

In contrast to the majority of video display devices which allow the user various modes for the whole display, NICK allows many different modes in the same display frame.

Principal features of NICK are:-

- * Mixed mode displays
- * User definable characters from fonts of 64, 128 and 256
- * 8-bit colour output (256 colours)
- * 2,4,16, and 256 colours per line chosen from 256
- * Maximum resolution (using interlace) 672 * 512
- * Cell based graphics, bitmap and characters
- * Characters any height from 1 to 256 scanlines
- * Choice of 256 border colours
- * User defined screen width and height
- * External colour input (unlimited sprites or TV camera)
- * Efficient use of RAM (can work with < 1K of RAM)
- * 4 colour-pairs in 84 column text mode
- * Special use of bits to increase colour options
- * Pointer-based memory mapping, for flexibility and speed

1. VIDEO RAM ADDRESSING

The Nick Chip is capable of addressing 64k of memory, referred to as the "VIDEO RAM", which consists of the four segments 0FCh to 0FFh. Thus the Nick chip can address any of the internal 64k or RAM in the Enterprise 64, but cannot address any additional RAM which is added.

The Nick Chip addresses this 64k directly, without going through the Dave Chip paging registers. For example it will always address the start of segment 0FDh as address 04000h, regardless of where the Z-80 is seeing this segment, indeed it may not even be paged in. Care must be taken to avoid confusing these video addresses with Z-80 addresses.

2. LINE PARAMETER TABLE

The Display generated by the Nick Chip is controlled by a block of data in video RAM referred to as the LINE PARAMETER TABLE. This consists of a series of LINE PARAMETER BLOCKS, each of which is 16 bytes long. The line parameter table can be positioned on any 16-byte boundary in video RAM and can contain any number of line parameter blocks. The user simply loads the address of the start of the line parameter table into a register in the Nick Chip and the display is then generated automatically.

Each line parameter block controls a section of the screen called a MODE LINE, and contains data to indicate what type of display this mode line is and the address of the video RAM containing the screen data. A mode line may consist of any number of scanline from 1 to 256. The Nick Chip generates the required number of scanlines for the first mode line and then goes on to the next mode line, reading the next line parameter block and so on.

The last line parameter block in the table must have a particular bit set to indicate that when this mode line is finished the Nick Chip should start again at the beginning of the line parameter table. Normally the total number of scanlines in all the modes lines in the table would be 312, so that each scan of the line parameter table generates one TV frame. However if interlacing or other special effects are required, the line parameter table may be larger.

The Nick Chip generates all the horizontal timing itself so that all scan lines will be 64uS long and have the correct blanking, sync and colour burst signals. However the vertical timing is under the control of the programmer. The line parameter table must contain suitable line parameter blocks to generate the vertical blanking and sync signals.

3. LINE PARAMETER BASE POINTER

The address of the start of the line parameter table must be loaded into the Nick Chip in order to generate a display. Only the top twelve bits are loaded since it must always be on a 16 byte boundary, and two separate 8 bit registers (LPL and LPH) in the Nick Chip are used.

The user can switch to a different display by simply loading a new line parameter base pointer value. The Nick Chip will finish its current scan of the line parameter table and then re-load the new value and start on the new line parameter table. There can be a problem if the Nick Chip happens to be in the middle of a re-load when the user writes out the new address. It is possible for the Nick Chip to re-load half of the old address and half of the new one, and start scanning the wrong section of memory. To avoid this problem, any such re-loading of the line parameter base register should be synchronised with a video interrupt to ensure that the re-load is not in progress. Video interrupts are described later.

When the Nick Chip is switched on it will initially be scanning garbage until a base pointer is loaded. However when a new value is loaded it will not be used until the Nick Chip reads a line parameter block with the RELOAD bit set, causing a delay before the display appears correctly. To avoid this problem, two otherwise unused bits in LPH are used to allow a forced re-load of the line parameter base pointer. In normal use these bits are not set to one. To force a re-load the following procedure must be followed.

1. Write the low eight bits to LPL.
2. Write the top four bits to LPH, bits 6 & 7 clear.
3. Write top four bits again, bit 6 set, bit 7 clear.
4. Write top four bits again, bits 6 and 7 set.

This procedure will guarantee that the new line parameter table is started immediately.

4. LINE PARAMETER BLOCKS

Each line parameter block consists of 16 bytes which the Nick Chip reads in during the horizontal blanking time at the start of a scanline. The sixteen bytes are:

- 0 SC Two's complement of the number of the number of scanlines in this mode line. Zero means 256 scanlines.

