

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.
- 10.
- 11.
- 12.
- 13.
- 14.
- 15.
- 16.
- 17.
- 18.
- 19.
- 20.
- 21.
- 22.
- 23.
- 24.
- 25.

26.27.28.29.30.31.32.33.34.35.36.37.38.39.

Building and Programming Australia's First Hobby Computer, the EDUC-8

Steven S. Pietrobon

Small World Communications
6 First Avenue, Payneham South SA 5070, Australia
steven@sworld.com.au

Presented at the

Institute for Telecommunications Research
University of South Australia
Mawson Lakes, South Australia

25 February 2009



Small World Communications

Introduction

- The Educational Digital Micro (μ) Computer (EDUC-8, pronounced “educate”) is Australia’s first hobby computer. It was designed by Jamieson Rowe of Electronics Australia and presented as a series of articles from August 1974 to August 1975.
- The machine was designed to teach the basic principles of programming a computer, as well as learning how a digital computer work.
- Mr. Rowe decided to use a discrete TTL implementation as microprocessor chips were very expensive, about \$110 in 1974 or \$730 today.
- The initial design had only 32 bytes of memory, as memory was also very expensive. However, with the falling cost of memory this was increased to 256 bytes by the time of the first construction article in October 1974.



Introduction (continued)

- In the first article, it was thought that this was the world's first hobby computer. However, Electronics Australia was beaten by the US magazine Radio Electronics with its MARK-8 computer using the Intel 8008 microprocessor, published in July 1974.
- As Mr. Rowe had prior experience with the popular DEC PDP-8 minicomputer, he decided to use an 8-bit version of the 12-bit instruction set of the PDP-8. The LINC-8 computer in the ITR lobby is in fact a PDP-8 with another PDP-8 coprocessor.
- To also reduce cost, a bit-serial implementation is used. That is, all data transfers to and from memory are performed serially. A basic instruction used either 48 or 72 clock cycles at 500 kHz, giving an instruction rate of about 10 kHz.
- A total of 100 74 series and 9000 series TTL chips were used. Two 1Kx1 bipolar TTL RAM chips were used for the memory.



Why?

- In 1977 I was 13 years old and in Year 8 at Marden High School. I was interested in electronics and had come across the EDUC-8 articles in the school library. I thought it would be fun to build my own computer, even though I did not know what I could use it for.
- You could buy the EDUC-8 as a kit for \$300, but that was very expensive back then (about \$1360 today). I asked my Dad for the money, but as I could not give a good reason and we did not have much money, I was not able to buy the kit.
- While looking through some old papers in 2006 I found an old Electronics Australia magazine from December 1977. This described an electronic poker machine project that I had started in high school but not finished due to lack of parts.
- I decided that I would finish this project and dug out the old printed circuit board (PCB).



Why? (continued)

- Using the local Dick Smith electronics store and online stores, I was able to buy all the parts and successfully completed the project.
- Remembering back to the 1970's, that old desire for an EDUC-8 returned. I now had the money, but the kit was long gone.
- The first step was easy. During the 1980's I obtained some old Electronics Australia magazines with all the EDUC-8 articles. These were torn out and stored in an envelope. I refound this envelope and so I had the instructions.
- The next steps would not be so easy...



Instruction Set

- Just like the PDP-8, the EDUC-8 uses the three most significant bits of an instruction as an operation code. Thus there are eight basic operation codes. The first six are memory reference instructions, one for operate instructions and one for input/output transfer (IOT) instructions.
- Also just like the PDP-8, the EDUC-8 has four registers, but 8-bits in length instead of 12.

AC = Accumulator Register

PC = Program Counter Register

MA = Memory Address Register

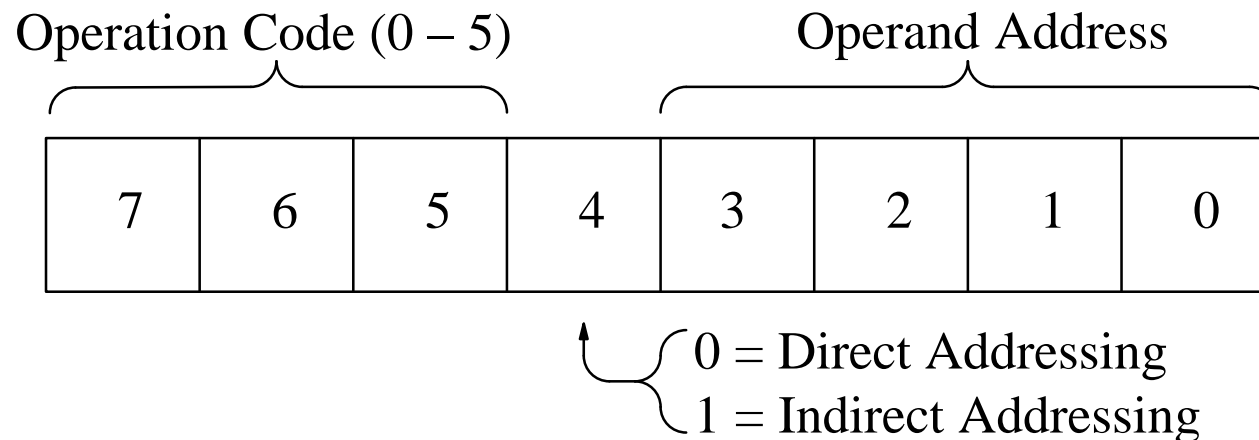
MB = Memory Buffer Register

- AC is used to store and perform all calculations in a program. PC is used to indicate the address of the current instruction. MA is used to address the memory. MB is used to store the output of the memory.



Memory Reference Instructions

- Similar to the PDP-8, the EDUC-8 has two addressing modes, direct and indirect. The memory is organised into 16 pages, each with 16 words, for 256 words in total.
- In direct addressing, the address A of the operand $M(A)$ is formed by PC7 to PC4 of the instruction address and by MB3 to MB0 of the instruction. This limits addressing to within only 16 words of the current page.
- In indirect addressing, the address of the operand is taken from $M(A)$. This allows an instruction to address any location in memory.



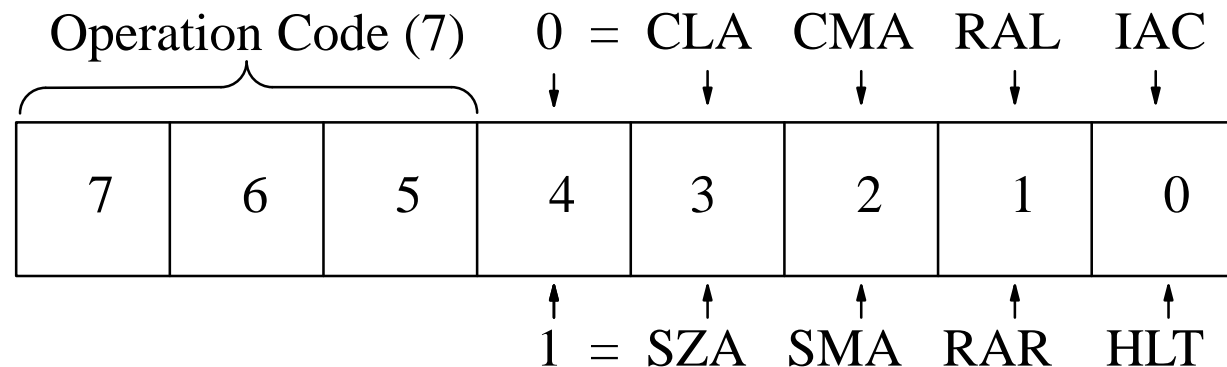
Memory Reference Instructions (continued)

- The six instructions with their operation code in brackets are:
- DCA (3) – Deposit AC at operand address and clear AC.
- TAD (1) – Two's addition operand with AC and store in AC. With a clear accumulator (CLA) or DCA instruction beforehand, this is also used to retrieve data from memory.
- AND (0) – Logical AND operand with AC and store in AC.
- ISZ (2) – Increment operand and store at operand address. If incremented operand is zero, skip next instruction. This is useful for loop programming.
- JMP (5) – Jump to operand address.
- JMS (4) – Jump to subroutine. Store address of next instruction at operand address. Jump to next instruction after operand address. Using a jump indirect (JMP I) instruction to the operand address allows a return from the subroutine.



Operate Instructions

- As only five bits are available compared with nine bits of the PDP-8, only a small subset are implemented.



