

C O M P L E M E N T A N D A D D

APRIL - MAY 1979

No. 2

NEWS

Hello Again!

First of all, I will keep you all informed of what's been happening in the world of SC/MP.

Since the first newsletter, I have been busy writing to all the Computing magazines to get news of the club published.

Practical Computing will definitely put in details of the club/newsletter, and maybe an article I sent.

Micom, who market software, are interested in a Mastermind program - if they can sell the program, club funds will be boosted considerably.

Eti said over the phone that they would say something in a forthcoming issue about the club.

WITH THIS NEWSLETTER

As a member of National Semiconductor's SC/MP users group, I have put in an issue of COMPUTE into every circulation (a different one in each of course!)

Also, some details of the MICOM assembler.

I cannot re-direct the additions to the first newsletter as I have recieved none back!

I have also included a listing of either my mastermind or maze program-at the risk of filling up too much space in the envelope.

HAVE YOU RECIEVED THE FIRST NEWSLETTER?

I have shuffled round the circulation lists this month so that if the newsletter or its copies are being held up somewhere, at least the same people will not suffer.

When the magazines publish the existance of the club, it may be worthwhile to let individuals control say two groups of users. Several of you have offered help in this way, and I would be grateful if when our numbers increase you could perhaps photocopy the newsletter for your groups - those who wish to control a group.

It might be advantageous to work on a regional level, that is I could deal with the South-East, another controlling the North etc.

For the moment though, there isn't really any need to split up the club in such a way.

SC/MP - 3?

Those of you with the Jan-Feb issue of Compute will see the headline 'National Develops'. The article then describes the 8070 and 8072 as :

- 1) Offering 64 bytes of ram; not to mention 2.5K of rom on the 8072.
- 2) 40-pin package, single 5V supply.
- 3) Stack Operation, 16 bit arithmetic(multiply divide..)
- 4) Ascii to bcd conversion in one instruction!
- 5) Two sense input, 3 flag outputs (sounds familiar??)
- 6) Bus features that sound an awful lot like the SC/MP2.

I shall write off for further info soon and keep you informed on developments. With a bit of luck the 8070 will be hardware compatible with SC/MP 2.

MICOM

The Micom assembler offers to reduce programming time by cutting out endless displacement calculations. The package I recieved was certainly professionally done, with lots of info, examples etc.

However, I found it a bit heavy going and feel that it is just too much bother to

- 1) code on special forms
- 2) keep loading different programs.

Some of you may find the package useful, of course only those of you with cassette interfaces.

I could not put the whole of the package in the newsletter, simply because there is so much of it, and I feel I may be infringing their copyright.

If anybody would like the whole package, then send me a large envelope and a stamp.

VDU

Shortly, I will be connecting up a Practical Electronics VDU to my MK14 to see what TV games I can invent.

An account of the mess I get into will appear in a later newsletter.

The info on Interrupts has been saved for the next time as this issue is large enough already.

MEETINGS

Is there any possibility of meeting each other? Probably not all at once - but several of us live in London, there are also groups in Devon/Cornwall.

USING BOOLEAN ALGEBRA

A quick glance through the SC/MP instruction set will describe such goodies as 'Exclusive or' (not used in rowing)
'Rotate right' (not a dance step)

This page will help you decide when you need to use any particular instruction within your program.

Instructions come under two main types :- single byte and double byte. The only difference really is that the double byte instruction needs an extra operand, that is to say that the first byte cannot supply all the information.

Take an add instruction; If you want to add 20 to your accumulator or you would code F4 (load imm.) and then follow it with the second byte - 20.

AND

The and instruction appears in both single and double byte formats, but this just means for single byte instructions the op-code on its own is enough.

The and function compares the accumulator with the contents of a memory location, or the extension register.

Where corresponding bits are set to 1, a 1 is left in the accumulator. That is, both bits must be 1 for the result to be 1.

E.g. 53 (0101 0011) and 29 (0010 1001) is 01 (0000 0001)
So already we've covered 10 possible op-codes!

One is AND IMMEDIATE which 'and's the acc. with the second byte of the instruction. Another is AND EXTENSION which AND's the accumulator with the extension register.

