T—Y—N—E—&—W—E—A—R

# TYNE & WEAR

# ATARI 8-BIT USER GROUP

Newsletter of TWAUG

Software
Editorial
Buy & Sell
Hardware

Reviews
Help line
Section
Repair Info

Public Domain Library

ISSUE #2

MARCH/APRIL 1993

U—S—E—R—G—R—O—U—P

# BRING YOUR EIGHT UP TO DATE with power products from COMPUTER SOFTWARE SERVICES

## THE BLACK BOX

The BLACK BOX is an add-on board for the Atari 600XL,800XL and 130XE 8-bit computers. It is a T-shaped board that plugs into the PBI port of the XL computer, or the ECI and cartridge ports of the 130XE. Connectors for both types of computers are built into the BLACK BOX so no adapter boards are necessary. A cartridge port is available on the board itself for 130XE users.

The BLACK BOX provides many unique and useful functions. The four primary functions are:-

* RS-232 serial modem port
* Parallel printer port
* SASI/SCSI hard disk port
* Operating System enhancements

The BLACK BOX is $199.95 for the basic unit, and $249.95 with an onboard 64K printer buffer.
Shipping and Handling extra.

## THE BLACK BOX ENHANCER

A must for all BLACK BOX owners. The BLACK BOX ENHANCER is a plug-in module for your BLACK BOX, enhancing the printer functions and adding an instantly available, full featured sector editor!

Installation of the BLACK BOX ENHANCER requires one simple solder connection. Only $49.95 plus shipping/handling.

## THE FLOPPY BOARD

Our latest and greatest product. The FLOPPY BOARD is an add-on expansion board for the BLACK BOX interface. It allows the use of the same inexpensive floppy drive mechanisms used in IBM computers. The FLOPPY BOARD is the first floppy drive interface to support "high density" floppy drive mechanisms in either 5.25 inch or 3.5 inch. Built into the FLOPPY BOARD are our BLACK BOX ENHANCER and a version of our SUPER ARCHIVER to allow copying of protected disks for 3.5 inch format. Included with the FLOPPY BOARD is our program to read and write to IBM or ST formatted disks. This makes the FLOPPY BOARD the best way to transfer files to and from your 8-bit.

The FLOPPY BOARD is only $149.95 plus shipping & handling.

## THE SUPER ARCHIVER

The SUPER ARCHIVER is not just another double density high speed upgrade. Rather it is the most powerful copying and disk editing tool available for the 8-bit computer. Copies of disks are exact duplicates of the originals, eliminating the need for patches,PDB files, or extra computer hardware.

The SUPER ARCHIVER is only $69.95 plus shipping/handling

## THE SUPER ARCHIVER II

The SUPER ARCHIVER II edits and copies all enhanced density programs plus retains all the features of the SUPER ARCHIVER.

The SUPER ARCHIVER II is only $99.95 plus shipping $ handling. NOTICE: if you already have THE SUPER ARCHIVER you may upgrade to S.A.II for only $29.95 plus shipping/handling. Software only.

## THE BIT WRITER

The Super Archiver BIT WRITER is capable of duplicating even the "uncopyable" Electronic Arts and Synapse Syn-series, which employ 34 full sector tracks. The BIT WRITER must be used with the SUPER ARCHIVER

The BIT WRITER is only $79.95 plus shipping/handling.

**For more information on these and other 8-bit products:**
**CONTACT**

**COMPUTER SOFTWARE SERVICES**
**PO BOX 17660**
**ROCHESTER, NEW YORK 14617**
**USA**

**ORDERING LINE: (716) 429-5689**
**FAX: (716) 247-7158**
**BBS: (716) 247-7157**

**or contact T.W.A.U.G. we will do our best to help**

# TWAUG NEWSLETTER

## EDITORIAL

Who to blame!!!

John Matthewson
David Ewens
Max Gerum

Please accept our apology for the mistakes in the first issue of the TWAUG newsletter. It was mainly due to the holiday period and wanting to release the newsletter by mid-January as promised some of the mistakes had been overlooked.

We had some phone calls from readers congratulating us on the lay out and content of the first issue. We also had promises of continued support and we in turn will do our utmost to publish articles that is of interest to all our readers. We thank you all very much for your support.

If you have friend(s) who would like to place an order for future issues just contact us and we will post an order form to your friend(s). We still can supply the first issue. Back issues will always be available.

This issue of TWAUG was produced with an Atari 65XE upgraded to 1 MEG, 2 Atari 1050 disk drives with US doubler and Happy enhancement in each and printed with a Citizen 120D. The software used was Textpro version 4.54 word processor and Daisy-Dot III print processor. The disk operating system I use is MYDOS 4.50, suitably enhanced for a 1 MEG Ramdisk.

The next issue will be ready by mid-May.

## CONTENTS

T.W.A.U.G.
P.O. BOX No 8
WALLSEND
TYNE & WEAR
NE28 6DQ

# DISK CONTENT

The content of this issue #2 disk is a variety of selected programs. Let's look at the games first of all, there are only three, one is an adventure game, one you must try to catch a chicken, and the third a jigsaw type puzzle for Micro-Painter pictures.

We have also included a times-table printer, it prints the multiplication tables, up to 15 direct to the printer, unfortunately only Epson printers and compatibles can be used. For those readers who haven't got an Epson printer the printout is on the back page. It is a very handy little program for education purposes.

HUSA: You must try and catch the chicken if you want a roast for your dinner. You have to run up ramps and along walls to catch that elusive chicken, but be careful not to fall or jump onto an egg or your chicken will be cooked. The chicken is leaving the eggs lying about all over the place, make sure you don't touch them. I have tried catching that elusive chicken and I was very unlucky, I cooked my own chicken within a few minutes, I mean lost all my lives, but it was fun. Sorry, there are no other instructions available. Use the joystick in port #1.

An Adventure game is on side B of the disk: I cannot tell you much about it, except it is called "GUARDIAN of the SACRED SWORD" The legend speaks of a sacred sword with great power which supposedly is buried in the church vault with the Knight who carried it in the first crusade. A small text file is on this disk but it doesn't say much about it, I am sure you adventurers will be able to solve this puzzle.

MICRO-PUZZLER: This is some sort of a jigsaw puzzle, this program loads a Micro-Painter screen file and divides the picture into 120 pieces, mixes and rotates them, and then it lets you put it back together again. You will find the instructions already printed out for you in this issue. Have fun!!!

REBOKIT.BAS: This is the Rebound Construction Kit, I printed out the full instructions in this issue, to save you having to do it. With this construction kit you can make your own Rebound screens as difficult or as easy as you like. I don't think I need to add anything more as you can read all about it in the printed text.

DISK LABELER: With this DSKLAB22.COM is a second program to alter the printer codes so that it will work with any printer. Load DRIVER22.COM separately with the DOS L Option and answer the prompts.

It is really a very good label printer, it prints on labels from 1" high to 3 1/2" wide or as high as necessary to print out all the titles on disk. You can include PrintShop icons, control the placement of the icon and print a title on the label in double width.

I am actually using this labeler. By adding a different type of icon to the label, I can see at a glance if the disk contains, commercial or public domain utilities. The title on the label is printed in inverse. You will find the documentation on side B of the disk.

DEMOS: There are also two demos on the disk. One demo is from the OL'HACKERS Atari User Group from New York. This demo is called "Dizzi Fingers", at the end of the music it will run a tongue in cheek demo from John E. of AURA.

Enjoy the programmes from this issue.

## Letter Section

It's a hard job to create a newsletter and I'm very pleased to see that there are still people who try to do it.

And you have made a very good job (don't worry about the small mistakes, I'm sure everybody makes mistakes), all is clear and well written, it is in fact great!!!

The only bad thing I found was that you didn't say where to find the hardware spoken about in the articles (like the 1020, or the voicemaster).

Well, now the question :

What do you think of a "demo review" or something like that ? (It probably would take one page to review the demos, new or old, if they are good).

Continental Atari friend. RED

REPLY:

Thank you for your kind letter, it is appreciated very much. Sorry about the mistakes I will try very hard to eliminate these silly mistyping errors in future.

The article on the Atari 1020 plotter was aimed at the 1020 plotter users to give them more info on how to use it to its full capacity. The only place where it may still be available could be from "MICRO DISCOUNT". The COVOX Voice Master may also still be available, but I believe only from the U.S.A..

Yes friend we will accept all articles for publication in this newsletter, as long it is of interest to 8-bit users.

---

Dear Friend

Here are a few lines to thank you for the 1st Issue of TWAUG.

I think it is a great new Newsletter for the Atari 8-bit and I enclose a cheque for £11 to commence with Issue No.2, I am a disk user. If you wish to know of me just ask David Ewens as I've been corresponding with him for a couple of years or more.

I've not filled in your order form as it would disfigure the Newsletter, my suggestion is that you insert a seperate sheet for membership and to order PD disks or whatever.

I would like to say that if the theme of the TWAUG Mag continues, I feel that you could have a winner.

A need for a question and answer section would not go astray. I have a Question: Why do you not mention the XF5517 And would disks formatted on this Double Sided m/c be compatible with the 1050 DD.

All my best wishes, Old Atari 8-bit user. Bill Hall

REPLY:

Thank you for your letter and wishing us success with our newsletter. Yes Bill we realised after the publication that the order form should have been inserted seperately.

As for a Question and Answer section, our intention was to include that section with the letters. If on the other hand a lot of questions are being asked, we would consider opening a seperate section.

Now to your question, why didn't we mention the XF551 drive? We cannot comment on this drive as nobody in our team owns one. We have been told that the XF551 formats a disk in the same manner as the 1050 on Side A and is therefore compatible. But the underside is formatted by the XF551 without having to flip the disk. When Side A is formatted it continues to format the underside from the centre outward. The compatibility stops there as the 1050 cannot read upside down.