NOP (700) No Operation

IAC (701) Increment AC

RAL (702) Rotate AC Left

CMA (704) Complement AC

CLA (710) Clear AC

CLA.IAC (711) AC = 1

CLA.CMA (714) AC = -1

NOP (720) No Operation

HLT (721) Halt

RAR (722) Rotate AC Right

SMA (724) Skip on Minus AC

SZA (730) Skip on Zero AC

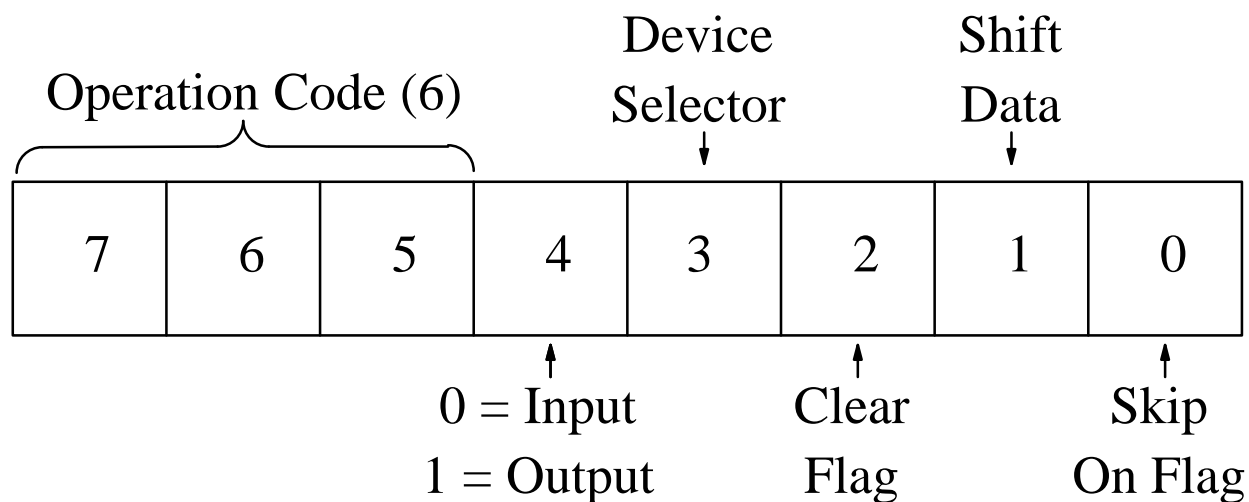
CMA.IAC (705) Two's complement AC

SZA.SMA (734) Skip on zero or minus AC



IOT Instructions

- Similar to the PDP-8, one bit is used to indicate an input or an output port and three bits to command the port.



SKF (601,611) Skip on input flag	SDF (621,631) Skip on output flag
KRS (602,612) Read input buffer	LDS (622,632) Load output buffer
RKF (604,614) Reset input flag	RDF (624,634) Reset output flag
KRB (606,616) Read input buffer, reset flag	
LDB (626,636) Load output buffer, reset flag	



Printed Circuit Boards

- There are eight single sided printed circuit boards containing 100 integrated circuits. The first two are vertical. The remaining six are horizontal and plug into the mother board.
- E8/F Front Panel Board
- E8/C Connector (Mother) Board
- E8/T Timing Board
- E8/D Decoder Board
- E8/M Memory Board
- E8/P Program Counter and Adder Board
- E8/A Accumulator Board
- E8/IOT Input Output Transfer Board



Obtaining Components

- Most of the integrated circuits (IC) were obtained from Unicorn Electronics in the US. Equivalent Intel 2115 1Kx1 SRAMs were used instead of the original Fairchild 93415 devices. Other sources included my own personal stock and stock from friends.
- The only IC that was difficult to obtain was a Fairchild 9001 JK flip flop, used to store the carry bit in the adder. This was sourced from American Microsemiconductor in the US. It was later found out that the 74105 could have been used.
- Other components such as the transformer, heat sinks, resistors, capacitors, transistors, screws, nuts, washers, and connectors were obtained from Unicorn, Dick Smith, Jaycar, Farnell and RS.
- Original parts that could not be obtained were the front panel switches, mother board connectors, case and printed circuits boards.



Front Panel Switches

- Single pole double throw (SPDT) switches, both standard (10) and momentary (5), from C&K were used. The leads were PCB mount with the switch itself panel mount.
- Obtained 10 SPDT switches with blue levers from Electronic Parts for Less and 5 SPDT momentary switches with black rockers from eBay. They were PCB instead of panel mount.
- Obtained red and black levers from Switches Plus to replace rockers and two of the blue levers. The PCB mounts were modified to be panel mount.

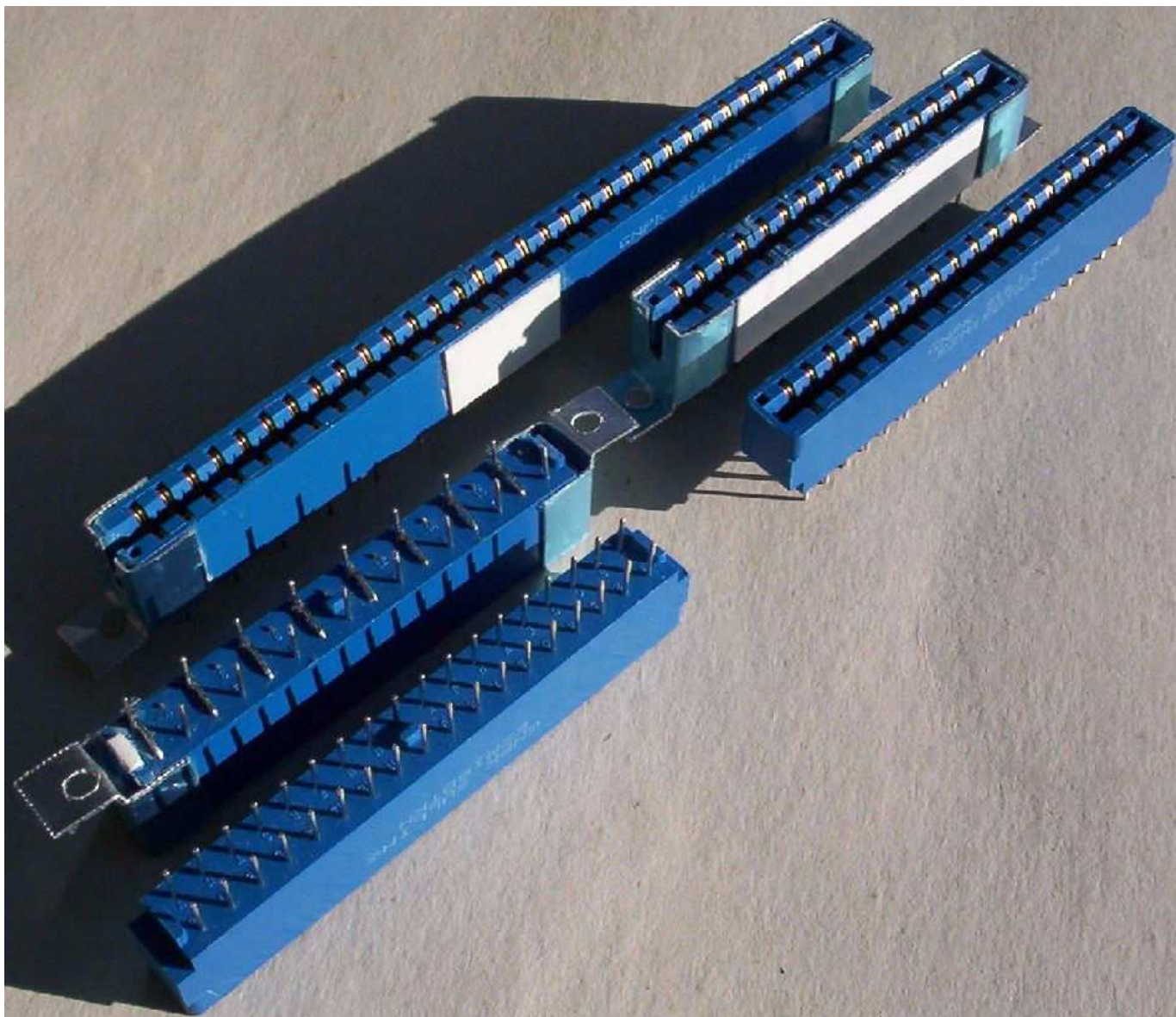


Connectors

- The original design used four 32 way and seven 16 way PCB mount single sided edge connectors with side mounts from McMurdo. Pin spacing was 3.81 mm (0.15 inch) and had an unusual up down pattern.
- Obtained a box of 18 way 3.96 mm (0.156 inch) double sided edge connectors without side mounts from eBay.
- These were cut up and reassembled using super glue and plastic card into 16 and 32 way connectors. Side mounts were made from left over material from the case. Pins were cut, bent and soldered to make up down pattern.
- Originally intended to cut connectors into three or four pin segments to obtain 3.81 mm pin spacing. However, as we could not obtain the original boards, we decided to make our own boards and change the pin spacing to 3.96 mm.



Connectors (continued)



Case

- The original case measured 293 mm wide, 103 mm high and 357 mm deep. It was made from 0.91 mm (20 gauge) steel with chocolate hammertone lacquer.
- Obtained two sheets of 0.5 x 300 x 900 mm aluminium sheet from Mitre 10. This had clear blue plastic covering on outside and white plastic covering on inside.
- After drawing dimensions, the sheets were cut and bent into five parts using the workshop in the Sir Charles Todd building. This included two side walls, bottom, top and rear lip.
- A bracket for mounting the front panel switches was also made. Rectangular holes were made by drilling, cutting and filing. Smaller holes were made using a drill.
- Unlike original case, the 16 pin paper tape connector was separated into a standard 6 pin output connector and a 15 pin D-type connector.



Case (continued)

Case parts after bending.

