```
**************************************************************************
**************************************************************************
**                                                                      **
**      OSWRCH  MAIN ROUTINE  entry from E0C5                           **
**                                                                      **
**      output a byte via the VDU stream                                **
**                                                                      **
**************************************************************************
**************************************************************************
;this routine takes up over 40% of the operating system ROM
;entry points are variable, as are the results achieved.
;tracing any particular path is relatively easy but generalising for
;commenting is not.  For clarity comments will not be as detailed as
;for later parts of the Operating system.

C4C0    LDX     &026A   ;get number of items in VDU queque
C4C3    BNE     &C512   ;if parameters needed then C512
C4C5    BIT     &D0     ;else check status byte
C4C7    BVC     &C4D8   ;if cursor editing enabled two cursors exist
C4C9    JSR     &C568   ;swap values
C4CC    JSR     &CD6A   ;then set up write cursor
C4CF    BMI     &C4D8   ;if display disabled C4D8
C4D1    CMP     #&0D    ;else if character in A=RETURN teminate edit
C4D3    BNE     &C4D8   ;else C4D8

C4D5    JSR     &D918   ;terminate edit

C4D8    CMP     #&7F    ;is character DELETE ?
C4DA    BEQ     &C4ED   ;if so C4ED

C4DC    CMP     #&20    ;is it less than space? (i.e. VDU control code)
C4DE    BCC     &C4EF   ;if so C4EF
C4E0    BIT     &D0     ;else check VDU byte ahain
C4E2    BMI     &C4EA   ;if screen disabled C4EA
C4E4    JSR     &CFB7   ;else display a character
C4E7    JSR     &C664   ;and cursor right
C4EA    JMP     &C55E   ;

********* read link addresses and number of parameters *****************

C4ED    LDA     #&20    ;to replace delete character

********* read link addresses and number of parameters *****************

C4EF    TAY             ;Y=A
C4F0    LDA     &C333,Y ;get lo byte of link address
C4F3    STA     &035D   ;store it in jump vector
C4F6    LDA     &C354,Y ;get hi byte
C4F9    BMI     &C545   ;if negative (as it will be if a direct address)
                        ;there are no parameters needed
                        ;so C545
C4FB    TAX             ;else X=A
C4FC    ORA     #&F0    ;set up negated parameter count
C4FE    STA     &026A   ;store it as number of items in VDU queque
C501    TXA             ;get back A
C502    LSR             ;A=A/16
C503    LSR             ;
C504    LSR             ;
C505    LSR             ;
C506    CLC             ;clear carry
C507    ADC     #&C3    ;add &C3 to get hi byte of link address
C509    STA     &035E   ;
C50C    BIT     &D0     ;check if cursor editing enabled
C50E    BVS     &C52F   ;if so re-exchange pointers
C510    CLC             ;clear carry
C511    RTS             ;and exit
```

```
;return with carry clear indicates that printer action not required.

;
********** parameters are outstanding **********************************
X=&26A = 2 complement of number of parameters X=&FF for 1, FE for 2 etc.

C512    STA     &0224,X ;store parameter in queque
C515    INX             ;increment X
C516    STX     &026A   ;store it as VDU queque
C519    BNE     &C532   ;if not 0 C532 as more parameters are needed
C51B    BIT     &D0     ;get VDU status byte
C51D    BMI     &C534   ;if screen disabled C534
C51F    BVS     &C526   ;else if cursor editing C526
C521    JSR     &CCF5   ;execute required function
C524    CLC             ;clear carry
C525    RTS             ;and exit
;
C526    JSR     &C568   ;swap values of cursors
C529    JSR     &CD6A   ;set up write cursor
C52C    JSR     &CCF5   ;execute required function
C52F    JSR     &C565   ;re-exchange pointers

C532    CLC             ;carry clear
C533    RTS             ;exit




*************************************************************************
*                                                                       *
*       VDU 1 send next character to printer only                       *
*                                                                       *
*       1 parameter required                                            *
*                                                                       *
*************************************************************************
;
C534    LDY     &035E   ;if upper byte of link address not &C5
C537    CPY     #&C5    ;printer is not interested
C539    BNE     &C532   ;so C532
C53B    TAX             ;else X=A
C53C    LDA     &D0     ;A=VDU status byte
C53E    LSR             ;get bit 0 into carry
C53F    BCC     &C511   ;if printer not enabled exit
C541    TXA             ;restore A
C542    JMP     &E11E   ;else send byte in A (next byte) to printer



*********** if explicit link address found, no parameters ***************

C545    STA     &035E   ;upper byte of link address
C548    TYA             ;restore A
C549    CMP     #&08    ;is it 7 or less?