Sorry we cannot be of more help, but we are sure some kind Atarian will write an article on this subject.

## Letter Section

Dear Sirs,

I write to congratulate you as a team on the first issue of your new newsletter, the "TWAUG Newsletter". As a 'senior citizen' who has not had the opportunity of receiving formal training in the use of computers, I can assure you that all the information contained in the newsletter was avidly read and much appreciated.

You are to be congratulated on the way that Daisy Dot III was managed to print such a professional-looking booklet. As for the contents of the booklet, I was particularly impressed by Max Gerum's review of DD III and most interested in the details printed of new products for the 8-bit Atari. James Cutler's article on the use of modems was interesting too,-perhaps in future issues he can give us a break-down of the cost involved in going on-line.

I am not all that interested in computer games but Rhys, my 13 year old son found "Dump that Waste" challenging as also is the "Rebound" game. The "Talking Math Wizard" is intriguing and possibly of value as a learning aid of nine/ten year old children.

In future issues I would be most interested if you could let us know in what ways other Atari 8-bit owners find the computers useful in their every-day lives. We all realise that the Atari is great for games but do children also find the Atari useful for their school work and if so in what way?

I mainly use the machine as a word-processor, spreadsheet manipulator and as the holder of a data base. A few years ago I produced a 180 page book (in Welsh) using my Panasonic KX-P1081's resident fonts. Had I known of the existence of Daisy Dot at the time I would have made a better fob of it,-particularly so if I also had access to Max Gerum's expertise.

I wish you well and am already looking forward to the next issue.

Yours faithfully,

(Arthur Ll. Morris).

---

My two colleagues had also letters from the U.S.A. congratulating us on our first newsletter and hoping that we can keep the Atari 8-bit alive. I cannot publish these letters as they are personal.

One letter came from the OL'HACKERS ATARI USER GROUP INC. NEW YORK, they too are receiving our newsletter.

And a letter was received by David from a club in SAN ANTONIO, called AAAUA and they also are reading our newsletter. The characters AAAUA stand for ALAMO AREA ATARI USER ASSOCIATION, they have about 45 local members, so it is a fairly large club.

COMPUTER SOFTWARE SERVICES from Rochester N.Y. are thanking us for the Black Box advert in our newsletter and they're wishing us a great success.

REPLY.

Many thanks for your nice letters, we appreciate your compliments very much.

---

TYNE&WEAR ATARI USER GROUP

# DAISY-DOT III
## USER'S GUIDE

Author David Richardson

Conversion of PSicons to DD3 fonts

by Max Gerum.

Hello I am back with more information, this time on how to convert Print Shop icons to Daisy-Dot 3 fonts. I will also show you a typical setup for printing those converted icons, and how to include a single icon in a text.

Since the publication of our newsletter we've had a number of requests for the Daisy-Dot 3 User's Guide manual. It also came to our notice that this manual isn't widely known and a lot of our readers were suprised what can be achieved with DD3.

After you've read this article and you still have any queries about converting icons just contact TWAUG and enclose a SAE for a speedy reply, and of course, I will try to put you on the right track.

When converting your own Print Shop icons you need two disk drives. If you have a 130XE you can use the RAMdisk, but with a 800XL that hasn't been upgraded you need two drives, the reason, Print Shop format isn't compatible with any of the DOS's I use.

There is a small point I would like to make, when converting your icons I would suggest to limit the icons you wish to convert to 10, you can always convert another set of ten in another session. You see 30 converted icons use up 301 sectors, and every time you want to make a printout, even if it is only one character, DD3 needs to access each font from the disk and load the entire file. You really would put your drive to task, it would be running non-stop for the duration of the printing and the printout time would be extremely long. On the other hand, if you can use the RAMdisk that would eliminate the heavy task to your drive and the printing would be much faster. I do all my printing via the RAMdisk, but I am lucky the RAMdisk I've got has over 8000 sectors.

Please also note, any icons you wish to include in any font set must be converted in the same session. You cannot put 9 icons in a font, exit the program and then decide to add some more. Neither can you delete an icon from the font, you would have to restart the conversion.

If however you decide to convert some icons, choose your icons carefully, copy them to a Print Shop formatted disk and make a printout of the directory. You must make sure that you type in the correct filename, the conversion program will ignore any mistyped filenames and will not display error messages. There are a number of PS utility programs available to format your disk for the PSicons, that will copy your icons to it and also print out a directory for you. Next prepare a couple of disks with your favourite DOS, there is no need to write DOS to the disk. If you use a RAMdisk make sure it has been set up. When you've done all that you will be ready to begin the conversion.

There are two BASIC programmes available, both have been produced by John McGowan and are public domain programmes. PSTODD3B.BAS converts each icon after you've entered the filename. PS2DD3.MOD is a modified version and you enter all the filenames up to 30 before the conversion begins.

Go to basic and type RUN"D:PSTODD3B.BAS" and Return.

The first prompt is (Source drive:?) enter the drive number, just type 1 if it is drive 1.

The next prompt is (Destination drive:?) if you are using D8: type 8 and Return, or 2 if your second drive is 2

Next you will be asked for the (Filename for save:?) enter a short filename only without extender as an extender will be added by the programme. I usually give a three or four character filename it is easier to remember when entering these filenames in a text.

The last prompt is (Graphic to convert:?) now you can enter the filename of the icons on the disk and press Return. You will now hear the computer clearing its throat a couple of times and the conversion begins, it will take approximately 10 to 12 seconds for each icon before you enter the next filename.

As I mentioned above 30 icons can be converted in one session, when the program has converted the 30th icon it will exit to BASIC. If on the other hand you only want to convert 10 icons enter the filenames in the same manner, after the last icon has been converted just press CONTROL-Q and Return, the conversion will then end.

Using PS2DD3.MOD is similar to the one above except that all the filenames are entered before the conversion begins. And again if only 10 icons are wanted, the rest of the prompts for the filename must be completed with the CONTROL-Q+Return.

When the conversion is completed you just type DOS, please do remember to swap the Print Shop disk with a DOS disk to read the directory and if you've saved to the RAMdisk copy the fonts to a prepared disk.

Those who know how to magnify fonts with DD3 will know that the font on the disk is a magnified DD3 font. Any of you readers who have never magnified a DD3 font wont know what it does so I will explain. When using the Magnify option of Daisy-Dot 3 you can magnify a font to a magnification factor of (4), this will add the same extender to that font, that you have on your disk after the conversion. Can you see why I said no extender need to be added, the program has added its own extender automatically. If you've given 'IC' as the filename, the directory of your disk will now read: "IC.NLQ, IC.NL2, IC.NL3, and IC.NL4.

These icons now appear as a magnified font it is now not possible to magnify them any more. But they only appear to be magnified by extender only, it is the way the conversion program works, as you can see from the example below, they are printed in normal size. There is a way around that, for an explaination on how to magnify the fonts you must wait for the next issue. Sorry!! Today I want to explain a little of what the conversion has done to the icon and what must be done before any printing can be carried out.

When the icons are converted they are split vertically into three parts, and in order to print these icons they need an identification, like a filename. But being split into three parts, does mean we must add three filenames. John McGowan solved this problem by using the keyboard characters, a printout of the table is below, as you know there are 90 printable characters on the keyboard. When using these characters we must use them in the same order that they are printed out when you use the DD3 Font Editor to print out an entire font. Thus you would add the first three characters to the first icon on the disk, for example (!"#) would represent icon 1 and 012 would represent icon 6. You must also set the spacing to zero otherwise a split icon will be printed.

Before printing can be carried out a word processor must be used to do the setting up. An example is printed below, this is how I typed it in the word processor, saved it to the RAM and then used DD3 to print the icons. This is how the icons are printed, as you can see below, with the setup shown. If you compare these icons with the larger set printed in last issue, you will notice that a

**magnification is possible.**

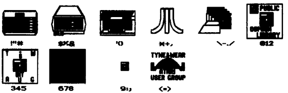
TABLE OF CHARACTERS

| ICON | CHARS | ICON | CHARS | ICON | CHARS |
|---|---|---|---|---|---|
| 1 | !"# | 2 | $%& | 3 | '() |
| 4 | *+, | 5 | \-./ | 6 | 012 |
| 7 | 345 | 8 | 678 | 9 | 9:; |
| 10 | <=> | 11 | ?@A | 12 | BCD |
| 13 | EFG | 14 | HIJ | 15 | KLM |
| 16 | NOP | 17 | QRS | 18 | TUV |
| 19 | WXY | 20 | ZI\\ | 21 | ]^_ |
| 22 | abc | 23 | def | 24 | ghi |
| 25 | jkl | 26 | mno | 27 | pqr |
| 28 | stu | 29 | vwx | 30 | yz \| |

Example set up

```
\FD8:ic\\S0!"# \PL$%& \PL'() \PL*+, \PL\-./ \PL012
\FD8:plain\\S2  !"#\T  $%&\T  '()\T  *+,\T  \-./\T  012
\FD8:ic\\s0345\T678\T9:;\T<=>
\FD8:plain\\s2  345\T  678\T   9:;\T  <=>
```

Printed result.



\L\S0\FD8:ic\<=> \S2\FD8:plain\This is a single icon setup with added text, example below.

TYNE&WEAR
ATARI
USER GROUP

This is a single icon setup with added text, but with this icon only one line can be written after it. You can include this icon anywhere in your text, the only drawback is the wide gap between lines.

See you in the next issue of TWAUG.

