Frodo

Christian Bauer

COLLABORATORS

	TITLE : Frodo		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Christian Bauer	January 18, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Frodo

1.1 Frodo Manual

Frodo
Version 2.0

What's new?
A Commodore 64 emulator
for Commodore-Amiga computers

1.2 What's new?

The most important changes to version 1.5:

- Settings can be saved into and loaded from arbitrary files
- Integrated a simple
  assembler/monitor
  - There is a speed display and (for 68060 owners) a speed limiter
- The graphics card handling has been completely redesigned
- .d64 and x64 files can be used directly (read-only)
- PlaySID can be used for sound output
- There is now a DeliTracker player for the SID card
- The right mouse button no longer quits the emulation but displays a menu
- Includes the single-cycle emulation

Frodo SC
Details are given in the
revision history
.

1.3 Overview

"Ú-queta i yéni avánier alye,
ú-queta i cirya ná vanwa!"

Frodo is a multitasking C64 emulator for the Commodore Amiga and the world’s first C64 emulator not bearing a "64" in its name. :-) (No, it has absolutely nothing to do with frodo.hiof.no, that’s a pure coincidence.)

Frodo was especially developed to reproduce the graphics of games and demos better than the existing emulators for the Amiga. Therefore Frodo has relatively high system requirements: The optimum configuration is a 68040/68060 and a Zorro-III graphics card. But on the other hand, Frodo can display raster effects correctly that only result in a flickering mess under "A64".

An Amiga with at least OS2.1, a 68020 processor and 2MB RAM is required. Frodo can use the built-in Amiga graphic capabilities as well as any graphics card whose modes are selectable with the standard screenmode requester.

Using a graphics card is strongly recommended as the emulator’s routines have been designed for a chunky display memory. It is possible to run the program on a stock A1200, but you won’t have any fun with it. There is, however, a special monochrome mode that has no color but is a lot faster than the full emulation. Using a 68060-50 with CyberVision gives about 140-160% of the original C64 speed in standard text mode with the emulation set for highest precision.

The program is not a 100% emulation (is there such?), the main missing things are:

- Only line-by-line raster effects possible
- Neither Amiga printers nor RS232 usable
- No emulation of a REU (RAM expansion unit)
- No snapshots possible

But the next version is coming real soon now... :-)

1.4 Installation

The archive contains the following files:

README	- Short description
Frodo	- Frodo main program
Frodo SC	- Frodo SC main program
Frodo.guide	- The german version of this manual

```
  Frodo_E.guide      - This manual
  Frodo Prefs        - Preferences file
  FrodoSIDCard.lha   - Archive with information on the
                SID card
                  Basic ROM.info     - Icon for Basic ROM
  Kernal ROM.info    - Icon for Kernal ROM
  Char ROM.info      - Icon for Character ROM
  #?.info            - Various icons
  Catalogs/#?        - Language files
  64prgs/#?          - Some sample programs
  src/#?             - Source
```

All you have to do is to unpack the archive somewhere onto your hard disk. The files in "Catalogs" don't need to be copied to LOCALE:, they can stay where they are. Frodo doesn't need any assigns, but copies of the original C64 ROMs are required. For legal reasons these are not included in the archive and you have to get them elsewhere.

Frodo looks for three ROM files named "Basic ROM", "Kernal ROM" and "Char ROM" in the same directory Frodo is in. If you already have ROM files for "A64 V2.0" (not the *.data files, these are not the real ROMs), you can simply rename them and use them with Frodo. Otherwise, they can be read from a real C64 by means of the following program:

```
  10 OPEN 1,8,2,"BASIC ROM,S,W"
  20 FOR X=40960 TO 49151:PRINT#1,CHR$(PEEK(X));:NEXT
  30 CLOSE 1
  40 OPEN 1,8,2,"KERNAL ROM,S,W"
  50 FOR X=57344 TO 65535:PRINT#1,CHR$(PEEK(X));:NEXT
  60 CLOSE 1
  70 POKE 56334,0:POKE 1,51
  80 FOR X=16384 TO 20479:POKE X,PEEK(X+36864):NEXT
  90 POKE 1,55:POKE 56334,1
 100 OPEN 1,8,2,"CHAR ROM,S,W"
 110 FOR X=16384 TO 20479:PRINT#1,CHR$(PEEK(X));:NEXT
 120 CLOSE 1
```

Unfortunately, Frodo has no function to get these files saved on a C64 disk to the Amiga, but you can transfer them via a null-modem cable or using A64 or any of the 1541-to-Amiga transfer utilities. (Frodo can indeed access C64 disk drives, but to start Frodo you must already have the ROMs %-).

