

COMPLEMENT AND ADD  
MK14 USERS NEWSLETTER

FEBRUARY-MARCH 1979

NO.1

Welcome to the first newsletter!

AS, if you have been sent the copy, you have seen, there are two newsletters circulating.

This is because I think the chain would be too long without the split, currently the number of members is 12 - and this will probably increase to 20-30 within the next month.

The aim of this newsletter will be:

- a) To provide a means of communication between users.
- b) To help newcomers to SC/MP with their Hardware and Software teething problems.
- c) To present any information supplied by Science of Cambridge and National Semiconductor.

The newsletter is free at present because the major cost - postage is shared amongst members. But a few stamps would always be helpful!

When you add your letters to the newsletter, keep them separate and then I can transfer them to another circulation.

By all means tag on your own software/ideas/problems.

Perhaps the first time round, you can introduce yourself and explain what you are doing with your machine and what extra hardware you have connected. ( Ram, Ram I/O, VDU, Floppy Disks etc?)

Can anyone think of a better name for the club than MK14 USERS CLUB?, or perhaps a better name than COMPLEMENT AND ADD.

The name is taken from a SC/MP instruction, and hopefully you will both complement the newsletter and add to it.

I have sent off letters to ETI PCW and S of C so perhaps we can attract a wider audience.

Probably the best way of exchanging software (programs ideas etc) is to mention what programs etc you have available and then those interested can say so. Obviously if the program is quite short, it can be added to the newsletter, otherwise the author could start his own mini circulation, after noting who wants what.

Don't forget also to say what hardware it requires i.e. RAM I O extra ram, Graph plotter...

To conserve paper, I have used both sides of the paper-hope you do not mind.

Whilst meetings may be out of the question, though I have met one member and spoken to three, have a look at the circulation lists to determine whether perhaps someone is within a train ride from you!



### Number Crunching

Has anyone devised a 32 bit divide routine?  
Somewhere I have a 32 bit multiply routine if anyone wants it.  
There is something fascinating about producing long primes!

### Games

Games are my speciality! (he said modestly!)

a) Maze. The program simulates a maze by asking you which direction you want to travel. I am currently improving this program to add a bit more fun.

b) Horserace.  
Three horses travel to the right of your display!  
The first one to fall off the edge wins.

c) Noughts and Crosses.  
This is the boring version which I adapted from a Basic program. The machine starts and wins or draws.  
The above 3 programs only need the 256 chunk.

d) Mastermind.  
This is the conventional machine type game but played with the MK14 working out your no.s. Needs extra Ram.

e) Quadpaw.  
I'm not surprised you haven't heard of it, as I invented this board game from an idea in Creative Computing.  
In the next newsletter I will give more details, if you are interested, as the basic idea can be extended to any small board game.

### Useful Utilities

a) Insert routine.  
Inserts one byte at the address given by P2.  
Very useful for when you discover that you had forgotten one byte.

b) Delete routine.  
Deletes one byte for when you enter the same thing twice.

These two occupy 8-9 bytes each! The person who can provide the shortest program will be first on the next circulation list!  
(Not Tony Stillwell who already has the solution)

c) I have written a fairly lengthy labels program that allows you to jump to a label, rather than work out displacements.

d) Very Quick Input Routine.  
This lets you enter in a program to XX12-FF by just entering in the program without all the MEMS and TERMS.

e) Decimal 6 digit to Binary routine. Uses about 30 bytes.

f) 60 byte BASIC with no instructions! It just says 'ready' or 'eh' depending on what is pressed.

I also have the modifications to TWONKY (see ETI) which enable the



I have connected up a surplus type keyboard to my machine, via the edge connector. So now I can enter my diary onto a cassette!

Over the next few weeks I shall dig up the ETI that gave the edge connector details, to appear in the next newsletter.

You might be interested to know how I added another  $\frac{3}{4}$ K to the board. I cheated by soldering the 2111s on top of each other, with the exception of the enable pins which were taken to some external log  $\phi$ .

