of this divider are used by the derived timing for register clocking pulses. 760.32kHz is also the rate at which the characters are loaded into the output register IC10 on the Memory A board.

From this point, we go into a divide-by-48, made up of IC3 and parts of IC2 and IC4. This divider generates the 48 possible character positions across the line, of which 32 are actually used and 16 are reserved.for retrace and overscan. The output of this divider is our horizontal rate, or 15,840 Hz. Note that this is negligibly faster than the usual interlaced 15,750 Hz.

A divide-by-12 in IC4 and IC5 counts horizontal lines for us, directly giving us the "what line is it?" commands for the character generator. It also gives us inputs for derived timing involving the line 1 transfer.

The output at this point is identified as an "0" clock on the internal test points on the timing board is 1320 Hz. Finally, IC6 and IC5 do a divide by 22 to give us the 22 possible character rows on the screen. Feedback via IC10 of the derived timing ("T" Clock) shortens what would normally be a divide-by 12 to a divide by 11. We use 16 of the 22 lines for characters and save 7 for retrace and overscan. Our final output frequency is the 60-Hz vertical rate.

All of the possible main timing chain signals appear on internal test points A through U, with U being the slowest and A being the fastest. The compliment of G, or G is also brought to a test point since it is useful in the derived timing.

Derived timing

The derived timing is on the same board as the main timing chain. It combines the continuous main timing chain waveforms into suitable "by bursts" signals needed for TV typewriter sequencing and control. Half of IC7 AND,s (Negative Logic) the N and S clocks to give us a 9-12 BLANK signal, an output that is high on each line 9-12, or counting from the top of the tv picture, on horizontal lines 9-12, 21-24, 33-36, 45-48, and so on. The waveform is used to generate the vertical space between characters as well as blanking the characters for the vertical retrace and overscan time. It works by inhibiting (stopping) clock pulses (A Clock) from marching video out of the output register IC10 on the page A memory board.

The same IC also ANDs (negative logic) the K, L, M, and N clocks to give us an output that is low only on each horizontal line 1. (Lines 1, 13, 25, 37. . . . . etc. .) The output is used directly as a line-1 transfer command that connects the line register to the memory only during lines 1, 13, etc.. so a new line of characters can be transferred from the memory to the line register. It is also used by IC10 to allow clocking of the memory only on lines chosen.

IC8 generates our line clock by negative logic ANDing the J and D clocks. This gives us 32 clock pulses per line, used to march the characters through the line register. IC9 provides a suitable time delay *after* each register clocking and then provides an output load command to the output register. The time delay is essential, for after you clock the line register, its output takes a brief amount of time to change. This changes the input on the character generator, which also takes a while to get its output correct. Only after we have the right output do we want to transfer the valid new character information into the output register.

Clocks for the main memory are called $\phi 1$ and $\phi 2$. We get these from IC10, which suitable combines some high frequency clocks (B and C) with the Line-1 output, the 32 pulses per line logic, and some other signals. The net result is a pair of 32 pulses per line, present only on lines 1, 13, 25,. . . and of the proper width to drive the clock driver circuitry on the memory boards. Note the clock levels are only TTL here; the transistors and inverters on the memory boards convert them to the full swing MOS levels at very low impedance to drive the capacitance of the memory clock lines.

The final third of IC10 does the shortening of the vertical interval for us, converting IC6 from a divide by 12 to a divide by 11.

IC11 and most of IC12 generate our horizontal and vertical sync pulses and combine them into a composite sync signal. The horizontal pulses are around 5.2 microseconds wide and happen once each horizontal line. The vertical pulses are around 1.5 milliseconds long and happen once each vertical frame. The position of the pulses is adjustable by changing the jumpers shown on the memory board, giving you four possible horizontal positions and three possible vertical positions for a total of 12 potential locations on the TV screen. In cases where a TV badly overscans or when you have a color set or something else you don't want to make any internal position adjustments on, a simple changing of these jumpers will center the picture for you. These could be made continuously adjustable, but the extra complexity of four monostables and two controls didn't seem worth the benefits.

About interlace

Interlace is not normally used, nor is it desirable on a stationary, words-only presentation. You can pick it up if you have to by using the INTERLACE RESET input, which when grounded, resets the entire vertical counting chain to its *maximum* count. When the reset is released, the next whole line of horizontal timing restarts the new frame. To synchronize the internal horizontal with an external system such as a video recorder, you can either lower the crystal slightly to get exactly 15,750 and use this as system timing, or else you can replace the crystal with a capacitor and voltage control IC1 by applying a +3 to +5 volt control signal onto pin 2 of the oscillator. A very simple phase lock loop system then can lock the typewriter terminal to the video recorder or whatever you may be interfacing.

The important point is that you don't need or want interlace for the majority of applications, and the only time you have to have it is when you must *superimpose* your message on top of some existing program material not under your control.

The cursor circuitry

The cursor board decides where and when a new character is to be entered. It also conditions the keyboard inputs, and optionally lets us use a line scan instead of a frame scan, and optionally controls the winking cursor.

It's easiest to look at this board in two parts. The input conditioning and sequencing is shown in Fig. 8, while the actual character position counter is shown in Fig. 9. Both circuits are on the same board and internally connected via tests points A through D.

Since virtually any keyboard or encoder could be used with the TV typewriter, a relatively elaborate conditioning circuit is provided. The input conditioning eliminates contact bounce. It also *waits* after a contact is made for the encoder in the keyboard to catch up and put out valid data. After that, it delivers an update command that lasts *exactly* one frame.

Sometime during the next frame, the character position counter decides where the new character gets put. If it gets entered at all depends on whether the pages are protected or not, and whether there is a CTRL or carriage return command present.

Keypressed signals consist of the KP input (22) going to ground. This input is filtered by R21, C16 to eliminate the worst of the keybounce and noise, particularly any noise on *break* or key release when you are using an elastomeric keyboard or something else without a snap action. Q1 unloads the filter and drives a Schmitt trigger in IC8 that gives us a clean, snap action, and adds noise immunity to the input.

The output of the Schmitt trigger trips a monostable IC9 that gives us around 10 milliseconds of delay. This makes sure the keyboard code is valid and everything is settled before we try entering any data. The output of the delay monostable is converted to a short pulse by C12.

The output pulse from IC8 trips a one-and-only-one synchronizer IC7. This consists of a set-reset flip flop driving a synchronous type D flip flop, and provides an output that lasts for one whole vertical interval, and only one whole vertical interval.

The one frame output goes directly to the update gate back-forth direction control in IC1 and is inverted to handle the update and cursor gates in IC5. How this is done will become more obvious when we talk about the character position counter later. Finally, the one-and-only-one output goes through a CTRL detector in IC8 that decides if a line feed, carriage return or CTRL command is being received. If it is, a CTRL output is generated that prevents the character from being entered. At the same time, a "move up a