For sound emulation, Frodo uses the "6581sid.library" of A64 V3.0 or the "playsid.library" of PlaySID V3.0 with are likewise not included, but can be obtained via Aminet.

1.5 Start/quit

After a double cklick on the icon, the window for the emulation

settings appears.

The actual emulation is started by a click on "OK". Then, at first a black screen should appear followed by the C64 startup message after a short moment.

You can quit from the running emulation by selecting the "Quit" menu item.

1.6 Settings

The settings window appears directly after starting Frodo, or by pressing F9 or selecting the "Settings..." menu item in the running emulation.

"Cycles per line (CPU)" and "Cycles per bad line (CPU)" control the relation of graphics speed and processor speed. The higher the values, the faster the emulated CPU, but the slower the graphics refresh speed. If a program is showing flickering lines or graphical flaws you should try to slightly alter both values.

With "Cycles per line (CIA)" you can control the speed of the CIA timers. Entering a higher value increases the frequency of cursor blinking and key repeat. Some programs don’t run correcly with the default value (e.g. "Ballblazer" which needs a value of 65).

The settings for the three "cycles" coming closest to an original PAL C64 are (63,23,63).

Under "Every (n)th frame" you can select if Frodo should skip frames when displaying the C64 graphics. The normal setting is "1", that is, every frame (every simulated raster beam sweep) is recalculated. If you change this to "2", for example, then only every second frame is calculated, immensely speeding up the display, though some raster effects may look a bit jerky.

"Joystick on Port 1/2" relates to the Amiga ports. Normally, you will want to have port 1 switched off if you have your mouse connected there. With "Swap joysticks" you can swap the assignment of the Amiga ports to the C64 ports without having to plug out and in your joysticks. E.g. if a C64 game is using a joystick on port 1 you can simply activate "Swap joysticks" and use a joystick in port 2 to play the game.

The keyboard layout normally matches that of a german keyboard (Y and Z swapped). With "American keyboard" you can prevent the swapping (preferable, if you are not using a german keyboard).

The gadget "Sprite collisions" determines if collisions between sprites and between sprites and graphics should be detected. In monochrome mode, collisions are always turned off. Turning off collisions will make you invincible in some games (sadly, your enemies are likely to become invincible, too :-/.

With the setting "Fast Reset" you can disable the memory test on C64 reset. On the Amiga, the memory test is not necessary and the reset procedure (F10) becomes much faster when it is disabled.

In the lower left corner of the screen, Frodo always displays how much percent of the speed of a real C64 the emulation achieves. When the field "Limit speed" is active, the emulation is slowed down when the speed exceeds 100%. If you set the value in "Every (n)th frame" so that the speed is just over 100% and activate the speed limiter, the emulation always runs at the original C64 speed, with the highest possible precision.

The setting "Screen type" determines the way Frodo controls the graphics. "8 bit RTG" is the most compatible setting that must be used for graphics cards (it should run on all cards). If you don't own a graphics card, you should select the faster setting "4 bit c2p4" instead (which, however, doesn't work on graphics cards). And then there is "1 bit Amiga", a particularly fast monochrome mode without sprite collisions (useful for "Elite", for example) which also doesn't work on graphics cards. After choosing the screen type, you can select the screen mode to be used with "Screen mode". These settings can not be altered in the middle of the running emulation.

The field "Allow direct video access" only has a function when using the CyberGraphX software. If it is checked, Frodo will write directly into the graphics card's display memory without any arbitration, if possible. This makes the emulation about 20% faster, but you should make sure that no other program is using the graphics card at the same time.

With "SID emulation" you set the type of sound emulation to be used. "None" disabled all sound, "SID card" uses the SID card , "6581sid.library" the library of A64, "playsid.library" that of PlaySID. For the latter two settings, the appropriate libraries must be present in LIBS:. If you don't own a SID card, "playsid.library" produces the highest sound quality, but with a higher demand of processing time than "6581sid.library".

Next to "Drive 8" to "Drive 11" you select how the disk drives 8, 9, 10 and 11 are to be emulated (see also here). There are three choices: "Dir", "D64" and "IEC". "Dir" emulates the drive in a directory of the Amiga file system. The name and path of the directory must be given in the string gadget next to the drive number (with a click on the right symbol you can also do this more comfortably with a file requester). For "D64", Frodo accesses a .d64 or x64 file containing an image of a complete C64 disk. The name of the file can either be entered into the string gadget or selected with the right symbol. When "IEC" is chosen, the belonging device number is accessed over the IEC cable as a real C64 device. If you have, for example, a real 1541 with drive number 8 connected to your Amiga, you have to set drive 8 to "IEC" to use the real 1541 (instead of an emulated one).