# Micro-Puzzler

Hey all you Micro-Painter owners! Have you ever wondered what you could do with all those beautiful screens you created, other than just look at them? Well, now there's Micro-Puzzler! This program will load a Micro-Painter screen file, divide it into 120 pieces, mix and rotate them, and then let you try to put it back together again - much like a jigsaw puzzle. As you may have guessed, the difficulty will be (mostly) determined by the complexity of the picture, so you can choose your own level by the screen you use.

Running the program

When you run the program, there will be a few seconds of initialization, after which you will be prompted by the words ENTER SCREEN FILE-NAME. You may now enter the name of the Micro-painter screen file that you want to use with Micro-Puzzler. If no device specification is given, disk drive 1 is assumed. If an error is encountered in trying to access this file, the program will return to the prompt for the screen filename. Instead of entering the screen filename, you may get a disk directory listing by hitting CTRL-D and then entering the drive number for which you want the directory. After listing the disk directory, the program will return to the prompt for the screen filename.

Once a valid screen filename is entered, the screen will be loaded, and the puzzle pieces will be shuffled and rotated. Then the new mixed-up screen will be displayed, along with the rectangular cursor in the upper left of the screen. Puzzle pieces are moved by exchanging positions of two pieces at a time. To do this, move the cursor (using the arrow keys) to one of the pieces you want to exchange and hit RETURN. Then move the cursor to the other piece to be exchanged and hit RETURN. While your moving from the first to the second piece, a secondary cursor will be left at the first position to mark the piece to be exchanged. After the exchange is made, the secondary cursor will disappear.

The only other type of puzzle piece manipulation that may be done at any time by pressing the R key. This will always rotate the piece within the primary cursor. An interesting and sometimes helpful phenomenon to note is that some of the colours of a puzzle piece may change when the piece is rotated. So if you see a colour that isn't on the original picture, chances are that piece is upside down.

For those of you who dont remember exactly what the original picture looked like, you may press the Atari key to toggle between the original and the mixed-up screen.

Once the picture is correct, you will be congratulated and may then press the ESC key to run again.

To quit, you must hit SYSTEM RESET.

If you get tired of puzzling before you complete the picture, you may save your current status on disk, if desired. To do this press the OPTION key and then enter a disk filename to which the status will be saved. Warning: this file has to be saved to the same disk containing the original screen file. Otherwise, when you try to reload your status, it will not work. To reload, just enter this status filename instead of the original Micro-Painter filename when prompted with ENTER SCREEN FILENAME.

Summary.

Datasoft's Micro-Painter is an excellent graphics program for the Atari. And, by using Micro-Puzzler, you can get even more enjoyment out of your Micro-Painter.

## THE VISICALC SPREADSHEET
(and some of its uses)

by Mark Stinson

### The Story

It was several months ago now. I was sitting in front of my 130XE, with Zork I booted up, trying to impersonate a proficient adventurer. While my fevered brain attempted yet another route through the maze, I was interrupted by a knock at the door. It was the postman delivering a package from New Atari User! I rushed downstairs, collected my prize, and began to unwrap it.

You may wonder why I would wish to purchase a package like Visicalc (unless you own one), and so would I had it not been for the advertisement in the magazine. The price indicated was twelve pounds, with the words "Priced at £159.90 when first released!". Those words were the reason for my purchase, and what a good move that turned out to be.

### The Quality

When you examine the packaging, and the manual, you can immediately see that this is a quality item. Visicalc comprises: the disk, a good sized and hard bound manual (170 pages), pocket reference card, strong cardboard sleeve, and outer box. It is very pleasing to know that you own some real business software which can be as productive as many expensive PC packages.

I decided to pay a visit to my local library, to look for books on spreadsheets. I never thought for one moment that there would be any books about Visicalc itself (in my experience, a visit to the library provides an opportunity to borrow books on the most outlandish subjects, except the subject you actually need). How wrong could I be? I used all six of my library tickets on books about spreadsheets, and three of these were written for Visicalc specifically. So, with Visicalc you know that you have a widely recognized software package, with plenty of supporting information available.

One touch I particularly liked was the command chart in the manual, which shows all the available commands in a 'family tree' format. Along with each command listed in the tree is a short description of its use, to facilitate quick reference. The 'family tree' is then broken down into smaller portions, in the following pages, to allow an in depth explanation of each command. The command charts span a total of sixty seven pages.

### An Overview

A lot of people will already know what a spreadsheet is, but for those with little or no knowledge of the subject, I will give a short description. A spreadsheet is basically an elaborate mathematical instrument. It is like a calculator, although it is far more versatile.

Imagine trying to write calculations out on paper, where one calculation leads on to another. You divide your paper into rows and columns (see figure 1), which has the effect of making a number of boxes. Different mathematical equations are placed in individual boxes, giving sets of calculations. This is exactly what Visicalc is, an electronic worksheet made up of 63 columns and 254 rows (16002 boxes).

The computer equivalent can only show a portion of this very large worksheet, on the TV or monitor. The display on the screen is called a "window", and shows an area of the spreadsheet. The rest of the spreadsheet can be viewed by moving the window with the arrow keys.

## THE VISICALC SPREADSHEET

continue from previous page

The cursor, with which we are all familiar from Basic, becomes elongated (to 9 characters) and fills one box (or cell). We can therefore enter any data, or formula into the cell currently highlighted by the cursor. However, formulae are 'hidden' behind the cell, and a formula's length can therefore be longer than the cell. Once a formula is entered, it becomes invisible, and the cell shows the result of the formula, rather than the formula itself.

### Some Calculating

We will now see how to manipulate data with Visicalc. However, let us first create an analogy. Think of your basic salary, and how you calculate your tax. You would need to know your tax code, and any pension payments. To work this out would be relatively simple using a calculator. However, if you wanted to repeat the calculation a large number of times it would be a tiresome job on the calculator; this is where a spreadsheet comes into its own.

To calculate tax with Visicalc, we would use the following method:

```
       A       B       C      D
1 Salary   Tax Code Pension Tax Paid
2 10000    501      600     @sum(A2-(b2*10+3000+c2))*.25+600
```

(note, the @ sign is simply a prefix which informs Visicalc that a formula is being entered)

I'm sure I heard a groan when we reached cell D2, but don't despair! This calculation is really quite simple (honest). Let us follow the code in cell D2:

**@sum(**

this informs Visicalc that a numerical calculation is to take place, and the bracket opens the start of the equation.

**A2-(**

This next section of the calculation, highlights cell A2, which is 'Salary'. It basically instructs Visicalc to take the contents of cell A2 (10000) and take away the result of the following calculation contained in the next set of parenthesis.

**B2*10+3000+c2))**

This part of the calculation lumps together the parts of your salary which are not taxed. The amount in b2 is the tax code, which is multiplied by ten to obtain the full allowance (in this case 5010). Next is the figure of 3000 which is the portion of taxable income now taxed at 20%. Finally, c2 is any pension payments (pension payments are not taxed);

***.25+600**

This final portion of the code instructs visicalc to take the balance of the calculations, multiply by 25% (basic rate tax) and add 600 (the new 20% tax rate on the first 3000 pounds of taxable pay).

There are two important features to mention here. Firstly, once the calculations are in place, each new salary to be worked on requires only the entry of salary, tax code and pension contribution. Secondly, the result of the calculation by cell D2 is printed (949.75) in D2. The formula stored at D2 is invisible, and only the result shows; the formula is still there however. To prove this we can change salary to 20000, and instantly, the result changes to 2849.75.

It is this versatility which makes a spreadsheet so useful. Large involved routines can be developed, and once they are in place, calculation becomes a breeze with the only alterations being to the data on which calculations take place.

## THE VISICALC SPEADSHEET

continue from pevious page

### The Command Structure

There are a great number of commands available with Visicalc, and they are divided into two distinct categories; Mathematical and Data Management.

### Mathematical

The Mathematical commands generally follow the usual rules of notation, which we already use with Basic. As you would expect, the usual symbols are used, + - * / ^.

However, there are a good number of special commands, or functions, which work on the following variables: v - a value, list - a combination of values/ranges separated by commas, range - portion of a row or column.

Each function is preceded by the @, and they comprise:

| | |
|---|---|
| @ABS | - absolute value of v |
| @AVERAGE | - Average of values in a list |
| @COUNT | - number of values in a list |
| @EXP | - natural exponential of v |
| @INT | - integer portion of v |
| @LN | - natural log of v |
| @LOG10 | - base 10 log of v |
| @LOOKUP | - compares v to a range and returns value in next row/column |
| @MAX | - returns maximum value in list |
| @MIN | - returns minimum value in list |
| @NPV | - returns Net Present Value (annual percentage depreciation) |
| @SQRT | - square root of v |
| @SUM | - sum of values in a list |

### Data Management

There are a great many 'data management' commands, which are used for a variety of functions including: save/restore spreadsheet, replicate data/formulae, set column widths, set justification, set integer/decimal format, etc.

All of these commands are extremely well documented in the manual, and nearly all of the commands found in expensive PC spreadsheet packages can be found in Visicalc.

### For the Serious User

There are many readers out there who no doubt use spreadsheets at the office, but do not yet own a spreadsheet for their Atari. Many of these readers will have questions about the functions available with Visicalc.

From my experience of Visicalc, and PC packages like SC5 and Lotus, Visicalc does an amazing job when you consider the memory available with our machine, and the limitations of an eight bit processor. Visicalc has many high level functions such as Net Present Value, lookup tables, Count, Average, Logs, and many, many more.

If you do not own Visicalc, whether you have any experience of spreadsheets or not, I would strongly advise you to obtain a copy, while you still can. Grailin and Micro Discount still had some recently. And by the way, I am still trying to solve Zork I, and what a brilliant game that is. You may also have noticed that this one is on sale with New Atari User at the incredible price of 2.95; buy this one also, it simply can't be missed.

