# App. F: Copper Lists

# (This feature is not implemented yet)

The Amiga co-processor
While AMOS Professional allows you to harness the power of the Amiga with the greatest of ease, it has to perform
a great deal of work behind the scenes when manipulating entire screens at great speed. The source of much of this
power is a special hardware chip called the "co- processor", or copper.

The copper is in effect a simple micro-processor, with its own separate programs, and its own unique memory
registers. It supports only three instructions, MOVE, WAIT and SKIP, and these commands insert values into the
computer's hardware registers at certain points on the display, which change the way pictures are drawn on the
screen.

These hardware registers hold the values that determine the precise appearance of the display, such as its size and
position, as well as the number of colours. For example, all the colour values used by AMOS Professional screens
are held in the colour registers from $180 to $1BE. Because the appearance of every line displayed on your screen is
controlled by the copper, a massive number of special effects can be created by changing these registers during a
program, using a list of instructions known as the "copper list".

## The Copper List

The copper list is executed automatically, fifty times every second, at the same time that the screen is re-drawn. This
is how the AMOS Professional RAINBOW commands work, waiting for a rainbow line to appear on screen and
then immediately poking a new value into the selected colour register. This causes dramatic colour changes,
depending on the position of the line in the display.

Exactly the same process can be applied to the rest of the display system, and by placing the appropriate value into
certain hardware registers at exactly the right moment, the position, type and size of the display can be changed at
will! Unfortunately, the copper list is notoriously difficult to manipulate, and many competent programmers have
failed to master its mysteries.

Although the copper is automatically managed by AMOS Professional, you cannot expect the system to teach you everything about the inner workings of the Amiga's hardware. Indeed, Francois Lionet has written AMOS Professional to save you the years of hard work and experience needed to gain such expert knowledge. However, for those expert programmers who insist on meddling with the copper directly, AMOS Professional includes a powerful trap- door into the realms of the co-processor. This allows advanced programmers to generate astounding effects, and also allows novices to send their displays berserk and crash their computers. You have been warned!

# Accessing the Copper

## *COPPER OFF*

*instruction*: turn off the standard copper list

```
Copper Off
```

If you ignore the warning in the last paragraph and use this instruction, the automatic copper generation that forms the backbone of the AMOS Professional system is turned off. From now on, you are on your own!

You should now understand that AMOS Professional actually holds two separate copper lists in memory, and the principle is very similar to the logical and physical screens of the DOUBLE BUFFER system.

The logical copper list is the list being created from AMOS Professional Basic, and it is completely invisible. The physical list holds the copper instructions that are generating the current TV display. It cannot be accessed from AMOS Professional at all, as this would corrupt the display completely. As a default, these copper lists are limited to 12k in length, which is the equivalent to approximately six thousand instructions. This limit may be increased using an option from the Interpreter set-up dialogue box.

Copper lists can be defined in one of three ways:

The first method is to enter the copper list using a combination of the COP MOVE and COP WAIT instructions, from AMOS Professional Basic.

The second way is to find the address of the logical copper list, using COP LOGIC. This can then be manipulated directly using DEEK and DOKE, allowing minor modifications to be made to the existing screen without having to generate a completely new copper list at all. This is perfect for the creation of rainbow effects.

The third alternative is for assembly language buffs. Copper lists can be generated using machine code, and as before, the current address is available via the COP LOGIC function. Note that this address will change during the course of a program, and it must be entered every time the machine code routine is called.

## Recommended Procedures

If you want to create copper lists from beginning to end, you must take personal control over the hardware Sprites, the display positioning, the location of screens, and their sizes. You must then ensure that the resulting screens have the correct amount of memory, before loading the appropriate registers with the addresses of the required bitmaps. This can be achieved with the LOGBASE function.

Additionally, if you intend to use DOUBLE BUFFER, a separate copper list must be produced for both the logical and physical screens. Here is the procedure:

- Define the copper list for the first screen.
- Switch copper lists using the COP SWAP command.
- Swap between the logical and physical screens with SCREEN SWAP.
- Define a copper list for the second screen.

Providing that all is well, you may access your screens using -all of the normal AMOS Professional drawing commands, including SCREEN COPY, DRAW, PRINT and PLOT. As well as this, there should be no problems using Blitter Objects.

However, multiple screens and Sprites are only supported by the standard AMOS Professional copper system, so you cannot use SCREEN OPEN, SCREEN DISPLAY, RAINBOWS or any of the SPRITE commands. If you need to generate such effects, you will have to program them for yourself! For those of you who wish to give up now, the following command may be useful.

### *COPPER ON*

*instruction*: re-start automatic copper generation

```
Copper On
```

The COPPER ON command re-starts all standard copper calculations, and returns AMOS Professional back to normal. The experts (and foolhardy) may now continue.

### *COP MOVE*

*instruction*: write a MOVE instruction to current copper list

```
Cop Move address,value
```

MOVE is an internal instruction used by the copper, and it is very similar to the AMOS Professional DOKE
command. It inserts a MOVE command into the current logical copper list, by copying a value from 0 to 65535 into
the selected register address. The address refers to a copper register from $7F to $1BE.

## *COP MOVEL*

*instruction*: write a long MOVE instruction to copper list

```
Cop Movel address,value
```

This is a special option from AMOS Professional Basic, which generates a matched pair of MOVE commands in
the new copper list. These load a 32-bit (long word) value into the selected address, exactly like a normal LOKE
instruction.

## *COP WAIT*

*instruction*: insert a WAIT instruction into copper list

```
Cop Wait x,y,xmask,ymask
```

The COP WAIT command enters a WAIT instruction at the current position in the copper list. WAIT forces the
copper to stop in its tracks until the screen has been drawn at the specified hardware coordinates x,y. The copper
then continues from the next instruction in the copper list.

WAIT is usually called immediately before a MOVE command, creating a pause until the display reaches a specific
screen line. The MOVE instruction is then used to change the attributes of the screen area below this line. Rainbows
are an excellent example of this technique, with each line of the rainbow generated with a pair of commands like
this:

```
Cop Wait 0,Y : Rem Y is starting coordinate of next colour shift
Cop Move $180,$777 : Rem $180 is address of colour 0 and $777 is new colour
```

The x-coordinate is a hardware coordinate from 0 to 448. Since the Amiga is only capable of performing this test
every four screen points, this coordinate is rounded to the nearest multiple of four.

The y-coordinate can be any value from 0 to 312. Normally, coordinates from 256 to 312 require special
programming, but AMOS Professional generates the correct instructions automatically, so there is no need for
concern! Here are some examples:

```
Cop Wait 0,300: Rem Wait for line 300
Cop Wait 12,10: Rem Wait for coordinates 12,10 to arrive
```

The optional xmask and ymask parameters are bit-mask values which allow for a pause until the screen coordinates satisfy a specific combination of bits. The default value is $1FF. For example:

```
```

## *COP RESET*

*instruction*: re-set copper list pointer

```
Cop Reset
```

This command is used to add a pair of MOVE commands, forcing the copper list to re-start from the very first instruction. This may be used to generate simple loops.

## *COP SWAP*

*instruction*: swap logical and physical copper lists

```
Cop Swap
```

The COP SWAP command switches over the logical and physical copper lists. The new copper list will now be flicked into place, and the results will be shown after the next vertical blank period. For example:

```
Cop Swap : Wait Vbl
```

## *COP LOGIC*

*instruction*: give address of logical copper list
address

```
Cop Logic
```

This command returns the absolute address of the logical copper list in memory. It can be used to manipulate the copper list directly from AMOS Professional Basic. Lists can also be generated by using assembly language.