When the symbol "Use IEC for other devices" is active, Frodo uses the IEC cable also for device numbers 4 to 7 and 12 to 15. So, if you have connected a

```
                C64 printer
                 with device numer 4, you should click on this
gadget to use the printer. If this gadget is not activated, all device
numbers except 8 to 11 give a "device not present error".
```

With "'/' -> '\' in file names" you control whether the '/' in C64 filenames will be translated to '\' and vice versa. The '/' character is used to access subdirectories on the Amiga, but as the C64 doesn't have subdirectories, it's a valid part of a C64 file name. This is a problem if a program wants to create a file with '/' in it as AmigaDOS would interpret the part before the '/' as a directory name and, finding no such directory, would return an error and the open would fail. Now simply activate this gadget and all '/'s will transparently be translated into '\', so in directory listings the '/' will still appear. If you turn off this gadget, you can of course use the '/' to access files in Amiga subdirectories from the C64.

By clicking "Save" the settings are saved on disk to be reloaded next time you start Frodo. "OK" will start the actual emulation (resp. return to it) and "Cancel" will discard your changes to the settings and quit Frodo (resp. discard the changes and return to the emulation).

With the menu items "Open..." and "Save As..." you can save the settings to arbitrary files. When you double-click on such a settings file, Frodo is started with the settings saved in it.

1.7 Keyboard layout

The keyboard layout closely resembles that of a C64, but the "Y" and "Z" keys are normally swapped (german keyboard layout). This can be prevented by clicking on "American keyboard" in the settings window.

The individual rows of the keyboard are mapped as follows (american keyboard, those with other keyboards may want to have a look at the german manual for the mapping of the two extra keys to the left of the Return key and the right of the left Shift key):

Option "American keyboard" off:

```
<- 1 2 3 4 5 6 7 8 9 0 + - £     [ ] / *
    Q W E R T Z U I O P @ *            -
    A S D F G H J K L : ;        <Joy> +
     Y X C V B N M , . /
```

Option "American keyboard" on:

```
<- 1 2 3 4 5 6 7 8 9 0 + - £     [ ] ^ =
    Q W E R T Y U I O P @ *            -
    A S D F G H J K L : ;        <Joy> +
     Z X C V B N M , . /
```

In addition, the following keys are used:

```
 Esc    - RUN/STOP
```

<-	- INS/DEL
Del	- CLR/HOME
Help	- RESTORE
Return	- RETURN
Enter	- RETURN
Amiga	- C=
Alt	- C=
F1-F8	- F1-F8
F9	- Opens the settings window
F10	- Resets the C64

So the famous key combination RUN/STOP-RESTORE corresponds to ESC-Help on the Amiga keyboard. But you don't have to thrash the Help key the same way as the RESTORE key on the original C64. :-)

The cursor (arrow) keys can be used as on the Amiga. I.e. "cursor up" corresponds to "Shift-cursor down" on the C64. The same applies to the function keys F2, F4, F6 and F8.

The numerical keypad is emulating a joystick in port 2 (with "Swap joysticks" one in port 1):

```
  7    8    9
       ^
       |
  4    5    6
 <-- Feuer -->
       |
       v
  1    2    3

  0
Feuer
```

1.8 File access

Frodo offers three possibilities to simulate 1541 drives with the device numbers 8, 9, 10 and 11:

1. Amiga directory, setting "Dir"

In this mode, the C64 programs and files are stored in a directory on your Amiga hard disk and can be LOADed and SAVEd as usual from the emulator. The paths to the directories of the simulated drives are given in the settings window under "Drive 8" to "Drive 11".

You can even load the directory with LOAD"$",8. All files marked as executable are displayed as "PRG", all the others as "SEQ". The file type "DIR" marks an Amiga subdirectory. To get into it you have to open the settings window and change the path for the drive, as the operating system of the C64 doesn't know about subdirectories. Though, it is possible to use a "/" in the C64 file name to access these subdirectories (e.g. LOAD"GAMES/ELITE",8), unless the '/'

translation setting is turned on.

For the opening of files, the file types "PRG" and "SEQ" and the access modes "R", "W" and "A" are supported. Wildcards (*,?) can be used, but you have to remember that files in Amiga directories have no specific order and the result of LOAD"*",8 is rather random. Files are always overwritten even if they are not opened with "@:". Floppy commands, relative files and direct block access are not implemented, you can however read the error channel.

