black and white receiver. When power is applied, you should be able to tune the trimmer capacitor to channels 2,3,4, or 5. You should get a completely blank screen and complete audio quieting with proper tuning. There is no sync yet. Do not go beyond this point till you are sure the modulator and oscillator are working properly.

**4. Build the timing board and the video output.** Add all parts to the timing board, picking three position jumpers at random. Add only IC10 and IC11 and their related parts to the memory A board. Apply power and check for any shorts on +5V. If all is well, you should have 16 rows of 32 white boxes on the screen, and the position jumpers should move you into 3 possible vertical and 4 possible horizontal locations. Pick the best one for the TV you are using. If you have any problems, check first to make sure the main timing chain is ending up with 60 Hz (not nothing or 54 Hz!). Then look for a composite sync output on pin 38, followed by composite video on pin 20. If you have a raster that is blank, look for output load 45, video clock 47, or IC10 problems. The boxes come about since you are loading open-circuit "1"'s onto IC10's inputs. Don't forget to remove the pin 20 ground from step 3!

This is the first plateau of the construction. Beyond this point, things are more or less self-checking. At this time it is a good idea to go through and check *every* terminal in the system with the self-check input, carefully noting everything, learning as much as you can about the timing, and looking for potential problems. (The boxes may be erased by temporarily shorting two output registers to ground. If you can't get to this point, an oscilloscope with a triggered sweep is almost essential for servicing. Beyond this point, the circuit more or less services itself.

DO NOT GO BEYOND THIS POINT UNLESS YOU HAVE 512 WHITE BOXES STABLY AND CLEARLY DISPLAYED WHERE YOU HAVE SELECTED ON THE TV SCREEN.

**5. Add the character generator.** First check the place where IC9 is to go for -12V on pin 1, +5V on pin 24, and -5V on pin 12. Add the pullup and pulldown resistors and solder IC9 in place. As with all the MOS IC's in this project, leave them in their protective foil or foam and quickly solder them in place with a small soldering *iron.* Always be sure all related circuitry is in place *before* adding any MOS IC. Very briefly apply power. You should get a screen full of @'s. IF YOU DON'T GET A SCREEN FULL OF @'s STOP IMMEDIATELY AND FIND OUT WHAT IS WRONG'

Once you have the screen full of at's, make up a jumper consisting of a 330-ohm resistor connected to +5 and briefly connect this one at a time to the signal end of R45 through R50. The character should change from "at" to A to D to H to P to blank and back again. If you get these characters, you'll probably have the rest of them, as we'll find out in the next step. Again, do not go beyond this point until you can display at your command "at", A's, D's, H's, P's, and blanks. You might like to use your self-test again with a blank screen to look at waveforms. Many TV's are reluctant to present an all-white screen with only tiny portions black, so positive logic signals (those that spend most of their time at ground) will generally look bad. These are in the minority. If you want to see them, get a good scope or a better TV. If you have any misgivings about anything on the project at this point, STOP and find out what the problems are. The sharpness of the characters will depend on the accuracy of the tuning and the quality of the TV, but with any reasonable TV, you should get a reasonable size display of good sharpness. Ghost images may mean tuning inaccuracy or transmission line reflections. Run your hand up and down the lead-in to see if this is a problem. Add a couple of 300-ohm resistors if this is a problem.

**6. Add the Line Register.** Check with the self-test for a line one transfer output (one white line–eleven black ones, repeated 16 times), and a line clock (32 narrow vertical white stripes). Add the remaining Page-A memory parts, ending up with IC8, following the usual MOS precautions. Once again, you should get a screen full of "ats", and running up and down the bus B1 through B6 (pins 9-14) should generate the now familiar at A-D-H-P-blank sequence. The only difference now is that we are loading and storing characters for eight lines at a time instead of just displaying them.

