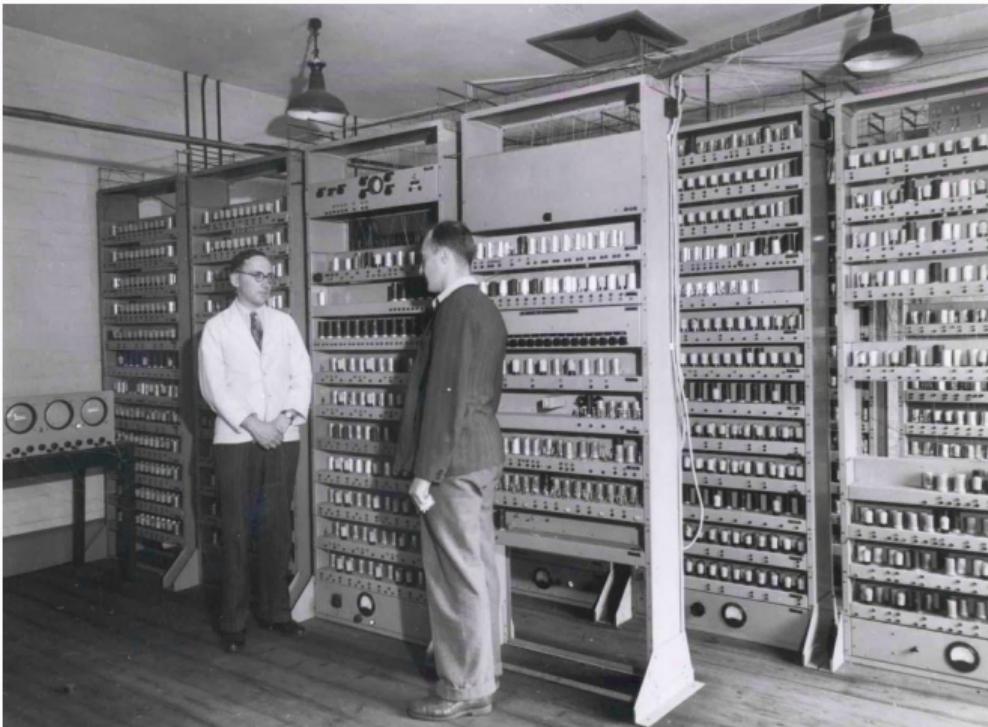


Mercury Delay Line Emulation: Reconstructing Memory for EDSAC

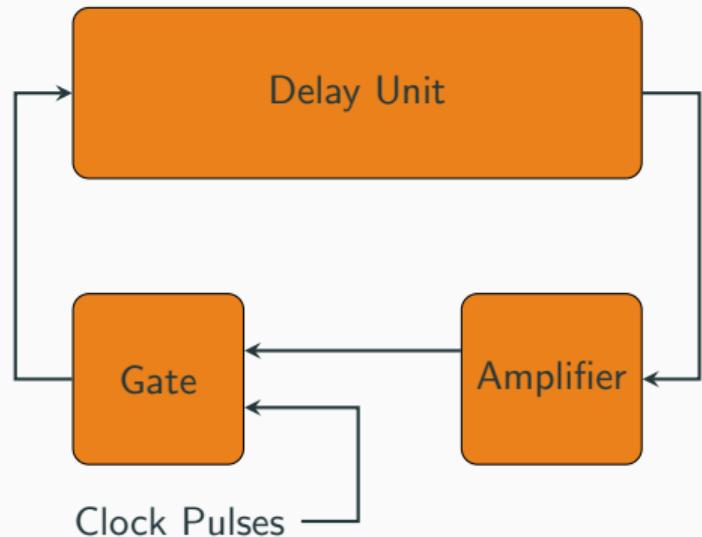
Joshua Tyler

August 30 2017