1 MB Mode byte.

b7	b6	b5	b4	b3	b2	b1	b0
VINT	COLOUR MODE	VRES	VIDEO MODE	RELOAD			

VINT - Set to cause a video interrupt at the start of this mode line - see later section.

COLOUR MODE - Controls how the display bytes are translated into colour bytes:

- 00 - Two colour mode
- 01 - Four colour mode
- 10 - Sixteen colour mode
- 11 - 256 colour mode

VRES - Set to allow full vertical resolution in graphics modes, reset for reduced vertical resolution

VIDEO MODE - Overall video mode, controls how display bytes are read from video RAM.

- 000 - VSYNC mode
- 001 - PIXEL mode
- 010 - ATTRIBUTE mode
- 011 - CH256 mode
- 100 - CH128 mode
- 101 - CH64 mode
- 110 - unused
- 111 - LPIXEL mode

2 LM b0...b5 - Left margin of display.
 b6 - ALTIND1 \ Control alternate colours in
 b7 - ALTIND0 / 2 colour character modes.

3 RM b0...b5 - Right margin of display.
 b6 - LSBALT \ Control alternate colours in
 b7 - MSBALT / 2 colour graphics modes.

4 LD1L \ Primary video data address, low byte first.
 5 LD1H /

6 LD2L \ Secondary video data address, low byte first.
 7 LD3H /

8 COL0 \
 9 COL1 |
 10 COL2 | First eight palette colours. Not used
 11 COL3 | in 256 colour mode.
 12 COL4 |
 13 COL5 |
 14 COL6 |
 15 COL7 /

5. LEFT and RIGHT MARGINS

Each horizontal scan line is divided up into 57 slots, with the Nick Chip able to read two bytes from video RAM in each slot. The first eight slots are used to read in the line parameter block, and the last three are used to provide refresh of the video RAM. This leaves 46 slots which are available for display.

The left and right margin bytes in the mode line define which slot number the active display starts and finishes on respectively. The area outside these margins will be border colour, apart from the necessary horizontal blanking and sync. In practice, because of TV overscan, some of the slots will be off the screen, and a practical limit to the display width is 42 slots, using a left margin value of 10 and a right margin of 52.

In the active section of display between the margins, the Nick Chip reads bytes from video RAM in a way depending on the video mode and produces either one or two bytes for display in each slot. The way these bytes are converted into pixels depends on the colour mode selected.

If the right margin is set to a lower value than the left margin then the entire mode line will be border colour, with no active display. Note that the margin positions have special meaning in VSYNC mode where they are used to generate the vertical sync pulses. This is described in the section on vertical synchronisation.

6. COLOUR MODE and the PALETTE REGISTERS

The colour mode controls the way in which display bytes are converted into colours. The Nick Chip has eight colour outputs, allowing 256 different colours to be generated. In 256 colour mode, each byte read from video RAM is simply output directly as a colour byte. In all other colour modes (two, four and sixteen), the display byte is split up into a number of pixels, and the values of these pixels are used to select one of up to sixteen palette colours to be displayed. The translation of colour bytes into actual colours is explained in a later section.

The first 8 palette colours (0...7) are defined by the last 8 bytes in the line parameter block. Thus these can be any eight colours from the total of 256, and can be chosen separately for each mode line. The remaining eight colours (in the range 8...16) are generated by taking the bottom 3 bits of the palette colour number, and forming a complete colour byte by putting five bits from the FIXBIAS register in the top five bits. The FIXBIAS register is a single register in the Nick Chip (port 080h) which is also used for various other functions described later. Thus it is possible to choose any of 32 groups of 8 colours for the remaining palette colours, but they will be the same for the whole screen (unless the FIXBIAS register is changed rapidly under interrupt control!).

The translation of display bytes into pixels is different for the three palette colour modes. Note particularly the ordering of bits in the pixels for 16 colour mode.

2 Colour Mode - 8 Pixels per Byte, each bit selects either palette colour 0 or 1 (but see notes on ALTIND0, ALTIND1, MSBALT and LSBALT in the video mode sections).

b7	b6	b5	b4	b3	b2	b1	b0
----	----	----	----	----	----	----	----

4 Colour Mode - 4 Pixels per Byte, each pair of bits selects one of the palette colours 0, 1, 2 or 3.

b7,b3	b6,b2	b5,b1	b4,b0
-------	-------	-------	-------

16 Colour Mode - 2 Pixels per Byte, each group of four bits selects any one of the sixteen palette colours. Note the order of bits in each pixel.

b7,b3,b5,b1	b6,b2,b4,b0
-------------	-------------

7. PIXEL GRAPHICS MODES

In PIXEL and LPIXEL graphics modes the LD2 pointer value is not used at all. The LD1 pointer points at the address in video RAM of the first byte to be displayed. Bytes are read in sequence from video RAM starting from the left margin and continuing to the right margin of each scanline. If VRES is set then the next scanline of this modeline continues at the next byte in video RAM, if VRES is clear then the LD1 pointer is reloaded so the same data will be displayed on each scanline. This allows memory to be saved by reducing vertical resolution.