The load and store cyclic sequence can be checked in the following way. Temporarily jumper the timing board R clock to bus B1. This should give you half at's and half blanks on the screen. The Q clock should give you two stripes of at's, four rows per stripe. The P clock should give you fonr stripes of at's, alternating two rows per stripe. And the 0 clock should alternate rows of at's and blanks, one row each. Now for the key "is it storing?" test. The J, K, L, M, and N clocks should NOT change the pattern from one of all "at's", but the I clock should turn the right half of the screen blank, giving you a vertical stripe. Similarly, the H clock should give you two vertical stripes, the G four, the F eight, the E sixteen. D, C, B, and A clocks should do nothing. If these tests are OK, the line register is probably working properly. If not, double check for line clock, line-one transfer, proper connections, etc...

Now, we can have some fun. Reconnect the B6 line to the 0 clock. Connect B5 to the I clock, B4 to the H clock, B3 to the G clock, B2 to the F clock and B1 to the E clock. You should get the entire alphabet, numbers, and punctuation spelled out, repeating every two lines.

DO NOT GO BEYOND THIS POINT UNLESS YOU ARE COMPLETELY AND STABLY DISPLAYING THE ENTIRE ALPHABET, REPEATED EIGHT TIMES.

**7. Debug the Cursor.** Add all components to the cursor board. Set the TV typewriter up for a screen full of at's and add the board. The display should not change, except for the possible appearance of the cursor. Check first the Cursor on-off switch and then the Clear to home the cursor. Now switch to repeat and check out the back forth and up-down cursor motion. Pins 2 and 3 on the connector must be grounded for up-down operation.

Next, go to a keyboard or a set of switches that has at least a working spacebar and carriage return key, and check out the cursor for one-at-a-time operation. If you have any troubles, use the internal test points, starting with H. H is a squared and delayed keypressed input. G is a brief pulse 10 milliseconds later. C is a gate that lasts one frame on keypressed command. B is its inverse, and A is identical to B but is present only when a CTRL is present, e.g. pins 2 and 3 on the connector or input bits A6 and A7 are grounded.

Finally, F shows the input to the character position counter, while E is the mid point in the counter. At F, you should have 512 pulses per normal frame. On a frame where an update is to take place or the cursor is to be moved, you should get an *additional* pulse in the upper left. A very brief one for left or upwards motion, and a longer one that overlaps exactly two normal clock pulses for right or downwards motion.

DO NOT GO ON TILL THE CURSOR IS WORKING TO YOUR SATISFACTION.

**8. Add the Memory Clock Driver.** Add all the remaining Memory A board parts *except* IC1 through IC6. Check at the two clock test terminals for two groups of 512 dots, using the self-test or preferably a good scope. If you have a scope, look at these points for a normally +5V waveform that drops to -12V for 200 ns sharply and repeating 512 times per frame. A short on the output of IC7. however brief, can damage the device. This is why R10 and R12 are needed.

DO NOT CONTINUE UNLESS YOU ARE REASONABLY CERTAIN THE MEMORY CLOCKS ARE PRESENT AND WORKING AT THE RIGHT AMPLITUDE!

**9. Add one memory.** Now add your keyboard or test switch group and IC5. As with the other MOS chips, leave these in their protective foam until immediately before use, then quickly solder them in place with a small soldering iron. When the protect and page select switches are properly set, you should be able to selectively store at's and P's wherever you want to on the screen. Note that holding the CLEAR down forces a load command onto the memory. This is useful for test, although you have to derive the encoder +5V temporarily from sonething besides the NOT CLEAR source.

**10. Add another memory.** IC6 this time. Now, you should get at's, zeros, blanks and P's wherever you want, and when the keyboard is properly set up, you should be able to clear the screen on command, again with the protect and page select switches properly set. Note that the keyboard encoder must put out either 100000 or a 000000 code during clearing. This easiest done be deriving the keyboard power from the unclear portion of the clear or home switch. DO NOT GO ON UNTIL YOU CAN ERASE