Version 3, November 2011 E1X2

Systems architectures for the Elliott 152, 153 and Nicholas computers.

Note: all references are listed in section E1/X5.

Systems architecture of the Elliott 152.

The Admiralty's Medium Range System 5 (MRS5), of which the 152 computer was a part, was the first major project to be undertaken at Elliott's new Borehamwood Research Laboratory. Out of it grew all subsequent Elliott involvement in digital computer design. The overall requirements of the MRS5 contract envisaged precision real-time digital control of a ship's anti-aircraft guns by a new type of accurate monopulse radar. Specifically, the radar was required to track an incoming hostile aircraft from a range of about eight miles, under the control of the 152 computer. Digital information on elevation, bearing and range was fed back to the 152, which then computed relevant trajectory prediction information and passed this digitally to the guns.

The MRS5 contract ran from October 1946 until the spring of 1950 when it was reduced in scope by the Admiralty. The significant point about the contracts was that they envisaged real-time, on-line, digital control at a point in history when analogue control was the norm. Since the project was classified as *secret*, technical information was written up in secret Borehamwood Internal Reports that are generally unavailable today. The information has, however, been summarised in the open literature in references 1 and 2.

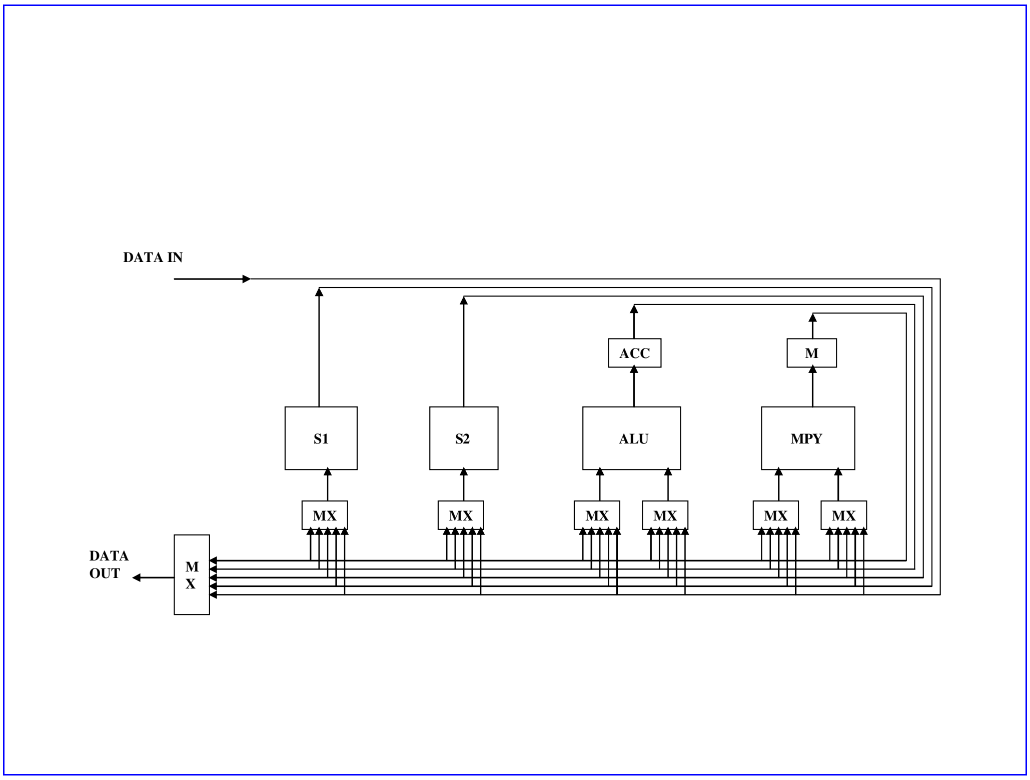
The first version of the 152 computer was completed in September 1950. It contained a fast multiplier; a single-word accumulator made from a circulating loop of delay elements with facilities for addition, subtraction, shift and collation (*AND)*; and a double-word (40-bit) accumulator. Data transfers were bit-serial. Arithmetic and logic circuits were based on sub-miniature pentodes (vacuum tubes). Storage was provided by 64 words of CRT (electrostatic RAM) memory, using the anticipation-pulse system invented by F C Williams at the Telecommunications Research Establishment (and later at Manchester University) – see reference 3.