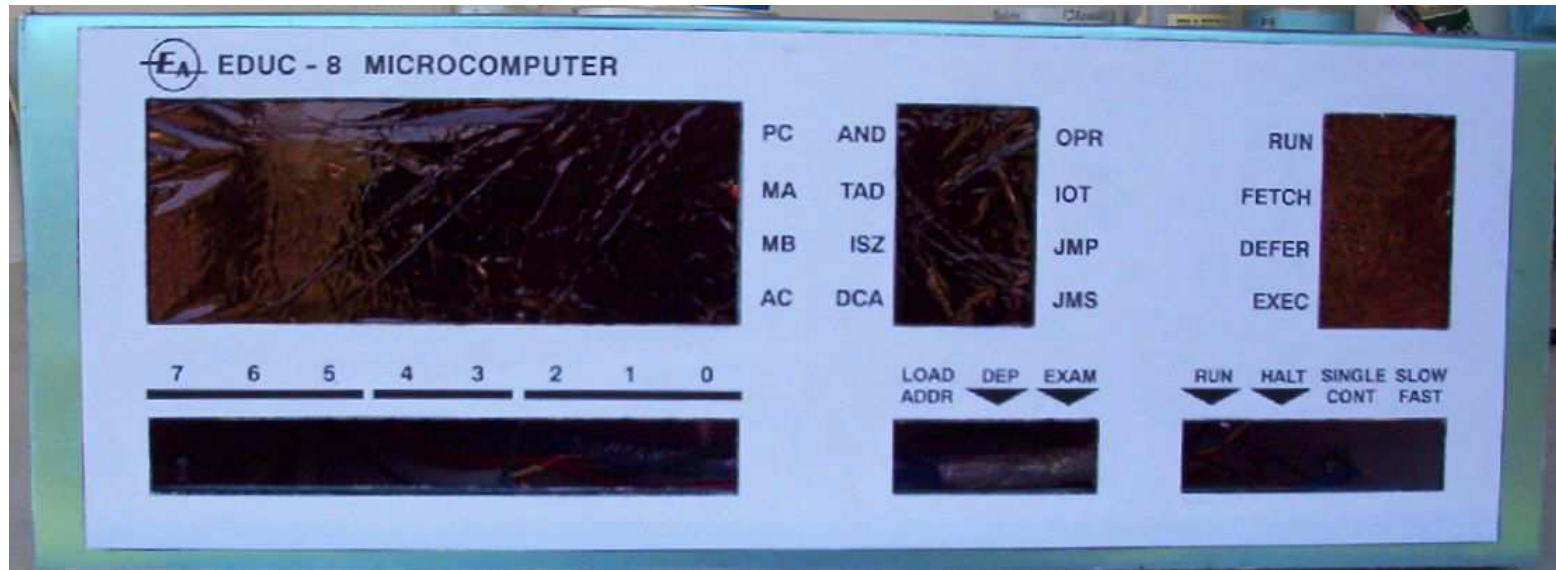


Switch bracket.



Front Panel Artwork

- A photo of the front panel of an original EDUC-8 was imported into Interleaf. This was sized to the appropriate dimensions.
- The lettering and window holes were recreated and printed using a laser printer onto white paper. This was glued onto the lid of the case.
- The rectangular holes for the windows were then cut out. Red cellophane from some old Christmas wrapping was placed behind the windows.

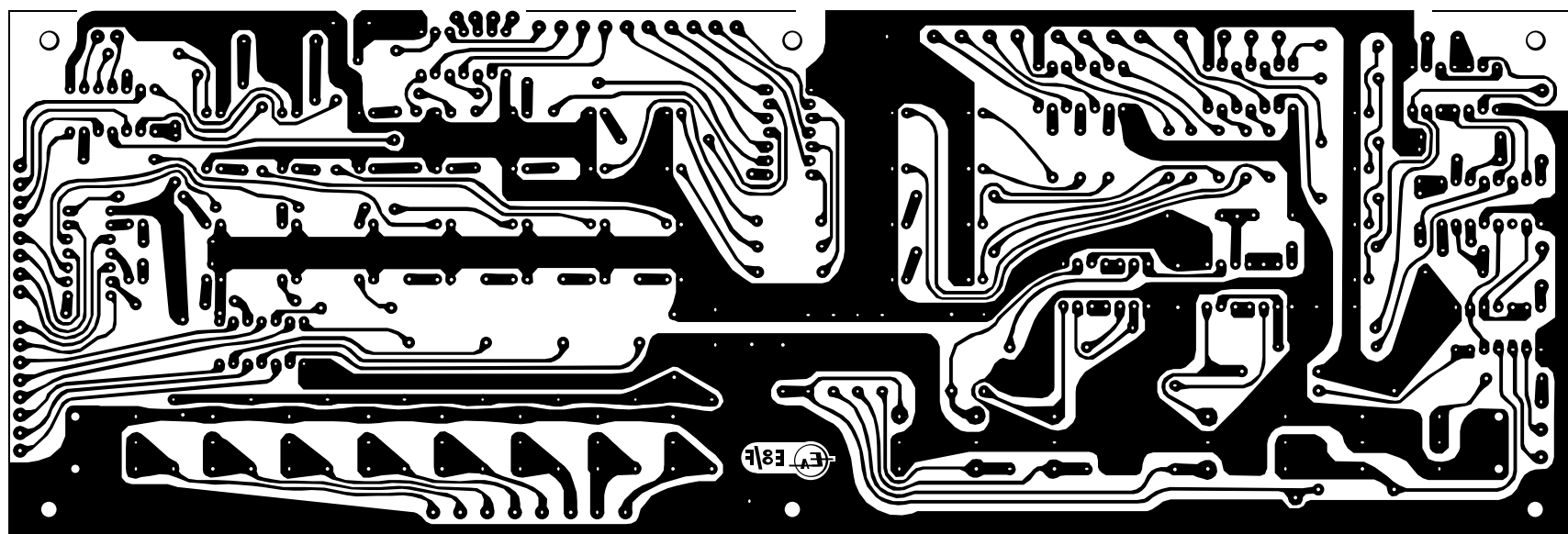
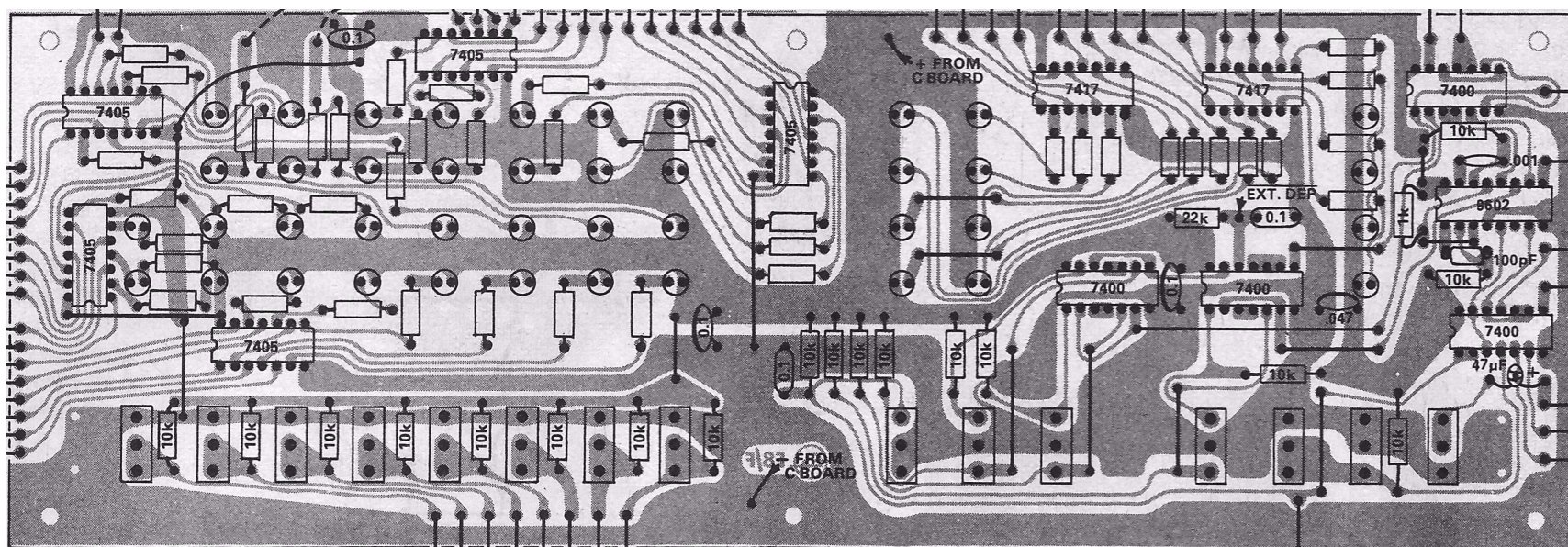


Printed Circuit Board Artwork

- After unsuccessfully trying for three months to obtain boards from original manufacturer, I decided to make my own boards.
- The artwork was no longer available, so I had to recreate the artwork from the articles. This was done by scanning the component layout diagrams which had the artwork in grey under the components.
- The JPEG scans were imported into Interleaf where the drawing package was used to recreate by hand every track, a long and tedious process for the eight boards.
- Four additional boards were also made.
- E8/X Extender Board
- E8/KI Keypad Input Board
- E8/S Octal Display Board
- E8/SR-T Serial Receive and Transmit Board