I will go into more details if anyone is prepared to make their machine look untidy.

#### CASSETTE INTERFACE

It might help some of you to know that I managed to get the interface working by adding a 2Meg resistor to the output of the cassette (earphone output) and changing the 470 resistor for a 470K. I also believe that the leds are shown back to front.

The positive side of the diode is the broad white stripe.

To show how reliable it is, you can even speed it up by 4 times, simply by reducing the delays by a factor of 4.  
(8-2, 20-8 etc)

#### MUSIC

I have not experimented with computer music much, apart from TWONKY the only contribution being an addition to MUSIC-BOX which lets you enter and play the tune as you go along.

#### THE NEXT NEWSLETTER

I shall try to get the next newsletter out by the end of March, and it would be a help if the newsletters were returned by then. If you have some urgent material for the next round, then just send it directly to me.

The next newsletter will contain any software you asked for some tips in the use of boolean algebra and something on the use of interrupts in SC/MP.

Also it will contain your letters of course, as not everyone have seen them the first time round.

Overleaf is a suggestion of the use of the monitor, intended for those who have only had the machine for a short while.

I look forward to hearing your views on this newsletter and your contributions to it.

Geoff Phillips,

London,



### Make Full Use of Your Monitor

From the moment you switch on a MK14 you are using the monitor contained within the 74LS571 proms.

It lets you enter your programs into the memory and execute them from any point you want.

If you have ever tried to input and output numbers, you should appreciate how hard your monitor works!

But apart from developing your programs using the monitor, you can use the monitor within your programs, as outlined below.

a) To translate a hexadecimal number to the 7 segment format.

There is a table at 010B which contains the code needed for every number between 0 and F.

To use this table, set P1 to 010B, load the number into the extension register, and then load from the pointer plus 80.

That is load from P1 plus the contents of e.

The value left in the accumulator will be the code required.

b) To Display a Message and input a number

Load 0184 to P3. Then load 0F00, or any convenient ram address to P2. Then put the message at 0F00-7 or wherever your ram is. This message will be in the 7 segment code and if at 0F00, will have to be stored by a program.

Finally jump to the routine with a 3F instruction.

If a command key is pressed, the monitor will return you to the instruction following the 3F; else if a number key is pressed then control will be passed to the third byte following the 3F. The number pressed will be left in the extension register.

c) To Display 6 digits and input a number

The method is as above, except that 013F is loaded to P3.

The first 2 digit no. is stored at P2 plus 0C, the next at P2 plus 0E and the last at P2 plus 0D. If P2 points to 0F00 then if you store at 0F0C:- 014523 then '0123 45' will be displayed when 3F is executed.

d) To Preset a register before entering your program

You can set any SC/MP register before running your program, as an alternative to actually setting the register within the program. To see where you have to put your values, see page 41. You will see that e.g. 0FFE is the extension reg., so that if you enter 0FFE 29 and then run a program, the value 29 will be loaded into the extension before the monitor passes control to your program.



I will leave most of this page blank so that you can add comments.

I hope that this newsletter is what you hoped it to be, if not then feel free to say what should be in it!

IN the next newsletter, which I will try to get out by early March I will put in some notes on interrupts in the SC/MP and some programming tips.

A large part of the next newsletter will be this newsletter! This is because, hopefully, interesting comments will have been added which would not have been seen by everyone, the first time round.

ANY

CHANGE

OF THE

TWOXRY

CONVERSIONS?

M. SEAMAN.

Page 1

COMPLEMENT AND ADD  
MK14 USERS NEWSLETTER  
FEBRUARY MARCH 1979

NO.1

Welcome to the first newsletter!

As, if you have been sent the copy, you have seen, there are two newsletters circulating.

This is because I think the chain would be too long without the split, currently the number of members is 12 - and this will probably increase to 20-30 within the next month.