The 152's arithmetic and routing sections occupied 280 glass printed-circuit boards (pcbs). A complete binary adder stage occupied one 5x3-inch pcb; the multiplier occupied 60 pcbs. The CRT store, addressing logic and program instruction generator used conventionally-wired circuits on 19-inch racks. The enhanced machine (March 1951), for which the (double-length) accumulator was duplicated, occupied 350 pcbs. The memory was arranged as two independently addressable banks. Each of two 6-inch CRTs held 32 words, giving a total RAM capacity in modern terms of 128 bytes (reference 302). This was the working store; program instructions were held separately (see below).

The physical appearance of the 152 is shown in the picture in the *Photo Gallery.* Each pcb chassis (or 'box') contained 18 pcbs; each rack contains between 3 and 8 chassis. An 8-chassis rack measured 6 feet x 31 inches x 5 inches and could contain up to 144 pcbs with approximately 600 sub-miniature valves (reference 158). All circuits were designed to work with HT supplies of +100v and –150 v, and bias supply of 6v. In the complete machine, 75% of the total pcbs were of just six basic types, the remainder being of special types.

The 152 employed 16-bit words for data and a digit period of 3 microseconds. The limiting factor determining the digit-period was the maximum speed at which the CRT store could be made to work reliably. Having fixed the digit-period, the next challenge was to design a fast multiplier. Using a series/parallel 'wiffle-tree' design, a multiplication in the 152 took 60 microseconds. The basic control cycle of the machine was therefore based upon a 'beat' of 60 microseconds, corresponding to 16 digit-periods plus a four-digit gap. Serial addition and other ALU operations could be performed within this 60-microsecond period. Instructions were 20 bits long, the program being held in separate ROM (see below).

If used in the conventional single-instruction, single-data mode of computer operation, the 152's arithmetic unit would not have given sufficient performance for real-time digital fire control. Therefore, provision was made for a number of instruction streams to be executed in parallel and for multiple simultaneous data-transfers to take place between the two RAM sections and various functional units within the CPU. A simplified conceptual diagram of the data paths is shown in the accompanying diagram, in which address, instruction and control signals have been omitted for clarity. In the diagram, S1, S2 are independently-addressable blocks of RAM; MX are multiplexers; ALU is the fixed-point serial Arithmetic and Logic Unit which provides addition, subtraction, AND, and shift operations; MPY is the fast multiplier unit. 'Data in' comes from the radar, which was also known as the Director in the Borehamwood reports. 'Data out' goes to the ship's guns.   Input signals from the ship's stable element (the gyro) has been omitted for clarity.



Conceptual view of information paths in the Elliott 152 computer.

In the 152 computer, each of four programs was stored in a read-only memory so that the calculations for bearing, range and elevation could proceed simultaneously. Other sequences co-ordinated the overall trajectory-prediction activity and controlled input and output to/from the 152. Each of the fixed programs was held on either a glass slide or a stiffened card (both systems were tried) with binary ones and zeros being represented by clear or opaque spots. The slides were read by a CRT flying-spot scanner.

There were two coaxial cables, connecting computer and radar: one carried a signal derived from the computer's clock, the other carried duplexed commands and data. On receipt of a command, the radar unit responded by sending back an integer. The computer sent 1,000 pulses per second to the radar (one pulse every millisecond). It was assumed that perhaps one pulse in three would not yield a meaningful return (because of glint, etc.), so there was a need to average over four radar returns when computing trajectories. Therefore, the 152 was required to perform the necessary axis-conversions and to produce one update of {range, bearing and elevation} every 4 milliseconds. Each of the {range, bearing, and elevation} parameters was represented to 14-bit accuracy, as were the sine and cosine of the bearing. 14 bits gave a precision of better than one minute of arc. In addition to any control and/or addressing commands, the data-rates coming into and out of the 152 computer were thus:

14 x 5 x 1000 = 70,000 bits/sec. inwards;
14 x 3 x 250 = 10,500 bits/sec. outwards.

In the sense that the programs were fixed, the 152 was not a *general-purpose* stored program computer. It was, however, probably the first tentative example of real-time, online, digital control. In America, the MIT Whirlwind real-time stored-program computer was first working reliably enough to do useful work by March 1951.