The remaining 7 cover the possibility of wanting the operand to be contained someplace in memory.

If for instance you want to and with the contents of OB66 you could do it by setting up register P2 to contain OB66 (Load OB XPAH Load OO XPAL) then all you do is add the pointer no. to D0-the base code, arriving at D2, and follow it by the 'displacement' - 66.

If you used P3 (there is no law against it), then code D3.

OR

The OR instruction appears in all the formats of the 'and'. It compares the accumulator with a second operand (extension reg. or contents of some address or an actual number) and leaves a 1 in the accumulator wherever at least one corresponding bit was set. E.g. 47 (0100 0111) OR 21 (0010 0001) gives: 67 (0110 0111)

EXCLUSIVE-OR

As above for OR except that each corresponding pair of bits must be different for a 1 to be set in the result.

E.g. 37 (0011 0111) XOR A5 (1010 0101) gives 92 (1001 0010)

USING BOOLEAN ALGEBRA CONTD

So what use are these weird logic functions?

Well, for a start don't forget that the SC/MP works on 8 bits at a time. It will quite happily add together lots of 8 bits, or store/load a byte.

But how do you breakdown a byte? Well the method is to use our new-found knowledge in Boolean Algebra. To isolate a bit/bits we have the and function to mask out the bits we want. So if we just want the least significant bit of the accumulator, e.g. as an odd or even test, we can and immediate with 01. This leaves us with either 0 or 1 in the accumulator, the other bits have been wiped out.

To force a bit to appear we must use the OR facility e.g. to make sure the flags are set we must or the status register with 07 - leaving the other bits as unchanged.

To invert a bit within a byte, use EXCLUSIVE-OR, leaving the corresponding bits-to-be-changed as 1.

Another major use of EX-OR is the comparison of 1 byte with another.

If we want to find out if the accumulator contains, say 28, all we must do is ex-or with 28 and test for 00 (00 when 28).

If some of these are not clear, then experiment with the machine or on paper.

Programs are usually littered with operations of this kind (can you see why D4 OF is useful), and the lack of them in Higher level languages (e.g. Basic, Cobol) provides at least one reason for their less efficiency.

A later article will be concerned with rotating/shifting and the various addressing modes available. You can never get away from using AND/OR/etc, after all, the SC/MP is really one large TTL package!!

TWONKY CONVERSIONS FOR MK14

The hardware does not need to change, though I just used a NE555 oscillator connected to Sense B. Luckily the program has been written inefficiently, and we can gain an extra two bytes.

The program occupies 0B00-FF and uses 0F00-FF as the tune playing store. An earphone or amplifier should be connected to Flag0.

OB01 C4 0F , OB09 CA 00 (mistake) OB0B C4 0B , OB19 C4 0F,
OB50 E4 07 , 98 05 , E4 07 32 90 B8 C4 EF 32 C4 0B 35
OB5F C4 0F , OBB2 C4 0B , OBEE 08.

Exec. 0B00.

ENOUGH POWER?

I used to have power supply problems - transformer or regulator over-heating, but since have invested in an ex-equipment 5V supply. The ad claims to the supply provides 5V or 7½V at 10A, though I doubt I've exceeded 3A.

The advert was still in the last ETI - the cost? £5.50 postage £1.25., which seems quite reasonable. The firm is Canon Components, 322,324 whitehorse Rd, Croydon, Surrey, CR0 2LF. 'phone 01-684 9872.

QUICK INPUT ROUTINE

The following routine saves time by allowing you to enter programs without the tedious Mems-Terms.

It can be loaded into any part of memory, and lets you enter a program into 0712-FF. The first 12₁₆ bytes are used by the monitor. It does not show the address being incremented but just a slight flicker for half a byte and a longer flicker for a whole byte. ABORT pressed at any time will terminate the program and give you the last location used. So to start at 0F12, this value will be 11.

