Modern TV Receiver Techniques

Part 14: Teletext

Eugene Trundle

Teletext is the oldest form of digital broadcasting technology. It has now been in use in the UK for twenty years, during which period teletext decoders have steadily shrunk in size and cost and the service has been improved. Teletext uses spare capacity in the field blanking interval to transmit data in the form of pulses that provide 'typing instructions' for pages of text and graphics. A TV set that contains a teletext decoder extracts this data from the overall transmission and stores it so that, when requested by the viewer, page readouts appear on the screen.

There are five teletext 'levels' that offer progressively better presentation of text and graphics on the screen. The 'higher' the level, the larger the memory and the more sophisticated the decoder required. Perhaps because of lack of enthusiasm amongst viewers we have, in the UK, progressed barely further than level 1, which is the most basic form of broadcast text. Access time however has been reduced by transmitting more data per TV field, by clever arrangement of the transmitted page sequences and by increasing the receiver's memory capacity.

The Field Blanking Interval

In Part 10 we saw that the field blanking period, during which the beams return from the bottom of the screen to the top to start a new field scan, lasts for a period equivalent to

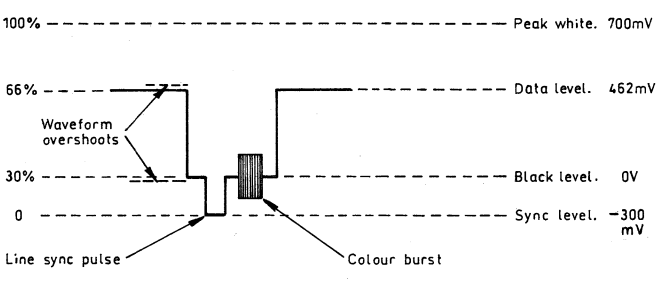

Fig. 1: Teletext data levels.

25 lines, i.e. 1·6msec. The field sync pulse group occupies 7·5 lines/480µsec of this space, leaving a period of 17·5 lines/1·12msec before the start of the video information for the first line of the next displayed TV field. This generous amount of time was originally allocated because early TV sets had a rather slow field flyback. Subsequently, with field timebases that had a faster flyback and advantages in technology at both ends of the transmission chain, it became possible to put some of these spare lines to good use: they were used to carry video insertion test signals (VITS) that provide signal identification, network switching and performance appraisal. The VITS occupy lines 19 and 20 in even fields and lines 332 and 333 in odd fields. The preceding twelve lines, 7-18 and 320-331, are used for text data.

Each text data line carries a colour burst to maintain the colour decoder subcarrier oscillator synchronisation and the usual line sync pulse. The text pulses occupy the rest of the line. This is the same period as the ordinary picture signal: 52µsec.

To prevent intercarrier buzz with f.m. sound systems the amplitude of the text pulses is limited to 66 per cent of the peak white signal, see Fig. 1. Binary zero is represented by the text signal hitting black level, binary one by the signal rising to the 66 per cent level. This represents 462mV with a standard 1V peak-to-peak video signal. The pulses are not square: they are filtered before transmission and further rounded in the receiver's i.f. and demodulator circuits. Thus by the time they reach the text decoder they are of approximately sinusoidal shape. In addition the pulses are transmitted in NRZ (non return to zero) form. This means that the signal remains high to indicate a string of binary ones and low when a string of binary zeros is being transmitted.

Text Data Format

The text data rate is 6·9375Mbits per second, which is 444 times the 625-line scan frequency. The pulses are not necessarily locked to the scan rate however. A data train representing 01010101 etc. to the end of the line will thus give rise to a quasi-sinewave at 3·46875MHz. At this rate there's room for 360 data bits on each TV line used for the purpose. These 360 bits produce a single row of text characters or graphics in the teletext screen display.

The 360 bits per line are divided into 45 eight-bit bytes. The first five of these bytes synchronise the decoder and provide an 'address' that indicates where the row is to be located in the text display. The remaining 40 bytes determine the characters in the row, which can thus have a maximum of 40 letters, numbers or graphic blocks. Where there's to be a blank or gap in the display the code for a blank space is sent so that data is always present.