C54B    BCC     &C553   ;if so C553
C54D    EOR     #&FF    ;invert it
C54F    CMP     #&F2    ;c is set if A >&0D
C551    EOR     #&FF    ;re invert

C553    BIT     &D0     ;VDU status byte
C555    BMI     &C580   ;if display disabled C580
C557    PHP             ;push processor flags
C558    JSR     &CCF5   ;execute required function
C55B    PLP             ;get back flags
C55C    BCC     &C561   ;if carry clear (from C54B/F)
```

```
*************** main exit routine ************************************

C55E    LDA     &D0     ;VDU status byte
C560    LSR             ;Carry is set if printer is enabled
C561    BIT     &D0     ;VDU status byte
C563    BVC     &C511   ;if nmo cursor editing  C511 to exit


***************** cursor editing routines ******************************

C565    JSR     &CD7A   ;restore normal write cursor

C568    PHP             ;save flags and
C569    PHA             ;A
C56A    LDX     #&18    ;X=&18
C56C    LDY     #&64    ;Y=&64
C56E    JSR     &CDDE   ;exchange &300/1+X with &300/1+Y
C571    JSR     &CF06   ;set up display address
C574    JSR     &CA02   ;set cursor position
C577    LDA     &D0     ;VDU status byte
C579    EOR     #&02    ;invert bit 1 to allow or bar scrolling
C57B    STA     &D0     ;VDU status byte
C57D    PLA             ;restore flags and A
C57E    PLP             ;
C57F    RTS             ;and exit
;
C580    EOR     #&06    ;if A<>6
C582    BNE     &C58C   ;return via C58C
C584    LDA     #&7F    ;A=&7F
C586    BCC     &C5A8   ;and goto C5A8 ALWAYS!!


****************** check text cursor in use ***************************

C588    LDA     &D0     ;VDU status byte
C58A    AND     #&20    ;set A from bit 5 of status byte
C58C    RTS             ;and exit

A=0 if text cursor, &20 if graphics


*************************************************************************
*                                                                       *
*                                                                       *
*       SET PAGED MODE  VDU 14                                          *
*                                                                       *
*                                                                       *
*************************************************************************

;
C58D    LDY     #&00    ;Y=0
C58F    STY     &0269   ;paged mode counter
C592    LDA     #&04    ;A=04
C594    BNE     &C59D   ;jump to C59D



*************************************************************************
*                                                                       *
*                                                                       *
*       VDU 2 PRINTER ON                                                *
```

```
*                                                                       *
*                                                                       *
*************************************************************************


C596    JSR     &E1A2   ;select printer buffer  and output character
C599    LDA     #&94    ;A=&94
                        ;when inverted at C59B this =1


*************************************************************************
*                                                                       *
*                                                                       *
*       DISABLE DISPLAY VDU 21                                          *
*       no parameters                                                   *
*                                                                       *
*************************************************************************


C59B    EOR     #&95    ;if A=&15 A now =&80: if A=&94 A now =1

C59D    ORA     &D0     ;VDU status byte set bit 0 or bit 7
C59F    BNE     &C5AA   ;



*************************************************************************
*                                                                       *
*       VDU 3 Printer off                                               *
*                                                                       *
*       No parameters                                                   *
*                                                                       *
*************************************************************************


C5A1    JSR     &E1A2   ;select printer buffer  and output character
C5A4    LDA     #&0A    ;A=10



*************************************************************************
*                                                                       *
*                                                                       *
*       VDU 15 paged mode off   No parameters                           *
*                                                                       *
*                                                                       *
*************************************************************************
A=&F or &A

C5A6    EOR     #&F4    ;convert to &FB or &FE
C5A8    AND     &D0     ;VDU status byte clear bit 0 or bit 2 of status
C5AA    STA     &D0     ;VDU status byte
C5AC    RTS             ;exit



*************************************************************************
*                                                                       *
*                                                                       *
*       VDU 4 select Text Cursor  No parameters                         *
*                                                                       *
*                                                                       *
```

```
*************************************************************************
;
C5AD    LDA     &0361   ;pixels per byte
C5B0    BEQ     &C5AC   ;if no graphics in current mode C5AC
C5B2    JSR     &C951   ;set CRT controller for text cursor
C5B5    LDA     #&DF    ;this to clear bit 5 of status byte
C5B7    BNE     &C5A8   ;via C5A8 exit



*************************************************************************
*                                                                       *
*                                                                       *
*       VDU 5 set graphics cursor                                       *
*                                                                       *
