

Manchester Baby Simulator Narration

By David Sharp
The University of Warwick
11th January 2001

19/6/48 *Routine* (should take about $\frac{1}{2}$ acc. X465 acc. time)

Step		C	29	30	31	RMX			
1	30, C	\bar{a}_n	\bar{a}_n	\bar{b}_n	\bar{b}_n	$\bar{b}_n = a_n$	01111	010	
2	C, 29	\bar{a}_n	\bar{a}_n	\bar{a}_n	\bar{b}_n		10111	110	
3	31, C	\bar{b}_n			\bar{b}_n		11111	010	
4	C, 31	\bar{b}_n		\bar{a}_n	\bar{b}_n		11111	110	
5	31, C	\bar{b}_n		\bar{a}_n	\bar{b}_n		11111	010	
6	C, 30	\bar{a}_n		\bar{b}_n	\bar{b}_n		01111	110	
7	29, C	\bar{a}_n					10111	010	
8	30, C(-)	\bar{a}_n					01111	001	
9	Test							011	
10	27, 6l(+)						11011	100	
11	31, C(-)	\bar{b}_n	\bar{a}_n	\bar{b}_n	\bar{b}_n	$\bar{a}_n = \bar{b}_n$	11111	001	
12	C, 31	\bar{b}_n			\bar{b}_n		11111	110	
13	28, C(-)	\bar{a}_n					00111	001	
14	Test							011	
15	0, 6l							all 0	
16	Stop	\bar{a}_n	\bar{b}_n	\bar{a}_n	\bar{b}_n			111	
0	0	all 0							
27	3	10111 11111							\bar{b}_n is the HCF
28	2	01000 00011							$\bar{a}_n = 0, \bar{b}_n = 1$ coprime
29	1	11111 11111 11111 11111 11111 11111 11111 11111							
30	\bar{b}_n	11101	01100	10001	11100	11000	00010	0	
30	\bar{a}_n	10000	10100	00110	11001	11010	10010	0	

($\bar{a}_n = 314, 159, 265$, $\bar{b}_n = 271, 828, 183$, coprime)

Geoff Tootill's Highest Common Factor program, written in his notebook 19/6/48.

Recent background¹

Between 1996 and 1998 much research was done by the Manchester Computer Conservation Society in building an accurate replica of the ‘Baby’ (as of 21st June 1948) for its 50th anniversary. A competition was also staged for people to write programs for the ‘Baby’, the winning program was run on the replica on the 21st June 1998. This rebuild of the replica raised the profile of the Baby and also caused a lot of research into details of the machine that might otherwise have been lost to history.

Historical accuracy

The display presented by the simulator is a stylized version of the original June 1948 Baby. I have made the interface as much like the original as is possible using Java’s GUI capability. Great pains have been taken to provide an accurate rendition of the Baby’s controls and idiosyncrasies. As well as this original interface, I have borrowed from the “player” notion of Martin Campbell-Kelly’s EDSAC simulator (Raul and Hashagen, p. 400) in providing an alternative modern day interface for loading and running the ‘Baby’.

The similarities between the original machine and the simulator interface are shown below in Figure 1. It would be easy for someone to learn to operate the original machine from the simulator.

