The Instruction Set

Recorded by Tom Kilburn

Cliff, in a recent conversation you asked me if I would take time to explain how I came to choose the particular instruction set in the Baby machine. It is difficult to recall in detail, of course, over so many years what in fact the determining factors were, but I will do my best.

One of these was certainly whether to use a multi-address code, say a double address code, or a single address code. I was quite convinced that whatever virtues a double address code might have for a non-immediate access machine a single address code was certainly the one to choose for a computer with an immediate access store. It was more efficient both in time and space and it left open for us to have two instructions per line later in the year when we expanded the machine which would not have been possible with a double address code. For the time being, however, two instructions per line was too complicated in equipment and so I resigned myself to programs for the machine which could be contained within the 32 lines of one cathode ray tube. Having decided on single address code, a minimal instruction set might be as shown in the first column of the attached sheet of paper. To store the content of the accumulator is obviously essential. To logically combine the content of the accumulator with the content of a store line together with nought is the minimum operation for the accumulator and the conditional transfer of control must also obviously be provided. It was very tempting to stay with this set of three as it would require the minimum building effort. I decided, however, that a slight extension of this set would be worthwhile. This was because within the severe restriction of 32 lines I wanted the problems attempted to be natural to the non-specialist. The programs should be aimed at solving the problems rather than the tiresome deficiencies of the code. So, although it is not strictly necessary to provide a unconditional transfer of control as it can be programmed using the other instructions, it is prudent and worthwhile to do so, as in the second column. Incidentally, because the control register contains an instruction, CI, whose address part is a control number and whose function digits are 000 it simplifies the design to give the unconditional transfer of control instruction the function digits 000 also. Having provided the unconditional transfer of control the condition –or test– can be separated from the transfer as in the third column to give a slight simplification of implementation.

I turn now to the accumulator. With an accumulator which had only a logical instruction the problems which could be attempted would certainly seem contrived and the programs which were written to solve them would be tortuous in the extreme. I decided to replace the logical instruction by an arithmetic one, as in the fourth column. This was more expensive in equipment and, what was more important, more expensive in building effort; but it would more than repay the effort expended. To have both addition and subtraction I considered would be needlessly extravagant and a subtracter was chosen because it can change the sign of a number and be programmed to perform additions.

An adder cannot be programmed to change the sign of a number nor to subtract. Having provided a subtracter, it is trivially easy and worthwhile to implement load negative as in the fifth column. A touch of elegance appears in column six with the invention of the relative transfer of control; it is certainly a luxury and an indulgence on my part. It is a refinement allowing the number in the store address, S, to remain the same irrespective of the position of a sub routine in the store. Here the number in a stored address is added to an address before that address is used. It is no surprise that the B-tube, or index register, was invented a few weeks later and I never had reason to regret the indulgence.

Finally, to column seven and the stop instruction. Why have a stop instruction? Why not?

Transcribed by Joanne Allison

04 July 1996

f							
0		s,C	s,C	s,C	s,C	s,C	s,C
1						c+s,C	c+s,C
2					-s,A	-s,A	-s,A
3	a,S	a,S	a,S	a,S	a,S	a,S	a,S
4 or 5	$\overline{a\&s}$,A	$\overline{a\&s}$,A	$\overline{a\&s}$,A	a-s,A	a-s,A	a-s,A	a-s,A
6	Test:s,C	Test:s,C	Test	Test	Test	Test	Test
7							Stop
	(1)	(2)	(3)	(4)	(5)	(6)	(7)

16 May 1996