Repairing HP9800 Desktop Calculators – Part 2

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

Memory System Theory of Operation

1 Introduction

Although the differences between the memory systems in the 3 models of clacualtor mean that the systems will be described separately, there are genearl features common to all of them which will be described first.

The memory system consistrs of the ROMs (both those containing the standard calculator firmware and those in optional extension modules), the RAM, and the memroy cotnrol circuitry. The latter contains a pair of 16 bit shift registers, the memory address (or 'M') register is serially loaded with the current memroy address by the processor. Its parallel output select a particular location in the ROM or RAM. The memory data (or 'T') register provides an interface betwee nthe 16 bit parallel memory data bus and the bit-serial processor. For read operations, the contents of the selected memory location are parallel-loded into the memroy data register and trasnfered serially to the processor. For write operations, the memroy data register is loaded serially by the processor and then its contents are transfered to the addressed RAM location.

The parallel outputs of the M register address a 32K word address space (the MSB, **M(15)**, which correspods to the indireciton bit of a procesosr address, is used to trigger an eaxternal DMA device in the HP9820 and HP9830 machines, and is not regarded as part of the memory address). This single address space contains the ROM and RAM devices.

The CPU microocde word bits 24 and 25, known as the 'S field' (with bit 24 as the MSB) are decoded by the memory control system. This field controls the bit-serial input to the second ALU input (signal **ALUSerIn**) and has the following meanings :

S=0 : Force 0 into the ALU

S=1 : Read the M register into the ALU

S=2 : Read the T register into the ALU

S=3 : Force 1 into the ALU

The ROM devices used are HP-designed 512*8 bit mask-programmed ROMs. They are somwahat unconventional, in that they are internally designed as a pair of 256*8 ROMS[1]. Each secion of the ROM is eanabled by a separate active-low input, these are conventionally linked to gate forms of address line **A8** and its inverse, so that when the ROM is accessed one (and only one) of the sections is eanabled. Another unconventional feature of these early MOS ROMs is that the enable inputs have to be taken to +12V votlage levels, rather than the (5V) TTL-compatible levels on all the other inputs and outputs.

This series of machines was one of the first major commercial applications for the Intel 1103 1K bit RRAM. These PMOS devices run from -16V supplies with a +3V bias supply, and have 16V votlage swings on the logic inputs and outputs. In the HP9800 series, the power supply pins are re-labelled as +16V and +19V (this makes no electrical differenec, you can connect th ecommon lead of your voltmeter anywhere), but of course bufferes are still needed to convert between the TTL level logic signals used elsewhere in the machine and the 16V ones used by the RAMs.

One further comment is that the processor requires ROM starting at locaiton 0 to provide the first instruciton to be executed at power-on, the 'counter constant' at locaiton 1 (this is a special bit pattern, (043060_8) used by the floating point microcode routines. However, the memory area from 1400_8 to 1777_8 must contain RAM to store the floating point data, the ssytem stack point, etc. This is no problem on the HP9810, which simply fils the first 512 words with a pair of ROMs and the next 512 words with a block of RAM but it slightly complicates the address decoder in the other calculators.

Physicaly, the memory system of the HP9810 and HP9820 machines is contained in a memtal boc plugged into the main backplane, Inside the box are the ROM and RAM PCBs and 3 boards of memory control circuity (**Memory Address**, **Memory Data**, and Memory Timing PCBs. The memory ssytem of the HP9830 consists of PCBs directly plugged into the main backplane, these consist of ROM and RAM PCBs, 2 boards (**Memory Addres** and **Memory Data**) contianing the memory control circutiry, and the **Ext ROM Seleector** PCB.

[1]The organisation of the device and the data it contains are determined by the final metalisation layer on the chip. 256*16 bit ROMs using the same underlying device are found in other HP products of the period.

2 The HP9810 Memory System

The memory system of the HP9810 is unconventional in that a 1K*1 bit RAM chip appears as a 512*2 bit device. The 16-bit wide system RAM uses only 8 chips, for example.