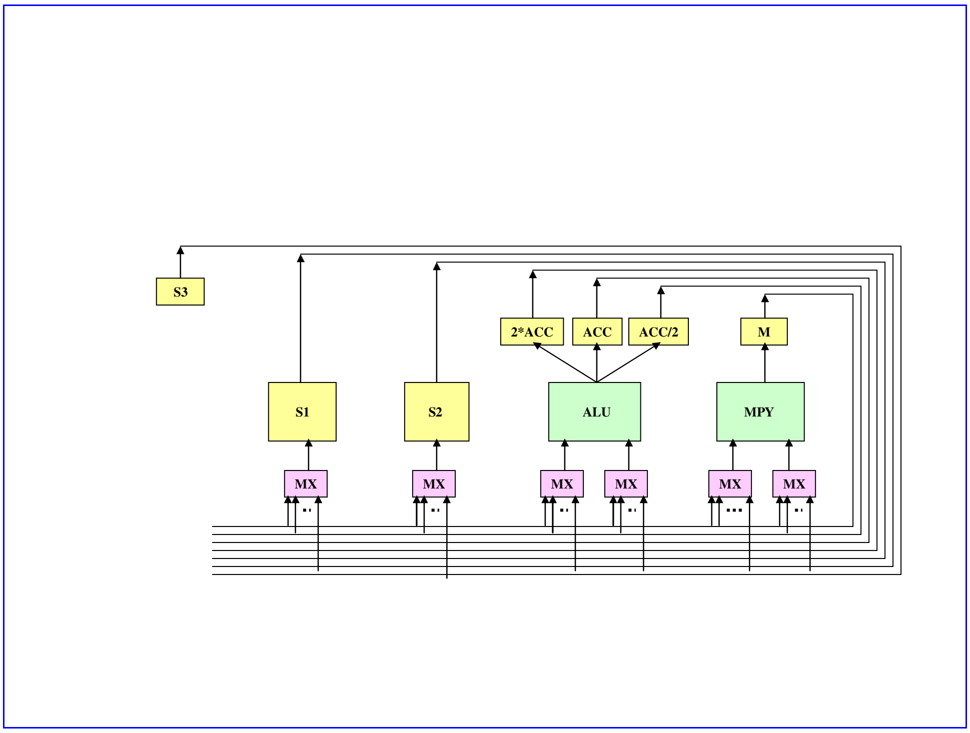
The 152 computer was, as John Bunt (who worked on the machine) has written, "a machine of remarkable power (when it worked), which employed every conceivable advance in technology known to man at the time. Printed circuits on glass plates, plated through holes, [deposited resistors], semiconductor diodes, plug-in units and so on. Some notable advances in techniques had been achieved in its construction, but unfortunately in a machine of its size, it was more often suffering from a breakdown than in robust health" (reference 1).

Systems architecture of the Elliott 153 – (the DF Computer).

The architectural philosophy of the 153 computer follows that of the Elliott 152, in respect of provision for multiple simultaneous data-transfers within the CPU. Like the 152, the Elliott 153 was classified as *secret,* so reference 1 is the only easily-accessible source of technical information. When compared with more conventional designs, the 153 achieves what would, 20 years later, be called *low-level parallelism* because it permits more than one distinct arithmetic operation to be performed within each instruction. An example is given in section E1X3.

The 153 employs 16-bit fixed-point data, for much the same trigonometrical reasons as the 152 and Nicholas (see below). Instructions in the Elliott 153 are 64 bits long, so as to allow multiple activities per order – again, reflecting the 152's philosophy. The digit-period is 3 microseconds because, in the initial design, the 153 was to have used the same CRT electrostatic memory as the Elliott 152. The time for every instruction in the 153 is 72 microseconds, being 16 digit-periods plus a gap of eight digit-periods.

The internal data paths for the 153 are illustrated conceptually in the accompanying diagram. The data-multiplexers, labelled MX in the diagram and referred to as *routers* in the original documents, are controlled by bits in each instruction, in the style of *microprogramming* – (see section E1X3). With the exception of parallel instruction fetching from disc, all data-transfers are bit-serial.



Interconnection of sub-units within the CPU of the Elliott 153.

For details of permissible interconnection, refer to the assignment of bits in the 153's instruction as described in section E1X3. We may summarise the allowable inputs to the Accumulator (ALU) and to the multiplier as follows:

{ a, hkys, S1, S2, S3, M, 2a, a/2 } .... { I/P 1 to ACC }

{ zero, S1, S2, S3, M, 2a} .... { I/P 2 to ACC }

{ 1, m, S2, S1, S3, a, 2a, a/2 }............................... { I/P to multiplicand }

{ retain, m, S2, S1, S3, a, 2a, prog.sel. hkys } .... { I/P to multiplier }

The 153’s Multiplier is a series/parallel (wiffle-tree) design, giving a single-length product within one instruction-time.