2. .d64/x64 image file, setting "D64"

Most C64 programs available on the Internet and on CD-ROM, expecially demos and games, are stored in files with the ending ".d64". Such a file holds all 683 blocks of a complete side of a 1541 disk, so that direct block accesses are possible from within the emulation. However, Frodo only supports read accesses.

Apart from .d64 files, Frodo can also use image files of the "x64" emulator, automatically detecting the file type.

3. Connecting a real 1541, setting "IEC"

For connecting real C64 drives you need a special cable to hook them to your Amiga. Drives connected this way can be used like from the C64, but speed loaders don’t work.

1.9 Printers

Frodo can’t use Amiga printers, but you can connect a C64 IEC printer using the IEC cable that is accessed as usual under the C64. You have to activate the gadget "IEC für andere Geräte verwenden" in the settings window to use the printer.

1.10 SAM

Frodo has a built-in machine language monitor that can be activated at any time by selecting the appropriate menu item: SAM (Simple Assembler and Monitor). It provides full access to the memory and hardware of the emulated C64.

SAM is controlled by a command-based interface, all numbers are in hex. With the command "h" you can display a list of all commands. The command "x" quits SAM and returns to Frodo.

All commands accessing the C64 memory use the memory configuration set with the "k" command. On starting up SAM, the configuration is set to the one the processor is in. Accesses from within SAM have the same effect as accesses of the emulated processor. This affects especially the I/O registers: a memory dump of $dc00-$de00 clears pending CIA interrupts as SAM reads from $dc0d and $dd0d. With the "v" commands, you can examine the state of the I/O chips without modifying the state of the emulation.

In all places where SAM expects a number (except in the assembler) you can also enter an expression of hex numbers containing '+', '-', '*', '/' and parens. Commands that create a longer output can be interrupted with CTRL-C.

Here is a description of all commands ('[]' marks a parameter than can be left out, '{}' marks a parameter that can be repeated many times. If a [start] parameter is left out, SAM continues at the address where the last command stopped):

```
 a [start]         Assemble
```

starts the assembler at the address "start". SAM always prints the address where the next instruction will be written to. The syntax of the instructions conforms to the standard except for shift/rotation instructions in the "accumulator" addressing mode. Those have to be entered without operand, e.g. "lsr" instead of "lsr a". Entering a blank line quits the assembler and returns to the command mode of SAM.

```
 b [start] [end]   Binary dump
```

displays the memory from "start" to "end" byte-wise binary. With this command, you can view character sets.

```
 c start end dest  Compare memory
```

compares the memory in the range from "start" to (and including) "end" with the memory at "dest". The addresses of all different bytes and the total number of differences (decimal) are printed.

```
 d [start] [end]   Disassemble
```

disassembles the memory from "start" to "end". Undocumented opcodes are markes with a star '*'.

```
 e                 Show interrupt vectors
```

shows the interrupt vectors of the 6510 and (if the Kernal ROM is mapped in) of the Kernal.

```
 f start end byte  Fill memory
```

fills the memory in the range from "start" to (and including) "end" with the value "byte".

```
 i [start] [end]   ASCII/PETSCII dump
```

shows the memory from "start" to "end" as ASCII/PETSCII characters.

```
 k [config]        Show/set memory configuration
```

"k" without parameters shows the memory configuration that is set for SAM, "k" with parameter modifies it. On exiting SAM, the configuration set with the processor port is reactivated. The 8 possible configurations are:

#	$a000-$bfff	$d000-$dfff	$e000-$ffff
0	RAM	RAM	RAM
1	RAM	Char ROM	RAM
2	RAM	Char ROM	Kernal ROM
3	Basic ROM	Char ROM	Kernal ROM
4	RAM	RAM	RAM
5	RAM	I/O	RAM
6	RAM	I/O	Kernal ROM
7	Basic ROM	I/O	Kernal ROM

```
 m [start] [end]   Memory dump
```

displays the memory from "start" to "end" as hexadecimal numbers and ASCII characters.

```
 n [start] [end]   Screen code dump
```

displays the memory from "start" to "end" as ASCII characters, interpreting each byte as a screen code of the standard character set.

```
 o [file]          Redirect output
```

When a file name is specified, all following output is redirected to this file. Entering "o" without parameters closes the file and directs the output into the window of SAM again.

```
 p [start] [end]   Sprite dump
```

displays the memory from "start" to "end" binary with three bytes per line. With this command, you can display sprite data.

```
 r                 Show 6510 registers
```

shows all 6510 registers and flags and the instruction at the address specified by the program counter. "DR" and "PR" are the data direction register and data register of the processor port.

```
 t start end dest  Transfer memory
```

transfers the memory from "start" to (and including) "end" to "dest". Source and destination may overlap.

```
 vc1               View CIA 1 state
```

shows the state of CIA 1 ($dc00).

```
 vc2               View CIA 2 state
```

shows the state of CIA 2 ($dd00).

```
 vs                View SID state
```

shows the state of the SID.

```
 vv                View VIC state
```

shows the state of the VIC.

```
 x                 Return to Frodo
```

quits SAM and returns to Frodo.

```
 : addr {byte}     Modify memory
```

writes the space-separated values "byte" into memory starting at "addr".

```
 ? expression      Calculate expression
```

calculates the value of the given expression and displays it in decimal and hexadecimal.