The text display has 23 rows: thus a maximum of 23 x 40 = 920 characters can be displayed on a single page to make up simple maps, diagrams and alphanumeric text. Upper- and lower-case letters can be produced as well as figures, symbols and 'building blocks' for graphics, in any of eight colours formed by the three primaries plus black and white.

Fig. 2 shows the code table. Starting at the top on the left-hand side the first four bits define the row number and the next three the column number. The codes are then used to determine the character so that capital A for example would be called up by byte 1000001 (bits 1-7) while byte 1111101 produces the symbol #. Notice that to change from a capital to a lower-case letter only bit six changes, and that the codes for numbers consist of their binary equivalents followed by the code 011.

The 32 codes in columns 0 and 1 are control codes that indicate whether the codes in the following columns represent alphanumeric characters or graphics and give some 'attribute', such as a colour, to the characters in the row. These control codes all have bits 6 and 7 at zero: they produce no display themselves, occupying a single blank space in the displayed row – usually in the background colour of the preceding character.

Protection

Because the data is subject to interference and noise during its journey from the transmitter to the decoder protection against data corruption is required. It's provided

$b_7 b_6 b_5$ →						0 0 0	0 0 1	0 1 0		0 1 1		1 0 0	1 0 1	1 1 0		1 1 1	
Bits	b4	b3	b2	b1	Col → / ↓ Row	0	1	2	2a	3	3a	4	5	6	6a	7	7a
	0	0	0	0	0	NUL (1)	DLE (1)			0		@	P	—		p	
	0	0	0	1	1	Alpha^n Red	Graphics Red	!		1		A	Q	a		q	
	0	0	1	0	2	Alpha^n Green	Graphics Green	"		2		B	R	b		r	
	0	0	1	1	3	Alpha^n Yellow	Graphics Yellow	£		3		C	S	c		s	
	0	1	0	0	4	Alpha^n Blue	Graphics Blue	$		4		D	T	d		t	
	0	1	0	1	5	Alpha^n Magenta	Graphics Magenta	%		5		E	U	e		u	
	0	1	1	0	6	Alpha^n Cyan	Graphics Cyan	&		6		F	V	f		v	
	0	1	1	1	7	Alpha^n White (2)	Graphics White	'		7		G	W	g		w	
	1	0	0	0	8	Flash	Conceal Display	(		8		H	X	h		x	
	1	0	0	1	9	Steady (2)	Contiguous (2) Graphics	)		9		I	Y	i		y	
	1	0	1	0	10	End Box (2)	Separated Graphics	*		:		J	Z	j		z	
	1	0	1	1	11	Start Box	ESC (1)	+		;		K	←	k		¼	
	1	1	0	0	12	Normal (2) Height	Black (2) Background	,		<		L	½	l		‖	
	1	1	0	1	13	Double Height	New Background	-		=		M	→	m		¾	
	1	1	1	0	14	SO (1)	Hold Graphics	.		>		N	↑	n		÷	
	1	1	1	1	15	SI (1)	Release (2) Graphics	/		?		O	#	o		■	

Fig. 2: ASCII codes used for teletext data transmission. Codes can be referred to by their column and row, e.g. 2/5 = %. Control characters marked (1) are reserved for compatibility with other data codes; those marked (2) are presumed before each row begins.

by adding an 'odd-parity bit' at the end of each of the seven-bit codes shown in Fig. 2 so that each becomes an eight-bit byte. The idea is that the parity bit added always results in an odd number of one bits in the byte. The decoder can then check whether any bytes with an even number of ones have been received and reject these as being corrupt. This simple system has two limitations: it doesn't work if an even number of bits in the byte are incorrect, and it doesn't provide data correction. Despite this it is adequate for use with text symbols, where a minor error often goes unnoticed, and is anyway corrected when the page is cyclically updated.

Some of the transmitted data requires greater protection however. If anything goes wrong with the data that determines the page number the wrong page could appear on the screen when it's called up by the viewer, while if the row address data is wrong the row will appear in the wrong place on the screen. To avoid this the bytes that provide this information and the real-time clock data are heavily protected by means of a Hamming code, in which every alternate bit is a parity bit: bits 2, 4, 6 and 8 convey data while bits 1, 3, 5 and 7 provide protection – see Table 1. This enables single-bit errors to be detected and corrected by inversion. The penality incurred is that the amount of information conveyed by the byte is reduced.

