AOZ Studio Beta - Bug #25

Rol.b and Ror.b cause syntax errors.

01/18/2020 01:57 PM - Brian Flanagan

Status:	Closed	**Start date:**	01/18/2020
Priority:	Normal	**Due date:**	
Assignee:	Francois Lionet	**% Done:**	0%
Category:		**Estimated time:**	0:00 hour
Target version:	0.9.3.2		
Affected version:	0.9.2.6		

Description

Example: Help_55, lines 157 & 160 from AMOSPro_Examples

RL: Rol.b 1,A : Return
'
'
RR: Ror.b 1,A : Return

Maybe these just aren't implemented yet, but if so, should produce Instruction not implemented error.

History

#1 - 01/21/2020 04:08 PM - Francois Lionet

- Status changed from New to Resolved

- Assignee set to Francois Lionet

- Target version set to 0.9.3

Fixed.

#2 - 01/21/2020 11:17 PM - Brian Flanagan

Francois Lionet wrote:

> Fixed.

Nope. These instructions still cause syntax errors in 0.9.3.
This example won't compile.
Is it possible that it is because the instructions are on the same line as the labels (RL & RR)?
Note that the error points to the dot after Rol or Ror.

The result is errors on the Rol.b and Ror.b commands:

First pass...
main.aoz:157:8: error: syntax error

main.aoz:160:8: error: syntax error
{}
Task failed, no code generated...

Here's the entire source for your convenience:

```
'*********************************
'*    AMOS Professional HELP 55   *               SUBJECTS COVERED
'*                                *
'*     (c) Europress Software     *                Rol. (line 11)
'*                                *                Ror. (line 45)
'*         Ronnie Simpson         *                Btst (line 58)
'*********************************                Bset (line 75)
'                                                  Bclr (line 85)
'                                                  Bchg (line 95)
'-----------------------------------------
'Rol
'-----------------------------------------
```

```
'rotate a binary number left
'
'                +--->Type
'                | +--->Number of times
'                | | +--->Variable
'             Rol.b 1,V
'
'There are 3 variations of the instruction:-
'
'             Rol.b (an 8 bit byte)
'             Rol.w (a word-2 bytes)
'             Rol.l (a long word-4 bytes)
'
'The binary representation of the variable will be rotated left the number of
'times stated and a wrap-around takes place for example:-
'
'             %11100111  rotated left would become  %11001111
'
'The same holds true for the word and longword versions.
'
'If the variable is replaced with an expression then this will be assumed to
'be an address in memory at which the bitwise rotations will take place.
'
'eg           A#=1234.56
'             Rol.w 2,Varptr(A#)
'             Print A#             (printed result: 842.216)
'
'
'Rol. can be used to quickly multiply a positive number by 2.
'
'
'-----------------------------------------
'Ror
'-----------------------------------------
'rotate a binary number right
'
'Ror. works in the same way as the above Rol. instruction only rotating to
'the right.
'Once again there are 3 possible variations of the instruction:-
'
'    Ror.b (1 Byte)          Ror.w (2 Bytes)         Ror.l (4 Bytes)
'
'Ror. can be used as a quick way to divide a positive number by 2.
'
'-----------------------------------------
'Btst
'-----------------------------------------
'Test a bit
'
'                +--->Position of bit
'                | +--->Variable name
'eg       B=Btst(7,A)
'
'The result of this function will be either -1 (true) if the tested bit
'is set or 0 (false) if it is clear.
'eg.             A=%10101010
'                B=Btst(7,A) : C=Btst(6,A)
'                Print B,C                     (result printed: -1    0)
'
'Variable name may be substituted by an address in memory.
'
'-----------------------------------------
'Bset
'-----------------------------------------
'set a bit
'
'eg.             Bset 7,A     (set bit 7 of variable A)
'
'Bset sets the named bit irrespective of its present state.
'Variable name may be substituted by an address in memory.
'
'-----------------------------------------
'Bclr
'-----------------------------------------
'clear a bit
```

```
'
'eg.            Bclr 2,B     (clear bit 2 of variable B)
'
'Bclr clears the named bit irrespective of its present state.
'Variable name may be substituted by an address in memory.
'
'-----------------------------------------
'Bchg
'-----------------------------------------
'change a bit
'
'eg.            Bchg 3,C     (change bit 3 of variable C)
'
'Bchg changes the named bit ie. if the bit is set to 1 then it will be
'changed to 0 and vice versa.
'Variable name may be substituted by an address in memory.
'
'-----------------------------------------
'WORKING EXAMPLE
'-----------------------------------------
Rem *** open screen and reserve some memory for screen zones and array
'
Screen Open 0,320,200,8,Lowres
Reserve Zone 15 : Dim X(8)
Palette $0,$F00,$F0,$F,$FF0,$F0F,$FF,$F95
Flash Off : Curs Off : Cls 0 : Paper 0
'
Rem *** change mouse shape
'
Change Mouse 2
'
Rem *** set out the screen zones
'
For N=1 To 8
   X(N)=320-20*N-20
   Box X(N),100 To X(N)+16,116
   Set Zone N,X(N),100 To X(N)+16,116
Next
Pen 2 : Locate 2,2 : Print Border$(Zone$("  ROLL LEFT   ",9),1)
Pen 2 : Locate 2,5 : Print Border$(Zone$("  ROLL RIGHT  ",10),1)
Pen 6 : Locate 2,8 : Print Border$(Zone$("  SET A BIT   ",11),1)
Pen 6 : Locate 2,11 : Print Border$(Zone$(" CLEAR A BIT  ",12),1)
Pen 5 : Locate 2,14 : Print Border$(Zone$(" CHANGE A BIT ",13),1)
Pen 1 : Locate 2,17 : Print Border$(Zone$("     QUIT     ",14),1)
'
Rem *** start the main loop
'
A=170
Do
   Pen 4 : Ink 4 : Locate 21,6 : Print "DECIMAL-";A;"  "
   Locate 19,9 : Print "BINARY-"+Bin$(A,8)
   For N=0 To 7
      If Btst(N,A)
         Ink 3 : Paint X(N+1)+1,102
      Else
         Ink 0 : Paint X(N+1)+1,102
      End If
   Next
   Repeat
      Pen 7 : Locate 0,20 : Print "     ^ ^ ^ ^    <---SELECT AN OPERATION"
      Locate 20,17 : Print "                "
      M=Mouse Zone
   Until Mouse Key and M>8
   If M=14 Then Edit
   Add M,-8
   On M Gosub RL,RR,SB,CB,CH
Loop
'
'
RL: Rol.b 1,A : Return
'
'
RR: Ror.b 1,A : Return
'
'
```

```
SB: Locate 0,20 : Print "          CLICK IN A BOX TO SET ANY BIT"
Locate 20,17 : Print "^ ^ ^ ^ ^ ^ ^ ^"
Repeat
   M=Mouse Zone
Until Mouse Key and M>0 and M<9
Dec M : Bset M,A : Return
'
'
CB: Locate 0,20 : Print "        CLICK IN A BOX TO CLEAR ANY BIT"
Locate 20,17 : Print "^ ^ ^ ^ ^ ^ ^ ^"
Repeat
   M=Mouse Zone
Until Mouse Key and M>0 and M<9
Dec M : Bclr M,A : Return
'
'
CH: Locate 0,20 : Print "       CLICK IN A BOX TO CHANGE ANY BIT"
Locate 20,17 : Print "^ ^ ^ ^ ^ ^ ^ ^"
Repeat
   M=Mouse Zone
Until Mouse Key and M>0 and M<9
Dec M : Bchg M,A : Return
```

