Brandon Buchanan
9.20.11

CSCI2160WorksheetF11MemoryAccess

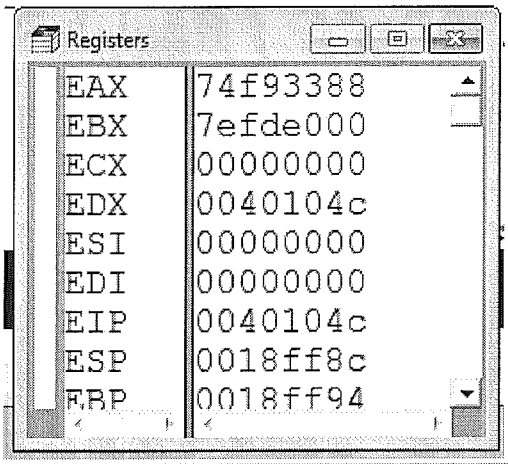
Assume that the prefix indicates the type. address of each of the below memory locations is given as indicated. Assume, also that the prefix name indicates the type of variable (w:WORD, d:DWORD, b:BYTE, q:QWORD)

wT: 0x00404014
bAns: 0x00404035
dVal: 0x00404022
qNum: 0x00404030

✓	mov ax,wT	ax:	4E 48	
✓	mov bl,bAns	bl:	00	
✓	mov ecx,dVal	ecx:	4952 204E	
✓	mov ebx, offset wT	ebx:	00404014	
✓	mov ax,wT+3	eax:	4557	
✓	mov cx,wT+3	cx:	4557	
✓	mov bl, wT+3	bl:	4557	
✓	mov dx,bAns[6]	dx:	42	
✓	mov edx,dVal]0]	edx:	4952 204E	
✓	mov eax,bAns[2]	eax:	FF	
✓	mov esi,offset qNum	esi:	00404030	
	mov eax,[esi]	eax:	00000000	
✗	mov edx,[esi+4]	edx:	00000004	FF EF 20 2E
✓	mov ebx, offset wT	ebx:	00404014	
✓	mov esi,4	esi:	00000004	
✗	mov eax,[ebx+esi+2]	eax:	7EFDE002	4F 54 20 54
✗	mov eax,[ebx][esi +4]	eax:	7EFDE004	54 20 4F 54

Given the prefix indicates the type of identifiers above determine which of the following are correct/incorrect and give the reason why.

✓ mov	eax, wT	no, eax is 4 bytes, wT is 2 byte	
✓ mov	al,wT[2]	no, al is 1 byte, wT is 2 byte	
✓ mov	edx,dVal[3]	correct	
✓ mov	wT,ax	correct	
✓ mov	wT,wY	no, cannot do memory to memory operations	
✓ mov	bAns,ax	no, bAns is 1 byte, ax is 2 bytes	
✓ mov	ebx,qNum	no, ebx is 4 bytes, qNum is 8 bytes	
✓ mov	ebx,offset wT	correct	
✓ mov	eax,dVal+1	correct	
✓ mov	bx,offset wT	no, bx is 2 bytes, wT is an 4 byte address	
✓ mov	eax, bl	no, eax is 4 bytes, bl is 1 byte	
✓ mov	ebx, 4	correct	
✓ mov	[ebx],4	no, [ebx] references the address	assembler has to know the size of 4, [ebx] is an address
✗ add	[ebx+esi],eax	no, [ebx+esi] references the address	valid
✓ mov	bx,[ebx+esi]	correct	
✗ mov	word ptr [ebx],4	no, word is 2 bytes, ebx is 4 bytes	4 is expanded to 2 bytes
✓ mov	dword ptr[ebx],24	correct	4 is expanded to 4 bytes
✓ mov	wT, [ebx]	correct	
✓ add	eax,bl	no, eax is 4 bytes, bl is 2bytes	
✓ mov	ax, offset wT	no, ax is 2 bytes, offset wT is 4 bytes	
✓ mov	dNum, si	no, dNum is 4 bytes, si is 2 bytes	
✓ mov	bl,275	no, bl is 1 byte, storing 275 needs 2 bytes	disregard sign unless specified
✓ mov	ebx,-1	correct	
✓ mov	bVal,-1	correct	
✗ mov	wAnswer,33999	no, wAnswer can store up to 32767	system doesn't care about sign
✗ add	bVal,ax	correct	bVal is 1 byte, ax is 2 bytes



Memory(&wT)

```
0x00404000 df ff 07 00 1b 00 05 05 00 00 00 00 00 00 27 00
0x00404010 37 00 4a 4f 48 4e 20 57 45 4e 54 20 54 4f 20 54
0x00404020 4f 57 4e 20 52 49 44 49 4e 47 20 4f 4e 20 41 20
0x00404030 50 4f 4e 59 2e 00 ef ff ff ff 41 42 43 41 42 43
0x00404040 c1 00 c1 00 c1 00 37 07 00 00 37 07 00 00 d3 7f
0x00404050 d3 7f d3 7f d3 7f 00 00 00 00 00 00 00 00 00 00
0x00404060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00404070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 42 53
0x00404080 74 72 69 6e 67 20 74 6f 20 70 72 69 6e 74 00 0a
0x00404090 0d 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x004040A0 00 00 00 00 00 00 0a 0d 45 6e 74 65 72 20 61 20
0x004040B0 73 74 72 69 6e 67 20 28 74 68 65 6e 20 45 4e 54
0x004040C0 45 52 29 3a 20 00 00 00 00 00 00 00 00 00 00 00
0x004040D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```