Repairing the HP9866A Thermal Printer

Dr A. R. Duell[1]

March 2011

[1]ard@p850ug1.demon.co.uk

Contents

Chapter 1

Introduction

This manual is a 'follow on' to my manual on repairing the HP9800 series of desktop calcualtors. It covers the HP9866A printer that was often used with the HP9830 mnachine and which, wjhile being a separaet product, is almost a part of that system

After an overview of the printer operation, this manual continuses with descriptions of the major sections of the printer electronics, including a state-by-state analysis of the control system finite-state machine. Two final chapters cover dismantlign the printer and locating faults.

I shall assume that the reader has already read the HP9800 manual and thus, since much of the general repair tips, necessary tools and test gear and safety information are the same for the HP9866 as for the HP9800 machines, this information will not be repeated here.

As in the case of the HP9800 calculators, the official HP service manual is mostly a 'boardswapper guide', although it does contain schematics of the **Logic PSU** and **Motor Driver** PCBs. A full schematic is however available and is almost essential for repairing this printer. The source listing of the control state machine is also available, but as this is included in a later chapter, it is not necessary ot obtain this lisitng.

Chapter 2

Overview of the Pritner Operation

2.1 Basic operation of the HP9866A

Since the operation of the HP9866A is not as well-known as that of the HP9800 machines themselves, this section will give an overview of the printer operation.

The design of the printer's control electronics was dictated by the design of the mechanism, and more particularly the printhead. This is fixed to the printer chassis, it does not move across the paper, and consists of 400 tiny heating elements in a horizontal row across the paper. These elements are not evenly spaced, rather they are in 80 groups of 5, forming the 5 dot columns of the 80 characters that this printer prints on each line. When the heating element is energized, it causes the (specially-coated) paper to darken.

Therefore, at the most basic level, the operation of the printer is as follows :

1. Read in characters from the host and store them in an internal line buffer memory until an end-of-line character is received

2. Read out the buffer store, one character at a time, convert it to the appropriate bit pattern for the next dot line of that character and send it to the print head, where it will be stored, but do not energise the print elements yet

3. When all 80 characters have been processed, turn on the selected elements for a given time

4. Advance the paper one dot line.

5. If not all 7 lines have been printed, go back to step 2 to print the next line of dots.

6. When all 7 lines have been printed, advance the paper to give the inter-line gap

Unfortunately, in practice, it's a little more complicated. To save storage in the printhead circuit, and more importantly to reduce the maximum current that might be drawn from the power supply, only a quarter of the printhead elements are energised at a time. On the first 'pass' through the buffer store, characters 0,4,8,12,... are printed. On the next pass 1,5,9,13,... and so on.

Also if fewer than 80 characters are send by the host before the end-of-line character, the remainder of the buffer will presumably contain random data. Therefore it is necessary to fill the remainder of the buffer with space characters when the end-of-line character is received before attempting to print the line.

Therefore a more complete description of the printer operation is :

1. Read in characters and store them in the buffer until an end-of-line character is received

2. When the end-of-line character is received, fill the remainder of the buffer with spaces

3. Read out every fourth character of the buffer and convert it to the correct bit pattern for the next line of the character, then end that to the printhead

4. When 20 characters have been processed in this way, energise the printhead elements for the standard time

5. If all 80 characters have not been processed, then select the next quarter of the buffer and go back to step 3

6. When the entire dot line has been printed, advance the paper one dot line and go back to step 3 to print the next dot lien of the characters

7. When all 7 dot lines have been printed, advance the paper for the inter-line gap

2.2 Implementation and layout

Before considering the theory of operation of the various sections of the printer it is helpful to have a basic idea of the design and layout. As with many computer-related devices, the electronics can be split into 2 main sections, namely the **data path** and the **control**.

The **data path** contains the buffer memory register which is implemented using 80-bit shift registers. It also includes the character generator ROM and the printhead storage register, again built from shift registers.

The **control** section is based round a 32-state finite-state machine. Associated with this are are several binary counters which keep track of the number of characters in the buffer store and the number of dot-lines printed.

Physically the electronics consists of 7 PCBs. The 2 main logic boards contain the **data path** and **control** sections. These are plugged into the **logic backplane** PCB along with a further PCB, the **Logic PSU**. Plugged into the side of this backplane is the **Printhead Driver** PCB which contains the printhead storage shift registers. 2 Further PCBs, the **Printhead PSU**, which provides the high voltage needed for the printhead elements, and the **Motor Driver** are plugged into connectors mounted on the main chassis. Also separately mounted are some of the larger power supply components such as the mains transformer.

Mechanically the printer mechanism is very simple. Since the printhead covers the full width of the line and does not move, the only motion is that of the paper. The platen roller is directly coupled to the stepper motor spindle, a pressure roller held against the platen is coupled to the paper loading roller by a toothed belt.

Chapter 3

Data Path Theory of Oepration

As mentioned earlier, the control system has 2 modes of operation. In the first, characters are loaded into the buffer store from the host. In the second, the contents of the buffer are printed. The **RotEn** signal determines the current mode. This signal is produced by the dot line counter. When it is low, the printer is in input mode, and when it is high the buffer shift register rotates (hence the name of the signal – ROTate ENable) and the buffer is read out so the contents can be printed. A clear understanding of this signal will make understanding the rest of the printer much easier.

The 2 operating modes of the data path circuitry will be described separately.

3.1 Input Mode

The input register consists of 7 D-type flip-flops U226, U17, U11 and U6 on the **Data Path** PCB. Characters are loaded into this register from the host by the **InClk** signal which is derived from an output of the control state machine by U21c. Note that the data lines from the host (**IFD(n)**) are active-low signals, so that the Q outputs of this register are also active low, while the Q/ outputs are active-high.

The outputs of 6 of the flip-flops are gated into 6 80-bit shift registers, since this printer only handles upper-case characters, 6 bits are sufficient. The least significant bit (**InData(0)**) will be described, all others are similar. The output of bit 0 of the input register, U22b, is fed to the AND-OR-Invert gate U15b. Since **RotEn** is low, U20f sets **InEn** high and the data is passed to the input of U14b, which is an 80 bit MOS shift register. The clock for this register is produced by buffering the **SClk** signal from the control state machine with U20a and the transistor Q1.

At the end of the input line, the remainder of the shift register must be filled with space characters. The **ForceSpace/** signal from the state machine goes low, and since **InEn** is high, **Space/** is driven low by U16a. This logically sets bit 5 of the input register and logically clears the remaining bits, thus forcing the register to contain the ASCII code for space. The state machine then produces the appropriate number of **SClk** pulses to fill the remainder of the shift register with space characters

There are 2 circuits that decode the input character and provide inputs to the state machine. When a linefeed character is received, the output of U5 (**LF**) is asserted (low). This indicates the end of the input line and causes the state machine to start the print cycle.

The other circuit asserts the **PrtChar** (printable character) signal. If the character is not a control code (that is, at least one of bits 5 and 6 is asserted, detected by U16d), and it's not DEL (detected by U10) or if it's a linefeed, the output of U16b goes high, indicating that the character is to be processed by the printer, this is fed to an input of the control state machine.

3.2 Print Mode

In this mode, **RotEn** is high, so **InEn** is set low by U20f. Again considering bit 0 of the data shift register, the output of the shift register U14b is inverted by U7b and fed back to the AND-OR-Invert gate U15b. This gates this signal back to the input of the shift register. Thus the shift register recirculates (rotates) and the data can be repeatedly red out. The output of U7b is further inverted by U7a, providing an active-high character code bit. The 6 character code bits are fed to the low 6 address inputs of the character generator ROM to select the appropriate character pattern, the high 3 address lines select one row of that pattern and come from the row counter, which will be described as part of the control section.

The character generator is U1, and is a 512*8 ROM of the same type as are used for the HP9800 firmware. It is permanently enabled (**CE** and **CS** are both connected to +12V). 5 outputs of this ROM, containing the bit pattern for the current row of the current character are inverted, e.g. by U13c, and fed via the backplane to the **Printhead Driver** PCB. There are 5 identical circuits on this board, corresponding to the 5 dot columns of each character, again only one – column 0 – will be described.

