lu*lu
    mulps  xmm7, xmm6 ;# rsq*lu*lu

    subps  xmm9, xmm1
    subps  xmm10, xmm4
    subps  xmm11, xmm7 ;# 3-rsq*lu*lu

    mulps  xmm9, xmm2
    mulps  xmm10, xmm5
    mulps  xmm11, xmm8 ;# lu*(3-rsq*lu*lu)

    movaps xmm4, [rsp + nb302_half]
    mulps  xmm9, xmm4 ;# rinvOO
    mulps  xmm10, xmm4 ;# rinvH1O
    mulps  xmm11, xmm4 ;# rinvH2O

    movaps [rsp + nb302_rinvOO], xmm9
    movaps [rsp + nb302_rinvH1O], xmm10
    movaps [rsp + nb302_rinvH2O], xmm11

    ;# O interactions
    ;# rsq in xmm0,xmm3,xmm6
    ;# rinv in xmm9, xmm10, xmm11

    movaps xmm1, [rsp + nb302_tsc]
    mulps  xmm0, xmm9 ;# r
    mulps  xmm3, xmm10
    mulps  xmm6, xmm11
    mulps  xmm0, xmm1 ;# rtab
    mulps  xmm3, xmm1
    mulps  xmm6, xmm1

    ;# truncate and convert to integers
    cvttps2dq xmm1, xmm0
    cvttps2dq xmm4, xmm3
    cvttps2dq xmm7, xmm6

    ;# convert back to float
    cvtdq2ps xmm2, xmm1
    cvtdq2ps xmm5, xmm4
    cvtdq2ps xmm8, xmm7

    ;# multiply by 4
    pslld  xmm1, 2
    pslld  xmm4, 2
    pslld  xmm7, 2

    ;# move to integer registers
    movhps xmm13, xmm1

```

```

movhlps xmm14, xmm4
movhlps xmm15, xmm7
movd    eax, xmm1
movd    r8d, xmm4
movd    r12d, xmm7
movd    ecx, xmm13
movd    r10d, xmm14
movd    r14d, xmm15
pshufd  xmm1, xmm1, 1
pshufd  xmm4, xmm4, 1
pshufd  xmm7, xmm7, 1
pshufd  xmm13, xmm13, 1
pshufd  xmm14, xmm14, 1
pshufd  xmm15, xmm15, 1
movd    ebx, xmm1
movd    r9d, xmm4
movd    r13d, xmm7
movd    edx, xmm13
movd    r11d, xmm14
movd    r15d, xmm15

```

```

mov rsi, [rbp + nb302_VFtab]

```

```

;# calculate eps

```

```

subps   xmm0, xmm2
subps   xmm3, xmm5
subps   xmm6, xmm8

```

```

movaps   [rsp + nb302_epsO], xmm0
movaps   [rsp + nb302_epsH1], xmm3
movaps   [rsp + nb302_epsH2], xmm6

```

```

;# Load LOTS of table data

```

```

movlps   xmm1, [rsi + rax*4]
movlps   xmm5, [rsi + r8*4]
movlps   xmm9, [rsi + r12*4]

```

```

movlps   xmm3, [rsi + rcx*4]
movlps   xmm7, [rsi + r10*4]
movlps   xmm11, [rsi + r14*4]

```

```

movhps   xmm1, [rsi + rbx*4]
movhps   xmm5, [rsi + r9*4]
movhps   xmm9, [rsi + r13*4]

```

```

movhps   xmm3, [rsi + rdx*4]
movhps   xmm7, [rsi + r11*4]
movhps   xmm11, [rsi + r15*4]

```

```

movaps   xmm0, xmm1
movaps   xmm4, xmm5
movaps   xmm8, xmm9

```

```

shufps xmm0, xmm3, 136 ;# 10001000
shufps xmm4, xmm7, 136 ;# 10001000
shufps xmm8, xmm11, 136 ;# 10001000
shufps xmm1, xmm3, 221 ;# 11011101
shufps xmm5, xmm7, 221 ;# 11011101
shufps xmm9, xmm11, 221 ;# 11011101

```

```

movlps xmm3, [rsi + rax*4 + 8]
movlps xmm7, [rsi + r8*4 + 8]
movlps xmm11, [rsi + r12*4 + 8]

```

```

movlps xmm12, [rsi + rcx*4 + 8]
movlps xmm13, [rsi + r10*4 + 8]
movlps xmm14, [rsi + r14*4 + 8]

```

```

movhps xmm3, [rsi + rbx*4 + 8]
movhps xmm7, [rsi + r9*4 + 8]
movhps xmm11, [rsi + r13*4 + 8]

```

```

movhps xmm12, [rsi + rdx*4 + 8]
movhps xmm13, [rsi + r11*4 + 8]
movhps xmm14, [rsi + r15*4 + 8]

```

```

movaps xmm2, xmm3
movaps xmm6, xmm7
movaps xmm10, xmm11

```

```

shufps xmm2, xmm12, 136 ;# 10001000
shufps xmm6, xmm13, 136 ;# 10001000
shufps xmm10, xmm14, 136 ;# 10001000
shufps xmm3, xmm12, 221 ;# 11011101
shufps xmm7, xmm13, 221 ;# 11011101
shufps xmm11, xmm14, 221 ;# 11011101

```

;/# table data ready in xmm0-xmm3 , xmm4-xmm7 , and xmm8-xmm11

```

movaps xmm12, [rsp + nb302_epsO]
movaps xmm13, [rsp + nb302_epsH1]
movaps xmm14, [rsp + nb302_epsH2]

```