Row Coding

The first row of a transmitted page, row 0, is the most important one since it contains the magazine and page numbers, the service name, the date and time. It's called the header row. To capture a page for display the viewer keys in the magazine and page numbers: when this keyed-in data corresponds with the transmitted code the relevant page is selected and displayed on the screen.

The data format for the header row is shown towards the

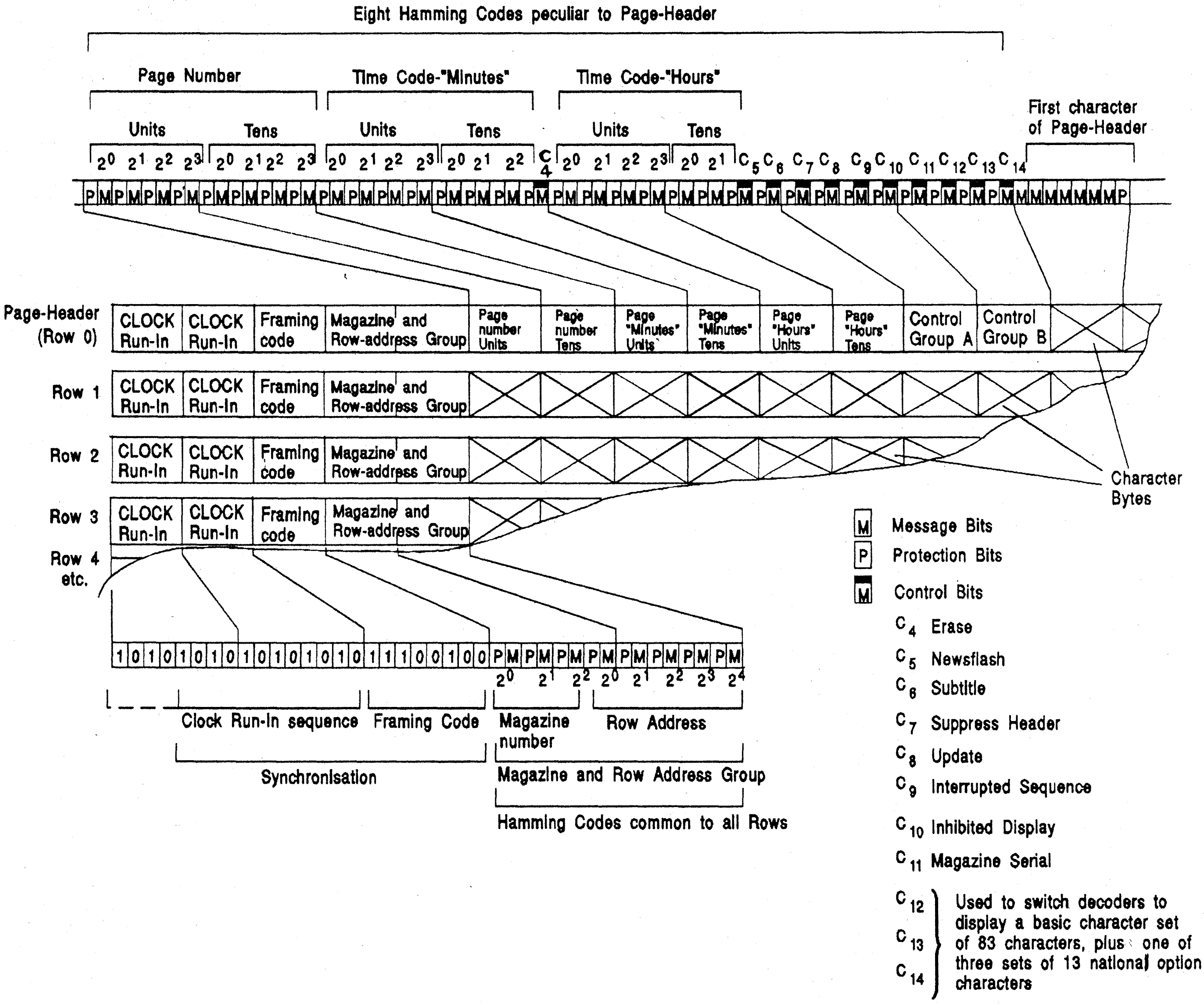

Fig. 3: Teletext row coding. The 'diamond-crossed' bytes carry display data, 32 characters in the header row and 40 in the others.