Start B00	nn	First locn - 1. i.e. last loc'n used.
B01	half-byte	
B02	C4 0F 36	P2 high gives program area, change as needed.
B05	C4 00 32	
B08	C4 01 37	
BOB	C4 3F 33	P3 points to monitor routine.
BOE	3F	Jump to monitor to await number.
BOF	90 FD	Ignore command keys!
B11	8F 66	Small delay.
B13	40 C8 EC	Put number pressed into half-byte.
B16	3F	Get low part of number.
B17	90 FD	Ignore if not no.
B19	C0 E7	Get half-byte.
B1B	1E 1E 1E 1E	Turn it round the right way.
B1F	58	(classic example of <u>OR</u>) or extn.
B20	01	Remember the complete no. in extn.
B21	A8 DE	Get nn - next location in this block.
B23	32	Set up P2 to this loc'n.
B24	40 CA 00	Put number input into next loc'n
B27	8F FF 8F 77	Longer delay - indicates whole byte.
B2B	90 D5	Back for next byte!

Obviously its easier to miss out a byte using this method, so to insert a byte use the routine:

C6 01 01 CA FF E4 88 9c F7 where OFFB,C points to the loc'n and the program is terminated by 88.

To remove a byte : C6 01 cA FE E4 88 9C F8 3F, where as above OFFB,C points to the right place and the prog. ends with 88. Note 3F should have been put after the first routine (sorry).

If you have any complaints about the newsletter then don't hesitate to write. (bad speellin' for instance)

I will try to get hold of some information on the 8072 (SC/MP3) from Nat Semi, and maybe a sample!!

PROBLEM If you have any MK14 puzzles, then send them either in the newsletter or directly to me.

I.e. step 1 shows 'OFxx xx'

step 3 shows '04xx xx'

The Mk14 is very good at problem solving - also very fast. I persuaded it to work out a trial and error problem, and it only took 3 seconds to go through all possible values.
(00.00-£99.99)

G. Phillips

London,

????????????????????

COMPLEMENT AND ADD
APRIL - MAY 1979
No. 2

[OCR version with checked hex code]

Hello Again! First of all, I will keep you all informed of what's been happening in the world of SC/MP.

Since the first newsletter, I have been busy writing to all the Computing magazines to get news of the club published.

Practical Computing will definitely put in details of the club/newsletter, and maybe an article I sent.

Micom, who market software, are interested in a Mastermind program - if they can sell the program, club funds will be boosted considerably.

Eti said over the phone that they would say something in a forthcoming issue about the club.

WITH THIS NEWSLETTER

As a member of National Semiconductor's SC/MP users group, I have put in an issue of COMPUTE into every circulation (a different one in each of course!)

Also, some details of the MICOM assembler. I cannot re-direct the additions to the first newsletter as I have recieved none back!

I have also included a listing of either my mastermind or maze program-at the risk of filling up too much space in the envelope.

HAVE YOU RECIEVED THE FIRST NEWSLETTER?

I have shuffled round the circulation lists this month so that if the newsletter or its copies are being held up somewhere, at least the same people will not suffer. When the magazines publish the existence of the club, it may be worthwhile to let individuals control say two groups of users. Several of you have offered help in this way, and I would be grateful if when our numbers increase you could perhaps photocopy the newsletter for your groups - those who wish to control a group.

It might be advantageous to work on a regional level, that is I could deal with the South-East, another controlling the North etc.

For the moment though, there isn't really any need to split up the club in such a way.

2)

SC/MP - 3? Those of you with the Jan-Feb issue of compute will see the headline National Develops The article then describes the 8070 and 8072 as :

1) Offering 64 bytes of ram; not to mention 2.5K of rom on the 8072.

2) 40-pin package, single 5V supply.

3) Stack Operation, 16 bit arithmetic (multiply divide..)

Ascii to bcd conversion in one instruction!

5) Two sense input, 3 flag. outputs (sounds familiar??)

6) Bus features that sound an awful lot like the SC/MP2.

I shall write off for further info soon and keep you informed on developments. With a bit of luck the 8070 will be hardware compatible with SC/MP 2.

MICOM

The Micom assembler offers to reduce programming time by cutting out endless displacement calculations. The package I received was certainly professionally done, with lots of info, examples etc.