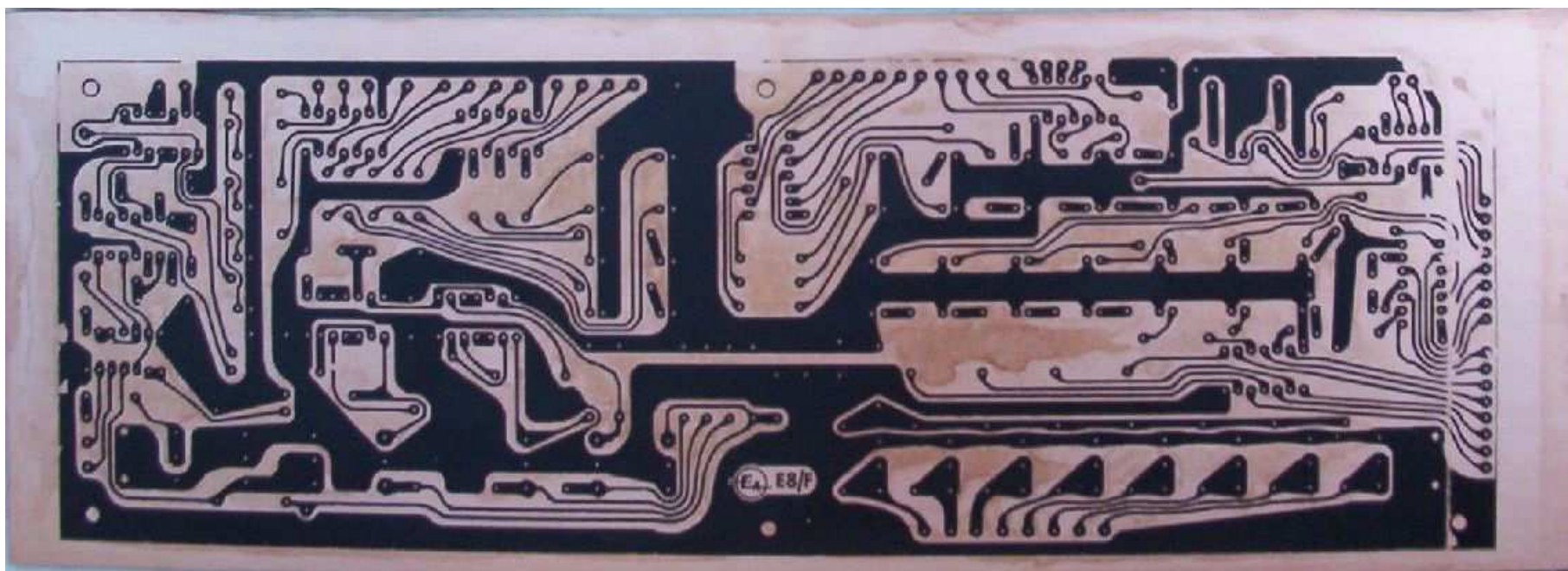


Printed Circuit Board Artwork (continued)



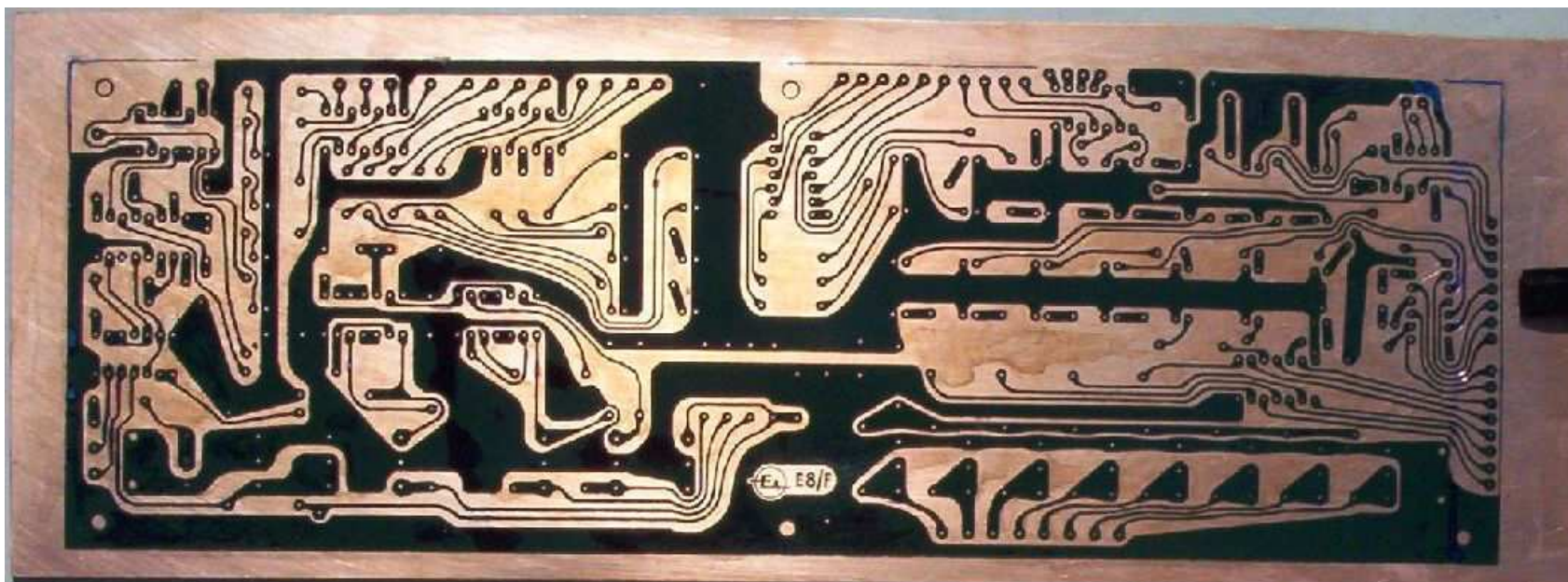
Toner Transfer System

- The Toner Transfer System was used to mask the PCB. This involves using a laser printer to print the artwork onto special Toner Transfer Paper using a reverse image of the artwork.
- The blank PCB is then cut to an appropriate size. I initially used a hacksaw, but later used a large paper guillotine. The toner is then transferred to the cleaned PCB using heat and pressure. I initially used an iron, but later used a laminator.



Toner Transfer System (continued)

- The toner paper is removed by soaking the board in water which then peels away.
- To seal the toner and protect it from the etching solution, green TRF is used. The green TRF is cut into the same size as the board and applied with heat and pressure in the same way as the toner. The green TRF is peeled away, leaving a seal on the toner.
- Any areas of toner that were not successfully transferred are filled in using a printed circuit marker.

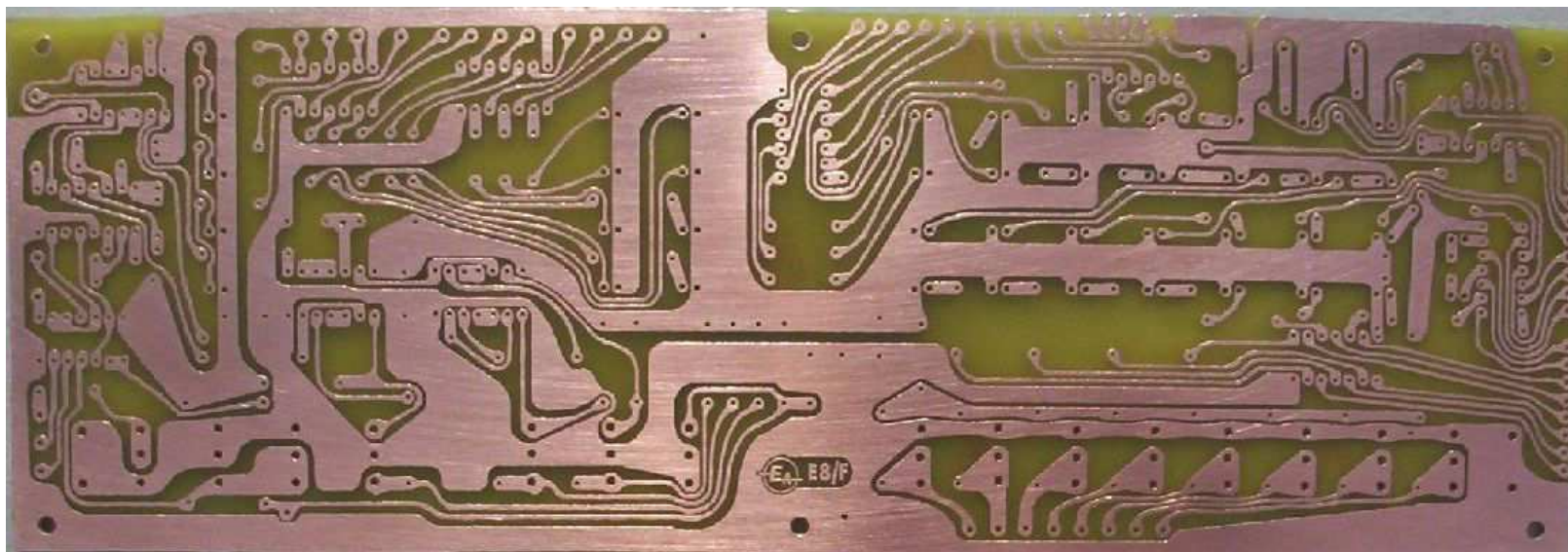
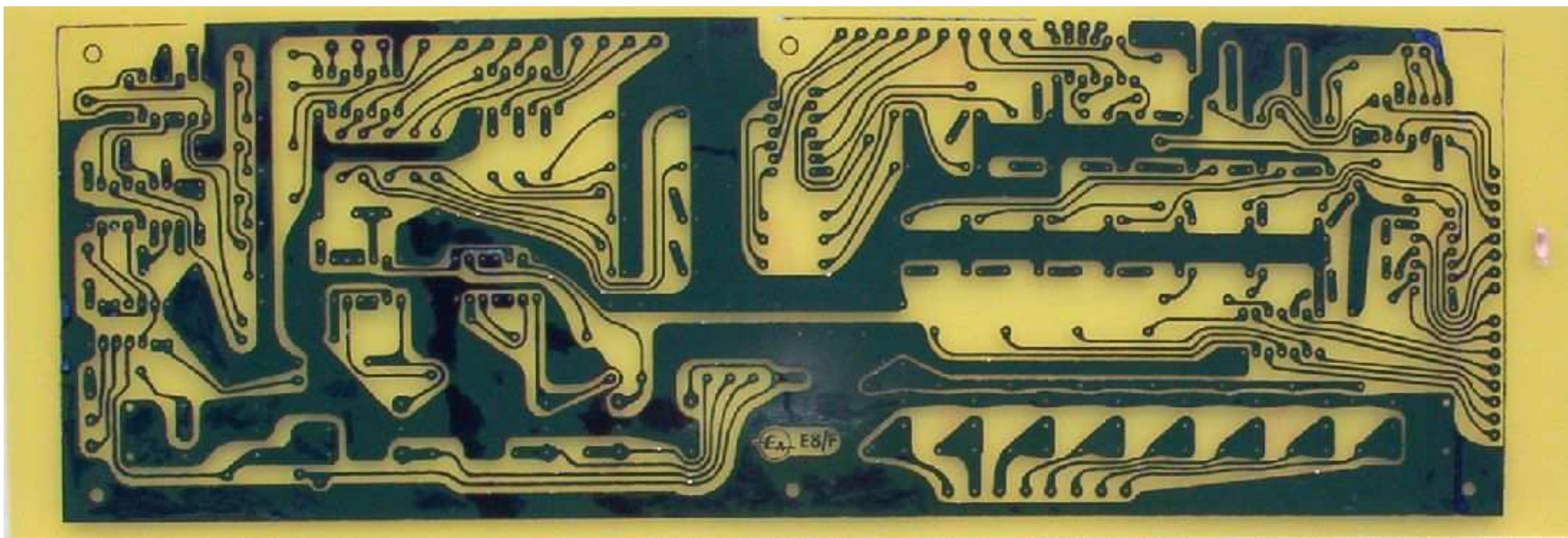