```

mulps xmm3, xmm12 ;# Heps
mulps xmm7, xmm13
mulps xmm11, xmm14
mulps xmm2, xmm12 ;# Geps
mulps xmm6, xmm13
mulps xmm10, xmm14
mulps xmm3, xmm12 ;# Heps2
mulps xmm7, xmm13
mulps xmm11, xmm14

```

```

addps xmm1, xmm2 ;# F+Geps
addps xmm5, xmm6
addps xmm9, xmm10

```

```

addps xmm1, xmm3 ;# F+Geps+Heps2 = Fp
addps xmm5, xmm7
addps xmm9, xmm11
addps xmm3, xmm3 ;# 2*Heps2
addps xmm7, xmm7
addps xmm11, xmm11
addps xmm3, xmm2 ;# 2*Heps2+Geps
addps xmm7, xmm6
addps xmm11, xmm10
addps xmm3, xmm1 ;# FF = Fp + 2*Heps2 + Geps
addps xmm7, xmm5
addps xmm11, xmm9
mulps xmm1, xmm12 ;# eps*Fp
mulps xmm5, xmm13
mulps xmm9, xmm14
movaps xmm12, [rsp + nb302_qqOO]
movaps xmm13, [rsp + nb302_qqOH]
addps xmm1, xmm0 ;# VV
addps xmm5, xmm4
addps xmm9, xmm8
mulps xmm1, xmm12 ;# VV*qq = vcoul
mulps xmm5, xmm13
mulps xmm9, xmm13
mulps xmm3, xmm12 ;# FF*qq = fij
mulps xmm7, xmm13
mulps xmm11, xmm13

;# accumulate vctot
addps xmm1, [rsp + nb302_vctot]
addps xmm5, xmm9
addps xmm1, xmm5
movaps [rsp + nb302_vctot], xmm1

movaps xmm10, [rsp + nb302_tsc]
mulps xmm3, xmm10 ;# fscal
mulps xmm7, xmm10
mulps xmm10, xmm11

movd eax, mm0 ;# restore j3 from mm0-mm3
movd ebx, mm1
movd ecx, mm2
movd edx, mm3

;# move j O forces to local temp variables
movlps xmm11, [rdi + rax*4] ;# jxOa jyOa - -
movlps xmm12, [rdi + rcx*4] ;# jxOc jyOc - -
movhps xmm11, [rdi + rbx*4] ;# jxOa jyOa jxOb jyOb
movhps xmm12, [rdi + rdx*4] ;# jxOc jyOc jxOd jyOd

movss xmm13, [rdi + rax*4 + 8] ;# jzOa - - -
movss xmm14, [rdi + rcx*4 + 8] ;# jzOc - - -
movhps xmm13, [rdi + rbx*4 + 8] ;# jzOa - jzOb -

```

```
movhps xmm14, [rdi + rdx*4 + 8] ;# jzOc - jzOd -
```

```
shufps xmm13, xmm14, 136 ;# 10001000 => jzOa jzOb jzOc jzOd
```

```
;/# xmm11: jxOa jyOa jxOb jyOb
```

```
;/# xmm12: jxOc jyOc jxOd jyOd
```

```
;/# xmm13: jzOa jzOb jzOc jzOd
```

```
xorps xmm0, xmm0
```

```
xorps xmm4, xmm4
```

```
xorps xmm8, xmm8
```

```
mulps xmm3, [rsp + nb302_rinvOO]
```

```
mulps xmm7, [rsp + nb302_rinvH1O]
```

```
mulps xmm10, [rsp + nb302_rinvH2O]
```

```
subps xmm0, xmm3
```

```
subps xmm4, xmm7
```

```
subps xmm8, xmm10
```

```
movaps xmm1, xmm0
```

```
movaps xmm2, xmm0
```

```
movaps xmm3, xmm4
```

```
movaps xmm5, xmm4
```

```
movaps xmm6, xmm8
```

```
movaps xmm7, xmm8
```

```
mulps xmm0, [rsp + nb302_dxOO]
```

```
mulps xmm1, [rsp + nb302_dyOO]
```

```
mulps xmm2, [rsp + nb302_dzOO]
```

```
mulps xmm3, [rsp + nb302_dxH1O]
```

```
mulps xmm4, [rsp + nb302_dyH1O]
```

```
mulps xmm5, [rsp + nb302_dzH1O]
```

```
mulps xmm6, [rsp + nb302_dxH2O]
```

```
mulps xmm7, [rsp + nb302_dyH2O]
```

```
mulps xmm8, [rsp + nb302_dzH2O]
```

```
movaps xmm14, xmm0
```

```
movaps xmm15, xmm1
```

```
addps xmm13, xmm2
```

```
addps xmm0, [rsp + nb302_fixO]
```

```
addps xmm1, [rsp + nb302_fiyO]
```

```
addps xmm2, [rsp + nb302_fizO]
```

```
addps xmm14, xmm3
```

```
addps xmm15, xmm4
```

```
addps xmm13, xmm5
```

```
addps xmm3, [rsp + nb302_fixH1]
```

```
addps xmm4, [rsp + nb302_fiyH1]
```

```
addps xmm5, [rsp + nb302_fizH1]
```

```
addps xmm14, xmm6
```

```

addps xmm15, xmm7
addps xmm13, xmm8
addps xmm6, [rsp + nb302_fixH2]
addps xmm7, [rsp + nb302_fiyH2]
addps xmm8, [rsp + nb302_fizH2]

```

