

RI

Conversion program

COLLABORATORS

	<i>TITLE :</i> RI		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Conversion program	October 9, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	RI	1
1.1	Overview of RI CopperFX Lib V2.5	1
1.2	RI CopperFX Lib V2.5	1
1.3	RI CopperFX Lib V2.5	2
1.4	RI CopperFX Lib V2.5	2
1.5	RI CopperFX Lib V2.5	3
1.6	RI CopperFX Lib V2.5	3
1.7	RI CopperFX Lib V2.5	3
1.8	RI CopperFX Lib V2.5	4
1.9	RI CopperFX Lib V2.5	4
1.10	RI CopperFX Lib V2.5	4
1.11	RI CopperFX Lib V2.5	4
1.12	RI CopperFX Lib V2.5	5
1.13	RI CopperFX Lib V2.5	5
1.14	RI CopperFX Lib V2.5	5
1.15	RI CopperFX Lib V2.5	6
1.16	Example Programs	7

Chapter 1

RI

1.1 Overview of RI CopperFX Lib V2.5

Overview

This is a library of commands that assist in setting up custom copperlists for your blitz mode games. It interfaces with the display library and so can only be used in conjunction with CopList objects. The commands in this library insert copper instructions into the custom space in a Coplist object - you must therefore have custom space in your CopList if you want to use them.

Custom space is given to the coplist object during initialisation - it is the last parameter of the InitCopList command.

An extension to this library allows you to use it with your own copper lists - enabling you to have direct control over the viewport (without having to use CopList objects). Note that this feature should only be used by people who know what they are doing - anyone tinkering with it will almost definitely crash their Amiga.

AGA warning: Some of the commands in this library are AGA only (A1200/A4000/CD32). They should not be used on non-AGA machines.

1.2 RI CopperFX Lib V2.5

Statement: CopperReset

Modes : Amiga/Blitz

Syntax: CopperReset coplist#,startline[,ccoffset]

This command sets up the copper library to work on a certain coplist object. It must be used before you can use any of the commands in this library.

coplist# is the number of the coplist you want to effect

startline is the vertical start position to store (for the commands DoColSplit and RedoColSplit).

The optional ccoffset parameter allows you to specify an offset into the custom area of the copperlist as a start position for the library. The ccoffset parameter is given in the form of the number of copper instructions from the start of the custom area.

1.3 RI CopperFX Lib V2.5

Statement/Function: DoColSplit

Modes : Amiga/Blitz

Syntax: DoColSplit cols_adr,numlines,colour_register

This command is AGA only at the moment. What it does is produce a nice aga fade going down the screen. The colours to fade from/to are given in the form of 6 longwords, the address of which is pointed to by cols_adr. The following structure could be used to store the colours:

```
Newtype.colourinfo
    r1.l
    g1.l
    b1.l
    r2.l
    g2.l
    b2.l
End Newtype
```

You would then assign a variable to be of type .colourinfo, and set the colour values in it. It would then be passed to the DoColSplit command using the & operator to pass the address of the variable:

e.g.

```
Deftype.colourinfo cols
cols\r1=0,0,0,255,255,255
DoColSplit &cols,256,0
```

The split will start at the current y counter value (set by CopperReset) and will go on for numlines vertical lines. It will effect the colour register supplied, which may be any aga register. The Y counter will be moved down to the end of the colour split after this command has finished, meaning that you can do multiple splits one after the other easily.

1.4 RI CopperFX Lib V2.5

Statement/Function: RedoColSplit

Modes : Amiga/Blitz

Syntax: RedoColSplit cols_adr,numlines,cc_offset

This command must be used after the DoColSplit. What it allows you to do is quickly update the colour information set up by the DoColSplit command without rebuilding the whole colour split. The parameters are the same except that cc_offset replaces the colour register parameter. For this command to work, you must start it at the same custom address as the DoColSplit was started at. This parameter is for you to pass the address to start the library. An easy way to do this is to store the current cc_offset BEFORE calling DoColSplit:

```
pos.w=GetCCOffset
DoColSplit &cols,256,0
;
; Change colours values in cols variable here!
;
RedoColSplit &cols,256,pos
```

1.5 RI CopperFX Lib V2.5

Statement/Function: CopperEnd

Modes : Amiga/Blitz
Syntax: CopperEnd

This command is used to tidy up the copperlist after you have finished adding custom commands. It is necessary if you're ever executing any WAIT commands (including DoColSplit) after vertical position 255. After this position, extra code is required to make sure the CopList display terminated properly. If you don't use it after going over 255 vertically, you will get screen corruption in your display.

1.6 RI CopperFX Lib V2.5

Statement/Function: CopperInfoBlock

Modes : Amiga/Blitz
Syntax: ad.l=CopperInfoBlock

Returns the address of the internal library information. This command is primarily for debugging by me. The data held within the structure is private, and no assumptions should be made about it by the user of this library.

