

## Bibliography

*Acorn User Magazine*, published monthly, Addison Wesley

*6502 Assembly Language Programming*, L.A. Leventhal,  
OSBORNE/Mc Graw Hill, Berkeley, California

*Acorn Electron Expansion Application Note*, Acorn Computers  
Limited, 1984

*Acorn Electron User Guide*, Acorn Computers Limited,  
Cambridge, 1983

*Beebug Magazine*, published every five weeks, BEEBUG, PO Box  
109, High Wycombe, Bucks.

*Programming the 6502*, Rodney Zaks, Sybex, 1980

*R6522 Versatile Interface Adapter Data Sheet*, Rockwell  
International, 1981 TTL Data Book, Texas Instruments Inc., 1980

*The BASIC ROM User Guide for the BBC Micro and Acorn  
Electron*, Mark Plumbley, Adder Publishing/ Acornsoft Limited,  
Cambridge, 1984

*The Advanced User Guide for the BBC Microcomputer*, Bray,  
Dickens and Holmes, Cambridge Micro Centre, 1983

# Glossary

**Address Bus** – a set of 16 connections, each one of which can be set to logic 0 or logic 1. This allows the CPU to address &FFFF (65536) different memory locations.

**Active low** – signals which are *active low* are said to be valid when they are at logic level 0.

**Analogue to digital converter (ADC)** – this is a chip which can accept an analogue voltage at one of its inputs and provide a digital output of that voltage.

**Asynchronous** – two devices which are operating independently of one another are said to be operating asynchronously.

**Baud Rate** – used to define the speed at which a serial data link transfers data. One baud is equal to 1 bit of data transferred per second. The standard cassette baud rate of 1200 baud is therefore equal to 1200 bits per second.

**Bidirectional** – a communication line is bidirectional if data can be sent and received over it. The data bus lines are bidirectional.

**Bit of memory** – this is the fundamental unit of a computer's memory. It may only be in one of two possible states, usually represented by a 0 or 1.

**Buffer** – a software buffer is an area of memory set aside for data in the process of being transferred from one device or piece of software to another.

**Byte of memory** – 8 bits of memory. Data is normally transferred between devices one byte at a time over the data bus.

**Chip** – derived from the small piece of silicon wafer or chip which has all of the computer logic circuits etched into it. A chip is normally packaged in a black plastic case with small metal leads to connect it to the outside world.

**Clock** – it is necessary to provide some master timing reference to which all data transfers are tied. The clock provides this synchronisation. A 16MHz clock is applied to the ULA. From this, the clock timing for the 6502 CPU is derived. See chapter 15 for a discussion of the clock timing requirements.

**CPU** – (Central processing unit) – the 6502A in the Electron, It is this chip which does all of the computing work associated with running programs.

**Cycle** – this is usually applied to the 6502 clock. A complete clock cycle is the period between a clock going high, low, then high again. See *clock*.

**Data bus** – a set of eight connections over which all data transactions between devices in the BBC microcomputer take place.

**Field** – a space allocated for some data in a register, or in a program listing, For example, in an Assembly language program, the first few spaces are allocated to the line number field, the next few spaces are allocated to the label field, and so on.

**Handshaking** – this type of communications protocol is used when data is being transferred between two asynchronous devices. Two handshaking lines are normally required. One of these is a *data ready* signal from the originating device to the receiving device, When the receiving device has accepted the data, it sends a *data taken* signal back to the originating device, which then knows that it can send the second lot of data and so on. This type of handshaking is used with the RS423 serial interface option.

**High** – sometimes used to designate logic ‘1’

**Interrupt** – this signal is produced by peripheral devices and is always directed to the 6502A CPU. Upon receiving an interrupt, the 6502 will normally run a special interrupt routine program before continuing with the task in hand before it was interrupted.

**Latch** – a latch is used to retain information applied to it after the data has been removed, It is rather like a memory location except that the outputs from the bits within the latch are connected to some hardware.

**LED (Light emitting diode)** – acts like a diode by only allowing current to pass in one direction. Light is emitted whilst current is passed.

**Low** – sometimes used to designate logic '0'.

**Machine code** – the programs produced by the 6502 BASIC Assembler are machine code. A machine code program consists of a series of bytes in memory which the 6502 can execute directly.

**Mnemonic** – the name given to the text string which defines a particular 6502 operation in the BASIC assembler. LDA is a mnemonic which means *load accumulator*.

**Opcode** – the name given to the binary code of a 6502 instruction, For example, &AD LV the opcode which means *load accumulator*.

**Open Collector** – this is a characteristic of a transistor output line, It simply means that the collector pin of the transistor is not driving a resistor load, ie it is open.

**Operand** – a piece of data on which some operation is performed. Usually the operand will be a byte in the accumulator of the 6502, or a byte in some memory location.

**Page** – a page of memory in the 6502 memory map is & 100 (256) bytes long. There are therefore 256 pages in the entire address space. 256 pages of 256 bytes each account for the 65536 bytes of addressable memory.

