BYTE NYBBLES

Jeremy O Jones/Alan Jones
Dept of Computer Science
Trinity College Dublin
Dublin 2 – IRELAND

A TMS-9900 Monitor

F000 MONIT(
F002 BOOT

Everyone has their own idea of what a good monitor should and should not do. The TMS-9900 monitor described here is aimed at a small system (without disks) with a terminal (64 by 32 screen size) for I/O (input/output). It has been designed such that programs (which may be cross-assembled elsewhere) can be debugged efficiently. To this end, the monitor contains an instant assembler, a disassembler and comprehensive user program tracing facilities. The instant assembler allows modifications in code to be made quickly, since calculating op codes is difficult because the op code fields are not aligned on nybble boundaries.

The monitor occupies slightly less than 256 bytes of memory and has been assembled to occupy hexadecimal locations F400 thru FFFE. Three workspaces are required in programmable memory and these are placed just below the monitor at F3E0, F3C0, and F3A0, respectively. A stack is required, and this starts just below the three workspaces. The stack is used for subroutine linkage and may be used by all user programs. The monitor also makes use of interrupt level 1 (for tracing) and the XOP vectors 0 to 5 inclusive (See figure 1). The description of the monitor is divided into two parts: how to use the monitor, and a general description of how the monitor has been implemented.

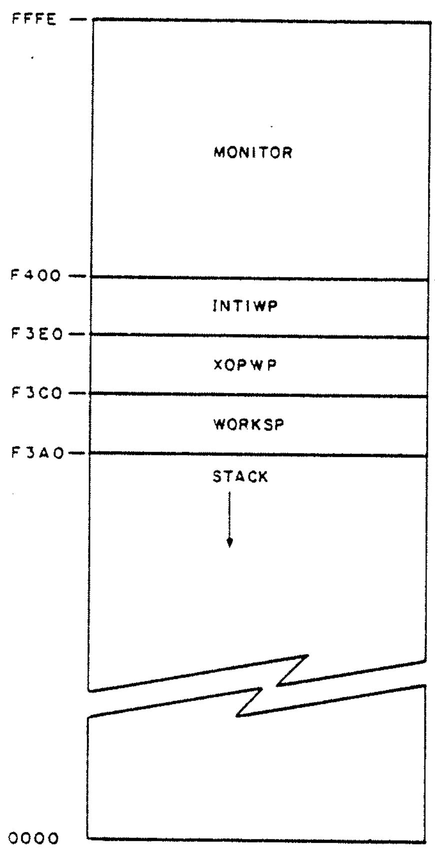

Figure 1: Memory usage in the Texas Instruments TMS-9900 system that uses the authors' monitor. The monitor occupies slightly less than 256 bytes of memory. Three workspaces are required, and they are placed just below the monitor. The stack, used for subroutine linkage, starts immediately below the three workspaces. The monitor also makes use of interrupt level 1 (for tracing) and the XOP vectors 0 to 5, inclusive.

How to Use the Monitor

When the monitor is started (by reset or load signals), it outputs << TMS9900 MONITOR >> to the terminal followed by the prompt >. The monitor is now ready to accept commands, all of which have a similar format:

COMMAND : : = {<HEXNO>} <COMMAND LETTER>

The optional hexadecimal number (only the last four digits count), unless otherwise stated, is taken to be a memory address. If no hexadecimal number has been input, then a default action takes place. The monitor keeps and updates two variables, low and last opened cell. The default action takes its address (hexadecimal number) from one of these. In all the examples that follow, the characters that the user inputs are underlined.

Command L

This command allows the monitor variable low to be set. The default action is to output the current value of low, else it is set to the hexadecimal number input. No other command changes the value of low, eg:

```
>2000L
>L
 LOW = 2000
>
```

Command ␣

This command (space) allows memory cells (locations) to be examined and changed. The default action is to open the last opened cell. It outputs the address of the opened cell and its current contents. To change the contents, input a hexadecimal number followed by a delimiter; otherwise, input a delimiter only. If a space is used as a delimiter, then the next cell is opened, a minus sign opens the previous cell, and any other character will return control to the monitor. This command updates the monitor variable last opened cell, eg:

```
>2000␣
 2000 1234␣
 2002 6789 05678␣
 2004 9ABC -
 2002 5678 <CR>
```

The symbol <CR> indicates a carriage return and is used throughout this article.

```
>C6P
 00C6     2CE0     XOP     @ ≠011P,3
 00CA     2CE0     XOP     @ ≠0108,3
 00CE     2CE0     XOP     @ ≠011D,3
 00D2     06A0     BL      @ ≠022E
 00D6     0204     LI      R4, ≠0013
 00DA     9064     CB      @ ≠0182(R4),R1
 00DE     1305     JEQ     ≠00EA
 00EO     0604     DEC     R4
 00E2     18FB     JOC     ≠00DA
 00E4     2CE0     XOP     @ ≠0126,3
 00E8     10F2     JMP     ≠00CE
 00EA     0A14     SLA     R4,1
 00EC     C124     MOV     @ ≠0196(R4),R4
 00F0     C0C3     MOV     R3,R3
 00F2     0454     B       *R4
 00F4     7FA0     SB      @ ≠0080,*R14+ <CR>
>
```

Listing 1: An example of the use of command P, a command in the authors' TMS-9900 monitor that lists the memory contents in disassembled form starting at the input address (in this case, hexadecimal C6). 16 disassembled instructions are listed. Inputting a space causes one more disassembled instruction to be output each time.

Command /

This command allows ASCII characters to be written into consecutive memory locations starting at the input address. The default action is to start at the last opened cell. Only characters greater than ASCII hexadecimal 20 (space) and also ASCII hexadecimal 00 may be input. Any other character (except ASCII hexadecimal 08 (backspace)) returns control to the monitor. The use of backspace allows a limited editing facility on the current line. This command updates the monitor variable last opened cell. Note that if ASCII 00 (control shift @) is input, then underscore is output.

```
>5010/
 5010 4142 /123456789ABCDEF0/
 5020 0000 /12 <CR>
>5010␣
 5010 3031 <CR>
>
```

Command P

This command lists memory contents in disassembled form starting at the input address. The default action is to start at address low. It lists 16 disassembled instructions, after which inputting further spaces will list one more disassembled instruction; any other character passes control back to the monitor (See listing 1).

Command .

This command enters the instant assembler mode. The default action is to open the last opened cell. The address of the opened cell, its current contents and a . are output. The assembly mnemonic is input *followed by a space.* The mnemonic is checked to see if it is legal; if not, then ?? is output. The operands must then be input using the same format as the disassembler output, using R to denote registers, etc. The instant assembler will prompt if it can, and control is passed back to the monitor if any character less than ASCII hexadecimal 20 is input. Any other illegal characters simply cause ?? to be output and the user is allowed to try again. If no address is input for a jump address, a displacement of 00 is used. This command updates the monitor variable last opened cell, eg:

```
>7000.
 7000     4CA0     .MOVA␣     ??
 7000     4CA0     .MOVB␣     @ ≠1234(R10),*R15+␣
 7004     7FE0     .LI␣       R1, ≠3<CR>
>7000␣
 7000     DFEA␣               (Check code produced)
 7002     1234␣
 7004     0201␣
 7006     0003 <CR>
>
```

Command O

This command outputs the contents of memory in block form starting at the input address. The default action is to start at address low. It will output 64 locations after which inputting further spaces will output further 32 word


```
>C6L
>O
 00C6  2CE0  011D  2CE0  0108  2CE0
 00D0  011D  06A0  022E  0204  0013  9064  0182  1305
 00E0  0604  18FB  2CE0  0126  10F2  0A14  C124  0196
 00F0  C0C3  0454  7FA0  0080  7FE0  070E  01C6  01E8
 0100  01EE  0200  0210  0222  3E20  544D  5339  3930
 0110  3020  4D4F  4E49  544F  5220  3C3C  000D  0A0A
 0120  2020  2020  3E00  203F  3F00  0A0D  0A20  2020
 0130  2020  0020  5354  4143  4B20  4F56  4552  464C<CR>
>
```

Listing 2: An example of the use of command O, which outputs the contents of memory in block form starting at the input address. 64 locations are output.

blocks. Inputting any other character passes control back to the monitor (See listing 2).

Command Z

This command searches memory sequentially for occurrences of the hexadecimal number input. The search begins at address low. If the hexadecimal number is found, then the address and contents are output. To change the hexadecimal contents input a hexadecimal number followed by a delimiter; otherwise input a delimiter only. If a space is used as a delimiter, then searching continues; other characters pass control back to the monitor. The Z routine will search all of memory (64 K bytes) and if the search is unsuccessful, then NOT FOUND is output.

```
>1234Z
 7000      1234   0␣      (Change Contents)
 7003      1234␣
 7F9A      1234␣
 NOT FOUND
>
```

Command M

This command allows a block of data to be moved from one part of memory to another. The instruction syntax is as follows:

<START ADDRESS> : <FINISH ADDRESS> <DESTINATION ADDRESS>

The software checks that the start address is less than the finish address to stop wrap around. After the three parameters have been input, the question YES or NO? is output. It must be answered with the letter Y if the move is to take place, eg:

```
>MOVE BYTES 100:200␣     TO A000␣
 YES OR NO? Y
>
```

Command ↑Z (Control Z)

This command clears the terminal screen (in our implementation) and branches to the start of the monitor.

Command G

This command causes execution to pass to the input address. The default action is to branch to the start of the monitor.

Command W

This command allows the user workspace to be set. It is intended to set a user program's workspace when the tracing facilities are used. The default action is to output the value of the current user workspace.

```
>1000W
>W
 WP = 1000
>
```

Instruction R

This instruction prints the contents of the registers taking the input address as the workspace printer. The default action uses the user workspace pointer.

```
>1000 R
 WP   =  1000
 R0   =  0400      R1   =  FFFF
 R2   =  8608      R3   =  870A
 R4   =  0000      R5   =  0012
 R6   =  0080      R7   =  865E
 R8   =  0012      R9   =  0002
 R10  =  0002      R11  =  830C
 R12  =  0000      R13  =  0001
 R14  =  4006      R15  =  0001
>
```

Command H

This command allows up to four breakpoints to be set in user programs. If during the T (Trace) or X (execute) instructions a breakpoint instruction is executed, control is passed to the monitor. The registers and environment of the user program may be examined and altered using any of the other monitor commands. The user program may be restarted from the breakpoint with further X, T, or S (Step) commands.

The H command allows the four breakpoints to be set in a user program. On receipt of the H command, BP1 SET AT XXXX is output. Inputting a hexadecimal number will change the address of the breakpointed instruction. Using a space as a delimiter will proceed to the next breakpoint; any other character will pass control to the monitor.

```
>H
 BP 1 SET AT 124C␣
 BP 2 SET AT 124E  1230␣  (Change Breakpoint 2)
 BP 3 SET AT 1260<CR>
>
```

Command T

This command allows a user program to be traced from the start address supplied. The default action is to continue the user program from where it was before. The instruction outputs the addresses of the next 32 instructions that the computer executes, followed by a ?. Inputting further spaces will trace further sets of 32 instructions; any other character passes control to the monitor. If a breakpointed instruction is executed, then control passes back to the monitor. Note: If the traced program outputs to the terminal, these characters will be mixed with the instruction addresses.

Command X

This command is similar to the T command, except that the instruction addresses are not output. Control is passed back to the monitor only if a breakpointed instruction is executed. It is possible to execute the monitor itself using the X command, except of course that the monitor program runs much more slowly.

Command S

This command allows a user program to be single-stepped from the input address. The default action is to continue where it was before. The software outputs a disassembly of the instruction that has just been executed and the contents of the status register.

Monitor Implementation

Firmly believing in structured programming, we felt it necessary to be able to nest subroutine calls to an unlimited depth. This is not possible with the BL (branch and link) instruction since it always puts the return address in R11 of the current workspace. This means that R11 must be saved elsewhere before further branch and link instructions are used.

The BLWP (branch and link workspace pointer) instruction allows unlimited nesting if a predefined workspace is used for each subroutine (gives nonreentrant code). Obtaining a completely new workspace at each subroutine call can be very wasteful of memory space and furthermore, many variables usually have to be accessed via the old workspace. Clearly this results in a situation where R15, R14 and R13 cannot be used (since they contain linkage data), and variables have to be accessed using indirect addressing, yielding too many unnecessary multiple-word instructions.

To overcome these limitations, we used the TMS-9900 XOP (extended operation) instruction to augment the instruction set. XOP0, XOP1, XOP4 and XOP5 have been coded to allow a good clean subroutine linkage that greatly outweighs the added execution time to perform an XOP instruction. All the XOP routines share the same workspace (XOPWP) and, consequently, interrupts are not allowed during their execution. The four routines use a stack that is pointed to by R10 of the XOPWP.

For subroutine calls, XOP0 is used. For example, to jump to subroutine REGISTER, the instruction XOP @ REGISTER, 0 would be used. The return address (address of the next instruction) is placed on the stack and the stack pointer is decremented. The next instruction executed is at address REGISTER. No register of the current workspace or the status is altered.

For subroutine return, XOP1 is used. For example, to return from the subroutine REGISTER, the instruction XOP R0, 1 would be used. R0 is a dummy parameter. The return address is obtained from the top of the stack and the stack pointer is incremented.

XOP4 and XOP5 are used to save and restore the workspace registers, respectively. For example, XOP RX, 4 pushes copies of registers 0 thru RX onto the stack while XOP RX,5 restores registers 0 thru RX by popping them off the stack and putting them in the current workspace. Register 9 of the XOPWP is used as a stack limit, as a precaution against the stack growing too far.