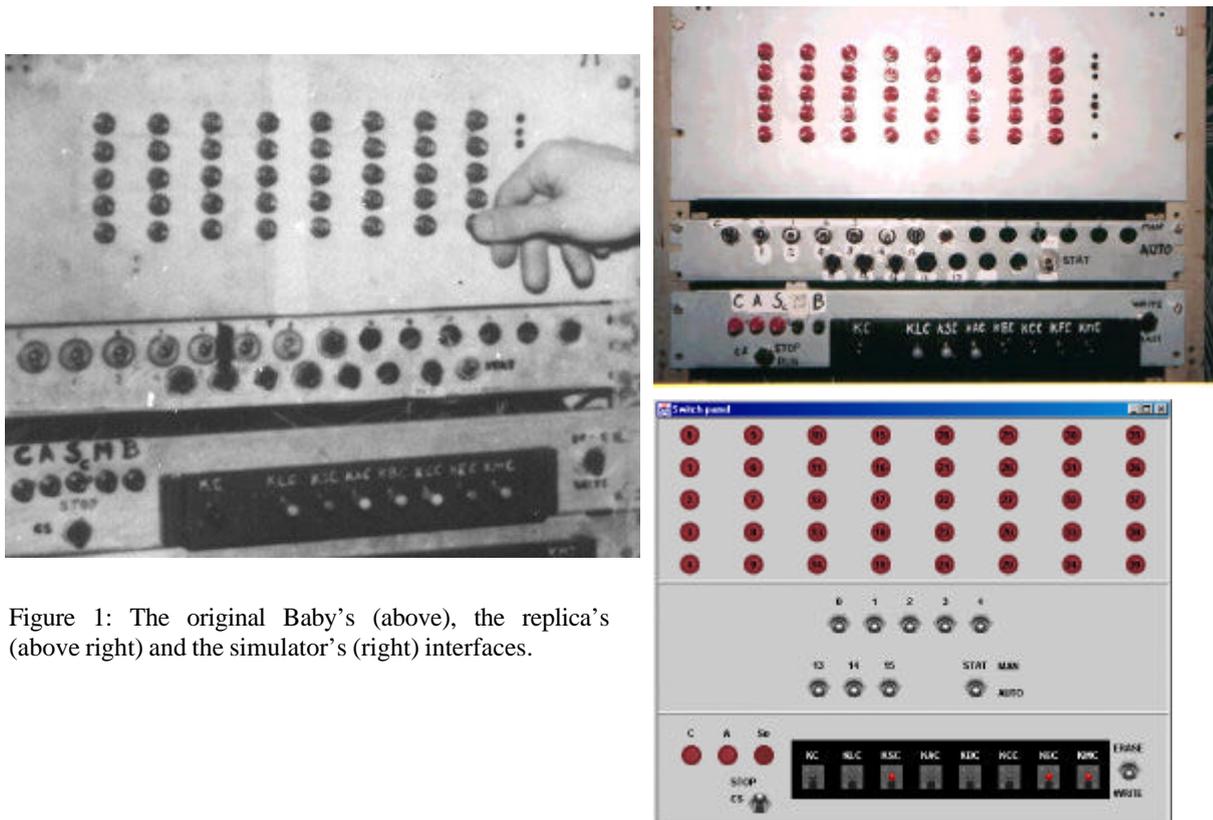


Figure 1: The original Baby’s (above), the replica’s (above right) and the simulator’s (right) interfaces.

¹ The history and widespread influence of the development of the Baby is outlined in the introduction to the user guide.

The level of historical authenticity achieved is extremely high and much better than any other previous simulator (and there have been several). This is mainly as a result of trying to cover the poorly defined boundary cases of the machine's operation² despite the fact that such discrepancies would have been barely noticeable when the machine is running.

Many issues and conflicting sources have arisen during the development of the simulator. The remainder of this document is a description of particularly poignant features that were taken into account, others that have not been considered but not implemented and a description of the sources of information available to me.

Switch panel

It's not clear whether any MAN/AUTO labels are on the STAT switch on the original though they certainly are on the replica (and in the wrong place, due to limited sources at the time of labeling). It is unclear whether erase is at top or bottom on the write/erase switch with secondary sources conflicting (MISIM and the replica)³. It's hard to tell on the original photo (Figure 1) though from looking at it I believe 'write' is at the bottom. The positions of the L and F staticisor switches are not clear on any photo, several holes and surplus switches have been left off my simulator though were unlikely to be present and certainly not used in June 1948.

What appears to be S on the button to select store to be displayed is, according to Chris Burton⁴, S₀ with the right part of the zero covered by masking tape (on both the original and replica).

The colours for the maroon typewriter buttons are sampled from the scanned photo (Figure 1) of the replica. I have attempted to make them appear slightly concave as the originals were.

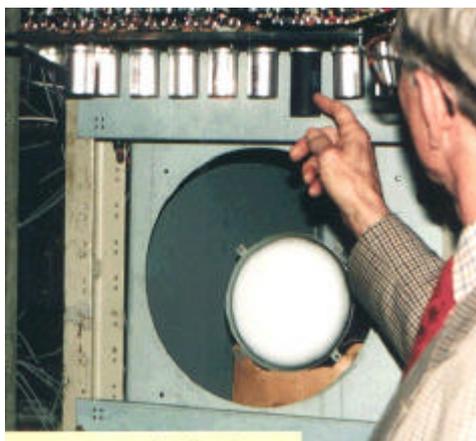


Figure 2: The display monitor on the replica.

