GA 18/30 Emulator

With

TSO/CYTOS

 Copyright (C) 1995-2014 Edward S. Quicksall & Developer Services, Inc

 This program is free software: you can redistribute it and/or modify

 it under the terms of the GNU General Public License as published by

 the Free Software Foundation, either version 3 of the License, or

 (at your option) any later version.

 This program is distributed in the hope that it will be useful,

 but WITHOUT ANY WARRANTY; without even the implied warranty of

 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

 GNU General Public License for more details.

 For a copy of the GNU General Public License see

 <http://www.gnu.org/licenses/>.

This is an emulation of the GA (General Automation) 18/30. The GA 18/30 is an upgrade of the IBM 1130. The GA 18/30 was released in 1969. It was intended to be a mix of the IBM 1130 and IBM 1800.

Major differences include:

* an interrupt structure very similar to the IBM 1800, not the IBM 1130
* additional instructions include compare and register transfer instructions
* special instructions and hardware were introduced around 1975 to allow TSO to run
* “aux core” – this is additional memory that is hidden from DM2 programs (“main core”)
* instructions to open and close aux core
* instructions to mask interrupts
* a shared storage subsystem allowing two 18/30’s to share the same storage
* mapped memory allowing multiple main core segments
* programmed interrupts
* locations 1, 2 and 3 do not correspond to index registers 1, 2 and 3

Most IBM 1130 programs will run on the GA 18/30 provided they use the appropriate drivers.

TSO is an operating system which is used to run virtual DM2 programs. CYTOS is a CLI which gives the ability to compile, link and run programs from multiple terminals with TSO.

Credits:

* TSO Supervisor – written by Don Nichols (formally president of DNA Systems)
* CYTOS – written by Timothy Mellon (chairman and majority owner of Pan Am Systems)
* DCIP – written by Bud Blackney (formally CEO of DNA Systems)
* XDUP – written by Joyce Stout (formally co-owner of DNA Systems)
* XBASE – written by Dave Morse (formally employee of DNA Systems)
* RJE, card reader spooling, IBM BI-SYNC emulators and this emulator – written by Eddy Quicksall (formally employee of DNA Systems)
* MSQ (Multiple Spooling Queues) – written by Ed Russell
* FORTRAN improvements – written by Dave Morse and others from University of Michigan
* Many support programs – written by Jim Cunningham (formally owner of Developer Services, Inc)
* Mag tape drivers – John Currey (formally employee of DNA Systems)
* Fast DISKZ and RENUMBER command – Richard J. Sherin

What is included

* 1830.exe – emulator
* cold0.cdi – coldstart card
* 1830.cfg – configuration file
* catpr.exe – a Windows version of the CATPR command
* run.bat – a batch file to run the program
* dsm0.dsk and dsm1.dsk – the simulated CDC Disk Storage Modules (similar to IBM 3330’s)
* README.DOCX – this document
* Documents
* ATTACH.pdf – how to use the ATTACH command
* cytos.txt – how to use CYTOS
* DNA Sort.txt – how to use the DNA sort
* DNA Systems Fortran IV Language.pdf – improvements to the University of Michigan FORTRAN
* dna\_manual.txt – how to use TSO
* Switches.pdf – how TSO interprets the data switches
* TSOE2.TXT – error messages
* XBASE.TXT – how use XBASE

Setup

The emulator was designed to run on MSDOS. The configuration provided here also runs on XP using command.com. The system uses EMS, XMS and DPMI memory.

To run on MSDOS 6.22:

Add these to config.sys:"

device=himem.sys"

device=emm386.exe ram"

The mag tape uses SCSI. A driver from Overland Data can be used (see <http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0CCwQFjAC&url=http%3A%2F%2Fwww.thecomputerarchive.com%2Farchive%2FStorage%2FControllers%2FOverland%2520Data%2520-%2520TX-8%25209-track%2520tape%2520controller.PDF&ei=ZrBsVLfRGsOZgwSR2ILYDw&usg=AFQjCNGhQ90xG8IAWCN9Z32AhQtFQIFpog&sig2=gKdYl9Iz9G6quIdRaU4JTw&bvm=bv.80120444,d.eXY>).

Here are the options:

\outright\api /16384 /m=16

To run on XP:

The emulator will run from an XP prompt (cmd.exe) but runs best if the MSDOS prompt (command.com) is used.

