fred@cbmvax

JOB 959

C64DX_SYSTEM_SPEC_UPDATE_1

ur C64DX Prototype Package should include the following items:

        C64DX computer
        ~~C64DX power supply~~
        9-pin D to 9-pin D (PC-style) Monitor cable (fits 1084S-P1/D1)
        C64DX Demo Disk #2
        C64DX SYSTEM SPECIFICATION (Wrapped in sanitary green band)
        C64DX SYSTEM_SPEC_UPDATE_1

ollow these instructions to hook up your C64DX System:

        Place the computer on a flat surface, within reach of a
        Power Outlet. Turn OFF all power switches on the computer and
        monitor before attempting to connect the Cables.

        The computer requires an RGB Amiga-style Monitor. We recommend
        the Commodore Model 1084S-P1 or 1084S-D1, which has a 9-pin D
        connector on the back of the unit. To connect the video cable,
        insert the 9-pin D connector into the port labeled RGB VIDEO on
        the back of the C64DX computer. Then insert the other end of
        the cable into the port labeled RGB on the back of the 1084S.

        The 1084S must then be set to RGB mode using the switches on the
        back of the monitor. Set the ANALOG/TTL switch to analog mode.
        Set the CVBS/RGB switch to RGB mode.

        To connect the power supply, insert the 4-pin circular DIN connector
        on the power cable into the port labeled POWER on the left side of
        the C64DX computer. Then insert the other end of the power cable
        into a wall outlet.

        Insert the C64DX DEMO DISK #2 into the Floppy drive. Turn ON the
        1084S Monitor. Now turn ON the C64DX computer, and the DEMO will
        automatically LOAD and RUN. Loading messages will appear on the
        screen, and the system will display several graphics pictures.
        ADDITIONAL graphics will be shown if a RAM EXPANDER is installed.

        The DEMO can be stopped by pressing the STOP key. It can be re-run
        by typing "BOOT", or by pressing the RESET button on the left side
        of the computer, as long as the DEMO DISK is still inserted in the
        Floppy Drive.

F YOU HAVE ANY PROBLEMS OR QUESTIONS, PLEASE CONTACT COMMODORE ENGINEERING
N WEST CHESTER, PA, USA.   attn: PAUL LASSA, GREG BERLIN, JEFF PORTER.

PLEASE NOTE: THIS C64DX COMPUTER IS A PRE-PRODUCTION ENGINEERING PROTOTYPE.

        Design and Testing is not yet completed, and as such, the unit may
        experience problems after being run for extended periods of time.
        Normal operation can be restored by pressing the RESET button, or
        allowing the unit to cool with the power OFF.

        RF and Composite VIDEO are not working with this release. GERMAN-
        PRINTED KEYBOARDS are not yet available, but the C64DX recognizes
        the GERMAN key sequences when CAPS/LOCK is ON. (see SPEC_UPDATE_1)

The following changes/updates/fixes have been made to the C64DX ROM
code since the March 1, 1991 C64DX System Specification was printed.
Please make note of them. Current ROM as of this update is 910501.

CHAR   Now works to spec and supports the following imbedded
                 control characters (although some are buggy; others
                 are planned):

                 ^F   6   flip
                 ^I   9   invert
                 ^O   15   overwrite
                 ^R   18   reverse field on
                         146   reverse field off
                 ^U   21   underline
                 ^Y   25   tilt
                 ^Z   26   mirror

                 When specifying a character set from ROM, note that
                 national versions of the C64DX will have the national
                 character set at $39000 and the C64 character set at
                 $3D000. In US/English systems, the default C64DX-mode
                 character set will be at $39000.

CLR ERR$ Clears BASIC error stuff, useful after a TRAP

CURSOR [<ON|OFF>,] [column] [,row] [,style]
        where: column,row = x,y logical screen position
                style   = flashing (0) or solid (1)
                ON,OFF   = to turn the cursor on or off

LINE   x0, y0 [,[x1] [,y1]]...
        where: (x1,y1)=(x0,y0) if not specified, drawing a dot.
                Additional coordinates (x2,y2), etc. draw a line from
                the previous point.

LOADIFF "file" [,U#,D#]
                Loads an IFF picture from disk. Requires a suitable
                graphic screen to be already opened (this may change).
                The file must contain std IFF data in PRG file type.
                IFF pics can be ported directly from Amiga (eg., using
                XMODEM). Returns 'File Data Error' if it finds data
                it does not like.