XOP2 and XOP3 are used for input and output to the terminal, respectively. XOP2 inputs a character from the terminal keyboard and places it at the specified location. For example, XOP RX,2 inputs one character and stores it in RX. XOP @ BUFFER,2 inputs one character and stores it at memory address BUFFER. XOP3 is really a print string operation in that it outputs characters from the specified address until a 00 byte has been output (the terminal is assumed to ignore the NUL code), eg:

```
LI      R0,     'AD'
LI      R1,     'D'
XOP     R1,     3       Prints "D"
XOP     R0,     3       Prints "ADD"
```

The monitor itself is quite straightforward, and a brief description should suffice. The first routine is an initialization routine, which clears the interrupt mask and initializes the interrupts and XOP vectors, low, last opened cell, stack pointer and the stack limit.

The next section is the monitor kernel. It recognizes the input command and jumps to the appropriate routine. The instructions command letters are in table INTAB, and the routine addresses are in SUBTAB. There is space for four more commands which we envisage will be used for cassette or disk drivers.

In this section, an assembler command WREN is used. This stands for write enable. The first 2 K bytes of memory is write protected. This prevents programs which are under development from running amok and overwriting themselves. This protection is implemented by external hardware, which simply inhibits the write enable signal to the memory. After a write enable instruction, only the next instruction is able to write into the protected memory.

An Example of the Use of the Tracing Facility in the Authors' Monitor.

1. The program is entered into memory using the instant assembler.
2. The user workspace is assigned.
3. A breakpoint is set on the last instruction.
4. The program is traced from the beginning.
5. Single step (from where trace ended).
6. Registers are examined (only R0 and R1 are in use).
7. The program is single stepped again.
8. The program is executed until the breakpointed instruction has been executed.
9. The final value of R0 is displayed.

```
>A000.
 A000   FFFF   .CLR␣    R0␣
 A002   FFFF   .LI␣     R1, =20␣
 A006   FFFF   .A␣      R1, R0␣
 A008   FFFF   .DEC␣    R1␣
 A00A   FFFF   .JOC␣    =A006␣
 A00C   FFFF   .SRL␣    R0, 1␣
 A00E   FFFF   .AI␣     R0, =200<CR>
>1020W
>H
 BP 1 SET AT 0000 A00E<CR>
>A000T
 A000 =>A002 =>A006 =>A008 =>A00A =>A006 =>A008 =>A00A =>
 A006 =>A008 =>A00A =>A006 =>A008 =>A00A =>A006 =>A008 =>
 A00A =>A006 =>A008 =>A00A =>A006 =>A008 =>A00A =>A006 =>
 A008 =>A00A =>A006 =>A008 =>A00A =>A006 =>A008 =>A00A => ? <CR>
>S
 A006   A001   A  R1, R0          ST = C001
>R
 WP  = 1020
 R0  = 0129     R1  = 0016
 R2  = 9000     R3  = 0200
 R4  = 3000     R5  = 0200
 R6  = 910E     R7  = 9106
 R8  = 000D     R9  = 0006
 R10 = 9000     R11 = 82E2
 R12 = 0000     R13 = 1040
 R14 = 85EA     R15 = C001
>S
 A008   0601   DEC    R1
>X
 BP AT A00E    WP = 1020     ST = C001
>1020␣
 1020    0308 <CR>
```

This simple feature has proved to be very useful and it is a rare occasion indeed when a program is completely destroyed (See figure 2).

Perhaps a brief description of the instant assembler and disassembler would be of interest. The starting point is to understand the op code structure (See table 2). It can be seen that the instruction set can be divided into nine basic types and a subroutine can be written to deal with each type.

Number	Type	+5V	GND
IC1	7474	14	7
IC2	7416	14	7
IC3	7474	14	7
IC4	7474	14	7
IC5	7416	14	7

Table 1: Power wiring table for figures 2 and 3.

Instant Assembler

The operation sequence is as follows

1. Obtain op code mnemonic from terminal. The mnemonic must be terminated with a space and cannot be more than four characters long.
2. The mnemonic is searched for in the mnemonic table to see if it is valid.
3. The mnemonic type and basic op code are found. This causes a branch to an appropriate routine to obtain the correct parameters.
4. Having obtained the parameters, the complete op code plus additional words are calculated.
5. Finally, the completely assembled instruction is written into memory at the address given by last opened cell, and this variable is incremented by 2, 4 or 6, appropriately.

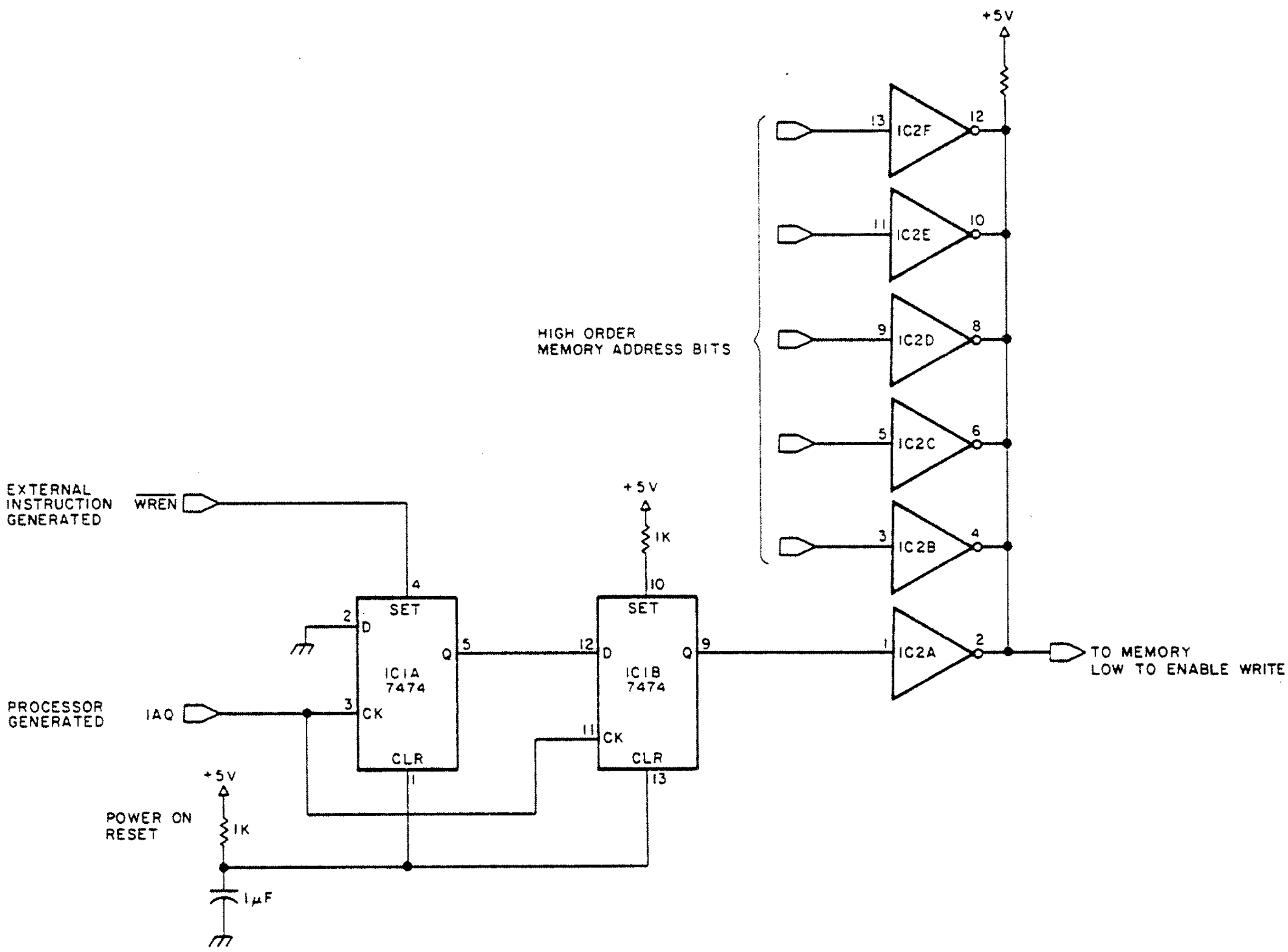

Figure 2: Hardware modification to write-protect the first 2 K bytes of memory in the TMS-9900 system. This prevents programs that are under development from overwriting themselves. After a write enable instruction, only the next instruction is able to write into the protected memory.

Disassembler

The disassembler is organized as a subroutine and is called by the instruction XOP @ DISASM, 0. This will disassemble the instruction pointed to by R12, after which R12 will have been incremented by 2, 4 or 6, appropriately. The sequence of operation is as follows

1. Determine the type of the instruction by systematically checking the op code.
2. Branch to the appropriate instruction type handler. Decode the parameters and output the disassembled instruction to the terminal.
3. Increment R12 appropriately and return.

The TRACE, EXECUTE AND STEP Instructions

These instructions require a small amount of additional hardware to cause an interrupt after the execution of an instruction (See figure 3). The circuitry uses the processor-generated signal IAQ (instruction acquisition), which goes high every time the processor obtains an instruction from memory. Consider the return from interrupt mechanism used by the trace routines:

''
''
TRON (turns trace hardware on)
RTWP (returns to user program).

Note that TRON and TROFF are alternate names for the instructions LREX and CKOF, respectively, and they are used to turn the trace interrupt logic on and off. On returning from an interrupt, the circuitry must generate an interrupt request on the second IAQ pulse, the first pulse being generated by the RTWP instruction and the second by the actual user instruction. The advantage of this mechanism is that the user program does not have to be altered in any way whatsoever for it to be traced.

Op Code	Op Code	Op Code	Instruction Type
0XXX –	00XX : ILL		
	01XX : ILL		
	02XX –	020X : LI	
		022X : AI	
		024X : ANDI	TYPE 6
		026X : ORI	
		028X : CI	
		02AX : STWP	TYPE 7
		02CX : STST	
		02EX : LWPI	TYPE 8
	03XX –	030X : LIMI	
		032X : ILL	
		034X : IDLE	
		036X : RSET	
		038X : RTWP	TYPE 9
		03AX : CKON	
		03CX : CKOF	
		03EX : LREX	
	04XX –	040X : BLWP	
		044X : B	
		048X : X	
		04CX : CLR	
	05XX –	050X : NEG	
		054X : INV	
		058X : INC	
		05CX : INCT	TYPE 3
	06XX –	060X : DEC	
		064X : DECT	
		068X : BL	
		06CX : SWPB	
	07XX –	070X : SETO	
		074X : ABS	
		078X : ILL	
		07CX : ILL	
	08XX : SRA		
	09XX : SRL		TYPE 5
	0AXX : SLA		
	0BXX : SRC		
	0CXX : ILL		
	0DXX : ILL		
	0EXX : ILL		
	0FXX : ILL		

Op Code	Op Code	Instruction Type
1XXX –	10XX : JMP	
	11XX : JLT	
	12XX : JLE	
	13XX : JEQ	
	14XX : JHE	
	15XX : JGT	
	16XX : JNE	
	17XX : JNC	
	18XX : JOC	TYPE 4
	19XX : JNP	
	1AXX : JL	
	1BXX : JH	
	1CXX : JOP	
	1DXX : SBO	
	1EXX : SBZ	
	1FXX : TB	
2XXX –	20XX : COC	
	24XX : CZC	
	28XX : XOR	
	2CXX : XOP	
3XXX –	30XX : LDCR	TYPE 2
	34XX : STCR	
	38CC : MPY	
	3CXX : DIV	
4XXX : SZC		
5XXX : SZCB		
6XXX : S		
7XXX : SB		
8XXX : C		
9XXX : CB		
AXXX : A		TYPE 1
BXXX : AB		
CXXX : MOV		
DXXX : MOVB		
EXXX : SOC		
FXXX : SOCB		

Note: ILL = ILLEGAL OP CODE

Table 2: TMS-9900 op code structure. For the purposes of the authors' instant assembler and disassembler, the instruction set can be divided into nine basic types, and a subroutine can be written to deal with each type.

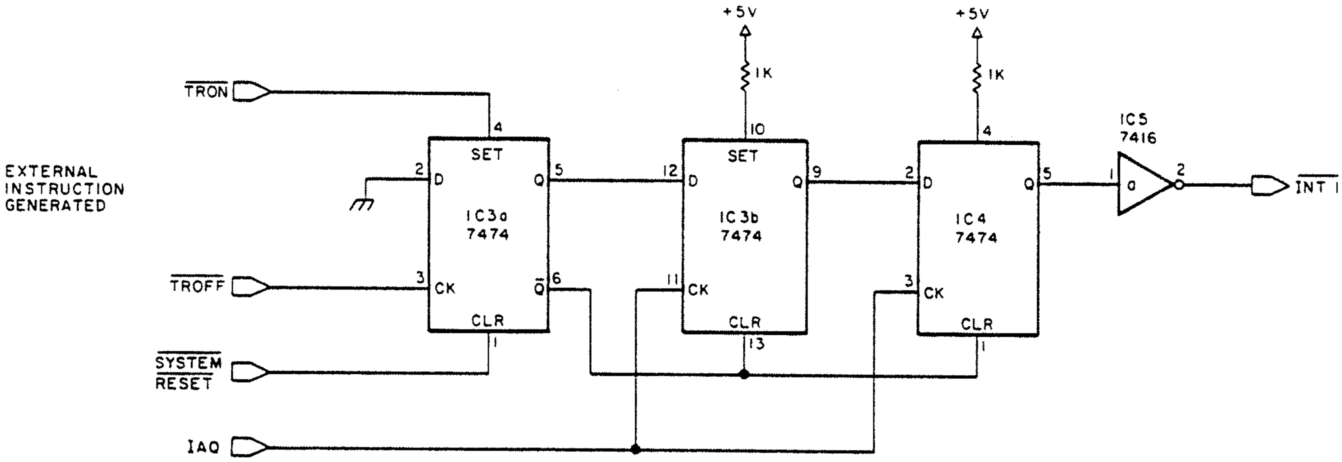

Figure 3: Hardware modification to the TMS-9900 system to enable the use of the authors' Trace, Execute, and Step instructions. The circuitry uses the processor-generated IAQ signal (instruction acquisition) which goes high every time the processor obtains an instruction from memory. TRON and TROFF are alternate names for the LREX and CKOF instructions, respectively. On returning from an interrupt, the circuitry must generate an interrupt request on the second IAQ pulse. With this modification, user programs can be traced without having to be altered.