*                                                                       *
*************************************************************************

C5B9    LDA     &0361   ;pixels per byte
C5BC    BEQ     &C5AC   ;if none this is text mode so exit
C5BE    LDA     #&20    ;set up graphics cursor
C5C0    JSR     &C954   ;via C954
C5C3    BNE     &C59D   ;set bit 5 via exit C59D



*************************************************************************
*                                                                       *
*                                                                       *
*       VDU 8  CURSOR LEFT     NO PARAMETERS                            *
*                                                                       *
*                                                                       *
*************************************************************************


C5C5    JSR     &C588   ;A=0 if text cursor A=&20 if graphics cursor
C5C8    BNE     &C61F   ;move cursor left 8 pixels if graphics
C5CA    DEC     &0318   ;else decrement text column
C5CD    LDX     &0318   ;store new text column
C5D0    CPX     &0308   ;if it is less than text window left
C5D3    BMI     &C5EE   ;do wraparound  cursor to rt of screen 1 line up
C5D5    LDA     &034A   ;text cursor 6845 address
C5D8    SEC             ;subtract
C5D9    SBC     &034F   ;bytes per character
C5DC    TAX             ;put in X
C5DD    LDA     &034B   ;get text cursor 6845 address
C5E0    SBC     #&00    ;subtract 0
C5E2    CMP     &034E   ;compare with hi byte of screen RAM address
C5E5    BCS     &C5EA   ;if = or greater
C5E7    ADC     &0354   ;add screen RAM size hi byte to wrap around
C5EA    TAY             ;Y=A
C5EB    JMP     &C9F6   ;Y hi and X lo byte of cursor position


**************** execute wraparound left-up*****************************

C5EE    LDA     &030A   ;text window right
C5F1    STA     &0318   ;text column


*************** cursor up ***************************************************
```

```
C5F4    DEC     &0269   ;paged mode counter
C5F7    BPL     &C5FC   ;if still greater than 0 skip next instruction
C5F9    INC     &0269   ;paged mode counter to restore X=0
C5FC    LDX     &0319   ;current text line
C5FF    CPX     &030B   ;top of text window
C602    BEQ     &C60A   ;if its at top of window C60A
C604    DEC     &0319   ;else decrement current text line
C607    JMP     &C6AF   ;and carry on moving cursor


******** cursor at top of window ***************************************

C60A    CLC             ;clear carry
C60B    JSR     &CD3F   ;check for window violatations
C60E    LDA     #&08    ;A=8 to check for software scrolling
C610    BIT     &D0     ;compare against VDU status byte
C612    BNE     &C619   ;if not enabled C619
C614    JSR     &C994   ;set screen start register and adjust RAM
C617    BNE     &C61C   ;jump C61C

C619    JSR     &CDA4   ;soft scroll 1 line
C61C    JMP     &C6AC   ;and exit


**********cursor left and down with graphics cursor in use **************

C61F    LDX     #&00    ;X=0 to select horizontal parameters


********** cursor down with graphics in use ****************************
;X=2 for vertical or 0 for horizontal

C621    STX     &DB     ;store X
C623    JSR     &D10D   ;check for window violations
C626    LDX     &DB     ;restore X
C628    SEC             ;set carry
C629    LDA     &0324,X ;current graphics cursor X>1=vertical
C62C    SBC     #&08    ;subtract 8 to move back 1 character
C62E    STA     &0324,X ;store in current graphics cursor X>1=verticaal
C631    BCS     &C636   ;if carry set skip next
C633    DEC     &0325,X ;current graphics cursor hi -1
C636    LDA     &DA     ;&DA=0 if no violation else 1 if vert violation
                        ;2 if horizontal violation
C638    BNE     &C658   ;if violation C658
C63A    JSR     &D10D   ;check for window violations
C63D    BEQ     &C658   ;if none C658

C63F    LDX     &DB     ;else get back X
C641    LDA     &0304,X ;graphics window rt X=0 top X=2
C644    CPX     #&01    ;is X=0
C646    BCS     &C64A   ;if not C64A
C648    SBC     #&06    ;else subtract 7

C64A    STA     &0324,X ;current graphics cursor X>1=vertical
C64D    LDA     &0305,X ;graphics window hi rt X=0 top X=2
C650    SBC     #&00    ;subtract carry
C652    STA     &0325,X ;current graphics cursor X<2=horizontal else vertical
C655    TXA             ;A=X
C656    BEQ     &C660   ;cursor up
C658    JMP     &D1B8   ;set up external coordinates for graphics
```

```
**************************************************************************
*                                                                        *
*                                                                        *
*       VDU 11 Cursor Up    No Parameters                                *
*                                                                        *
*                                                                        *
**************************************************************************

C65B    JSR     &C588   ;A=0 if text cursor A=&20 if graphics cursor
C65E    BEQ     &C5F4   ;if text cursor then C5F4
C660    LDX     #&02    ;else X=2
C662    BNE     &C6B6   ;goto C6B6




**************************************************************************
*                                                                        *
*                                                                        *
*       VDU 9 Cursor right    No parameters                              *
*                                                                        *
*                                                                        *
**************************************************************************

C664    LDA     &D0     ;VDU status byte
C666    AND     #&20    ;check bit 5
C668    BNE     &C6B4   ;if set then graphics cursor in use so C6B4
C66A    LDX     &0318   ;text column
C66D    CPX     &030A   ;text window right
C670    BCS     &C684   ;if X exceeds window right then C684
C672    INC     &0318   ;text column
C675    LDA     &034A   ;text cursor 6845 address
C678    ADC     &034F   ;add bytes per character
C67B    TAX             ;X=A
C67C    LDA     &034B   ;text cursor 6845 address
C67F    ADC     #&00    ;add carry if set
C681    JMP     &C9F6   ;use X and Y to set new cursor address