The **PHData(0)** signal coming into the **Printhead Driver** PCB is inverted by U4a and fed to the 20 bit shift register consisting of U9 and U6. The character rows are shifted into this register by the **PHCLk** signal from the control state machine, inverted by U4b. This shift register provides 20 of

the printhead data signals, those for the first dot column in each character, **Drv(0), Drv(5)...Drv(95).** The 5 registers thus provide all100 signals needed to control the selected section of the printhead. These signals are wired to the connectors for the 4 printheads.

Each printhead takes in 25 of the signals and contains circuitry to energise the appropriate elements selected by the **Bank0...Bank 3** signals. Again, the operation of these signals will be described with the control logic.

Chapter 4

Control System Theory of Operation

4.1 Control state machine

As was mentioned earlier, the control system is based around a finite state machine with 32 states. Therefore the state can be represented as a 5 bit number which is stored in the 5 D-type flip-flops U2a, U6b, U6a, U18b and U2b on the **Control** PCB. The outputs of these flip-flops form the address for the 32 byte ROM U12 which contains the state table.

A description of this state table, which could be considered to be the control firmware of the printer, will be given later, but at present only the hardware will be described.

The next state is of course determined by the logic levels on the D inputs of the flip-flops. The highest 2 state bits are simply determined by 2 of the ROM outputs, while the **CC** (Condition Code) signal selects one of 2 ROM outputs for each of the lower 3 state bits. For example, consider the input to U2a, which holds the lowest state bit.

If **CC** sis low, then U1d sets **CC/** high. Therefore U1c is inhibited and its output is forced high. U1a inverts **SMD(2)** and passes it to U1b, which inverts it again., So in this case, the input to U2a is **SMD(2)**. If **CC** is high, then U1a is inhibited, while U1c inverts and passes on **SMD(7)**, which is then re-inverted by U1b. So in this case the input to U2a is **SMD(7)**.

So, depending on the state of **CC**, the new state is given by the SMD bits as follows ;

CC=0 : 6,4,0,1,2

CC=1 : 6,4,3,5,7

There is, of course, no reason why the ROM can't be programmed so the 2 possible next states have the same value, meaning that the state of the **CC** signal is irrelevant. Needless to say, such unconditional transitions are quite common in the state machine.

The **CC** signal is thus the control input to the state machine. It comes from the 16 input multiplexer U14, which selects the various signals that need to be tested. The first 8 inputs of this multiplexer as used in print mode, the last 8 in input mode, since the D select input is driven by **RotEn/**. The other 3 select inputs are driven by 3 of the ROM outputs – thus the next state is partially determined by the signal to be tested.

The 16 output decoder U13 is driven by the lower 4 state flip-flops. It provided the output signals from the state machine to the rest of the logic. Thus each output signal is produced in different stares (since the highest state flip-flop is not used by this decoder). However, since **SCLk**, the clock signal to the buffer shift register, is required in 4 states, 2 of the outputs of the decoder are logically ORed by U20d.

4.2 Master Clock and Reset

The clock for the control state machine comes initially from the RC oscillator formed from U8a, U8b and U8c on the **Control** PCB. This provides a clock signal of approximately 3.5MHz which clocks the 2 cross-coupled JK flip-flops U7a and U7b. These cycle through the stares 00, 10, 11, 01, 00... and thus produce a pair of signals of a quarter the frequency of **MCLk** in phase quadrature. The output of U7b is used to clock the state machine, that of U7a enables the decoder U13 when it is low. This has 2 consequences. The first is that the outputs of U13 pulse every cycle of the clock divider even if the state machine remains in the same state. The second is that the outputs of U13 re disabled when the state machine is changing state, thus presenting glitches on these outputs.

If the **Wait/** pin on the test connector is pulled low, U20c will cause the clock divider to halt in state with OutEn/ low, that is with U13 enabled. This facility does not seem to be particularly useful.

In order to start the system off in a known state, the **Init/** signal is pulsed low by the **Logic PSU** at power-on. This is buffered by U27c and U27f on the **Control** PCB and clears all the state flip-flops, and other parts of the logic. This also asserts the **Rst** signal via U16c, which clears the interface logic. This logic can also be cleared by **IFInit**, provided the printer is in input mode – the inverted form of **IFInit** is ANDed with **RotEn/** by U16a.

4.3 Interface Handshake

In order to describe the interface handshake, the particular example of an H(830 host will be used.

The trailing edge of the **LatchEn** signal on the HP9830 **I/O Backplane** sets the flip-flop U3a. The Q/ output of this flip-flop goes low, causing the printer to appear busy via U1b. The Q output asserts **PrtStb/** via U2c.

This strobe signal is fed to the **Control** PCB in the HP9866 where is it called **IFStb/**. It is received by U23c, and provided the printer has paper and that the manual feed button is not pressed (that is, **PaperOut/** and **Feed/** are both high), the output of U22b goes low. This is an input to the state machine input multiplexer, U14.

The state machine now begins the character read sequence. When it asserts **InCLk/**, the data is latched into the input register. The trailing edge of this signal sets U22a on the **Data Path** PCB, which asserts **IFFLg/** via U20c.

Back on the HP9830 **I/O Backplane**, this flag signal, now called **PFLg/** clears U3a and thus removes the strobe. It also causes the busy signal to be maintained via U1b.

When the HP9866 has processed the incoming character, the state machine asserts the **CLrFLg** signal. This resets U22a on the **Data Path** PCB and de-asserts **IFFLg/**. The printer is now ready for another character.

4.4 Control Counters

It is necessary for the control system to keep a count of the number of characters stored or printed and the current dot line. In order to simply the control system, this information is not stored in the state machine, instead separate counters are used. There are 3 such counters, one counts the characters transfered to or from the buffer shift register; the second counts the rows of dots being printed and the last is a simple 4 state counter used to count the shifts when the shift register is moved on by 4 characters during each section of the print cycle. These counters will be described individually.

4.4.1 Character Counter

the character counter is an 80-state counter consisting of the divide-by-10 counter U4 and the divide-by-8 counter U10 on the **Control** PCB., It therefore counts from 0 to 79 in BCD.

Two signals increment this counter – they are logically ORed by U16b. During the character input operation, the **PHOut/** signal (which does not enable the printhead elements in this mode) is gated via U15d to produce **IncCC**. During the print cycle, the **PHClk** signal from the state machine, which loads the bit pattern into the printhead shift registers increments this counter. It is important to realise here that the count is not an 'index' of the characters in the data shift register. During the print cycle, the bit pattern of every fourth character is transfered to the printhead shift register, but the counter is only increments once for each character transfered.

The outputs of this counter are decoded by NAND gates to produce condition inputs for the control state machine. The output of U16d, **Newbank/** is asserted when the character count is divisible by 20, that is, when a quarter of the characters have been transferred to the printhead shift register, and thus this section of the dot line should be printed. The **Char0/** signal is asserted by U9a when the count is zero, that is, when the count has wrapped around and all characters have been processed. Finally, U15c produces **Fristback/** when the first quarter of the dot line is being printed.

Two other circuits are associated with this counter. The SR flip-flop produced by cross-coupling U15b and U22c is cleared when a dot line is printed and set when the character counter is incremented. It thus indicates whether any characters have been loaded into the buffer shift register. The output of this flip-flop, **Empty/**, is combined with **Char0** by U11b. Its output, **Start/** therefore indicates whether the fact that the character counter is 0 indicates. that the buffer is full or not

The other related circuit is located on the **Printhead Driver** PCB. Here, the NOR gates in U1 and U2 decode the top 2 bits of the character counter, and when **PHEn/** is asserted (the source of this signal will be described later), the appropriate bank signal is turned on, energising the printhead elements. Since all 4 of these circuits operate in a similar manner, only one will be described.