 Edit Windows\System32\Config.NT

 dos=high, umb

 device=%SystemRoot%\system32\himem.sys

 files=40

Make a shortcut to C:\WINDOWS\system32\command.com /c run.bat

Right click and select Properties:

 Programm Tab:

 -Working Directory = wherever 1830.exe is

 Memory Tab:

 -Conventional memory: auto

 -Initial environment: auto

 -EMS: 16384

 -XMS: 16384

 -DPMI: auto

Running the emulator

 Running 1830.exe without a coldstart card image will immediately stop and if “g” (go) is typed a small test program will output periods every second. Using run.bat will use the coldstart card.

E:\>run /?

1830 emulation by Eddy Quicksall (Eddy@Quicksall.com)

for Developer Services, Inc. (Wichita, KS)

usage: 1830 [master] [options]

master = 1 for master, 0 for slave (TSO clustering)

options:

/h or /? for help

/v is verbose

/sxxxx sets the data switches to hex xxxx

/bcdiFile means to 'boot and go' using cdiFile (Boot Card Image)

/ccfgFile = name of configuration file (default = 1830.cfg)

ctrl/F1 = 1830 HALT (enters the panel routine)

ctrl/F10 = 1830 Console Interrupt

To see panel help, execute 1830 and type '?' at the '\*' prompt

E:\>

To exit the emulator, press ctrl/F1 then at the \* prompt, type q<cr>.

Output of the emulator will look like this:

1830 emulation by Eddy Quicksall (Eddy@Quicksall.com)

for Developer Services, Inc. (Wichita, KS)

usage: 1830 [master] [options]

 master = 1 for master, 0 for slave (TSO clustering)

 options:

 /h or /? for help

 /v is verbose

 /sxxxx sets the data switches to hex xxxx

 /bcdiFile means to 'boot and go' using cdiFile (Boot Card Image)

 /ccfgFile = name of configuration file (default = 1830.cfg)

ctrl/F1 = 1830 HALT (enters the panel routine)

ctrl/F10 = 1830 Console Interrupt

To see panel help, execute 1830 and type '?' at the '\*' prompt

Shared disks are NOT supported, need an ID (0 or 1)

Printer 0 (P1403) = lpt1

 channels = -1 -1 -1 -1 -1 -1 -1 0 -1 -1 -1 -1

Printer 1 (P1132) = p1132.out

 channels = -1 -1 -1 -1 -1 -1 -1 0 -1 -1 -1 -1

DSM's: dsm0.dsk dsm1.dsk

Tape 0 has 0 buffers allocated

EDDY\_TESTING is set -- don't use on DSI system.

LOG\_ENABLED is set

Press ◄┘ to continue.

MTCA ports: COM1=3F8, COM2=2F8

EMS segment: CB00

no user option

cart id table update complete

$acct not found in let/flet

TS0 00/00/80 2036 HRS

PLEASE LOGON

==>

Logging on

When you execute you will get the message “PLEASE LOGON”. You can log onto 2202, 2203, 2205, 7ff1 or 031d. The user ID must be JC and the project must be TSO. The password is X. You can change these, see the CYTOS.TXT manual. Here is an example:

 LOGON JC TSO 2202

 PASSWORD> X

If logging in at a terminal, press Enter (CR) at the terminal to get the attention of the system then use a different cartridge ID for the LOGON (e.g., 2203 or 2205).

After logging in, you will probably want to access the standard library files. Do that by attaching ID’s 0002 and 0003 as follows: ATTACH (2,3). Most programs mentioned below will require these cartridge ID’s to be attached.

Terminal setup

Up to 16 terminals are possible with MSDOS using a DigiBoard. However this build does not support the DigiBoard. This build supports up to 4 terminals (COM1, COM2, COM3 and COM4). The characteristics are:

 Speed: 57600 BAUD

 Data frame: 7 bit

 Parity: None

 Stop bits: 1

Flow control: none or x-on/x-off (x-on/x-off can also be used from the terminal’s keyboard by pressing ctrl/q for x-on and ctrl/s for x-off)