The position and size of the stop lamp on the original is not clear so it has been placed to the left of the display monitor (mainly to reduce the size of the window). On the replica it is to the left of the CRT at about 10 o'clock near the upright of the rack⁵, see Figure 2 ("Both Geoff Tootill and Tom Kilburn have agreed that it is a plausible place though they can't guarantee it." – Chris Burton).

² For example, in manual mode when the 'present instruction' displayed is not the same as the instruction that is executed because that is taken from the staticisor switches.

³ See Appendix, question 21.

⁴ See Appendix, question 14.

⁵ See Appendix, question 16.

The three display select buttons are part of a unit where pushing one locks it in and releases the others, this is simulated when by pressing a button, the previously pressed button is released. Switches which were unconnected in June 1948, KBC, KEC (which clears all storage) and KMC (which clears a multiplier tube, added later) have been left on the diagram since their absence would cause the central black panel not to be representative of the original. In fact, even more true to the original than the replica (which has colours them inaccurately⁶) the colours of these black, grey and red switches is correct on the simulator. I have purposely positioned the STAT switch separated from the F staticisor switches as it was on the original.

The only known inaccuracies are the M and B switches which are present on the replica but not on the simulation⁷. They were present in June 1948 although were not actually used until later. So as to not enlarge the window too much unnecessarily, they have been left off the switch panel. Also, according to Chris Burton, it was only the KC switch which spring returns⁸ (i.e. has to be held down to remain in that position). On the simulator the other clearance switches also spring return as it would be too easy to leave them switched on and counter-intuitive to have to flick them off again.

Display

The CRT display was circular as is shown on the simulator - it was left resting on an empty valve box to keep it showing through the hole in the casing, the valve box is not displayed.

The display of the Control and Accumulator (misrepresented on other simulators) has been correctly performed. As a result, the Accumulator value is repeated down the display when selected, as are the CI (Control Instruction) and PI (Present Instruction) values when the control is displayed. The PI only existed when the machine was running and for that reason only the CI is displayed when the machine is stopped.

As described in the introduction as an example of the intricate simulation, when a manual instruction is executed (that is one encoded on the staticisor switches) the PI displayed is the one in the store addressed by the CI, not the one that is executed which is encoded on the switches⁹.

The action line is the line in the store that is specified by the L staticisor switches (when in manual mode) or the Control Instruction (when in automatic mode). When the CRT display is showing the store, the action line was always brighter than any of the other lines, this has been implemented correctly.

The only known inaccuracy is that the PI should be visible momentarily if the control is displayed when a single instruction is executed (by pressing KC) at the moment this isn't done.

⁶ See Appendix, question 18.

⁷ See Appendix, questions 12 and 13.

⁸ See Appendix, question 20.

⁹ See Appendix, question 23.

Execution

The exact speed of the original Baby was never firmly determined, approximations (in Chris Burton's Programmers Reference Manual, section 2.7) suggest 700 instructions per second and timings taken for historical programs seem to compare very well with the simulator if calculated at this speed. The HFR989 program is said to execute in 30 seconds on the Baby, the simulator's real-world timer also reports this.

Corruption of the control and store was not just possible but likely if parts of the switch panel were used when the machine was running. In simulating the machine I have accurately represented as many different types of corruption as possible. In general, holding down a key which affects a line while the Baby is running corrupts every line encountered while that key is held down. That also includes corruption of the PI as it is fetched from the store and any operand lines that are used by that instruction. Operations that affect in this way include typewriter keys setting or clearing bits, lines being cleared etc. It should be noted that a further complication to the simulation was that manual instructions (encoded on the switches) are not corrupted, as they are not fetched from the store, however their operands are.

It is essential that the L and F stat switches be all set to on (down) so that the instruction being fetched can get from the store to the PI. For any that aren't set, the appropriate bit is simply not fetched to the PI. Notice how all the switches necessary are set to the right configuration when using the modern controls.