For example, for the second quarter of the line, **PHSelA/** is 1 and **PHSelB** is 0. Therefore, when **PhEN/** goes low, the output of U1b goes high, the outputs of U2a, U1a and U2b all remain low. The output of U1b going high turns on the PNP transistor U8d via U3a and thus applies the printhead power supply voltage to the Bank2 signal.

4.4.2 Row Counter

This is a 10 state counter consisting of U12 on the **Data path PCB**. It is incremented by the **RowClk** output from the control state machine and keeps a count of the number of dot rows printed. When this counter is in

state 0, **RotEn** is set low by U19a, selecting the character input mode. At the start of the print cycle, this counter is incremented, asserting **RotEn** and selecting the print cycle mode. The output of U19b goes low during the 7th row of dots; this is the last row and thus the state machine tests this at the end of each printed row to check if the complete character line has been printed.

4.4.3 Shift Counter

In order to count the shifts when moving on 4 characters during the print cycle, a simple 4 state counter on the **Control** PCB is used. It consists of the interconnected D-type flip-flops U17a and U17b. During the print cycle, the **ForceSpace/** signal (which has no other use in this cycle) increments this counter via U20a. It counts in the sequence 00, 01, 11, 10, 00...but the exact sequence is not important for the operation of the printer. What is important is that the output of U11a, **LnFlg/** is asserted every 4 cycles. This signal is a condition input to the state machine.

4.5 Timing Monostables

The printhead takes a significant time to change the state of the thermal paper, similarly the stepper motor takes a significant time to move the paper up by one dot row. Three monostable multivibrators are used to determine these times.

In order to print the current dot pattern (already loaded into the shift registers on the **Printhead Driver** PCB, the state machine asserts **PHOut/.** Since **RotEn** is high, the monostable U24 is triggers via U20b, thus asserting **PhEN/** for about 6ms. This signal energises the printhead elements via the decode circuit on the **Printhead Driver** PCB. Monostable U26a is also triggered, it has a time of around 7.16ms. This output of this monostable is gated via U25 and synchronised by U18a to provide a condition input to the state machine which then waits until the printhead has 'burned' the pattern onto the paper before continuing.

Similarly, to move the paper up one dot row, the sate machine produces the **AutoFeed/** signal. This triggers the monostable U26b, with a period of 8.2ms. This is similarly fed via U25 to the state machine input to cause the state machine to wait for the motor to finish advancing the paper before attempting to print another row of dots.

4.6 Paper Sensor

The presence of paper in the printer is detected by a changeover microswitch under the chassis. This microswitch is debounced by the SR flip-flop U28a and U28b on the **Control** PCB to produce the **PaperOut** signal and its inverse. This signal has 3 effects : It inhibits the input strobe signal via NAND gate U22b, thus preventing the printer form accepting characters from the host; It turns on transistor Q1 and thus causes the 'Load' indicator on the front of the instrument to illuminate; and finally it assets the **IFPout** signal via U20b on the **Data Path** PCB to indicate to the host that the paper has run out, however this signal is not used by the HP9830.

Chapter 5

The COntrol State Machine

As was mentioned earlier, the central section of the control system of this printer is a 32-state state machine. This is sufficiently small to be easily understandable, so a state-by-state description of it will be given.

From the current state and the contents of the ROM location associated with that state it is possible to determine the condition that will be tested, the possible next states and which (if any) output signals are generated. And thus a 'source listing' of the state machine can be produced. This will be given below in 3 parts, the character entry routine; the (quite short) paper advancing routine and finally, of course, the print cycle.

Each state will be given as a pair of hexadecimal numbers followed by a 1-line description. The first number is the current state. the second number is the contents of the state machine ROM for that state. The description decodes these numbers to give the condition that is tested, the possible next states and the signal generated (if any). This will be followed by a longer description of how this state affects the printer.

5.1 Character Input

This routine reads characters form the host interface. until a linefeed is received. It ignores non-printable cheaters and stores others in the buffer shift register. When a linefeed is read, then if no characters have been loaded since the last linefeed, the paper is simply advanced, otherwise a full print cycle is started.

```
00 : 20    if(Stb/) goto (00,02) , ClrFlg
```

The character entry routine starts in state 0, which is also the state where it starts on a hardware reset (after power-on, the printer needs to load characters before it can print them). The interface flag flip-flop is cleared, indicating

to the host that the printer is ready to accept a character. The machine remains in this state until it gets a strobe signal from the host indicating that there is a character to load. it then goes to state 2

```
02 : 2B    goto (06)
06 : C2    if(PrtChar) goto (12,11) , InClk
```

State 2 is a no-operation state, added simply due to the interaction between the states, next states, condition and output signals. It goes directly to state 6. This state outputs **(InClk) which loads the character into the input register and also sets the interface flag flip-flop, indicating the printer is busy. The character is tested to see if it is one that the printer should handle (that it is not a control character other than linefeed). If it is not to be processed, the state machine skips over the input routine, thus ignoring this character.**

```
12 : F4    if(LF/) goto (19,1B)
```

The character is now tested to see if it's a line feed.

```
1B : CF    if(Empty/) goto (17,15)
```

If it was a linefeed, the next test is whether the buffer is empty (that is, whether the 'empty' flip-flop is set). If there are characters in the buffer, start a print cycle, Otherwise just feed the paper.

```
19 : C1    if(Start/) goto (14,11)
```

The state machine gets to this state if it is a printable character. If the character counter is 0 and the buffer is non-empty, it indicates that the buffer is completely full. In that case, ignore the character, thus truncating the line.

```
14 : 9D    goto (0D) , SClk
```

The state machine gets to this state if it's a printable character and there is still space in the buffer shift register. It now clocks the shift register to load in this character/

```
0D : C4    goto (11) , PHOut
```

. In the character entry mode, PHOut will increment the character counter (as one more character has been loaded) and set the 'empty' flip-flop (as there is now something in the buffer)

```
11 : C0    if(Stb/) goto (10,11)
```

The state machine gets to this state when the current input character has been processed. It waits for the host to deassert the strobe signal

```
10 : 00    goto (00) , ClrFlg
```

When the strobe signal becomes inactive, the state machine clears the interface flag and goes back to the start to process the next character.

5.2 Paper Feed

This routine drives the stepper motor and increments the dot row counter until the latter has got to 0. It thus completes the interline gap if characters have been printed and feeds the paper by an entire character line for a blank line.

```
15 : 94    goto (09) , Autofeed
```

Send a step pulse to the motor control circuit to advance the paper by one dot row

```
09 : 93    if(Timing) goto (0E,09)
```

Wait until the timing monostable circuit U26b (on the Control PCB) has timed out, This gives the motor and mechansim time to move.

```
0E : FF    goto (1F) , RowClk
```

Increment the dot row counter

```
1F : C5    if(RotEn) goto (15,11)
```

Go around again if the counter has not got back to 0 (that is, if RotEn is still 1). Otherwise go back and wait for the strobe signal from the host to be inactive and then wait for next character to be sent.

5.3 Print Cycle

This routine prints the contents of the buffer shift register. It begins by filling the remainder of the shift register with space characters, in the process moving the input characters to the start of the lien. It then prints each dot row of the characters by sending each fourth character from the buffer to the printhead shift register via the character generator. Once this section of the line is printed, the remaining 3 quarters are printed in turn. Then the paper is advanced by one dot row and the process repeated until the entire characters have been printed.

The first part of this routine, therefore, fills the remainder of the buffer with spaces.

```
17 : 89    if(Start/) goto (04,05) , ForceSpace
```

Set the input register flip-flops to a space character. Exit this loop when the character counter has wrapped around to 0. This can be thought of as a while(buffer not full) loop.

```
04 : 10    goto (08) , SClk
```

Clock the space into the data buffer shift register.

```
08 : EF    goto (17) , PHClk
```

Increment the character counter and go round again to load the next space character.

