

Re: Share Your Experience with 3DNow, SSE, SSE2 etc.

Re: Share Your Experience with 3DNow, SSE, SSE2 etc.

Source: <http://sci.tech-archive.net/Archive/sci.image.processing/2008-08/msg00037.html>

- *From:* aruzinsky <aruzinsky@xxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 4 Aug 2008 10:52:46 -0700 (PDT)
-

I forgot prefetchN = 8.

On Aug 4, 11:42 am, aruzinsky <aruzin...@xxxxxxxxxxxxxxxxxxxxxxxx> wrote:

On Aug 4, 12:15 am, Hendrik van der Heijden <h...@xxxxxx> wrote:

aruzinsky schrieb:

Thank you for your input. I would feel more confident if you tested my code, though.

Then post a pointer to your code, sources preferred.

Hendrik

Simple enough to cut and paste here, but this is an untested excerpt that assumes the arrays are a multiple of 16 floats aligned on 16 byte boundaries.

```
void equalSSE2(int mn, float *aax, float *bbx, int prefetchN)
{
    int NN = 16, MN = mn/NN, MN4 = NN*MN, N = mn-MN4;
    __asm
    {
        push eax
        push ecx
        push edx
        push esi
        mov ecx, prefetchN
```

Re: Share Your Experience with 3DNow, SSE, SSE2 etc.

```
    imul ecx, 64
    mov  eax, aax
    mov  edx, bbx
    mov  esi, MN
    test esi, esi
    jle $L1
    align 16;
$L2:
    movaps xmm0, [edx]
    movaps xmm1, [edx+16]
    movaps xmm2, [edx+32]
    movaps xmm3, [edx+48]

    PREFETCHNTA [edx+ecx]

    movntpd [eax], xmm0
    movntpd [eax+16], xmm1
    movntpd [eax+32], xmm2
    movntpd [eax+48], xmm3

    add  edx, 64
    add  eax, 64

    dec esi
    jnz  $L2
$L1:
    pop  esi
    pop  edx
    pop  ecx
    pop  eax
}
//code for array elements MN4 to mn-1 would go here
```

}– Hide quoted text –

– Show quoted text –