# BUY, SALE and CONTACT section

## FOR SALE

12" GREEN SCREEN MONITOR for Atari 800XL or XE machines. New when purchased 1992, used 6hrs. £25. Buyer collects. (unable to pack & send, dissabled).
Contact: Bill Hall, 38 Old Hall RD, Tingley, Wakefield, WF3 1QE.
Phone: 0532-524108

## WANTED

I need some good quality DS/DD disks, about 100 to 300 wanted. I will pay between 5p and 15p per disk.
J. HAWORTH,
86 ASHTREE RD.
OADBY,LEICS,LE2 5TD

Does anyone have or know how to get a BOOK called RANDOM ALLEY Adventures for the Atari 8-bit. If so please call:
Michael on 091-2859356

Does anyone have a 'QS3' interface by Qume, (any condition) for sale. Please call Dave:
on 091-2710086

Atari Lightgun and Lightgun software, reasonable price paid.
Call Mark on 0602-720597 anytime.

I'm looking for an 850 voice master, and old Atari Users. Please send me you pricelist (with the cost of postage please).
Write to:
DELELIS Christian
321, rue Leon Blum
62232 ANNEZIN
FRANCE

## CONTACT

Wanted contact with lucky owner of this excellent new programming language called 'QUICK'!. To swap hints, tips, programs, games and general knowledge.
Write to:
James HAWORTH,
86 ASHTREE RD.
OADBY, LEICS, LE2 5TD

HI ! I am back again with another ad!!!
Who wants a GREAT disk full of HiScores for most 8-Bit games?? there is also a booklet which you can have too. It is an essential part of a games players collection...
AND IT's FREE!!! Just send a SAE to
Also send in ANY Scores of your own.
J.HAWORTH,
86 ASHTREE RD.
OADBY, LEICS, LE2 5TD

Contact wanted with any 8-bitter anywhere in the world! To swap hints, tips, games etc...You must be interesting and have a lot to say!!! I don't care how fast you are at writing, but I love a good read.!!!
Write to:
James HAWORTH,
86 ASHTREE RD.
OADBY, LEICS, LE2 5TD

TYNE&WEAR
ATARI
USER GROUP

## PLAYER/MISSILE GRAPHICS

by Tomohawk April 1992

If there is one hardware feature that performs the most pretty tricks in software today, that feature would probably be Player/Missile graphics! With a single PMG, you can create a 2nd cursor or maybe a drawing pointer (as used in most graphic-art packages), an animated character perhaps, or even a screenful of scrolly stars (as seen in the BIG demo).
This article is aimed at programmers with a fair understanding of hardware and shadow memory and also boundaries.

### PLAYERS and MISSILES.

There are 4 players and 4 missiles in the Atari. They occupy a total of 27 hardware registers and 6 shadow registers. Players are 8 bits wide and missiles are just 2 bits wide.

| | | | | |
|---|---|---|---|---|
| (W) | 53248 | - 53251 | HPOSPO | - P3 |
| | 53252 | - 53255 | HPOSMO | - M3 |
| | 53256 | - 53259 | SIZEPO | - P3 |
| | 53260 | | SIZEMO | - M3 |
| | 53261 | - 53264 | GRAPHPO | - P3 |
| | 53265 | | GRAPHMO | - M3 |
| | 53266 (704) | - 53269 (707) | COLPMO | - PM3 |
| | 53275 (623) | | PRIOR (GPRIOR) | |
| | 53277 | | GRACTL | |
| | 53278 | | HITCLR | |
| | 54272 (559) | | DMACTL (SDMCTL) | |
| | 54279 | | PMBASE | |
| (R) | 53248 | - 53251 | MOPF | - M3PF |
| | 53252 | - 53255 | POPF | - P3PF |
| | 53256 | - 53259 | MOPL | - M3PL |
| | 53260 | - 53263 | POPL | - P3PL |

(W) WRITE.

53248 - 53251 HPOSPO-P3

All 4 players have Horizontal positions, controlled by these locations. Vertical positions are somewhat different, described later.

53252 - 53255 HPOSMO-M3

Alike players, the 4 missiles have Horizontal positions. Vertical movement is explained later.

53256 - 53259 SIZEPO-P3

Each player also has a size of width. Although, players remain 8 bits wide, each bit can appear normal, double or quadruple width. This size is selected from the lowest 2 bits of the SIZEP registers:

```
BITS  7 6 5 4 3 2 1 0 DEC SIZE

       ..unused.. 0 0  0  NORMAL WIDTH
                  0 1  1  DOUBLE   "
                  1 1  3  QUAD.    "
```

## PLAYER/MISSILE GRAPHICS
continued from previous page

53260 SIZEMO-M3

Missiles also have a size of width selection. With the missiles, all their widths are selected with just ONE register. This is achieved by using each of the four "bit-pairs" of the byte, for each missile width:

```
BITS  7 6 5 4 3 2 1 0

M#     -3-  -2-  -1-  -0-
```

The bit combinations to select the missile sizes are identical to the players.

53261 - 53264 GRAPHPO-P3

Designing your player is done alike character-redefinition, though, altering the GRAPHP registers independent of DMA (see DMACTL and PMBASE), will cause the same value to be echoed on every scan-line of the graphic. There are several methods of altering a graphic at chosen intervals, but it is recommended that PMBASE should be used if a shape is required.

53265 GRAPHMO-M3

As a missile is only 2 bits wide, all 4 missiles can be designed in just the ONE register. The echo of the value in this GRAPHM register applies equally with missiles as with players

```
BITS  7 6 5 4 3 2 1 0

M#     -3-  -2-  -1-  -0-
```

53266 - 53269 (704 - 707) COLPMO-PM3 (PCOLRO-3)

Each player has its own colour register that it shares with its relative missile. The shadow registers are given in the brackets.
Although each graphic is only 1 colour throughout, there are several simple methods to obtain multi-coloured graphics. One way is to overlap graphics (see PRIOR).

53275 (623) PRIOR (GPRIOR)

One of the nice features about graphics is the ability to prioritise players, missiles and playfields with ease and speed (see bits 3, 2, 1 and 0).

BIT 5 DEC 32

MULTIPLE-COLOUR GRAPHIC ENABLE.
Enabling bit 5 of PRIOR causes a logical "OR" of COLPMO with COLPM1 and COLPM2 with COLPM3 where overlaps occur on the screen.

BIT 4 DEC 16

FIFTH PLAYER ENABLE.
Enabling bit 4 causes all 4 missiles to assume the colour of COLPF3 (color 3) 53273 (711). They will not assume any relative positioning with each other. You'll need to perform this task yourself.

BITS 3 2 1 0 DEC 8 4 2 1

## PLAYER/MISSILE GRAPHICS

continued from previous page

PRIORITY SELECT.
The table below details the 4 priority combinations between players, missiles and playfields:

| Priority | B3=1 | B2=1 | B1=1 | B0=1 |
|---|---|---|---|---|
| High | Pf0 | Pf0 | PM0 | PM0 |
| | Pf1 | Pf1 | PM1 | PM1 |
| | PM0 | Pf2 | Pf0 | PM2 |
| | PM1 | Pf3+P5 | Pf1 | PM3 |
| | PM2 | PM0 | Pf2 | Pf0 |
| | PM3 | PM1 | Pf3+P5 | Pf1 |
| | Pf2 | PM2 | PM2 | Pf2 |
| | Pf3+P5 | PM3 | PM3 | Pf3+P5 |
| Low | BG | BG | BG | BG |

Note:
If multiple bits are set, all conflicting priorities will turn black when overlapped on screen.

ie.
If bits 3 and 1 are simmultaneously set, players/missiles 0 and 1 would 'black-out' with playfields 0 and 1 of the same priority level (I like to call this, "a Power-cut").

DIRECT MEMORY ACCESS (DMA).
DMA is a special technique used by the Antic chip to directly access the memory for use as graphics capabilities. When enabled, Antic will temporarily halt the CPU, thus, stealing just enough cycles (time) to perform its necessary tasks.
Dynamic refresh, playfield and P/M are the only three types of DMA in our Atari, and of these, we are only concerned with the P/M DMA. When enabled at DMACTL, the GRAPH registers are loaded with a different item for each single/double scan line, pointed to by PMBASE and turned on to the display by GRACTL.

53277 GRACTL

| BITS 1 | 0 | DEC | STATUS |
|---|---|---|---|
| 0 | 0 | 0 | DISABLE PLAYERS AND MISSILES |
| 0 | 1 | 1 | ONLY MISSILES ENABLED |
| 1 | 0 | 2 | ONLY PLAYERS ENABLED |
| 1 | 1 | 3 | ENABLE PLAYERS AND MISSILES |

If you are not using DMA, you should turn your players/missiles off by changing their Horizontal positions to 0 (see HPOSP and HPOSM).