The line buffer now full and ready for printing

```
05 : E6    goto (13) , Autofeed
```

Advance the paper by one dot row

```
13 : E3    if(Timing) goto (16,13)
```

Wait for the timing monostable to time out to give the motor and mechanism time to move the paper

```
16 : 3E    if(Row7) goto (0B,0E) , InClk
```

Check to see if 7 dot rows have been printed – that is, the complete character. if so, the go to the paper feed routine, other wise continue with printing this character line

```
0B : 7B    goto (1E)
1E : 84    goto (01) , RowClk
```

Increment the dot row counter. This will automatically set RotEn, thus selecting the print (rather than input) mode for the printer control system.

```
01 : 50    goto (18)
18 : BF    goto (0F) , PHClk
```

Clock the printhead shift registers. This will transfer the bitmap for the current row of the current character into these shift registers

```
0F : DA    if(char0/) goto (1A,1D)
```

Have all 80 characters in the line been processed. If so, then skip the 4 character rotation at the end of the line so that the shift register is left in the same state at the start and end of the line. This is essential so that the dot rows of each character are aligned.

```
1A : AF    goto (07)
07 : 59    goto (1C) , ForceSpace
```

Otherwise, rotate the line buffer by a total of 4 characters to select the next character to be printed in this pass. The ForceSpace signal now increments the 4-state Shift Counter

```
1C : A7    if(LnFlg) goto (07,03) , SClk
```

Rotate the buffer shift register and go round again until LnFlg is asserted. This is a total of 4 times round this loop.

```
03 : D8    if(newbank/) goto (18,1D)
```

Have 20 characters – a complete quarter of the line – been loaded into the printhead shift register? If not, go round again.

```
1D : 32    goto (0A) , PHOut
```

Trigger the timing monostables to activate the printhead and 'burn' this part of the line onto the paper.

```
0A : 31    if(Timing) goto (0C,0A)
```

Wait for the timing monostable to time out to give the printhead time to act.

```
0C : 8C    if(firstbank/) goto (01,05) , SClk
```

Rotate the line buffer by one character to move the next quarter into position. If the character counter has wrapped around then all 4 quarters have been printed, so go back to the start to advance the paper by one dot row and start to print the next dot row of these characters. if not, then go back to print the next quarter of the line.

Chapter 6

Motor Control Theory of Operation

The paper is advanced by a 2-phase stepper motor that is directly coupled to the platen roller. One step of this motor corresponds to one dot row of paper movement. The motor control circuit consists of a circuit to generate a clock pulse to advance the paper, a 4-state counter to generate the 2 signals in phase quadrature and buffer circuits to amplify these 2 signals to actually drive the motor windings. The first 2 sections are located on the Control PCB, the last is contained on the Motor Driver PCB (along with its power supply circuit). These 3 sections will now be described separately.

The motor clock signal can come from 2 sources. For automatic paper advance during printing, the state machine output decoder, U13, assets the AutoFeed/ signal. For manual paper advance, the *Feed* button on the front panel is debounced by the SR flip-flop U28c and U28d to provide the Feed signal. When this is asserted, and providing the printer is not i nthe middle of printing a line – that is that RotEn/ is high (not asserted, NAND gate U12a is enabled and passes the 100Hz signal from the oscillator u23a, U23e, U23d. The (active-low) output of this gate is logically ORed with the active-low AutoFeed/ signal by U15a to provide the MotorClk signal. Thus, when the state machine outputs the AutoFeed/ signal, a single motor clock pulse is produced and the paper is advanced by one dot row. And when the *Feed* button is pressed, a continuous stream of motor clock pulses is produced and the motor runs continuously.

The motor counter consists of the 2 coupled D -type flip-flops U21a and U21b. They count in the sequence 00, 01, 11, 10,

00...The outputs of these flip-flops are thus the 2 required drive signals in phase quadrature .They, and their inverses, are buffered by the open-collector inverters U27a, U72b, U27d, U27e and fed via the logic backplane and chassis wiring to the Motor Driver PCB.

The Motor Driver PCB contains 2 identical driver stages (one for each motor winding) and a simple unregulated power supply. The latter produces ±17V by rectifying and smoothing the output of a centre-tapped 14-0-14VAC transformer secondary winding,

Since the 2 driver stages are identical, only one will be described, that associated with the ϕA signal which originates from the flip-flop U21a on the Control PCB, When this flip-flop is clear, ϕA is high and ϕA/ is low. The former causes Q5 and thus Q1 on the Motor Control PCB to be cut off, while the latter turns on Q9 and thus supplies base current to Q6 and Q2, turning them on too. Therefore the MA signal is driven to the negative supply rail.

When U21a on the Control PCB is set, ϕA is low and ϕA/ is high. The former now provides base current to the PNP transistor Q1 on the Motor Driver PCB which in turn provides base current to Q1, turning it on. Transistor Q9 receives no base current and this provides no base current to Q6 which in turn provides no base current to Q2. Q2 is therefore cut off. The MA signal is thus driven to the positive supply rail.

The motor winding is connected between MA and ground and thus the polarity of the current through it depends on the state of the flip-flop U21a on the Control PCB as required..

Chapter 7

Power Supply Theory of Operation

7.1 Mains Input and Transformer

Mains enters the printer via a IEC60320 C14 plug with integral filter on the rear panel. It then passes via the 1.5A fuse in the live wire to the double-pole on/off switch to the voltage selector switches and the mains transformer primary windings.

The mains transformer has 2 primary windings, each is 120V with a tap at 100V. The voltage selector switches connect them as follows for different mains input voltages :

100V :] The 2 100V sections of the primary windings are connected in parallel

120V : The 2 complete 120V primary windings are connected in parallel.

220V : The 100V section of one winding is connected in series with the 100V section of the other.

240V : Both 120V windings are connected in series.

In all cases, the cooling fan is connected across one of the 120V windings. Thus this fan operates at 120V, using the mains transformer primary windings as an autotransformer to obtain this voltage.

The mains transformer has 5 independent secondary windings which are used as follows :

18V : (Blue wires) used for the -12V logic supply

18V : (Orange wires) Used for the +12V logic supply and the Power-OK circuit

14-0-14V : Yellow wires with brown centre tap) This winding powers the motor driver circuitry as described earlier

10V : (Green wires) Used for the +5V logic power supply.

39V : (Grey wires) Used for the printhead power supply.

7.2 Logic Power Supply

The Logic PSU PCB provides +12V, -12V and +5V regulated supply rails for the printer's control electronics using linear regulator circuits. It also provides the Init/ power-on reset signal.

7.2.1 +12V Supply

One of the 18V secondaries of the mains transformer is bridge-rectified and smoothed, producing approximately 24V across C1. This voltage is regulated down to 12V by the 3-terminal regulator U1.

7.2.2 -12V Supply

This uses an identical circuit to the +12V supply powered from the other 18V secondary winding. However, i nths case, the output terminal of the regulator is connected to ground. Since the action of the regulator IC is to maintain this connection at 12V above the common terminal, this common terminal (and the -ve side of C2, etc) is at -12V with respect to logic ground, as required[1]

7.2.3 +5V Power Supply

The +5V supply, used for almost all the digital electronics in the printer, is based around the 723 IC U3 on the Logic PSU PCB. This IC contains a 7.15V reference voltage source, an error amplifier and a drive transistor.

[1] There is no such thing as ground...

The input to this regulator circuit comes from the 10V secondary winding of the mains transformer. This is rectified and smoothed by chassis-mounted components and produces about 12V DC across the 18000μF capacitor.

The regulator control circuitry is located on the Logic PSU PCB. The inverting input of the error amplifier in U3 is connected to the +5V supply line on the Logic Backplane. The non-inverting input is connected to a divided-down version of the 7.15V reference. Due to the values of the resistors R 1 and R3, the non-inverting input +5VRef is kept at 5V. The action of the error amplifier is then to control the output transistor inside the IC to make these 2 voltages equal. This output transistor controls Q1 (on the Logic PSU PCB, which in turn controls the pass transistor Q1 on the main chassis. The overall action, therefore ,is to maintain the logic supply rail at +5V.

