

## Chapter 7

### The Manchester Mark 1<sup>1</sup>

S. H. Lavington

Upon arrival at Manchester in December 1946, Williams and Kilburn set about perfecting a digital store, at first using the commercially available type CV1131 12-inch diameter cathode ray tubes [Kilburn, 1948; Williams and Kilburn, 1949]. The principle of a two-state electrostatic store can be visualised from the following simple experiment. Start with a focussed CRT beam and turn the beam current on (thus producing a charged “dot”) and off again repeatedly. *Negative* voltage pulses will be induced by capacitive coupling in a pick-up plate placed close to the outer surface of the CRT screen. Now move the beam whilst it is on so as to write a “dash” on the screen, then move the beam back whilst the current is off, and then switch on the current again. This time a *positive* voltage pulse is induced. With dots and dashes representing logical 0’s and 1’s, readable as negative and positive voltage signals, a binary storage system is available. Other representations such as a “focus/defocus” system were also used. Now although the electrostatic charge leaks away in about 0.2 seconds, automatic refreshing (re-writing) of the information in less than 0.2 seconds is a simple matter electronically. (cf a modern MOS solid-state store.) Since the refresh rate is rapid, long term drifts in electrode supply voltage, etc are not critical and a robust store can be made from standard components. In contrast, the mercury acoustic delay-line stores chosen by other workers had to be constructed to close physical tolerances. The biggest advantage of the CRT store was that it allowed random access whereas other contemporary systems were sequential.

By the Autumn of 1947 the Manchester group had successfully stored 2048 digits for a period of hours [Kilburn, 1948] and the way was clear for the construction of a prototype computer “to subject the system to the most searching tests possible” [Williams et al., 1951]. Kilburn took the initiative with the logical design. The “baby machine,” as it was called, had a specification which may be expressed in modern terminology as follows:

32-bit word length

Serial binary arithmetic using two’s complement integers

Single-address format order code

Main store: 32 words, extendable to 8192 words, random access

<sup>1</sup>Excerpted from S. H. Lavington, *A History of Manchester Computers*, NCC Publications, Manchester, England, 1975, pp. 7–10.

*Editor’s note:* Further discussion of the Mark 1 can be found in the above publication.

Computing speed: 1.2 milliseconds per instruction.

The instruction format had three bits assigned to the function field, 13 bits to the address field and the remaining 16 bits were unused. The main store consisted of a single CV 1131 Williams Tube, with each 32-digit line occupying about 10 cms on the screen and being scanned in 272 microseconds. A complete “beat” of 306 microseconds consisted of  $32 \times 8.5$  microsecond digit periods plus a four digit fly-back time. The rhythm of the whole processor was synchronised to this store beat. There was notional provision for extending up to 256 Williams tubes to yield a total storage capacity of 8192 words. The arithmetic unit was based on a serial subtractor and the logic employed EF50 pentode tubes, used widely for wartime applications. Using this technology, flip-flops (bistable circuits) were extremely costly and temporary storage throughout the central machine was implemented with Williams Tubes wherever possible. Thus the accumulator and control register (instruction counter) were Williams Tubes. One incidental advantage of the use of CRT’s was that the contents of main store, accumulator and control register could be viewed on a monitor CRT during or after a computation—so providing a simple output mechanism. Input for the prototype was via a 32-position keyboard and operators control switches.

The machine first ran a program in June 1948 [Williams and Kilburn, 1948] and as far as can be ascertained it was therefore the world’s first stored-program computer. A complete diagram of the prototype Mark 1 is given in Williams et al. [1951] and Fig. 1 is a simplified version showing the main flow of information. The Williams Tube which implemented the control register was also used to hold the present instruction (PI) itself subsequent to its being read out of main store. Either the value of control or the value of this PI could be fed from the “control” Williams Tube to

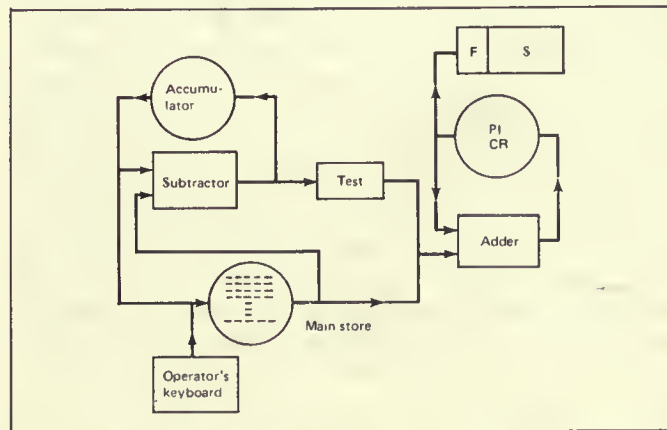


Fig. 1. A simplified diagram of the prototype Mark 1 showing the main information paths.