However, I found it a bit heavy going and feel that it is just too much bother to

- 1) code on special forms
- 2) keep loading different programs.

Some of you may find the package useful, of course only those of you with cassette interfaces.

I could not put the whole of the package in the newsletter, simply because there is so much of it, and I feel I may be infringing their copyright.

If anybody would like the whole package, then send me a large envelope and a stamp.

VDU

Shortly, I will be connecting up a Practical Electronics VDU to my MK14 to see what TV games I can invent.

An account of the mess I get into will appear in a later newsletter.

The info on Interrupts has been saved for the next time as this issue is large enough already.

MEETINGS

Is there any possibility of meeting each other? Probably not all at once - but several of us live in London, there are also groups in Devon/Cornwall.

3)

USING BOOLEAN ALGEBRA

A quick glance through the SC/MP instruction set will describe such goodies as 'Exclusive or' (not used in rowing)

Rotate right! (not a dance step) This page will help you decide when you need to use any particular instruction within your program. Instructions come under two main types :- Single byte and double byte. The only difference really is that the double byte instruction needs an extra operand, that is to say that the first byte cannot supply all the information.

Take an add instruction; If you want to add 20 to your accumulator you would code F4 (load imm.) and then follow it with the second byte - 20.

AND

The and instruction appears in both single and double byte formats, but this just means for single byte instructions the op-code on its own is enough.

The and function compares the accumulator with the contents of a memory location, or the extension register.

Where corresponding bits are set to 1, a 1 is left in the accumulator. That is, both bits must be 1 for the result to be 1.

E.g. 53 (0101 0011) and 29 (0010 1001) is 01 (0000 0001) So already we've covered 10 possible op-codes! One is AND IMMEDIATE which and's the acc. with the second byte of the instruction. Another is AND EXTENSION which AND'S the accumulator with the extension register. The remaining to cover the possibility of wanting the operand to be contained some place in memory.

If for instance you want to and with the contents of 0B66 you could do it by setting up register P2 to contain 0B66 (Load 0B XPAH Load 00 XPAL) then all you do is add the pointer no. to D0-the base code, arriving at D2, and follow it by the 'displacement' - 66. If you used P3 (there is no law against it), then code D3.

The OR instruction appears in all the formats of the 'and'. It compares the accumulator with a second operand extension reg. or contents of some address or an actual number) and leaves a 1 in the accumulator wherever at least one corresponding bit was set. E.g. 47 (0100 0111) OR 21(0010 0001) gives:67 (0110 0111)

EXCLUSIVE-OR

As above for OR except that each corresponding pair of bits must be different for a 1 to be set in the result.

E.g. 37 (0011 0111) XOR A5 (1010 0101) gives 92 (1001 0010)

4)

USING BOOLEAN ALGEBRA CONTD

So what use are these weird logic functions? Well, for a start don't forget that the SC/MP works on 8 bits at a time. It will quite happily add together lots of 8 bits, or store/load a byte.

But how do you breakdown a byte? Well the method is to use our new found knowledge in Boolean Algebra. To isolate a bit/bits we have the and function to mask out to the bits we want. So if we just want the least significant bit of the accumulator, e.g. as an odd or even test, we can and immediate with 01. This leaves us with either 0 or 1 in the accumulator, the other bits have been wiped out.

To force a bit to appear we must use the OR facility e.g. to make sure the flags are set we must or the status register with 07 - leaving the other bits as unchanged.

To invert a bit within a byte, use EXCLUSIVE-OR, leaving the corresponding bits-to-be-changed as 1.

Another major use of EX-OR is the comparison of 1 byte with another.

If we want to find out if the accumulator contains, say 28, all we must do is ex-or with 28 and test for 00 (00 when 28).

If some of these are not clear, then experiment with the machine or on paper.

Programs are usually littered with operations of this kind (can you see why D4 0F is useful), and the lack of them in higher level languages (e.g. Basic, Cobol) provides at least one reason for their less efficiency.

A later article will be concerned with rotating/shifting and the various addressing modes available. You can never get away from using AND/OR/etc, after all, the SC/MP is really one large TTL package!!