Notes

* The familiar \*, ? and arrow up characters as used in a DOS prompt are not supported in CYTOS. The exception is with the DESCRIBE command; the \* may be used as a wildcard in the position of the extension. E.g., DESCRIBE THIS.\* but DESCRIBE THIS\*.\* will give an error.
* Print spooling. To understand the print spooling, see “I/O SPOOLING LOGIC” and “APPENDIX G” in the dna\_manual.txt.
* Multiple terminals. This allows multiple users to access the TSO system.
* Pause. A PAUSE will not stop other terminals. A PAUSE is displayed to the user as $PAUSE. See “VIRTUAL DM-2 PAUSE PROCESSING” in dna\_manual.txt.
* When finished with a terminal or console session, type “LOGOFF” or “DONE”. This will flush your spooling queues. Type ctrl/Enter or ctrl/F10 to get the LOGON prompt again.
* Printing will go through the queue and directly to the printer if no other terminal is currently printing. Sometimes it is useful to send all output to the queue without printing. To enable that feature and other options, run QCON. To clear the queue, run Q and use R for the FUNCTION, then S for the INDEX, then press Enter for each prompt.
* If you don’t change 1830.cfg, 1403 printer output goes to LPT1 and 1132 printer output goes to P1132.OUT.
* If you type BATCH and you are at the console, it will attempt to read from the card reader. Since there is no card reader you will get a $PAUSE 4000. At this point you can type “TYPE” and the card input will come from the terminal (you won’t get a prompt). Type any DM2 command. Typing // CYTOS will return you to the CYTOS logon prompt.
* The immediate stop “button” is ctrl/F1. At that point type “?” to see your options. Typing Q will exit the emulator. There is no “program stop”. The internal debugger only works from the console panel routine.
* If you run on XP and you don’t have multiple processors, the emulator will hog so much time that it will be difficult to use other windows. In that case press ctrl/F1 (immediate stop), use the other windows and then enter G to continue emulation.
* The EXPLAIN command uses the file CYTXP.S to be present. It is on cartridge 7FF3, therefore mount 7FF3 (e.g., M (2202, 7FF3)).
* To access “attach” cartridges, run the emulator in “batch only” mode. This is accomplished by setting data switch 6 during boot. To do this use the option /s0200 when executing the emulator. For example:

run /s0200

* To clear the print queue during boot, run the emulator with switch 5 on. For example:

run /s0400

* The CHECKSTOP CORE DUMP option mentioned in dna\_manual.txt has been disabled. Therefore if an invalid instruction is encountered the emulator will enter the panel routine which will stop all terminals. Look at the console and notice the IAR; use the ‘U’ command to unassembled the surrounding code.
* Interrupt request is ctrl/enter.

Shared computers (TSO clustering)

Two computers running TSO can share the same storage. Connect a serial line between the two computers using com1. On one computer use the option 0 and on the other use the option 1. The same DSMs must be available to both computers (usually on the network). Each time a DSM is accessed, TSO will send a signal over the serial line (distributed lock). If the opposing computer currently has a lock then the computer sending the signal will suspend the area (terminal). When the opposing computer releases the lock it will send a signal to the partner computer which will wake up the area and try to get the lock again. This does not guarantee read/modify/write continuity … that is the responsibility of the user software.

Writing assembler programs

* You can’t use WAIT. To pause use ‘BSI L $PRET’ where $PRET is equated to /28.
* Don’t expect locations 1, 2 and 3 to be index registers.
* The interrupt transfer vector may not be in the 1130 locations. I will have to see how we did that.
* When you start an I/O with XIO be sure to immediately check for BUSY. If you don’t get BUSY then it means the I/O did not start. If you check for READY before the XIO you could hit a window where it was ready when you checked but by the time you hit the XIO it became not ready. The XIO is protected for one instruction which gives time for you to test for BUSY. Don’t forget to set $IOCT because TSO will wait for that to go to zero before swapping to another area.
* After starting an I/O, don’t use any timing loops or a WAIT instruction as other interrupts will bounce off of the WAIT.
* If you hit an invalid instruction the emulator will enter the panel routine with a \* prompt. That stops all emulation. Type ? to see your options. To exit the faulting program type ‘R IAR’ then enter 0038 ($EXIT) and ‘G’ followed by Enter twice.