If the current drawn from this regulator becomes too great, the op-amp U4, which monitors the voltage across the 0.1Ω sense resistor R10, activates the current sense circuit inside U3, thus turning off the pass transistor and turning off the +5V power rail.

The 18V secondary winding used to supply the +12V power supply is rectified, clamped by the zener diode CR11 and smoothed by C8. The voltage across this capacitor therefore indicates that the mains input is present. Should the mains fail, then transistor Q5 will turn off, allowing Q4 to turn on. This will short the +5VRef signal to ground and thus cause U3 to reduce the output voltage to zero.

Protection for the logic ICs in the event of a failure in this regulator circuit is provided by the crowbar circuit based around the thyristor (SCR) Q2. If the +5V supply rail rises too high, zener diode CR12 conducts and applies a gate drive current to Q2. Q2 then turns on, shorting the +5V line to ground protecting the logic. This will probably blow the 7A fuse in the regulator circuit.

7.2.4 Init/ Signal

This signal provides a power-on reset to the control state machine to ensure the printer starts in the correct state. Capacitor C9 is charged from the +5V line at power-on via a 39k resistor. When the voltage across it exceeds the input threshold of U5a (and providing the mains input is present), U5a's output goes low, causing the output of U5b to go high. Feedback form this out-

put to U5a's input provides hysteresis to prevent the circuit from oscillating.

The output of U5b is therefore low at power-on and goes high a short time later when C9 is charged. This output is logically ORed with the Initin/ signal from the test connector (which can be used to reset the printer during testing) by U5c and U5d to provide the Init/ signal to the printer logic.

The power supply for U5 comes from the +12V via a simple linear regulator circuit consisting of zener diode CR15 buffered by the emitter follower Q3.

7.3 Printhead Power Supply

The 20V power supply needed for the printhead elements is produced by a switching regulator contained on the Printhead PSU PCB along with the associated chassis-mounted components.

The 39V secondary winding of the mains transformer is bridge-rectified and smoothed, resulting in approximately 52V across the 10000μF chassis-mounted capacitor. This voltage is used to power the step-down switching regulator

The chopper transistor for this regulator is Q2 on the main chassis. When it is turned on, current flows from the 52V supply to the load via the inductor on the Printhead PSU PCB. This inductor causes the load voltage to change relatively slowly as the magnetic field builds up in its core. When the chopper transistor is turned off, the magnetic field in the inductor core collapses and current flows from the inductor via the chassis-mounted flyback diode, thus again supplying power to the load. By varying the on:off ratio of the chopper transistor base drive signal, the output voltage of the power supply can be controlled.

The chopper transistor is controlled by the PHPSUDrv signal. When it is high, transistor Q6 is saturated. This turns on Q2, and in turn Q4, which provides base current to the chopper transistor. When PHPSUDrv is low, then all these transistors are cut off. Should the chopper transistor base current rise too high, the excessive voltage drop across R25 (0.15Ω) will turn on transistor Q5, thus preventing Q2 from turning on and thus disabling the chopper transistor.

The chopper control circuitry therefore controls the high:low

ratio[2] of the PHPSUDrv signal depending on the output voltage. This circuit operates as follows: U4 is wired as a relaxation oscillator which charges and discharges the 6.8nF capacitor on its inverting input. The osc signal is thus a non-linear ramp signal. This is fed to the inverting input of U5, which is used as a comparator. The non-inverting input of U5 comes from the error amplifier U2 which compares the actual printhead voltage (PH+), divided down to produce PHSense with the reference voltage from a 6.4V zener diode.

The output of U5 is high when the ramp voltage is less than the output of U2. If the printhead supply voltage rises too high, the output of U2 is becomes lower so the output of U5 is high for a shorter time. Similarly if the output voltage is too low, the output of U5 is high for a longer time.

The output of U5 is fed into the AND gate produced by inverting the output of the NAND gate U6a using U6b. The other 2 inputs of this NAND gate shut the power supply down if the logic power supplies are not present (that is, if Q7 is turned on or if the +5V lien is totally missing) or if the printhead current is too high (comparator U3's output goes low).

As in the case of the +5V logic supply, the printhead supply is protected by a crowbar circuit. Should the output voltage rise too high, Op-amp U1, again sued as a comparator) will change state. The inverting input of this amplifier is kept at a stable voltage by a zener diode, the non-inverting input is a divided down version of the printhead power supply voltage PH+. When the latter rises too high, the output of U1 goes high, turning on transistor Q1 and providing gate drive to the thyristor Q3. This then shorts the input to the chopper circuit to ground, blowing the 10A fuse and disabling the power supply.

[2]Or mark:space ratio, a term from telegraph signalling instruments.

Chapter 8

Dismantling the Printer

8.1 General Dismantling

To gain access to the internals of the HP9866A for repair, begin by removing the covers. The top cover is held on by 2 screws on the rear of the machine, after removing these the cover lifts off leaving the paper access door attached to the mechanism.

The 5 plug-in PCBs are not exposed. These have colour-coded ejector handles which match the card guides and also give the last digit of the PCB part number using the standard resistor colour code. To the left of the print mechanism are :

Data Path : Brown handles

Control : Red handles

Logic PSU : Orange handles

While on the right side of the print mechanism are :

Printhead PSU : Yellow handles

Motor Driver : Green handles – The motor is connected to an edge connector on the top of this PCB

Each set of PCBs is retained by a hold-down bracket retained by 2 screws. Remove these nest; it is easier to remove the countersunk screw holding the bracket to the side pane before the pan head screw holding it to the mechanism. The 2 hold-downs are different, but they are colour-coded to match the PCBs they fit on, so there should be no confusion when the time comes to reassemble

the printer. After removing the hold-downs, unplug and remove the 5 PCBs. AS mentioned earlier the motor wiring is connected to the Motor Driver PCB, unplug this edge connector before removing the PCB. The Data Path PCB contains the MOS character generator ROM and MOS shift registers and should therefore be stored in an anti-static bag.

Next turn the printer upside-down and remove the base cover which is retained by 6 obvious countersunk screws. Loosen the 2 screws at the front of the foot rails that go onto the plastic tabs of the front cover. Turn the printer the right way up again and loosen the 2 screws at the front of the top flanges of the side plates. Pull the front cover forwards, then loosen the 3 screws that hold the switch bracket to it. Slide out the switch bracket and completely remove the front cover.

The Printhead is removed next. Remove the 2 pan head screws and large washers from the front edge of the mechanism side plates that hold the head against the platen roller. Then remove the 2 countersunk screws fixing it to the chassis plate. Swing the printhead forward and down and unplug the 4 ribbon cables from the connectors on the Printhead Driver PCB. Put the printhead carefully aside.

With the printhead removed, the Printhead Driver PCB comes out next. Unplug the 20V power input edge connector from the right hand side of this PCB and then remove the 7 screws holding the PCB to the chassis. Slide the PCB to the right to unplug it for the edge of the Logic Backplane and lift the PCB out from the underside of the machine.

Before removing the Mechanism, the card guides for the plug-in PCBs must be removed. Although all 20 card guides are identical, the mounting methods for the 4 groups are not. Thy are removed in the following manner :

Front Left : Unscrew the 2 screws on the plate, remove the plate and 3 card guides. Remove the 2 screws holding the fixing bar in place (one to the side plate, one of the mechanism) and remove this bar.

Front Right : Unscrew the 2 screws on the plate, note that the left hand screw is the longer one (it goes through the mounting bar into a tapped bush on the motor mount). Remove the plate and 2 card guides. Remove the mounting bar by unscrewing the countersunk screw holding it to the side plate.

Rear : Each set of rear guides is retained by a bracket fixed by 2 screws. Remove the screws, bracket and guides.

With the printer upside-down, remove the 2 screws holding the paper-out microswitch bracket to the mechanism chassis. Slide the microswitch out and leave it hanging on the wires. Remove the 2 screws on the rear that tap into the right-hand mechanism sideplate followed by the 8 screws on the underside of the chassis. With the printer standing on its rear panel, lift the complete mechanism out.