top in Fig. 3. There's first a clock run-in sequence that consists of two bytes carrying a series of ones and zeros – 101010 etc. This synchronises the decoder's bit-sampling clock oscillator. Next comes a framing code byte whose purpose is to identify the start of the data bytes in the following pulse train. Its pattern is 11100100, chosen to give a reliable reset signal to the rest of the decoder and to be amenable to correction by a circuit that's able to put right single-bit errors. Fig. 4 shows the principle of the framing code detector: when the and gate's output goes high it signals to the decoder that the next bit is the first one of the first data byte. Thereafter the circuit is reset at eight-bit intervals by a byte clock, thus individually partitioning off the bytes.

The next two bytes, which are Hamming-protected, provide the magazine and row-address information. In the header row this is followed by page number and time code (not necessarily real time) bytes to enable the viewer to select a particular page at a specified time. Two bytes that provide control information relating to the whole page follow – the codes that govern news flash, subtitle, update and similar functions, all with Hamming protection.

Eight data bytes of the header row have thus been used, leaving 32 for the display of characters. Except for the page number, the header row text is the same for all the pages of any one transmission: it shows the name of the service (Ceefax, Skytext, etc.) and the day, date and real time in hours, minutes and seconds.

The following rows have the same clock run-in and framing codes as the header row. These are followed by individual magazine and row-address bytes, which are the only ones that have Hamming protection in an 'ordinary' row. Bytes 6 to 45 contain eight-bit codes for the text or graphics in the line. Each byte is transmitted with the LSB (least significant bit) first and the parity bit last.

Control Codes

Some further explanation is required for columns 0 and 1 in Fig. 2. At the beginning of each transmitted row the decoder assumes that what follows will be steady, normal-height white alphanumeric characters on a black background, also that any graphic blocks will be contiguous (joined together). Any change from this, at any point along the line, requires the insertion of a control byte to tell the decoder. Thus if the second half of a row is to be concealed (say it's a quizz answer), the 'question' text will be followed by the code 0001100 then the answer code. The answer will then be revealed only when the viewer keys 'reveal'. The next row will be reset automatically. Thus if this is to be concealed the first byte must again be 0001100. Similarly if one word in a row is to flash it must be

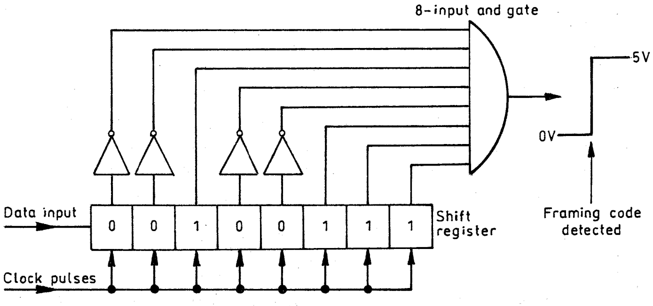

Fig. 4: Framing code detector: only the bit pattern 11100100 can open the and gate to produce a 0-5V transition. The clock pulses step the bits rightwards through the shift register.

preceded by code 0001100 then code 1001000.

Suppose that we want to transmit a row with red graphics at the left changing to green graphics half way across, perhaps as part of a map or diagram. After the row coding the first byte will be 1000100 to set the display mode followed by the graphics codes. Just before the transition to green point code 0100100 is sent to change the colour, followed by more graphics codes. There doesn't have to be a gap in the display where the control byte is inserted: in the graphics-hold mode (column 1, row 14, Fig. 2) the decoder repeats the previous graphic symbol in the control code space. This can't be done with alphanumeric characters, so the control byte in this case produces a gap in the display: it appears in the background colour.

Because bits 6 and 7 of these control codes are both at zero they are sent straight to the character generator at the output end of the decoder.