********: text cursor down and right ************************************

C684    LDA     &0308   ;text window left
C687    STA     &0318   ;text column

********: text cursor down ***********************************

C68A    CLC             ;clear carry
C68B    JSR     &CAE3   ;check bottom margin, X=line count
C68E    LDX     &0319   ;current text line
C691    CPX     &0309   ;bottom margin
C694    BCS     &C69B   ;if X=>current bottom margin C69B
C696    INC     &0319   ;else increment current text line
C699    BCC     &C6AF   ;
C69B    JSR     &CD3F   ;check for window violations
C69E    LDA     #&08    ;check bit 3
C6A0    BIT     &D0     ;VDU status byte
C6A2    BNE     &C6A9   ;if software scrolling enabled C6A9
C6A4    JSR     &C9A4   ;perform hardware scroll
C6A7    BNE     &C6AC   ;
C6A9    JSR     &CDFF   ;execute upward scroll
C6AC    JSR     &CEAC   ;clear a line

C6AF    JSR     &CF06   ;set up display address
C6B2    BCC     &C732   ;
```

```
********** graphic cursor right ***************************************

C6B4    LDX     #&00    ;


************** graphic cursor up  (X=2) *********************************

C6B6    STX     &DB     ;store X
C6B8    JSR     &D10D   ;check for window violations
C6BB    LDX     &DB     ;get back X
C6BD    CLC             ;clear carry
C6BE    LDA     &0324,X ;current graphics cursor X>1=vertical
C6C1    ADC     #&08    ;Add 8 pixels
C6C3    STA     &0324,X ;current graphics cursor X>1=vertical
C6C6    BCC     &C6CB   ;
C6C8    INC     &0325,X ;current graphics cursor X<2=horizontal else vertical
C6CB    LDA     &DA     ;A=0 no window violations 1 or 2 indicates violation
C6CD    BNE     &C658   ;if outside window C658
C6CF    JSR     &D10D   ;check for window violations
C6D2    BEQ     &C658   ;if no violations C658

C6D4    LDX     &DB     ;get back X
C6D6    LDA     &0300,X ;graphics window X<2 =left else bottom
C6D9    CPX     #&01    ;If X=0
C6DB    BCC     &C6DF   ;C6DF
C6DD    ADC     #&06    ;else add 7
C6DF    STA     &0324,X ;current graphics cursor X>1=vertical
C6E2    LDA     &0301,X ;graphics window hi X<2 =left else bottom
C6E5    ADC     #&00    ;add anny carry
C6E7    STA     &0325,X ;current graphics cursor X<2=horizontal else vertical
C6EA    TXA             ;A=X
C6EB    BEQ     &C6F5   ;if X=0 C6F5 cursor down
C6ED    JMP     &D1B8   ;set up external coordinates for graphics




*************************************************************************
*                                                                       *
*                                                                       *
*       VDU 10  Cursor down     No parameters                           *
*                                                                       *
*                                                                       *
*************************************************************************

C6F0    JSR     &C588   ;A=0 if text cursor A=&20 if graphics cursor
C6F3    BEQ     &C68A   ;if text cursor back to C68A
C6F5    LDX     #&02    ;else X=2 to indicate vertical movement
C6F7    JMP     &C621   ;move graphics cursor down




*************************************************************************
*                                                                       *
*                                                                       *
*       VDU 28   define text window         4 parameters                *
*                                                                       *
*                                                                       *
*************************************************************************
;parameters are set up thus
;0320  P1 left margin
;0321  P2 bottom margin
```

```
;0322  P3 right margin
;0323  P4 top margin
;Note that last parameter is always in 0323

C6FA    LDX     &0355   ;screen mode
C6FD    LDA     &0321   ;get bottom margin
C700    CMP     &0323   ;compare with top margin
C703    BCC     &C758   ;if bottom margin exceeds top return
C705    CMP     &C3E7,X ;text window bottom margin maximum
C708    BEQ     &C70C   ;if equal then its OK
C70A    BCS     &C758   ;else exit

C70C    LDA     &0322   ;get right margin
C70F    TAY             ;put it in Y
C710    CMP     C3EF,X  ;text window right hand margin maximum
C713    BEQ     &C717   ;if equal then OK
C715    BCS     &C758   ;if greater than maximum exit

C717    SEC             ;set carry to subtract
C718    SBC     &0320   ;left margin
C71B    BMI     &C758   ;if left greater than right exit
C71D    TAY             ;else A=Y (window width)
C71E    JSR     &CA88   ;calculate number of bytes in a line
C721    LDA     #&08    ;A=8 to set bit  of &D0
C723    JSR     &C59D   ;indicating that text window is defined
C726    LDX     #&20    ;point to parameters
C728    LDY     #&08    ;point to text window margins
C72A    JSR     &D48A   ;(&300/3+Y)=(&300/3+X)
C72D    JSR     &CEE8   ;set up screen address
C730    BCS     &C779   ;home cursor within window
C732    JMP     &CA02   ;set cursor position



*************************************************************************
*                                                                       *
*                                                                       *
*       OSWORD 9    read a pixel                                        *
*                                                                       *
*                                                                       *
*************************************************************************
;on entry &EF=A=9
;         &F0=X=low byte of parameter block address
;         &F1=Y=high byte of parameter block address
;       PARAMETER BLOCK
;bytes 0,1 X coordinate, bytes 2,3 Y coordinate
;EXIT with result in byte 4 =&FF if point was of screen or logical colour
;     of point if on screen

C735    LDY     #&03    ;Y=3 to point to hi byte of Y coordinate
C737    LDA     (&F0),Y ;get it
C739    STA     &0328,Y ;store it
C73C    DEY             ;point to next byte
C73D    BPL     &C737   ;transfer till Y=&FF lo byte of X coordinate in &328
C73F    LDA     #&28    ;
C741    JSR     &D839   ;check window boundaries
C744    LDY     #&04    ;Y=4
C746    BNE     &C750   ;jump to C750