The Logic Backplane is removed by first removing the foot rails. These are retained by the screws that held the front cover in place and 3 further screws. Under each rail is a m,tale chassis rail. The one of the left side is now free and can easily be removed, that on the right side is retained by a further (small) countersunk screw on the front. Then unplug the chassis wiring and input cable edge connectors from the Logic backplane, undo its 6 fixing screws, and lift it out.

8.2 Removing the Rear Panel

To completely separate the rear panel from the printer chassis is a fairly major job. All the above parts must be removed, and then continue as follows :

Remove the cable clips from the switch bracket to free the wiring and unscrew the 2 screws holding the mains on/off switch to the bracket. Unclip all the grey plastic cable clips under the chassis and feed the mains switch cable through the chassis to the underside. Next desolder all the transformer secondary winding leads. On the underside of the chassis, there are 2 orange and 2 blue wires on the logic backplane edge connector and 2 yellow and a brown wire on the Motor Driver PCB connector. Above the chassis there are 2 grey wires to the input of the printhead supply bridge rectifier and 2 green wires to the input of the +5V supply bridge rectifier. When reassembly, note that the polarity of identically-coloured wires is unimportant. Feed the transformer wires through the chassis to the top side.

Next remove the plastic bumper strips from the sides of the rear panel . They are retained by 3 screws each. Often these strips have decayed with time and become brittle, it is then probably best not to refit them but to simply refit the screws into their tapped holes.

With the bumpers removed,, remove the 4 countersunk screws that retain the rear panel to the side panels and remove the rear panel, freeing the switch cable grommet from the chassis plate..

This gives complete access to all parts mounted on the rear panel. However, it is often only necessary to work on the cooling fan or the connector panel. In this case it is not necessary to desolder the transformer secondary wires. Instead, first remove all the parts described in the previous section (up to and including the Logic Backplane. Remove the bumper strips and the 4 countersunk screws under them. Then stand the printer on the rear panel and lift the side panels / chassis up to separate them. Free the switch grommet and turn the chassis assembly about a vertical axis to allow access to the fan and connector panel.

The connector panel is retained by 4 nuts on studs on the bezel and can easily be removed. The fan can be overhauled in the same was as the fans in the HP9800 calculators; this procedure was described in that series of articles.

8.3 Dismantling the Printhead

The printhead pressure springs (2 per printhead hybrid circuit) can be removed by unscrewing their housings with a 5/16" nut-driver. The hybrid circuits can then be removed by loosening the 2 setscrews that clamp each one to its heatsink, but note that they must be carefully aligned when refitting them – the procedure is in the service manual. Hewlett-Packard specifically warn against dismantling the printhead carrier/heatsink unit, my experience is that it can be dismantled and reassembled but that aligning it is very difficult and it is best not to dismantle this unless necessary

8.4 Dismantling the Mechanism

With the printer mechanism removed for the printer, dismantle it in the following way: First remove the coupling bolt that links the motor shaft to the platen roller using a 5/16" nutdriver. If the mechanism is not seized, it is easy to rotate the roller to allow access to the bolt head. If the mechanism is seized, remove the 4 motor fixing screws and rotate the motor body to get the bolt to a suitable position for removal.

Next, if the motor screws have not been removed, remove them. Then pull the motor off the side of the mechanism. Undo the 3 screws thus exposed and remove the motor mount. When the motor is refitted, fit the coupling bolt loosely, then screw the motor to the motor mount and finally tighten the coupling bolt.

Continue dismantling by removing the screws from the ends of the paper feed roller shafts on the left side of the unit. These retain the belt sprockets which should then be drawn off together with the toothed coupling belt. Unscrew the pivot nuts for the paper feed bearing plates (using a 1/4" nutdriver) and remove their screws from inside the printer mechanism frame. Take off the bearing plates.

On the left side, remove the 6 screws (4 pan head, 2 countersunk). Remove the left hand side plate, the thin spacer plate on the inside of it and the 3 rollers. The platen roller is the plain one with a bearing in the left hand side plate, the front feed roller is divided into 3 sections while the rear feed roller has 4 sections. When refittng the feed rollers, the send with the flats and the axial tapped hole goes to the left side of the machine.

If necessary, remove the similar 6 screws on the right hand side plate and separate the mechanism chassis parts.

The various ball races can be removed from the spindles or chassis parts using conventional methods (a 3-legged puller or suitable drifts in a bench vice) if it is necessary to replace them.

I have never hand to rebuild the rollers in an HP9866A so I can't describe how to do it. Suffice it to say for the moment that it does not look to be straightforward, owing to the fact that the feed rollers are divided into sections.

The Motor can be dismantled for repair (after removing it form the mechanism) by removing the 4 slot-headed through-bolts from the front face. The remove the front cover. Often the rotor comes away with this cover, separate them by tapping the ed of the spindle on a wooden bench top. Recover the crinkle washer and the shims from the rear bearing housing and then remove the rear cover of the motor by carefully tapping it free.

8.5 Initial reassembly

Reassemble the printer mechanism by reversing the dismantling procedure. Refit the rear panel and reconnect the transformer

leads if they were desoldered. Refit the logic backplane and reconnect the 2 edge connectors to it. It is best to leave the mechanism out of the chassis until the initial electrical tests have been performed.

Chapter 9

Troubleshooting

It is difficult to say how far a non-working HP9866A printer needs to be dismantled before attempting to troubleshoot it . For a newly-acquired machine,. the covers, plug-in PCBs, printhead and printhead driver should all be removed, it is probably worth also removing the mechanism and logic backplane and separating the rear panel to clean and oil the cooling fan. For a unit which has been in use and which has recently developed a fault, it may be repair it after dismantling rather less of the machine, but a warning is necessary here :

The printhead power supply can supply enough current at a fairly high voltage (20V) to do extensive damage to the rest of the printer's electronics. The input to this power supply, of over 50V is possibly also dangerous to the repairer. And if, due to a fault ,the printhead enable signals – the BankN lines on the printhead connectors – are held asserted, then the printheads will burn out. Replacement printheads are very difficult to obtain.

For this reason, I strongly recommend removing the Printhead PSU PCB and the printhead when repairing any part of the control logic. After the repairs are completed and the unit seems to work, refit the Printhead PSU and check that none of the BankN signals are held on. On then refit the printhead and test the printer fully. There was a 'Test Fixture' which connected to the test connector on the Printhead PSU PCB and which allowed the output voltage of this PSU to be reduced, thus protecting the printhead if there was any fault in the logic. Although this fixture is mentioned in the official service manual no internal details are given and thus although it was *probably* a (variable) resistor network that connected to the input potential divider on U2, I cannot be certain, ad

my repair procedures will not make use of such a device.

9.1 Power Supply Troubleshooting

With all the plug-in PCBs, the printhead and the Printhead Driver PCB removed, set the rear-panel voltage selector switches appropriately and check there is a good fuse of the correct rating in the mains fuseholder.

If the rear panel or Logic Backplane have been removed, then refit them. Reconnect the 2 edge connectors to the logic backplane.

As in the case of the HP9800 calculators, it is difficult to safely operate the mains-on/off switch when the switch bracket is removed from the front cover and mains is applied, so turn on this switch first and used a switched mains socket (if you have them on your workbench) or a switched cable to control the power.

As in the case of the HP9800 calculators, electrical breakdown on the mains input side is very uncommon. However, it is worth checking taut there earth pin of the mains input connector is connected to the metal chassis of the unit. And if you have a high-voltage insulation tester ('Megger') then check the insulation resistance between the live and neutral pins (strapped together) and the case with the mains switch on the unit turned on.

With the printer standing on one side, connect the mains and switch on. If the cooling fan runs then firstly it is working correctly and secondly (and more importantly) the mains fuse has not blown. In this case check the AC voltages from the transformer secondary windings at the logic backplane edge connector and the DC voltages across each of the 2 large capacitors on the chassis. If either of the latter are missing then suspect the associated bridge rectifier.