2.1 M register and address decoder

The M register is a 16 bit shift register formed by cascading U2, U3, U8 and U1 on the **Memory Address** PCB. To load it from the ALU output, microcode bit μ**(27)** is low¿ This enables U9d, which passes the **Bitclk** signal to the shift register. The mode input is low, so the register shifts right on each **Bitclk** pulse, and the ALU output is shifted into the most significant end.

To read the register, the S field contains 1. This is decoded by U16c on the **Memory Timing** PCB, and the **RotEn** signal is high. Back on the Memory Address PCB, U9c is enabled, and for a simple read, the mode input of the shift register is high. The register thus parallel-loads on each **Bitclk** pulse, but owing to its connections, this results in a rotation to the right.. The LSB of the register is transfered to the ALU via the first gate of the AND-OR-Invert gate U17 on the **Memory Timing** PCB, enabled by the **RotEn** signal.

The 2 operations can occur simultaneously. In this case, the register performs a serial load operation, and the LSB is transfered to the ALU via the AND-OR-Invert gate.

The parallel outputs of the M register are decoded to provide chip-select signals for the ROM and RAM devices. The output of U9a is asserted (high) when the high nybble of the address is X000. This enables the decoder U17 via U11a when the **MemEn** signal is high. This signal, from U11b, is asserted if the R field contains 3 or 6, corresponding to memory read or write operations. Similarly, U10 is enabled via U9b and U11d when the high nybble of the address is X001. These decoders thus divide the lowest 8K words of the processor address space into sixteen 512-word blocks, which is the size of either a pair of ROMs or a set of RAMs.

The ROM selection outputs from these decoders are converted to 12V levels (as required by the ROM chips) using open-collector inverter and transistor circuit such as the one built round U15f. When the chip select line is not asserted, the input to U15f is high, so the output, and the select input to the ROM are low. The transistor is cut off. When the input to U15f goes low, its output is pulled high by the 4k7 resistor, the transistor is turned on, bringing

the ROM select line to +12V, enabling that ROM

In a similar way, the RAM selection outputs are converted to 16V levels. For example, the **RawRAMCS(0)** signal from the decoder is logically ORed with the **Rfsh/** signal by U11c, the output of this gate will be high as will the output of U12b if the **RAMStb** signal from the memory timing PCB is high. The output of U12b is inverted and converted to 16V levels by U18e and a complementary pair of emitter-followers.

The **RAMEn** signal, from the NAND gate U4, is asserted if any of the RAM areas are selected.

The lower 9 outputs of the M register are used to select a location within a memory device. In the case of the ROM, the lowest 8 bits are simply buffered, for example by U7c, these NAND gates being enabled by the **MemEn** signal, delayed by U5a and an RC timing network. Bit 8 is used to enable one or other half of the ROM as described in the introduction, these 2 signals are produced by U13a and U13b, enabled by U14b, which occurs during the later part of a memory cycle (**ROMEn/** asserted) which doesn't address a RAM block.

The RAM addressing is slightly more complicated since, being dynamic RAMs, a refresh address has to be generated too. The top 5 address lines are converted to 16V logic levels by circuits such as the one based round U20f and sent to the RAM board (this circuit, of course, operates in the same manner as the ROM selection buffers). The low nybble of the M register is fed to the **Memory Timing** PCB, where it enters the AND-OR-Invert gates U14 and U15. These act as multiplexers, switching the RAM address buffer inputs between the output of the M register and the output of the refresh address counter, U13. RAM address input 4 (**RAMAddr(4)**) is used to select between the 2 half-width locations storing a word, and will be described below.

2.2 T Register

The 'T', or Memory Data, register is located on the **Memory Data** PCB, and consists of the four 4-bit registers at locations U1, U6, U11, and U14 cascaded in the conventional way. The serial input/output facilities of this register are used to communicate with the bit-serial processor, the parallel signals are used to communicate with the memory device data bus.

To load the T register from the processor, microcode bit μ**(26)) is low. This causes the output of U21a on the memory Timing PCB to go high, enabling U11c and passing the (Bitclk** signal to the T register. The top gate of the AND-OR gate U22 on the **Memory Timing** PCB is enabled, so

the output of the ALU is gated to **TRegSin** and thus loaded into the register.