Page Transmission and Access Time

We've seen that each text line transmitted during the field blanking period can produce one row of text on the screen. So with 24 rows in the display 24 lines will be required. As the current rate is 12 teletext lines per field, it takes two fields to transmit a single teletext page. At a field rate of 50 per second, 25 pages a second can be transmitted – 1,500 pages a minute. Thus access time – the period between requesting a page and its appearance on the screen – largely depends on the number of pages being transmitted. If 750 pages are being transmitted the worst-case access time will be 30 seconds and the average wait 15 seconds – this is with the pages transmitted repeatedly in sequence.

To improve the system's 'user-friendliness' high-priority pages such as indexes and commonly-used pages are transmitted more often than others. Another time saver is the 'row-adaptive' feature, which enables the totally blank rows that occur in many text pages to be omitted from the data stream. It's also possible to interleave the text rows of several pages to give a perceived improvement in the access time: this is why each row has a magazine number (see Fig. 3). Subtitles (page 888) have the highest priority: they are transmitted on the first available line after the cueing point.

These techniques are limited in their effectiveness: in most cases they shorten the access time to some pages while lengthening the wait for others. From the user's point of view the most effective reduction in page access time comes from the use of the Fastext system.

Memory Capacity and Fastext

As the parity bits don't have to be stored the memory capacity required for a single page of text is 24 (rows) x 40 (characters) x 7 (bits), which comes out at 6,720 bits. This is the amount of memory that was provided in the earliest text decoders. Later types have a two-page memory (2K x 7 bits in practice) and automatically store the page following the one being displayed. Thus if you select page 100, page 101 will also be captured and held and can be called up for immediate display. This action prompts the decoder to search for, acquire and store the next page (102) in the now unused half of the memory, overwriting the stored page 100 with the page 102 data. Philips Components uses this system in the popular Euro-CCT (Computer Controlled Teletext) decoder.

A Fastext decoder incorporates a larger memory that can hold the selected page and the next seven, which are automatically loaded in sequence by the decoder. This is based on the premise that viewers usually step through pages sequentially. As each page is discarded the next page in the sequence is loaded. Thus virtually instantaneous page display is achieved.

The broadcaster's text editor can assist with the Fastext function by anticipating the user's requirements and sending additonal instructions to the decoder, based on the current page and its contents, for extra page acquisition. This works as follows. The bottom row of the text page has four colour-coded prompts which may typically be news, sports, travel and financial information. The remote control handset has four buttons with corresponding colours. Pressing any one of these selects magazine and page numbers simultaneously, loading appropriate pages into the memory for instant selection. If travel is selected for example the coloured prompts may change to air, sea, road delays and continue, the latter providing an escape from the travel menu.

A 'packet' system, in which additional data rows are transmitted but not necessarily displayed on screen, is used for Fastext data acquisition. The packets contain additional control data rather than characters and are sent before the teletext pages (which become packets 0 to 23) so that the control data arrives before the actual page. Packet 24, which is displayed as a row, is used for the Fastext prompts while packet 27 provides the page-linking data.

There are eight packets in the text specification for various purposes that include extended character sets (primarily for other languages) and general-purpose data

Table 1: Hamming code protection.

Message	*Hamming coded bits* 1	2	3	4	5	6	7	8
0	1	0	1	0	1	0	0	0
1	0	1	0	0	0	0	0	0
2	1	0	0	1	0	0	1	0
3	0	1	1	1	1	0	1	0
4	0	0	1	0	0	1	0	0
5	1	1	0	0	1	1	1	0
6	0	0	0	1	1	1	0	0
7	1	1	1	1	0	1	0	0
8	0	0	0	0	1	0	1	1
9	1	1	1	0	0	0	1	1
10	0	1	1	0	0	0	1	1
11	1	1	0	1	1	0	0	1
12	1	0	0	0	0	1	0	1
13	0	1	0	1	1	0	1	1
14	1	0	1	1	1	1	1	1
15	0	1	0	1	0	1	1	1

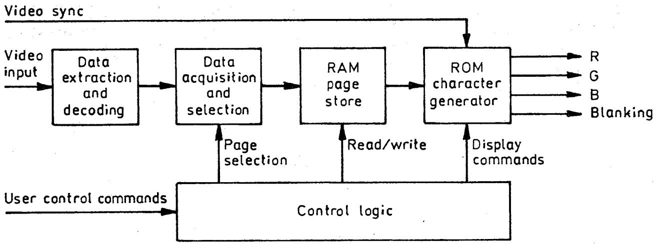

Fig. 5: Basic decoder block diagram to show the working principle.

transmission. Packet 30 is an interesting one, providing PDC (Programme Delivery Control) for VCRs. More on this later.