Etching Boards

- The boards were etched using a large vertical etching tank from Compu-tronics with Ammonium Perchlorate solution. This is heated to 50 C using a fish tank heater with air bubbled through to mix the solution.
- The solution starts off clear and then turns blue from the dissolved copper. After a few minutes, the boards are etched.
- After etching, the boards are drilled using a vertical drill stand and carbide drill bits (0.6 to 1.6 mm).
- The green TRF and toner are removed using a steel scrubbing brush.
- A clear PCB lacquer is then sprayed to prevent corrosion.

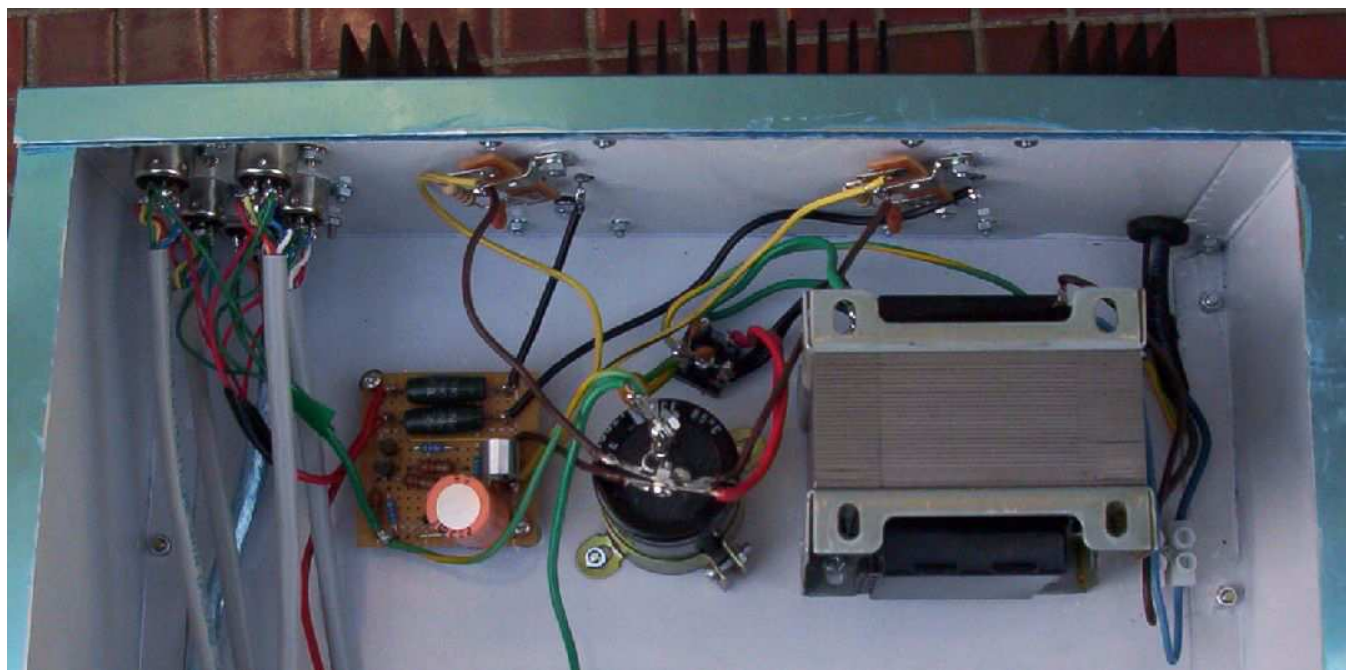


Etching Boards (continued)



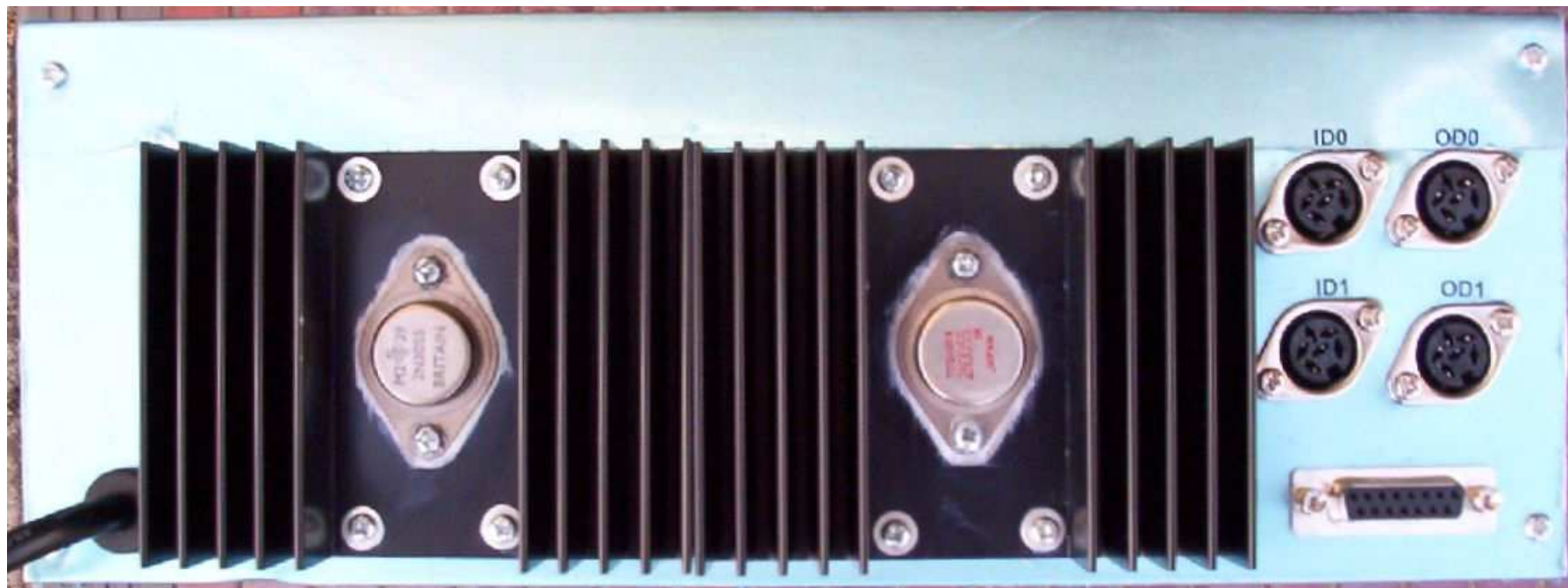
Power Supply

- The EDUC-8 uses a custom power supply to generate +5 Volts at up to 6 Amps, 30 Watts in total. 15 W is used by the computer, with the remaining 15 W available for peripherals. The original design used a 240V/10V 60VA transformer. Used 230V/2x9V 100VA transformer from RS.
- This is then regulated with a PB40 bridge rectifier, followed by two 2N3055 NPN power transistors in parallel (cooled with two large heat sinks), and a 22,000 μ F 6.38 A ripple current capacitor. A 6.2 V Zener diode is used to control the voltage.



Peripheral Interface

- Each of the two input and two output 6-pin DIN connectors had +5V and GND to power the peripheral, $\overline{\text{CLOCK}}$ and $\overline{\text{DATA}}$ to transmit or receive data serially, $\overline{\text{FLAG}}$ for reading the flag, and $\overline{\text{RESET FLAG}}$ for clearing the flag.
- A 15-pin D connector can be connected to a paper tape reader for programming the computer (originally a 16-pin connector combined with an input port). Eight bits go to the switch register, a $\overline{\text{DEP}}$ input to deposit the data into memory, and a $\overline{\text{LA}}$ input to load the address into MA.