To read the T register into the processor, the S field contains 2. This causes the output of U16d on the **Memory Timing** PCB to go high, which enables the T register clock via U21a and U11c as before. Now the second gate in U22 is enabled, so the LSB of the T register (**T(0)**) is gated back to the serial input, the register rotates. The second gate in the **ALUSerIn** data selector, U17, is enabled so the LSB of the T register is transfered to the ALU input.

If both operations occur simultaneously, then only the first gate of U22 is enabled so the T register is loaded from the ALU as before, but the second gate of U17 is also enabled, so the LSB of the T register is transfered to the ALU input.

As an aside, the third gate in U17 is used for force a '1' into the ALU input when the S field contains 3 and the fourth gate feeds the serial output of the I/O register on the **I/O Interface** PCB into the ALU when enabled by the I/O control circuitry during input instructions.

The parallel interface between the T register and memory will be described below.

2.3 Memory timing and control

The memory timing circuit is based around the 4-bit counter U5 on the **Memory Timing** PCB. The outputs of this counter are decoded by U10, giving the **State(n)** signals. Normally this counter is held reset by U4c, since both **State(0)** and the output of U12b are low. A memory cycle is stated by the **MemEn** signal going high. This forces to the output of U12c high and that of U12b high. The reset input of the counter thus goes low, and the counter starts counting, the **State(n)** lines going low in sequence. The output of U12b goes low again when the memory accessing microinstruction ends after 12 clock cycles, but the output of U4c remains low until the **State(0)** signal goes low again. The system is now back in the initial state, ready for another memory cycle.

There are 3 types of memory access operation :

2.3.1 ROM Read

In state 4, and all subsequent states, the output of U21d goes high, asserting **ROMEn/** via U23c and bringing **ROMCE** from 12V to 0V. The ROM address lines are enabled as described above. Since this is a read operation, **R(6)** is low, so **TMode** is driven high by U16b, setting the T register for parallel loading.

The T register is loaded by the **TLd** signal, in state 5 (which is essentially a dummy operation) and again in state 11, due to U3b, U11d and U11b. Since **R(3)** is high. **THold** is forced low by U16a and since U2a is never set, **TShift** is also kept low.

On the **Memory Data** PCB, the **RAMEn** signal is low, so the RAM Read buffers are disabled. Thus (considering bit 15), the output of U16C is high, the ROM data passes through U2b into the T register. In the case of bit 14, both U5c and U5d are disabled, their outputs are '1', so U2a passes the ROM data into the T register. The other bits follow the same pattern.

2.3.2 RAM Read

The RAM read cycle has to be considered one state at a time (All ICs are on the **Memory Timing** PCB unless otherwise specified) :

State 2 : U8a is cleared by U3d, thus asserting **RAMStb** and enabling the selected RAM via the gates on the **Memory Address** PCB. U8b is directly set, so the **Precharge** input to the RAM goes high.

State 5 : U8a is set, deasserting **RAMStb**. **TLd** is asserted via U3b U11d and U111b, the RAM data outputs are converted to TTL levels on the Memory Data PCB (e.g by U18b) and transfered to the odd-numbered bits of the T register, the **ROMD** lines all being high since no ROMs are enabled..

State 6 : U8b is cleared by U3c, thus the **Precharge** signal goes low. U9b is toggled via U23f, thus **RAMAddr(4)** goes high, selecting the location containing the other half of the word.

State 8 : U2a is set via U11a, thus asserting **TShift** (which remains asserted for the rest of the cycle).

State 9 : U8a is again set, this is a repeat of State 2 to read the second half of the word.

State 11 : The **TLd** signal is again asserted by U3b, U11d and U11b. The RAM outputs are transfered to the odd-numbered bits of the T register as in state 6, but since **TShift** is asserted, the transfer gates, for example U5c on the **Memory Data** PCB are enabled and the existing contents of the odd bits of the T register are transfered to the even bits.

State 12 : U8a is set, disabling the RAM select signal.

Sate 15 and end of cycle : All flip-flops are returned to their initial states, the state counter is halted.