Table 1

Decimal value of function bits	An early notation	Modern mnemonic	Explanation of operation
0	s, C	JMP S	Absolute indirect unconditional jump: set the control register equal to the contents of address S.†
1	c+s, C	JRP S	Relative indirect unconditional jump: add the contents of address S to the control register.†
2	-s, A	LDN S	Load negative: set the accumulator equal to the negated contents of address S.
3	a, S	STO S	Store: copy the contents of the accumulator to address S.
4 or 5‡	a-s, A	SUB S	Subtract: set the new value of the accumulator equal to the former contents minus the contents of address S.
6	Test	CMP	Compare against zero: the value in the accumulator is tested. If it is less than zero, one is added to the control register thus causing the next sequential instruction to be skipped.
7	Stop	STP	Stop: cease automatic mode, and await manual commands from the operator's keyboard.

†Note that +1 was always added to the control register at the end of every order, so the programmer used JMP and JRP to point to an instruction one before the line he intended to jump to.

‡To economise on logic elements only partial decoding of the function bits was carried out.

an 8-bit (extendable to 16-bit) flip-flop register known as the staticisor. This staticised function bits (F) and operand addresses (S) during the execution of an order, and then staticised the address of the next order during the instruction-fetch phase. An interesting anomaly was that numbers were stored with the least significant digit on the left—a system which makes sense to engineers if not to mathematicians!

As has been noted, two's complement was used to represent negative numbers, though the now familiar rules for addition/subtraction and the formation of the complement of a number were not implemented in the usual way. The main emphasis of the project at this time was to prove the practicability of the Williams Tube for realising the stored-program concept and so the arithmetic logic was kept as simple as possible. The subtractor was the *only* arithmetic facility provided, it being preferred to an adder because a subtractor can be used without alteration to form complements and to perform additions whereas the converse is not true. As may be seen from Fig. 1 an operand entered the accumulator by being fed through the subtractor; this "subtract from zero" thus complemented the operand before it reached the accumulator. The effect of a positive load could be programmed in two orders by performing a negative load and then subtracting the contents of the accumulator from a zero operand. An addition of two quantities p, q was programmed by a four-order sequence as follows:

load negative p  
subtract q

store the resulting  $(-p -q)$

load negative to achieve  $(+p +q)$

The order code of the the prototype had provision for eight functions, specified as in Table I.

Three demonstration programs were run on the prototype machine, the first one involving determination of the highest factor of an integer by a method which would give a long run, the result of which could be easily checked. To quote [Williams and Kilburn 1948], "the highest proper factor of  $2^{18}$  was found by trying in a single routine every integer from  $2^{18}-1$  downward, the necessary divisions being done not by long division, but by the primitive process of repeated subtraction of the divisor. Thus about 130,000 numbers were tested, involving some 3.5 million operations. The correct answer was obtained in a 52-minute run. The instruction table in the machine contained 17 entries."

The original program was written by Tom Kilburn. G C Tootill, an engineer on loan to Manchester from TRE from mid-1947 to mid-1949, also wrote programs for the prototype and a notebook kept by him over the period 4th June to 28th November 1948 has survived. From entries in this notebook it seems that Kilburn's program was first run on Monday 21st June.

## References

Kilburn [1948]; Williams and Kilburn [1948]; Williams and Kilburn [1949]; Williams, Kilburn, and Tootill [1951].

## APPENDIX 1 MARK 1 ISP DESCRIPTION

```

MARK1 :=
  begin

! The Manchester Mark-I architecture is described.

! The Mark-I was an early (circa 1948) computer.

**MP.State**

  M[0:8191]<0:31>,

**PC.State**

  PI\Present.Instruction<0:15>,
    f\function<0:2> := PI<0:2>,
    s<0:12> := PI<3:15>,
  CR\Control.Register<0:12>,
  ACC\Accumulator<0:31>,

**Instruction.Execution**{tc}

  icycle\instruction.cycle{main} :=
    begin
      REPEAT
        begin
          PI ← M[CR]<0:15> next
          DECODE f =>
            begin
              #0 := CR ← M[s],
              #1 := CR ← CR + M[s],
              #2 := ACC ← - M[s],
              #3 := M[s] ← ACC,
              #4:#5 := ACC ← ACC - M[s],
              #6 := IF ACC lss 0 => CR ← CR + 1,
              #7 := STOP()
            end next
          CR ← CR + 1
        end
      end
    end
  end
end

```