

Architecture 1001: x86-64 Assembly ShiftExample2.c

Xeno Kovah – 2021
xeno@darkmentor.com

All materials is licensed under a Creative Commons “Share Alike” license.

- <http://creativecommons.org/licenses/by-sa/4.0/>

You are free:



to Share — to copy, distribute and transmit the work



to Remix — to adapt the work

Under the following conditions:



Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).



Share Alike — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license.

ShiftExample2Unsigned.c

Note: Compiled with /O2 “Maximize Speed”, to clear away a bit of cruft

```
#include <stdlib.h>
```

```
int main(int argc, char **argv)
```

```
{
```

```
    unsigned int a, b, c;
```

```
    a = atoi(argv[1]);
```

```
    b = a * 8;
```

```
    c = b / 16;
```

```
    return c;
```

```
}
```

```
main:
```

```
00000000140001000    sub    rsp,28h
```

```
00000000140001004    mov    rcx,qword ptr [rdx+8]
```

```
00000000140001007    call  qword ptr [00000000140004178h]
```

```
0000000014000100D    shl   eax,3
```

```
00000000140001010    shr   eax,4
```

```
00000000140001013    add   rsp,28h
```

```
00000000140001017    ret
```

 Where are the multiply and divide instructions?! 

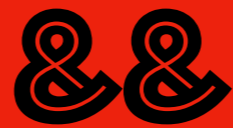
ShiftExample2Unsigned.c

Takeaway

- When a multiply or divide is by a power of 2, compilers prefer shift instructions as a more efficient way to perform the computation



STOP



**STEP THROUGH
THE ASSEMBLY**



CHECK YOUR UNDERSTANDING



STOP



**STEP THROUGH
THE ASSEMBLY**



CHECK YOUR UNDERSTANDING