ACEReference.doc

COLLABORATORS			
	TITLE : ACEReference.doc		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 5, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

1 ACEReference.doc **1**

1.1 Main Menu . . . 1
1.2 Introduction . . . 1
1.3 Getting started . . . 1
1.4 History . . . 2
1.5 Stop bits . . . 4
1.6 What is the ACE SuperOptimizer? . . . 4
1.7 Who is it for? . . . 5
1.8 Installation . . . 5
1.9 Using the ACE SuperOptimizer . . . 5
1.10 SuperOptimizer options . . . 6
1.11 SuperOptimizer in action . . . 7
1.12 Running SuperOptimized ACE programs . . . 9
1.13 SuperOptimizer limitations . . . 9
1.14 Future versions . . . 10
1.15 A note to PD libraries and reviewers . . . 10
1.16 Disclaimer . . . 11
1.17 Contacting the author . . . 11
1.18 Final word . . . 11

Chapter 1

ACEReference.doc

1.1 Main Menu

```
                        +--------------------------+
                | ACE SuperOptimizer V1.42 |
                +--------------------------+


                Introduction

                Getting started

                History

                Stop bits
```

1.2 Introduction

```
                                    Introduction
                        ------------


                What is the ACE SuperOptimizer?

                Who is it for?
```

1.3 Getting started

```
                                    Getting started
                        ---------------
```

Installation

Using the SuperOptimizer

SuperOptimizer options

SuperOptimizing ACE programs

SuperOptimizer in action

Limitations

1.4 History

```
                              History
                              -------

          ! = bugfix.
          m = minor update.
          M = major update.


30/12/1995   First public release.

01/01/1996 m Added the RemoveLine.b program to the SuperOptimizer archive.
             David forgot to include it, so I wrote my own version in ACE...
             It uses two 8K buffers for reads/writes and is FAAAASSST :)

01/01/1996 ! I have discovered some bugs when receiving ACE V2.37
             - The program is now converting all movea mnemonics to move.
               Problem showed up when SuperOptimising gadtools.b program.
               A68k will compile the resulting code even if the dedicated
               mnemonic is not used.
               I think that most humans don't use the movea(ddress) aniway.

             - Found a tricky bug in optlevel 6, cmp.x Dn,Dn is doing a false
               optimisation when a byte is compared to an immediate value.
               Ex.: IF Peek(FileLineAddress)<>35% THEN will be compiled to
                    something like this.

                   ACE optimized output       SuperOptimizer V0.99
                   --------------------       --------------------
                   _lab27:                    _lab27
                      move.l  -16(a5),a0         move.l  -16(a5),a0
                      move.b  (a0),d0            move.b  (a0),d0
                      ext.w   d0                 ext.w   d0
                      cmp.w   #0,d0           ;; tst.w   d0
                      bge.s   _lab28             bge.s   _lab28
                      not.w   d0                 not.w   d0       This code
                      move.w  #255,d1            move.w  #255,d1  only appears
                      sub.w   d0,d1              sub.w   d0,d1    for bytes.
                      move.w  d1,d0           ;; move.w           wrong!
                   _lab28:                    _lab28:
                      move.w  d0,-(sp)        ;; move.w
```

```
                move.w  #35,d1          ;; MOVEQ
                move.w  (sp)+,d0        ;; move.w
                moveq   #-1,d5             moveq   #-1,d5
                cmp.w   d1,d0              cmp.w   #35,d1   wrong!
                bne.s   _lab29             bne.s   _lab29
                moveq   #0,d5              moveq   #0,d5
              _lab29:                   _lab29:
                move.l  d5,d0              move.l  d5,d0
                cmpi.l  #0,d0           ;; tst.l
                bne.s   _lab30             bne.s   _lab30
                jmp     _lab31             jmp     _lab31
```

This bug only appeared when comparing a byte value to a word or a long value. This is because extra code is added to cope with byte values becoming negative when extended to word or long. Bug is fixed.

- Bug fixed when optimizing additional stack operations. MOveq #immediate value, can only be used if destination is a data register.