**Parallel** – parallel data transfers occur when data is sent along two or more lines at once. The system data bus for example has eight lines operating in parallel.

**Peripheral** – any device connected to the 6502 central processor unit, such as the Plus 1, Plus 3 interface etc., but not including the memory.

**Poll** – most of the hardware devices on the Electron expansion modules will generate interrupts to the 6502 CPU. If interrupts have been enabled, the CPU has to find out which device generated the interrupt. It does this by successively reading status bytes from each of the hardware devices which could have caused an interrupt. This successive reading of devices is called polling.

**RAM** – (Random access memory) the main memory in the Electron is RAM because it can be both written to and read from.

**Refresh** – all of the RAM in the Electron is dynamic memory. This means that it has to be refreshed every few milliseconds so that data is not lost. The refreshing function is performed by the ULA as it accesses memory regularly for video output.

**Register** – the 6502 and the Electron ULA contain registers. These are effectively one byte memory locations which do not necessarily reside in the main memory map. All software on the 6502 makes extensive use of the internal registers for programming. The bits in most peripheral registers define the operation of a particular piece of hardware, or tell the processor something about that peripheral's state.

**Rollover** – this is a function provided on the keyboard to cope with fast typists. Two keys can be pressed at once. The previous key with a finger being removed, and the next key with the finger hitting the key. The software in the operating system ensures that rollover normally operates correctly.

**ROM (Read only memory)** – as the name implies, ROM can only be read from and cannot be modified by being written to. The MOS and BASIC are contained in one large 32K byte ROM chip.

**Serial** – data transmitted along only one line is transmitted serially. Serial data transmission is normally slower than parallel data transmission, because only one bit instead of several bits are transferred at a time. Communication with the cassette interface is carried out serially.

**Stack** – a page of memory in the 6502 used for temporary storage of data. Data is pushed onto a stack in sequence, then removed by pulling the data off the stack. The last byte to be pushed is the first byte to be pulled off again. The stack is used to store return addresses from subroutines, Page &01 is used for the stack in the Electron.

**ULA (Uncommitted logic array)** – this large chip is responsible for most of the system control on the Electron. It contains a large number of logic gates. The connection between the gates is defined when the chip is manufactured.

# Index

!BOOT status 73  
 \*. 14  
 \*/ 14  
   filing system call 106  
 \*BASIC 14  
 \*CAT  
   filing system call 108  
 \*CODE 14,49,113  
 \*EXEC 14  
   close files 38  
   file handle 67  
 \*FX 15  
 \*HELP 15  
 \*KEY 15  
 \*LINE 15,113  
 \*LOAD 15  
 \*MOTOR 15,50  
 \*OPT 15,50  
   filing system call 106  
 \*ROM 15,51  
   data format 176  
   example ROM 178  
   get byte call 175  
   initialise ROM call 158,174  
 \*ROM filing system 172  
 \*RUN 15  
   filing system call 108  
 \*SAVE 15  
 \*SPOOL 15  
   close files 38  
   file handle 67  
 \*TAPE 15,51  
 \*TV 15,52  
 1MHz bus 217  
 1MHz clock generation 213  
 6502  
   stack area 187  
 6502 clock speed 212

## A

ADC 31  
   channel read 44  
   conversion complete event 120  
   conversion type 64  
   current channel 63  
   maximum channel number 64  
 Arguments (files) 98  
 Auto-boot  
   ROM call 154  
 Auto-repeat  
   delay 28,66  
   period 29,66

## B

BASIC  
   paged ROM socket 63  
 BEL  
   channel 72  
   duration 73  
   frequency 73  
   SOUND information 72  
 Blank/restore palette 36  
 BPUT  
   fast tube 55  
 BREAK  
   effect 68  
   interception 84  
   last type 85  
 Break-points 117  
 BRK  
   paged ROM active 63  
   Service ROM call 155  
   vector 116  
 BRKV 116  
 Buffers  
   character entry event 120





External hardware	207	Get character at cursor	49
		from buffer	52
F		from input stream	10
		GSINIT	12
		GSREAD	12
Fast tube BPUT	55		
File options select	50		
Files		H	
attributes	96,98	Handle	
close SPOOL/EXEC	38	filing system	108
EOF check	106	Hardware	
EXEC handle	67	external	207
open/close	105	internal	197
read byte	100	introduction	192
read/write	95	Hardware scroll example	199
read/write group of bytes	102	High-order address	47
SPOOL handle	67	HIMEM	190
system calls	94	read	48
write byte	101		
Filing system		I	
deselect	108	I/O read/write	53
handle range	108	I/O processor	
initialise	160	read memory	89
*ROM	172	write memory	90
workspace claim	153	INKEY	45
zero page workspace	186	Input buffer full event	120
Filing system calls	94	Input character interpretation	76
Firm keys		Input line	88
language call	149	Input source flags	59
pointer	69	Input stream	
status	77	selection	23
string	70	Insert value into buffer	127
Flag		INSV	127
*ROM/*TAPE	62	Internal hardware	197
printer destination	84	Interrupts	135
RS423 control	65	delays	213
RS423 use	64	example	141
Tube presence	80	interception	139
user	23,82	ROM call if unknown	155
Flashing colours		ULA mask	69
counter	65	vectors	119
mark duration	27,66	Interval timer	89
reset cycle	54	Interval timer event	121
space duration	28,65	IRQ	
Flushing buffers	30,34	input pin	210
FRED	53,217		
G			
Get byte (OSBGET)	100		