```

movaps [rsp + nb302_fixO], xmm0
movaps [rsp + nb302_fiyO], xmm1
movaps [rsp + nb302_fizO], xmm2
movaps [rsp + nb302_fixH1], xmm3
movaps [rsp + nb302_fiyH1], xmm4
movaps [rsp + nb302_fizH1], xmm5
movaps [rsp + nb302_fixH2], xmm6
movaps [rsp + nb302_fiyH2], xmm7
movaps [rsp + nb302_fizH2], xmm8

```

```

;# xmm14 = fOx
;# xmm15 = fOy
;# xmm13 = fOz
movaps xmm0, xmm14
unpcklps xmm14, xmm15
unpckhps xmm0, xmm15

```

```

addps xmm11, xmm14
addps xmm12, xmm0

```

```

movhlps xmm14, xmm13 ;# fOzc fOzd

```

```

movlps [rdi + rax*4], xmm11
movhps [rdi + rbx*4], xmm11
movlps [rdi + rcx*4], xmm12
movhps [rdi + rdx*4], xmm12
movss [rdi + rax*4 + 8], xmm13
movss [rdi + rcx*4 + 8], xmm14
shufps xmm13, xmm13, 1
shufps xmm14, xmm14, 1
movss [rdi + rbx*4 + 8], xmm13
movss [rdi + rdx*4 + 8], xmm14

```

```

;# move j H1 coordinates to local temp variables
mov rsi, [rbp + nb302_pos]
movlps xmm0, [rsi + rax*4 + 12] ;# jxH1a jyH1a - -
movlps xmm1, [rsi + rcx*4 + 12] ;# jxH1c jyH1c - -
movhps xmm0, [rsi + rbx*4 + 12] ;# jxH1a jyH1a jxH1b jyH1b
movhps xmm1, [rsi + rdx*4 + 12] ;# jxH1c jyH1c jxH1d jyH1d

```

```

movss xmm2, [rsi + rax*4 + 20] ;# jzH1a - - -
movss xmm3, [rsi + rcx*4 + 20] ;# jzH1c - - -
movhps xmm2, [rsi + rbx*4 + 20] ;# jzH1a - jzH1b -
movhps xmm3, [rsi + rdx*4 + 20] ;# jzH1c - jzH1d -

```

```

movaps xmm4, xmm0

```



```
unpcklps xmm0, xmm1 ;# jxH1a jxH1c jyH1a jyH1c
unpckhps xmm4, xmm1 ;# jxH1b jxH1d jyH1b jyH1d
movaps xmm1, xmm0
unpcklps xmm0, xmm4 ;# x
unpckhps xmm1, xmm4 ;# y
```

```
shufps xmm2, xmm3, 136 ;# 10001000 => jzH1a jzH1b jzH1c jzH1d
```

```
;/# xmm0 = H1x
;/# xmm1 = H1y
;/# xmm2 = H1z
```

```
movaps xmm3, xmm0
movaps xmm4, xmm1
movaps xmm5, xmm2
movaps xmm6, xmm0
movaps xmm7, xmm1
movaps xmm8, xmm2
```

```
subps xmm0, [rsp + nb302_ixO]
subps xmm1, [rsp + nb302_iyO]
subps xmm2, [rsp + nb302_izO]
subps xmm3, [rsp + nb302_ixH1]
subps xmm4, [rsp + nb302_iyH1]
subps xmm5, [rsp + nb302_izH1]
subps xmm6, [rsp + nb302_ixH2]
subps xmm7, [rsp + nb302_iyH2]
subps xmm8, [rsp + nb302_izH2]
```

```
movaps [rsp + nb302_dxOH1], xmm0
movaps [rsp + nb302_dyOH1], xmm1
movaps [rsp + nb302_dzOH1], xmm2
mulps xmm0, xmm0
mulps xmm1, xmm1
mulps xmm2, xmm2
movaps [rsp + nb302_dxH1H1], xmm3
movaps [rsp + nb302_dyH1H1], xmm4
movaps [rsp + nb302_dzH1H1], xmm5
mulps xmm3, xmm3
mulps xmm4, xmm4
mulps xmm5, xmm5
movaps [rsp + nb302_dxH2H1], xmm6
movaps [rsp + nb302_dyH2H1], xmm7
movaps [rsp + nb302_dzH2H1], xmm8
mulps xmm6, xmm6
mulps xmm7, xmm7
mulps xmm8, xmm8
addps xmm0, xmm1
addps xmm0, xmm2
addps xmm3, xmm4
addps xmm3, xmm5
addps xmm6, xmm7
```

```

addps xmm6, xmm8

;# start doing invsqrt for jH1 atoms
rsqrtps xmm1, xmm0
rsqrtps xmm4, xmm3
rsqrtps xmm7, xmm6

movaps xmm2, xmm1
movaps xmm5, xmm4
movaps xmm8, xmm7

mulps xmm1, xmm1 ;# lu*lu
mulps xmm4, xmm4 ;# lu*lu
mulps xmm7, xmm7 ;# lu*lu

movaps xmm9, [rsp + nb302_three]
movaps xmm10, xmm9
movaps xmm11, xmm9

mulps xmm1, xmm0 ;# rsq*lu*lu
mulps xmm4, xmm3 ;# rsq*lu*lu
mulps xmm7, xmm6 ;# rsq*lu*lu

subps xmm9, xmm1
subps xmm10, xmm4
subps xmm11, xmm7 ;# 3-rsq*lu*lu

mulps xmm9, xmm2
mulps xmm10, xmm5
mulps xmm11, xmm8 ;# lu*(3-rsq*lu*lu)

movaps xmm4, [rsp + nb302_half]
mulps xmm9, xmm4 ;# rinvOH1
mulps xmm10, xmm4 ;# rinvH1H1
mulps xmm11, xmm4 ;# rinvH2H1

movaps [rsp + nb302_rinvOH1], xmm9
movaps [rsp + nb302_rinvH1H1], xmm10
movaps [rsp + nb302_rinvH2H1], xmm11