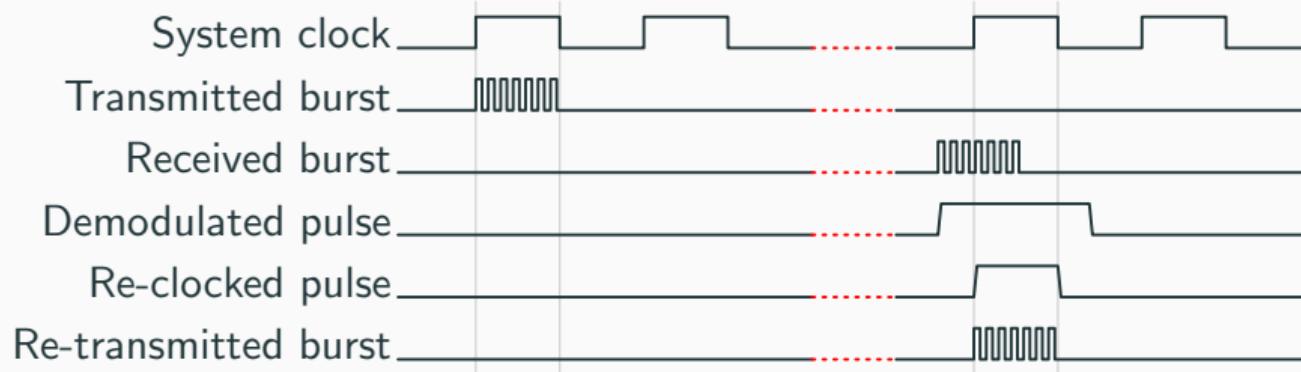
What is EDSAC?



How did memory work in EDSAC?



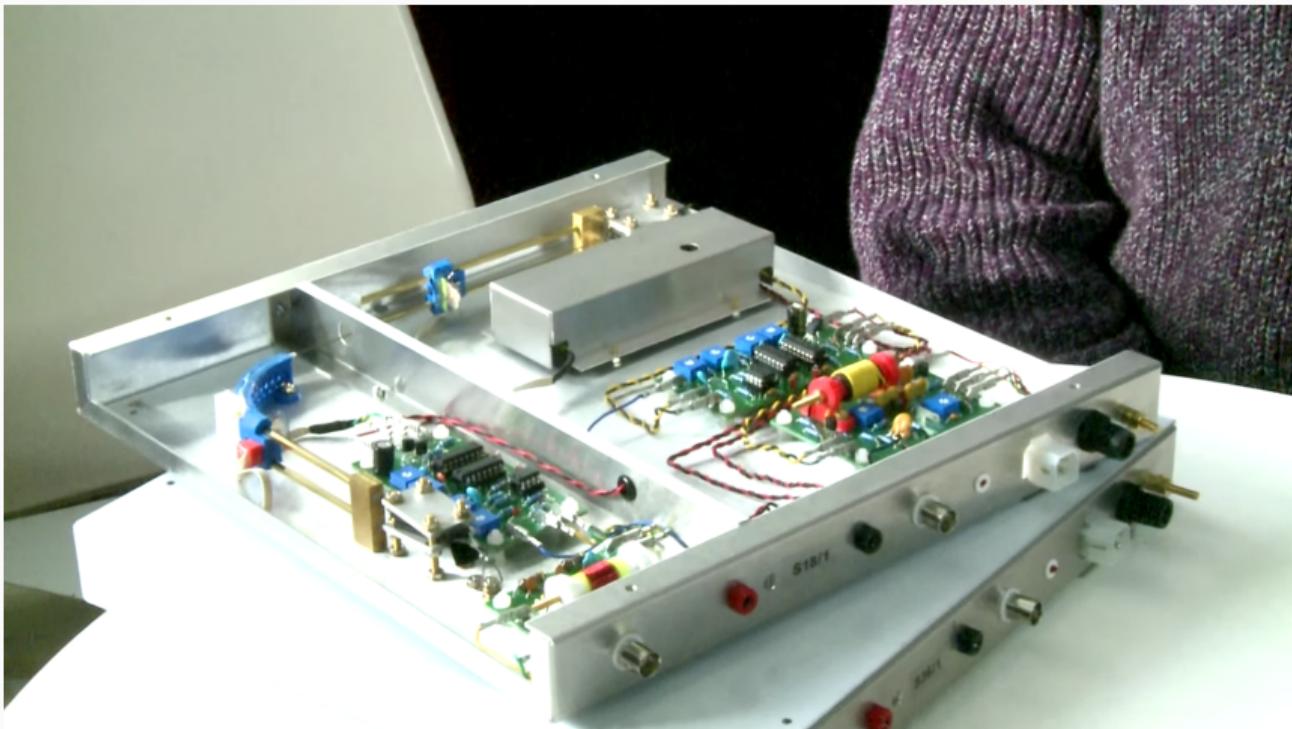
How did memory work in EDSAC?