04/01/1996 m Did a little bit of fine-tuning.
- move.l #0,An becomes SUB.l An,An when optimizing absolute moves.
- some move.l #value,Dn were not optimized to MOveq when optimizing absolute moves.

06/01/1996 ! Found a bug when optimizing additional stack operations.
- Ex.: B=-B*A where A and B are both of type SHORTINT. It would produce this false code.

```
;; move.w
   neg.w  (sp)
   move.w -2(a4),d0    variable A in D0
;; move.w
   muls   -4(a4),d0    A * B
   move.w  d0,-4(a4)   store result in B
```

As you can see, B is never negated but the stack is...

Bug is fixed and this code is generated instead :

```
   move.w -4(a4),-(sp) push B onto the stack
   neg.w  (sp)         negate B
   move.w -2(a4),d0    variable A in D0
;; move.w
   muls   (sp)+,d0     A * (-B)
   move.w  d0,-4(a4)   store result in B
```

Could come up with a full registerized version in the future.

10/01/1996 m FPuts, buffered writing, is used instead of _Write when creating the SuperOptimized source. This will give a noticeable speed increase. This however makes the SuperOptimizer KS1.3 incompatible.

12/05/1996 ! Corrected a serious bug when optimizing absolute moves.

```
13/05/1996 M Added float-math optimizations, ACE is burning rubber!
             Extended the absolute move optimizations, so that external calls
             are also optimized.
             Levels 9 and 10 have changed place.
             Added version information.

15/06/1996 ! Corrected a bug when doing integer math optimizations.
             Found the bug when optimizing the new BobStars program that
             is now using arithmetic shifts.
```

1.5 Stop bits

```
                                        Stop bits
                                ---------
```

Future versions

A note to PD libraries and reviewers

Disclaimer

Contacting the author

Final word

1.6 What is the ACE SuperOptimizer?

```
What is the ACE SuperOptimizer?
-------------------------------

The SuperOptimizer is a freely distributable, 'intelligent' peephole-optimising
A68K-compatible assembly source code optimizer that has been specially designed
for use with ACE.
It can, in most cases, double the amount of optimisations when used in conjunction
with ACE's internal peephole optimizer.

It runs under Wb 2.x and up.

It will happily run in 1024K, but more than this is required for very large
assembly source files.
A 68020+ system with some fast memory is highly recommended but the program
does not depend on it.

The following files constitute a complete ACE SuperOptimizer package:


  - SuperOptimizer        : the program.

  - SuperOptimizer.guide  : what you are reading now...
```

```
   - RemoveLine.b          : ACE version of RemoveLine, for use with acpp or dcpp.
                             Very fast preprocessed source line 'shaver'.

The ACE SuperOptimizer is written entirely in ACE and has been tested with
Enforcer V37.62, © Michael Sinz.

A68K and Blink are used to assemble and link the code produced by the
ACE SuperOptimizer. They can be found in the ACE archive.

The complete ACE SuperOptimizer package may be freely distributed.
```

1.7 Who is it for?

```
Who is it for?
--------------

The ACE SuperOptimizer is intended only for anyone who already knows ACE and wants
the following:

  - Even faster program execution.

  - Smaller code.
```

1.8 Installation

```
Installation
------------

You will need to open a shell to install the ACE SuperOptimizer.
Installation consists of:

  - Extracting the archive found in the single supplied ACE SuperOptimizer
    archive.
    Copying the SuperOptimizer to the ACE:bin drawer.
    Copying the guide and it's icon to the ACE:docs drawer.

    ACE, ofcourse, has to be installed to be of any use. ;-)

That's it!
```

1.9 Using the ACE SuperOptimizer

```
Using the ACE SuperOptimizer
----------------------------

There are two ways to use the SuperOptimizer:

  - From the shell/CLI.
  - Via an Integrated Development Environment: AIDE.
```

```
Whichever environment you choose to work with the SuperOptimizer, read on.

The SuperOptimizer expects 3 parameters from the command line, none of them
may be ommitted.

It's general format is: SuperOptimizer optlevel asm_infile asm_outfile
The SuperOptimizer expects all assembly source files to have a ".s" extension.

If you have an assembly program called foo.s, that has been created by the ACE
compiler, you would invoke the SuperOptimizer thus:

  SuperOptimizer 12 foo.s foo_opt.s

This would produce foo_opt.s, an A68K-compatible assembly source (text) file
with maximum optimization level.

To use the SuperOptimizer from AIDE, you will have to modify the AIDE BASIC
source and recompile it.
My hopes for the future are, that the ACE SuperOptimizer will become an integral
part of the ACE package.
```