54272 (559) DMACTL (SDMCTL)

| BITS | DEC | STATUS |
|---|---|---|
| 7 and 6 | 128 and 64 | unused |
| 5 | 32 | ENABLE DMA FETCH INSTRUCTION |
| 4 | 16 | =0 FOR DOUBLE-LINE RESOLUTION<br>=1 FOR SINGLE-LINE " |
| 3 and 2 | | |

PLAYER/MISSILE GRAPHICS
continued from previous page

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | NO P/M DMA |
| 0 | 1 | 4 | ENABLE MISSILE DMA ONLY |
| 1 | 0 | 8 | ENABLE PLAYER DMA ONLY |
| 1 | 1 | 12 | ENABLE PLAYER AND MISSILE DMA |

1 and 0

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | NO | PLAYFIELD |
| 0 | 1 | 1 | NARROW | " |
| 1 | 0 | 2 | STANDARD | " |
| 1 | 1 | 3 | WIDE | " |

Notes on above bits:
Bit 5 should always be set so that you may see what is going on, though, if you don't have this bit set, the majority of processes should execute approx. 30% faster.
These bits may appear a little complex at first, but all you'll need to do is select which bits you need and then sum their decimal values:

ie. if you wanted:

| | | | |
|---|---|---|---|
| BIT 5 | DEC | 32 | ENABLE DMA FETCH INSTRUCTION |
| BIT 4 | DEC | 16 | SINGLE-LINE RESOLUTION |
| BIT 3 & 2 | DEC | 8 + 4 | ENABLE PLAYER AND MISSILE DMA |
| BIT 1 | DEC | 2 | STANDARD PLAYFIELD |

All Bits final sum = 62

This is the value that would need to be POKEd into the shadow register 559 (SDMCTL).

| RESOLUTION | MEMORY REQUIREMENTS |
|---|---|
| DOUBLE-LINE<br>(DLR) | 4 PLAYERS OF 128 BYTES EACH<br>ALL 4 MISSILES TAKE 128 BYTES TOTAL<br>= (4*128)+128 = 640 BYTES<br>1K BOUNDARY |
| SINGLE-LINE<br>(SLR) | 4 PLAYERS OF 256 BYTES EACH<br>ALL 4 MISSILES TAKE 256 BYTES TOTAL<br>= (4*256)+256 = 1280 BYTES<br>2K BOUNDARY |

54279 PMBASE

The area of memory where the graphics begin is

256 * PMBASE

P/M graphics must begin on a 1K boundary for double-line resolution and a 2K boundary for single-line resolution. The 1/2K from PMBASE is actually organised as in the diagram below:

PLAYER/MISSILE GRAPHICS
continued from previous page

```
DOUBLE-LINE        0 ***PMBASE*** 0       SINGLE-LINE
RESOLUTION           *          *         RESOLUTION
                     *  UNUSED  *
                     *   AREA   *
                     *          *
                +384 ************ +768
                     *          *
                     * MISSILES *
                     *  0 - 3   *
                     *          *
                +512 ************ +1024
                     *          *
                     * PLAYER 0 *
                     *          *
                +640 ************ +1280
                     *          *
                     * PLAYER 1 *
                     *          *
                +768 ************ +1536
                     *          *
                     * PLAYER 2 *
                     *          *
                +896 ************ +1792
                     *          *
                     * PLAYER 3 *
                     *          *
               +1024 ************ +2048
```

The offset from PMBASE and the beginning of the missiles is unused, it is not cleared or altered and is therefor free for use.

VERTICAL PLAYER/MISSILE MOVEMENT.
There is no hardware feature for vertical movement of P/M's. The only way to move your desired shape vertically, is to actually copy the memory where the shape is into a higher or lower position, erasing the previous image.
This task is very time consuming for basic, and is just too slow for fast-action arcade games, however, if your a very clever basic programmer, there is ONE way of accomplishing fast vertical movement without the need for machine-code!
This method would involve using a basic string dimensioned to 1/2K. The only snag is that its address must start on a 4K boundary!

(R)

53248 - 53263 P/M & PLAYFIELD COLLISIONS.

Another nice feature about P/M's are these 16 memory locations, where upon being read, various on-screen collisions can be detected.

```
BITS  7 6 5 4 3 2 1 0
   ..unused.. X X X X
```

All of the 16 collision registers take the above form. Only the lowest 4 bits can be active and each bit represents the Player/Missile or Playfield being collided with.

ie.
* 53257  M1PL

## PLAYER/MISSILE GRAPHICS

continued from previous page

If this location has a decimal value of 2, then bit 1 is set, thus, Player 1 has collided with Missile 1.

note:
P & PL = Player PF = Playfield M = Missile.

53278 HITCLR

Once a collision has occured, the value remains in the register. You should always clear the register before every joystick detection in a game and to do this you should POKE a value into HITCLR.

P/M GRAPHICS IN YOUR OWN GAME.
There are 2 types of P/M's you can have in your own game. You can either have them with a designer shape, or with the same bit pattern all the way down the screen.
If You do not want to design a full blown shape in your graphic, then you should ignore steps 2 and 6.

1. CALL YOUR PLAYFIELD.
   A simple graphic call will suffice.

2. ENABLE P/M DMA AND RESOLUTION.
   see DMACTL.

3. DETERMINE HORIZONTAL POSITIONS.
   see HPOS's.

4. DETERMINE P/M COLOURS AND SIZE.
   see COL's and SIZE's.

5. DETERMINE GRAPHIC SHAPE.
   see PMBASE for full blown shape, otherwise, see GRAPH.

6. ENABLE DMA TO SCREEN.
   see GRACTL.

NB* Information gathered from Technical Reference Notes, Mapping the Atari [REV], De Re Atari and Page-6 magazine.

# LIZARD.

Review: by Mark Fenwick

'Tiger Developments', one of the few companies producing new quality software for the 8-Bit, presents Lizard.

The story. The planets of the Reptilian System have recently formed an alliance, with the aim of taking over the earth in order to setup breeding colonies in the ideal conditions found there. The planets of the Reptilian System are each ruled by a tyrant beast, which hide behind powerful force fields. Intelligence sources say that these force fields are solar powered and that by destroying the solar panels the force fields can be brought down. Thus your mission is to destroy the five tyrant beasts using any methods at your disposal. Well, that's got you worried hasn't it!

The game starts with your player, complete in space suit and jetpack, left of five planets, each with its own name. You can select which planet you wish to enter first, I don't think it matters what order. Press the j/stick button to enter planet. You're now at the side of the door you entered and are now ready to do battle, should you wish to leave then simply turn and enter the door, where you return to the planets screen. Once on a planet move right, there's plenty of aliens coming at you ready to be shot at. You'll notice boxes scattered about, if you place your man over these and press 'P' you pick up what ever item is in it. Items are: Power, Smart Bomb, Super Gun and Bonus points. When you pick up a Smart Bomb a coloured border will come around your bomb icon on your console, to activate press 'SPACE' Super Gun works for a limited time only and allows quick firing.

The object boxes contain coloured letters, when you pick up a letter it will be displayed on your console, use arrow keys left/right to move your object cursor. The real object of the game is to obtain five letters of the same colour, once this is achieved you must go to one of the main computers and press 'L' to log on. Use the joystick left/right/fire to select letters for input. If you input the letters in the correct order you'll be given a Mega-Gun, to kill the Tyrant beast, otherwise you must log on again. As there are a number of combinations with five letters to find the correct order, this part alone will keep you busy.

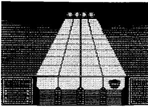
To complete the game you must get all five codes correct, as well as killing each tyrant beast on each planet in turn. Beware not to shoot all the solar panels on your first run, otherwise you'll lift the force field at the end of the level. You cannot kill the beast with the weapon you are armed with and soon get killed.

Lizard is a combination of Zybex, but offers more to do than just a shoot 'em up. Each planet is different with excellent graphics and colours, the screen scrolls smoothly as you move left and right. This isn't a game you'll complete easily, its a good idea to map each planet, so you don't power up too early, and you can note what boxes hold what letters. I mapped each planet, which took roughly 2 hours, pressing 'OPTION' to pause the screen. I found the game very enjoyable as you've tasks to carry out as well as shooting everything in sight.

Lizard is available direct from 'Tiger Developments', priced £4.99, a very good price when you consider how many of us are left, so if you've received a pirate copy, please send for an original. If there's not enough interest, they may decide not to produce any other software! My opinion: Well worth a place in anybodies collection!

## HYPNOTIC LAND

Review: by Mark Fenwick



Well, here we are, a brand new Rom for '93, yes a Rom. It's a while since we saw a Rom for the trusty 8-bit. Tracked down by Page 6, produced in Italy from Lindasoft.

This game is very similar to the Lynx version called 'Klax'. Your aim is to supply energy to a distant planet in danger, the energy being the coloured mineral elements which you must collect.

The object of the game is to place coloured balls into their corresponding coloured cups. As there are four ramps, the balls must be re-directed according to colour. If your cup at the base of the ramp is green, then you must place green balls only into this cup. Changing the direction of the falling balls can be achieved by placing direction arrows on each ramp via the joystick. Arrows can only be placed on the red squares of each column, three arrows in total but only one per row. By pressing j/stick button you choose between left and right arrows to re-direct the balls. When the ball hits the arrow it will move accordingly to the adjacent column. Sounds too easy!

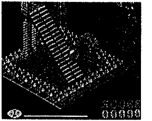
You start play on level one. A nice bit of music begins, 'Lambada' I think. I did try every key to switch it off after a while. You'll notice the right of the screen states 'errors, shots 9'. Errors is really equivelant to lives. Each time you make a mistake i.e. if you drop an incorrectly coloured ball in your cup, let your coloured ball corresponding to cup colour fall off the end of ramp, or off the side and smash. In each case you'll get an error. You're allowed five errors in total for the whole of your game, so take care. Each time you catch a correct ball five arrows will appear at the top left of the screen.