*************************************************************************
*                                                                       *
*                                                                       *
```

```
*       OSWORD 11       read pallette                                   *
*                                                                       *
*                                                                       *
************************************************************************
;on entry &EF=A=11
;         &F0=X=low byte of parameter block address
;         &F1=Y=high byte of parameter block address
;       PARAMETER BLOCK
;bytes 0,logical colour to read
;EXIT with result in  4 bytes:-0 logical colour,1 physical colour
;     2,3 both 0.  corresponds to reading VDU 19

C748    AND     &0360   ;number of logical colours less 1
C74B    TAX             ;put it in X
C74C    LDA     &036F,X ;colour pallette
C74F    INY             ;increment Y to point to byte 1

C750    STA     (&F0),Y ;store data
C752    LDA     #&00    ;issue 0s
C754    CPY     #&04    ;to next bytes until Y=4
C756    BNE     &C74F   ;

C758    RTS             ;and exit



************************************************************************
*                                                                       *
*                                                                       *
*       VDU 12  Clear text Screen               0 parameters            *
*                                                                       *
*                                                                       *
************************************************************************
;
C759    JSR     &C588   ;A=0 if text cursor A=&20 if graphics cursor
C75C    BNE     &C7BD   ;if graphics cursor &C7BD
C75E    LDA     &D0     ;VDU status byte
C760    AND     #&08    ;check if software scrolling (text window set)
C762    BNE     &C767   ;if so C767
C764    JMP     &CBC1   ;initialise screen display and home cursor

C767    LDX     &030B   ;top of text window
C76A    STX     &0319   ;current text line
C76D    JSR     &CEAC   ;clear a line

C770    LDX     &0319   ;current text line
C773    CPX     &0309   ;bottom margin
C776    INX             ;X=X+1
C777    BCC     &C76A   ;if X at compare is less than bottom margin clear next



************************************************************************
*                                                                       *
*                                                                       *
*       VDU 30  Home cursor                     0  parameters           *
*                                                                       *
*                                                                       *
************************************************************************

C779    JSR     &C588   ;A=0 if text cursor A=&20 if graphics cursor
C77C    BEQ     &C781   ;if text cursor C781
C77E    JMP     &CFA6   ;home graphic cursor if graphic
C781    STA     &0323   ;store 0 in last two parameters
C784    STA     &0322   ;
```

```
************************************************************************
*                                                                      *
*                                                                      *
*       VDU 31  Position text cursor            2  parameters          *
*                                                                      *
*                                                                      *
************************************************************************
;0322 = X coordinate
;0323 = Y coordinate

C787    JSR     &C588   ;A=0 if text cursor A=&20 if graphics cursor
C78A    BNE     &C758   ;exit
C78C    JSR     &C7A8   ;exchange text column/line with workspace 0328/9
C78F    CLC             ;clear carry
C790    LDA     &0322   ;get X coordinate
C793    ADC     &0308   ;add to text window left
C796    STA     &0318   ;store as text column
C799    LDA     &0323   ;get Y coordinate
C79C    CLC             ;
C79D    ADC     &030B   ;add top of text window
C7A0    STA     &0319   ;current text line
C7A3    JSR     &CEE8   ;set up screen address
C7A6    BCC     &C732   ;set cursor position if C=0 (point on screen)
C7A8    LDX     #&18    ;else point to workspace
C7AA    LDY     #&28    ;and line/column to restore old values
C7AC    JMP     &CDDE   ;exchange &300/1+X with &300/1+Y



************************************************************************
*                                                                      *
*                                                                      *
*       VDU  13         Carriage  Return        0 parameters           *
*                                                                      *
*                                                                      *
************************************************************************


C7AF    JSR     &C588   ;A=0 if text cursor A=&20 if graphics cursor
C7B2    BEQ     &C7B7   ;if text C7B7
C7B4    JMP     &CFAD   ;else set graphics cursor to left hand columm

C7B7    JSR     &CE6E   ;set text column to left hand column
C7BA    JMP     &C6AF   ;set up cursor and display address

C7BD    JSR     &CFA6   ;home graphic cursor



************************************************************************
*                                                                      *
*                                                                      *
*       VDU 16 clear graphics screen            0 parameters           *
*                                                                      *
*                                                                      *
************************************************************************

C7C0    LDA     &0361   ;pixels per byte
```

```
C7C3    BEQ     &C7F8   ;if 0 current mode has no graphics so exit
C7C5    LDX     &035A   ;Background graphics colour
C7C8    LDY     &035C   ;background graphics plot mode (GCOL n)
C7CB    JSR     &D0B3   ;set graphics byte mask in &D4/5
C7CE    LDX     #&00    ;graphics window
C7D0    LDY     #&28    ;workspace
C7D2    JSR     &D47C   ;set(300/7+Y) from (300/7+X)
C7D5    SEC             ;set carry
C7D6    LDA     &0306   ;graphics window top lo.