Useful programs

Check dna\_manual.txt for useful programs referred to as CORELOAD(S). Browse through the .S files and you may find some that are not mentioned in dna\_manual.txt. To execute many of the standard programs, you will need to attach cartridge IDs 0002 and 0003 (ATTACH (2, 3) LIST).

Some programs can take arguments on the command line (but not all). If arguments are on the command line they are given as the 3rd parameter enclosed in single quotes (‘). For example, LETP,,’2202’ will print a list of LET entries. Some other programs take arguments as the 2nd parameter without using the single quotes. For example SYSOUT HERE would send all system output to the terminal.

* Q – control spooling queues. Use the command “P” to post a queue to the printer (e.g., to post queue 1234, type P1234).

To clear the queue:

==> Q

 R

 S

 (CR)

 (CR)

 (CR)

 Y

* SYSOUT will direct you printer output to HERE, P1132 or P1403 (e.g., SYSOUT P1403)
* WHERE – searches all DSM’s to find the file. Does not take arguments, respond to the prompt.
* QCON – used to control the spooling queues. Arguments can be used: e.g. QCON,,’CPR’ will
set the queue to prevent Close, Post and Remove (i.e., cause all output to go into a single queue entry)

The qcon setting invoked by the user remain in effect until:

1. The system is re IPL’d

2. QCON is run changing the current settings

* CLOCK – set the date and time
* ASYBD - Prompts for an async area #, tells the current speed setting and allows modification.
Speeds are:
* SLOW - Prompts for an area # and will allow it to run only if nothing else going on.
* FAST - Changes SLOW back to normal.
* ENDAL - Allows any area to be turned OFF or ON
* RENUMBER PROG1.S PROG2.S – will renumber a FORTRAN program.
* TELL – Send a message to another terminal or the console
* CGTYP – Change cartridge type (system, non-system, attach or spooling)
* LOOKQ – Look at a queue item.
* OZAP – The OZAP command can be used to patch a .O (OZAP *prog.o location old new\_value(s)*).
* CIZAP – The CIZAP command can be used to patch a .L (CIZAP *prog.l location old new\_value(s)*).
* ZAP – The ZAP command can be used to patch a phase (ZAP *phase location old new\_value(s)*).
* SORT – A comprehensive sort program. See “DNA Sort.PDF”
* WATCH – This program will watch all areas. But it requires an ADDS terminal emulation since it uses ADDS cursor positioning commands.
* WPROG – Will monitor the programs and mounts for each terminal. WPROG,,’1’ will monitor just area 1.

Interesting source files

* $USER.S – The initialization program executed with a LOGON. In this file you can automatically configure each terminal as required.
* $STRT.S – Executed when a coldstart is forced.
* DONE.S – Executed by the user when he is done

Windows utility programs

These utility programs run from Windows. They must be executed from the same folder that holds the DSMs.

- catpr.exe catpr <ctid> can be used to dump LET on cartridge *ctid*

- getfl80.exe getfl80 <dsname> can be used to extract the file given by dsname (e.g. getfl80 tmp4.s(220d)). The file will be converted from 80 character EBCDIC to ASCII when it is copied.

- ldfil.exe ldfil <dsname> can be used to copy a file given by dsname to a DSM. The file (dsname) must already exist on the cartridge and must be large enough to receive the file. The file will be converted from ASCII to 80 character EBCDIC when it is copied. To make a new file use the CYTOS command ALLOCATE DSNAME.S(ctid) CARDS *XX.*

- vmware If you run the emulator with an XP virtual machine (e.g., vmware) you can setup named pipes for the serial lines that the terminals use. With that done you can bring up a serial program that accepts named pipes and run several terminals from within Windows. The pipe system in Windows will buffer up characters and feed them into the vmware virtual machine. The emulator will buffer up to 16 characters and feed them into TSO with 55ms spacing. That overcomes an overrun problem that could occur during the task switch from Windows to vmware. This is useful since you can scroll back with most serial programs but scrolling is not supported at the console.

