

Mirek,

Here it is: The unconditional alignment. I took your idea, ditched my own, and modified your Fill3a as follows:

```
void inline Fill3T(void *b, dword data, int len){
    switch(len){
        case 3: ((dword *)b)[2] = data;
        case 2: ((dword *)b)[1] = data;
        case 1: ((dword *)b)[0] = data;
        case 0: return;
    }
    __m128i q = _mm_set1_epi32(*(int*)&data);
    __m128i *w = (__m128i*)b;

    if(len >= 32) {
        __m128i *e = (__m128i*)b + (len>>2) - 8;
        if(len > 1024*1024 / 16 && ((uintptr_t)w & 3) == 0) { // for really huge data, bypass the cache
            _mm_storeu_si128(w, q); // Head align
            int s=-((int)((uintptr_t)b)>>2)&0x3;
            w = (__m128i*) ((dword*)b) + s;
            do {
                _mm_stream_si128(w++, q);
                _mm_stream_si128(w++, q);
                _mm_stream_si128(w++, q);
                _mm_stream_si128(w++, q);
                _mm_stream_si128(w++, q);
                _mm_stream_si128(w++, q);
                _mm_stream_si128(w++, q);
                _mm_stream_si128(w++, q);
            }while(w<=e);
            _mm_sfence();
        }
        else
            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*) (((dword*)b) + len - 4), q); // Tail align
}

```

I made some other changes too and this one is slightly faster on short transfers while equals Fill3a() on longer ones. The improvement is more significant on MSBT19 / MSBT19x64.

In order to get real fast short transfers, the function must be 'inline'. I think this necessitates two variants of the final function. (I have seen that BufferPainter paints most of the time with really short fills, so inlining really makes a difference there.)

Best regards,

Tom

P.S. My cache threshold is still at 8M...