C7D9    SBC     &0302   ;graphics window bottom lo
C7DC    TAY             ;Y=difference
C7DD    INY             ;increment
C7DE    STY     &0330   ;and store in workspace (this is line count)
C7E1    LDX     #&2C    ;
C7E3    LDY     #&28    ;
C7E5    JSR     &D6A6   ;clear line
C7E8    LDA     &032E   ;decrement window height in pixels
C7EB    BNE     &C7F0   ;
C7ED    DEC     &032F   ;
C7F0    DEC     &032E   ;
C7F3    DEC     &0330   ;decrement line count
C7F6    BNE     &C7E1   ;if <>0 then do it again
C7F8    RTS             ;exit



*************************************************************************
*                                                                       *
*                                                                       *
*       VDU 17          Define text colour              1 parameter     *
*       COLOUR                                                          *
*                                                                       *
*************************************************************************
;parameter in &0323

C7F9    LDY     #&00    ;Y=0
C7FB    BEQ     &C7FF   ;jump to C7FF



*************************************************************************
*                                                                       *
*                                                                       *
*       VDU 18          Define graphics colour          2 parameters    *
*       GCOL                                                            *
*                                                                       *
*************************************************************************
;parameters in 323,322

C7FD    LDY     #&02    ;Y=2

C7FF    LDA     &0323   ;get last parameter
C802    BPL     &C805   ;if +ve its foreground colour so C805
C804    INY             ;else Y=Y+1

C805    AND     &0360   ;number of logical colours less 1
C808    STA     &DA     ;store it
C80A    LDA     &0360   ;number of logical colours less 1
C80D    BEQ     &C82B   ;if none exit
C80F    AND     #&07    ;else limit to an available colour and clear M
C811    CLC             ;clear carry
C812    ADC     &DA     ;Add last parameter to get pointer to table
C814    TAX             ;pointer into X
C815    LDA     &C423,X ;get plot options from table
```

```
C818    STA     &0357,Y ; colour Y=0=text fgnd 1= text bkgnd 2=graphics fg etc
C81B    CPY     #&02    ;If Y>1
C81D    BCS     &C82C   ;then its graphics so C82C else
C81F    LDA     &0357   ;foreground text colour
C822    EOR     #&FF    ;invert
C824    STA     &D3     ;text colour byte to be orred or EORed into memory
C826    EOR     &0358   ;background text colour
C829    STA     &D2     ;text colour byte to be orred or EORed into memory

C82B    RTS             ;and exit
;
C82C    LDA     &0322   ;get first parameter
C82F    STA     &0359,Y ;text colour Y=0=foreground 1=background etc.
C832    RTS             ;exit



;
C833    LDA     #&20    ;
C835    STA     &0358   ;background text colour
C838    RTS             ;



**************************************************************************
*                                                                        *
*                                                                        *
*       VDU 20          Restore default colours         0 parameters    *
*                                                                        *
*                                                                        *
**************************************************************************
;
C839    LDX     #&05    ;X=5

C83B    LDA     #&00    ;A=0
C83D    STA     &0357,X ;zero all colours
C840    DEX             ;
C841    BPL     &C83D   ;until X=&FF
C843    LDX     &0360   ;number of logical colours less 1
C846    BEQ     &C833   ;if none its MODE 7 so C833
C848    LDA     #&FF    ;A=&FF
C84A    CPX     #&0F    ;if not mode 2 (16 colours)
C84C    BNE     &C850   ;goto C850

C84E    LDA     #&3F    ;else A=&3F

C850    STA     &0357   ;foreground text colour
C853    STA     &0359   ;foreground graphics colour
C856    EOR     #&FF    ;invert A
C858    STA     &D2     ;text colour byte to be orred or EORed into memory
C85A    STA     &D3     ;text colour byte to be orred or EORed into memory
C85C    STX     &031F   ;set first parameter of 5
C85F    CPX     #&03    ;if there are 4 colours
C861    BEQ     &C874   ;goto C874
C863    BCC     &C885   ;if less there are 2 colours goto C885

                        ;else there are 16 colours
C865    STX     &0320   ;set second parameter
C868    JSR     &C892   ;do VDU 19 etc
C86B    DEC     &0320   ;decrement first parameter
C86E    DEC     &031F   ;and last parameter
C871    BPL     &C868   ;
C873    RTS             ;
;
********* 4 colour mode ************************************************
```

```
C874    LDX     #&07    ;X=7
C876    STX     &0320   ;set first parameter
C879    JSR     &C892   ;and do VDU 19
C87C    LSR     &0320   ;
C87F    DEC     &031F   ;
C882    BPL     &C879   ;
C884    RTS             ;exit

;********* 2 colour mode ************************************************

C885    LDX     #&07    ;X=7
C887    JSR     &C88F   ;execute VDU 19
C88A    LDX     #&00    ;X=0
C88C    STX     &031F   ;store it as
C88F    STX     &0320   ;both parameters



*************************************************************************
*                                                                       *
*                                                                       *
*       VDU 19   define logical colours                 5 parameters    *
*                                                                       *
*                                                                       *
*************************************************************************
;&31F=first parameter logical colour
;&320=second physical colour

C892    PHP             ;push processor flags
C893    SEI             ;disable interrupts
C894    LDA     &031F   ;get first parameter and
C897    AND     &0360   ;number of logical colours less 1
C89A    TAX             ;toi make legal  X=A
C89B    LDA     &0320   ;A=second parameter
C89E    AND     #&0F    ;make legal
C8A0    STA     &036F,X ;colour pallette
C8A3    TAY             ;Y=A
C8A4    LDA     &0360   ;number of logical colours less 1
C8A7    STA     &FA     ;store it
C8A9    CMP     #&03    ;is it 4 colour mode??
