
Subject: Re: BufferPainter::Clear() optimization
Posted by [mirek](#) on Tue, 28 Apr 2020 08:20:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

Current Fill(RGBA * assembler code

```
4000EEE0  cmp r8d,byte +0x10
4000EEE4  jl 0x14000ef13
4000EEE6  movd xmm0,edx
4000EEEA  pshufd xmm0,xmm0,0x0
4000EEEF  nop
4000EEF0  mov eax,r8d
4000EEF3  movdqu [rcx],xmm0
4000EEF7  movdqu [rcx+0x10],xmm0
4000EEFC  movdqu [rcx+0x20],xmm0
4000EF01  movdqu [rcx+0x30],xmm0
4000EF06  add rcx,byte +0x40
4000EF0A  lea r8d,[rax-0x10]
4000EF0E  cmp eax,byte +0x1f
4000EF11  jg 0x14000eef0
4000EF13  add r8d,byte -0x1
4000EF17  cmp r8d,byte +0xe
4000EF1B  ja 0x14000ef59
4000EF1D  lea r9,[rel 0x4000ef5c]
4000EF24  movsxd rax,dword [r9+r8*4]
4000EF28  add rax,r9
4000EF2B  jmp rax
4000EF2D  mov [rcx+0x38],edx
4000EF30  mov [rcx+0x34],edx
4000EF33  mov [rcx+0x30],edx
4000EF36  mov [rcx+0x2c],edx
4000EF39  mov [rcx+0x28],edx
4000EF3C  mov [rcx+0x24],edx
4000EF3F  mov [rcx+0x20],edx
4000EF42  mov [rcx+0x1c],edx
4000EF45  mov [rcx+0x18],edx
4000EF48  mov [rcx+0x14],edx
4000EF4B  mov [rcx+0x10],edx
4000EF4E  mov [rcx+0xc],edx
4000EF51  mov [rcx+0x8],edx
4000EF54  mov [rcx+0x4],edx
4000EF57  mov [rcx],edx
4000EF59  ret
```

and the central snippet from the memsetd variant....

```

40001565 movaps xmm0,[rel 0x402c60a0]
4000156C nop dword [rax+0x0]
40001570 movups [rsi+rdx*4],xmm0
40001574 movups [rsi+rdx*4+0x10],xmm0
40001579 movups [rsi+rdx*4+0x20],xmm0
4000157E movups [rsi+rdx*4+0x30],xmm0
40001583 movups [rsi+rdx*4+0x40],xmm0
40001588 movups [rsi+rdx*4+0x50],xmm0
4000158D movups [rsi+rdx*4+0x60],xmm0
40001592 movups [rsi+rdx*4+0x70],xmm0
40001597 movups [rsi+rdx*4+0x80],xmm0
4000159F movups [rsi+rdx*4+0x90],xmm0
400015A7 movups [rsi+rdx*4+0xa0],xmm0
400015AF movups [rsi+rdx*4+0xb0],xmm0
400015B7 movups [rsi+rdx*4+0xc0],xmm0
400015BF movups [rsi+rdx*4+0xd0],xmm0
400015C7 movups [rsi+rdx*4+0xe0],xmm0
400015CF movups [rsi+rdx*4+0xf0],xmm0
400015D7 add rdx,byte +0x40
400015DB add rdi,byte +0x8
400015DF jnz 0x140001570

```

Interesting...

Benchmarking code

```

#include <CtrlLib/CtrlLib.h>

using namespace Upp;

GUI_APP_MAIN
{
    Color c = Red();

    int len = 4000 * 2000;

    Buffer<RGBA> b(len);

    for(int i = 0; i < 1000; i++) {
        {
            RTIMING("memsetd");
            memsetd(b, *(dword*)&(c), len);
        }
        {
            RTIMING("Fill");

```

```
    Fill(b, c, len);  
  }  
}  
}
```

CLANGx64, 2700x

TIMING Fill : 2.73 s - 2.73 ms (2.73 s / 1000), min: 2.00 ms, max: 4.00 ms, nesting: 0 - 1000

TIMING memsetd : 2.78 s - 2.78 ms (2.78 s / 1000), min: 2.00 ms, max: 5.00 ms, nesting: 0 - 1000

MSBT19x64

TIMING Fill : 2.89 s - 2.89 ms (2.89 s / 1000), min: 2.00 ms, max: 5.00 ms, nesting: 0 - 1000

TIMING memsetd : 2.90 s - 2.90 ms (2.90 s / 1000), min: 2.00 ms, max: 5.00 ms, nesting: 0 - 1000