The aim of this newsletter will be:

- a) To provide a means of communication between users,
- b) to help newcomers to SC/MP with their Hardware and Software teething problems.
- c) To present any information supplied by Science of Cambridge and National Semiconductor.

The newsletter is free at present because the major cost - postage is shared amongst members. But a few stamps would always be helpful!

When you add your letters to the newsletter, keep them separate and then I can transfer them to another circulation.

By all means tag on your own software/ideas/ problems.

Perhaps the first time round, you can introduce yourself and explain what you are doing with your machine and what extra hardware you have connected. ( Ram, Ram I/O, VDU, Floppy Disks etc?)

Can anyone think of a better name for the club than MK14 USERS CLUB?, or perhaps a better name than COMPLEMENT AND ADD.

The name is taken from a SC/MP instruction, and hopefully you will both complement the newsletter and add to it.

I have sent off letters to ETI POW and S of C So perhaps we can attract a wider audience.

Probably the best way of exchanging software programs ideas etc) is to mention what programs etc you have available and then those interested can say so. Obviously if the program is quite short, it can be added to the newsletter, otherwise the author could start his own mini circulation, after noting who wants what.

Don't forget also to say what hardware it requires i.e. RAM I/O extra ram, Graph plotter..

To conserve paper, I have used both sides of the paper - hope you do not mind.

Whilst meetings may be out of the question, though I have met one member and spoken to three, have a look at the circulation lists to determine whether perhaps someone is within a train ride from you!

2)

Number Crunching

Has anyone devised a 32 bit divide routine? Somewhere I have a 32 bit multiply routine if anyone wants it. There is something fascinating about producing long primes!

## Games

Games are my speciality! (he said modestly!)

- a) Maze. The program simulates a maze by asking you which direction you want to travel. I am currently improving this program to add a bit more fun.
  - b) Horserace. Three horses travel to the right of your display! The first one to fall off the edge wins.
  - c) Noughts and Crosses. This is the boring version which I adapted from a Basic program. The machine starts and wins or draws.
- The above 3 programs only need the 256 chunk.

## d) Mastermind

This is the conventional machine type game but played with the MK14 working out your no.s. Needs extra Ram.

e) Quad pawn. I'm not surprised you haven't heard of it, as I invented this board game from an idea in Creative Computing. In the next newsletter I will give more details, if you are interested, as the basic idea can be extended to any small board game.

## Useful Utilities

### a) Insert routine.

Inserts one byte at the address given by P2. Very useful for when you discover that you had forgotten one byte.

### b) Delete routine.

Deletes are by te for when you enter the same thing twice.

These two occupy 8-9 bytes each! The person who can provide the shortest program will be first on the next circulation list! (Not Tony Stillwell who already has the solution!)

c) I have written a fairly lengthy labels program that allows you to jump to a label, rather than work out displacements. What you do is code A0 nn where nn is a label. Then whenever you want to jump to that location you just enter nn as the second part of the instruction.

e.g.

C4 10

A0 55

02

F4 FF

9C 55

gets translated to

C4 10

02

F4 FF

9c F9. The extra ram and ram I/O are needed though you could get by after changing the program

### d) Very Quick Input Routine.

This lets you enter in a program to XX12-FF by just entering in the program without all the MEMS and TERMS.

e) Decimal 6 digit to Binary routine. Uses about 30 bytes.

f) 60 byte BASIC with no instructions! It just says ready' or 'eh' depending on what is pressed.

I also have the modifications to TWONKY (see ETI) which enable it to run with extra RAM. Twonky for those who do not read ETI is a music generator rather like a musical doorbell that has gone wrong!

I've written a program that lets you put your diary on cassette using an external keyboard and cassette interface.

Has anyone built the prom blower board yet? I'm sure we would all be interested to know of any difficulties encountered when building something onto the MK14.