Reconstruction Effort



Reconstruction Effort



Where I come in

- If you're going to cheat – cheat big!
- Using modern technology to create delay lines for EDSAC as close to the original as possible in terms of:
 - Form
 - Function
 - Electrical interface
 - N.B. This includes being 'phantom powered'
- Creation of a test harness which can emulate the memory interface of EDSAC

An Ultrasonic Memory Unit for the EDSAC

By M. V. WILKES, M.A.*
and
W. RENWICK, B.Sc.†

Introduction

THE successful completion of the ENIAC by the Moore School of Electrical Engineering in Philadelphia marked a new stage in the application of electronic devices to computing.^{1,2} This machine suffers, however, from two major drawbacks: its great complexity—it contains in all 18,000 valves—and its very limited memory capacity. The latter defect, although unimportant when solving the ordinary differential equations for which the machine was primarily intended, seriously limits its sphere of application. In order to deal with problems in partial differential equations, for example, one really needs storage capacity for hundreds or even thousands of numbers.

Fig. 5. General view of the computer racks with a completed battery of delay units in the foreground

design presents many problems of interest to electronic engineers; it is thought that a description of its main features may be of interest in advance of the completion of the whole machine.

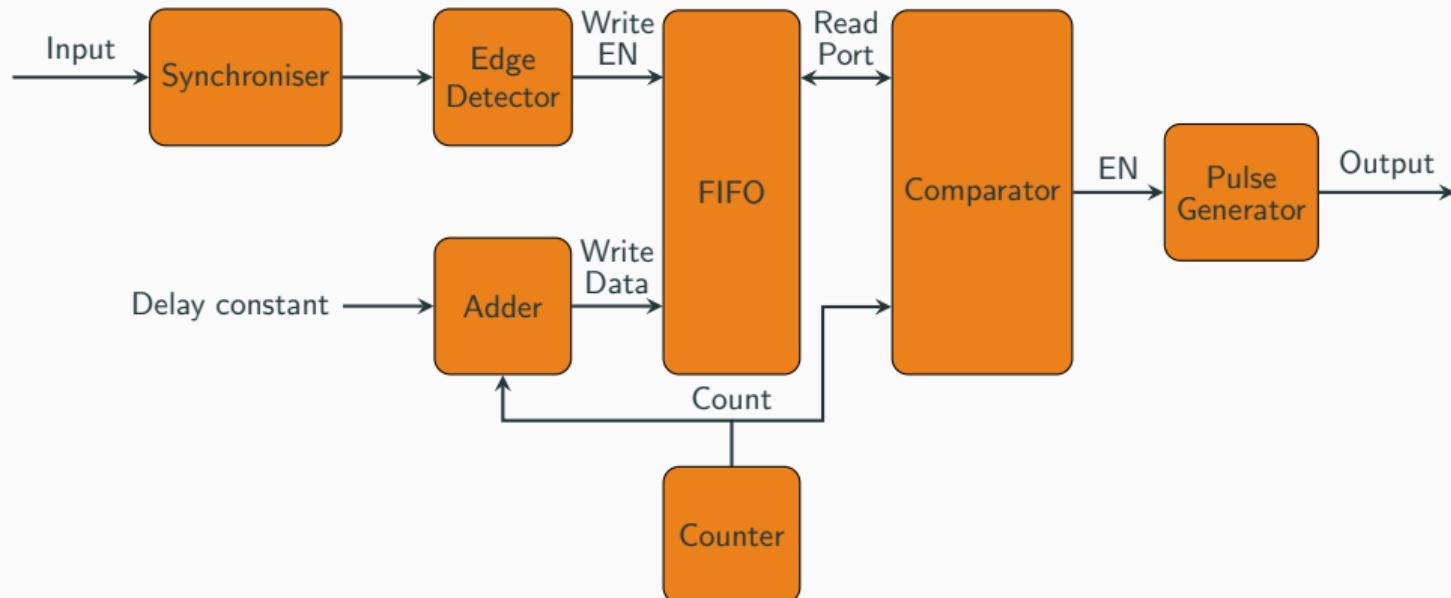


train of pulses, the presence of a pulse indicating a "1" and the absence of a pulse a "0." Fig. 1 will make this clear. This system is adopted for the EDSAC and all internal operations in the machine