The interrupt handler uses a separate workspace (INTIWP). Registers 2 thru 5 of this workspace contain the addresses of the breakpointed instructions, and it is the contents of these four registers that are changed by the H instruction. On acknowledgement of a trace interrupt, the appropriate handler, as determined by a pointer contained in R10, is executed. This pointer has been previously set up by the T, X and S routines. If the mode is Trace or Execute, then the breakpoint table is searched, if a match is found, the breakpoint message is output and control is passed back to the monitor. It is important to note that R15, R14 and R13 contain the environment of the traced program. In the step mode (S), a call to the disassembler is made. This displays in mnemonic form the last instruction executed together with the status register, and passes control back to the monitor. The user program may be examined or even changed before resuming execution.

The Move Instruction

The routine for this instruction is not quite as simple as it first seems. Basically, there are two ways of moving data from one part of memory to another to ensure that the new contents do not overwrite the old contents before the old contents have been moved (See figure 4).

Conclusion

This monitor has been in constant use for a period of several months debugging on-going projects (BASIC and Pascal interpreters) and has proved to be an invaluable working tool.■

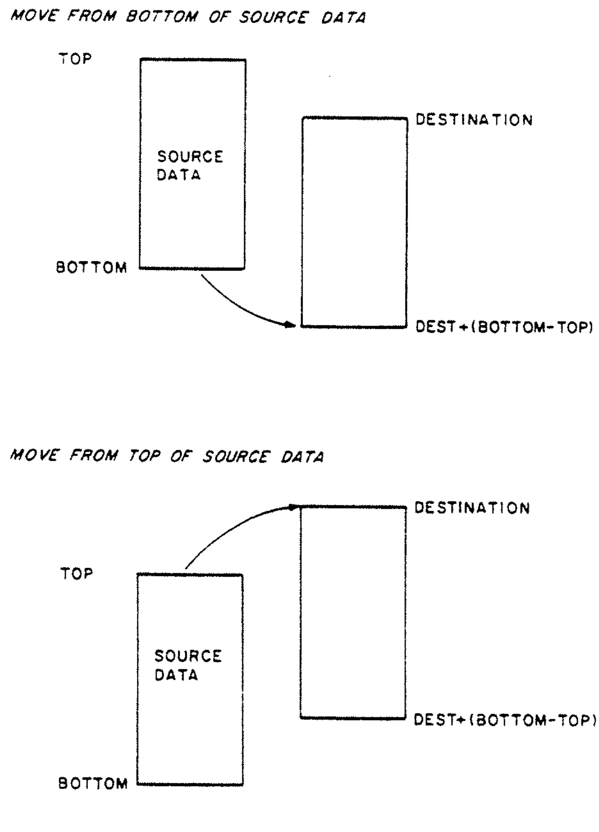

Figure 4: The two cases of the move instruction. Shown are the two ways to move data from one part of the memory to another to ensure that the new contents do not overwrite the old contents before the latter have been moved.

Listing 3: An example of the actual disassembler output.

```
0D40    02E0    LWPI    #E020
0D44    020C    LI      R12, # 0060
0D48    1F01    TB      #0001
0D4A    13FE    JEQ     #0D48
0D4C    1D08    SBO     #0008
0D4E    04C4    CLR     R4
0D50    D064    MOVB    @ # C000 (R4), R1
0D54    9060    CB      @ #0CEA, R1
0D58    141A    JHE     #0D8E
0D5A    06C1    SWPB    R1
0D5C    0241    ANDI    R1, #003F
0D60    C081    MOV     R1, R2
0D62    0A22    SLA     R2, 2
0D64    A042    A       R2, R1
0D66    04C3    CLR     R3
0D68    1F00    TB      #0000
0D6A    16FE    JNE     #0D68
0D6C    D161    MOVB    @ # 0880 (R1), R5
0D70    0583    INC     R3
0D72    0581    INC     R1
0D74    1F04    TB      #0004
0D76    16FE    JNE     # 0D74
0D78    3205    LDCR    R5, 8
0D7A    1D08    SBO     #0008
0D7C    1F04    TB      #0004
0D7E    13FE    JEQ     #0D7C
0D80    0283    CI      R3, #0005
0D84    1AF3    JL      #0D6C
0D86    0584    INC     R4
0D88    8804    C       R4, @ #7FC8
0D8C    1AE1    JL      #0D50
0D8E    1E08    SBZ     #0008
0D90    1D0A    SBO     #000A
0D92    04C2    CLR     R2
0D94    0403    CLR     R3
0D96    1020    SBO     #0009
0D98    0582    INC     R2
0D9A    0583    INC     R3
0D9C    0283    CI      R3, #0200
0DA0    1AFC    JL      #0D9A
0DA2    1E09    SBZ     #0009
0DA4    04C3    CLR     R3
0DA6    0583    INC     R3
0DA8    0283    CI      R3, #0600
0DAC    1AFC    JL      #0DA6
0DAE    0282    CI      R2, #0009
0DB2    1AF0    JL      #0D94
0DB4    1F01    TB      #0001
0DB6    13FE    JEQ     #0DB4
0DB8    1E0A    SBZ     #000A
0DBA    2C40    XOP     R0, 1
```