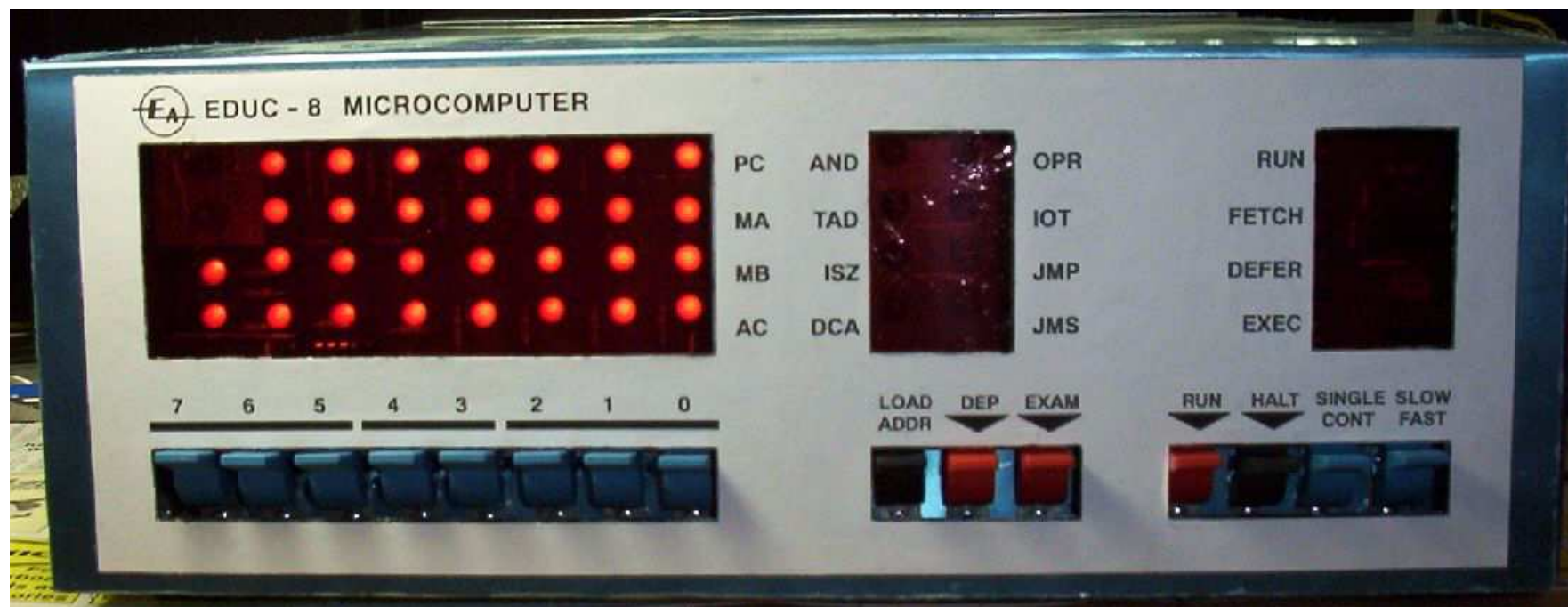
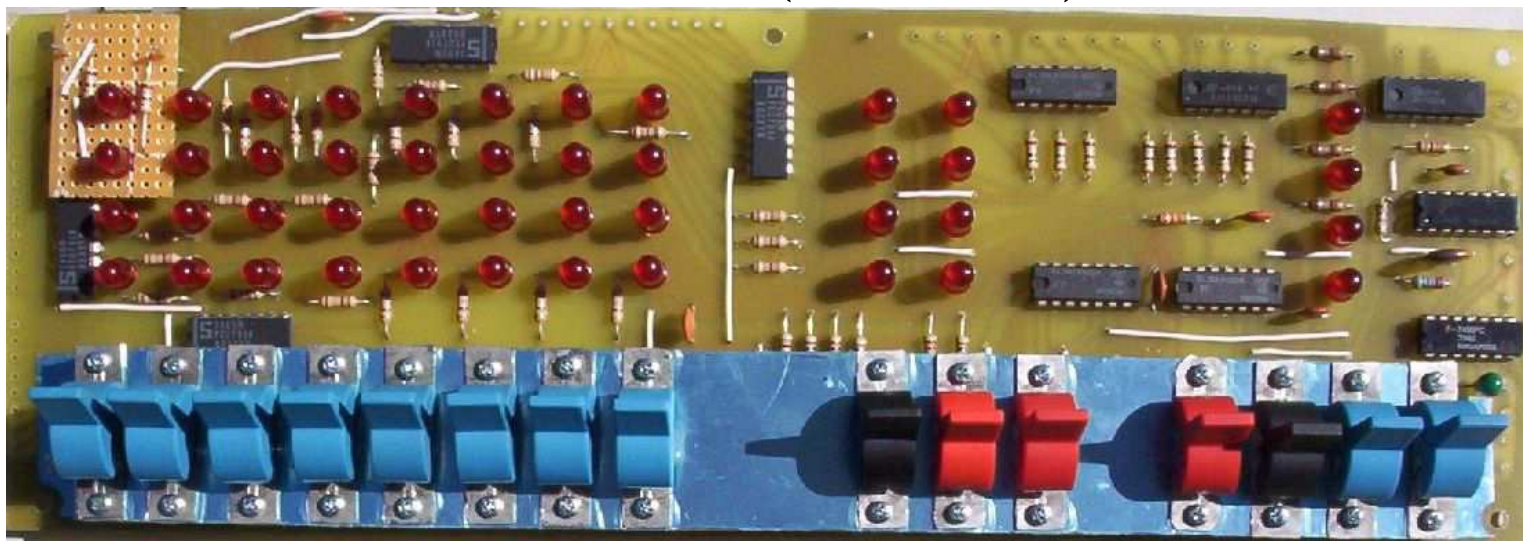


Front Panel

- Very similar to the PDP-8, the front panel of the computer allows it to be programmed and read the status of the computer and its memory contents.
- The left side of the computer uses LEDs to display the contents of the four internal registers PC, MA, MB and AC. Below this are eight switches which form the switch register (SR).
- In the middle are eight LEDs to display the operation code of the current instruction. Below this are the programming switches; LOAD ADDR to load the address from SR into PC, DEP to deposit SR into address PC and then increment PC, and EXAM to examine the contents of PC and then increment PC.
- On the right are four LEDs to display the computer status, RUN for a program is running, FETCH for the fetch instruction cycle, DEFER for an indirect address instruction cycle, and EXEC for the execute instruction cycle. Below are four switches; RUN to start a program, HALT to stop a program, SINGLE/CONT for single instruction or continuous operation, and SLOW/FAST for a slow (2 Hz) or fast (500 kHz) clock.
- In SINGLE mode, RUN is pressed once for each instruction.

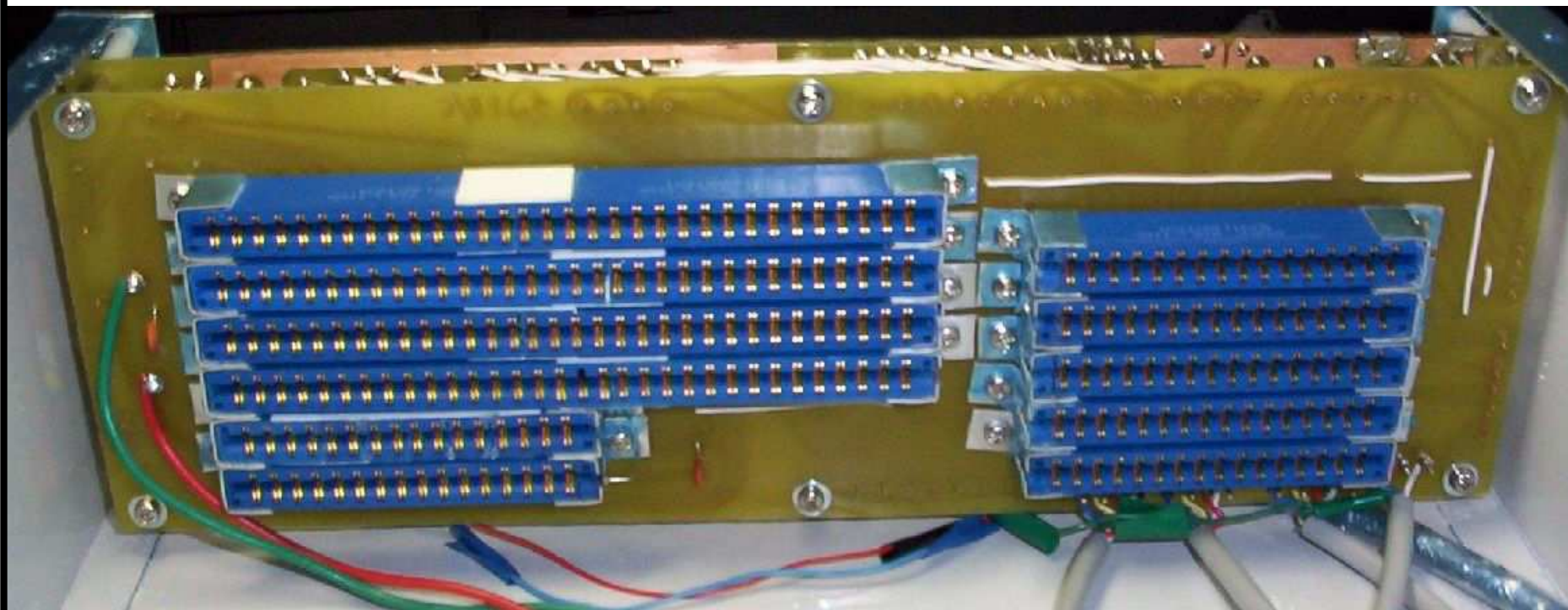


Front Panel (continued)