You might be interested to know that I added my extra memory by actually putting in a new ram chip (2111) onto the existing ones (after removing them from the board). As long as you solder quickly and carefully, you should be okay. The enables on the ram chip must be taken to some extra decoding logic. I will go into more details if anyone is interested and is prepared to make their machine look untidy!

(3)

I have connected up a surplus type keyboard to my machine, via the edge connector. So now I can enter my diary onto a cassette!

Over the next few weeks I shall dig up the ETI that gave the edge connector details, to appear in the next newsletter.

You might be interested to know how I added another K to the board. I cheated by soldering the 2111s on top of each other, with the exception of the enable pins which were taken to some external logic

I will go into more details if anyone is prepared to make their machine look untidy.

#### CASSETTE INTERFACE

It might help some of you to know that I managed to get the interface working by adding a 2Meg resistor to the output of the cassette (earphone output) and changing the 470R resistor for a 470K I also believe that the leds are shown back to front. The positive side of the diode is the broad white stripe. To show how reliable it is, you can even speed it up by 4 times, simply by reducing the delays by a factor of 4. (8-2, 20-8 etc)

#### MUSIC

I have not experimented with computer music much, apart from TWINKY the only contribution being an addition to MUSIC-BOX which lets you enter and play the tune as you go along.

#### THE NEXT NEWSLETTER

I shall try to set the next newsletter out by the end of March, and it would be a help if the newsletters were returned by then. If you have some urgent material for the next round, then just so it directly to me.

The next newsletter will contain any software you asked for, some tips in the use of boolean algebra and something on the use of interrupts in SC/MP.

Also it will contain your letters of course, as not everyone. have seen them the first time round.

Overleaf is a suggestion of the use the monitor, intended for those who have only had the machine for a short while.

I look forward to hearing your views on this newsletter and your contributions to it,

Geoff Phillips, [my old home address removed]

(4)

#### Make Full Use of Your Monitor.

The monitor program lets you enter numbers and confirms it visually. It also lets you execute your program from anywhere in memory. Without it, you would have to enter data via switches and observe the results on LEDs. So the monitor



within your 74LS571's is essential to your machine. But apart from using it for developing your programs you can use it within your programs as outlined below.

a) To translate a hexadecimal number to the 7 segment format:

There is a table at 010B which contains the code needed for every number between 0 and F. To use this table, set P1 to 010B, load the number into the extension register, and then load from the pointer plus 80. That is load from P1 plus the contents of e. The value left in the accumulator will be the 7 segment code required.

b) To Display a Message and input a number:

Load 0184 to P3. Then load 0F00, or any convenient ram address to P2. Then put the message at 0F00-7 or wherever your ram is. This message will be in the 7 segment code and if at 0F00, will have to be stored by a program. Finally jump to the routine with a 3F instruction (exchange ptr 3) The program will then jump to a part of the monitor and your message displayed. When a key is pressed control will be passed back to i) The instruction immediately following the 3F if MEM GO or TERM. Or ii) Two bytes following the 3F if a number key was pressed. So you could verify that a number key was pressed by following 3F by 90 FD (jump -3). The number pressed will be left in the extension register.

c) To display a two digit number and wait for an input as above set p2 to 0f00 and p3 to 013F and store the number at 0F0D. The same will happen as above except that the number will be converted and displayed on the right. You could also display a 4 digit number on the left by storing the high part of the number at 0F0C and the low part at 0F0E.

Suppose you store 12 at 0F0C 34 at 0F0E and 56 at 0F0D, 1234 56 will be displayed when you execute 3F.

d) A number input by your program can be translated to an actual number by loading 1BD into P3 and the number into the accumulator. See page 46-47. [of the instruction manual] In all these display and input routines, the number input is left in the extension register.

e) To Preset a register before entering your program

You can set any SC/MP register before running your program, as an alternative to actually setting the registers within that program. To see where you have to put your values, see page 41. You will see that e.g. OFFE is the extension reg so that if you enter OFFE 29 and then run a program, the value 29 will be loaded into the extension before the monitor passes control to your program.