1.10 SuperOptimizer options

```
SuperOptimizer options
----------------------
The full command line syntax for the SuperOptimizer is:

  SuperOptimizer optlevel asm_infile asm_outfile

The 'optlevel' defines which level of optimisation has to be done (1 to 12).

  1  Only optimizes stack operations.
  2  Do level 1 optimizations plus move.x to moveq optimizations.
  3  Do level 1 and 2 optimizations plus cmpi.x #0 to tst.x optimizations.
  4  Do level 1 to 3 optimizations plus tst.x optimizations.
  5  Do level 1 to 4 plus logical optimizations.
  6  Do level 1 to 5 plus cmp.x Dn,Dn optimizations.
  7  Do level 1 to 6 plus integer math optimizations.
  8  Do level 1 to 7 plus immediate zero move optimizations.
  9  Do level 1 to 8 plus additional stack optimizations.
  10 Do level 1 to 8 plus float-math and library base move optimizations.
  11 Do level 1 to 10 plus absolute long optimizations.
  12 Do level 1 to 11 plus unsigned long multiplication optimizations.

So allways use optlevel 12 for maximum optimization.

Beware, the asm_infile and asm_outfile have to be different files.

It may occur that a program stops running correctly with maximum optimization, so
lower the 'optlevel' until it works again.
This will make it very easy for me to hunt down any existing bugs.

You could also use the 'optlevel' to see what is optimized at a different
```

optimization level.

1.11 SuperOptimizer in action

SuperOptimizer in action.

This section is only included to get your attention!

All the programs below have been optimized with optimization level 12 and are working as they should.

Remember, the SuperOptimizer works best in conjunction with ACE's internal peephole optimizer.

The figures are form V1.0 of the SuperOptimizer, V1.41 is producing even better results.

Program	ACE V2.35 peephole optimizer	SuperOptimizer	Total
Ehb.b	121	112	233
Ham.b	212	128	340
Hopnet.b	1631	1325	2956
Iff.b	69	38	107
Maze.b	476	921	1397
Pattern.b	84	110	194
Shuttle.b	336	321	657
Ifs.b *	613	377	990
Galcolsim.b	839	1031	1870
Jovian.b	1073	1122	2195
Messier.b	248	323	571
Ackermann.b	67	79	146
Calc.b	25	11	36
Lines.b	82	51	133
Pcwall.b	201	223	424
Sieve.b	25	43	68
Henon.b	218	167	385

Kx.b	59	38	97
Lorenz.b	223	192	415
Sierp.b	71	60	131
Acegadgets.b	180	45	225
Events.b	128	42	170
Aterm.b	108	82	190
Print.b	178	209	387
Seq.b	65	39	104
Hello.b	4	5	9
Library.b	49	47	96
Openwindow.b	312	262	574
Reqtools.b	20	38	58
Caps.b	19	14	33
Ledfade.b	51	43	94
Ledflash.b	36	16	52
Arr_of_str.b	23	22	45
Fact.b	29	27	56
Hanoi.b	40	40	80
Linkedlist.b	21	33	54
Task.b	14	26	40
Bc.b	377	525	902
Easter.b	205	141	346
Play.b	285	241	526
Sound.b	96	70	166
Boxit.b	67	46	113
Bst.b	438	495	933
Dragon.b	73	57	130
Flower.b	44	18	62
Snowflake.b	104	70	174

Torus.b	41	20	61
Tree.b	75	52	127
ReqEd V1.11	2336	2765	5101
Acecalc.b	597	761	1358
A-A.b	514	1243	1757
Fd2bmap.b	496	573	1069
Uppercacer.b	448	582	1030
Indent.b	215	307	522
Stratego.b !	3368	3749	7117
Life.b	967	991	1958
Lisp.b %	???	1226	1226
SuperOptim.	2858	4345	7203

* added t!=TIMER and PRINT TIMER-t!
! Optimized, compiled and linked ok. NO test-run.
% ACE's peephole optimizer is turned off because of negation bug when optimized.
  Also, optimization level 12 will produce false code.