#3 - 01/22/2020 10:31 PM - Baptiste Pillot

Error confirmed in 0.9.3.1.

- source : https://www.amos2.fr/ide/Amos2/Ide/Program/423
- compile result :

```
    main.aoz:157:8: error: syntax error
    main.aoz:160:8: error: syntax error
```

#4 - 01/22/2020 10:31 PM - Baptiste Pillot

- Status changed from Resolved to Feedback

#5 - 01/24/2020 10:11 AM - Francois Lionet

- Status changed from Feedback to Resolved

- Target version changed from 0.9.3 to 0.9.3.2

Fixed.

#6 - 02/01/2020 07:42 AM - Baptiste Pillot

- File main.aoz added

- Status changed from Resolved to Feedback

This now compiles, but :

"Illegal text window parameter at line: 130, column: 9."

Lines that cause errors (I commented them successively) :

- 130 : Pen 5 : Locate 2,14 : Print Border$(Zone$(" CHANGE A BIT ",13),1)
- 131 : Pen 1 : Locate 2,17 : Print Border$(Zone$("   QUIT   ",14),1)
- 147 : Pen 7 : Locate 0,20 : Print "   ^ ^ ^ ^   <---SELECT AN OPERATION"
- 148 : Locate 20,17 : Print "   "

Once I commented all these lines, the program runs, but if I click between "CLEAR A BIT" and the button line above, it throws another error :

"Illegal text window parameter at line: 171, column: 5"

#7 - 02/03/2020 06:44 PM - Francois Lionet

- File Annotation 2020-02-03 191342.png added

- Status changed from Feedback to Resolved

OK just tried, Help_55 works. (see picture)

#8 - 02/22/2020 10:33 AM - Brian Flanagan

- Status changed from Resolved to Closed

Verified it's working now! Closing issue.

Files

main.aoz	5.57 KB	02/01/2020	Baptiste Pillot
Annotation 2020-02-03 191342.png	24 KB	02/03/2020	Francois Lionet