C8AB    PHP             ;save flags
C8AC    TXA             ;A=X
C8AD    ROR             ;rotate A into &FA
C8AE    ROR     &FA     ;
C8B0    BCS     &C8AD   ;
C8B2    ASL     &FA     ;
C8B4    TYA             ;A=Y
C8B5    ORA     &FA     ;
C8B7    TAX             ;
C8B8    LDY     #&00    ;Y=0
C8BA    PLP             ;check flags
C8BB    PHP             ;
C8BC    BNE     &C8CC   ;if A<>3 earlier C8CC
C8BE    AND     #&60    ;else A=&60 to test bits 5 and 6
C8C0    BEQ     &C8CB   ;if not set C8CB
C8C2    CMP     #&60    ;else if both set
C8C4    BEQ     &C8CB   ;C8CB
C8C6    TXA             ;A=X
C8C7    EOR     #&60    ;invert
C8C9    BNE     &C8CC   ;and if not 0 C8CC

C8CB    TXA             ;X=A
C8CC    JSR     &EA11   ;call Osbyte 155 pass data to pallette register
C8CF    TYA             ;
```

```
C8D0    SEC             ;
C8D1    ADC     &0360   ;number of logical colours less 1
C8D4    TAY             ;
C8D5    TXA             ;
C8D6    ADC     #&10    ;
C8D8    TAX             ;
C8D9    CPY     #&10    ;if Y<16 do it again
C8DB    BCC     &C8BA   ;
C8DD    PLP             ;pull flags twice
C8DE    PLP             ;
C8DF    RTS             ;and exit



*************************************************************************
*                                                                       *
*                                                                       *
*       OSWORD 12    WRITE PALLETTE                                     *
*                                                                       *
*                                                                       *
*************************************************************************
;on entry X=&F0:Y=&F1:YX points to parameter block
;byte 0 = logical colour;  byte 1 physical colour; bytes 2-4=0

C8E0    PHP             ;push flags
C8E1    AND     &0360   ;and with number of logical colours less 1
C8E4    TAX             ;X=A
C8E5    INY             ;Y=Y+1
C8E6    LDA     (&F0),Y ;get phsical colour
C8E8    JMP     &C89E   ;do VDU19 with parameters in X and A



*************************************************************************
*                                                                       *
*                                                                       *
*       VDU     22              Select Mode   1 parameter               *
*                                                                       *
*                                                                       *
*************************************************************************
;parameter in &323




C8EB    LDA     &0323   ;get parameter
C8EE    JMP     &CB33   ;goto CB33



*************************************************************************
*                                                                       *
*                                                                       *
*       VDU 23 Define characters                9 parameters            *
*                                                                       *
*                                                                       *
*************************************************************************
;parameters are:-
;31B character to define
```

```
;31C to 323 definition

C8F1    LDA     &031B   ;get character to define
C8F4    CMP     #&20    ;is it ' '
C8F6    BCC     &C93F   ;if less then it is an instruction to set CRT
                        ;controller goto C93F
C8F8    PHA             ;else save parameter
C8F9    LSR             ;A=A/32
C8FA    LSR             ;
C8FB    LSR             ;
C8FC    LSR             ;
C8FD    LSR             ;
C8FE    TAX             ;X=A
C8FF    LDA     &C40D,X ;get font flag mask from table (A=&80/2^X)
C902    BIT     &0367   ;font flag
C905    BNE     &C927   ;and if A<>0 C927 storage area is established already
C907    ORA     &0367   ;or with font flag to set bit found to be 0
C90A    STA     &0367   ;font flag
C90D    TXA             ;get back A
C90E    AND     #&03    ;And 3 to clear all but bits 0 and 1
C910    CLC             ;clear carry
C911    ADC     #&BF    ;add &BF (A=&C0,&C1,&C2) to select a character page
C913    STA     &DF     ;store it
C915    LDA     &0367,X ;get font location byte (normally &0C)
C918    STA     &DD     ;store it
C91A    LDY     #&00    ;Y=0 so (&DE) holds (&C000 -&C2FF)
C91C    STY     &DC     ;
C91E    STY     &DE     ;

C920    LDA     (&DE),Y ;transfer page to storage area
C922    STA     (&DC),Y ;
C924    DEY             ;
C925    BNE     &C920   ;

C927    PLA             ;get back A
C928    JSR     &D03E   ;set up character definition pointers

C92B    LDY     #&07    ;Y=7
C92D    LDA     &031C,Y ;transfer definition parameters
C930    STA     (&DE),Y ;to RAM definition
C932    DEY             ;
C933    BPL     &C92D   ;

C935    RTS             ;and exit

;
C936    PLA             ;Pull A
C937    RTS             ;and exit
;



************ VDU EXTENSION ********************************************

C938    LDA     &031F   ;A=fifth VDU parameter
C93B    CLC             ;clear carry
C93C    JMP     (&0226) ;jump via VDUV vector


********** set CRT controller ****************************************

C93F    CMP     #&01    ;does A=1
C941    BCC     &C958   ;if less (0) then set CRT register

C943    BNE     &C93C   ;if not 1 jump to VDUV
```

```
C945    JSR     &C588   ;A=0 if text cursor A=&20 if graphics cursor
C948    BNE     &C937   ;if graphics exit
C94A    LDA     #&20    ;else A=&20
C94C    LDY     &031C   ;Y=second VDU parameter
C94F    BEQ     &C954   ;if 0 C954
C951    LDA     &035F   ;last setting of CRT controller register

C954    LDY     #&0A    ;Y=10
C956    BNE     &C985   ;jump to C985

C958    LDA     &031D   ;get third
C95B    LDY     &031C   ;and second parameter
C95E    CPY     #&07    ;is Y=7
C960    BCC     &C985   ;if less C985
C962    BNE     &C967   ;else if >7 C967
C964    ADC     &0290   ;else ADD screen vertical display adjustment

C967    CPY     #&08    ;If Y<>8
C969    BNE     &C972   ;C972
C96B    ORA     #&00    ;if bit 7 set
C96D    BMI     &C972   ;C972
C96F    EOR     &0291   ;else EOR with interlace toggle

C972    CPY     #&0A    ;Y=10??