Each ball after will add to this until the end of the screen, then you're home free. Well, not really it's level 2 then, and with another level so the game gets harder. This time you've two cups, different colours, too! You've also got a nice little chap in the shape of an octopus, who tries to steal your balls. (Now, now!) You can shoot the Octopus by keeping your j/stick button pressed and moving your target over him. If he's holding a ball and you shoot him while he's over the grid he'll drop the ball back into play. You've 9 shots so you should manage to kill yourself a couple of Octopus's. Occasionally you'll see a shining ball descend, drop this in your cup and pick up a nice little bonus. The gameplay is very good. It did take me a while to get to grips with what was what. It was a while before I managed to get off level one. The gameplay is simple but hard, as you need quick reflexes at times. This makes you more determined to get further. I managed to progress to level 3 eventually. This is where things really speed up, two cups the thieving octopus plus the speed of the balls, you begin to sweat! If you want to pause for breath then press [SELECT] then again to resume play.

Overall I rate this game, it makes a change from a shoot 'em up. It's easy to play and addictive with nice colours and good graphics with the landscaped background giving depth. My only criticism is the errors (lives) I feel it would have been better to have five per level rather than five per game. Plus it would have been nice to switch the music off for a while, but 30 minutes and you'll want to adjust the volume! Currently available from Page 6 at £12.95 not a bad price either. My opinion: Buy it!

## AIRBALL

Review: by Mark Fenwick



Airball, from Microdeal US. Only recently has this game 'Rom' been heard of in the U.K. The reason being, Atari's lack of good marketing. Airball, packed in the 'XE Video Game' packaging to promote the failed XE Game System, was never released in the U.K. Its just a pity Atari only flooded the U.K. with hardware only, lacking the software desperately waited for. Well you know that!

The story begins where an evil wizard turns you in to a ball, for what reason, I don't know. You are then placed in a mansion containing 150 rooms. While in the mansion, you must pick up objects, but your main goal is to find the spellbook in order to become human again! Easy ? No, for the Wizard has also given you a slow puncture and placed spikes and prickly floor tiles in most of the rooms. You can inflate yourself in certain rooms containing pumps.

Airball is very similar in game play to 'Head over Heels' and 'Molecule Man' with the play being diagonal movement. The graphics are highly detailed and give a true 3D prospective. There are many nice touches like stair cases, archways, ramps and crates which you can move, giving each room depth.

You start play with four lives, in a room on top of an air pump, where you automatically begin to inflate. Beware, too much air and you'll pop! Once you're sufficently inflated, move off the pump in the desired direction. Movement is by diagonal direction on the j/stick, to bounce press fire. There are four exits to this first room, each leading to a different location. Remember, there's plenty of spikes and prickly floors to contend with so watch your step, sorry, roll, touch one of these and you'll burst and fly around the room. You'll come across various objects, which you can pick up for bonus points i.e. Gold and Gems. Simply move over the item you wish to pick up. You'll also find a flashlight and lantern, to be used in some of the unlit rooms. A good idea, like with all multiscreen games of this type, is to make yourself a map. This way you can easily locate airpumps when you become low on air.

I feel Airball is an excellent game, very well put together, nice clean hi-res graphics, plenty of depth ie. 150 rooms, with a nice concept. My only critism is the music, although its good quality, it does become annoying after a while. Atari really got their act together, producing really good software, just before they decided to abandon the 8-Bit.

Airball is currently available from 'Micro Discount' at £17.95, a price which five years ago would have been acceptable. Todays present situation, I feel £10 could be shed from the price. The reason being Atari is not going to produce any more games for your 8-Bit, and although Airball is new to the U.K. it is five years old. So Atari themselves wont be seeing much of your £17.95, as they're no longer producing. However, the game is definitely worth adding to your collection, the price you pay is up to you. You could choose to wait, see if Derek Fern will lower his price, this could be a gamble though, as only he knows what stock he has! In either case, my opinion: Buy it!!

# REBOKIT

With Rebound Creation Kit, you can construct your own fiendish screens for my Rebound game(Antic, January 1987).

## GETTING STARTED

The blinking cursor in the upper left hand corner is your "drawing" cursor. At the bottom of the screen is a display of various shapes from Rebound, plus a blinking "selection" cursor.

To create screens, use the joystick to draw, using the shape selected with the selection cursor. The program supports all eight joystick directions with off-screen wrap-around.Pressing the joystick trigger with the drawing cursor over an empty space puts the selected shape on the screen. Pressing the trigger over an occupied space blanks it.

[OPTION] and [SELECT] move the selection cursor.

The [START] key toggles between the two sets of shapes available for selection.

Use the keyboard to add text messages to the screen. Use inverse and lowercase letters for more colours.

Type [CONTROL][S] to save a screen to disk. You will be prompted to type in a file name and a screen number. Note: Before using the SAVE option, move the drawing cursor to the position on the screen where you want flip to start out.

Type [CONTROL][L] to load a previosly saved screen.

Type [SHIFT][1] to [5] to adjust the colours of a screen.

Each screen can use different colours. Push the joystick up and down to adjust the luminance, left and right to adjust the hue, and press the trigger when done.

Type [CONTROL][Q] to exit the Rebound Creation Kit.

Press [SHIFT][CLEAR] to clear the screen.

The Rebound Creation Kit uses all the objects described in the Rebound game instructions plus these new ones:

New walls-To change the look of the game, you can use different wall shapes, solid blocks, or outline blocks.

Fake walls - These look like solid walls in the game (in the Rebound Creation Kit they have an identification mark), But Flip can bounce through them. Some will kill Flip, some will hide him and some will let him bounce over them. You can intermix these with solid blocks to create mazes.

Special - Certain shapes may be used by pressing special keys on the keyboard. Note that invisible barriers are shown in the Rebound Creation Kit as special shapes.

After you complete a screen, exit the program via[CONTROL][Q] and LOAD the Rebound game program. Then ENTER the screen file created (using the name you typed in). RUN the program to make sure your screen is in place. (You can adjust the FIRSTSCREEN variable in line 4 to your screen number.) Then, SAVE your new Rebound game under a new file name.

## SCREEN DESIGN HINTS

Warning: there are two rules to follow when designing a Rebound screen:

1. There must be at least one Ogart to save on a screen, and there must be a way for Flip to save each Ogart on a screen.
2. You can't put just one transporter on a screen. And if more than two are used, they may not work properly - Flip could end up in the middle of a wall.

Make sure your screens are possible to complete, but not too easy. Try to use the objects in fresh new combinations that challenge the player to use Flip in new ways. Try something bizzare.

You might use the text to help give a theme to a screen, or to possibly give a clue, a warning, or a pat on the back. Make efficient use of the screen, not wasting much space. Try to make sure that there are no dead ends (like a one way flow of RotogartEatos that can't be switched).

# CRACKING THE CODE

by Keith Mayhew and Roy Smith

Re-printed by M.Gerum

This article first appeared in "The U.K. ATARI Computer Owners Club" later renamed to "MONITOR".

## PART 2

Since the last issue, you have had plenty of time to practice working with binary and hex numbers. This second part will discuss briefly the general internal layout of the machine and proceed to some introductory machine code instructions.

### BASICS

The microprocessor used in the ATARI computers is the popular and well documented 6502. The 6502 has sixteen address lines and eight data lines to communicate with the rest of the computer. In part 1 we showed a diagrammatic representation of binary codes in columns, and we showed up to eight columns of bits. If you can imaging another eight columns added to the first eight this would represent our total number of address lines. So the columns would represent: 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768. If every column contained a '1' then the largest number accessable would be the total of all sixteen columns, which would be 65535. If every column contained a '0' then the lowest number is obviously zero, which means the overall range of address is 65536. Now is a good time to introduce an abbreviation, that is the 'K' representing 1024. If 65536 is divided by 1024 then it can be written as '64K'. Each one of the 65536 addresses can be thought of as a box, into which a piece of data may be stored. The name given to one of these boxes is the 'byte', which is further sub-divided into eight bits (one bit is equivalent to a binary digit). As we know, eight bits can represent a number from 0 to 255, giving a total of 256 different numbers. Thus all 65536 memory locations have a number between 0 to 255 held in them, which is transferred via the eight data lines.

64K of available memory sounds like an awful lot! But that is the trouble, it is not all available to you for prógram and data storage. In the ATARI without BASIC, 48K of memory is left free for program storage in RAM (Random Access Memory). The rest of the 64K is allocated to the ROM (Read Only Memory), which is where the operating system is held. The general I/O devices are also allocated some of the memory, these devices include the cassette recorder, the display and the keyboard.

### THE 6502 PROCESSOR

The processor contains several different registers. All of these registers are eight bits wide except for the 'program counter', often denoted as 'PC', which is sixteen bits wide. The other registers are the 'acumulator' or 'A', the two index registers 'X' & 'Y', the processor flags register 'P', and the stack pointer 'S'.

The PC register is sixteen bits wide because it needs to point to any byte in the full 64K memory space, the byte pointed to is the byte where information will be accessed. The most often used register is the accumulator, this is where most calculations and values are held. The two index registers are general purpose and can be used by themselves or in conjunction with the accumulator. The P register contains the current status of the processor and the S register points to the current position in the stack. This is only a brief introduction to the 6502's registers and these will be covered in greater depth later.

### THE PAGING CONCEPT.

Due to the 'architecture' or internal layout of the 6502, the memory is set out in 'pages' consisting of 256 bytes per page, this conviently divides into the 65536 locations to give 256 pages. Why are the memory locations laid out in this way? The reason lies with the fact that the 6502 is an eight bit microprocessor, and as the address bus is sixteen bits wide, it is composed of two eight bit words (where a word is an expression used to describe a set of bits of a certain length; note, an eight bit word is often referred to as a byte). The upper eight bit word points to one of 256 pages and the lower bit word points to a byte within that page, see figure 1.

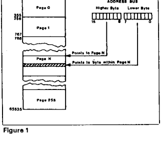


Figure 1

# CRACKING THE CODE

continue from previous page

This paging feature is not generally of any importance, as the complete memory block appears to the programmer as one continuous area of memory locations. However, it is important to consider the paging feature when programming in machine code for two reasons, the least of which is a slight time delay when crossing a page boundary. The other reason is connected with limitations to some 'addressing' modes, where that mode is restricted to only one page in size. More on addressing modes later.