Listing 4: The Texas Instruments' TMS-9908 monitor

```
*** TMS9900 CROSS ASSEMBLER ***

>> TMS9900 MONITOR V2.0 <<

  ADDR     OP   OP1   OP2            ST.NO.

                                      10              PRINT
                                      20              CREF
                                      30              NOOBJ
                                      40   *****************************************************
                                      50   *                                                   *
                                      60   * TMS9900 MONITOR,INSTANT ASSEMBLER AND DISASSEMBLER *
                                      70   *                                                   *
                                      80   *****************************************************
                                      90   *
                                     100              TITLE=>> TMS9900 MONITOR V2.0 <<
                                     110   *
        *0000*                       120   R0         EQU       0
        *0001*                       130   R1         EQU       1
        *0002*                       140   R2         EQU       2
        *0003*                       150   R3         EQU       3
        *0004*                       160   R4         EQU       4
        *0005*                       170   R5         EQU       5
        *0006*                       180   R6         EQU       6
        *0007*                       190   R7         EQU       7
        *0008*                       200   R8         EQU       8
        *0009*                       210   R9         EQU       9
        *000A*                       220   R10        EQU       10
        *000B*                       230   R11        EQU       11
        *000C*                       240   R12        EQU       12
        *000D*                       250   R13        EQU       13
        *000E*                       260   R14        EQU       14
        *000F*                       270   R15        EQU       15
                                     280   *
        *000D*                       290   CR         EQU       #0D                     CARRIAGE RETURN
        *000A*                       300   LF         EQU       #0A                     LINE FEED
                                     310   *
        *F3E0*   EFED                320   INT1WP     EQU       #F3E0
```

```
      *F3C0*             XOPWP    EQU    #F3C0
      *F3A0*             WORKSP   EQU    #F3A0
                         *
0000                              INIT   #F400
                         *
                         * MONITOR KERNEL
                         *
                         * INITIALISATION
                         *
F400  0360               INITIAL  RSET                            CLEAR INTERRUPT MASK
F402  02E0 F3A0                   LWPI   WORKSP
F406  04C0                        CLR    R0                       SET UP RESET AND INTERRUPT 1 VECTORS
F408  0201 F474                   LI     R1,RESTAB
F40C  03A0               INIT0    WREN                            WRITE ENABLE
F40E  CC31                        MOV    *R1^,*R0^
F410  0280 0008                   CI     R0,8
F414  16FB                        JNE    INIT0
F416  0200 0040                   LI     R0,#40                   SET UP XOP0-XOP5 VECTORS
F41A  0201 F47C                   LI     R1,XOPTAB
F41E  0202 F3C0                   LI     R2,XOPWP
F422  03A0               INIT1    WREN
F424  CC02                        MOV    R2,*R0^
F426  03A0                        WREN
F428  CC31                        MOV    *R1^,*R0^
F42A  0280 0058                   CI     R0,#58
F42E  16F9                        JNE    INIT1
F430  04CF                        CLR    R15                      LOW
F432  04CE                        CLR    R14                      ADDRESS OF LAST OPEN CELL
F434  04E0 F3FE                   CLR    @2*R15+INT1WP            CLEAR "USER PROGRAM" STATUS
F438  0200 EC00                   LI     R0,WORKSP-#800+#60       INITIALISE STACK LIMIT
F43C  C800 F3D2                   MOV    R0,@2*R9+XOPWP
F440  C820 F474 F3D4     INIT2    MOV    @RESTAB,@2*R10+XOPWP     INITIALISE STACK POINTER
                         *
F446  2CE0 F49D          MONITOR  XOP    @MESS01,3                PRINT ">> TMS9900 MONITOR <<"
F44A  2CE0 F488                   XOP    @MESS00,3
F44E  2CE0 F49D          MON00    XOP    @MESS01,3                PRINT PROMPTER
F452  06A0 F5AE                   BL     @HEXIN                   OBTAIN INSTRUCTION
F456  0204 0013                   LI     R4,19                    20 POSSIBLE INSTRUCTIONS
F45A  9064 F502          MON01    CB     @INTAB(R4),R1            SEARCH INTAB
F45E  1305                        JEQ    MON03
F460  0604                        DEC    R4
F462  18FB                        JOC    MON01
F464  2CE0 F4A6          MON02    XOP    @MESS02,3                PRINT " ??"
F468  10F2                        JMP    MON00
F46A  0A14               MON03    SLA    R4,1                     BRANCH TO APPROPRIATE ROUTINE
F46C  C124 F516                   MOV    @SUBTAB(R4),R4
F470  C0C3                        MOV    R3,R3                    TEST HEXIN FLAG
F472  0454                        B      *R4
                         *
                         * TABLES AND MESSAGES
                         *
F474  F3A0 F400 F3E0     RESTAB   WORD   WORKSP,INITIAL,INT1WP,INT1
F47A  FABE
```

```
F47C   F546 F568 F56E      850  XOPTAB  WORD   XOP0,XOP1,XOP2,XOP3,XOP4,XOP5
F482   F580 F590 F5A2
F488   3E20 544D 5339      860  MESS00  TEXT   '> TMS9900 MONITOR <<'-
F48E   3930 3020 4D4F
F494   4E49 544F 5220
F49A   3C3C 00
F49D     0D 0A0A           870  MESS01  BYTE   CR,LF,LF
F4A0   2020 2020 3E00      880          TEXT   '    >'-
F4A6   203F 3F00           890  MESS02  TEXT   ' ??'-
F4AA   0A0D 0A             900  MESS03  BYTE   LF,CR,LF
F4AD     20 2020 2020      910          TEXT   '     '-
F4B2   00
F4B3     20 5354 4143      920  MESS04  TEXT   ' STACK OVERFLOW AT'-
F4B8   4B20 4F56 4552
F4BE   464C 4F57 2041
F4C4   5400
F4C6   40                  930  ATHASH  BYTE   '@'
F4C7     23 00             940  HASH    BYTE   '#',0
F4C9     2C 00             950  COMMA   BYTE   ',',0
F4CB     20 2020 2020      960  R       TEXT   '     '
F4D0   5200                970  RR      BYTE   'R',0
F4D2                       980          EVEN
F4D2   3031 3233 3435      990  HEXTAB  TEXT   '0123456789ABCDEF'
F4D8   3637 3839 4142
F4DE   4344 4546
F4E2   3020 3120 3220     1000  NUMTAB  TEXT   '0 1 2 3 4 5 6 7 8 9 101112131415'
F4E8   3320 3420 3520
F4EE   3620 3720 3820
F4F4   3920 3130 3131
F4FA   3132 3133 3134
F500   3135
F502   1A                 1010  INTAB   BYTE   #1A                     CONTROL Z(CLEARS SCREEN)
F503     2F 2F2F 2F47     1020          TEXT   '////GMZOWRPXLHST/. '
F508   4D5A 4F57 5250
F50E   584C 4853 542F
F514   2E20
F516   F446 0000 0000     1030  SUBTAB  WORD   MONITOR,0,0,0,0,GO,MOVE,FIND,OUTPUT,WP,REGIST,PRINT
F51C   0000 0000 FE76
F522   FDBA FE28 FEB0
F528   FE96 FB16 FB64
F52E   FA4C FE80 FA02     1040          WORD   XCUTE,LOW,SETBP,STEP,TRACE,CHAR,INSTANT,OPEN
F534   FA52 FA40 F638
F53A   F7DE F5F4
                          1050  *
F53E   001F               1060  MASK32  WORD   #001F
F540   000F               1070  MASK16  WORD   #000F
F542   0007               1080  MASK8   WORD   #0007
F544   0003               1090  MASK3   WORD   #0003
                          1100  *
                          1110  * XOP0
                          1120  *
                          1130  * SUBROUTINE CALLING MECHANISM
                          1140  *
```

```
F546    0360             1150  XOP0     RSET
F548    064A             1160           DECT    R10                   DECREMENT STACK POINTER
F54A    824A             1170           C       R10,R9                CHECK FOR STACK OVERFLOW
F54C    1B0A             1180           JH      XOP0AA
F54E    02E0 F3A0        1190  XOP0XXX  LWPI    WORKSP                STACK OVERFLOW HANDLER
F552    C0A0 F3DC        1200           MOV     @2*14+XOPWP,R2
F556    2CE0 F4B3        1210           XOP     @MESS04,3
F55A    06A0 F5D8        1220           BL      @HEXOUTX
F55E    0460 F440        1230           B       @INIT2
F562    C68E             1240  XOP0AA   MOV     R14,*R10              PUSH RETURN ADDRESS ON STACK
F564    C38B             1250           MOV     R11,R14
F566    0380             1260           RTWP
                         1270  *
                         1280  * XOP1
                         1290  *
                         1300  * SUBROUTINE RETURN MECHANISM
                         1310  *
F568    0360             1320  XOP1     RSET
F56A    C3BA             1330           MOV     *R10^,R14
F56C    0380             1340           RTWP
                         1350  *
                         1360  * XOP2
                         1370  *
                         1380  * TERMINAL INPUT EXTENDED OPERATION
                         1390  *
F56E    0360             1400  XOP2     RSET
F570    020C 0040        1410           LI      R12,#40               SET CRU BASE ADDRESS
F574    1F08             1420  XOP2AA   TB      #08                   SPIN ON TERMINAL STATUS
F576    16FE             1430           JNE     XOP2AA
F578    35C0             1440           STCR    R0,7                  PARITY BIT=0 ALWAYS
F57A    1E08             1450           SBZ     #08
F57C    D6C0             1460           MOVB    R0,*R11
F57E    0380             1470           RTWP
                         1480  *
                         1490  * XOP3
                         1500  *
                         1510  * TERMINAL OUTPUT EXTENDED OPERATION
                         1520  *
F580    0360             1530  XOP3     RSET
F582    04CC             1540           CLR     R12                   SET CRU BASE ADDRESS
F584    1F08             1550  XOP3AA   TB      #08                   SPIN ON TERMINAL STATUS
F586    16FE             1560           JNE     XOP3AA
F588    323B             1570           LDCR    *R11^,8
F58A    1E08             1580           SBZ     #08
F58C    16FB             1590           JNE     XOP3AA                CONTINUE UNTIL #00 HAS BEEN OUTPUTED
F58E    0380             1600           RTWP
                         1610  *
                         1620  * XOP4
                         1630  *
                         1640  * PUSHES R0-RX ONTO STACK
                         1650  *
F590    0360             1660  XOP4     RSET
F592    C04D             1670           MOV     R13,R1
```

```
F594  064A        XOP4AA  DECT    R10
F596  824A                C       R10,R9            CHECK FOR STACK OVERFLOW
F598  12DA                JLE     XOP0XXX
F59A  C6B1                MOV     *R1^,*R10
F59C  82C1                C       R1,R11
F59E  12FA                JLE     XOP4AA
F5A0  0380                RTWP
                  *
                  * XOP5
                  *
                  * POPS RX-R0 OFF STACK
                  *
F5A2  0360        XOP5    RSET
F5A4  C6FA        XOP5AA  MOV     *R10^,*R11
F5A6  064B                DECT    R11
F5A8  834B                C       R11,R13
F5AA  14FC                JHE     XOP5AA
F5AC  0380                RTWP
                  *
                  * SUBROUTINE HEXIN                  INPUTS A HEX NO. INTO R2
                  * USES R1,R2,R3,R4
                  *
F5AE  04C1        HEXIN   CLR     R1
F5B0  04C2                CLR     R2
F5B2  0703                SETO    R3                SET FLAG
F5B4  2C81        HEXIN0  XOP     R1,2              GET CHARACTER
F5B6  2CC1                XOP     R1,3              PRINT CHARACTER
F5B8  0204 000F           LI      R4,15
F5BC  9064 F4D2   HEXIN1  CB      @HEXTAB(R4),R1
F5C0  1604                JNE     HEXIN2
F5C2  04C3                CLR     R3                CLEARS FLAG
F5C4  0A42                SLA     R2,4              SHIFTS HEX DIGIT INTO R2
F5C6  E084                SOC     R4,R2
F5C8  10F5                JMP     HEXIN0
F5CA  0604        HEXIN2  DEC     R4
F5CC  18F7                JOC     HEXIN1
F5CE  C0C3                MOV     R3,R3             TEST HEXIN FLAG
F5D0  045B                RT
                  *
                  * SUBROUTINE HEXOUT                 OUTPUTS CONTENTS OF R2
                  * USES R0,R1,R2,R3
                  *
F5D2  2CE0 F4B1   HEXOUT  XOP     @MESS03+7,3       PRINTS " "
F5D6  2D03        HEXOUT0 XOP     R3,4              SAVE R0-R3
F5D8  04C0        HEXOUTX CLR     R0
F5DA  0203 0004           LI      R3,#04
F5DE  0BC2        HEXOUT1 SRC     R2,12
F5E0  C042                MOV     R2,R1
F5E2  0241 000F           ANDI    R1,#000F
F5E6  D021 F4D2           MOVB    @HEXTAB(R1),R0
F5EA  2CC0                XOP     R0,3
F5EC  0603                DEC     R3
F5EE  16F7                JNE     HEXOUT1
```

```
F5F0  2D43              XOP   R3,5           RESTORE R3-R0
F5F2  045B              RT
                  *
                  * INSTRUCTION ' '
                  *
F5F4  1601        OPEN    JNE   OPEN00       JNE ON HEXIN FLAG
F5F6  C382                MOV   R2,R14
F5F8  024E FFFE   OPEN00  ANDI  R14,#FFFE    MAKE ADDRESS EVEN
F5FC  2C20 F622   OPEN01  XOP   @PADDRC,0    PRINT ADDRESS AND CONTENTS
F600  06A0 F5AE           BL    @HEXIN
F604  1602                JNE   OPEN02       JNE ON HEXIN FLAG
F606  03A0                WREN
F608  C782                MOV   R2,*R14
F60A  0281 2000   OPEN02  CI    R1,' '       LAST CHARACTER=" "?
F60E  1602                JNE   OPEN03
F610  05CE                INCT  R14
F612  10F4                JMP   OPEN01
F614  0281 2D00   OPEN03  CI    R1,'-'       LAST CHARACTER="-"?
F618  1602                JNE   MON00A
F61A  064E                DECT  R14
F61C  10EF                JMP   OPEN01
F61E  0460 F44E   MON00A  B     @MON00
                  *
                  *
F622  2CE0 F4AB   PADDRC  XOP   @MESS03+1,3  PRINT INDENTATION
F626  C08E                MOV   R14,R2       PRINT ADDRESS
F628  06A0 F5D6           BL    @HEXOUT0
F62C  C09E                MOV   *R14,R2      PRINT CONTENTS OF ADDRESS
F62E  06A0 F5D2           BL    @HEXOUT
F632  2CE0 F4B0           XOP   @MESS03+6,3  PRINT "  "
F636  2C40                XOP   R0,1         RETURN
                  *
                  * INSTRUCTION /
                  *
F638  1601        CHAR    JNE   CHAR00       JNE ON HEXIN FLAG
F63A  C382                MOV   R2,R14
F63C  0203 2F00   CHAR00  LI    R3,'/'
F640  C10E        CHAR01  MOV   R14,R4       SET START ADDRESS
F642  2C20 F622           XOP   @PADDRC,0    PRINT ADDRESS AND CONTENTS
F646  2CC3                XOP   R3,3         PRINT "/"
F648  2C81        CHAR02  XOP   R1,2         GET CHARACTER
F64A  0281 0800           CI    R1,#0800     BACKSPACE?
F64E  1605                JNE   CHAR03
F650  8384                C     R4,R14
F652  13FA                JEQ   CHAR02
F654  2CC1                XOP   R1,3         PRINT BACKSPACE
F656  060E                DEC   R14
F658  10F7                JMP   CHAR02
F65A  C041        CHAR03  MOV   R1,R1        IS CHARACTER CONTROL SHIFT @
F65C  1303                JEQ   CHAR04
F65E  0281 2000           CI    R1,' '
F662  1ADD                JL    MON00A       RETURN TO MONITOR
F664  03A0        CHAR04  WREN
```

```
F666   DF81             2740          MOVB   R1,*R14^
F668   1602             2750          JNE    CHAR05
F66A   0201 5F00        2760          LI     R1,'_'              UNDERSCORE
F66E   2CC1             2770  CHAR05  XOP    R1,3                PRINT CHARACTER
F670   27A0 F53E        2780          CZC    @MASK32,R14
F674   16E9             2790          JNE    CHAR02
F676   2CC3             2800          XOP    R3,3                PRINT "/"
F678   10E3             2810          JMP    CHAR01
                        2820  *
                        2830  *
                        2840  * INSTANT ASSEMBLER
                        2850  *
                        2860  * TABLES
                        2870  *
F67A   535A 4320 535A   2880  TYPE1   TEXT   'SZC SZCBS  SB  C   CB  A   AB  '
F680   4342 5320 2020
F686   5342 2020 4320
F68C   2020 4342 2020
F692   4120 2020 4142
F698   2020
F69A   4D4F 5620 4D4F   2890          TEXT   'MOV MOVBSOC SOCB'
F6A0   5642 534F 4320
F6A6   534F 4342
F6AA   434F 4320 435A   2900  TYPE2   TEXT   'COC CZC XOR XOP LDCRSTCRMPY DIV '
F6B0   4320 584F 5220
F6B6   584F 5020 4C44
F6BC   4352 5354 4352
F6C2   4D50 5920 4449
F6C8   5620
F6CA   424C 5750 4220   2910  TYPE3   TEXT   'BLWPB   X   CLR NEG INV INC INCT'
F6D0   2020 5820 2020
F6D6   434C 5220 4E45
F6DC   4720 494E 5620
F6E2   494E 4320 494E
F6E8   4354
F6EA   4445 4320 4445   2920          TEXT   'DEC DECTBL  SWPBSETOABS ILL?ILL?'
F6F0   4354 424C 2020
F6F6   5357 5042 5345
F6FC   544F 4142 5320
F702   494C 4C3F 494C
F708   4C3F
F70A   4A4D 5020 4A4C   2930  TYPE4   TEXT   'JMP JLT JLE JEQ JHE JGE JNE JNC '
F710   5420 4A4C 4520
F716   4A45 5120 4A48
F71C   4520 4A47 4520
F722   4A4E 4520 4A4E
F728   4320
F72A   4A4F 4320 4A4E   2940          TEXT   'JOC JNO JL  JH  JOP SBO SBZ TB  '
F730   4F20 4A4C 2020
F736   4A48 2020 4A4F
F73C   5020 5342 4F20
F742   5342 5A20 5442
F748   2020
```

```
F74A  5352 4120 5352   TYPE5    TEXT   'SRA SRL SLA SRC '
F750  4C20 534C 4120
F756  5352 4320
F75A  4C49 2020 4149   TYPE6    TEXT   'LI  AI  ANDIORI CI  '
F760  2020 414E 4449
F766  4F52 4920 4349
F76C  2020
F76E  5354 5750 5354   TYPE7    TEXT   'STWPSTST'
F774  5354
F776  4C57 5049 4C49   TYPE8    TEXT   'LWPILIMI'
F77C  4D49
F77E  494C 4C3F 4944   TYPE9    TEXT   'ILL?IDLERSETRTWPWRENCKOFLREX'
F784  4C45 5253 4554
F78A  5254 5750 5752
F790  454E 434B 4F46
F796  4C52 4558
F79A  F67A F6AA F6CA   TYTAB    WORD   TYPE1,TYPE2,TYPE3,TYPE4,TYPE5
F7A0  F70A F74A
F7A4  F75A F76E F776            WORD   TYPE6,TYPE7,TYPE8,TYPE9,TYTAB
F7AA  F77E F79A
F7AE  0A08 0406 0603   TYSHIFT  BYTE   10,8,4,6,6,3,3,3,3,0
F7B4  0303 0300
F7B8  4000 2000 0400   TYBASE   WORD   #4000,#2000,#0400,#1000,#0800,#0200,#02A0,#02E0,#0320
F7BE  1000 0800 0200
F7C4  02A0 02E0 0320
F7CA  F898 F8AC F8CC   TYSUB    WORD   T1,T2,T3,T4,T5,T6,T7,T8,T9
F7D0  F8D4 F904 F91A
F7D6  F92A F934 F860
F7DC  2E00             DOT      WORD   '.'
                       *
                       * GET MNEMONIC AND PLACE IN R8,R9
                       *