;# H1 interactions
;# rsq in xmm0,xmm3,xmm6
;# rinv in xmm9, xmm10, xmm11

movaps xmm1, [rsp + nb302_tsc]
mulps xmm0, xmm9 ;# r
mulps xmm3, xmm10
mulps xmm6, xmm11
mulps xmm0, xmm1 ;# rtab
mulps xmm3, xmm1
mulps xmm6, xmm1

```

```

mov rsi, [rbp + nb302_VFtab]

;# truncate and convert to integers
cvttps2dq xmm1, xmm0
cvttps2dq xmm4, xmm3
cvttps2dq xmm7, xmm6

;# convert back to float
cvtdq2ps xmm2, xmm1
cvtdq2ps xmm5, xmm4
cvtdq2ps xmm8, xmm7

;# multiply by 4
pslld xmm1, 2
pslld xmm4, 2
pslld xmm7, 2

;# move to integer registers
movhps xmm13, xmm1
movhps xmm14, xmm4
movhps xmm15, xmm7
movd eax, xmm1
movd r8d, xmm4
movd r12d, xmm7
movd ecx, xmm13
movd r10d, xmm14
movd r14d, xmm15
pshufd xmm1, xmm1, 1
pshufd xmm4, xmm4, 1
pshufd xmm7, xmm7, 1
pshufd xmm13, xmm13, 1
pshufd xmm14, xmm14, 1
pshufd xmm15, xmm15, 1
movd ebx, xmm1
movd r9d, xmm4
movd r13d, xmm7
movd edx, xmm13
movd r11d, xmm14
movd r15d, xmm15

;# calculate eps
subps xmm0, xmm2
subps xmm3, xmm5
subps xmm6, xmm8

movaps [rsp + nb302_epsO], xmm0
movaps [rsp + nb302_epsH1], xmm3
movaps [rsp + nb302_epsH2], xmm6

;# Load LOTS of table data
movlps xmm1, [rsi + rax*4]

```

```
movlps xmm5, [rsi + r8*4]
movlps xmm9, [rsi + r12*4]
```

```
movlps xmm3, [rsi + rcx*4]
movlps xmm7, [rsi + r10*4]
movlps xmm11, [rsi + r14*4]
```

```
movhps xmm1, [rsi + rbx*4]
movhps xmm5, [rsi + r9*4]
movhps xmm9, [rsi + r13*4]
```

```
movhps xmm3, [rsi + rdx*4]
movhps xmm7, [rsi + r11*4]
movhps xmm11, [rsi + r15*4]
```

```
movaps xmm0, xmm1
movaps xmm4, xmm5
movaps xmm8, xmm9
```

```
shufps xmm0, xmm3, 136 ;# 10001000
shufps xmm4, xmm7, 136 ;# 10001000
shufps xmm8, xmm11, 136 ;# 10001000
shufps xmm1, xmm3, 221 ;# 11011101
shufps xmm5, xmm7, 221 ;# 11011101
shufps xmm9, xmm11, 221 ;# 11011101
```

```
movlps xmm3, [rsi + rax*4 + 8]
movlps xmm7, [rsi + r8*4 + 8]
movlps xmm11, [rsi + r12*4 + 8]
```

```
movlps xmm12, [rsi + rcx*4 + 8]
movlps xmm13, [rsi + r10*4 + 8]
movlps xmm14, [rsi + r14*4 + 8]
```

```
movhps xmm3, [rsi + rbx*4 + 8]
movhps xmm7, [rsi + r9*4 + 8]
movhps xmm11, [rsi + r13*4 + 8]
```

```
movhps xmm12, [rsi + rdx*4 + 8]
movhps xmm13, [rsi + r11*4 + 8]
movhps xmm14, [rsi + r15*4 + 8]
```

```
movaps xmm2, xmm3
movaps xmm6, xmm7
movaps xmm10, xmm11
```

```
shufps xmm2, xmm12, 136 ;# 10001000
shufps xmm6, xmm13, 136 ;# 10001000
shufps xmm10, xmm14, 136 ;# 10001000
shufps xmm3, xmm12, 221 ;# 11011101
shufps xmm7, xmm13, 221 ;# 11011101
shufps xmm11, xmm14, 221 ;# 11011101
```

```
 ;# table data ready in xmm0-xmm3 , xmm4-xmm7 , and xmm8-xmm11
```

```

movaps xmm12, [rsp + nb302_epsO]
movaps xmm13, [rsp + nb302_epsH1]
movaps xmm14, [rsp + nb302_epsH2]

mulps  xmm3, xmm12  ;# Heps
mulps  xmm7, xmm13
mulps  xmm11, xmm14
mulps  xmm2, xmm12  ;# Geps
mulps  xmm6, xmm13
mulps  xmm10, xmm14
mulps  xmm3, xmm12  ;# Heps2
mulps  xmm7, xmm13
mulps  xmm11, xmm14