ULA register	197		
IRQ1V	119,139		
IRQ2V	119,139		
<b>J</b>			
		blank/restore palette	36
		claim service ROM call	158
		input pin	210
		release service ROM call	158
		routine area	189
JIM	53	zero page workspace	185
<b>K</b>		<b>O</b>	
Key number table	39	One megahertz bus	217
Keyboard		One megahertz clock generation	213
auto-repeat delay	28,66	Operating system	
auto-repeat period	29,66	calls	9
disable	68	commands	14
scan	40,46	high water mark (OSHWM)	47
soft key status	77	variables	56
status byte	69	vectors	110
status LEDs	37	workspace	81,85
translation table address	58	zero page workspace	186
vector	125	Operating system call summary	230
Keys pressed information	38	OS commands	14
KEYV	125	OS version	22,46
<b>L</b>		OSARGS	98
Language		OSASCI	11
exclusive workspace	188	OSBGET	100
zero page workspace	185	OSBPUT	101
Language entry	51	OSBYTE	16
Language ROMs	148	summary	18
Line filling	222	OSCLI	14
Line input OS WORD	88	OSEVEN	14
<b>M</b>		OSFILE	94
Memory clear on BREAK	68	OSFIND	105
Memory usage	183	OSFSC	106
MODE		OSGBPB	102
read	49	OSHWM	190
<b>N</b>		primary	60
		read	47,60
		soft character explosion	32
		OSNEWL	11
		OSRDCH	10
		OSRDRM	13
		OSWORD	87
		summary	87
		OSWRCH	9
		Output buffer empty event	120
		Output stream	
NETV	123	read/write	81
NMI	136	selection	24

## P

PAGE	47
Paged mode lines	74
Paged ROMs	143
active at BRK	63
allocation	217
BASIC socket	63
copyright string	147
current language number	85
enter language	51
extended vectors	171,189
firm keys	149
header format	144
info table address	57
issue service call	52
language entry	144
language ROMs	148
OS commands	15
paging register	215
pointer table address	57
polling semaphore	35,62
priority (Plus 1)	217
read byte from selection	13
selection register	215
selection register	202
service entry	145
service ROMs	152
title string	146
Tube relocation address	147
type byte	145
version number	146
version string	147
workspace table	189
Paged ROMs connector (Plus 1)	232
Palette	
blank/restore	36
read	92
ULA register	54,205
write	92
Pixel value	91
PLOT numbers	221
Plus	1
page &D usage	190
printer buffer example	162
ROM connector	232
ROM priority	217
Polling	
semaphore	35,62

service ROM call	160
POS	49
Printer	
buffer example	130
character in buffer ROM call	160
destination flag	84
driver going dormant	42
ignore character	26,84
output destination selection	26
user vector	121

## R

Read byte from ROM	13
Read character (OSRDCH)	10
Read character definition	91
Read input line	88
Remove value from buffer	127
REMOV	127
Reset output	210
pin	
ROM accessing	212
ROM connector (Plus 1)	232
ROM filing system	
select	51
ROM/Cassette flag	62
RS423	
baud rate	27
control flag	65
error event	121
mode	61
use flag	64
workspace	188

## S

Screem memory	191
Screen	
blank/restore palette	36
pixel value	91
Screen mode dependent clock	212
Screen mode layouts	223
Select input stream	23
Select output stream	24
Serial ROMs	172
Service call semaphore	35,62
Service ROM call	52
Service ROM calls	
*HELP	156



USERV	113	interrupt	119
execute code	49	interrupt	139
Utility zero page workspace	186	keyboard	125
		summary	230
		unused	134
		user	113
		user print	121
		VDU extension	124
<b>V</b>		Version	
VDU		operating system	22
abandon queue	75	operating system	46
extension vector	124	Vertical sync	
paged mode lines	74	event	120
queue items	75	wait	32
read graphics cursor positions	93	VPOS	49
read palette	92		
read status	37		
read variable	56		
variables origin	58	<b>W</b>	
write palette	92	Wait for vertical sync	32
VDU code summary	219	Write a new line (OSNEWL)	11
VDUV	124	Write character (OSASCII)	11
Vectors	110	Write character (OSWRCH)	9
BRK	116	<b>Z</b>	
buffer maintenance	126	Zero page usage	184
default table	134		
Econet	121		
event	119		
extended	171		