MACHINE CODE INSTRUCTIONS

Machine code programs are stored in memory as a series of binary numbers. The 6502 starts execution by reading the first byte pointed to by the Program Counter register (PC), this will inform the 6502 which command it is to implement, and is referred to as the operation code or op-code byte. This op-code is followed by either none, one or two additional bytes depending on the type of instruction being executed. These bytes are automatically read by the 6502, incrementing the PC register by one each time. These bytes are often referred to as the operand bytes and their purpose is either to point to a memory address where the op-code will perform its function, or the operand will actually be the data which the op-code is to act upon. In other words in one instance the op-code needs the actual value of the operand byte as the data for its operation, whereas in the other case the op-code retrieves its data from the location pointed to by the operand byte(s). Unfortunately, unlike BASIC which reports errors to the user, the 6502 gets totally confused when it retrieves an op-code which is not one of its recognised instructions. This will cause the 6502 to 'hang up' or 'crash' and is irreversible apart from turning off and on and starting all over again!

THE HARDWARE STACK

The 6502, as mentioned earlier maintains a stack pointer (S). The S register is eight bits wide and can therefore address anywhere within a page size. In the 6502, the S register addresses page one. Page zero is reserved because it has great importance to some of the 6502's instructions, therefore the stack, which could be on any page, has simply been allocated to the next available page. This stack is referred to as the hardware stack because it is supported by the 6502, other stacks do exist but are supported by software techniques, for example BASIC keeps a 'run time' stack to place the return address for a subroutine.

The stack is used for high speed, temporary storage of numbers, where it is impractical to use the limited number of internal registers. The internal register A,X and Y are used for quick manipulation of numbers, but if you need to save some numbers temporarily whilst executing other routines, the stack is ideal. The only other way of saving these numbers would be to place them in RAM, this is slightly slower, but more importantly you need to keep a track of the memory location being used, whereas the stack being hardware maintained, keeps track for you. So, why not use the stack for all storage? There are two reasons, first the stack is limited to one page of storage i.e. 256 bytes. The other reason is that the stack is a sequential storage system, which means that numbers can only be retrieved in the order in which they were stored. This type of stack is called a LIFO stack, standing for Last In First Out.

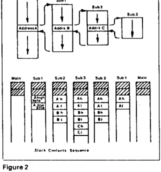


Figure 2

A particular use for the stack is subroutines. When the subroutine is called, the current contents of the PC register is stored on the stack, then it is loaded with the address of the subroutine which is then executed, upon completion the PC register reloaded from the stack with the address of where it left off from the main program. It should be noted that if in a subroutine numbers are stored on the stack, they obviously must be retrieve within that subroutine before returning to the main program, otherwise the return address will be incorrect, and will certainly cause undesired results. If subroutines are nested, i.e. one subroutine calls another, see figure 2, then care should be taken to ensure that not more than 256 bytes are stored on the stack or 128

# CRACKING THE CODE

continue from previous page

subroutines maximum, otherwise information will be overwritten because the stack loops around on itself. In practice less than 128 subroutines can be called because there will probably be information already stored on the stack.

ADDRESSING MODES

There are six main types of addressing modes: Immediate mode is when, as mentioned earlier, the operand of the instruction is the actual data required, see figure 3. Absolute mode is the other case mentioned earlier where the operands point to the location of the data required, see figure 4, note this mode covers the complete memory area including page-0. Short addressing or page-0 addressing, is the same as absolute except only one byte is needed to point anywhere in page-0, see figure 5. Remember, absolute addressing covers this page as well, it's just slower and needs a second byte. Indexed addressing uses an absolute base address and adds to it the contents of either the X or Y registers to give the final computed address, see figure 6. Indirect uses two locations in page-0 to point to another memory location, see figure 7. The last type is Implied mode which needs no data and just performs a set, internal function. These are only the basic addressing modes, and can be combined in many different ways, which will be covered in due course. These addressing techniques apply to all instructions, with a few exceptions. The main point to understand is that all of these different addressing modes are really only different ways of accessing data, with the exception of implied mode.

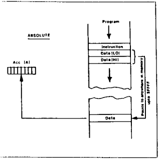


Figure 4

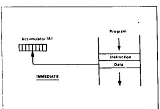


Figure 3

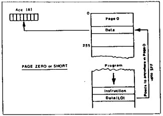


Figure 5

# CRACKING THE CODE

continue from previous page

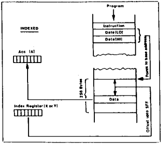


Figure 6

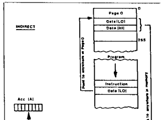


Figure 7

THE PROCESSOR'S FLAGS

Almost all of the instructions that can be used, affect the processor's flag register (P). Different instructions affect different bits of the register, where each bit will be either true (1) or false (0). What an instruction actually does will depend on the status of a particular bit in the flag register. Referring to figure 8, each bit has the following meaning:

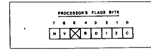


Figure 8

Bit 7, (N), a '1' indicates that a result was negative, a '0' indicates that a result was positive.

Bit 6, (V) a '1' indicates that a result had overflowed, a '0' indicates that it had not.

Bit 5 is not used in the P register.

Bit 4, (B) a '1' indicates that an interrupt was caused by the BRK instruction, a '0' if it was not.

Bit 3, (D), a '1' indicates decimal mode is in use, a '0' indicates binary mode is in use.

Bit 2, (I) a '1' indicates that interrupts are enable or allowed, a '0' indicates that they are disabled.

Bit 1, (Z), a '1' indicates that the result of an operation was zero, a (0) indicates the result was non-zero.

Bit 0, (C), a '1' indicates that a carry has occurred, a '0' indicates that there is no carry.

Bits two and four, concerning interrupts will be covered in greater depth in a future article, but all the others lead us onto arithmetic operations.

SUMS!

To finish, we have to get down to some mathematics, sorry, but it really is necessary!

(1) Unsigned Numbers: The numbers we have encountered so far have all been 'unsigned', this means that they have all been positive and we have no way of representing a negative number yet. to show the relevance of the carry (C) and overflow (V) flag we will demonstrate some binary arithmetic on eight bit numbers:

# CRACKING THE CODE

continue from previous page

| | Binary | Decimal |
|---|---|---|
| | 00010101 | 21 |
| (+) | 00000111 | + 7 |
| (=) | 00011100 | 28 |

O.K., how did we go about adding up the binary numbers? Starting from the right hand column, we add 1+1 which equals 2. Two in binary is '10' so we put the zero in the answer line and carry the '1' into the next column. Now we have 1+0+1 which equals 2 again. So once again we put the zero in the answer line and carry the '1' into the third column. This column is now 1+1+1, which is 3. Three in binary is '11', so we put '1' in the answer column and carry the other '1' into the fourth column. The fourth column is 1+0+0 which is '1', this one is entered into the answer column and there is nothing to carry forward. The next column is just 1+0 which is a '1' in the answer column. The last three columns are all 0+0's which of course is '0'!

Let's do another example:

| | Binary | | Decimal |
|---|---|---|---|
| | 10011000 | | 152 |
| (+) | 10001110 | + | 142 |
| (=) | 100100110 | | 294 |

Following the same procedure as shown before, the answer comes out nine bits long due to the carry from the last column. As the 6502 only works in eight bits then any two eight bit numbers will need up to nine bits when they are added together, so the maximum would be:

| | Binary | Decimal |
|---|---|---|
| | 11111111 | 255 |
| (+) | 11111111 | 255 |
| (=) | 111111110 | 510 |

The 6502 places this ninth bit or the 'carry' into the carry flag (C) of the register (P) i.e. sets the carry flag to a '1'. Note that the carry flag is left set until reset by the programmer.

(2) Signed Numbers: Numbers can be assigned a polarity of either positive or negative, the way this is done is to allocate the most significant bit (bit 7) as a zero to represent positive or a one to represent negative. Here are some examples:

00001011 = 11, decimal.

The far left hand bit (msb) is a '0' which means the number will be positive, in this case + 11.

01111111 = 127, decimal

Again the number is positive, but using this method + 127 is the highest number we can represent.

10001011 =- 11, decimal

The one in bit 7 represents negative, therefore the number is minus 11.

11111111 =- 127, decimal.

Again this is the largest negative number which can be represented with this method. Adding two signed numbers can result in complications:

| | Binary | Decimal |
|---|---|---|
| | 01010111 | + 87 |
| (+) | 01110000 | (+) +112 |
| (=) | 11000111 | - 71 Wrong |

By carrying a '1' from bit 6 into bit 7, the sign has been accidentally changed to a minus number to give an incorrect result. When adding these numbers on the 6502, it has no way of knowing that they are actually signed numbers so it makes no adjustments for you, so it is up to the programmer to correct the result if this occurs. The 6502 does however give an indication that the signed bit has been changed by placing a '1' into the overflow flag (V) of the register (P). Therefore by clearing (V) before arithmatic and looking at (V) after, to see if an overflow has occurred into bit 7, any necessary adjustments can be made. Using the signed method is not very reliable as there are many instances when the result will be incorrect even when no overflow is encountered:

| | Binary | Decimal |
|---|---|---|
| | 00010011 | +19 |
| (+) | 10011011 | (+) -27 |
| (=) | 10101110 | -46 Wrong |

Obviously, plus 19 added to minus 27 should equal minus 8 and not 46. It is interesting to note that the resulting answer is in the correct sign i.e. minus, and + 19 added to plus 27 is plus 46, so you can see where this wrong answer came from!