addps  xmm1, xmm2  ;# F+Geps
addps  xmm5, xmm6
addps  xmm9, xmm10
addps  xmm1, xmm3  ;# F+Geps+Heps2 = Fp
addps  xmm5, xmm7
addps  xmm9, xmm11
addps  xmm3, xmm3  ;# 2*Heps2
addps  xmm7, xmm7
addps  xmm11, xmm11
addps  xmm3, xmm2  ;# 2*Heps2+Geps
addps  xmm7, xmm6
addps  xmm11, xmm10
addps  xmm3, xmm1  ;# FF = Fp + 2*Heps2 + Geps
addps  xmm7, xmm5
addps  xmm11, xmm9
mulps  xmm1, xmm12  ;# eps*Fp
mulps  xmm5, xmm13
mulps  xmm9, xmm14
movaps xmm12, [rsp + nb302_qqOH]
movaps xmm13, [rsp + nb302_qqHH]
addps  xmm1, xmm0  ;# VV
addps  xmm5, xmm4
addps  xmm9, xmm8
mulps  xmm1, xmm12  ;# VV*qq = vcoul
mulps  xmm5, xmm13
mulps  xmm9, xmm13
mulps  xmm3, xmm12  ;# FF*qq = fij
mulps  xmm7, xmm13
mulps  xmm11, xmm13

;# accumulate vctot
addps  xmm1, [rsp + nb302_vctot]
addps  xmm5, xmm9
addps  xmm1, xmm5
movaps [rsp + nb302_vctot], xmm1

movaps xmm10, [rsp + nb302_tsc]

```

```
mulps xmm3, xmm10 ;# fscal
mulps xmm7, xmm10
mulps xmm10, xmm11
```

```
movd eax, mm0 ;# restore j3 from mm0-mm3
movd ebx, mm1
movd ecx, mm2
movd edx, mm3
```

```
    ;# move j H1 forces to local temp variables
movlps xmm11, [rdi + rax*4 + 12] ;# jxH1a jyH1a - -
movlps xmm12, [rdi + rcx*4 + 12] ;# jxH1c jyH1c - -
movhps xmm11, [rdi + rbx*4 + 12] ;# jxH1a jyH1a jxH1b jyH1b
movhps xmm12, [rdi + rdx*4 + 12] ;# jxH1c jyH1c jxH1d jyH1d
```

```
movss xmm13, [rdi + rax*4 + 20] ;# jzH1a - - -
movss xmm14, [rdi + rcx*4 + 20] ;# jzH1c - - -
movhps xmm13, [rdi + rbx*4 + 20] ;# jzH1a - jzH1b -
movhps xmm14, [rdi + rdx*4 + 20] ;# jzH1c - jzH1d -
```

```
shufps xmm13, xmm14, 136 ;# 10001000 => jzH1a jzH1b jzH1c jzH1d
```

```
    ;# xmm11: jxH1a jyH1a jxH1b jyH1b
    ;# xmm12: jxH1c jyH1c jxH1d jyH1d
    ;# xmm13: jzH1a jzH1b jzH1c jzH1d
```

```
xorps xmm0, xmm0
xorps xmm4, xmm4
xorps xmm8, xmm8
```

```
mulps xmm3, [rsp + nb302_rinvOH1]
mulps xmm7, [rsp + nb302_rinvH1H1]
mulps xmm10, [rsp + nb302_rinvH2H1]
```

```
subps xmm0, xmm3
subps xmm4, xmm7
subps xmm8, xmm10
```

```
movaps xmm1, xmm0
movaps xmm2, xmm0
movaps xmm3, xmm4
movaps xmm5, xmm4
movaps xmm6, xmm8
movaps xmm7, xmm8
```

```
    mulps xmm0, [rsp + nb302_dxOH1]
    mulps xmm1, [rsp + nb302_dyOH1]
    mulps xmm2, [rsp + nb302_dzOH1]
    mulps xmm3, [rsp + nb302_dxH1H1]
    mulps xmm4, [rsp + nb302_dyH1H1]
    mulps xmm5, [rsp + nb302_dzH1H1]
    mulps xmm6, [rsp + nb302_dxH2H1]
```

```
    mulps xmm7, [rsp + nb302_dyH2H1]
    mulps xmm8, [rsp + nb302_dzH2H1]
```

```
movaps xmm14, xmm0
movaps xmm15, xmm1
addps xmm13, xmm2
addps xmm0, [rsp + nb302_fixO]
addps xmm1, [rsp + nb302_fiyO]
addps xmm2, [rsp + nb302_fizO]
```

```
addps xmm14, xmm3
addps xmm15, xmm4
addps xmm13, xmm5
addps xmm3, [rsp + nb302_fixH1]
addps xmm4, [rsp + nb302_fiyH1]
addps xmm5, [rsp + nb302_fizH1]
```

```
addps xmm14, xmm6
addps xmm15, xmm7
addps xmm13, xmm8
addps xmm6, [rsp + nb302_fixH2]
addps xmm7, [rsp + nb302_fiyH2]
addps xmm8, [rsp + nb302_fizH2]
```

```
movaps [rsp + nb302_fixO], xmm0
movaps [rsp + nb302_fiyO], xmm1
movaps [rsp + nb302_fizO], xmm2
movaps [rsp + nb302_fixH1], xmm3
movaps [rsp + nb302_fiyH1], xmm4
movaps [rsp + nb302_fizH1], xmm5
movaps [rsp + nb302_fixH2], xmm6
movaps [rsp + nb302_fiyH2], xmm7
movaps [rsp + nb302_fizH2], xmm8
```

```
;/# xmm14 = fH1x
;/# xmm15 = fH1y
;/# xmm13 = fH1z
movaps xmm0, xmm14
unpcklps xmm14, xmm15
unpckhps xmm0, xmm15
```

```
addps xmm11, xmm14
addps xmm12, xmm0
```

```
movhlps xmm14, xmm13 ;/# fH1zc fH1zd
```

```
movlps [rdi + rax*4 + 12], xmm11
movhps [rdi + rbx*4 + 12], xmm11
movlps [rdi + rcx*4 + 12], xmm12
movhps [rdi + rdx*4 + 12], xmm12
movss [rdi + rax*4 + 20], xmm13
movss [rdi + rcx*4 + 20], xmm14
```