MOD (number, modulus)
                New function.

MOUSE ON [,[port] [,[sprite] [,[hotspot] [,X/Yposition]]]]
MOUSE OFF
        where: port   = (1...3) for joyport 1, 2, or either (both)
                sprite   = (0...7) sprite pointer
                hotspot = x,y offset in sprite, default 0,0
                position = normal, relative, or angluar coordinates
                Defaults to sprite 0, port 2, last hotspot (0,0), and
                position. Kernel doesn't let hotspot leave the screen.

PAINT x, y [,color]
                    Working, but not completely to spec. Uses draw pen
                    color and fills emptyness to any border.

RND(0)   Improved for better "randomness".....Uses unused POT
                    of second SID chip. PCB must allow lines to float.

SET DISK #   (without [TO #] parameter) allows user to clear DS$
                    message and specify which drive next DS$ comes from.

SET VERIFY <ON|OFF>
                    The new DOS65 defaults to verify-after-write OFF.
                    This command works with 1581 drive, too.

* Negative Coordinates are now allowed for all graphics commands. Some
  commands require their arguments to be "onscreen", such as PAINT.

* BASIC errors now force text mode, and TYPE, LIST, DISK, KEYLOAD,
  LOADIFF now catch all DOS errors. Autoboot filename= AUTOBOOT.C64DX.*

* Opening an RS-232 channel, command string allows setting new features:

            1   baud (0-16, where 16=MIDI rate)
            2   word len
            3   parity
            4   stop bits (not used)
            5   duplex
            6   xline
            7   xon char (0=incoming flow control disabled)
            8   xoff char (0=outgoing flow control disabled)
            9,10   input buffer pointer (page lo, hi)
            11,12   output buffer pointer (page lo, hi)
            13   high water mark (point at which xoff is xmitted)
            14   low water mark (point at which xon is xmitted)

  For debug purposes, the border color will change if an RS232 buffer
  overflow occurs. To differentiate between a GET# of a null and a
  'no data' null, test bit 3 of STatus (same as C64).

* Support for latest DOS controller chip, F011D, includes error LED
  blink (border color still changes too, for now). Changes to improve
  FASTLOAD speed and improve SAVE speed. Will work with F011C chip,
  but error LED does not blink. Requires latest 'ELMER' PAL for disk
  LED to work correctly for either controller chip. External drive LED
  will not work correctly until new PCB & F016 chip are designed. New
  DOS functions include COPY D0 TO D1, ability to change sector skews
  for files (U0>S#) and directory (U0>D#), and directory compress
  (i.e., empty trash) via "E" command. Physical interleave is now 7.

* The DOS COPY/CONCAT bugs have been fixed, and COPY now allows forms
  such as COPY D0,"*.SRC" TO D1,"*" and COPY D0,"*=SEQ" TO D1,"*".
  Directory/partition paths not yet implemented, but will be.

*   The Monitor parser now allows PETSCII input/conversion:

```
'A                           prints ASC() value of character
>1800 'text                  puts text into memory
LDA #'A
```

*   IRQ runs during graphics (Kernel finds its own base page). IRQ
    still does not run during DOS activity (not sure if they ever will).

*   The following Kernel Jump Table Entries have moved (and are still
    subject to further changes):

```
FF05     nirq              ;IRQ handler
FF07     monitor_brk       ;BRK handler (Monitor)
FF09     nnmi              ;NMI handler
FF0B     nopen             ;open
FF0D     nclose            ;close
FF0F     nchkin            ;chkin
FF11     nckout            ;ckout
FF13     nclrch            ;clrch
FF15     nbasin            ;basin
FF17     nbsout            ;bsout
FF19     nstop             ;stop key scan
FF1B     ngetin            ;getin
FF1D     nclall            ;clall
FF1F     monitor_parser    ;monitor command parser
FF21     nload             ;load
FF23     nsave             ;save

FF25     talk
FF27     listen
FF29     talksa
FF2B     second
FF2D     acptr
FF2F     ciout
FF31     untalk
FF33     unlisten

FF35     DOS_talk
FF37     DOS_listen
FF39     DOS_talksa
FF3B     DOS_second
FF3D     DOS_acptr
FF3F     DOS_ciout
FF41     DOS_untalk
FF43     DOS_unlisten

FF45     Get_DOS
FF47     Leave_DOS
FF49     ColdStartDOS           <<< new
FF4B     WarmStartDOS           <<< new
```