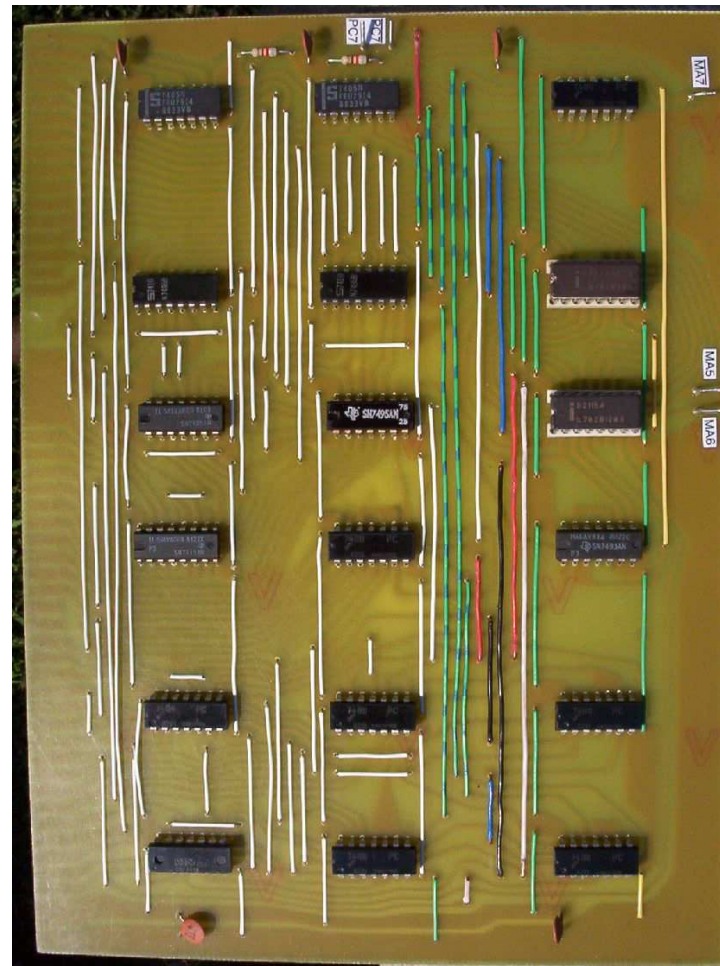
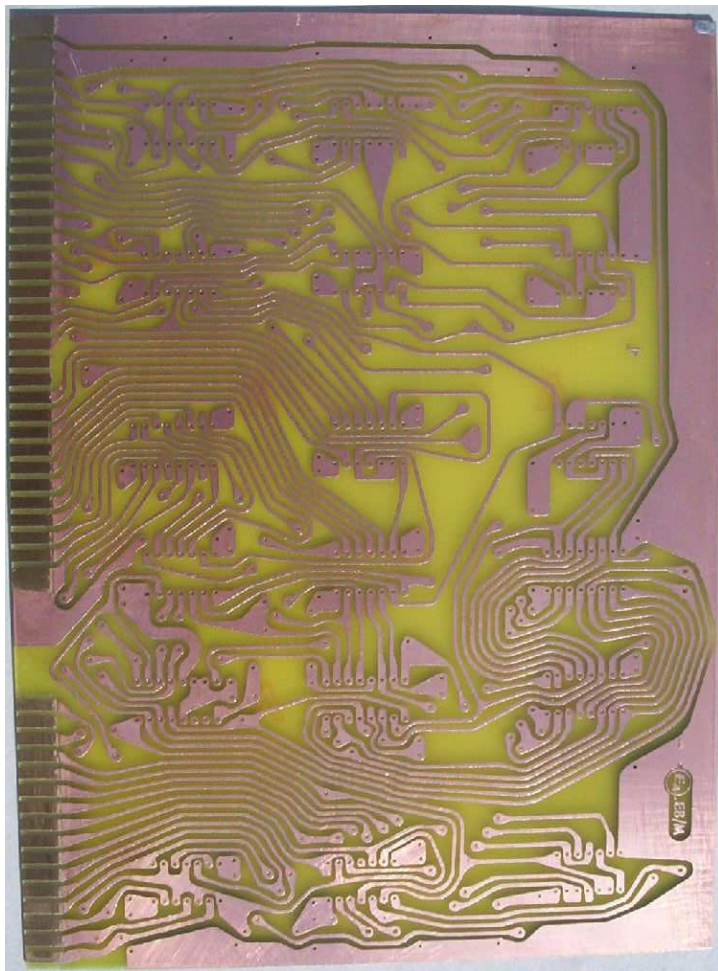
Connector Board

- The connector or mother board is bolted behind the front panel. It allows the six computer boards to be easily plugged into the computer.
- Three boards have a 32 and 16 pin connector, two with two 16 pin connectors and one with one 32 pin connector. Wires along the top and sides of the boards are soldered between the front panel and connector boards. The power supply and peripheral wiring are soldered to the connector board.



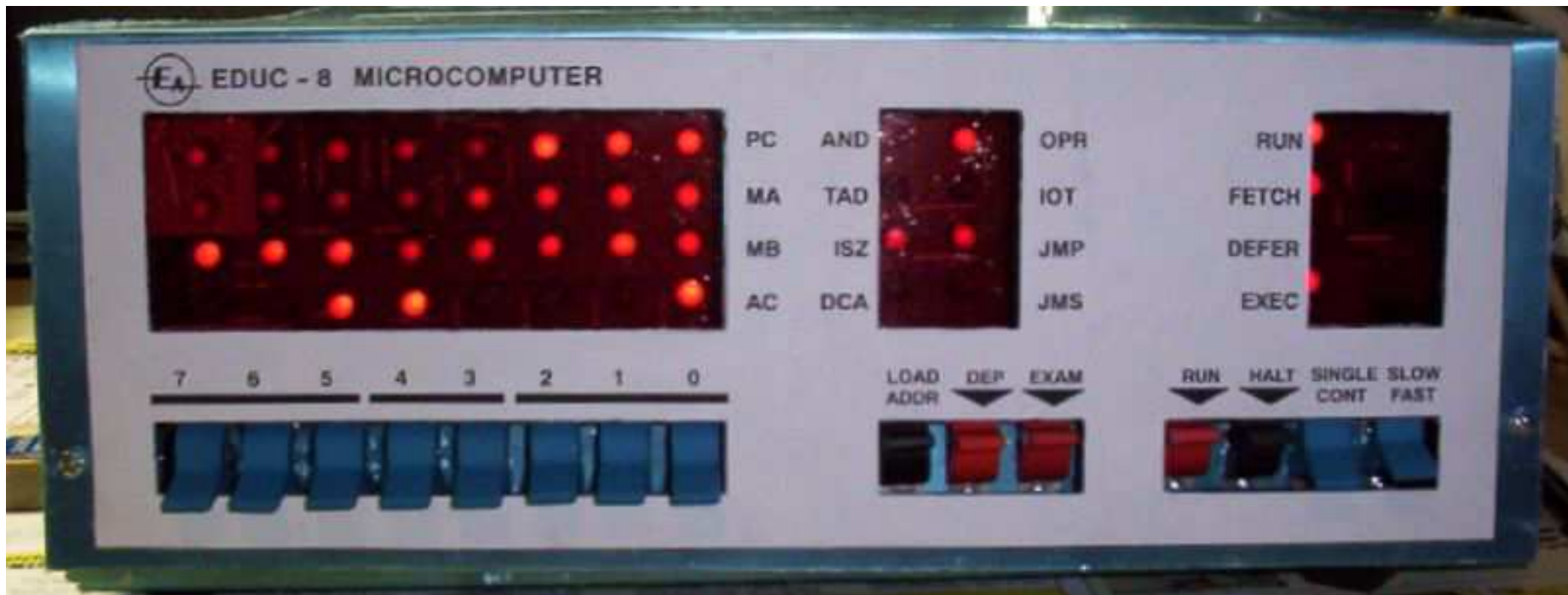
Main Boards

- The six main boards were produced in the same way as the front panel and connector board, except the edge connector contacts were gold plated at Delway Brass. Each board requires many wire links to be soldered first.



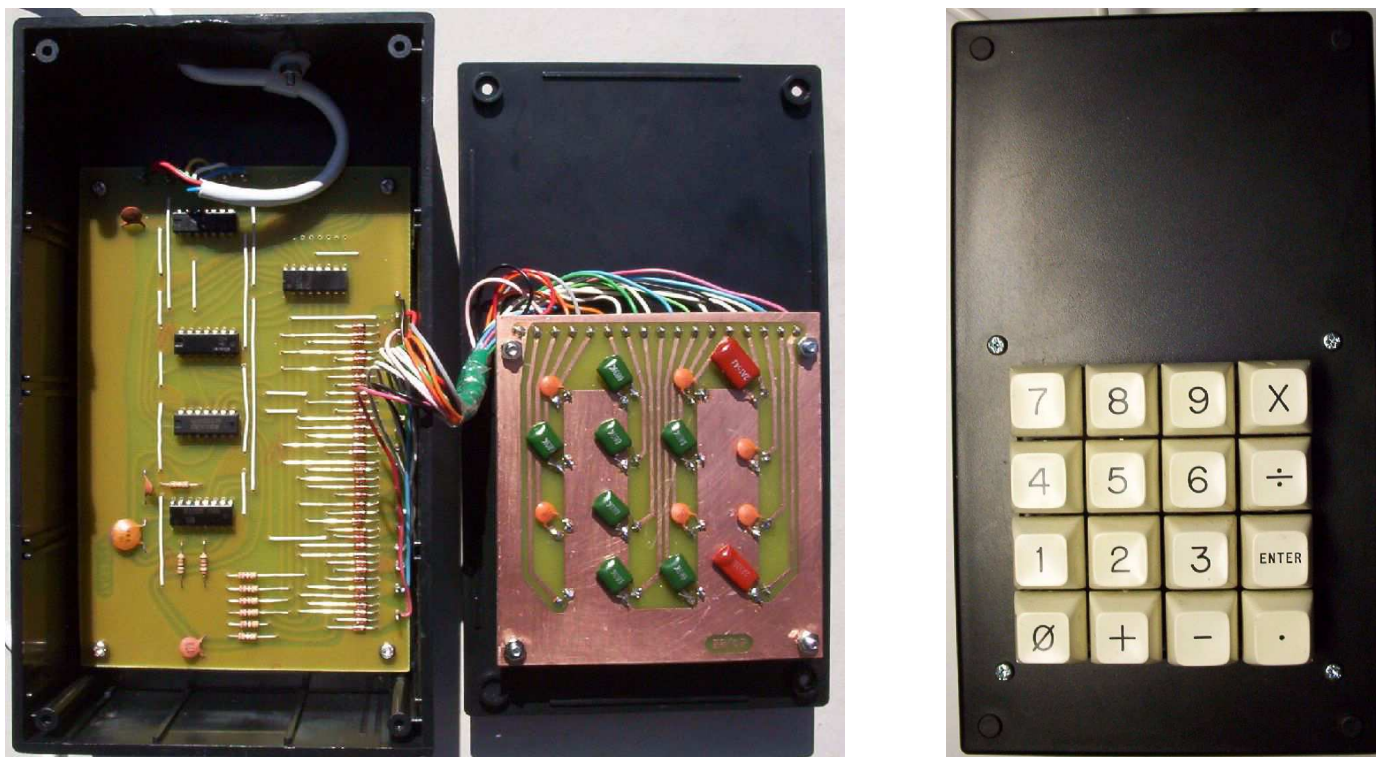
Debugging

- As the computer is built up, it can be tested incrementally, following the instructions in the articles. Four faulty 7400 quad NAND gate ICs were found and replaced. The faulty parts were found using the extender board, a multi-meter, a CRO and operating in SLOW mode.
- The first program which counts AC from 0 to 255 was run with success!