```

shufps xmm13, xmm13, 1
shufps xmm14, xmm14, 1
movss [rdi + rbx*4 + 20], xmm13
movss [rdi + rdx*4 + 20], xmm14

    mov rsi, [rbp + nb302_pos]
    ;# move j H2 coordinates to local temp variables
movlps xmm0, [rsi + rax*4 + 24] ;# jxH2a jyH2a - -
movlps xmm1, [rsi + rcx*4 + 24] ;# jxH2c jyH2c - -
movhps xmm0, [rsi + rbx*4 + 24] ;# jxH2a jyH2a jxH2b jyH2b
movhps xmm1, [rsi + rdx*4 + 24] ;# jxH2c jyH2c jxH2d jyH2d

movss xmm2, [rsi + rax*4 + 32] ;# jzH2a - - -
movss xmm3, [rsi + rcx*4 + 32] ;# jzH2c - - -
movss xmm5, [rsi + rbx*4 + 32] ;# jzOb - - -
movss xmm6, [rsi + rdx*4 + 32] ;# jzOd - - -
movlhps xmm2, xmm5 ;# jzOa - jzOb -
movlhps xmm3, xmm6 ;# jzOc - jzOd -

movaps xmm4, xmm0
unpcklps xmm0, xmm1 ;# jxH2a jxH2c jyH2a jyH2c
unpckhps xmm4, xmm1 ;# jxH2b jxH2d jyH2b jyH2d
movaps xmm1, xmm0
unpcklps xmm0, xmm4 ;# x
unpckhps xmm1, xmm4 ;# y

shufps xmm2, xmm3, 136 ;# 10001000 => jzH2a jzH2b jzH2c jzH2d

;# xmm0 = H2x
;# xmm1 = H2y
;# xmm2 = H2z

movaps xmm3, xmm0
movaps xmm4, xmm1
movaps xmm5, xmm2
movaps xmm6, xmm0
movaps xmm7, xmm1
movaps xmm8, xmm2

subps xmm0, [rsp + nb302_ixO]
subps xmm1, [rsp + nb302_izO]
subps xmm2, [rsp + nb302_ixH1]
subps xmm3, [rsp + nb302_izH1]
subps xmm4, [rsp + nb302_ixH2]
subps xmm5, [rsp + nb302_izH2]
subps xmm6, [rsp + nb302_ixH1]
subps xmm7, [rsp + nb302_izH1]
subps xmm8, [rsp + nb302_ixH2]

movaps [rsp + nb302_dxOH2], xmm0
movaps [rsp + nb302_dyOH2], xmm1
movaps [rsp + nb302_dzOH2], xmm2

```



```

mulps xmm0, xmm0
mulps xmm1, xmm1
mulps xmm2, xmm2
movaps [rsp + nb302_dxH1H2], xmm3
movaps [rsp + nb302_dyH1H2], xmm4
movaps [rsp + nb302_dzH1H2], xmm5
mulps xmm3, xmm3
mulps xmm4, xmm4
mulps xmm5, xmm5
movaps [rsp + nb302_dxH2H2], xmm6
movaps [rsp + nb302_dyH2H2], xmm7
movaps [rsp + nb302_dzH2H2], xmm8
mulps xmm6, xmm6
mulps xmm7, xmm7
mulps xmm8, xmm8
addps xmm0, xmm1
addps xmm0, xmm2
addps xmm3, xmm4
addps xmm3, xmm5
addps xmm6, xmm7
addps xmm6, xmm8

```

```

;# start doing invsqrt for jH2 atoms
rsqrtps xmm1, xmm0
rsqrtps xmm4, xmm3
rsqrtps xmm7, xmm6

```

```

movaps xmm2, xmm1
movaps xmm5, xmm4
movaps xmm8, xmm7

```

```

mulps xmm1, xmm1 ;# lu*lu
mulps xmm4, xmm4 ;# lu*lu
mulps xmm7, xmm7 ;# lu*lu

```

```

movaps xmm9, [rsp + nb302_three]
movaps xmm10, xmm9
movaps xmm11, xmm9

```

```

mulps xmm1, xmm0 ;# rsq*lu*lu
mulps xmm4, xmm3 ;# rsq*lu*lu
mulps xmm7, xmm6 ;# rsq*lu*lu

```

```

subps xmm9, xmm1
subps xmm10, xmm4
subps xmm11, xmm7 ;# 3-rsq*lu*lu

```

```

mulps xmm9, xmm2
mulps xmm10, xmm5
mulps xmm11, xmm8 ;# lu*(3-rsq*lu*lu)

```

```

movaps xmm4, [rsp + nb302_half]

```

```

    mulps  xmm9, xmm4 ;# rinvOH2
    mulps  xmm10, xmm4 ;# rinvH1H2
mulps  xmm11, xmm4 ;# rinvH2H2

    movaps [rsp + nb302_rinvOH2], xmm9
    movaps [rsp + nb302_rinvH1H2], xmm10
    movaps [rsp + nb302_rinvH2H2], xmm11

    ;# H2 interactions
;# rsq in xmm0,xmm3,xmm6
;# rinv in xmm9, xmm10, xmm11

    movaps xmm1, [rsp + nb302_tsc]
    mulps  xmm0, xmm9 ;# r
    mulps  xmm3, xmm10
    mulps  xmm6, xmm11
    mulps  xmm0, xmm1 ;# rtab
    mulps  xmm3, xmm1
    mulps  xmm6, xmm1