If the fan does not start, check any one of the AC voltages. If it is present, then the most likely problem is a fault in the cooling fan itself.

If the mains fuse blows, the n the most likely problem is that one of the bridge rectifiers or smoothing capacitors has shorted. Remove the mechanism (if you have not already done so) for access to these parts. If these all test good, and indeed if the mains fuse continues to blow with the transformer secondary winding leads disconnected from the bridge rectifiers, then unfortunately the mains transformer is suspect.

At this stage it is worth checking the 2 power transistors and the

diode mounted on the main chassis. With the PCBs removed these can be tested at the terminals of the associated edge connectors. Again, if replacement is necessary, the mechanism has to come out.

It is worth pointing out that many of the plug-in PCBs have test connectors on their top edges which carry many of the useful signals needed for the following tests. Consult the schematic diagrams for the connection details

Now refit the mechanism and card guides if they were removed. Refit the Logic PSU PSU and apply power again. Check the 3 voltage outputs form this board on the backplane pins. Again, should the mains fuse fail at switch-on, check the input rectifiers and smoothing capacitors for shorts. The $\pm$12V power supply circuits are very easy to sort out, if there is voltage across the smoothing capacitor then the only likely fault is a defective e 3-terminal regulator. The 5V power supply is a little more complex, Check the 7A fuse on the PCB. If this has failed, then most likely the pass transistor has shorted (or maybe Q1 on the PCB has shorted)(, causing the output to rise too high and tripping the crowbar circuit. If the pass transistor and fuse are OK and there is still no output, check the input voltages to U3. If +5VRef is missing then either U3 has failed or Q4 is turned on, possibly because the AC-present detection circuit is malfunctioning. Of course it;'s possible for a fault in U4 or its associated components to shut the power supply down by indicating an over-current situation where none, in fact, exists. Finally on this PCB check that the Init/ signal is logic high If not, check the 5V Vcc supply to U5, U5 itself and the associated components.

If the +5V power supply is present, the power-on indicator on the switch bracket should be glowing. If not, then the bulb has most probably burnt out. Replace it. Check the paper-out (Load) indicator by grounding the White/Blue wire on the logic backplane edge connector. Again, the most likely fault here is a burnt-out bulb.

Switch off the mains,insert the Printhead PSU, and switch on again. Check the output voltage from this board, either at the edge connector that would connect to the Printhead Driver PCB or across either of the 2 largest electrolytic capacitors (C1, C2) on the PCB. As this is a switch-mode PSU, it is herder to debug then the linear supplies previously dealt with, but not impossible. Begin by checking he voltage on the emitter of the chopper transistor Q2, mounted on the main chassis. If this is missing,, it is likely that

the 10A fuse on the PCB has failed. This, in turn, may be due to problems with the chopper transistor, the inductor on the PCB or the output capacitors C1 and C2, also on the PCB.

If the chopper is getting power, but there is no output from this part of th PSU, check the osc (oscillator) signal on the test connector with an oscilloscope., A non-linear ramp should be present here. Now check TP1 on the PCB with a logic probe. This point should be toggling, if not, then check the inputs to U6a, Either the comparator U5 is not driving the rest of the circuit (in which case check U5 and U2), or one of the other circuits is shutting the power supply down, possibly due to a lack of one of the other voltages. This should not be difficult to trace.

If TP1 is toggling and there is still not output, check the transistors Q2, Q4, Q5 and Q6 that drive the chopper. After getting this power supply working, disconnect the mains input and remove the Printhead PSU PCB again and set it aside until the printer logic is essentially working.

9.2 Getting the Motor to Run

Next (with the Logic PSU still fitted), refit the Data Path, Control and Motor Driver] PCBs. Connect the motor edge connector to the top of the last PCB. As an aside, the only reason for having the Data Path fitted is to make RotEn a logic 0, thus enabling the manual paper feed logic

Apply power again and carefully (so as not to touch the mains connections on the on/off switch) press the feed button. The motor should turn. If not then start by checking the output of U15a on the Control PCB. It should be toggling while the feed button is pressed. If not, then check that Autofeed is not being held low due to a fault in the state machine or the decoder U13. Check that the motor oscillator is running (that the output of U23d is toggling) and the inputs to U22a.

Once MotorClk is correct, check the 4 motor drive signals, ϕA, ϕA/, ϕB and ϕB/ at the backplane connector. Lack of activity here suggests a problem in the motor counter *U21a and U21b) or the associated open-collector inverters.

When these signals are correct, move on to the Motor Driver PCB.First check the (unregulated) power supply voltages on C1 and C2. Then, using an oscilloscope, look at the MA and MB

signals on the motor connector. Problems here are generally caused by defective transistors on this PCB. Or course the problem could be an open-circuit winding in the motor itself, This is easy to detect (using an ohmmeter across the terminals of the motor edge connector when it is unplugged from the Motor Driver PCB), but much harder to correct.

Of course with no paper in the printer, the Load lamp should be on. Operate the paper detection microswitch with a suitable tool and check that the lamp goes out when it is depressed. If not, check the debounce flip-flop U28a and U28b on the Control PCB and the driver transistor Q1.

9.3 Repairing the Control Logic

When working on the control system, the Printhead PSU PCB and the printhead should be removed, but all other parts should be fitted.

First check SMCLk and OutEn/ on the Control PCB test connector. There should be a clock signal of about 850-900kHz on each one. If not, check the master clock circuit (expect about 3.5MHz on the output of U8c) and the clock divider circuit. Without the correct clocks, the state machine can't possibly work properly

The easiest way to locate faults in the control logic is to use a logic analyser to trace the operation of the state machine. The *Inverted* form of the 5 state bits – SMA(0)/...SMA(4)/ are available on the test connector on the Control PCB. Since the state changes on the rising edge of SMClk, the analyser should be clocked on the falling edge of this signal so that it samples stable signals.

Two further comments are necessary. Firstly, if any of the state flip-flops, U2a, U6b, U6a, U18b and U2b are faulty, the inverted signals on the test connector may not reflect the actual signal being sent to the state machine ROM. And secondly, remember that every state transition is conditional, the CC signal cannot be inhibited. Unconditional transitions are simply the result of having the 2 possible next states actually being the same. But if the input gates to the state flip-flops are malfunctioning, the actual next state may (incorrectly) depend on the CC signal. If the state machine seems to go in an impossible sequence, then investigate this area.

9.3.1 Character Input

Connect the printer to a suitable machine, such as an HP9830, which can send individual characters. Use a suitable rod to operate the paper-out microswitch so that it appears that paper is loaded (if necessary remove the microswitch from the mechanism). It us helpful to have the printer standing on one side panel so that the test connectors on top of the PCBs and the Logic Backplane pins are both accessible.

It is useful to have a way to reset the printer electronics. A momentary SPST switch (e.g. a 'push to make' button) connected between InitIn/ and ground on the Logic PSU test connector is a good way to do this.

Monitor IFFlg/ on the Logic Backplane or interface connector and send a printable character (other than a linefeed). If IFFlg/ pulses low then the character input section of the state machine is basically working.

If not, then if logic analyser is available, connect this to the SMA(n) signals on the Control PCB test connector. Reset the printer and if possible set the logic analyser to trigger if the state is no longer 00. If the logic analyser does not support that sort of trigger, then trigger when SMA(1)/ goes to 0.

Send a character as before and see what states the state machine goes through. If the sequence is totally crazy, then suspect a problem with the state machine flip-flops, the input gates or the state machine ROM itself.

If no logic analyser is available, then monitor SMA(1) with the logic probe. While this will not indicate anything like as much, it will at least indicate that he state machine is doing something as the machine leaves state 00.

If nothing happens at all when a character is send, then look at Stb/ at the output of U22b on the Control PCB. If that is not changing state, then check to be sure that both PaperOut/ and Feed/ are high. Debug the debounce SR flip-flops if necessary. Then check CC and CC/ (input and output of U1d) to make sure they're behaving correctly. And then, check the state machine flip-flops, gates and ROM.

