Repairing the HP9866A printer – Part 2

Tony Duell (ard@p850ug1.demon.co.uk)

Control State Machine Firmware

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.

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.

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 inter-line 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.

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.

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..

Power Supply Theory of Operation

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.

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.

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.

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]

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

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.

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.

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 output 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.

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.

All sections of the printer electronics have now been described. The final article in this series will cover the practical details of dismantling the printer for repair and tracing faults in the electronics.