Thus, at the end of the cycle, The 2 register contains the contents of the first location read (**RAMAddr(4)**=0) in its even-numbered bits and the contents of the second location read (**RAMAddr(4)**=1) in its odd-numbered bits.

2.3.3 RAM Write

The operation of the **RAMStb** and **Precharge** signals is identical to a read cycle. U16a brings **THold** high during states 0-7, since **R(3)** is low.

State 2 : RAM selected and Precharge brought high as before

State 4 : Since **THold** is high and **TShift** is low, the even-numbered bits of the T register are applied to the RAM data inputs via, e.g., U5d and U10d on the **Memory Data** PCB. U9a on the **Memory Timing** PCB is cleared for one cycle, pulsing the **R/W/** signal low and writing this data into the RAM.

State 5 ; U8a is set, removing the RAM select signal

State 6 : U8b is cleared, removing the **Precharge** signal. U9b is set to select the second RAM location to be used (as above)

State 8 : U16a is now disabled, so **THold** goes low. U2a is set via U11a, so **TShift** goes high/

State 9 : Start the second RAM cycle, as in State 2

State 11 : Since **TShift** is high and **THold** low, the odd-numbered bits of the T register are gated to the RAM inputs (e.g. via U5c and U10c on the **Memory Data** PCB). U9a is cleared for one cycle (as in State 4) thus writing the odd-numbered data bits to the RAM.

State 12 : Set U8a to disable the RAM select signal as before

State 15 and end of cycle : All flip-flops are returned to their initial conditions ready for another cycle.

Thus, at the end of the cycle, the even numbered bits of the T register have been stored in the first RAM location (**RAMAddr(4)**=0) and the odd-numbered bits of T in the second RAM location (**RAMAddr(4)**=1), as required for them to be correctly read back by a RAM Read cycle.

2.4 RAM Refreshing

The dynamic RAM ICs used in the HP9810 need to be refreshed and this is performed by circuitry on the **Memory Timing** PCB.

The monostable U1 sets the time between refresh cycles. When it times out, it causes U6a to be set, thus removing the rest signal from U6b. At the end of the next microinstruction, the **RfshSt** signal is asserted via U12d and U4d, so U6b is set. Thus the output of U18a goes high, asserting the **Rfsh** signal and inhibiting the microcode clock via U20d.

When **Rfsh/** is low, the reset input of the state counter is also low, thus this counter free-runs. The memory timing circuitry performs dummy RAM ready cycles. U9a is held set, so the **R/W/** input to the RAM can never go low and writing is thus impossible. U16b is inhibited, so no data is transfered to the T register. RAM address bit **RAMAddr(4)** is provided by U9b as usual.

The AND-OR-Invert gates U14 ad U15 switch the low 4 RAM address lines from the outputs of the M register to the outputs of the 4-bit counter U13. This is incremented at the end of every complete refresh memory cycle by U12a. When all 32 addresses have been accessed, as required by the RAM's specifications. (a total of 16 cycles), **RfshCy** goes low. This triggers U1, deasserting the **Rfsh** signal. The machine then continues running until U1 times out again and the RAM is refreshed once more.

2.5 Memory PCBs

The ROM and RAM PCBs, along with the plug-in ROM modules simply contain the appropriate memory devices and little more needs to be said about them. One minor curiosity is that the user program in the HP9810 is stored as a sequence of 6-bit keycodes, and thus the program storage RAM is logically 6 bits wide. Due to the fact that the physical RAM width is half the logical RAM width, the program RAM is expanded in multiples of 3 chips. I suspect this is the only machine where this is the case.

2.6 Memory system test connector

There is a test connector on the top edge of the memory box backplane in both the HP9810 and HP9820 machines. It carries the parallel outputs from the M and T registers and can be used to check the proper functioning of these registers or to trace machine code programs. The pinout of this connector is the same in the 2 machines.

3 The HP9820 Memory System

The HP9820 uses 16-bit wide memory throughout, both ROM and RAM. This simplifies the design of the T register and memory timing circuits, but causes additional complexity in the address decoder.