There are a few known inaccuracies that have not been implemented because specifics of the corruption are not confirmed. It was thought better not to implement a corruption than to implement it incorrectly. It has been suggested that when the Baby runs in manual mode, the CI is incremented for each manual instruction executed, however it doesn't increment when just a single manual instruction is executed¹⁰. I have seen it described that if the write/erase switch is set to erase when running in automatic mode then all the action lines encountered are cleared (see Programmers Reference Manual section A3.3.3). However, on consultation the author was unsure of the accuracy of that statement so that feature has been left out. The only known inaccuracy is that if any of bits 13 to 15 (the function bits in PI) in the CI become set then the PI will become corrupted, exact details of this corruption are not available so this has only been implemented in part. Since there are so many combinations of corruptions, I would expect there to be a few other cases that are not correctly implemented though I have done my best to avoid this.

In concluding, the accuracy of the program execution is extremely good, much better than any other known simulator. In fact the simulator is such a marked improvement over previous efforts that Christopher Burton has suggested that the Manchester Computer Conservation Society might like to feature this simulator in their demonstrations.

¹⁰ See Appendix, question 26.

Sources

While many sources obtained in the writing of this simulator date from the late 1940s, most of those proved fairly impenetrable and at a lower level than required i.e. circuit diagrams. As a result, modern information from the rebuild has also been used in interpreting the historical sources and producing the simulator.

One of the main sources used in the rebuild was the copy of the circuit diagrams that D.B.G. Edwards and A.A. Robinson made into their notebooks upon joining the team in November 1948 (although many developments had been made since June). Geoff Tootill's notebook was also available, as were several photos of the Baby in various stages of development after June 1948. It is precisely this continued development of the same machine immediately after 21st June that has caused myself and others so many problems in accurately determining the original set up.

Sources used in the research of this simulator are outlined below. All Internet URLs are verified correct as of 10/1/01.

Web

Burton, C., *The Programmers Reference Manual*
<http://www.cs.man.ac.uk/prog98/ssemref.html>

Burton, Christopher, P., 'Baby's Legacy - The Early Manchester Mainframes',
<ftp://ftp.cs.man.ac.uk/pub/CCS-Archive/misc/>

Some programs are available as the winners from the 1998 programming competition:

<http://www.cs.man.ac.uk/prog98/prizewinners.html>

Other programs historic programs were made available by Chris Burton. See the 'Programs provided' section at the end of the User Guide for details.

Molyneux, Andrew, "MISIM". The simulator used for the programming competition though does not simulate the switch panel anything like correctly.

http://www.cs.man.ac.uk/prog98/alt_simulators.html

Bibliography

Burton, Christopher, P., 'Rebuilding the First Manchester Computer', pp. 379-386, *The First Computers – History and Architecture*, eds. Rojas, Raul and Hashagen, Ulf, Massachusetts Institute of Technology, 2000.

Campbell-Kelly, Martin, 'Past into Present: The EDSAC Simulator', pp. 397-416, *The First Computers – History and Architecture*, eds. Rojas, Raul and Hashagen, Ulf, Massachusetts Institute of Technology, 2000.

Campbell-Kelly, Martin and Aspray, William, *Computer – A history of the information machine*, BasicsBook, 1996, p. 100, 106, 126.

Lavington, Simon, *Early British Computer – The story of vintage computers and the people who built them*, Manchester University Press, 1980, pp. 36-43.

Articles

Williams, F. C., Kilburn, T. and Tootill, G. C., ‘Universal High-speed digital computers: a small-scale experimental machine’, IEE, *The proceedings of the institution*, vol. 98, part II, no. 61, February 1951. The version used is that transcribed to electronic format by Chris Burton, particular reference was made to figure 9.

In depth discussions via email with Christopher P. Burton who led the team building the replica of the Baby, see Appendix.

The Computer that Changed the World CD ROM, produced by Europress for the University of Manchester. Available from <http://www.computer50.org/mark1/cd.html>
Referenced CD material

Lavington, Simon, *A History of Manchester Computers*.

Video presentations by Christopher Burton, Geoff C. Tootill and Tom Kilburn.

Kilburn, Tom, ‘From Cathode Ray Tube to Ferranti Mark 1’, *The Bulletin of the Computer Conservation Society*, vol. 1, no. 2, Autumn 1990.

Williams, F. C. and Kilburn, T., ‘Electronic Digital Computers’, *Nature*, 3^d August 1948, no. 162, 487.