The 153 has two banks of random-access memory, shown as S1 and S2 in the diagram. These were implemented as CRT electrostatic memory initially, but were soon changed to

Elliott's more reliable nickel delay-line storage system. Each of S1 and S2 contains 34 words: 10 are implemented as single-word delay lines, and the remaining 24 words are addressed as six groups of four words – ie, they are implemented as four-word lines. The DF calculation algorithms requires 'about 60 to 70 words of fast storage'. This could have been met by two memory systems, each of 32-word capacity. However, each would have required 5 address bits. At the time, this was felt to consume too many bits in the 153's instruction. It was therefore decided to limit each store to 16 addressable locations, 10 of which were single-word and the rest of which could be up to four words per addressable location. This arrangement permitted hardware economies, whilst not significantly adding to an already complex instruction format. For each of the quad-word locations, any one of the four words is only available to software every fourth cycle-time. Therefore, the programmer has to take note of the running-order of a sequence of instructions before using one of the quad-words. Because the 153 is bit-serial, it can be arranged that a new operand may be written into an address at the 'same' time as the original contents are being read out.

S1 and S2 are primarily used as operand caches. The 153's main store is a magnetic disc with multiple fixed read-write heads, similar in concept to the magnetic drums used on several early computers. The disc holds program instructions, test programs, DF station parameters, operational data and run-time working-space. Read/write block transfers between the disc and the RAM caches are under programmer control. Refer also to the overall system diagram given in section E1/X4.

The 153's disc is 15.25 inches (38.74 cms) in diameter, made of duraluminium coated with iron oxide. It spins at 3000 rpm. Phase-modulation recording is employed. Each disc track has 256 x 24-bit locations – (16 bits plus an 8-bit 'gap'). The disc is organised as follows. Side A has 48 tracks for program storage, plus a clock track and an address track. The 48 program tracks are organised as 12 groups of four. Two tracks are used for an input routine (Initial Orders) and for test programs. A group of four tracks can be read simultaneously, thus allowing one quad-word instruction to be read per machine cycle. Side B of the disc has 40 tracks. These are employed for storing incoming DF records associated with up to 255 'target incidents', station constants for up to 20 DF stations, and a directory. Fast electronic track-selection is used for side A; slower relay-tree selection is used for side B. Side A is normally read-only; side B is read/write. The total disc storage in modern 8-bit bytes, as seen by a programmer, is about 45 Kbytes, calculated as: (48 tracks x 256 x 2) + (40 tracks x 256 x 2).

The 153's central processor (CPU) technology is based on the use of sub-miniature pentode thermionic valves (vacuum tubes) and semiconductor diodes. Type CV466 pentodes and type CV425 germanium diodes predominate. Use is made of a family of standard plug-in circuits, equivalent to the printed-circuit board, or package, of the 1970s. Each of the 153's packages is roughly the size of a small paperback book. The Borehamwood Laboratories of Elliott Brothers had been leaders in the field of printed-circuit development since about 1948. A wiffle-tree multiplier gives the 153 its required speed for multiplication. The CPU consumes about 9.2 KW of power, provided by a bank of trickle-charged batteries. The CPU is water cooled, with forced-air cooling. The whole installation occupies 15 cabinets, each being the size of a modest wardrobe.

Systems architecture of the Elliott Nicholas.

Somewhat untypically for historic computers, the details of Nicholas software are easier to deduce from surviving documents than the hardware details. Indeed, the facts surrounding its birth (see reference 4) suggest that there would have been little in the way of formal hardware documentation for Nicholas. Certainly, the definitive list of early Borehamwood Internal Reports only mentions one Nicholas publication: a programming guide (see Reference 5).

Nicholas is a serial machine with a single-address instruction format. The design requirement was for a modest, general-purpose, computer that could be constructed rapidly from existing Borehamwood know-how, to a fixed cost. The architecture is, at first sight, straightforward by the standards of 1952. There are three hardware-related factors of interest:

(a) the elegant logical design by Charles Owen, which required substantially no modification from inception to completion, and the short time of five months between the decision to built and the running of the first test program (reference 6);

(b) the predominant use throughout the computer of just three types of standard logic circuit;

(c) the simple, low-level hardware, support for subroutines.

