

Forum: [Hack Forums](#)

Thread: [AMD64 \(64-bit mode\) and \[RSP\] addressing "anomaly"?](#)

DracoRexifer - March 7, 2019, 1:45 pm

I have been looking at JWASM output for various addressing modes for the NOT instruction and found this strange code output:

000000F2	F65076	not BYTE PTR 076h[RAX]
000000F5	F65176	not BYTE PTR 076h[RCX]
000000F8	F65276	not BYTE PTR 076h[RDX]
000000FB	F65376	not BYTE PTR 076h[RBX]
000000FE	F6542476	not BYTE PTR 076h[RSP]
00000102	F65576	not BYTE PTR 076h[RBP]
00000105	F65676	not BYTE PTR 076h[RSI]
00000108	F65776	not BYTE PTR 076h[RDI]

When the register used for indirect addressing is RSP, there is an extra byte (24) that precedes the offset (76). This also happens when using R12:

0000023A	41F65076	not BYTE PTR 076h[R8]
0000023E	41F65176	not BYTE PTR 076h[R9]
00000242	41F65276	not BYTE PTR 076h[R10]
00000246	41F65376	not BYTE PTR 076h[R11]
0000024A	41F6542476	not BYTE PTR 076h[R12]
0000024F	41F65576	not BYTE PTR 076h[R13]
00000253	41F65676	not BYTE PTR 076h[R14]
00000257	41F65776	not BYTE PTR 076h[R15]

It probably has to do with R12 having the same value in the ModRM byte (value of 4) but the REX byte (41) makes it 12. However, 24 is not one of the listed prefixes and is actually the opcode for:

AND AL, imm8

Can anyone explain this?

I just figured it out. The 24 is the SIB byte. When the r/m field in the ModRM byte is 4, then it means that a SIB byte follows. This is the only way that the SP can be used as an index register. But that is something I would probably never want to do.