The bytes read from video RAM are translated according to the colour mode as described above. The only difference between PIXEL and LPIXEL mode is that in PIXEL mode two bytes are read in each slot, whereas in LPIXEL mode only one byte is read. Thus LPIXEL mode only has half the horizontal resolution, and only uses half the amount of RAM.

Two special control bits are provided for use with 2-colour pixel graphics modes. These are MSBALT and LSBALT which are in the top two bits of the left margin byte. They are provided for simulating high resolution text modes in PIXEL mode, and allow different coloured characters to be displayed.

If MSBALT is set then bit-7 of all display bytes will be forced to zero before displaying. If bit-7 was set then palette colours 2 and 3 will be used instead of colours 0 and 1 for the remaining pixels in this byte.

If LSBALT is set then bit-0 of all display bytes will be forced to zero before displaying. If bit-0 was set then palette colours 4...7 will be used instead of palette colours 0...3 for the remaining pixels in this byte.

8. CHARACTER MODES - 64, 128 and 256

In the character modes, LD2 points to a character font and LD1 points to the character codes to be displayed. One character code is read, and the font byte for it fetched, in each slot, allowing up to 42 characters per line. VRES should always be clear for character modes. The three character modes vary only in the number of characters in the font, and therefore the amount of RAM required for the font. CH128 will be used as an example in this description, the other two modes follow trivially.

The font for CH128 mode must always start on a 128 byte boundary, and the address stored in LD2 is in fact the font address divided by 128 (thus the top 9 out of 16 bits). The remaining 7 bits of address are generated by the Nick Chip.

The first 128 bytes of the font are the display bytes for the top scanline of each of the 128 characters. The next 128 bytes are the display bytes for the second scanline and so on. There must be sufficient lines in the font for the SC count in the line parameter block.

The character codes are read sequentially from the LD1 address and the appropriate display byte for the current scanline obtained from the font. This display byte is translated into colours just as in the pixel graphics modes, depending on the colour mode. For producing ASCII type characters, 2-colour mode is normally used but other colour modes can be used for doing character based graphics, with fewer pixel across each character.

Two control bits are provided to allow different coloured characters in 2-colour text modes. These are ALTIND0 and ALTIND1 which are the top two bits of the right margin byte. These are particularly useful in 64 and 128 characters modes when the top bits of the character codes are not used.

If ALTIND1 is set then characters codes with bit-7 set will use palette colours 2 and 3 instead of 0 and 1 for displaying this character.

If ALTIND0 is set then character codes with bit-6 set will use palette colours 4...7 instead of 0...3 for displaying this character.

9. ATTRIBUTE GRAPHICS MODE

In attribute graphics mode, LD2 points to two colour bitmap graphics data, and LD1 point to an area of cell based graphics attributes. When using attribute mode, the colour mode should always be set to 2-colour, any other colour modes will have unpredictable results.

The pixel data is read in just the same way as for 2-colour LPIXEL graphics mode, but the corresponding attribute byte defines which two palette colours should be used for displaying this single byte as below:

b7	b6	b5	b4	b3	b2	b1	b0
Background colour				Foreground colour			

The correspondence between attribute bytes and pixel data bytes depends on VRES. If VRES is set then there is a one to one correspondence, so the attribute data will take up the same amount of RAM as the pixel data. If VRES is clear then the attribute data will repeat for each scanline in the mode line. Thus each attribute byte will correspond to a number of bytes vertically above one another, in the same way as the character code corresponds to several lines in character modes.

10. VSYNC MODE - Vertical Synchronisation

In VSYNC mode, no data is displayed on the screen, the video output is held at blanking level, and the margins define the start and end of the vertical sync pulse. Note also that colour burst is disabled during VSYNC lines.

During the active vertical sync period, the normal horizontal sync pulses will be inverted to prevent loss of horizontal sync. This is essential to produce a stable display.

A line parameter table must contain a sequence of line parameter blocks to generate the correct vertical blanking and sync pulses. The recommended sequence is as follows, the last 12 bytes of each LPB should be zero so just the first four bytes are listed here.