1.7 RI CopperFX Lib V2.5

Statement: CopperCommand

Modes : Amiga/Blitz
Syntax: CopperCommand copins1,copins2

This command allows you to manually insert copper instructions into the current set coplist object. The copper instruction is given as two words which are stored straight into the coplist.

1.8 RI CopperFX Lib V2.5

Statement: CopperMove

Modes : Amiga/Blitz

Syntax: CopperMove register,value

This command allows you to insert a move instruction into the copperlist. The first parameter should be a hardware register address (given as an offset from \$0), the second should be a value to move into it. The value parameter must be a word.

1.9 RI CopperFX Lib V2.5

Statement: CopperWait

Modes : Amiga/Blitz

Syntax: CopperWait x,y

This command allows you to insert a wait instructino into the copperlist. The horizontal and vertical position to wait for are given by x,y. The copper has a horizontal resolution though, of 4 low resolution pixels, thus your x coordinate will be rounded down to the nearest multiple of 4.

1.10 RI CopperFX Lib V2.5

Statement: CopperSkip

Modes : Amiga/Blitz

Syntax: CopperSkip x,y

This command allows you to insert a wait instruction into the copperlist. The horizontal and vertical position to wait for are given by x,y. The copper has a horizontal resolution though of 4 low resolution pixels, thus your x coordinate will be rounded down to the nearest multiple of 4.

1.11 RI CopperFX Lib V2.5

Function: GetCCOffset

Modes : Amiga/Blitz

Syntax: offset=GetCCOffset

Gets the current custom copper instruction offset. Used if you want to keep track of how far through your custom area you are, or in conjunction with Do/RedoColSplit. The return value is the number of instructions from the start of the custom area.

1.12 RI CopperFX Lib V2.5

Statement: CopperAGACol

Modes : Amiga/Blitz

Syntax: CopperAGACol register, r, g, b

Setting AGA colours is a pain in the arse. This instruction allows you to do it easily in your copperlist by doing all the extra work for you. Just supply the colour register number to move the data into and the r, g, b values. This command generates 4 copper instructions inside your copperlist.

1.13 RI CopperFX Lib V2.5

Statement: CopperResetAdr

Modes : Amiga/Blitz

Syntax: CopperReset copperlist, startline

This command sets up the copper library to work on a user created copperlist. The pointer passed is the location of space inside the copperlist that this library can use to store the commands you specify. Note that no range checking is performed inside your custom copperlist - you must manually ensure that the size allocated will be large enough for your needs.

1.14 RI CopperFX Lib V2.5

Statement: ColSplitBplcon3

Modes : Amiga/Blitz

Syntax: ColSplitBplcon3 bplcon3

This command allows you to control the contents of the lower half of register BPLCON3 during colour splits set up by this library. The value you provide will be placed into the destination copperlist everytime that the copperlist needs to change BPLCON3.

Contents of lower byte of BPLCON3

BIT#	REGISTER	DESCRIPTION
07	SPRES1=0	Determine resolution of all 8 sprites (x=0,1):
		+-----+-----+-----+-----+-----+-----+-----+-----+

	SPRES1	SPRES0	SPRITE RESOLUTION
	0	0	ECS defaults (LORES,HIRES=140ns,SHRES=70ns)
	0	1	LORES (140ns)
	1	0	HIRES (70ns)
	1	1	SHRES (35ns)
06	SPRES0=0		
05	BRDRBLNK=0		"Border area" is blanked instead of color (0). Disabled when ECSENA low.
04	BRDNTRAN=0		"Border area" is non minus transparant (ZD pin is low when border is displayed). Disabled when ECSENA low.
03	X		don't care- but drive to 0 for upward compatibility!
02	ZDCLKEN=0		ZD pin outputs a 14MHz clock whose falling edge coincides with hires (7MHz) video data. this bit when set disables all other ZD functions. Disabled when ESCENA low.
01	BRDSPRT=0		Enables sprites outside the display window. disabled when ESCENA low.
00	EXTBLKEN=0		Causes BLANK output to be programmable instead of reflecting internal fixed decodes. Disabled when ESCENA low.

(Information obtained from AGA.GUIDE)

1.15 RI CopperFX Lib V2.5

RI CopperFX Lib V2.5

©1996 Red When Excited Ltd

Undocumented commands added by Toby Zuijdveld 02/03/1999
mailto: hotcakes@abacus.net.au

Overview

Command Index

ColSplitBplcon3

CopperAGACol

CopperCommand

CopperEnd

CopperInfoBlock

CopperMove

CopperReset
CopperResetAdr
CopperSkip
CopperWait
DoColSplit
GetCCOffset
ReDoColSplit
 CopperResetAdr

Examples

Main Document

Library Index

1.16 Example Programs

Example Programs

EXAMPLE 1 - Filling the screen with 4096 colours using the copper... :

Load Example 1
Compile It!

EXAMPLE 1 - Doing A Copper Colour Fade :

Load Example 1
Compile It!