The entire program is written by David Sharp, the only other credit due is for some of the graphical icons which have been borrowed from MISIM. Thanks must also go to Christopher Burton for all the help and advice with historical accuracy, providing me with invaluable photos, giving good suggestions and testing the simulator.

Appendix

Extracts from emails between David Sharp (simulator author) and Christopher Burton (Baby replica team leader) between 27/12/00 and 10/1/01, reproduced with permission. This is present for completeness since it highlights my attempts to get every detail as accurate as possible and several obscure features that I could find no documentation about.

Q 1) I have been reading your reference manual with some interest but am curious about the sources of the information (is it just the IEE paper?).

A. The IEE paper is the primary source. It has been verified by examination of the actual circuits in the machine, which clarify the operation of the switches. Other verification, or perhaps better to say, confirmation, comes from Geoff Tootill's notebook in the Manchester Museum of Science and Industry. But I would say that the latter is not needed to get an accurate simulation.

Q 2) When you say that the IEE paper was 'verified by examination of the actual circuits in the machine', what do you mean?

A. Many of the relevant circuit diagrams exist in notebooks kept by DBG Edwards and AA Robinson. The logic structure is so simple that it is easy to make sure that the circuit diagrams and the programming definition are identical.

Q 3) Is there any way to get a look at Geoff Tootill's notebook or a copy of it?

A. The Museum of Science and industry in Manchester could make you a photocopy, but would charge you for it. You could probably examine the original by appointment. My own copy is very worn and uncopyable (I know, someone else asked.) Several of the pages are in facsimile on the CD-ROM "The Computer that Changed the World". Several are also in a paper by Brian Shelburne and me in Annals of the History of Computing July/September 1998.

Q 4) I note from the old competition website that this simulator was deemed officially representative of the baby, was this done methodically or just by a certain amount of experimentation?

A. Methodically. The author wrote a version. When we tried a few examples, we found that there were various errors, mainly in the Man Machine Interface. I wrote the Programmers Reference, confirming with colleagues that it correctly represented the way the circuits worked (by this time the replica was working), then Andy used that to correct M1SIM.

Q 5) According to your programmers reference manual, if a typewriter key is pressed when the machine is not running then the action line (which I deduce from m1sim to be the line selected by the switches on the panel) is updated accordingly. However if the machine is running, then "operation of the Typewriter will affect all the Action Lines encountered while the button is pressed". Does this mean that every line that is loaded into the PI while the button is pressed is adjusted.

A. Yes. An action line is one that is accessed, either to get the instruction or as an operand, either reading or writing. Every line that is accessed while the button is pressed will be affected, either a one or a zero written into that place, depending on the position of the write/erase switch. Exercise. Devise a program which makes use of this, eg a reaction time program.

Q 6) I notice that m1sim stops the machine running if a typewriter key is pressed, I take it this isn't correct behaviour?

A. Incorrect behaviour - one of the reasons we want to change M1SIM.

Q 7). I also notice that from the docs, the accumulator and CI and PI should be repeated all the way down the monitor, but again m1sim doesn't do this, is there any particular reason for that?

A. Misunderstanding by the author of M1SIM. Another reason we want to change it.

You said you have some of the competition entries. Did you get those from the web site? Are they the winners entries? The only ancient programs available are the ones analysed in our Annals paper, namely, Turing's Long Division, Tootill's HCF, and Kilburn's Highest Factor. Have you got those already?

Q 8). Sorry to really nit-pick but is the action line overwritten before or after it's accessed, i.e. is it corrupted immediately or does the first access work ok and corruption occur afterwards? I would presume it's overwritten before/during fetching and hence reads corrupt words.

Therefore, and here's the complication, the value fetched into the present instruction would be corrupted and could therefore affect the operand which is to be corrupted, can you confirm?

A. Correct.

The answer is found in Fig 9 of the IEE paper. A 1 is written in at the same time as the corresponding digit is read, so corruption occurs after reading. A 0 is written in at the same time as reading, so corruption occurs as you read. (Note that the output of the gate with Ha on it just below the typewriter amplifier is a pulse at the appropriate digit time, every Action time, whether writing a 1 or a 0.

Q 9). Would pressing and holding the KLC switch while running corrupt the present instruction also?

