

Hi Mirek,

Yes, I'm nuts... still working at this hour.

Anyway, here's a new version - Fill3T3 - that can actually handle all alignment variations (even those not handled by 7a). Please benchmark and check for correctness:

```
never_inline void FillStream(dword *b, dword data, int len){
```

```
    while((uintptr_t)b & 15){ // Try to align
        *b++=data;
        len--;
    };
    __m128i *w = (__m128i *)b;
    __m128i q = _mm_set1_epi32((int)data);
    if(len>=16){
        __m128i *e = w + (len>>2) - 3;
        do{
            _mm_stream_si128(w++, q);
            _mm_stream_si128(w++, q);
            _mm_stream_si128(w++, q);
            _mm_stream_si128(w++, q);
        }while(w<e);
    }
    if(len & 8) {
        _mm_stream_si128(w++, q);
        _mm_stream_si128(w++, q);
    }
    if(len & 4) {
        _mm_stream_si128(w++, q);
    }
    _mm_sfence();
    _mm_storeu_si128((__m128i*)(b + len - 4), q); // Tail align
}
```

```
void inline Fill3T3(dword *b, dword data, int len){
    if(len<4){
        if(len&1) *b++ = data;
        if(len&2){ *b++ = data; *b++ = data; }
        return;
    }
}
```

```
    __m128i *w = (__m128i *)b;
    __m128i q = _mm_set1_epi32((int)data);
```

```

if(len >= 32) {
    if(len>1024*1024 && (((uintptr_t)b & 3)==0)){
        FillStream(b,data,len);
        return;
    }

    __m128i *e = w + (len>>2) - 7;
    do{
        _mm_storeu_si128(w++, q);
        _mm_storeu_si128(w++, q);
        _mm_storeu_si128(w++, q);
        _mm_storeu_si128(w++, q);
        _mm_storeu_si128(w++, q);
        _mm_storeu_si128(w++, q);
        _mm_storeu_si128(w++, q);
        _mm_storeu_si128(w++, q);
    }while(w<e);
}
if(len & 16) {
    _mm_storeu_si128(w++, q);
    _mm_storeu_si128(w++, q);
    _mm_storeu_si128(w++, q);
    _mm_storeu_si128(w++, q);
}
if(len & 8) {
    _mm_storeu_si128(w++, q);
    _mm_storeu_si128(w++, q);
}
if(len & 4) {
    _mm_storeu_si128(w++, q);
}
_mm_storeu_si128((__m128i*)(b + len - 4), q); // Tail align
}

```

Best regards,

Tom
