

Computer Architecture Project5.

컴퓨터과학과 2013147513 조영재

vectorization을 사용하는 이유는, 두 배열의 원소들을 반복문을 돌며 곱할 때, 반복에 따른 연산의 종속성이 존재하지 않다면, vectorization을 이용해 빠르게 계산을 수행할 수 있기 때문에 vectorization을 사용한다.

```
0000000000000000 <main>:
0: 31 f6          xor     %esi,%esi
2: b9 20 00 00    mov     $0x20,%ecx
7: 83 c6 08       add     $0x8,%esi
a: 81 fe 00 02 00 00    cmp     $0x200,%esi
10: 74 31         je      43 <main+0x43>
12: 31 c0         xor     %eax,%eax
14: 0f 1f 40 00    nopl    0x0(%rax)
18: 8b 90 00 00 00 00    mov     0x0(%rax),%edx
1e: 4b 83 c0 04     add     $0x4,%rax
22: 0f af 90 00 00 00 00    imul    0x0(%rax),%edx
29: 8b 90 00 00 00 00    mov     %edx,0x0(%rax)
2f: 4b 39 c0       cmp     %rcx,%rax
32: 75 e4         jne     18 <main+0x18>
34: 83 c6 08       add     $0x8,%esi
37: 4b 80 c3 20     add     $0x20,%rcx
3b: 81 fe 00 02 00 00    cmp     $0x200,%esi
41: 75 cf         jne     12 <main+0x12>
43: 31 c0         xor     %eax,%eax
45: c3          retq

74 0000000000000000 <main>:
75: 0: 31 f6          xor     %esi,%esi
76: 2: 83 c6 08       add     $0x8,%esi
77: 5: 81 fe 00 02 00 00    cmp     $0x200,%esi
78: b: 74 70         je      7d <main+0x7d>
79: d: 0f 1f 00      nopl    (%rax)
80: 10: 0d 4e fc       lea     -0x4(%rsi),%ecx
81: 13: 31 c0         xor     %eax,%eax
82: 15: 31 d2        xor     %edx,%edx
83: 17: c1 e9 02      shr     $0x2,%ecx
84: 1a: 83 c1 01      add     $0x1,%ecx
85: 1d: 0f 1f 00      nopl    (%rax)
86: 20: 66 0f 6f 88 00 00 00    movdqa 0x0(%rax),%xmm1
87: 27: 00
88: 28: 83 c2 01      add     $0x1,%edx
89: 2b: 4b 83 c0 10     add     $0x10,%rax
90: 2f: 66 0f 6f 80 00 00 00    movdqa 0x0(%rax),%xmm0
91: 36: 00
92: 37: 66 0f 6f d1    movdqa %xmm1,%xmm2
93: 3b: 66 0f 6f 88 00 00 00    movdqa 0x0(%rax),%xmm1
94: 42: 00
95: 43: 66 0f f4 80 00 00 00    pmuludq 0x0(%rax),%xmm0
96: 4a: 00
97: 4b: 66 0f 70 c9 08    pshufd $0x8,%xmm0,%xmm0
98: 50: 66 0f 73 d2 2b    pslrq  $0x20,%xmm2
99: 55: 66 0f 73 d1 2b    pslrq  $0x20,%xmm1
100: 5a: 66 0f f4 ca     pmuludq %xmm2,%xmm1
```

왼쪽이 vec_mult_o2.dis이고 오른쪽이 vec_mult_o3.dis이다.

가장 눈여겨볼 차이점은, 오른쪽에는 pmuludq라는 명령어가 있는 반면, 왼쪽에는 단순히 imul명령어가 있다는 것이다. 또한, 오른쪽에는 xmm레지스터를 사용하는데 반해 왼쪽에서는 xmm레지스터를 사용하지 않는다는 것이다.

pmuludq라는 명령어를 찾아보니, multiply packed unsigned quadword integer라고 되어 있었다. 아마도 이것이 vector곱을 해주는 명령어 인 것 같다.