1.11 The SID card

"Fanuilos, le linnathon."

Though Frodo has a sound emulation over the Amiga's hardware (by means of the "6581sid.library") it can access a real C64 sound chip (SID) on a slot card in the Amiga for the real sound experience. This card maps the SID into the address range starting at $a00001 (odd addresses).

The schematics, a DeliTracker player and additional information can be found in the archive "FrodoSIDCard.lha".

1.12 The IEC cable

> "This is Ohm of Borg.
> Resistance is voltage divided by electric current." :-)

The best way to transfer your old C64 software to the Amiga is to connect a real C64 disk drive. To do this, you need a special cable with a small circuit that, together with Frodo's software, will expand your Amiga by the so-called "IEC" bus system that is used on the C64 for connecting disk drives and printers.

If you already have such a cable for the "A64" you most probably can also use it with Frodo (untested). The cable described in the german "Amiga Magazin" issue 8/88, p.36 is not usable with Frodo as it has inverted output lines.

Building an IEC cable is fairly simple, it connects the parallel port of the Amiga with the round 6-pin connector on the C64 devices.

```
 Parallel port                      IEC connector
    Amiga                               1541
 DSUB, 25-pin                        DIN, 6-pin

  Pin  Name                           Name  Pin
   14  +5V ------------*--*--+
                       |  |  |
              7406    |¯||¯||¯|
            (74LS05)  | || || | 1k each
                      |_||_||_|
              1|\ 2    |  |  |
    5  PB3 ----| O-----*--+--+------- ATN    3
               |/         |  |
                          |  |
              3|\ 4       |  |
    6  PB4 ----| O--*-----*--+------- CLK    4
               |/   |        |
                    |        |
              5|\ 6 |        |
    7  PB5 ----| O--+-*------*------- DATA   5
               |/   | |
                    | |
    8  PB6 ---------+ |
                      |
    9  PB7 -----------+

   25  GND -------------------------- GND    2
```

The 7406 (a 74LS05 will do as well) must be connected to +5V and GND of course. The trained technician will notice that this is the same circuit found inside the real C64.

[Important note: If you plan to build and connect such a cable, please remember that you do it on your own risk. I will not take any responsibility if there is blue smoke coming out from the back of your computer!]

1.13 The single-cycle emulation 'Frodo SC'

"Frodo SC" is a special version of Frodo that doesn't work with a line-based emulation but instead with a cycle-based one. That means that the emulator switches between 6510 and VIC in every emulated ø2 clock phase. By doing this, Frodo SC achieves an extreme precision (nearly all $d011 and $d016 effects can be emulated), but at the expense of speed.

In the settings options, Frodo SC differs from Frodo in only a few points:

- Frodo SC requires a 68060 and a graphics card to run at a halfway descent "speed" (the maximum is about 1/4 of the speed of an original C64)
- The only supported screen type is "8 bit RTG"
- The "cycles per line" settings have no effect as the timing of Frodo SC is hardcoded

Apart from that, Frodo SC is operated in the same way as Frodo and also uses the same settings. Frodo SC has only a few incompatibilities to a real C64:

- The current version of Frodo SC has neither sprite display nor sprite collision detection, but the sprite timing is correct
- The undocumented opcodes RRA and ISB have no decimal mode
- Some opcodes like $8b/$ab don't create the same (partly unpredictable) results as on a real 6510
- BRK cannot be interrupted by IRQ/NMI
- The CIA timers start three cycles and stop two cycles too early
- Color register modifications are visible 7 pixels too late
- On BA low and AEC high, the VIC always reads $f in D8-D11
- The sprite data fetch ignores the state of BA

1.14 Sample programs

The directory "64prgs" contains some little sample programs to show the capabilities of Frodo/Frodo SC. Except for "dadb" and "lrborder", they are all loaded with LOAD"<name>",8,1 and started with SYS49152. To do so, you have to set "Drive 8" in the settings to "Dir" and enter '64prgs' so Frodo will find the programs.