C974    BNE     &C985   ;if not C985
C976    STA     &035F   ;last setting of CRT controller register
C979    TAY             ;Y=A
C97A    LDA     &D0     ;VDU status byte
C97C    AND     #&20    ;check bit 5 printing at graphics cursor??
C97E    PHP             ;push flags
C97F    TYA             ;Y=A
C980    LDY     #&0A    ;Y=10
C982    PLP             ;pull flags
C983    BNE     &C98B   ;if graphics in use then C98B

C985    STY     &FE00   ;else set CRTC address register
C988    STA     &FE01   ;and poke new value to register Y
C98B    RTS             ;exit



*************************************************************************
*                                                                       *
*                                                                       *
*       VDU 25          PLOT                    5 parameters            *
*                                                                       *
*                                                                       *
*************************************************************************
;
C98C    LDX     &0361   ;pixels per byte
C98F    BEQ     &C938   ;if no graphics available go via VDU Extension
C991    JMP     &D060   ;else enter Plot routine at D060


********** adjust screen RAM addresses **********************************

C994    LDX     &0350   ;window area start address lo
C997    LDA     &0351   ;window area start address hi
C99A    JSR     &CCF8   ;subtract bytes per character row from this
C99D    BCS     &C9B3   ;if no wraparound needed C9B3

C99F    ADC     &0354   ;screen RAM size hi byte to wrap around
C9A2    BCC     &C9B3   ;

C9A4    LDX     &0350   ;window area start address lo
```

```
C9A7    LDA     &0351   ;window area start address hi
C9AA    JSR     &CAD4   ;add bytes per char. row
C9AD    BPL     &C9B3   ;

C9AF    SEC             ;wrap around i other direction
C9B0    SBC     &0354   ;screen RAM size hi byte
C9B3    STA     &0351   ;window area start address hi
C9B6    STX     &0350   ;window area start address lo
C9B9    LDY     #&0C    ;Y=12
C9BB    BNE     &CA0E   ;jump to CA0E



*************************************************************************
*                                                                       *
*                                                                       *
*       VDU 26  set default windows             0 parameters            *
*                                                                       *
*                                                                       *
*************************************************************************


C9BD    LDA     #&00    ;A=0
C9BF    LDX     #&2C    ;X=&2C

C9C1    STA     &0300,X ;clear all windows
C9C4    DEX             ;
C9C5    BPL     &C9C1   ;until X=&FF

C9C7    LDX     &0355   ;screen mode
C9CA    LDY     C3EF,X  ;text window right hand margin maximum
C9CD    STY     &030A   ;text window right
C9D0    JSR     &CA88   ;calculate number of bytes in a line
C9D3    LDY     &C3E7,X ;text window bottom margin maximum
C9D6    STY     &0309   ;bottom margin
C9D9    LDY     #&03    ;Y=3
C9DB    STY     &0323   ;set as last parameter
C9DE    INY             ;increment Y
C9DF    STY     &0321   ;set parameters
C9E2    DEC     &0322   ;
C9E5    DEC     &0320   ;
C9E8    JSR     &CA39   ;and do VDU 24
C9EB    LDA     #&F7    ;
C9ED    JSR     &C5A8   ;clear bit 3 of &D0
C9F0    LDX     &0350   ;window area start address lo
C9F3    LDA     &0351   ;window area start address hi
C9F6    STX     &034A   ;text cursor 6845 address
C9F9    STA     &034B   ;text cursor 6845 address
C9FC    BPL     &CA02   ;set cursor position
C9FE    SEC             ;
C9FF    SBC     &0354   ;screen RAM size hi byte


*************** set cursor position ***********************************


CA02    STX     &D8     ;set &D8/9 from X/A
CA04    STA     &D9     ;
CA06    LDX     &034A   ;text cursor 6845 address
CA09    LDA     &034B   ;text cursor 6845 address
CA0C    LDY     #&0E    ;Y=15
CA0E    PHA             ;Push A
CA0F    LDA     &0355   ;screen mode
CA12    CMP     #&07    ;is it mode 7?
CA14    PLA             ;get back A
```

```
CA15    BCS     &CA27   ;if mode 7 selected CA27
CA17    STX     &DA     ;else store X
CA19    LSR             ;divide X/A by 8
CA1A    ROR     &DA     ;
CA1C    LSR             ;
CA1D    ROR     &DA     ;
CA1F    LSR             ;
CA20    ROR     &DA     ;
CA22    LDX     &DA     ;
CA24    JMP     &CA2B   ;goto CA2B

CA27    SBC     #&74    ;mode 7 subtract &74
CA29    EOR     #&20    ;EOR with &20
CA2B    STY     &FE00   ;write to CRTC address file register
CA2E    STA     &FE01   ;and to relevant address (register 14)
CA31    INY             ;Increment Y
CA32    STY     &FE00   ;write to CRTC address file register
CA35    STX     &FE01   ;and to relevant address (register 15)
CA38    RTS             ;and RETURN
```