F7DE  1601             INSTANT  JNE    INSTOO              JNE ON HEXIN FLAG
F7E0  C382                      MOV    R2,R14
F7E2  02A0             INSTOO   STWP   R0
F7E4  0220 0010                 AI     R0,2*8              R0->R8
F7E8  0208 2020                 LI     R8,'  '             INITIALISE R8,R9
F7EC  C248                      MOV    R8,R9
F7EE  2C20 F622                 XOP    @PADDRC,0           PRINT ADDRESS AND CONTENTS
F7F2  2CE0 F7DC                 XOP    @DOT,3              PRINT "."
F7F6  04C2                      CLR    R2                  CHARACTER COUNT=0
F7F8  2C81             INST01   XOP    R1,2                GET CHARACTER
F7FA  2CC1                      XOP    R1,3                PRINT CHARACTER
F7FC  0281 4100                 CI     R1,'A'              IN ['A'..'Z'] ?
F800  1A08                      JL     INST02
F802  0281 5A00                 CI     R1,'Z'
F806  1805                      JH     INST02
F808  DC01                      MOVB   R1,*R0^
F80A  0582                      INC    R2
F80C  0282 0005                 CI     R2,#5               GET AT MOST 5 CHARACTERS
F810  16F3                      JNE    INST01
F812  2CE2 F4AD        INST02   XOP    @MESS03+3(R2),3     PRINT REMAINING SPACES
F816  0281 2000                 CI     R1,' '              CHECK LAST CHARACTER
```

```
F81A  1B3B                        JH    ERROR
F81C  1A2E                        JL    INST10              RETURN TO MONITOR
                          *
                          * SEARCH FOR MNEMONIC IN TABLE
                          *
F81E  0207 F67A                   LI    R7,TYPE1
F822  8217                INST03  C     *R7,R8
F824  1603                        JNE   INST04
F826  8267 0002                   C     @#0002(R7),R9
F82A  1306                        JEQ   INST05
F82C  0227 0004           INST04  AI    R7,4
F830  0287 F79A                   CI    R7,TYTAB
F834  16F6                        JNE   INST03
F836  102D                        JMP   ERROR
                          *
                          * FIND MNEMONIC TYPE AND CALCULATE BASIC OPCODE
                          *
F838  04C4                INST05  CLR   R4                  R4:=OPTYPE
F83A  0203 F79C                   LI    R3,TYTAB+2
F83E  8CC7                INST06  C     R7,*R3^
F840  1A02                        JL    INST07
F842  0584                        INC   R4
F844  10FC                        JMP   INST06
F846  61E3 FFFC           INST07  S     @-4(R3),R7
F84A  D024 F7AE                   MOVB  @TYSHIFT(R4),R0
F84E  0980                        SRL   R0,8
F850  0A07                        SLA   R7,0                SHIFTS BY VALUE IN R0
F852  0A14                        SLA   R4,1
F854  A1E4 F7B8                   A     @TYBASE(R4),R7      R7:=BASIC OPCODE
F858  04CA                        CLR   R10                 NO OF WORDS IN INSTRUCTION
F85A  C024 F7CA                   MOV   @TYSUB(R4),R0
F85E  0450                        B     *R0
                          *
F860  2C81                T9      XOP   R1,2                GET CHARACTER
F862  2CC1                INST08  XOP   R1,3                PRINT LAST CHARACTER
F864  02A0                        STWP  R0
F866  0220 000E                   AI    R0,2*7              R0->R7
F86A  03A0                INST09  WREN
F86C  CFB0                        MOV   *R0^,*R14^
F86E  060A                        DEC   R10
F870  18FC                        JOC   INST09
F872  0281 2000                   CI    R1,' '
F876  1B0D                        JH    ERROR
F878  13B4                        JEQ   INST00
F87A  0460 F44E           INST10  B     @MONOO              RETURN TO MONITOR
                          *
F87E  0281 2C00           DELIM   CI    R1,','              CHECK CH IN [' ',',']
F882  1304                        JEQ   DELIMOO
F884  0281 2000                   CI    R1,' '
F888  1B04                        JH    ERROR
F88A  1AF7                        JL    INST10
F88C  2CE0 F4C9           DELIMOO XOP   @COMMA,3            PRINT ","
F890  045B                        RT
```

```
                          *
F892  2CE0 F4A6           ERROR  XOP   @MESS02,3      PRINT " ??"
F896  10A5                       JMP   INST00
                          *
F898  06A0 F97C           T1     BL    @SIX           INSTRUCTION OF TYPE1
F89C  06A0 F87E                  BL    @DELIM
F8A0  A1C0                       A     R0,R7
F8A2  06A0 F97C                  BL    @SIX
F8A6  0A60                T100   SLA   R0,6
F8A8  A1C0                T101   A     R0,R7
F8AA  10DB                       JMP   INST08
                          *
F8AC  06A0 F97C           T2     BL    @SIX           INSTRUCTION OF TYPE2
F8B0  06A0 F87E                  BL    @DELIM
F8B4  A1C0                       A     R0,R7
F8B6  0287 2C00                  CI    R7,#2C00       CHECK IF XOP,LDCR OR STCR
F8BA  1A03                       JL    T200
F8BC  0287 3400                  CI    R7,#3400
F8C0  1A02                       JL    T201
F8C2  2CE0 F4D0           T200   XOP   @RR,3          PRINT "R"
F8C6  06A0 F942           T201   BL    @GETRC
F8CA  10ED                       JMP   T100
                          *
F8CC  06A0 F97C           T3     BL    @SIX           INSTRUCTION OF TYPE3
F8D0  A1C0                       A     R0,R7
F8D2  10C7                       JMP   INST08
                          *
F8D4  2CE0 F4C7           T4     XOP   @HASH,3        INSTRUCTION OF TYPE4
F8D8  06A0 F5AE                  BL    @HEXIN
F8DC  16C2                       JNE   INST08         DEFAULT FOR JUMP INSTRUCTIONS
F8DE  0287 1D00                  CI    R7,#1D00       IS INSTRUCTION SBO,SBZ,TB
F8E2  140B                       JHE   T400
F8E4  608E                       S     R14,R2         CALCULATE JUMP OFFSET
F8E6  0642                       DECT  R2
F8E8  0812                       SRA   R2,1
F8EA  0282 FF80                  CI    R2,-128
F8EE  1178                       JLT   ERR1
F8F0  0282 007F                  CI    R2,127
F8F4  1575                       JGT   ERR1
F8F6  0242 00FF                  ANDI  R2,#00FF
F8FA  0282 00FF           T400   CI    R2,#FF
F8FE  1B70                       JH    ERR1
F900  A1C2                       A     R2,R7
F902  10AF                       JMP   INST08
                          *
F904  2CE0 F4D0           T5     XOP   @RR,3          INSTRUCTION OF TYPE5
F908  06A0 F942                  BL    @GETRC
F90C  06A0 F87E                  BL    @DELIM
F910  A1C0                       A     R0,R7
F912  06A0 F942                  BL    @GETRC
F916  0A40                       SLA   R0,4
F918  10C7                       JMP   T101
                          *
```

```
F91A  2CE0 F4D0   T6       XOP   @RR,3             INSTRUCTION OF TYPE 6
F91E  06A0 F942            BL    @GETRC
F922  A1C0                 A     R0,R7
F924  06A0 F87E            BL    @DELIM
F928  1005                 JMP   T8
                  *
F92A  2CE0 F4D0   T7       XOP   @RR,3             INSTRUCTION OF TYPE7
F92E  06A0 F942            BL    @GETRC
F932  10BA                 JMP   T101
                  *
F934  2CE0 F4C7   T8       XOP   @HASH,3           INSTRUCTION OF TYPE8
F938  06A0 F5AE            BL    @HEXIN
F93C  C202                 MOV   R2,R8
F93E  058A                 INC   R10
F940  1090                 JMP   INST08
                  *
                  * GETRC                          OBTAINS REGISTER NO OR CONSTANT NUMBER
                  *
F942  02A6        GETRC    STWP  R6
F944  0226 000A            AI    R6,2*5            R6->R5
F948  0205 2020            LI    R5,'  '
F94C  0202 0003            LI    R2,3              AT MOST 3 CHARACTERS
F950  2C81        GETRC00  XOP   R1,2              GET CHARACTER
F952  0281 3000            CI    R1,'0'            IN ['0'..'9'] ?
F956  1A07                 JL    GETRC01
F958  0281 3900            CI    R1,'9'
F95C  1B04                 JH    GETRC01
F95E  2CC1                 XOP   R1,3              PRINT CHARACTER
F960  DD81                 MOVB  R1,*R6^
F962  0602                 DEC   R2
F964  16F5                 JNE   GETRC00
F966  0202 001E   GETRC01  LI    R2,2*15
F96A  8162 F4E2   GETRC02  C     @NUMTAB(R2),R5    TEST IF VALID NUMBER
F96E  1303                 JEQ   GETRC03
F970  0642                 DECT  R2
F972  18FB                 JOC   GETRC02
F974  1035                 JMP   ERR1
F976  0912        GETRC03  SRL   R2,1
F978  C002                 MOV   R2,R0             R0:=HEX EQUIVALENT OF DECIMAL NUMBER
F97A  045B                 RT
                  *
                  * SIX                            OBTAINS SIXBIT SOURCE/DESTINATION FIELD
                  *
F97C  C30B        SIX      MOV   R11,R12           SAVE RETURN ADDRESS
F97E  2C81                 XOP   R1,2              GET CHARACTER
F980  2CC1                 XOP   R1,3              PRINT CHARACTER
F982  0281 5200            CI    R1,'R'            REGISTER ADDRESSING MODE
F986  1603                 JNE   SIX00
F988  06A0 F942            BL    @GETRC
F98C  045C                 B     *R12              RETURN
F98E  0281 2A00   SIX00    CI    R1,'*'            INDIRECT ADDRESSING MODE
F992  160E                 JNE   SIX03
F994  2CE0 F4D0            XOP   @RR,3             PRINT "R"
```

```
F998   06A0 F942            BL     @GETRC
F99C   0220 0010            AI     R0,#10          AMODE:=1
F9A0   0281 2800            CI     R1,'+'          AUTO INCREMENT REGISTER MODE
F9A4   1604                 JNE    SIX02
F9A6   2CC1                 XOP    R1,3            PRINT "+"
F9A8   0220 0020            AI     R0,#20          AMODE:=3
F9AC   2C81                 XOP    R1,2            GET CHARACTER
F9AE   045C         SIX02   B      *R12            RETURN
F9B0   0281 4000    SIX03   CI     R1,'@'          SYMBOLIC ADDRESSING MODE
F9B4   1615                 JNE    ERR1
F9B6   2CE0 F4C7            XOP    @HASH,3         PRINT "#"
F9BA   06A0 F5AE            BL     @HEXIN
F9BE   C28A                 MOV    R10,R10         TEST R10
F9C0   1602                 JNE    SIX04
F9C2   C202                 MOV    R2,R8
F9C4   1001                 JMP    SIX05
F9C6   C242         SIX04   MOV    R2,R9
F9C8   058A         SIX05   INC    R10
F9CA   04C0                 CLR    R0
F9CC   0281 2800            CI     R1,'('          INDEXED ADDRESSING MODE
F9D0   1608                 JNE    SIX07
F9D2   2CE0 F4D0            XOP    @RR,3           PRINT "R"
F9D6   06A0 F942            BL     @GETRC
F9DA   0281 2900            CI     R1,')'
F9DE   1302                 JEQ    SIX06
F9E0   0460 F892    ERR1    B      @ERROR
F9E4   2CC1         SIX06   XOP    R1,3            PRINT ")"
F9E6   2C81                 XOP    R1,2            GET CHARACTER
F9E8   0220 0020    SIX07   AI     R0,#20          AMODE:=2
F9EC   045C                 B      *R12
                    *
                    *
                    * INSTRUCTION H
                    *
F9EE   000A 0A      SMESS0  BYTE   CR,LF,LF
F9F1     20 2020 2020       TEXT   '     BP '-
F9F6   4250 2000
F9FA   2053 4554 2041
                    SMESS1  TEXT   ' SET AT'-
FA00   5400
FA02                        EVEN
FA02   0205 F3E0    SETBP   LI     R5,INT1WP
FA06   0225 0008            AI     R5,2*4          R5->BP 1 (R4,R5,R6,R7)
FA0A   0208 3100            LI     R8,'1'
FA0E   2CE0 F9EE    SETBP00 XOP    @SMESS0,3       PRINT "     BP "
FA12   2CC8                 XOP    R8,3            PRINT BP NO.
FA14   2CE0 F9FA            XOP    @SMESS1,3       PRINT "SET AT"
FA18   C0B5                 MOV    *R5^,R2         PRINT CONTENTS
FA1A   06A0 F5D2            BL     @HEXOUT
FA1E   2CE0 F4B0            XOP    @MESS03+6,3     PRINT "  "
FA22   06A0 F5AE            BL     @HEXIN          GET NEW CONTENTS(IF REQUIRED)
FA26   1602                 JNE    SETBP01         JNE ON HEXIN FLAG
FA28   C942 FFFE            MOV    R2,@-2(R5)
FA2C   0281 2000    SETBP01 CI     R1,' '
```

```
FA30  1605                         JNE    SETBP02
FA32  0228 0100                    AI     R8,#100
FA36  0288 3500                    CI     R8,'5'
FA3A  16E9                         JNE    SETBP00
FA3C  0460 F44E            SETBP02 B      @MON00
                           *
                           * INSTRUCTION T
                           *
FA40  0204 0020            TRACE   LI     R4,#20
FA44  C804 F3F2                    MOV    R4,@2*9+INT1WP
FA48  04C4                         CLR    R4
FA4A  1005                         JMP    SETUP
                           *
                           * INSTRUCTION X
                           *
FA4C  0204 0002            XCUTE   LI     R4,2
FA50  1002                         JMP    SETUP
                           *
                           * INSTRUCTION S
                           *
FA52  0204 0004            STEP    LI     R4,4
                           *
FA56  0360                 SETUP   RSET                             DISABLE INTERRUPTS
FA58  C804 F3F4                    MOV    R4,@2*10+INT1WP
FA5C  C0C3                         MOV    R3,R3                     TEST HEXIN FLAG
FA5E  1602                         JNE    SETUP00
FA60  C802 F3FC                    MOV    R2,@2*14+INT1WP           SET UP NEW START ADDRESS
FA64  02E0 F3E0            SETUP00 LWPI   INT1WP                    CHANGE WORKSPACE
FA68  C30E                         MOV    R14,R12                   ADDRESS OF NEXT INSTRUCTION
FA6A  27E0 F540                    CZC    @MASK16,R15
FA6E  1601                         JNE    SETUP01
FA70  058F                         INC    R15
FA72  03C0                 SETUP01 TRON                             TURN TRACE LOGIC ON
FA74  0380                         RTWP                             RETURN TO PROGRAM BEING TRACED
                           *
                           * INTERRUPT 1 HANDLER
                           *
FA76  FA96 FAC0 FB02       I1TAB   WORD   INT1AA,INT1CC,INT1GG      TRACE,XCUTE,STEP
FA7C  3D3E 00              IMESS0  TEXT   '=>'-
FA7F    20 2057 503D       IMESS1  TEXT   '  WP='-
FA84  00
FA85    18 2C              IMESS2  BYTE   #18,#2C                   CONTROL CHARACTERS TO POSITION CURSER
FA87    20 2053 543D       IMESS3  TEXT   '  ST='-
FA8C  00
FA8D    00                         EVEN
FA8E  03E0                 INT1    TROFF                            RESET TRACE LOGIC
FA90  C02A FA76                    MOV    @I1TAB(R10),R0            JUMP TO APPROPRIATE HANDLER
FA94  0450                         B      *R0
                           *
FA96  2660 F542            INT1AA  CZC    @MASK8,R9
FA9A  1602                         JNE    INT1BB
FA9C  2CE0 F4AB                    XOP    @MESS03+1,3               PRINT INDENTATION
FAA0  C08C                 INT1BB  MOV    R12,R2                    PRINT ADDRESS OF LAST INST.
```

```
FAA2  06A0 F5D6             BL    @HEXOUTO
FAA6  2CE0 FA7C             XOP   @IMESSO,3          PRINT "=>"
FAAA  0609                  DEC   R9
FAAC  1609                  JNE   INT1CC
FAAE  04C1                  CLR   R1
FAB0  2CE0 F4A8             XOP   @MESS02+2,3        PRINT "?"
FAB4  2C81                  XOP   R1,2               GET CHARACTER
FAB6  0281 2000             CI    R1,' '
FABA  161F                  JNE   INT1FF
FABC  0209 0020             LI    R9,#20
                  *
                  * SEARCH BREAKPOINT TABLE
                  *
FAC0  0202 0004   INT1CC    LI    R2,4
FAC4  02A0                  STWP  R0
FAC6  0220 0008             AI    R0,2*4             R0->BP 1
FACA  8330        INT1DD    C     *R0^,R12           IS INSTRUCTION BREAKPOINTED?
FACC  1305                  JEQ   INT1EE
FACE  0602                  DEC   R2
FAD0  16FC                  JNE   INT1DD
FAD2  C30E                  MOV   R14,R12            R12:=ADDRESS OF NEXT INSTRUCTION
FAD4  03C0                  TRON                     TURN TRACE LOGIC ON
FAD6  0380                  RTWP                     RETURN TO USER PROGRAM
                  *
FAD8  2CE0 F9EE   INT1EE    XOP   @SMESS0,3          PRINT INDENTATION+"   BP"
FADC  2CE0 F9FF             XOP   @SMESS1+5,3        PRINT "AT"
FAE0  C08C                  MOV   R12,R2
FAE2  06A0 F5D2             BL    @HEXOUT
FAE6  2CE0 FA7F             XOP   @IMESS1,3          PRINT "WP="
FAEA  C08D                  MOV   R13,R2             PRINT USER WORKSPACE
FAEC  06A0 F5D6             BL    @HEXOUTO
FAF0  2CE0 FA87             XOP   @IMESS3,3          PRINT "ST="
FAF4  C08F                  MOV   R15,R2             PRINT USER STATUS
FAF6  06A0 F5D6             BL    @HEXOUTO
FAFA  02E0 F3A0   INT1FF    LWPI  WORKSP
FAFE  0460 F44E             B     @MON00
                  *
FB02  2D0B        INT1GG    XOP   R11,4              SAVE R0-R11
FB04  2C20 FBDA             XOP   @DISASM,0
FB08  2D4B                  XOP   R11,5              RESTORE R0-R11
FB0A  2CE0 FA85             XOP   @IMESS2,3          PRINT "ST="
FB0E  C08F                  MOV   R15,R2             PRINT STATUS
FB10  06A0 F5D6             BL    @HEXOUTO
FB14  10F2                  JMP   INT1FF             RETURN TO MONITOR
                  *
                  * INSTRUCTION R
                  *
FB16  0204 F3FA   REGIST    LI    R4,2*13+INT1WP
FB1A  C114                  MOV   *R4,R4             R4:=USER WORKSPACE BY DEFAULT
FB1C  C0C3                  MOV   R3,R3              TEST HEXIN FLAG
FB1E  1601                  JNE   REGISTO
FB20  C102                  MOV   R2,R4
FB22  0244 FFFE   REGISTO   ANDI  R4,#FFFE           MAKE ADDRESS EVEN
```

```
FB26  2CE0 F4AA            XOP   @MESS03,3             PRINT INDENTATION
FB2A  2CE0 FA81            XOP   @IMESS1+2,3           PRINT "WP="
FB2E  C084                 MOV   R4,R2                 PRINT WORKSPACE ADDRESS
FB30  06A0 F5D6            BL    @HEXOUTO
FB34  04C5                 CLR   R5
FB36  04CA                 CLR   R10
FB38  0209 0D0A   REGIST1  LI    R9,#0D0A
FB3C  2CC9                 XOP   R9,3                  PRINT CR,LF
FB3E  2CE0 F4C8   REGIST2  XOP   @R,3                  PRINT "R"
FB42  C265 F4E2            MOV   @NUMTAB(R5),R9
FB46  2CC9                 XOP   R9,3                  PRINT REGISTER NO
FB48  0209 203D            LI    R9,' ='
FB4C  2CC9                 XOP   R9,3
FB4E  C0B4                 MOV   *R4^,R2               PRINT REGISTER CONTENTS
FB50  06A0 F5D2            BL    @HEXOUT
FB54  05C5                 INCT  R5
FB56  0285 0020            CI    R5,#20
FB5A  1315                 JEQ   MON00B