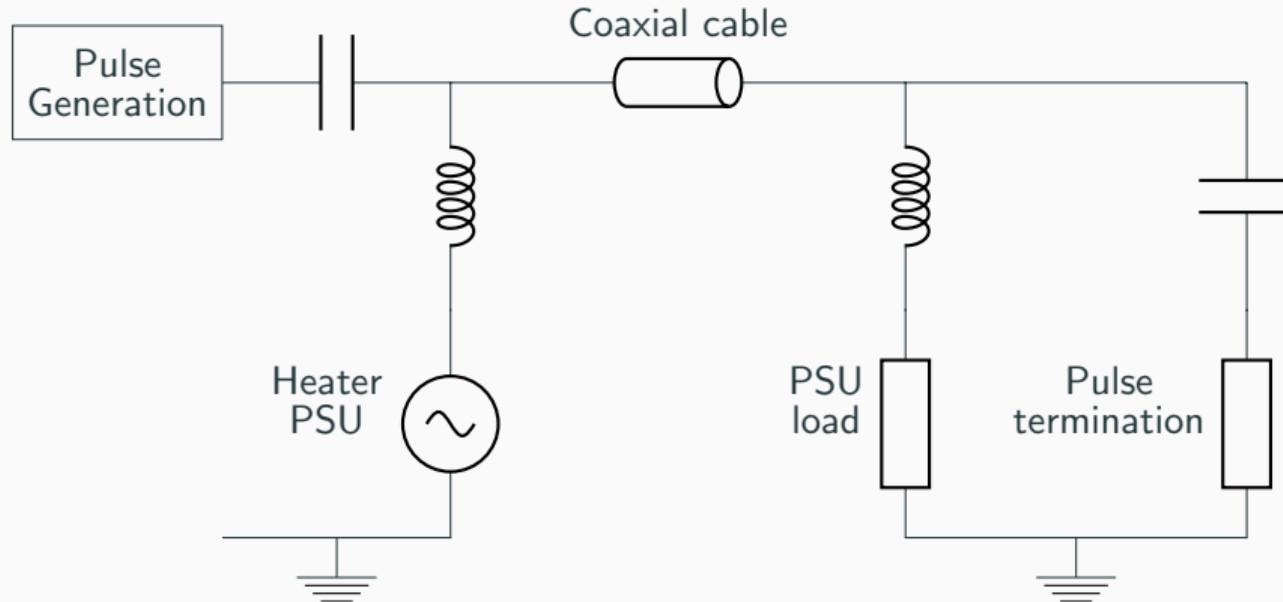
Specification

Item	Specification	Justification
1	Must be capable of producing a delayed copy of the EDSAC pulse train presented to its input	This is the primary function of the device.
2	Must be powered from the input signal driven by EDSAC, with only minimal non-intrusive modifications made to EDSAC.	The goal of the device is to faithfully recreate the appearance and electrical interface of EDSAC, and thus large modifications such as power supply wires must be avoided.
3	Must be able to have a nominal delay of 1.15 ms, adjustable by at least $\pm 10\%$.	1.15 ms is the nominal delay of a long tube, as discussed in the report. An adjustable delay allows synchronisation with the system clock, with may vary .
4	Must have a maximum per burst deviation from the nominal delay of 50 ns.	This ensures that the delay line output will be able to synchronise with the clock of EDSAC. 50 ns is the maximum deviation derived in the report
5	Must be able to interface with input waveforms AC coupled bursts of 13.5 MHz carrier, with peak voltages in the range of 25 V to 35 V.	This is necessary to mimic the performance of the original delay line, 25 V to 35 V is the range derived in the report.
6	Must be able to have an adjustable nominal output voltage in the range of 10 mV to 100 mV (peak to peak), driving into 70Ω .	An adjustable output voltage in this range allows compatibility with both the original electrical interface, and that used by the reconstruction effort, 10 mV to 100 mV is the range derived in the report.
7	Must be encapsulated in a metal tube of 4.44 cm outer diameter, and 165.5 cm length.	This diameter allows the design to have the same appearance of the main memory store tubes of the original EDSAC design, the width and diameter are discussed in the report
8	Must be accompanied by a testing device capable of emulating the signals produced by EDSAC.	This allows the delay line to be tested separately to the reconstruction project.