;# truncate and convert to integers
    cvttps2dq xmm1, xmm0
    cvttps2dq xmm4, xmm3
    cvttps2dq xmm7, xmm6

;# convert back to float
    cvtdq2ps  xmm2, xmm1
    cvtdq2ps  xmm5, xmm4
    cvtdq2ps  xmm8, xmm7

;# multiply by 4
    pslld  xmm1, 2
    pslld  xmm4, 2
    pslld  xmm7, 2

;# move to integer registers
    movhlps xmm13, xmm1
    movhlps xmm14, xmm4
    movhlps xmm15, xmm7
    movd  eax, xmm1
    movd  r8d, xmm4
    movd  r12d, xmm7
    movd  ecx, xmm13
    movd  r10d, xmm14
    movd  r14d, xmm15
    pshufd xmm1, xmm1, 1
    pshufd xmm4, xmm4, 1
    pshufd xmm7, xmm7, 1
    pshufd xmm13, xmm13, 1
    pshufd xmm14, xmm14, 1
    pshufd xmm15, xmm15, 1
    movd  ebx, xmm1

```

```

movd    r9d, xmm4
movd    r13d, xmm7
movd    edx, xmm13
movd    r11d, xmm14
movd    r15d, xmm15

mov     rsi, [rbp + nb302_VFtab]

;# calculate eps
subps   xmm0, xmm2
subps   xmm3, xmm5
subps   xmm6, xmm8

movaps   [rsp + nb302_epsO], xmm0
movaps   [rsp + nb302_epsH1], xmm3
movaps   [rsp + nb302_epsH2], xmm6

;# Load LOTS of table data
movlps   xmm1, [rsi + rax*4]
movlps   xmm5, [rsi + r8*4]
movlps   xmm9, [rsi + r12*4]

movlps   xmm3, [rsi + rcx*4]
movlps   xmm7, [rsi + r10*4]
movlps   xmm11, [rsi + r14*4]

movhps   xmm1, [rsi + rbx*4]
movhps   xmm5, [rsi + r9*4]
movhps   xmm9, [rsi + r13*4]

movhps   xmm3, [rsi + rdx*4]
movhps   xmm7, [rsi + r11*4]
movhps   xmm11, [rsi + r15*4]

movaps   xmm0, xmm1
movaps   xmm4, xmm5
movaps   xmm8, xmm9
shufps   xmm0, xmm3, 136 ;# 10001000
shufps   xmm4, xmm7, 136 ;# 10001000
shufps   xmm8, xmm11, 136 ;# 10001000
shufps   xmm1, xmm3, 221 ;# 11011101
shufps   xmm5, xmm7, 221 ;# 11011101
shufps   xmm9, xmm11, 221 ;# 11011101

movlps   xmm3, [rsi + rax*4 + 8]
movlps   xmm7, [rsi + r8*4 + 8]
movlps   xmm11, [rsi + r12*4 + 8]

movlps   xmm12, [rsi + rcx*4 + 8]
movlps   xmm13, [rsi + r10*4 + 8]
movlps   xmm14, [rsi + r14*4 + 8]

```

```
movhps xmm3, [rsi + rbx*4 + 8]
movhps xmm7, [rsi + r9*4 + 8]
movhps xmm11, [rsi + r13*4 + 8]
```

```
movhps xmm12, [rsi + rdx*4 + 8]
movhps xmm13, [rsi + r11*4 + 8]
movhps xmm14, [rsi + r15*4 + 8]
```

```
movaps xmm2, xmm3
movaps xmm6, xmm7
movaps xmm10, xmm11
```

```
shufps xmm2, xmm12, 136 ;# 10001000
shufps xmm6, xmm13, 136 ;# 10001000
shufps xmm10, xmm14, 136 ;# 10001000
shufps xmm3, xmm12, 221 ;# 11011101
shufps xmm7, xmm13, 221 ;# 11011101
shufps xmm11, xmm14, 221 ;# 11011101
```

;/ table data ready in xmm0-xmm3 , xmm4-xmm7 , and xmm8-xmm11

```
movaps xmm12, [rsp + nb302_epsO]
movaps xmm13, [rsp + nb302_epsH1]
movaps xmm14, [rsp + nb302_epsH2]
```

```
mulps xmm3, xmm12 ;# Heps
mulps xmm7, xmm13
mulps xmm11, xmm14
mulps xmm2, xmm12 ;# Geps
mulps xmm6, xmm13
mulps xmm10, xmm14
mulps xmm3, xmm12 ;# Heps2
mulps xmm7, xmm13
mulps xmm11, xmm14
```

```
addps xmm1, xmm2 ;# F+Geps
addps xmm5, xmm6
addps xmm9, xmm10
addps xmm1, xmm3 ;# F+Geps+Heps2 = Fp
addps xmm5, xmm7
addps xmm9, xmm11
addps xmm3, xmm3 ;# 2*Heps2
addps xmm7, xmm7
addps xmm11, xmm11
addps xmm3, xmm2 ;# 2*Heps2+Geps
addps xmm7, xmm6
addps xmm11, xmm10
addps xmm3, xmm1 ;# FF = Fp + 2*Heps2 + Geps
addps xmm7, xmm5
addps xmm11, xmm9
mulps xmm1, xmm12 ;# eps*Fp
mulps xmm5, xmm13
mulps xmm9, xmm14
```