Console

The console is the window shown after starting up. It is a simple DOS window so it is not possible to scroll back. Pressing ctrl/f1 will enter the panel routine (it is not possible to enter the panel routine from a terminal window). From the panel routine registers and memory can be displayed. An example follows:

==>

Stop AUX=clsd MSK=00 PIR=0000 IRR=0000 ISR=0000 Lvl=255

ACC=1081 EXT=0040 XR1=1E95 XR2=0000 XR3=8000 SWS=0000 +

01D2 70FF MDX -01 (01D2=70FF)

\*

Typing ‘S’ at the ‘\*’ prompt will display more registers. An example follows:

\*s

Current map = N0 N1 N2 N3, map registers = N0 N1 N2 N3

DSM : Lev 0, bit 0 IRQ = off, DSW = 0401, SubSts=A400, SubILSW=0000, Rsrv=0000

Timr: Lev 0, bit 5 IRQ = off, DSW = 0000, A=stopped, B=stopped, C=running

TTY : Lev 1, bit 1 IRQ = off, DSW = 0000, mode = Receive & echo

CI : Lev 1, bit 5 IRQ = off, DSW = 0000

Card: Lev 1, bit 3 IRQ = off, DSW = 0001

Prt0: Lev 1, bit 4 IRQ = off, DSW = 0000

Prt1: Lev 1, bit C IRQ = off, DSW = 0000

Tape: Lev 2, b 0-7 IRQ = off, DSW = 0000

MTCA: Lev 2, bit B IRQ = off, sub ILSW = 00000000, DSW's =

8000 8000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

ILSW: 0 = 0000 1 = 0000 2 = 0000 3 = 0000 4 = 0000 5 = 0000

CAR1(6)=0000 SCR1(7)=0000 CAR2(8)=0000 SCR2(9)=0000 CAR3(A)=0000

SCR3(B)=0000 CAR4(C)=C092 SCR4(D)=0000 CAR5(E)=0000 SCR5(F)=0000

Stop AUX=clsd MSK=00 PIR=0000 IRR=0000 ISR=0000 Lvl=255

ACC=1081 EXT=0040 XR1=1E95 XR2=0000 XR3=8000 SWS=0000 +

01D2 70FF MDX -01 (01D2=70FF)

Console data switches = 0000 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

 . . . . . . . . . . . . . . . .

\*

Typing ‘U’ will unassembled. An example follows:

\*u 1855

1855 4400 00F2 BSI L 00F2

1857 7400 00EE MDX L 00EE,00 (00EE=0000)

1859 70FD MDX -03 (1857=7400)

185A 6600 0000 LDX L2 0000

185C 2000 LDS 0

185D C8CE LDD -32 (182C=0000 0040)

185E 4C00 18C3 BSC L 18C3

1860 6AFA STX 2 -06 (185B=0000)

1861 D8CA STD -36 (182C=0000 0040)

1862 74FE 185F MDX L 185F,-02 (185F=18C3)

1864 6200 LDX 2 00

1865 700C MDX 0C (1872=28E9)

1866 6AF4 STX 2 -0C (185B=0000)

1867 6200 LDX 2 00

1868 7005 MDX 05 (186E=D8BD)

1869 6AF1 STX 2 -0F (185B=0000)

186A 6202 LDX 2 02

186B 7002 MDX 02 (186E=D8BD)

186C 6AEE STX 2 -12 (185B=0000)

186D 6204 LDX 2 04

\*

Below is an example of using the ‘T’ command. It is a trace of executing the $dump code. First I positioned at \0040 then I started pressing ‘t’ and Enter (to follow the source, check dna\_manual.txt APPENDIX D):

0040 D8D9 STD -27 (001A=1890 4400)

0041 4023 BSI 23 (0065=0025)

0042 282A STS (006D=2000)

0043 CC80 003F LDD I 003F (1234=0000 0000)

0045 69D1 STX 1 -2F (0017=C8FC)

0046 D0D2 STO -2E (0019=C0FC)

0047 C8F4 LDD -0C (003C=0001 0004)

0048 4400 00F2 BSI L 00F2

004A C0F1 LD -0F (003C=0001)

004B 7007 MDX 07 (0053=D0C2)

004C 0000

004D 0000

004E 0000

004F 0000

0050 0000

0051 0000

0052 C0EB LD -15 (003E=FFFF)

0053 D0C2 STO -3E (0016=04FE)

\*t

Trace AUX=clsd MSK=00 PIR=0000 IRR=0000 ISR=0000 Lvl=255

ACC=0050 EXT=7002 XR1=0000 XR2=0B18 XR3=007F SWS=0000 CE+

0041 4023 BSI 23 (0065=0025)