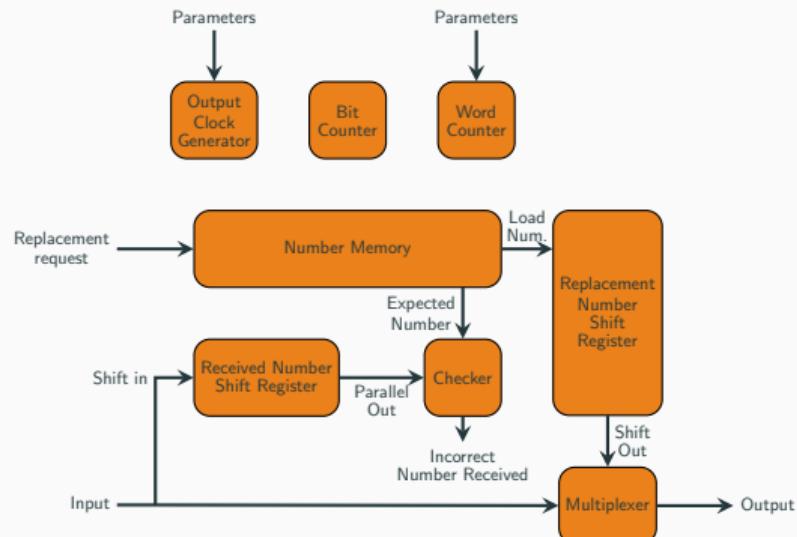
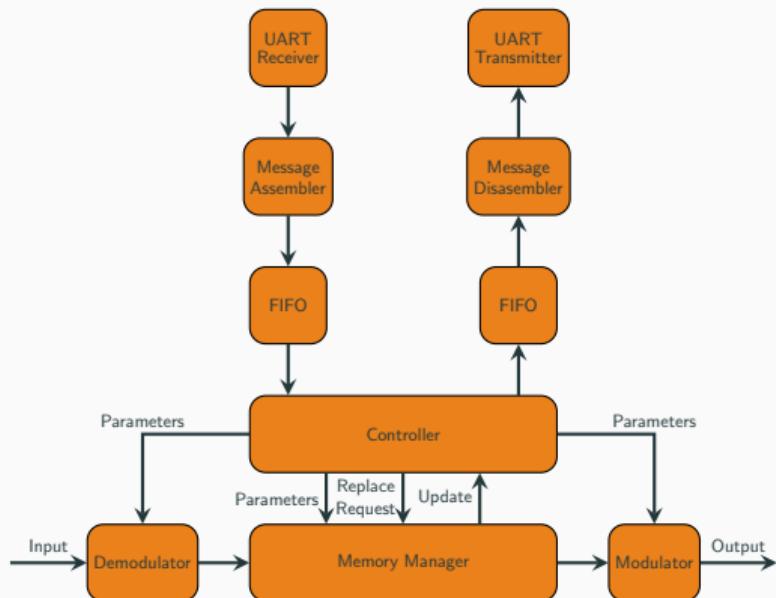
What I've done – FPGA delay line



What I've done – Phantom power



What I've done – Digital test harness



What I've done – Test harness software

mem_gui

System options

Number of numbers: 16 FPGA Clock Frequency (MHz): 54.0

Port options

Serial port: 0 Lattice Lattice FT232R Interface Cable Baud rate: 9600 Connect

Modulator options

Half period time (in clock cycles): 2 Send

The signal will be modulated with 13.500000MHz

Demodulator options

Pulse width (in clock cycles): B1 Send

Memory manager options

Pulse width (in clock cycles): 4B Pulse gap (in clock cycles): 59 Send

The bit time is 1.981481us, giving a circulating bit rate of 504.672897kHz.
The delay time for all bits is therefore 1.14133ms.