The programs were written by Pasi Ojala, Marko Mäkelä and Andreas Boose. If you like, you can try them on other emulators or on a real C64.

A short description of each program:

3fff : Opens the top and bottom border and displays swinging letters ("Zyklen pro Zeile (CPU)" should be set to 60 for this one)

colorbars: Flickering colors

```
  d011h3   : 8-way soft scrolling without moving a single byte in
             memory (cursor keys to scroll left/right, 't' for text mode,
             'g' for bitmap mode), requires Frodo SC
  de00all  : A program running in the address space $de00-$dfff (press
             space), requires Frodo SC
  dycp     : Scrolling with different Y character position
  fld      : Demonstrates the FLD effect (flexible line distance)
  lrborder : Opens the left/right border, requires Frodo SC
  sprsync  : Stable raster routine by synchronizing with a sprite,
             requires Frodo SC
  stretch  : Variably expanded sprites, requires Frodo SC (altough
             Frodo SC currently has no sprite display, you can see that
             the timing is correct)
  tech-tech: Horizontal scrolling with large amplitude (use joystick
             in port 2 to control)
  text26   : Displays (nearly) 26 lines of text
```

1.15 Technical info

"Known?" said Gandalf.
"I have known much that only the Wise know, Frodo."

In contrast to A64 that is biased towards the Amiga hardware, Frodo tries to exactly imitate C64 hardware features. Now the 64's hardware (esp. the graphics chip "VIC") has a rather simple design resulting in many of the internal processes coming to the "outside". So there are lots of "undocumented features" you can do effects with the designers never dared to dream about.

Frodo uses a line-by-line emulation, i.e. the function of the VIC and the processor (6510) are emulated for one raster line of the C64 screen at times. In practice, Frodo runs VIC and 6510 alternately for 63 simulated cycles each (corresponding to one raster line). At first, it emulates the processor for 63 cycles, then switches over to the VIC that paints one pixel row to the screen, then again 63 cycles processor, and so on... With the exception of monochrome mode, all graphics is internally calculated for a chunky mode display. The Amiga mode uses a chunky->planar conversion to display the chunky pixels on screen.

Even though this is a heavy simplification of the processes in a real C64, it lets you perfectly emulate many graphical effects possible on the 64, e.g. FLD, FLI, DYCP, hyperscreen and many more. But this method has one big disadvantage: Changes made to VIC registers by the processor in the middle of a raster line will only take effect at the start of the next line. E.g. you can't change the border color in the middle of a line, the color change takes place in the next line. Therefore, very sophisticated techniques depending on the exact position of a register change can't be emulated. For instance, it is no problem to open the top and bottom border, but opening the left and right border is impossible (and therefore not implemented in the emulation).

Frodo SC goes one step further by switching between VIC and 6510 in every cycle and precisely emulating the internal functions. Modifications to

VIC registers become visible immediately in the next clock phase and therefore it can even emulate effects that depent on the exact position of a register change within a raster line, e.g. special FLI routines, opening the left/right border, linecrunch, DMA delay, multiple repeated sprite lines and executing programs in open address spaces ($de00-$dfff) and in the color RAM. The 6510 emulation is also more precise and does the same memory accesses as the real 6510, even the "unnecessary" ones that come from design weaknesses of the 6510 and are not needed for the function of single opcodes (e.g. in an instruction sequence like INX:INX:INX:INX, the 6510 reads every opcode twice).

1.16 The source

For those confused by the numbers: A small description of all modules:

1541fs.#?	- 1541 emulation in Amiga directories
1541d64.#?	- 1541 emulation in .d64/x64 files
6510#?	- The processor
6526#?	- The CIAs (in/output, keyboard, timers)
6569#?	- The VIC (graphics)
6569mono.i	- Routines for monochrome graphics
6581.#?	- The SID (sound)
Boopsi.o	- Part of GadToolsBox
c2p4.asm	- Chunky->planar conversion
debug.#?	- For debugging
Display.#?	- Handling of the emulator screen
ExpTable.bin	- Included by 6569mono.i
FrodoGUI.#?	- GadToolsBox user interface of settings window
Frodo_rev.#?	- Version information
IEC.#?	- IEC bus routines
LocStrings.#?	- For localization, created by CatComp
Main.asm	- The main module
Makefile	- The makefile
Prefs.#?	- Handling of the settings window
SAM.#?	- The monitor "SAM"

The makefile has been written for PhxAss 4.x and the freeware version of DICE V2.06.