3.1 M register and address decoder

The M register itself is located on the **Memory Address** PCB, and is very similar to that in the HP9810. It is a 16 bit shift register formed by cascading U4, U3, U9 and U2. It is serially loaded from the ALU output when the microcode bit μ**(27)** is low, This enables the **SRClk** signal via U10a and U122. The mode inputs of the shift registers are low, so the register shift right and the output of the ALU is loaded into the MSB.

When the S field contains 1, the output of U6b on the **Memory Timing** PCB goes high. This enables U12a on the **Memory Address** PCB. The M register rotates as described in the case of the HP9810. The LSB of the M register is transfered to the ALU via the second gate of the AND-OR-invert gate data selector U25 on the **Memory Timing** PCB, enabled by U6b. As in the HP9810, simultaneous reading and writing are possible, and operate in the same way as in that machine.

The parallel outputs of the M register are decoded and used to enable the various memory devices. The output of U5c on the **Memory Address** PCB goes high when the address is in the first 4K of the address space, and thus decoder U14, enabled by U23d, decodes this part of the space into 512-word blocks. Of course **Memcycle.** the output of U23d is high for a memory cycle when the R field contains either 3 or 6. The output of U5a is high for addresses of the form $00xxxxxxxxxxxx_2$ or $0x000xxxxxxxxx_2$. Since in this part of the address decoder, the output of U5a is ANDed with **M(12)** by U12, the second group of addresses is masked out here, and the decoder U15 decodes the second 4K part of the address space.

3.1.1 ROM addressing

There are 14 ROM blocks in the HP9820 (8 for the system ROMs and 2 in each of the 3 plug-in modules). Thirteen of these blocks are selected by outputs of these decoders, which are buffered to the required 12V levels by inverter and transistor circuits, such as that built round U19b on the **Memory Address**

PCB. The operation of this circuit has been described as part of the HP9810 memory system.

ROM 1 is the exception to this. It appears in the first half of blocks 1 and 14. When either block is selected, the output of U16d goes high. If **M(8)** is also low (signifying the first half of a 512-word block), the output of U10f will also be high. This enables ROM 1 via U1d and U19e.

The lowest address lines are passed through e.g. U1b (enabled whenever a memory cycle is occurring) and fed to the ROM devices. Address line 8 is a little more complicated. Normally, **RawCS(14)** is high, thus the output of U16 is the same as **M(8)**. This is used by U13a and U13b to enable the appropriate half of the ROM when **ROMHAddrEn** is high. This comes from U17b, and occurs during a memory cycle which does not access RAM (and therefore accesses ROM). When **RawCS(14)** is low – when the second half of ROM 1 is accessed, then ROM address input 8 is forced high by U16a, thus selecting the second half of ROM 1 (of course, during accesses to the first half of block 1, **M(8)** is low anyway, thus selecting the first half of ROM 1 here).

The output of U17d (**StatEN**) goes high when the last word of the ROM in block 1 is read. This word is modified if an extension RAM PCB is fitted to the machine by logic on the **Memory Data** PCB.

3.1.2 RAM addressing

The standard 1K words of RAM appears in the second half of block 1, the second half of block 2 and all of block 15. The output of U16d is high whenever blocks 1 or 14 are accessed, and thus the output of U18c is high whenever the second half of either of these blocks is accessed. The output of U17a is clearly high when block 15 is accessed. These are logically ORed by U17c, the output of which goes low to enable the standard RAM.

The expansion RAM (also 1K words in size) is enabled for addresses of the form $01000xxxxxxxxx_2$ by U12b and U23c. When either RAM area is accessed, the output of U16c goes high, enabling the RAM output buffers on the **Memory Data** PCB and disabling ROM access via U17b.

The outputs of the RAM decoders are logically ORed with the **Rfsh** signal (so that all RAM is enabled on a refresh cycle) by U11c and U11d, and then ANDed with the memory timing signal **RAMCSEn** by U18d and U18a. They are then converted to 16V levels and fed to the RAM chips.