(3) Two's Complement: Because of the problems shown above, there is a need for a better system for representing signed numbers. In 'two's complement' positive numbers are still represented by a zero in bit 7 and then the rest of the bits will be the value of the number up to plus 127. To invert the sign of any number i.e. plus to minus and minus to plus, the following functions are applied to the number: first each bit is inverted i.e. '0' becomes '1' and '1' becomes '0', and then 00000001 is added to the resultant number.

# CRACKING THE CODE

continue from previous page

To represent minus 27, we first write the binary code for plus 27, then we invert each bit, and add one to give the two's complement representation of minus 27.

```
        Binary            Decimal
       00011011              +27

       11100100   invert each bit.
   (+) 00000001 add one to number.
       11100101             (-27)
```

By applying exactly the same procedure to a two's complement 'negative' number the positive value of it is given.

```
        Binary            Decimal
       11100101            (-27)

       00011010   invert each bit.
    (+)00000001 add one to number.
       00011011             +27
```

To add plus 19 and minus 27 together we would use the following procedure: First find the two's complement of 27 to give minus 27:

```
        Binary            Decimal
       00011011              +27

       11100100           invert.
   (+) 00000001          add one.
       11100101             (-27)
```

Next add the two signed numbers in two's complement form;

```
       Binary             Decimal
      00010011                +19
   (+) 11100101        (+)  (-27)
       11111000              (-8)
```

The '1' in bit 7 of the answer tells us the result is negative and is a two's complement representation of minus 8. Just to show that this represents -8 we can take the two's complement again but bear in mind that the sign will change to positive:

```
       Binary             Decimal
       11111000              (-8)

      00000111            invert.
   (+)00000001           add one.
      00001000                +8
```

Now we have a usable sign system where we can ensure correct results. However, care should still be taken with the overflow which will still accidentally change bit 7, which must then be changed back again to give the correct sign.

(4) Decimal mode: To display an eight bit binary number on the screen is reasonably easy in hex format, as two four bit blocks can be considered, each representing a hex character. But to display this on the screen in decimal characters is comparatively difficult. For this reason the 6502 has a 'decimal' mode, which uses two four bit blocks to represent two decimal characters. The codes of 0 to 9 are all that are needed and so 10 to 15 are unused, two of these characters are placed in each byte so that the complete byte can represent anything from 00 to 99 in decimal. This form of storing decimal numbers is termed Binary Coded Decimal (BCD). Listed below are the four bit binary codes for '0' to '9'.

| Decimal | BCD |
|---|---|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |

Using this information, let's try some examples in eight bits:

```
      BCD           Decimal

   0101 0001
    5 & 1             =51

   1001 0010
    9 & 2             =92

   0110 1000
    6 & 8             =68
```

In this decimal mode the 6502, when adding numbers, automatically does an internal carry between bits four and five, if the number exceeds the decimal value for that block i.e. 9. This mode is activated when the (D) flag is set to one in the (P) register.

This completes most of the basic ground-work needed for a thorough understanding of machine code, next time we will be covering assembly language programming in some depth.

## ATARI XL/XE SOFTWARE

## EXCEL DISK MAGAZINE

EXCEL 23 - out March 1st.........£2.50
A SIX ISSUE SUBSCRIPTION IS ONLY £15!
BACK ISSUES are all £2.50 each, ANY 5 for £10. or 20 issues for ONLY £30!!!

### THE LATEST XL/XE PD/SOFWARE DISKS

Double sided/single density unless stated otherwise.
All £1.50 or 5 for £10. Send a stamp for a huge list!
ANALOG TEXT ADVENTURES:12 Aventure games in Basic & machine code! (1050)
TRIVIA TREK: An excellent trivia quiz!
VENGEANCE OF XIOP: Space War simulation!
DUNGEON MAKER & KRISS KROSS: Dungeons and Dragon's playing aid plus a super problem solving puzzle program! (1050)
DOS 2.5: German and Spanish versions!
ADVENTURE COMPANION 3 & 4: More help for those stuck on adventure games!
MUSIC SHOW DEMO +7: More great demos!
THE TOP PART 2: Plus more great demos!
TEXAS POKER & YANKEE BLACKJACK: Card games with documentation etc.!
MEGABLAST & NFL FOOTBALL: Two players blasting plus US football sim! (1050)
AUTOCROSTIC & CROSSWORD: Word puzzles!

Some PD games available on tape!

All available from:
R.STUART: 21, STRONSAY WAY,
BROOMLANDS, IRVINE,
STRATHCLYDE, KA11 1HZ, SCOTLAND

All cheques/PO's payable to R.STUART - Overseas please add £1 for pp

## FUTURA :

## XL/XE NEWSLETTER

Each issue is packed with

ARTICLES and SOFTWARE
Articles include 8-bit info, news, trivia, profiles, reviews, programming DIAMOND GOS and VCS columns, etc...
Software includes the latest UTILITIES, GAMES, and DEMOS.
Send now for the latest issue:
Disk......................................£1.95
Printed copy & cassette.....£3.95
Cassette of programs.........£2.95
Please make cheques/POs payable to S.J.MURRAY
and send to -

Stuart J.Murray,
North of Scotland Atari User Group,
71 Walker Road,
TORRY, Aberdeen AB1 3DL

## BOOT !

## The Official Newsletter of LACE

THE LONDON ATARI COMPUTER ENTHUSIASTS.

As a member of LACE you will have access to: A monthly group meeting within the Greater London area. A quarterly magazine containing solely 8-bit orientated articles. A yearly Cover Disk/Cassette of exclusive PD. Discount on any PD purchased from our library. a dedicated menu on the City BBS.

All this can be yours for £7.00 per year (4 issues of BOOT).

Membership application form can be optained by sending your name address to :

L.A.C.E.
143 Richmond Road
Leytonstone
London E11 4BT

## 8:16

The Alternative Atari Newsletter

### The Bournemouth and Poole Atari User Group

The BAPAUG club produces a quarterly newsletter for the 8 and 16 bit user. The subscription rates for the 4 issues is £6.50.

Anyone wishing to receive the 8:16 newsletter should send their address to:

Mr.Colin Hunt
246 Wimborne Road
Oakdale, Poole
Dorset BH15 3EF

## TIMES TABLE PRINTOUT

```
1*1=1        2*1=2        3*1=3        4*1=4        5*1=5
1*2=2        2*2=4        3*2=6        4*2=8        5*2=10
1*3=3        2*3=6        3*3=9        4*3=12       5*3=15
1*4=4        2*4=8        3*4=12       4*4=16       5*4=20
1*5=5        2*5=10       3*5=15       4*5=20       5*5=25
1*6=6        2*6=12       3*6=18       4*6=24       5*6=30
1*7=7        2*7=14       3*7=21       4*7=28       5*7=35
1*8=8        2*8=16       3*8=24       4*8=32       5*8=40
1*9=9        2*9=18       3*9=27       4*9=36       5*9=45
1*10=10      2*10=20      3*10=30      4*10=40      5*10=50
1*11=11      2*11=22      3*11=33      4*11=44      5*11=55
1*12=12      2*12=24      3*12=36      4*12=48      5*12=60
1*13=13      2*13=26      3*13=39      4*13=52      5*13=65
1*14=14      2*14=28      3*14=42      4*14=56      5*14=70
1*15=15      2*15=30      3*15=45      4*15=60      5*15=75


6*1=6        7*1=7        8*1=8        9*1=9        10*1=10
6*2=12       7*2=14       8*2=16       9*2=18       10*2=20
6*3=18       7*3=21       8*3=24       9*3=27       10*3=30
6*4=24       7*4=28       8*4=32       9*4=36       10*4=40
6*5=30       7*5=35       8*5=40       9*5=45       10*5=50
6*6=36       7*6=42       8*6=48       9*6=54       10*6=60
6*7=42       7*7=49       8*7=56       9*7=63       10*7=70
6*8=48       7*8=56       8*8=64       9*8=72       10*8=80
6*9=54       7*9=63       8*9=72       9*9=81       10*9=90
6*10=60      7*10=70      8*10=80      9*10=90      10*10=100
6*11=66      7*11=77      8*11=88      9*11=99      10*11=110
6*12=72      7*12=84      8*12=96      9*12=108     10*12=120
6*13=78      7*13=91      8*13=104     9*13=117     10*13=130
6*14=84      7*14=98      8*14=112     9*14=126     10*14=140
6*15=90      7*15=105     8*15=120     9*15=135     10*15=150


11*1=11      12*1=12      13*1=13      14*1=14      15*1=15
11*2=22      12*2=24      13*2=26      14*2=28      15*2=30
11*3=33      12*3=36      13*3=39      14*3=42      15*3=45
11*4=44      12*4=48      13*4=52      14*4=56      15*4=60
11*5=55      12*5=60      13*5=65      14*5=70      15*5=75
11*6=66      12*6=72      13*6=78      14*6=84      15*6=90
11*7=77      12*7=84      13*7=91      14*7=98      15*7=105
11*8=88      12*8=96      13*8=104     14*8=112     15*8=120
11*9=99      12*9=108     13*9=117     14*9=126     15*9=135
11*10=110    12*10=120    13*10=130    14*10=140    15*10=150
11*11=121    12*11=132    13*11=143    14*11=154    15*11=165
11*12=132    12*12=144    13*12=156    14*12=168    15*12=180
11*13=143    12*13=156    13*13=169    14*13=182    15*13=195
11*14=154    12*14=168    13*14=182    14*14=196    15*14=210
11*15=165    12*15=180    13*15=195    14*15=210    15*15=225
```

(C) RALPH BRADLEY 1990/2