1.17 Copyright and distribution

The program "Frodo", this manual and the source code may be freely distributed as long as they remain unchanged (archiving and packing is allowed) and all files are included. You must not make any profit by selling Frodo, especially the price of a disk containing Frodo may not exceed US$ 5,- (or equivalent amounts in other currencies). Please feel free to distribute Frodo over bulletin board systems and networks and as part of shareware/freeware CD-ROMs.

Anyone using this program agrees to incur the risk of using it for himself. In no way can the author be held responsible for any damage directly or indirectly caused by the use or misuse of this manual and/or the program.

The rights on the source code remain at the author. It may not - not even in parts - used for commercial purposes without explicit written permission by the author. Permission to use it for non-commercial purposes is hereby granted als long as my copyright notice remains in the program. You are not allowed to use the source to create and distribute a modified version of Frodo.

The user interface of the program was designed with GadToolsBox
© Copyright 1991-1993 Jaba Development.

Parts of the program are © Copyright 1992-1993 Jaba Development.

1.18 Bug reports

If you find a bug or a misfeature in Frodo, or have an idea how to make some things better, then please drop me a note so I'll be able to improve Frodo in the future. My address can be found here .

I don't think the emulation can be made much faster without changing the concept but I'll be happy to find someone showing me how to do it.

Besides, I'm looking for translations of Frodo.catalog in other languages. An elvish version would be especially welcome. :-)

1.19 Credits

"The Silmaril as lantern light
And banner bright with living flame
To gleam thereon by Elbereth
Herself was set, who thither came."

The following persons deserve special thanks from me as they made a significant contribution to the development of Frodo:

- Frank Wille <frank@phoenix.owl.de>, whose PhxAss was used for developing Frodo
- Matthew Dillon, whose DICE was used for developing Frodo
- Jaba Development, whose GadToolsBox developed Frodo's user interface
- Peter McGavin <peterm@maths.grace.cri.nz>, who wrote the chunky-to-planar conversion
- Andreas Boose <a.boose@ldb.han.de> and Marko Mäkelä <marko.makela@hut.fi>, who provided me with precious information on the VIC and on the C64 in general
- Markus Winklbauer <sysop@ng-box.wwbnet.de>, for the docs for the Picasso card
- Lord Prayer for the docs for the Merlin card
- Teemu Suikki <Teemu.Suikki@lut.fi> for the idea of skipping frames

- The following people for the catalog files:

```
    Polish   : Pawel Soltysinski <polonus@felix.univ.szczecin.pl> and
               Pawel Hernik <spi-ph@srv1.tu.kielce.pl>
    Italian  : Giovanni Gentile <gentile@dsi.unimi.it>
    Swedish  : Marcus Comstedt <marcus@amicus.rydnet.lysator.liu.se>
    Finnish  : Jarkko Lindblad <lindblad@cc.helsinki.fi>
    Danish   : Claus Østergaard <dos@dator.dk>
    Norwegian: Kjell Irgens <kjelli@stud.cd.uit.no>
    Croatian : Marko Sehtl <Marko.Sehtl@public.srce.hr>
    Hungarian: László Török <phoenix@master.fok.hu>
```

- J.R.R.Tolkien for the suggestion for the name of the emulator
- All the people who have sent me suggestions and comments

1.20 The author

My address is:

Christian Bauer
Max-Planck-Str.60
55124 Mainz
Germany

```
EMail:
  bauec002@goofy.zdv.uni-mainz.de  (More reliable but polled
  cebix@ng-box.wwbnet.de           (Polled daily, but slower connection)
```

Questions, criticism, suggestions and
bug reports
are always
welcome. EMail is preferred.

Frodo is not a shareware program, but I won't reject any
gifts. :-)

1.21 History

"Those days, the Third Age of Middle-earth,
are now long past, and the shape of all lands
has been changed."