The address lines to the RAM chips Fit into 3 groups. **M(0)...M(4)** are fed to AND-OR-Invert gates on the **Memory Timing** PCB (e.g U22a) which elect

between the M register outputs and the refresh address counter. The outputs of those are buffered to 16V levels (e.g. by U17a) and fed to the RAMs, as described in the HP9810 memory system.

M(5)...M(7) are simply converted to 16V levels on the **Memory Address** PCB (e.g.by the circuit around U21e) and fed to the RAMs.

The highest 2 RAM address line circuitry is also on the **Memory Address** PCB but is made more complicated by the fact that the standard RAM is split between 3 blocks. When block 1 is accessed, U16b and U11b are inhibited, so the outputs of both are high. When the second half of block 14 is accessed, the output of U16b goes low, that of U11b is high (by virtue of the fact that **M(9)** is low in block 14). And throughout block 15, the output of U16b is the inverse of **M(8)** while U11b's output goes low (since **M(9)** is high as is **RawCS(1)**). Thus a different 256 words of RAM are accessed in each case. When the expansion RAM is accessed, the outputs of U16b and U11b are simply the inverses of **M(8)** and **M(9)**.

3.2 T Register

The T register is a 16 bit shift register comprising U2, U5, U3 and U4 on the **Memory Data** PCB. As in all HP9800 machines, it communicates serially with the processor while the parallel inputs and outputs are used for the parallel memory data bus.

To load the T register from the processor, μ**(26)** is low, This enables the **Bitclk** signal to the R register via U7c, U11b and U13a on the **Memory Timing** PCB. The ALU output is passed to the MSB of the T register via the third gate in the AND-OR data selector U18 on the **Memory Timing** PCB, so data is transfered from the ALU to the T register.

To read the T register into the ALU, the S field contains 2. This causes the output of U9d on the **Memory Timing** PCB to go high, enabling the T register clock as before. The second gate in U18 is enabled, so the T register recirculates (the LSB output is fed back to the serial input). The fourth gate in the ALU data selector U25 is enabled so the serial output of the T register is read into the ALU.

If both operations are requested at the same time, the T register is read into the ALU as normal, but the third (not the second) gate of U18 is enabled, so the T register is loaded from the ALU output.

The remaining gates in U18 allow the T register to be loaded from an external DAM device (fourth gate) or to recirculate when being read by a DMA device

(first gate). Little more will be said about DMA devices here, since I have never seen one and thus have no precise details on how they operate.

There are 2 remaining gates in U25. The first gate is used to force a '1' into the ALU when the S field contains 3. The third gate transfers the I/O register contents to the ALU under the command of the input/output system. These operate identically to their counterparts in the HP9810.

The parallel outputs of the T register are converted to the RAM's 16V levels, e.g. by the circuit containing U8c, and fed to the data inputs of the RAMs.

The data outputs from the RAMs are converted to TTL levels by the comparator circuits such as U14a, which are enabled when the RAM is accessed. The outputs are then inverted and logically ORed with the ROM data outputs, for example by U16, and fed into the parallel inputs of the T register. Since ROM and RAM are never both enabled at the same time, this simply passes the data from an enabled memory device, unchanged, to the T register.

There is one exception to this. **TIn(10)** will be forced high by U10 when the last word of the ROM in block 1 is read (that is, when **StatEn** is asserted) and if an expansion RAM board is fitted (when U11a's output is high since its input is connected to ground by track on this extension RAM board). Thus the machine can tell if the board is present.

The timing of the parallel load will be described in the next section.

3.3 Memory timing and control

The memory timing circuit is again based around a 4 bit counter, U1 on the **Memory Timing** PCB, the outputs of which are decoded by U5. Normally the counter is held reset by U6a, but when a memory cycle occurs, the output of U7d goes high, releasing the reset condition. The counter thus counts until **State(0)** goes low at the end of the cycle. The **ROMEn/** signal is asserted by U11d and U21f when the state counter gets to state 4, and remains asserted for the rest of the cycle.

There is essentially only one type of memory cycle in the HP9820, which occurs as follows :

State 1 : U16b is set, removing the reset signal from U15a and allowing the **Precharge** signal to be asserted later.