A. Yes.

Q 10). The other question on this would be, are manual instructions taken from the line and function switches also corrupted in the same way?

A. No. Manual instructions are not affected by the typewriter. See Fig 9.

Here is a scanned image [Figure 1] of the replica, during construction. The typewriter buttons should have their bit number painted on the end of the maroon-coloured button in black. The end of the button is slightly concave. The toggle switches are fairly clear. The post office switches are standard, with black handles apart from three with grey handles. Only KP spring returns.

Q 11). For an instruction where the action line is not used, such as STOP and TEST, would the action line get corrupted by a key being held down - i.e. is the action line fetched irrespective of whether the instruction needs it or not?

A. An Action line is any line being addressed in the store which is not a Scan line. There are two action lines in an ordinary instruction - one to fetch the instruction, and one to fetch or write the operand. So even STOP has an action line to get the STOP instruction into PI. It also addresses the line specified in its L-bits - there is nothing to stop it, though the data goes nowhere of course as the OTG will be closed (see Fig 11b [of the IEE paper]).

Typewriter inserting a one or a zero in the store is independent of what instruction function is being currently executed. Every instruction causes two action lines to be addressed, one from the CI, one from the L-bits of the instruction (i.e of the PI). If the typewriter button is pressed during any action line, then that line is zapped in that bit position.

Q 12). What do the M and B buttons next to the A,C,Sc selectors do?

A. Could be considered a point of non-authenticity! The panels are based on photos taken 15th December 1948, by which time the machine had been fitted with Multiplier and B-tube. Strictly speaking they should be unused at 21st June 1948.

Q 13). What type of buttons are the M and B buttons?

A. The five buttons are a unit where pushing in one button locks it, and releases any button already in. (This mechanism is removed in the typewriter of course). So any one of the three CRTs can be selected for display on the monitor. You may choose to leave five buttons in the GUI but only have 3 do anything. (In fact, pressing those will leave the store displayed). The switches used originally had two types of plastic button, either a top hat shape, or else a plain cylinder. I could only obtain the cylinder type, so had the top hat shape made specially. When I counted up how many to have made, I forgot to include the two spare button positions. You can choose whether to represent the original SSEM or the replica! By the way, when you get the CD-ROM, there is a 1949 BBC newsreel film which shows operation of the typewriter quite clearly, though the resolution is not as good as the real film.

Q 14). What does the subscript c on the store selector, 'Sc', stand for?

A. Actually it is S subscript 0, for store tube zero. A bit of masking tape has covered half of it. Again, by December 1948 they were just putting in extra storage tubes, and the control panel below was prepared for another bank of five switches.

Q 15). Why is the MAN AUTO labels in a different place to the STAT switch on the modern picture?

A. The replica panels were made before I had obtained that black and white picture, and was the best estimate I could make of what it was like derived from other photos. I probably ought to repaint the control panel. The Stat switch should be labelled MAN/AUTO as described in the PRM. I think we have just done it in ball point pen on the replica!

Q 16). Is there any chance of an image of the location of the stop lamp relative to the monitor?

A. No original photo is convincing. I placed it hanging down behind the chassis just above the monitor panel at the left. Both Geoff Tootill and Tom Kilburn have agreed that it is a plausible place though they can't guarantee it. It is just a little 1/4" dia neon bulb hanging down. Nothing like M1SIM. It is a little to the right of the four holes in a square in the panel I am pointing at in the picture. [Figure 2] ... I am not pointing at the neon lamp, but at the chassis behind which the neon lamp hangs. The neon lamp is much further to the left near the upright of the rack.

Q 17). Can you tell me what the other switches are next to the line and function switches? In particular the furthest left, labelled 'c'?

A. Again, when the address range and function range had increased by December 1948, more switches were used. There are quite likely to have always been 7 line switches from the experiments in 1947 when they were trying as many lines as they could. These are the ones labelled 0, 1 etc above the switch. On the replica, only 0 to 4 are wired in. "C" was added later in 1948 when they increased the word length to 40 bits and put two instructions in one word. It defines which half of a word. We don't use it.

We only use the left hand three function switches labelled 13, 14, 15.