DB	256-3,	VBLANK,	63,	0
DB	256-2,	VBLANK,	6,	63
DB	256-1,	VBLANK,	63,	32
DB	256-19,	PIXEL,	6,	63

The first of these LPB's gives three lines of blanking. Since the left margin is set to 63 which is never reached, the sync will not be turned on. The sync is turned on near the start of the second LPB, and remains on throughout this LPB because the right margin of 63 is never reached. The sync is turned off half way through the single scan line third LPB, giving 2.5 lines of sync. The fourth LPB provides nineteen more lines of blanking to make a total of 25 lines, but in 2-colour PIXEL mode so that the colour burst is re-enabled. In this LPB the left margin is set to 6 to ensure that the border colour does not get displayed, the palette must be set to all black.

These LPBs make a total of 25 lines, leaving 287 lines left for the display. In practice, because of overscanning, it is not possible to use more than about 256 of these for display, the rest must provide top and bottom borders.

11. VIDEO INTERRUPTS

If a line parameter block has the VINT bit set in its mode byte, then the VIRQ output of the Nick Chip will be made low for the duration of that mode line. This is fed into the Dave Chip where the start of the low pulse will cause a Z-80 interrupt, assuming that the video interrupts are enabled in the Dave Chip. Details of how to program the Dave Chip interrupts will be found in the Dave Chip specification, but there are several points to note here.

Because the VIRQ output is held low for the duration of the mode line, and because the Dave Chip input is edge triggered, then two consecutive mode lines with VINT set will only cause one interrupt because the second mode line will merely hold VIRQ low without producing an edge. It is therefore necessary to have a mode line with VINT clear to allow VIRQ to return high before another interrupt can be generated.

Once the interrupt has occurred, it is latched inside the Dave Chip and must be reset here before interrupts can be re-enabled. This is normal practice with Dave Chip interrupts. If the interrupt is cleared and interrupts are enabled again before the VINT mode line has finished, then another interrupt WILL NOT be triggered because the Dave Chip input is edge triggered.

12. DETERMINATION OF COLOURS

As described above, the Nick Chip generates a colour byte for each point on the screen, depending on the video mode, colour mode, palette registers and so on. The 8-bits of this colour value are converted into separate RED, GREEN and BLUE signals which are available on the monitor connector, and are also converted into a PAL encoded, UHF modulated signal which can be fed directly into a television.

Of the eight bits, there are three each for red and green and two bits for blue, thus allowing eight levels of red and green and four levels of blue. The layout of bits is:

b7	b6	b5	b4	b3	b2	b1	b0
G2	R2	B1	G1	R1	B0	G0	R0
green		red		blue		green	
		blue		red		green	
				blue			

The three bits for each colour are combined according to the formulas below. B0, G0 and R0 are the most significant bits of each colour, and the weighting for the blue levels takes into account the fact that there are only four levels of blue.

$$\begin{aligned}\text{total RED} &= (R2 + R1*2 + R0*4) / 7 \\ \text{total GREEN} &= (G2 + G1*2 + G0*4) / 7 \\ \text{total BLUE} &= (B1 + B0*2) / 3\end{aligned}$$

13. CONTROL REGISTERS

Z-80 port numbers 080h to 08Fh reserved for Nick Chip registers although only the first four of these (080h to 083h) are used at present. These registers are write only, the values cannot be read back.

080h FIXBIAS

b0..b4 Colour bias to be used for top 5 bits of palette colours 8-15.

b5..b6 Controls priority of external colour inputs on ECO-EC3, allowing 16 colour sprites to be implemented. The external colour will only be selected when the /EXTC input is low and the display is active (not border).

=00 ECO-EC3 always select the corresponding palette colour.

=01 ECO-EC3 select the corresponding palette colour if the internal display is generating a palette colour in the range COL8-COL15.

=10 ECO-EC3 select the corresponding palette colour if the internal display is generating a palette colour in the range COL8-COL15 OR the external colour is in the range COL0-COL7 (EC3 low).

=11 ECO-EC3 select the corresponding palette colour if the internal display is generating a palette colour in the range COL8-COL15 OR the external colour is in the ranges COL0-COL3 or COL8 - COL11 (EC2 low).

b7 VC1 output of Nick Chip. Set to disable the internal speaker in the Enterprise.

081h BORDER Eight bit colour value which will be displayed outside the margins of all mode lines.

082h LPL Low part (A4...A11) of line parameter base pointer. The lowest four bits of the address are generated by the Nick Chip.

083h LPH

- b0..b3 Top four bits (A12...A15) of line parameter base pointer.
- b4..b5 Not used.
- b6 Set for normal operation. Clear to inhibit clocking of line parameter counter.
- b7 Set for normal operation. Clear to force an immediate re-load of the line parameter base pointer.

+++++++ END OF DOCUMENT ++++++