TWONKY CONVERSIONS FOR MK14

The hardware does not need to change, though I just used a NE555 oscillator connected to Sense B. Luckily the program has been written inefficiently, and we can gain an extra two bytes.

The program occupies OB00-FF and uses OF00-FF as the tune playing store. An earphone or amplifier should be connected to Flag0.

OBO1 C4 OF, OB09 CA 00 (mistake), OBOB C4 OB, OB19 C4 OF, OB50 E4 07, 98 05, E4 07 32 90 B8 C4 EF 32 C4 OB 35

OB5F C4 OF, OBB2 C4 OB, OBEE 08.

Exec. OBOO.

5)

ENOUGH POWER?

I used to have power supply problems - transformer or regulator over-heating, but since have invested in an ex-equipment 5V supply. The ad claims to the supply provides 5V or 7.5V at 10A, though I doubt I've exceeded 3A.

The advert was still in the last ETI - the cost? £5.50 postage £1.25., which seems quite reasonable. The firm is Canon Components, 322, 324 whitehorse Rd, Croydon, Surrey, CRO 2LF. 'phone 01-684 9872.

QUICK INPUT ROUTINE

The following routine saves time by allowing you to enter programs without the tedious Mems-Terms.

It can be loaded into any part of memory, and lets you enter a program into 0?12-FF. The first 16 bytes are used by the monitor. It does not show the

address being incremented but just a slight flicker for half a byte and a longer flicker for a whole byte. ABORT pressed at any time will terminate the program and give you the last location used. So to start at OF12, this value will be 11. Start B00 nn First locn - 1. i.e. last loc'n used.

B01 half-byte

B02 C4 0F 36 P2 high gives program area, change as needed

B05 C4 00 32

B08 04 01 37

B0B C4 3F 33 P3 points to monitor routine.

B0E 3F Jump to monitor to await number.

B0F 90 FD Ignore command keys!

B11 8F 66 Small delay.

B13 40 C8 EC Put number pressed into half-byte.

B16 3F Get low part of number.

B17 90 FD Ignore if not no.

B19 CO E7 Get half-byte.

B1B 1E 1E 1E 1E Turn it round the right way.

B1F 58 (classic example of OR) or extn.

B20 01 Remember the complete no. in extn.

B21 A8 DE Get nn - next location in this block.

B23 32 Set up P2 to this locn.

B24 40 CA 00 Put number input into next locn

B27 8F FF 8F 77 Longer delay - indicates whole byte. B2B 90 D5 Back for next byte! Obviously its easier to miss out a byte using this method, so to insert a byte use the routine:

C6 01 01 CA FF E4 88 9C F7 where OFFB,C points to the loc'n and the program is terminated by 88.

To remove a byte : C6 01 CA FE E4 88 9C F8 3F, where as above OFFB,C points to the right place and the prog. ends with 88. Note 3F should have been put after the first routine (sorry).

6)

ANY COMPLAINTS?

AS I said before, I have not yet received back any newsletters, though plenty of other interesting correspondence. It is interesting that s. of C. have not actually contacted me but have instead referred people interested in a club to me.

If you have any complaints about the newsletter then don't hesitate to write. (bad speellin' for instance)

Our numbers should steadily increase with the publication in CT and (yet to come) Practical computing.

I will try to get hold of some information on the 8072 (S from Nat Semi, and maybe a sample!!

PROBLEM

If you have any MK14 puzzles, then send them either in the newsletter or directly to me.

How can you make the monitor suddenly change from displaying 'OFXX XX' to '04xx Xx' where xx is don't care or not telling-you!! You must not use any program to do it (only the monitor) and the display should change just by entering 'term'. I.e. step 1 shows 'OFXX XX'

step 2 press term (after pressing what you like) & before you release, the display should not change.

step 3 shows '04xx xx' Note: for those with new monitors, display alters as soon as 4 pressed!, but until 4 released display shows 'OFxx'

The M14 is very good at problem solving - also very fast. I persuaded it to work out a trial and error problem, and it only took 3 seconds to go through all possible values. (00.00-£99.99)

See you next newsletter!