FB5C  2560 F544            CZC   @MASK3,R5
FB60  13EB                 JEQ   REGIST1
FB62  10ED                 JMP   REGIST2
                  *
                  * INSTRUCTION P
                  *
FB64  C30F        PRINT    MOV   R15,R12               DEFAULT ACTION
FB66  C0C3                 MOV   R3,R3                 TEST HEXIN FLAG
FB68  1601                 JNE   PRINT00
FB6A  C302                 MOV   R2,R12
FB6C  024C FFFE   PRINT00  ANDI  R12,#FFFE
FB70  020D 0010            LI    R13,16                INSTRUCTION COUNT
FB74  2C20 FBDA   PRINT01  XOP   @DISASM,0             CALL DISASSEMBLER
FB78  060D                 DEC   R13                   DECREMENT INSTRUCTION COUNT
FB7A  16FC                 JNE   PRINT01
FB7C  058D                 INC   R13
FB7E  2C81                 XOP   R1,2                  GET CHARACTER
FB80  0281 2000            CI    R1,' '                MORE INSTRUCTIONS TO BE DISASSEMBLED ?
FB84  13F7                 JEQ   PRINT01
FB86  0460 F44E   MON00B   B     @MON00
                  *
                  * DISASSEMBLER
                  *
                  * TABLES
                  *
FB8A  FC70 FC70 FC7A  C0ATAB  WORD  CODE00,CODE00,CODE02,CODE02,CODE04,CODE04,CODE04,CODE04
FB90  FC7A FC94 FC94
FB96  FC94 FC94
FB9A  FCB0 FCB0 FCB0          WORD  CODE08,CODE08,CODE08,CODE08,CODE00,CODE00,CODE00,CODE00
FBA0  FCB0 FC70 FC70
FBA6  FC70 FC70
FBAA  FCD8 FCD8 FCD8  C2ATAB  WORD  DSIX,DSIX,DSIX,KSIX,KSIX,KSIX,DSIX,DSIX
FBB0  FCEA FCEA FCEA
FBB6  FCD8 FCD8
FBBA  FCF6 FCF6 FCF6  C02ATAB WORD  IMMR,IMMR,IMMR,IMMR,IMMR,ONER,ONER,IMM
```

```
FBC0   FCF6 FCF6 FD0C
FBC6   FD0C FD00
FBCA   FD00 FC3A FC3A                WORD    IMM,DISASM2,DISASM2,DISASM2,DISASM2,DISASM2,DISASM2,DISASM2
FBD0   FC3A FC3A FC3A
FBD6   FC3A FC3A
                              *
FBDA   2CE0 F4AB              DISASM  XOP     @MESS03+1,3          PRINT INDENTATION
FBDE   C08C                           MOV     R12,R2               PRINT ADDRESS
FBE0   06A0 F5D6                      BL      @HEXOUT0
FBE4   C17C                           MOV     *R12^,R5             R5:=OPCODE
FBE6   C085                           MOV     R5,R2                PRINT OPCODE
FBE8   06A0 F5D2                      BL      @HEXOUT
FBEC   C185                           MOV     R5,R6
FBEE   09C6                           SRL     R6,12
FBF0   1605                           JNE     DISASM1
FBF2   C185                           MOV     R5,R6
FBF4   0976                           SRL     R6,7
FBF6   C026 FB8A                      MOV     @C0ATAB(R6),R0
FBFA   0450                           B       *R0
FBFC   0286 0003              DISASM1 CI      R6,#0003
FC00   1B2A                           JH      CODE4
FC02   0286 0002                      CI      R6,#0002
FC06   141A                           JHE     CODE2
                              *
FC08   C005                           MOV     R5,R0
FC0A   0240 0F00                      ANDI    R0,#0F00
FC0E   0960                           SRL     R0,6
FC10   0220 F70A                      AI      R0,TYPE4
FC14   06A0 FD8C                      BL      @PMNEM
FC18   C085                           MOV     R5,R2
FC1A   0242 00FF                      ANDI    R2,#00FF
FC1E   0285 1D00                      CI      R5,#1D00
FC22   1407                           JHE     CODE100
FC24   0282 0080                      CI      R2,#0080
FC28   1A02                           JL      CODE101
FC2A   0262 FF00                      ORI     R2,#FF00
FC2E   0A12                   CODE101 SLA     R2,1
FC30   A08C                           A       R12,R2
FC32   2CE0 F4C7              CODE100 XOP     @HASH,3              PRINT "#"
FC36   06A0 F5D6                      BL      @HEXOUT0
FC3A   2C40                   DISASM2 XOP     R0,1                 RETURN
                              *
FC3C   C185                   CODE2   MOV     R5,R6
FC3E   0246 1C00                      ANDI    R6,#1C00
FC42   0986                           SRL     R6,8
FC44   C006                           MOV     R6,R0
FC46   0220 F6AA                      AI      R0,TYPE2
FC4A   06A0 FD8C                      BL      @PMNEM
FC4E   0916                           SRL     R6,1
FC50   C026 FBAA                      MOV     @C2ATAB(R6),R0
FC54   0450                           B       *R0
                              *
FC56   0A26                   CODE4   SLA     R6,2
```

```
FC58  C006                      MOV   R6,R0
FC5A  0220 F66A                 AI    R0,TYPE1-16
FC5E  06A0 FD8C                 BL    @PMNEM
FC62  2C20 FD30                 XOP   @SIXBITS,0
FC66  2CE0 F4C9                 XOP   @COMMA,3           PRINT ","
FC6A  2C20 FD34                 XOP   @SIXBITD,0
FC6E  2C40                      XOP   R0,1               RETURN
                          *
FC70  0200 F77E           CODE00 LI   R0,TYPE9
FC74  06A0 FD8C                 BL    @PMNEM
FC78  2C40                      XOP   R0,1               RETURN
                          *
FC7A  C185                CODE02 MOV  R5,R6
FC7C  0246 01E0                 ANDI  R6,#01E0
FC80  0936                      SRL   R6,3
FC82  C006                      MOV   R6,R0
FC84  0220 F75A                 AI    R0,TYPE6
FC88  06A0 FD8C                 BL    @PMNEM
FC8C  0916                      SRL   R6,1
FC8E  C026 FBBA                 MOV   @C02ATAB(R6),R0
FC92  0450                      B     *R0
                          *
FC94  C005                CODE04 MOV  R5,R0
FC96  0240 03C0                 ANDI  R0,#03C0
FC9A  0940                      SRL   R0,4
FC9C  0220 F6CA                 AI    R0,TYPE3
FCA0  06A0 FD8C                 BL    @PMNEM
FCA4  0285 07C0                 CI    R5,#07C0
FCA8  1402                      JHE   CODE040
FCAA  2C20 FD30                 XOP   @SIXBITS,0
FCAE  2C40                CODE040 XOP R0,1               RETURN
                          *
FCB0  C005                CODE08 MOV  R5,R0
FCB2  0240 0300                 ANDI  R0,#0300
FCB6  0960                      SRL   R0,6
FCB8  0220 F74A                 AI    R0,TYPE5
FCBC  06A0 FD8C                 BL    @PMNEM
FCC0  C185                      MOV   R5,R6
FCC2  2C20 FD14                 XOP   @REG,0
FCC6  2CE0 F4C9                 XOP   @COMMA,3           PRINT ","
FCCA  0935                      SRL   R5,3
FCCC  0245 001F           CFIELD ANDI R5,#001F           PRINTS CONSTANT FIELD'
FCD0  C265 F4E2                 MOV   @NUMTAB(R5),R9
FCD4  2CC9                      XOP   R9,3
FCD6  2C40                      XOP   R0,1               RETURN
                          *
FCD8  2C20 FD30           DSIX  XOP   @SIXBITS,0         PRINTS SIXBIT FIELD + REGISTER
FCDC  2CE0 F4C9                 XOP   @COMMA,3           PRINT ","
FCE0  C185                      MOV   R5,R6
FCE2  0966                      SRL   R6,6
FCE4  2C20 FD14                 XOP   @REG,0
FCE8  2C40                      XOP   R0,1               RETURN
                          *
```

```
FCEA   2C20 FD30      7930  KSIX     XOP   @SIXBITS,0        PRINTS SIXBIT FIELD + CONSTANT
FCEE   2CE0 F4C9      7940           XOP   @COMMA,3          PRINT ","
FCF2   0955           7950           SRL   R5,5
FCF4   10EB           7960           JMP   CFIELD
                      7970  *
                      7980  *
                      7990  *
FCF6   C185           8000  IMMR     MOV   R5,R6             PRINTS REGISTER NO. + IMMEADIATE OPERAND
FCF8   2C20 FD14      8010           XOP   @REG,0
FCFC   2CE0 F4C9      8020           XOP   @COMMA,3          PRINT ","
FD00   2CE0 F4C7      8030  IMM      XOP   @HASH,3           PRINT "#"
FD04   C0BC           8040           MOV   *R12^,R2
FD06   06A0 F5D6      8050           BL    @HEXOUTO
FD0A   2C40           8060           XOP   R0,1              RETURN
                      8070  *
FD0C   C185           8080  ONER     MOV   R5,R6             PRINTS REGISTER
FD0E   2C20 FD14      8090           XOP   @REG,0
FD12   2C40           8100           XOP   R0,1              RETURN
                      8110  *
FD14   0246 000F      8120  REG      ANDI  R6,#000F          PRINTS REGISTER NO. E.G. R1,R10
FD18   0A16           8130           SLA   R6,1
FD1A   2CE0 F4D0      8140           XOP   @RR,3             PRINT "R"
FD1E   C266 F4E2      8150           MOV   @NUMTAB(R6),R9
FD22   0286 0014      8160           CI    R6,#14
FD26   1402           8170           JHE   REG00
FD28   0249 FF00      8180           ANDI  R9,#FF00
FD2C   2CC9           8190  REG00    XOP   R9,3
FD2E   2C40           8200           XOP   R0,1              RETURN
                      8210  *
FD30   C185           8220  SIXBITS  MOV   R5,R6             DECODES AND PRINTS SIXBIT SOURCE FIELD
FD32   1002           8230           JMP   SIXBIT
FD34   C185           8240  SIXBITD  MOV   R5,R6             DECODES AND PRINTS SIXBIT DEST. FIELD
FD36   0966           8250           SRL   R6,6
FD38   0246 003F      8260  SIXBIT   ANDI  R6,#003F          R6:=SIXBIT FIELD
FD3C   C1C6           8270           MOV   R6,R7
FD3E   0246 000F      8280           ANDI  R6,#000F          R6:=REGISTER
FD42   0947           8290           SRL   R7,4              R7:=ADDRESSING MODE
FD44   1603           8300           JNE   SIXBITO
FD46   2C20 FD14      8310           XOP   @REG,0            REGISTER ADDRESSING MODE
FD4A   101F           8320           JMP   SIXBIT3
FD4C   0287 0002      8330  SIXBITO  CI    R7,#2             SYMBOLIC ADDRESSING MODE
FD50   1611           8340           JNE   SIXBIT2
FD52   2CE0 F4C6      8350           XOP   @ATHASH,3         PRINT "@#"
FD56   C0BC           8360           MOV   *R12^,R2
FD58   06A0 F5D6      8370           BL    @HEXOUTO
FD5C   0286 0000      8380           CI    R6,#0000          INDEXED ADDRESSING MODE
FD60   1314           8390           JEQ   SIXBIT3
FD62   0201 2800      8400           LI    R1,'('            PRINT "("
FD66   2CC1           8410           XOP   R1,3
FD68   2C20 FD14      8420           XOP   @REG,0
FD6C   0201 2900      8430           LI    R1,')'
FD70   2CC1           8440           XOP   R1,3              PRINT ")"
FD72   100B           8450           JMP   SIXBIT3
```

```
FD74  0201 2A00       SIXBIT2 LI    R1,'*'         INDIRECT REGISTER ADDRESSING MODE
FD78  2CC1                    XOP   R1,3
FD7A  2C20 FD14               XOP   @REG,0
FD7E  0287 0003               CI    R7,#0003
FD82  1603                    JNE   SIXBIT3
FD84  0201 2B00               LI    R1,'+'         AUTO INCREMENT ADDRESSING MODEBT
FD88  2CC1                    XOP   R1,3
FD8A  2C40            SIXBIT3 XOP   R0,1           RETURN
                      *
FD8C  2CE0 F4B0       PMNEM   XOP   @MESS03+6,3    SUBROUTINE TO PRINT OPCODE MNEMONIC
FD90  04CA                    CLR   R10
FD92  C230                    MOV   *R0^,R8
FD94  C250                    MOV   *R0,R9
FD96  2CC8                    XOP   R8,3
FD98  2CE0 F4B1               XOP   @MESS03+7,3    PRINT " "
FD9C  045B                    RT
                      *
                      * INSTRUCTION M
                      *
FD9E  4F56 4520 4259  MMESS0  TEXT  'OVE BYTES '-
FDA4  5445 5320 00
FDA9    54 4F20 00    MMESS1  TEXT  'TO '-
FDAD    59 4553 204F  MMESS2  TEXT  'YES OR NO? '-
FDB2  5220 4E4F 3F20
FDB8  00
FDB9    00                    EVEN
FDBA  2CE0 FD9E       MOVE    XOP   @MMESS0,3      PRINT "OVE BYTES "
FDBE  06A0 F5AE               BL    @HEXIN
FDC2  162B                    JNE   MOVE04         JNE ON HEXIN FLAG
FDC4  C202                    MOV   R2,R8          R8:=LOW ADDRESS
FDC6  0281 3A00               CI    R1,':'
FDCA  1627                    JNE   MOVE04         ERROR
FDCC  06A0 F5AE               BL    @HEXIN
FDD0  1624                    JNE   MOVE04         JNE ON HEXIN FLAG
FDD2  8088                    C     R8,R2
FDD4  1B22                    JH    MOVE04
FDD6  C1C2                    MOV   R2,R7          R7:=HIGH ADDRESS
FDD8  2CE0 FDA9               XOP   @MMESS1,3      PRINT " TO"
FDDC  06A0 F5AE               BL    @HEXIN
FDE0  161C                    JNE   MOVE04         JNE ON HEXIN FLAG
FDE2  C182                    MOV   R2,R6
FDE4  2CE0 F4AA               XOP   @MESS03,3      PRINT INDENTATION
FDE8  2CE0 FDAD               XOP   @MMESS2,3      PRINT "YES OR NO?"
FDEC  2C81                    XOP   R1,2           GET CHARACTER
FDEE  2CC1                    XOP   R1,3           PRINT CHARACTER
FDF0  0281 5900               CI    R1,'Y'
FDF4  1610                    JNE   MON00C         RETURN TO MONITOR
FDF6  8206                    C     R6,R8
FDF8  130E                    JEQ   MON00C         LOW=DESTINATION
FDFA  1B04                    JH    MOVE01
FDFC  DDB8            MOVE00  MOVB  *R8^,*R6^
FDFE  81C8                    C     R8,R7
FE00  1B0A                    JH    MON00C         RETURN TO MONITOR
```

```
FE02  10FC            8960          JMP    MOVE00
FE04  81C6            8970  MOVE01  C      R6,R7
FE06  1BFA            8980          JH     MOVE00
FE08  6188            8990          S      R8,R6
FE0A  A187            9000          A      R7,R6
FE0C  D597            9010  MOVE02  MOVB   *R7,*R6
FE0E  0607            9020          DEC    R7
FE10  0606            9030          DEC    R6
FE12  8207            9040          C      R7,R8
FE14  14FB            9050          JHE    MOVE02
FE16  0460 F44E       9060  MONOOC  B      @MONOO
FE1A  0460 F464       9070  MOVE04  B      @MONO2               PRINTS " ??"
                      9080  *
                      9090  * INSTRUCTION Z
                      9100  *
FE1E  4E4F 5420 464F  9110  FMESSOO TEXT   'NOT FOUND'-
FE24  554E 4400
FE28                  9120          EVEN
FE28  C24F            9130  FIND    MOV    R15,R9
FE2A  C1C2            9140          MOV    R2,R7
FE2C  91F9            9150  FINDOO  CB     *R9^,R7
FE2E  1604            9160          JNE    FIND01
FE30  06C7            9170          SWPB   R7
FE32  91D9            9180          CB     *R9,R7
FE34  06C7            9190          SWPB   R7
FE36  1307            9200          JEQ    FIND02
FE38  83C9            9210  FIND01  C      R9,R15
FE3A  16F8            9220          JNE    FINDOO
FE3C  2CE0 F4AA       9230          XOP    @MESS03,3            PRINT INDENTATION
FE40  2CE0 FE1E       9240          XOP    @FMESSOO,3           PRINT "NOT FOUND"
FE44  10E8            9250          JMP    MONOOC
FE46  2CE0 F4AA       9260  FIND02  XOP    @MESS03,3            PRINT INDENTATION
FE4A  C089            9270          MOV    R9,R2
FE4C  0602            9280          DEC    R2
FE4E  06A0 F5D6       9290          BL     @HEXOUTO             PRINT ADDRESS
FE52  C087            9300          MOV    R7,R2
FE54  06A0 F5D2       9310          BL     @HEXOUT              PRINT CONTENTS
FE58  2CE0 F4B0       9320          XOP    @MESS03+6,3          PRINT "  "
FE5C  06A0 F5AE       9330          BL     @HEXIN
FE60  1606            9340          JNE    FIND03               JNE ON HEXIN FLAG
FE62  03A0            9350          WREN
FE64  DA42 FFFF       9360          MOVB   R2,@>FFFF(R9)
FE68  06C2            9370          SWPB   R2
FE6A  03A0            9380          WREN
FE6C  D642            9390          MOVB   R2,*R9
FE6E  0281 2000       9400  FIND03  CI     R1,' '
FE72  13E2            9410          JEQ    FIND01
FE74  1000            9420          JMP    MONOOC
                      9430  *
                      9440  * INSTRUCTION G
                      9450  *
FE76  16CF            9460  GO      JNE    MONOOC               JNE ON HEXIN FLAG
FE78  0452            9470          B      *R2
```

```
                          *
                          * INSTRUCTION L
                          *
FE7A   4C4F 573D 00       LMESSO   TEXT   'LOW='-
FE7F     00                        EVEN
FE80   1602               LOW      JNE    LOW00            JNE ON HEXIN FLAG
FE82   C3C2                        MOV    R2,R15
FE84   10C8                        JMP    MON00C
FE86   2CE0 F4AA          LOW00    XOP    @MESS03,3        PRINT INDENTATION
FE8A   2CE0 FE7A                   XOP    @LMESSO,3        PRINT "LOW="
FE8E   C08F                        MOV    R15,R2           PRINT USER WORKSPACE
FE90   06A0 F5D6                   BL     @HEXOUTO
FE94   10C0                        JMP    MON00C
                          *
                          * INSTRUCTION W
                          *
FE96   1603               WP       JNE    WP00             JNE ON HEXIN FLAG
FE98   C802 F3FA                   MOV    R2,@2*R13+INT1WP
FE9C   10BC                        JMP    MON00C
FE9E   2CE0 F4AA          WP00     XOP    @MESS03,3        PRINT INDENTATION
FEA2   2CE0 FA81                   XOP    @IMESS1+2,3      PRINT "WP="
FEA6   C0A0 F3FA                   MOV    @2*R13+INT1WP,R2
FEAA   06A0 F5D6                   BL     @HEXOUTO
FEAE   10B3                        JMP    MON00C
                          *
                          * INSTRUCTION O
                          *
FEB0   C10F               OUTPUT   MOV    R15,R4
FEB2   C0C3                        MOV    R3,R3
FEB4   1601                        JNE    OUTPUTO
FEB6   C102                        MOV    R2,R4
FEB8   0244 FFFE          OUTPUTO  ANDI   R4,#FFFE         MAKE ADDRESS EVEN
FEBC   C144                        MOV    R4,R5
FEBE   0245 FFF0                   ANDI   R5,#FFF0
FEC2   0225 0080                   AI     R5,#80
FEC6   2CE0 F4AB          OUTPUT1  XOP    @MESS03+1,3      PRINT INDENTATION
FECA   C084                        MOV    R4,R2
FECC   06A0 F5D6                   BL     @HEXOUTO         PRINT ADDRESS
FED0   2CE0 F4B1                   XOP    @MESS03+7,3      PRINT " "
FED4   C0B4               OUTPUT2  MOV    *R4^,R2
FED6   06A0 F5D2                   BL     @HEXOUT
FEDA   8144                        C      R4,R5
FEDC   1304                        JEQ    OUTPUT3
FEDE   2520 F540                   CZC    @MASK16,R4
FEE2   13F1                        JEQ    OUTPUT1
FEE4   10F7                        JMP    OUTPUT2
FEE6   2C81               OUTPUT3  XOP    R1,2             GET CHARACTER
FEE8   0281 2000                   CI     R1,' '
FEEC   1694                        JNE    MON00C
FEEE   0225 0040                   AI     R5,#40           AN EXTRA 4 LINES
FEF2   10E9                        JMP    OUTPUT1
                          *
                                   END
```