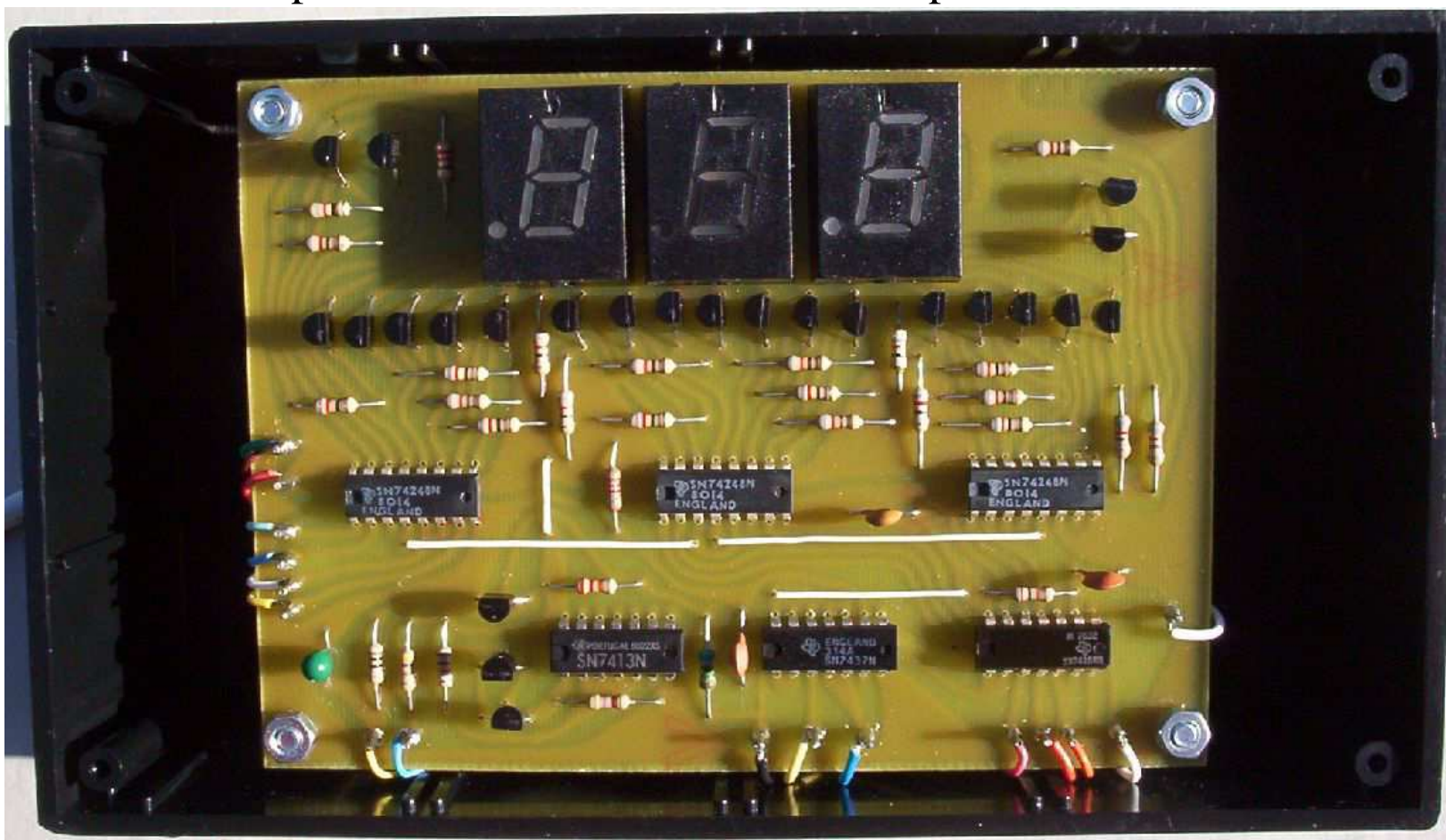
Simple Keypad

- A simple 16 digit keypad peripheral was also constructed. The original used a keypad with mercury bounceless switches. We used a similar keypad from a Commodore 64 system, but with non-bounceless switches.
- During testing, a wiring error was found in the PCB artwork which was easily fixed. The non-bounceless switches were causing improper operation. This was fixed by replacing a 7400 IC with a 74132 quad NAND gate IC with Schmidt trigger inputs. Also, capacitors were added across the switch inputs.



Octal Display

- Also constructed was an octal display using three old DL750 7-segment LED displays from Electronic Surplus in the US. The display can be used to display 8 bits either in 377 or 737 format. The display also includes a timer which uses the FLAG to operate. This worked first time no problem.



Programming

- The 16 word page length (compared to 128 words for the PDP-8) makes programming the EDUC-8 very difficult.
- For working with arrays, the index variable must be in the same page as the instruction in order to use indirect addressing to the value in the array. The PDP-8 solves this problem by using a bit to address page 0. Thus, the index variable can be in page 0 and able to be referenced from any location in the 4K address space. Due to the reduced word length, the EDUC-8 does not have page 0 addressing.
- We have written an assembler called E8ASY to make programming the EDUC-8 a little easier. This required 689 lines of Pascal code. Instructions have to be grouped into 16 word pages, leaving room for variables or constants in each page. The assembler automatically inserts the variables or constants into each page.
- A memory test program was used to test the memory. We have also written programs using the keypad and octal display for writing and displaying code, eight digit addition subtraction, eight digit multiplication and eight digit division/modulos.



Programming (continued)

- Here's a simple program in assembly used to read the memory contents of EDUC-8 with the keypad and octal display.
- Each 16 word page begins with a / followed by the page number. In this example, we only have one page. Variables and constants are declared at the beginning of the program.

```

IDX,          VAR                      /Program Index

/0
START,        CLA
              TAD I  IDX
              LDS 0                    /write(P[IDX])
              TAD IDX                  /AC := IDX
TESTCHAR,     SKF 0                    /if CHARFLAG = 0
              JMP TESTCHAR            /then TESTCHAR else
              RKF 0                    /clear FLAG
              INC  IDX                 /IDX := IDX+1
              JMP  START

```



Programming (continued)

- Here's the output of E8ASY. All codes and addresses are in 737 format. The start address of the program to be examined, e.g., 020 is loaded into address 011, the IDX variable. The start address of the actual program is then loaded, in this case 000 and the program started by pressing the RUN switch.

```

/0
000 START,          CLA          710
001                TAD I  IDX    131
002                LDS 0         622 /write(P[IDX])
003                TAD IDX       111 /AC := IDX
004 TESTCHAR,      SKF 0         601 /if CHARFLAG = 0
005                JMP TESTCHAR  504 /then TESTCHAR else
006                RKF 0         604 /clear FLAG
007                INC  IDX       211 /IDX := IDX+1
010                JMP START     500
011 IDX,           VAR          000 /Program Index

```



Other Peripherals

- The main construction articles for the EDUC-8 were from October 1974 to January 1975. Articles from March 1975 to August 1975 described constructing interfaces to various peripherals.
- March: Interfacing EDUC-8 with punched paper tape.
- April: EDUC-8 with a Philips 60SR printer unit.
- May: A full ASCII-type input keyboard for EDUC-8.
- June: Teaching your EDUC-8 to play a melody.
- July: Interfacing EDUC-8 to teleprinters and magnetic tape.
- August: Interfacing a Burroughs self-scan display panel to your EDUC-8.
- We have obtained most of the parts or their equivalents and hope to complete all these projects. The most difficult to obtain was a paper tape punch and reader.



Expansion and Conclusions

- A page 0 bit would work by having instructions reference the first half of page 0 or the second half of the current page. This takes advantage of the fact that there can be no more than eight variables or constants in a page, corresponding to eight memory reference instructions.
- To expand the number of ports, one IOT port would be used to select up to 256 additional ports. The other IOT port would access the selected port.
- To expand the memory up to 64KB, two of these ports would be used to select a field of 256 words of memory. Just like the PDP-8, one would be an instruction field buffer (IFB) and the other a data field (DF). To prevent wild program execution, the IFB is copied into the instruction field (IF) only during a JMP or JMS instruction. The DF is used in indirect addressing.
- The computer, keypad and octal display cost \$1113 for parts and supplies and \$717 for the tools to make the PCBs. Total cost \$1830.
- This has been a fun and challenging exercise into learning how early computers worked, how they were built and were then programmed.



<http://www.sworld.com.au/steven/educ-8/>