Decoder Overview

Fig. 5 shows a teletext decoder in the simplest possible block diagram form. The first block removes the data pulses from the video signal and decodes them. The next block selects the data requested by the viewer, according to instructions sent via the control logic block, and passes it to the page memory (RAM) whose write-in control signal also comes from the control block. Read-out from the RAM is again governed by the control block: it occurs at a slower rate, a complete field scan period being required for a page in the memory to be read out – it's written in during short 52μsec bursts at varied intervals. The data from the RAM is passed to a character-generator ROM which uses it to produce the characters, graphics and symbols shown in the body of Fig. 2 and to give them the colours and attributes dictated by the control data stored with the character codes. The ROM chip's outputs consist of RGB signals in serial data form plus a blanking pulse train. They are all passed to an RGB signal processing chip of the type described in Parts 6 and 7 of this series, after which the RGB signals are applied to the picture tube.

Decoder Operation

Fig. 6 provides a more detailed idea of the internal workings of the decoder. A composite video signal (video plus sync) at about 2V peak-to-peak is fed in at the left-hand side. The sync pulses are stripped from it and fed to field sync and line divider stages. They are also passed to the set's timebase generators and continue (generated within the decoder) when there's no TV signal present so that stored pages can, if required, be viewed 'after hours'.

The two clock run-in bytes synchronise and phase lock the data clock, which supplies 6·9375MHz pulses to the data separator to enable it to make zero or one decisions with each received bit. An adaptive slicer is used here, like the one described on page 505 of the June 1993 issue. The serial pulse train produced in this way must next be converted to parallel form. This is done by clocking the pulses into an eight-bit shift register then stepping them out into an eight-line bus each time the clock divider is reset by the framing code detector shown in Fig. 4. The eight data bits in the byte thus captured are held in the data latch (centre of Fig. 6) ready for transfer into the memory.

The bytes are checked for corruption, one at a time, in the parity-check block which puts up an error-inhibit flag in the read/write control block when a byte with incorrect parity is detected.