256 BYTES FROM FEFF TO FFFF

```
** CROSS REFERENCE LISTING **

LABEL NAME   DEFN  VALUE STATEMENT NUMBERS

ATHASH        930  F4C6  8350
CO2ATAB      6900  FBBA  7590
COATAB       6870  FB8A  7040
C2ATAB       6890  FBAA  7360
CFIELD       7810  FCCC  7960
CHAR         2550  F638  1040
CHAR00       2570  F63C  2550
CHAR01       2580  F640  2810
CHAR02       2610  F648  2790  2680  2650
CHAR03       2690  F65A  2630
CHAR04       2730  F664  2700
CHAR05       2770  F66E  2750
CODE00       7480  FC70  6880  6870
CODE02       7520  FC7A  6870
CODE04       7620  FC94  6870
CODE040      7700  FCAE  7680
CODE08       7720  FCB0  6880
CODE100      7250  FC32  7190
CODE101      7230  FC2E  7210
CODE2        7290  FC3C  7090
CODE4        7390  FC56  7070
COMMA         950  F4C9  8020  7940  7870  7790  7440  3810
CR            290  000D  5230   900   870
DELIM        3760  F87E  4390  4300  3960  3880
DELIM00      3810  F88C  3770
DISASM       6930  FBDA  6740  6290
DISASM1      7060  FBFC  7010
DISASM2      7270  FC3A  6910
DOT          3050  F7DC  3160
DSIX         7860  FCD8  6890
ERR1         5140  F9E0  4980  4720  4240  4210  4190
ERROR        3840  F892  5140  3790  3720  3430  3300
FIND         9130  FE28  1030
FIND00       9150  FE2C  9220
FIND01       9210  FE38  9410  9160
FIND02       9260  FE46  9200
FIND03       9400  FE6E  9340
FMESS00      9110  FE1E  9240
GETRC        4540  F942  5110  4890  4840  4430  4370  4320  4290  4030
GETRC00      4580  F950  4660
GETRC01      4670  F966  4620  4600
GETRC02      4680  F96A  4710
GETRC03      4730  F976  4690
GO           9460  FE76  1030
HASH          940  F4C7  8030  7250  4990  4460  4100
HEXIN        1900  F5AE  9330  8810  8750  8700  5360  5000  4470  4110  2300   690
HEXIN1       1930  F5B4  2010
HEXIN1       1960  F5BC  2030
HEXIN2       2020  F5CA  1970
```

```
HEXOUT      2100  F5D2  9880  9310  6980  6580  6180  5340  2490
HEXOUT0     2110  F5D6  9850  9700  9590  9290  8370  8050  7260  6950  6470  6330  6240  6210  5910  2470
HEXOUT1     2140  F5DE  2200
HEXOUTX     2120  F5D8  1220
HEXTAB       990  F4D2  2170  1960
I1TAB       5770  FA76  5840
IMESS0      5780  FA7C  5920
IMESS1      5790  FA7F  9680  6450  6190
IMESS2      5800  FA85  6310
IMESS3      5810  FA87  6220
IMM         8030  FD00  6910  6900
IMMR        8000  FCF6  6900
INIT0        460  F40C   490
INIT1        530  F422   580
INIT2        640  F440  1230
INITIAL      420  F400   840
INST00      3110  F7E2  3850  3730  3090
INST01      3180  F7F8  3270
INST02      3280  F812  3230  3210
INST03      3360  F822  3420
INST04      3400  F82C  3370
INST05      3470  F838  3390
INST06      3490  F83E  3520
INST07      3530  F846  3500
INST08      3640  F862  4500  4260  4120  4080  3930
INST09      3670  F86A  3700
INST10      3740  F87A  3800  3310
INSTANT     3090  F7DE  1040
INT1        5830  FA8E   840
INT1AA      5870  FA96  5770
INT1BB      5900  FAA0  5880
INT1CC      6040  FAC0  5940  5770
INT1DD      6070  FACA  6100
INT1EE      6150  FAD8  6080
INT1FF      6250  FAFA  6340  5990
INT1GG      6280  FB02  5770
INT1WP       320  F3E0  9690  9650  6380  5670  5660  5630  5490  5270   840   610
INTAB       1010  F502   710
KSIX        7930  FCEA  6890
LF           300  000A  5230   900   870
LMESS0      9510  FE7A  9570
LOW         9530  FE80  1040
LOW00       9560  FE86  9530
MASK16      1070  F540  9910  5690
MASK3       1090  F544  6620
MASK32      1060  F53E  2780
MASK8       1080  F542  5870
MESS00       860  F488   670
MESS01       870  F49D   680   660
MESS02       890  F4A6  5960  3840   750
MESS03       900  F4AA  9860  9830  9670  9560  9320  9260  9230  8840  8600  8550  6930  6440  5890  5350  3280  2500
                        2450  2100
MESS04       920  F4B3  1210
```