The Nicholas word length is 32 bits, two instructions being packed into every word. Addressing, however, is always to a word boundary. Numbers are held as two's complement fractions, with the binary point being *two* places from the most-significant end. Ed Hersom has said (reference 4): "I decided that the binary point should be two from the most-significant end because I expected to be undertaking a lot of trigonometrical calculations such as axis conversions. The most-significant bit would be the sign, and the next to it the overflow bit (I never gave the idea of a separate overflow register a single thought, again on economy grounds). The overflow bit also allowed a function like sin(x) to equal 1, or even slightly more due to rounding error, without causing severer mistakes".

Magnetostrictive nickel delay lines are used exclusively for storage throughout Nicholas. There are five short lines (each 32 bits), used respectively for the accumulator, multiplicand, instruction-register, program counter and address-counter (also called *coincidence counter*). Apart from the single-length accumulator and the program-counter (called the *Order Number Register* in the original literature) Nicholas has no other programmer-accessible central registers.

The digit-period is 3 microseconds, this being the period that had been adopted for the first Elliott computer (the 152). The revised version of standard Borehamwood logic circuits, developed towards the end of the 152 project and start of the 153 project by Charles Owen, were used by Owen in the design of Nicholas (reference 1). There were basically just three types of circuit board, each based on sub-miniature pentodes with germanium diodes for the logic gating. The whole computer contained about 250 thermionic valves (vacuum tubes) (mainly of type VX8030/8046) and 250 germanium diodes. There were about 80 circuit boards in the complete Nicholas.

The main memory holds 1024 32-bit words (ie equivalent in modern terminology to

4Kbytes) in 64 long delay lines. A 65th line is used to control a master oscillator in order to minimise the effects of temperature variations on the effective time-delay of the memory lines. The physical unit of storage in the main memory is the 16-word long delay line, which consists of a loosely-coiled spiral of nickel wire, held by paper strips about half an inch above an 18-inch square aluminium plate (reference 4). A photograph of an experimental plate is shown in reference 12. 65 such plates, spaced vertically about one inch apart, are housed in a tall aluminium cabinet about six feet (1.85 metres) high.

For addressing purposes, it is arranged that eight 16-word lines are connected in series with repeater amplifiers between them to form a *loop*. There are therefore eight such 128-word loops in the complete Nicholas memory. This arrangement of loops gives economy of electronic logic circuitry at the expense of average access-time. This average access time is 64 word-times, whereas it could have been 8 word-times if the basic long delay lines had not been grouped into loops. (We note in passing that the same memory technology could even have been made *random-access*, if one-word short delay lines had been used throughout the 1,024-word store).

Input is via an electro-mechanical 5-track teleprinter paper tape reader, transferring at a rate of 26 characters/second. Output is either to a paper tape punch or to a page printer, at the standard teleprinter speed of 7 characters/second (reference 7).

There are 47 distinct operations in the basic Nicholas instruction set, including a ’conditional shift-and-add’ order used to good effect in the software multiply routine (reference 1). The instructions are described more fully in section E1X3 from the programmers’ viewpoint.

At the hardware level, the Nicholas arithmetic and logic unit (ALU) has a basic repertoire of about 13 actions. To describe these, we use the following notation:

a, a’: the old and new contents of the accumulator register;
x, x’: the old and new values of an operand at memory-address n;
t: a 5-bit character read in from paper tape to the l.s. end of the accumulator
p: a character sent to the teletype for punching/printing.

The 13 possible ALU actions are:

function	***explanation***	***optional side-effect***
a’ = a	no operation	(could also inhibit PC incr.)
a’ = 0	clear the accumulator	(could be made persistent)
a’ = -a	negate	(could also inhibit PC incr.)
a’ = x	load acc	
x’ = a	store acc	
a’ = a + x	add	
a’ = a – x	subtract	
a' = x – a	reverse-subtract	
a' = a & x	AND (ie, *collate*)	
a' = 2a	shift left (result may be neg.)	(could be made persistent)
a' = a/2	shift right (arithmetic)	(could be made persistent)
a' = t	input a character	
p = a	output a character.	