The Hamming-coded bytes are similarly checked, and

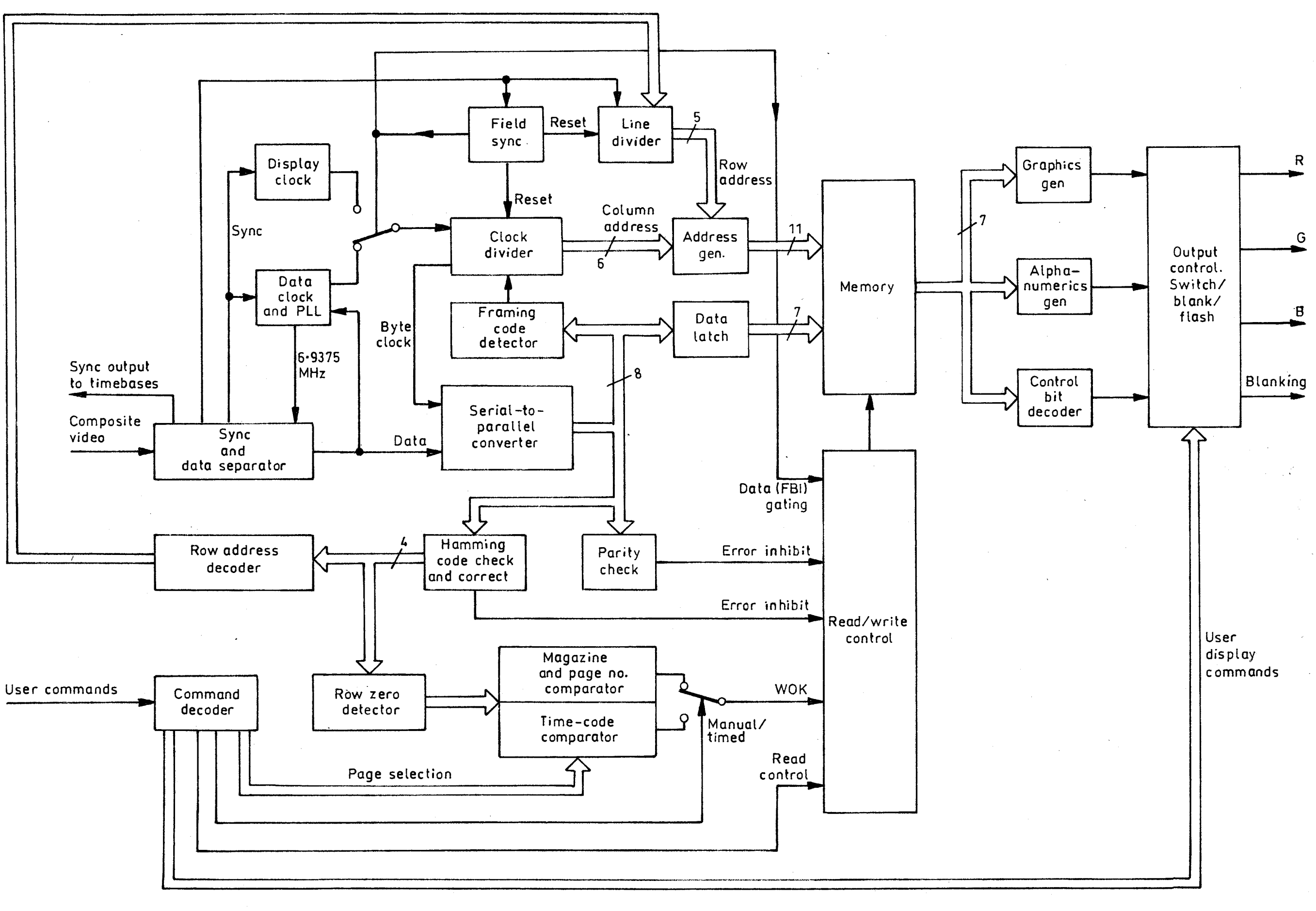

Fig. 6: A more detailed decoder block diagram to show the basic operation: practical processing arrangements are much more complex than this.

corrected if necessary, in the Hamming checker. This again inhibits the write-into-memory cycle should it detect an uncorrectable byte error. The data output from the Hamming block is in four-bit parallel form – four bits because, as we saw in Table 1 and Fig. 3, the other four bits in the byte are parity bits. It passes to the row address decoder which tells the memory, via the address generator, where the following 40 bytes of data are to be stored; and also to the row 0 detector which comes to life when it sees the row 0 code in the two bytes immediately after the framing code. The row 0 detector produces magazine, page and time-code data which are latched into comparators for updating each time a new header row is transmitted. The other input to each comparator consists of data entered by the viewer. When the magazine and page numbers at both sides of the comparator match exactly a WOK (Write OK) signal is passed to the read/write section, permitting data to be written into the memory from the seven-bit data latch. In the timed-page mode, transmitted and user-logged time codes are compared to produce the WOK command.

Thus the conditions required for a write-into-memory operation are: that the magazine and page numbers (and if required the time code) matches that requested by the viewer; that the data is complete and uncorrupted; and that the control bits in the header row are present and correct. When all these conditions have been satisfied the data is written into a memory address specified by the row and column address lines. In most decoders the memory has a two- or eight-page capacity that's automatically filled with data representing one or more linked pages as described earlier. Memory access time must be well under 1µsec, which is the nominal period between displayed characters.

Data from the viewer's infra-red remote control handset is fed into the command decoder in serial form, typically via an I2C bus. After serial-to-parallel conversion it's passed to the comparators and to the character output contol section to operate such functions as reveal and superimpose.

Next Month

We've now seen how the text data is acquired and stored in memory: next month we'll look at readout and character generation, then relate all this to practical teletext decoders and the chips they use.