Test mode

Run Reset parameters to current values

Memory updaters

Address: 0 Value (hex): 0 Send

Fill with 0s Fill with 1s Fill with random data

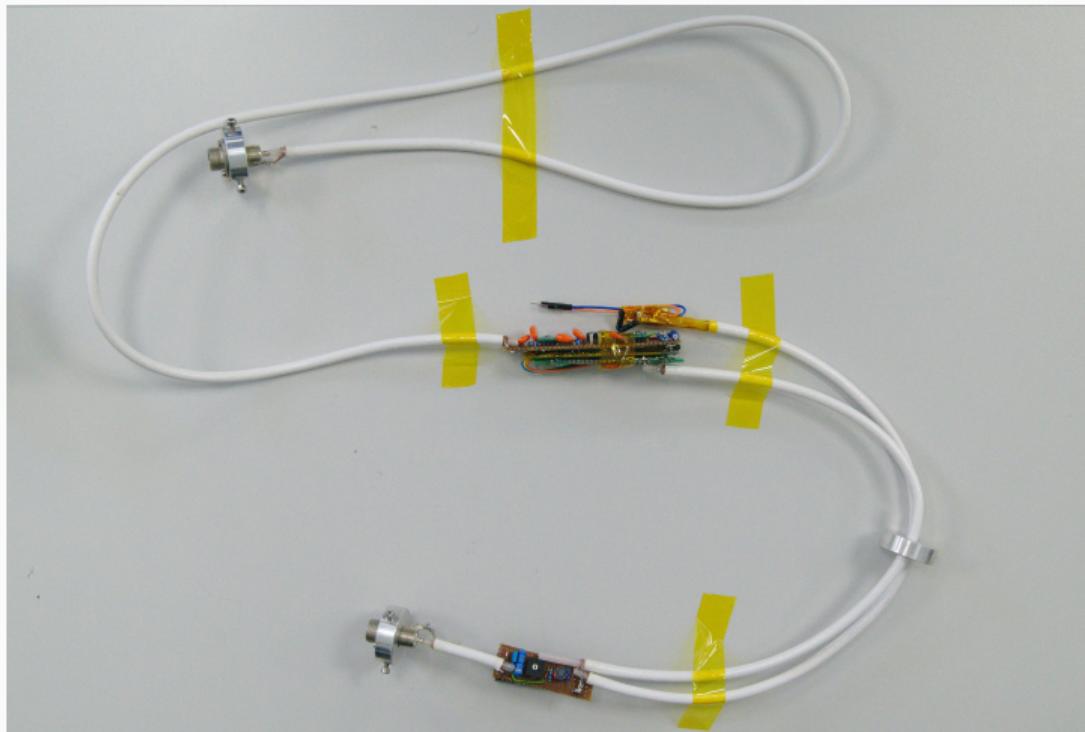
Memory viewer

Address	Data (Hex)
0	717374d2
1	3a72d1795
2	14Be871be
3	7a2042825

Log

Sent:5
Received: Ack; sys status: run: 0
Sent:203
Received: Ack; mod params: cycles per half period: 2
Sent:5104
Received: Ack; demod params: pulse width: 81
Sent:76601006
Received: Ack; mem params: no nums: 16 test mode: 0 pulse width: 48 pulse gap: 59
Sent:105
Received: Ack; sys status: run: 1
Sent:717374d2e0001
Received: Ack; replace num: addr: 0 data: 30454271277
Received: replace num done: addr: 0 data: 0
Sent:3a72d17950101
Received: Ack; replace num: addr: 1 data: 15689652117
Received: replace num done: addr: 1 data: 0
Sent:14Be871be0201
Received: Ack; replace num: addr: 2 data: 5518160318
Received: replace num done: addr: 2 data: 0
Sent:7a20428250301
Received: Ack; replace num: addr: 3 data: 32782952485
Received: replace num done: addr: 3 data: 0
Sent:729690e880401
Received: Ack; replace num: addr: 4 data: 30764764808
Received: replace num done: addr: 4 data: 0
Sent:260e5d72c0501
Received: Ack; replace num: addr: 5 data: 10215610156
Received: replace num done: addr: 5 data: 0
Sent:7b991f6e0601
Received: Ack; replace num: addr: 6 data: 33178121966
Received: replace num done: addr: 6 data: 0
Sent:6ab77b560701
Received: Ack; replace num: addr: 7 data: 28646538070
Received: replace num done: addr: 7 data: 0
Sent:46d4b954b0801
Received: Ack; replace num: addr: 8 data: 20751029579
Received: replace num done: addr: 8 data: 0
Sent:45b7696090901
Received: Ack; replace num: addr: 9 data: 18714367497
Received: replace num done: addr: 9 data: 0
Sent:75f59e8c0a01
Received: Ack; replace num: addr: 10 data: 31664086668
Received: replace num done: addr: 10 data: 0
Sent:2f04aac00001
Received: Ack; replace num: addr: 11 data: 12834876108
Received: replace num done: addr: 11 data: 0

What I've done – System integration



What I've done – Problems

- Quite a few!
- Phantom powering not as simple as first thought
- Crosstalk between input and output stages
- High speed, high voltage, high power amplifier design

Conclusion

- Success!
- What was achieved:
 - Analysis of EDSAC's 'memory problem'
 - A working recreation of EDSAC's delay lines that meets the intended goal
 - SPICE simulation of the design integrating with EDSAC
 - Pending integration testing next week
 - A working test harness capable of generating the correct output signals, and analysis of the input signals

Demo

What I've done – Analogue interface

