line" or a "move down a line" command is delivered to IC5.

We can also optionally "force feed" the one-and-only-one in IC7 with the LINESCAN input, which updates and moves us one character per frame. This option is handy for clocks and calculators but is not used for normal typewriter use. Also, we have set the circuit up so that *any* CTRL command gives us a carriage feed in order to save parts. If you want to you can add extra decoding and logic to independently bring out as many machine commands as you want to.

A keypressed command is random with respect to the frame by frame system timing. So, something between a very small amount of time and an almost full frame has to go by before the one-and-only-one can start with the next frame. The set-reset flip flop in IC7 absorbs this time difference. Up to the entire next frame may be needed for character entry, depending on where the character is. Thus, it takes *two* whole frames worst case to enter a character via the keypressed input. One to synchronize and one to actually enter. This gives us a 33 millisecond fastest possible update rate or about 30 characters per second. The normal computer teletypes run about 10 characters per second maximum; thus the TV typewriter can easily handle their data rates.

Notice that, in interests of economy, the character information lines A1-A7 are unconditioned. This means that the selected character must be valid when the keypressed delay in IC9 ends and must *stay* valid for at least 33 milliseconds after that. For the vast majority of manually operated keyboards, this is no problem at all. For some special or faster systems, you might like to add latches to the input to store the valid data for the length of time it is needed. While you can, in theory update all 512 characters in a single frame, this takes a bunch of more complicated circuitry, and if you can, run your system at less than 30 characters a second (CPS).

This rate will take you about 17 seconds to fill the screen at 30 CPS and around 51 seconds at the 10 CPS typewriter rate. If you have two pages, you could fill one while using the other by changing the update and protect jumpers around.

Character position counter

The character position counter circuit is shown in Fig. 9. Many cursors use an add-subtract or up-down counter that's static and a big comparator to find out when the next character is to go in its proper place. While this works, it's big, expensive, and takes a lot of fancy parts. It's also a bear to debug. We use a much simpler system here that gets the same job done without the need for up-down counters or comparators. It's called a *phase shift counter.* All we do is have a divide-by-512 counter that goes around just like the system timing does, for it is driven by the 512 clock pulses that run the memory. The counter runs *continuously* in bursts just like the memory does. Once each frame, the output suddenly drops, indicating that this is the place to put a new character and that the cursor should also be shown at this time. If we don't tamper with the inputs, the counter always drops on the same place in each frame.

Now, the trick is to back the counter up or move it forward *with respect to the system clock pulses.* Add an extra pulse one frame, and the output drops one count *earlier,* backing us up one character. Hold back one pulse per 512 and the counter goes ahead one character, *changing its relative phase or character position with respect to the system timing.* To make things a little bit easier, we either throw in a short extra count to back up, or a very long extra count that's so long it overrides *two* system clock pulses to go forward. One extra minus two held back is the same as holding one back and a lot easier to do.

So much for the normal character-to-character operation. To get a carriage return, you break the divide-by-512 counter into a divide-by-32 and a divide-by-16, the former for characters across a line and the latter for lines. For carriage return, you reset the character counter to its highest count and hold back or add one line count pulse from the line counter. This returns you to the left and up or down a line at the beginning of the next frame. We either add a brief pulse to move us up a line or a long pulse that overlaps two normal ones to move us down a line.

Finally, to home or get to the upper lefthand corner, we reset both counters to their highest state, and release the counters immediately at the beginning of a frame. This starts everything off on the right foot for a new sequence.

The output of the character position counter drops immediately before the desired character position. It loads a winking cursor command into the proper slot on line 1 of this particular character group. And, if we are in an update cycle, and if we are on an unprotected page, it allows entry of the character by switching the memory from recirculate to update just this one character position.

Turning to the actual circuitry of Fig. 9., the input clock is ANDed with an add character command in IC1. IC2 and half of IC3 form the character counter, while IC4 counts character lines. IC4's output is controlled and distributed by the cursor and memory updates by IC5, under control of the update command from the one-and-only-one and the cursor off-on switch. At the *end* of an update, test point C suddenly drops pulses one of the AND gates in IC1. If pins 27 and 30 are open, this pulse is so brief that it gets added to the normal $\phi 1$ clock pulses and we get an *extra* count pulse, backing us up one character. Short pins 27 and 30, and the pulse is so long that it starts before the first normal clock pulse and lasts till *after* the second normal one goes away. Here, we add one pulse but block two, leaving us in the hole by one pulse. The character counter moves forward one character.

Linefeed is controlled by the flip flop in IC5. To go up a line, the flip flop produces a very brief pulse. To go down a line, the flip flop is set and held exactly long enough to block two normal count pulses. Again, we add one and block two, moving us down a line. Meanwhile the character counter is reset to its maximum count, so that at the beginning of the next frame, we start at the lefthand side.

The final flip flop in IC3 is set on a clear command and released on the beginning of the next field. This holds the counters in the upperleft position until the system clear is released and a new frame begins. The counters are always loaded or cleared to their *maximum* count. This way, the first system clock pulse at the beginning of the frame sets us to zero, instead of one, making sure everything ends up where it belongs.

Construction

Because of the complexity of this project the construction MUST be done in progressive stages and should not be started until the complete story is on hand and thoroughly understood. If one step seems to present problems, DO NOT GO BEYOND ANY OPERATION THAT DOES NOT SEEM CORRECT. STOP AND FIND OUT WHY! Here, very briefly is the suggested building and debugging sequence.

1. Mount the stack connectors. First, very carefully inspect the PC boards for any possible problems. Minute shorts will be extremely hard to find later. Note the connectors are more or less alternated so that the stack fits together one and only one way. Be sure everything on each board follows the same pattern. Be sure the notch on each board goes the same way. After the stack neatly snaps together, add all the jumpers and all the bypass capacitors on all the boards as well as the protection diodes on the memory boards.

2. Build the power supply. If the transformer has two secondaries, be sure to connect them aiding rather than bucking. The 5-V supply uses the 6-V transformer outputs, while the + and -12-V supplies use the 12-V taps. Be sure to watch the proper polarity on everything particularly the Zeners. Use a proper heatsink for IC1, All the switches can be mounted, being very careful not to short anything underneath the switches with the switch pads when they are bent and soldered in place. Check out the supply, looking for +5 on pins 58 and 59, -5 on 57 -12 on 56 and +12 at the optional keyboard power point. Mount the binding posts for ground and self-test. Do not proceed till all the voltages are correct.

3. Build and check the rf modulator. Wind the coil first, 6 turns on a 3/8" mandril, spaced out to 1" long. Vertical mounting leads are then attached, making sure the tap is at precisely one turn. The tab on Q1 is between the emitter and extra case connection. Eight inches of twinlead are attached to the output, and a new piece of twinlead is taped or plastic bolted to that with a 2" overlap, forming as output attentuator and ending up with a suitable connector. To test the rf modulator, temporarily short the video output pin 20. This tells the modulator to put out *maximum* signal. Apply the output to a suitable TV, preferably a high-quality, small-screen