```

movaps xmm12, [rsp + nb302_qqOH]
movaps xmm13, [rsp + nb302_qqHH]
addps xmm1, xmm0    ;# VV
addps xmm5, xmm4
addps xmm9, xmm8
mulps xmm1, xmm12    ;# VV*qq = vcoul
mulps xmm5, xmm13
mulps xmm9, xmm13
mulps xmm3, xmm12    ;# FF*qq = fij
mulps xmm7, xmm13
mulps xmm11, xmm13

;# accumulate vctot
addps xmm1, [rsp + nb302_vctot]
addps xmm5, xmm9
addps xmm1, xmm5
movaps [rsp + nb302_vctot], xmm1

movaps xmm10, [rsp + nb302_tsc]
mulps xmm3, xmm10 ;# fscal
mulps xmm7, xmm10
mulps xmm10, xmm11

movd eax, mm0 ;# restore j3 from mm0-mm3
movd ebx, mm1
movd ecx, mm2
movd edx, mm3

;# move j H2 forces to local temp variables
movlps xmm11, [rdi + rax*4 + 24] ;# jxH2a jyH2a - -
movlps xmm12, [rdi + rcx*4 + 24] ;# jxH2c jyH2c - -
movhps xmm11, [rdi + rbx*4 + 24] ;# jxH2a jyH2a jxH2b jyH2b
movhps xmm12, [rdi + rdx*4 + 24] ;# jxH2c jyH2c jxH2d jyH2d

movss xmm13, [rdi + rax*4 + 32] ;# jzH2a - - -
movss xmm14, [rdi + rcx*4 + 32] ;# jzH2c - - -
movss xmm1, [rdi + rbx*4 + 32] ;# jzH2b - - -
movss xmm2, [rdi + rdx*4 + 32] ;# jzH2d - - -
movlhps xmm13, xmm1 ;# jzH2a - jzH2b -
movlhps xmm14, xmm2 ;# jzH2c - jzH2d -

shufps xmm13, xmm14, 136 ;# 10001000 => jzH2a jzH2b jzH2c jzH2d

;# xmm11: jxH2a jyH2a jxH2b jyH2b
;# xmm12: jxH2c jyH2c jxH2d jyH2d
;# xmm13: jzH2a jzH2b jzH2c jzH2d

xorps xmm0, xmm0
xorps xmm4, xmm4
xorps xmm8, xmm8

mulps xmm3, [rsp + nb302_rinvOH2]

```

```
mulps xmm7, [rsp + nb302_rinvH1H2]
mulps xmm10, [rsp + nb302_rinvH2H2]
```

```
subps xmm0, xmm3
subps xmm4, xmm7
subps xmm8, xmm10
```

```
movaps xmm1, xmm0
movaps xmm2, xmm0
movaps xmm3, xmm4
movaps xmm5, xmm4
movaps xmm6, xmm8
movaps xmm7, xmm8
```

```
mulps xmm0, [rsp + nb302_dxOH2]
mulps xmm1, [rsp + nb302_dyOH2]
mulps xmm2, [rsp + nb302_dzOH2]
mulps xmm3, [rsp + nb302_dxH1H2]
mulps xmm4, [rsp + nb302_dyH1H2]
mulps xmm5, [rsp + nb302_dzH1H2]
mulps xmm6, [rsp + nb302_dxH2H2]
mulps xmm7, [rsp + nb302_dyH2H2]
mulps xmm8, [rsp + nb302_dzH2H2]
```

```
movaps xmm14, xmm0
movaps xmm15, xmm1
addps xmm13, xmm2
addps xmm0, [rsp + nb302_fixO]
addps xmm1, [rsp + nb302_fiyO]
addps xmm2, [rsp + nb302_fizO]
```

```
addps xmm14, xmm3
addps xmm15, xmm4
addps xmm13, xmm5
addps xmm3, [rsp + nb302_fixH1]
addps xmm4, [rsp + nb302_fiyH1]
addps xmm5, [rsp + nb302_fizH1]
```

```
addps xmm14, xmm6
addps xmm15, xmm7
addps xmm13, xmm8
addps xmm6, [rsp + nb302_fixH2]
addps xmm7, [rsp + nb302_fiyH2]
addps xmm8, [rsp + nb302_fizH2]
```

```
movaps [rsp + nb302_fixO], xmm0
movaps [rsp + nb302_fiyO], xmm1
movaps [rsp + nb302_fizO], xmm2
movaps [rsp + nb302_fixH1], xmm3
movaps [rsp + nb302_fiyH1], xmm4
movaps [rsp + nb302_fizH1], xmm5
movaps [rsp + nb302_fixH2], xmm6
```

```

movaps [rsp + nb302_fiyH2], xmm7
movaps [rsp + nb302_fizH2], xmm8

;# xmm14 = fH2x
;# xmm15 = fH2y
;# xmm13 = fH2z
movaps xmm0, xmm14
unpcklps xmm14, xmm15
unpckhps xmm0, xmm15

addps xmm11, xmm14
addps xmm12, xmm0

movhlps xmm14, xmm13 ;# fH2zc fH2zd

movlps [rdi + rax*4 + 24], xmm11
movhps [rdi + rbx*4 + 24], xmm11
movlps [rdi + rcx*4 + 24], xmm12
movhps [rdi + rdx*4 + 24], xmm12
movss [rdi + rax*4 + 32], xmm13
movss [rdi + rcx*4 + 32], xmm14
shufps xmm13, xmm13, 1
shufps xmm14, xmm14, 1
movss [rdi + rbx*4 + 32], xmm13
movss [rdi + rdx*4 + 32], xmm14

;# should we do one more iteration?
sub dword ptr [rsp + nb302_innerk], 4
jl .nb302_single_check
jmp .nb302_unroll_loop

```