If the state machine appears to be running but IFFlg never goes low, then check InClk/ on the Logic Backplane. Reset the printer and send a character. if InClk/ never toggles then check the state machine output decoder U13 on the Control PCB. If InClk does

toggle, then check U22a on the Data Path PCB. If IFFlg goes low and never returns high, then reset and send another character while monitoring ClrFlg/ on the Logic Backplane. If this signal is toggling, then check U21a, U21d and U22a on the Data Path PCB. Again, if not, check the decoder U13 on the Control PCB. Of course if the interface flag flip-flop U22a (on the Data Path PCB) seems to be working correctly but IFFlg/ is not, then suspect the open-collector buffer U20c.

Once the printer is working well enough to accept characters, there are a few other signals that are worth checking. Of course it is now no longer necessary to reset the printer before sending each character. First check that SClk is pulsing low for each character entered. If not, the fault is likely to be U20d or U13 on the Control PCB. then check that the character counter is correctly incrementing after each character – its outputs are available on the test connector on the Control PCB. If not, then it is possible that the characters are not being treated as printable, check the input latch and the character testing gates U10, U16d, U16c and U16b on the Data Path PCB.

9.3.2 Paper Feed

Reset the printer to clear all the counters and then send a single linefeed character. If you are monitoring the state machine with the logic analyser, check that it does the transition form state 1B to state 15.

In any case, the stepper motor should turn the rollers to advance the paper by one character line (if there was paper present). There are 3 possible problems here :

If the motor doesn't run at all (and, incidentally, IFFlg/ remains low) then it is likely that AutoFeed/ is never going low. Check the state machine and its output decoder.

If the motor advances one step and IFFlg/ remains low, then it is likely the state machine is stuck in state 09, waiting for the motor timing monostable to time out. You can check this my monitoring the SMA(n)/ signals with a logic probe. if this is the case, then check U22b the associate gates and the flip-flop U18a on the Control PCB.

The last possibility is that the motor starts running and never stops. The general cause of this is that the condition at state 1F is always true, so the state machine stays in the paper feed

loop. Again, a logic analyser will prove the point. This generally means that the row counter or RotEn on the Data Path PCB is the problem. The counter outputs can be checked on the test connector of this PCB, RotEn is the output of U19a. If the motor is running, then the counter should be incrementing on each step, this is

9.3.3 Print Cycle

Now send a reasonably long line of printable characters (but fewer than 80) and a linefeed. The correct behaviour is for the rollers to move by one character line but rather slower than before (as the printer is processing each dot row). As before there are several easily-spotted failure modes.

If the printer 'hangs up' with IFFlg/ l;low having not moved the paper at all, it is likely that it is stuck in the buffer-fill loop, states 17,04,08. The easiest way to check this is with a logic analyser, if one is not available, continual SClk and PHClk pulses (at a third of the state machine clock frequency) are a symptom of this problem. Check that the character counter is incrementing and find out if Start/ is ever going low. Finally suspect the state machine condition multiplexer U14.

If the paper advances one dot row and the printer hangs then the state machine has entered the print cycle loop but for some reason it is not completing it. Again a logic analyser is the best way of finding out what the state machine is doing, the most likely problems are :

It is stuck in the 'shift by 4' loop at 07,1C, waiting for LnFlg/. This can be detected with a logic probe by a continual stream of pulses on ForceSpace/ and nothing on SClk (The latter distinguishes it for the buffer-fill loop). In this case, check if the output of U20a on the Control PCB is toggling. If not, check this gate, if it is, investigate the 4-state counter U17a and U71b.

It is stuck in the transfer characters loop from 18 to 03. Generally this means that NewBank/ is never being asserted. Check the character counter and the gates that produce this signal.

It is stuck in the complete dot-line loop because Firstbank/ is never being asserted so the condition at state 0C never passes. Check U15c on the Control PCB.

It is hanging in state 0A waiting for the timing monostable U26a (on the Control PCB) to time out. Again, this is easy to detect

wit ha logic probe as the state machine flip-flops will be static in that state. Check the monostable and associated gates.

If the printer starts to feed the paper and never stops, it is likely that Row7 is never being asserted, and this the condition at state 16 never passes. Check that the row counter is incrementing and then check the output of U19b on the Data Path PCB.

Once the print loop is basally functional, send the line of characters and linefeed again, and check PHEn. It should be toggling. If not, check U24 on the Control PCB.

9.4 Locating Data Path Problems

In general problems in the data path are easier to find than those in the control system because, if the latter is working, the printer will accept characters and attempt to print them. Many faults can thus be located by examination of the printed output.

Begin by refitting the Printhead PSU PCB – the only electronic section now not fitted should be the printhead. Power up, and check that PH+ is at about 20V and that all of the BankN are 'dead'. Now, with the paper out microswitch operated as before, send a line of text followed by a linefeed. After this, again check that all of the BankN signals are dead. If not, Find out why not before fitting the printhead. Check the PHEn/ signal. If it is stuck low, check the timing monostable U24 on the Control PCB. If PHEn/ is high, then check the bank select NOR gates, the drivers and the transistor array U8 on the Printhead Driver PCB. Do not fit the printhead before curing such faults, a burnt-out printhead is not pleasant.

Now, power down, refit the printhead power up and load a roll of thermal printer paper[1]. Check that that the 'Load' lamp on the switch bracket has gone out, if not, find out why the paper is not operating the paper out microswitch.

Again print a line of text (ending in a linefeed). This time something should appear on the paper, and there are several possibilities :

The printer prints correctly. There is not much more to do. Just refit the casing parts and enjoy hard copy output on the HP9830.

The entire line is blank. This normally means the printhead

[1]Thermal fax paper works perfectly.

is not being driven. Check PHEn/ and the bank select gates. No data being loaded into the dot shift registers on the Printhead Driver PCB can cause this (it can also cause this, so check PHClk. Of course the printer might be printing space characters in all positions dues to a problem with the input latch or the data shift registers on the Data Path PCB, but this is rare. Also rare, fortunately, is a totally data character generator ROM which can produce either entirely blank or entirely dark characters.

If every fourth character is blank, it suggests that the associated BankN signal is never being driven. Check its driver transistor in U8 on the Printhead Driver PCB and the associated gates.

If the same dot column of every character is either blank or always dark, then the problem is most likely in the dot data path. Check the appropriate PHData(n) signal on the test connector of the Data Path PCB. If it is not changing state during the print cycle, then check the inverter that drives it and the character generator ROM. If PHData(n) is toggling, then check the inverter that receives that bit on the Printhead Driver PCB ,ad the associated dot shift register.

If characters on the right side of the printout are correct, but those on the left side have either missing or dark dot columns then most likely the dot shift register for that bit of PHData(n) is defective. Remember that characters are loaded from the right hand end of the line (so that the first character ends up being shifted all the way to the left), so a fault in the shift register that stops the bits flowing any further will cause this fault.

A single missing dot column is most likely due to a problem with that printhead. Assuming a replacement printhead is unavailable, it may be best to move the defective one to the rightmost position and try to only print short lines.

If some or all characters are printed wrongly but are still valid character patters, then the problem is most likely in character code data path. Check the input register flip-flops, the data storage shift registers and their input gates and the character generator ROM.

If the first dot row of the line is correct, but subsequent rows are wrong (particularly if only wrong for some character codes), then suspect the recirculation gates on the shift registers.

If the first line is correct but subsequent lines have their dot patterns rotated to the left (in 4 character increments) then the problem is most likely in the control system,m. Check the char0 signal and ensure that the state machine correctly skips the rota-

tion loop at the end of each dot line.

Minor defects in the characters such as missing or extra dots are most likely due to the character generator ROM. Since this is an HP custom part (it is a mask programmed ROM of a similar type to the ROMs used in the HP9800 calculators), it may be simplest to live with this problem if it is not too severe.