1.12 Running SuperOptimized ACE programs

```
Running SuperOptimized ACE programs
-----------------------------------
Since the ACE SuperOptimizer reads an assembly source file and writes an
optimized version of it, all optimized files have to be assembled and linked.

 eg.
  SuperOptimizer 12 Shuttle.s Shuttle_Opt.s: produce the optimized assembly ↩
     version.
  ACE:bin/A68K Shuttle_Opt.s                : assemble it.
  Then type in the following line to make a ready to run executable.
  ACE:bin/blink Shuttle_Opt.o LIB ACELIB:startup.lib+ACELIB:db.lib+ACELIB:ami.lib
```

1.13 SuperOptimizer limitations

```
SuperOptimizer limitations
--------------------------

It highly depends on the output of the ACE compiler and has been tested
with V2.35 and V2.37 of ACE.
```

Since the SuperOptimizer has been specially designed to optimize assembly
source files produced by the ACE BASIC compiler, it will not work with
'alien' assembly source files.
Don't even try to SuperOptimize a SuperOptimized assembly source!

Also take note that the SuperOptimizer has been written with ACE's own peephole
optimizer in mind.
So, you must enable the ACE peephole optimizer because optlevel 12 will
definitely produce false code! Should be rather safe with lower optlevels.
Try it by increasing the optlevel with 1 at a time.

Even if the performance of the SuperOptimizer will not deteriorate when assembly
comments are included, the performance of ACE's peephole optimizer will.
So, you should also disable the inclusion of assembly comments for best
performance.
Don't be surprised...the SuperOptimized assembly source won't contain any
comments. They are simply removed to spare memory!

The current version cannot optimize assembly source files that contain more
than 20.000 lines.
This however is not a real limitation since large programs can be split into
separate modules that can be linked together.

The SuperOptimizer will no longer work on KS1.3 machines because I'm
using FPuts for writing the optimized source.

1.14 Future versions

Future versions

The SuperOptimizer will be completely redesigned!

This will take some time, so be patient...

However, if you find bugs in this version you should contact me, so I
can update the SuperOptimizer.

1.15 A note to PD libraries and reviewers

A note to PD libraries and reviewers

I hope that you will find this program good enough to include it in your
library or to give it a small review.

I'd appreciate it if you would check with me (if possible) to ensure you
have the latest version of the ACE SuperOptimizer, before including it in
your library or reviewing it for a magazine.

Lastly, the wider the ACE SuperOptimizer travels the happier I am, so I'm

pleased to see it turning up in the odd PD library.
Please note however that I don't wish people to profit financially from the distribution of it. You may charge a fee which covers the cost of the disk and the copying thereof, but no more.

1.16 Disclaimer

Disclaimer

Although every care has been taken in the development and testing of the SuperOptimizer, the author will not be held liable for damages caused either directly or indirectly as a result of the use of the SuperOptimizer.

1.17 Contacting the author

Contacting the author

I am contactable via snail-mail and telephone only:

Manuel ANDRE
Arbeidersstraat Nr. 9
2600 BERCHEM
Belgium

Tel.: 32-3-230-66-10
\/
|
-->Belgium.

Direct modem to modem access possible, but only on special request.

1.18 Final word

Final word

Let me offer my thanks to David Benn, he deserves an ACE!
I would also like to thank Michael Sinz for his truly wonderfull Enforcer.
It helped a lot to get rid of all(?) bugs.

I hope you enjoy the ACE SuperOptimizer and find it useful.
But what I REALLY want is feedback. If you have any problems, requests, queries or suggestions, I want to hear from you.

Happy Optimizing!

Regards, Manuel ANDRE
17 May 1996