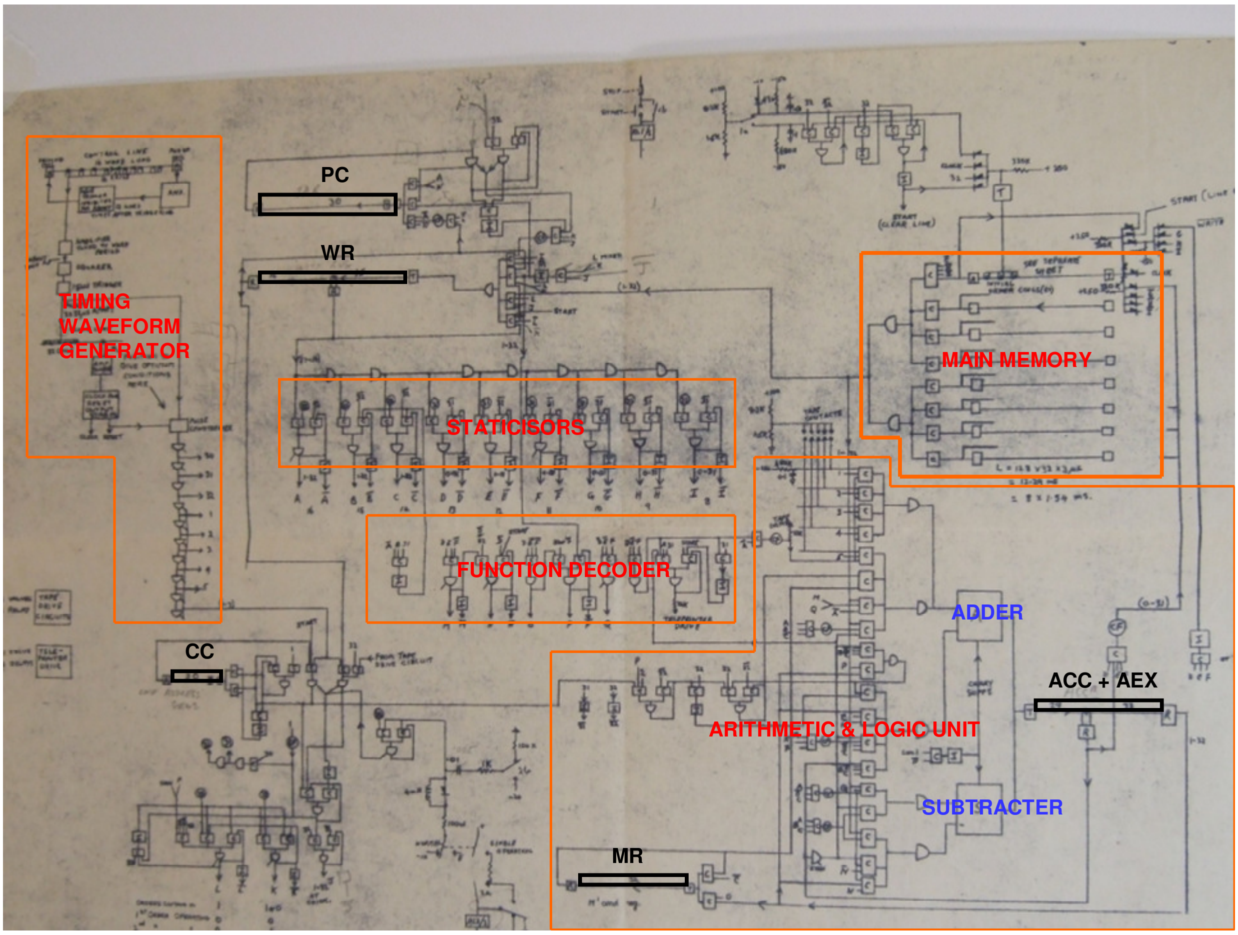
Some of these 13 actions could be combined – for example, 'shift and add'. Five of the actions could optionally be accompanied by one of two unusual side-effects:

(a) the normal incrementing of the Program Counter (or *Order Number Register)* could be suppressed at the conclusion of the instruction;
(b) the ALU action could be made to persist until the next successful jump instruction was encountered.

Side-effect (a) allowed a sequence of subsidiary instructions to be obeyed, at the conclusion of which the main program thread would be resumed. This is conceptually equivalent to the effect of a subroutine. Side-effect (b) was used by systems programmers for the efficient implementation of such functions as multiplication and division. More information on the Nicholas instruction set is given in section E1X3.

The original Nicholas logic diagram is reproduced below, with the following registers indicated:

PC	Program Counter	10 bits
WR	Working Register	16 bits
ACC	Accumulator	32 bits
AEX	Accumulator extension, used eg during multiplication	32 bits (+ 1 bit)
MR	Multiplicand register	32 bits
CC	Coincidence Counter, for matching addresses	10 bits.


PC
WR
TIMING WAVEFORM GENERATOR
STATICISORS
MAIN MEMORY
FUNCTION DECODER
ADDER
CC
ACC + AEX
ARITHMETIC & LOGIC UNIT
SUBTRACTER
MR