The ball dolly toggle switches are obvious. The Fswitches are actually a RAF wartime type with a somewhat cylindrical bakelite dolly, black, not nickel plated like the others. (Same as the Write/Erase switch and the prepulse switch CS)

Q 18). Why don't the light grey KLC, KSC, KAC in the modern photo appear to be the same ones that are light grey in the old photo?

An error. We ought to put them right. Thanks for pointing it out. In fact, looking again at the old photo they may have been white, but grubby with use. You should try to be authentic. Starting with KC, I suggest:

KC KLC KSC KAC KBC KCC KEC KMC
black, - grey, red, grey, grey, grey, red, red - .

In the replica we only use KC, KLC, KSC, KAC, KCC, which seems to be what they had in June 1948. KEC means clear everything and was added later in 1948. KMC means clear multiplier tube; we have wired it to trigger a PC to upload a program direct into the store of the replica. Incidentally, the black panel with the switches is a standard GPO telephone exchange 10-position switch panel.

Q 19). I take it the KAC was unattached as I've not seen it mentioned elsewhere?

A. No, KAC was Key Accumulator Clear and was in use June '48.

Q 20). You say that 'Only KP spring returns', I can't see a KP switch, do you mean KSP i.e. KC?

A. I meant KC. They originally called the pulse which starts an instruction the "Completion Pulse" and by June 1948 were just changing over to call it the "Prepulse". The key meant "Key Completion" in typical GPO telephone exchange nomenclature. Geoff Tootill was fond of telephone exchange circuits and no doubt that is why he used that nomenclature. The IEE paper was based on Geoff Tootill's MSc thesis, and was published later. So there is a mixture of names for the same key. KSP was not used in June 1948.

Q 21). I'm trying to work out whether write or erase should be at the top on the switch since sources conflict:

modern photo = write at top

old photo = write at bottom (if my eyes don't deceive me)

m1sim = write at bottom

Can you confirm which was correct in the June 48 baby?

A. Old photo and M1SIM.

Q 22). Were the M & B buttons were actually present in June 1948 or not?

A. The buttons come in banks of five - they were present but unused.

Q 23). When the Baby is in manual mode and is running, does the instruction encoded on the Lstat and Fstat switches appear as the present instruction on when the control is displayed?

A. The C tube only has a PI on it during the execution of an instruction - on Single Shot there is a single fleeting view of PI.

I would have to check on the machine, but I am pretty confident that on execution of a manual instruction, the PI displayed on C will be the one which comes out of the store addressed by the CI. The instruction executed is of course the one on the switches. See Fig 9.

I think the CI does not change, i.e. you can do a manual instruction without losing your place in a program, on Single Shot, but it counts up on Run. Can't remember the logic of that just now.

[Having tried the simulator] I keyed in and ran a program without trouble - it is quite realistic. The speed looks OK though I would need to run some of our programs to be sure.

On the real machine, the action line is slightly brighter than the other 31 lines. This is because the line is being accessed more often. It is a useful feature at stop, because you can see which line is currently being selected. For example, CI may be zero, and the top line is bright. You set the Stat switch to Manual, and the

bottom line is bright because the L switches are down. Setting the L switches to a line to insert bits is easy - you can see where you are going to overwrite before keying.

Q 24). Can I just confirm that if the write/erase switch is set to erase when running then all action lines get blanked? I think I got that from your PRM A3.3.3 "Should it be left in the "Erase" position, then all action lines would be corrupted." ?

A. I can't confirm or deny that, because I can't remember the mechanism at the moment. According to Fig 9 it should have no effect unless you press a typewriter button. But I must have written the sentence in the PRM for some reason. Can we leave it an open question pro tem?

The C tube only has a PI on it during the execution of an instruction - on Single Shot there is a single fleeting view of PI.

Q 25). I've taken a look at how I could do this but it would be very awkward and probably wouldn't work too well (i.e. would be displayed for too long to be realistic) so won't implement it.

A. That's OK. In fact we sometimes frig the real machine so PI persists to make life easier when debugging faults. Perfectly acceptable to mention that you know how it should work but that it is an implementation detail to fix it!

Q 26). Do you mean that I should keep incrementing the CI every time a manual instruction is executed at run?

A. I think so. I will have to check the logic. May be several days as I probably will have to consult a colleague. Tell your supervisor that it is an arcane point which is still under investigation!