\*t

Trace AUX=clsd MSK=00 PIR=0000 IRR=0000 ISR=0000 Lvl=255

ACC=0050 EXT=7002 XR1=0000 XR2=0B18 XR3=007F SWS=0000 CE+

0066 C0CB LD -35 (0032=0000)

\*t

Trace AUX=clsd MSK=00 PIR=0000 IRR=0000 ISR=0000 Lvl=255

ACC=0000 EXT=7002 XR1=0000 XR2=0B18 XR3=007F SWS=0000 CEZ

0067 E8A9 OR -57 (0011=0000)

\*t

Trace AUX=clsd MSK=00 PIR=0000 IRR=0000 ISR=0000 Lvl=255

ACC=0000 EXT=7002 XR1=0000 XR2=0B18 XR3=007F SWS=0000 CEZ

0068 4C20 0066 BSC L 0066,Z

\*t

Trace AUX=clsd MSK=00 PIR=0000 IRR=0000 ISR=0000 Lvl=255

ACC=0000 EXT=7002 XR1=0000 XR2=0B18 XR3=007F SWS=0000 CEZ

006A 7000 MDX 00 (006B=4C80)

\*t

Trace AUX=clsd MSK=00 PIR=0000 IRR=0000 ISR=0000 Lvl=255

ACC=0000 EXT=7002 XR1=0000 XR2=0B18 XR3=007F SWS=0000 CEZ

006B 4C80 0065 BSC I 0065 (0042=282A)

\*t

Trace AUX=clsd MSK=00 PIR=0000 IRR=0000 ISR=0000 Lvl=255

ACC=0000 EXT=7002 XR1=0000 XR2=0B18 XR3=007F SWS=0000 CEZ

0042 282A STS (006D=2000)

\*t

Trace AUX=clsd MSK=00 PIR=0000 IRR=0000 ISR=0000 Lvl=255

ACC=0000 EXT=7002 XR1=0000 XR2=0B18 XR3=007F SWS=0000 EZ

0043 CC80 003F LDD I 003F (1234=0000 0000)

\*t

Trace AUX=clsd MSK=00 PIR=0000 IRR=0000 ISR=0000 Lvl=255

ACC=0000 EXT=0000 XR1=0000 XR2=0B18 XR3=007F SWS=0000 EZ

0045 69D1 STX 1 -2F (0017=C8FC)

\*

If the emulator encounters an invalid op/code it will enter the panel routine and print “Invalid op/code”. There is an example:

Invalid op/code AUX=clsd MSK=00 PIR=0000 IRR=0000 ISR=0000 Lvl=255

ACC=0050 EXT=7002 XR1=0000 XR2=0B18 XR3=007F SWS=0000 E+

1238 0000

\*

Typing ‘?’ will display a full list of commands:

\*?

Ctrl/F1 = Immediate STOP

Ctrl/F10 = Console Interrupt

Ctrl/S,Q = Pause/Continue TTY output

Ctrl/Enter= Ctrl/@ = NUL

B X N = Break at X after N passes (N = 1 if not given)

BC X = Clear breakpoint number X (BC \* to clear all breakpoints)

BL = Break list

C = Close all printers

C N = Close printer N

D A LB C = Dump starting at location A length B or to location C

E A B .. = Enter value B, etc. at location A

E = Enter values at next location

G XXXX = Go to XXXX (XXXX is optional)

Ix name = IPL from "name" (x: c=Cards, w=Write file, p=Platter, blank=reset)

L 0 | 1 | 2 | 3 = Log execution (0 = off, 1 = on, 2/3 = dump, 12 = real time)

Lx 1 | 12 = Log XIO's

L 1 xxxx = Log and break at loc xxxx

Lr A B = Log range is from A to B

Lc = Loc clear

O = Open all printers

O N = Open printer N (0=1403, 1=1132)

O N XXXX = Open printer N (0=1403, 1=1132) as XXXX

P = Proceed (does not trace through AUX or ISR)

Q = Quit

R = Registers

R XXX = Register XXX (IAR XR1 XR2 XR3 ACC EXT)

S = Status of devices

S XXXX = Switches are XXXX

T = Trace

U A LB C = Unassemble starting at location A length B or to location C

W name = Write core to disk file "name"

X = Execution time

Alt/d = Go to DOS

**\***