State 2 : Start of the dummy RAM cycle, used only during refresh. U15b is cleared, asserting **RAMCSEn**, setting U15a via U9b and thus asserting **Precharge** to the RAM

State 5 : Set U15b, deasserting **RAMCSEn** and removing the set signal from U15a. End of the first RAM cycle.

State 6 : Clear U15a, deasserting **Precharge.** Toggle U4a. This is a remnant of the half-width RAM addressing of the HP9810, and is used here only to supply a bit of the refresh address and thus refresh two RAM locations in a single refresh cycle.

State 9 : Clear U15b, starting the second RAM cycle as in state 2.

State 11 : On RAM write cycle, U16a is toggled low for 1 clock cycle (the output of U22a is high), so **RAMWr/** is asserted, writing the data to RAM. For read cycles, **TMode** is high (since U12d's output is high), and thus **TLd** pulses high in this state. This parallel-loads the T register with the contents of the selected ROM or RAM location.

State 12 : End of the second RAM cycle, as state 5.

State 15 and end of cycle : All flip-flops are returned to their initial states, ready for another cycle, the counter is halted.

Thus at the end of the cycle, the appropriate transfer between the T register and memory has occurred.

3.4 RAM Refresh

The Dynamic RAM refresh circuit is located on the **Memory Timing** PCB. A discrete transistor astable multivibrator releases normally holds U4b reset. When a refresh cycle is to occur, this reset is released, and at the end of the next microinstruction, U14b is set, asserting the **Rfsh** signal and inhibiting the microcode clock via U19c. When **Rfsh** is asserted, the reset input of U1 is held low via U7d and U6a, so the memory state counter runs. As described above, 2 memory cycles are performed for each complete cycle of this counter. **RAMWr** is inhibited since U16a is held in the set state and T register loads are inhibits via U12d and U8a. Thus no actual data transfers occur in these cycles.

The AND-OR-invert data selectors such as U23 switch the low 5 RAM address lines from the outputs of the M register to the outputs of the refresh address counter on the **Memory Timing** PCB. The LSB of the refresh address is provided by U4a, which as described above is toggled during the state counter cycle. The remaing 4 bits are provided by a 4 bit counter, U20, which is

incremented at the end of each state counter cycle. Thus all 32 refresh addresses are generated in sequence. When all locations have been refreshed, U14a is set via U21a, inhibiting U9a, deasserting **Rfsh** and enabling the microcode clock/ U14a is cleared when U14b is cleared by the multivibrator, ready for another refresh cycle.

3.5 DMA flip-flop

Since I do not own a DMA peripheral, I am unable to describe DMA cycles in detail. However, since this flip-flop disables the system microcode clock, it will be described.

The DMA control flip-flop is U4b on the **Memory Timing** PCB. When a memory cycle is not occuring (**Memcycle** is low), it is held reset. Therefore, when a memory cycle begins, both inputs of U13c are high and thus **DisμClk(2)** goes low, This would inhibit **μClk** via U12b on the **Clock** PCB.

However, if no DMA operations are occuring, **EMB/** is high. Therefore, when U4a is set during the memory cycle, the output of U9c goes high amd U4b is set. This re-enables the microcode clock before the start of the next micoinstrucion, in fact this flip-flop has no real effect on the machine. But during DMA operations, will rempain clear, preventing further microinstructions from being executed untl the DMA operation has completed.

3.6 Memory PCBs

As in the HP9810, the HP9820 memory PCBs contain the appropriate memory devices only. The standard and expansion RAM PCBs are identical and simply contain sixteen 1103 DRAM devices. The 2 ROM PCBs are electrically identical both to each other and to the HP9810 ROM PCB. The boards differ, of course, in the programming for the ROM devices fitted, and also in the part numbers, both for the complete PCB and the ROMs themselves that are etched on the board. The optional ROM modules again are electrically identical to those in the HP9810, but of course with different ROMs.

3.7 Memory system test connector

The test connector on the top edge of the memory box is identical in connections and function to that on the HP9810 memory box.

Sorry, HP9830 owners, that there was little for you in this article. Your memory ssystem will be described next time.