MMESS0	8650	FD9E	8690															
MMESS1	8660	FDA9	8800															
MMESS2	8670	FDAD	8850															
MON00A	2420	F61E	2720	2390														
MON00B	6810	FB86	6610															
MON00C	9060	FE16	9960	9710	9660	9600	9550	9460	9420	9250	8950	8910	8890					
MON00	680	F44E	9060	6810	6260	5440	3740	2420	760									
MON01	710	F45A	740															
MON02	750	F464	9070															
MON03	770	F46A	720															
MONITOR	660	F446	1030															
MOVE	8690	FDBA	1030															
MOVE00	8930	FDFC	8980	8960														
MOVE01	8970	FE04	8920															
MOVE02	9010	FE0C	9050															
MOVE04	9070	FE1A	8820	8780	8760	8740	8710											
NUMTAB	1000	F4E2	8150	7820	6530	4680												
ONER	8080	FD0C	6900															
OPEN	2260	F5F4	1040															
OPEN00	2280	F5F8	2260															
OPEN01	2290	F5FC	2410	2370														
OPEN02	2340	F60A	2310															
OPEN03	2380	F614	2350															
OUTPUT	9750	FEB0	1030															
OUTPUT0	9790	FEB8	9770															
OUTPUT1	9830	FEC6	9980	9920														
OUTPUT2	9870	FED4	9930															
OUTPUT3	9940	FEE6	9900															
PADDRC	2450	F622	3150	2590	2290													
PMNEM	8550	FD8C	7760	7660	7570	7490	7420	7340	7150									
PRINT	6680	FB64	1030															
PRINT00	6720	FB6C	6700															
PRINT01	6740	FB74	6800	6760														
R	960	F4CB	6520															
R0	120	0000	8580	8570	8530	8200	8100	8060	7910	7840	7750	7740	7730	7720	7700	7650	7640	7630
			7620	7600	7590	7560	7550	7500	7480	7460	7410	7400	7370	7360	7330	7320	7270	7140
			7130	7120	7110	7050	7040	6070	6060	6050	5850	5840	5170	5070	4940	4900	4740	4380
			4330	4310	4070	3970	3920	3910	3890	3680	3660	3650	3610	3600	3550	3540	3240	3120
			3110	2510	2180	2170	2120	1460	1440	630	620	570	560	540	500	480	470	440
R1	130	0001	9950	9940	9400	8880	8870	8860	8730	8520	8510	8470	8460	8440	8430	8410	8400	6790
			6780	5980	5970	5950	5390	5160	5150	5120	5080	4970	4950	4930	4910	4860	4820	4810
			4800	4640	4630	4610	4590	4580	3780	3760	3710	3640	3630	3290	3240	3220	3200	3190
			3180	2770	2760	2740	2710	2690	2660	2620	2610	2380	2340	2170	2160	2150	1960	1940
			1930	1900	1720	1710	1670	710	560	510	470	450						
R10	220	000A	8560	6490	5840	5060	5010	4490	3690	3590	1810	1710	1690	1680	1330	1240	1170	1160
			640															
R11	230	000B	6300	6280	4790	1830	1820	1810	1720	1570	1460	1250						
R12	240	000C	8360	8040	7240	6960	6940	6720	6710	6680	6170	6110	6070	5900	5680	5180	4960	4850
			4790	1540	1410													
R13	250	000D	9690	9650	6770	6750	6730	6200	1830	1670								
R14	260	000E	6110	5680	4150	3680	3100	2780	2740	2670	2640	2580	2560	2480	2460	2400	2360	2330
			2280	2270	1330	1250	1240	600										

```
R15         270  000F  9750  9580  9540  9210  9130  6680  6320  6230  5710  5690   610   590
R2          140  0002  9870  9840  9780  9690  9650  9580  9540  9470  9390  9370  9360  9300  9280  9270  9140  8830
                       8790  8770  8720  8360  8040  7240  7230  7220  7200  7170  7160  6970  6940  6710  6570  6460
                       6420  6320  6230  6200  6170  6090  6040  5900  5660  5380  5330  5050  5030  4740  4730  4700
                       4680  4670  4650  4570  4480  4250  4230  4220  4200  4180  4170  4160  4150  3280  3260  3250
                       3170  3100  2560  2480  2460  2330  2270  2150  2140  2000  1990  1910  1200   540   520
R3          150  0003  9760  6690  6400  5640  3530  3490  3480  2800  2600  2570  2210  2190  2130  2110  2040  1980
                       1920   790
R4          160  0004  9910  9890  9870  9840  9800  9790  9780  9750  6570  6460  6430  6420  6390  6380  5630  5600
                       5550  5500  5490  5480  3600  3580  3570  3540  3510  3470  2640  2580  2020  2000  1960  1950
                        800   780   770   730   710   700
R5          170  0005  9970  9890  9820  9810  9800  8240  8220  8080  8000  7950  7880  7820  7810  7800  7770  7720
                       7670  7620  7520  7290  7180  7160  7110  7020  6990  6970  6960  6620  6600  6590  6530  6480
                       5380  5330  5280  5270  4680  4560
R6          180  0006  9030  9010  9000  8990  8970  8930  8900  8830  8380  8280  8270  8260  8250  8240  8220  8160
                       8150  8130  8120  8080  8000  7890  7880  7770  7590  7580  7550  7540  7530  7520  7400  7390
                       7360  7350  7320  7310  7300  7290  7080  7060  7040  7030  7020  7000  6990  4640  4550  4540

R7          190  0007  9300  9190  9180  9170  9150  9140  9040  9020  9010  9000  8970  8940  8790  8490  8330  8290
                       8270  4380  4310  4250  4130  4070  4000  3980  3970  3920  3890  3580  3560  3530  3490  3410
                       3400  3380  3360  3350
R8          200  0008  9040  8990  8940  8930  8900  8770  8720  8590  8570  5420  5410  5310  5290  5030  4480  3360
                       3140  3130
R9          210  0009  9390  9360  9270  9210  9180  9150  9130  8580  8190  8180  8150  7830  7820  6560  6550  6540
                       6530  6510  6500  6000  5930  5870  5050  3380  3140  1690  1170   630
REG        8120  FD14  8480  8420  8310  8090  8010  7900  7780
REG00      8190  FD2C  8170
REGIST     6380  FB16  1030
REGIST0    6430  FB22  6410
REGIST1    6500  FB38  6630
REGIST2    6520  FB3E  6640
RESTAB      840  F474   640   450
RR          970  F4D0  8140  5100  4880  4420  4360  4280  4020
SETBP      5270  FA02  1040
SETBP00    5300  FA0E  5430
SETBP01    5390  FA2C  5370
SETBP02    5440  FA3C  5400
SETUP      5620  FA56  5560  5510
SETUP00    5670  FA64  5650
SETUP01    5720  FA72  5700
SIX        4790  F97C  4060  3950  3900  3870
SIX00      4860  F98E  4830
SIX02      4960  F9AE  4920
SIX03      4970  F9B0  4870
SIX04      5050  F9C6  5020
SIX05      5060  F9C8  5040
SIX06      5150  F9E4  5130
SIX07      5170  F9E8  5090
SIXBIT     8260  FD38  8230
SIXBIT0    8330  FD4C  8300
SIXBIT2    8460  FD74  8340
SIXBIT3    8530  FD8A  8500  8450  8390  8320
SIXBITD    8240  FD34  7450
```

```
SIXBITS     8220  FD30  7930  7860  7690  7430
SMFSS0      5230  F9EE  6150  5300
SMFSS1      5250  F9FA  6160  5320
STEP        5600  FA52  1040
SUBTAB      1030  F516   780
T1          3870  F898  3040
T100        3910  F8A6  4040
T101        3920  F8A8  4440  4340
T2          3950  F8AC  3040
T200        4020  F8C2  3990
T201        4030  F8C6  4010
T3          4060  F8CC  3040
T4          4100  F8D4  3040
T400        4230  F8FA  4140
T5          4280  F904  3040
T6          4360  F91A  3040
T7          4420  F92A  3040
T8          4460  F934  4400  3040
T9          3630  F860  3040
TRACE       5480  FA40  1040
TYBASE      3030  F7B8  3580
TYPE1       2880  F67A  7410  3350  3000
TYPE2       2900  F6AA  7330  3000
TYPE3       2910  F6CA  7650  3000
TYPE4       2930  F70A  7140  3000
TYPE5       2950  F74A  7750  3000
TYPE6       2960  F75A  7560  3010
TYPE7       2970  F76E  3010
TYPE8       2980  F776  3010
TYPE9       2990  F77E  7480  3010
TYSHIFT     3020  F7AE  3540
TYSUB       3040  F7CA  3600
TYTAB       3000  F79A  3480  3410  3010
WORKSP       340  F3A0  6250  1190   840   620   430
WP          9640  FE96  1030
WP00        9670  FE9E  9640
XCUTE       5550  FA4C  1040
XOP0        1150  F546   850
XOP0AA      1240  F562  1180
XOP0XXX     1190  F54E  1700
XOP1        1320  F568   850
XOP2        1400  F56E   850
XOP2AA      1420  F574  1430
XOP3        1530  F580   850
XOP3AA      1550  F584  1590  1560
XOP4        1660  F590   850
XOP4AA      1680  F594  1730
XOP5        1800  F5A2   850
XOP5AA      1810  F5A4  1840
XOPTAB       850  F47C   510
XOPWP        330  F3C0  1200   640   630   520
NOLABELS=  230 NOPROBES=   390 ALPHA=      1.69
SYMBOL TABLE 57.35 % FULL
```