This emulator started as a player routine for C64 music (therefore the SID card) until I saw A64 V3.0 and was so disappointed of it that I began to write a "real" C64 emulator (around christmas 1994).

```
V0.4  05.Jan.1995
  - First released version

V0.5  10.Jan.1995
  - Starting Frodo on a 68000 correcly gave a requester but then
    the computer crashed. I told you this only works on 68020+ :-)
  - Removed two enforcer hits in the memory configuration routine
  - Sprite cycles are now calculated correctly
  - Switching off a sprite in the middle of its display blanked it
    immediately. Now it is drawn completely.
  - Graphics speed for EGS/Picasso improved by means of a line
    buffer (20% faster), other small optimizations to the graphics
  - VIC interrupt flag is now cleared on every write access to $d019
    (a hack for RMW commands doing two write accesses)
  - CIA timers B are now capable of counting underflows of timer A
    (for "Ballblazer" :-)
  - PLP and RTI clear the break flag
  - Completely new 1541 emulation, 4 drives
  - CPU cycle counter heavily optimized (again 25% gain in speed)

V1.0  25.Jan.1995
  - PHP now sets the break flag on stack
  - HandleRESET erroneously set the I flag (don't trust a data sheet,
    even if it's from the manufacturer...)
  - A RESET will set the VIC bank to 0
  - All illegal opcodes implemented
  - New algorithm for the VIC's VC that is more close to the real VIC
  - The VBlank routines now jump back to Periodic6569 as there was
    a non-existant -1st line
  - TOD clocks implemented
  - ECM+BMM, ECM+MCM and ECM+BMM+MCM now correctly display a black
    screen
  - Access to SID card is now disable-able (!? :-)
  - Some bugs in 1541 emulation corrected
  - Use of IEC devices over an adaptor cable is now possible
  - Selecting the directories for the 1541 emulation can now be done
    with a directory requester
  - Turning off the screen using bit 4 of $D011 is working
  - Setting for "raster frequency" removed
  - Interrupt handling completely rewritten
  - Some internal optimizations
  - During a reset, the CBM80 signature is now only overwritten if
    it is present
  - Frodo now works on non-AGA machines
  - Sprite-sprite collisions implemented
  - Swapping of Y and Z keys is now a settings item

V1.1 was an internal test version with bitplane graphics
     (removed again in V1.2)

V1.2  09.Feb.1995
  - Average graphics speed is now higher, especially in multicolor and
    bitmap modes
  - New, very fast monochrome mode
  - Sprite routines are more efficient. No processor time is wasted
    when all sprites are off.
  - Sound emulation with 6581sid.library when SID card is disabled
  - Minimal screen resolution is now 384$\times$272
  - Sprite collisions can be turned off
  - 1541 emulation now handles sequential files and access modes and
    ignores drive numbers and the '@'
  - Frodo is now multilingual by means of "locale.library"
  - Overscan resolutions can now be used
  - Modes like PAL:Lowres are now selectable under OS2.x
  - Option to translate '/' in file names to '\' and vice versa
  - Joysticks are now only polled in the VIC vertical blank routine
  - Amiga graphics routines now also use double buffering
  - Option to accelerate the memory test on reset

V1.3  22.Feb.1995
  - Bug in Close1541 corrected (could crash on quit)
  - CyBERgraphics support
  - 1541 emulation now handles wildcards
  - Option to skip frames implemented

V1.3a 04.Mar.1995
  - Corrected a bug with opening the prefs window with F9 on the
    Picasso card

V1.4  05.May.1995
  - Pause6510 and Resume6510 now use signals
  - Keyboard routines improved, the option "American keyboard" will
    now map "=" and "^" to the numerical keypad
  - Sprite routines completely rewritten, sprite graphics priorities and
    collisions are now taken into account
  - Branches over $ffff/$0000 work
  - Joystick emulation with numerical keypad
  - Polish and italian catalog files

V1.4a 12.Mai.1995
  - Bug in joystick routines corrected
  - Danish, finnish and swedish catalog files

V1.5  06.Jun.1995
  - Merlin graphics card support
  - Norwegian catalog files

V2.0  12.Dec.1995
  - Completely new graphics handling, removed all graphics card specific
    code
  - More precise emulation of the DEN bit in register $d011
  - Small optimizations to the VIC and 6510 routines
  - On startup the RAM is initialized (almost) like on the real C64
  - 100% correct ADC/SBC in decimal mode
  - Better emulation of addresses 0 and 1
  - Better NMI triggering
  - Fixed CIA interrupt handling
  - Pressing F9 could cause pending interrupts to be lost
  - 1541 Open routine checks for file name buffer overflow
  - Closing 1541 channel 15 closes all open files
  - Read-only .d64 and x64 disk image file support
  - Four types of SID emulation: None, the SID card, 6581sid.library
    and playsid.library
  - Menus for the prefs window and the emulation window
  - Loadable/saveable preferences, prefs files are Frodo's "projects"
  - Integrated assembler/monitor (SAM)
  - Speedometer and speed limiter
  - Corrected the SID card schematics and included a DeliTracker player
    module for the card
  - Hungarian catalog files
```

1.22 The future

```
  "Ai! Palan ú ná metta Eldaloaron!"

These are some of the things on planned for the next versions:
 · Possibilities to save and restore the state of the emulation
   (memory and register snapshot)
 · Usage of Amiga printers and modems (rather unlikely, as Frodo has
   not been written to run PrintShop on it :-)
```