

Glossary

Accumulator – a register used to perform mathematical operations. The 6502 has one accumulator, A, which can deal with 8-bit integers.

Addressing Mode – specifies how any data will be used by a machine code instruction.

ASCII (American Standard Code for Information Interchange) – the ASCII code of a character is the value of the byte which is used to store it in the computer.

Assembler – a program which converts a series of mnemonics into a machine code program.

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.

BNF (Backus Naur Form) – a way of writing down the syntax of a computer language.

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.

Command – similar to a BASIC statement, but it can only be executed if it is typed in at the keyboard directly (i.e. in *command mode*), rather than as part of a BASIC program. For example, 'AUTO' is a command.

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

Disassembler – a program which converts a series of bytes in a machine code program into assembler mnemonics.

Field – a space allocated for some data in a register, or in a program listing, or in a storage area. For example, in a Variable Descriptor Block, the first field contains a pointer to the location of the variable, and the second field contains the type of the variable .

Flag – a bit (or byte) which is used to signal a particular condition. For example, the N (negative) flag in the 6502 is set if the number just calculated is negative.

Heap – BASIC uses a HEAP to store the variables used during a program. Data can be added on top of a heap, but once used, the space cannot be recovered until it is completely cleared.

High – sometimes used to designate logic ‘1’ Indirection – pointing to a variable in memory with the indirection operators ?, ! or \$, rather than using a value directly. For example, !&4000 points to the 4-byte integer variable in locations &4000 to &4003.

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.

Keyword – a special word (sometimes called a Reserved Word) which BASIC uses for a special purpose. For example, PRINT is a keyword which is put before items to be printed out.

Linked list – a list of items in memory, where each item contains a pointer to the next one. The end of the list is usually marked by a null pointer in the last item. A base pointer is used to point to the first item in the list.

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*.

Opcod – the name given to the binary code of a 6502 instruction. For example, &AD is the opcode which means *load accumulator* (absolute addressing).

Operand – a piece of data on which some operation is performed. This could be a number in a BASIC prog -a , or it could be a byte in the accumulator of the 6502.

Operator – a symbol or device which takes one or two operands to produce a single result. If an operator takes one operand, it is a *unary* operator; if it takes two operands, it is a *binary* operator. For example, the '\$' operator takes the number following it, and gives as a result the static string at that location.

Overflow – a condition caused when the result of a calculation is too large to be represented properly.

Overlay – a part of a program which is loaded into memory while the main program is running. Large programs can be run in a computer by splitting them up into several overlays, and each one will only be loaded in when they are needed.

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.

Page zero – the locations from &0000 to &00FF. These are very useful on the 6502, because any machine code instructions which use them are shorter and faster than those which use any other section of the memory.

Peripheral – any device connected to the 6502 central processor unit, such as the printer port, disc interface etc., but not including the memory.

Program – a BASIC program is a sequence of statements which the BASIC interpreter is to execute one after the other. A machine code program is a sequence of bytes which the 6502 is to execute one after the other as machine code instructions.

RAM (Random Access Memory) – the main memory in the BBC microcomputer and the Electron is RAM because it can be both written to and read from.

Register – a location which can be written to or read from, usually for a special purpose, but which is not necessarily in the main memory map of the computer. The 6502 and peripheral devices contain registers, and BASIC uses a series of page zero locations as if they were its own registers.

ROM (Read only memory) – as the name implies, ROM can only be read from and cannot be modified by being written to.

Stack – the 6502 and BASIC each use a stack 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 6502 stack is used to store return addresses from subroutines; the BASIC stack is used to store temporary results during a calculation, and other data inside a PROC or FN call.

Statement – a sequence of symbols which tells the BASIC interpreter to perform a certain action. For example, the statement 'A= 10' tells BASIC to assign the value 10 to the variable 'A' (this is an *assignment statement*).

Static string – a string whose characters are stored in memory starting at a fixed location. The string is terminated by a &0D byte (carriage return character), which is not counted as one of the characters of the string. For example, \$&2000 is the static string whose first character is stored in location &2000.

String Information Block – this block is used to reference the characters of a dynamic string on the BASIC HEAP. It contains a pointer to the start of the string, the amount of memory allocated to the string, and the current length of the string. The *String Information Block* is held in the *value field* of the *Variable Information Block* of a string variable.

Token – a single byte which is used by BASIC to represent a keyword. This saves memory when programs are stored. For example, &80 is the token for 'AND'.

Variable – is used to hold a number or a string (depending on its type). Named variables are stored on the BASIC HEAP (or in page 4 if they are resident integer variables), but indirected variables (accessed using the \$, ? and ! operators) can be anywhere in memory.

Variable Descriptor Block – this is passed between routines inside BASIC as a description of a variable, once its location and type has been found. It consists of a pointer to the value of the variable, and a byte which gives the type of the variable.

Variable Information Block – the format used to store variables (and FN/PROC locations) on the BASIC HEAP. It consists of a pointer to the next *Variable Information Block*, the name of the variable, and the value of the variable.