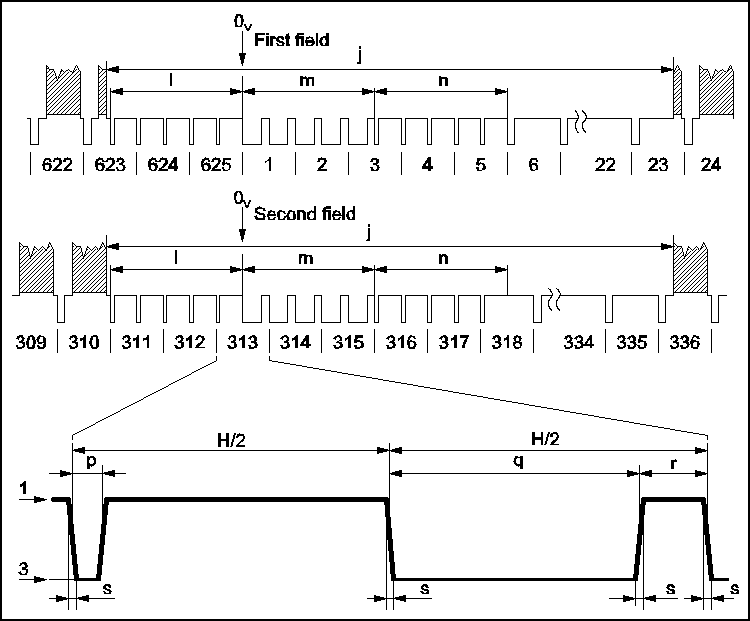
**Release: Video Board Schematic**

Pass me a beer.

I’m quite impressed that as a teenager I had any idea how PAL video worked at all. I’m now in my 50s and I still haven’t really got it.

I’ve only schematised the main digital part of the video board (because the rest of it is a bit hopeless is the real reason), and this isn’t so bad.

The basics : IC10A and IC12A decode write to Video RAM (this shadows the lowest 2k of ROM from 0000-07FF).

IC1 and IC2 count horizontally, using IC2/Q3 to blank horizontally. IC3 and 4 do the same vertically, latching IC5A to set to blank vertically. The character positions (e.g. divide by 8 – note that IC3/4 divide by two initially e.g. IC3Q0 is not connected) are passed into address multiplexors IC6/7 which takes data from this on VRAM write. These are then fed into 128x8 RAM IC9 (Yay ! - Motorola – again) which can receive data from the bus gated via IC8 on VRAMWRITE.

This data out – the character – is fed into the character generator ROM IC11 (along with Q1-A3 of IC3) and multiplexed by IC14 (using Q0-Q2 of IC1), and the gated by the two blanking signals from the counters.

There’s one obvious mistake here (there’s probably half a dozen others !). There are no delays between the reset pulses (HSYNC and VSYNC) and video output. So No front porch time, no period before vertical display starts. Oops.

I would probably fix it by having them trigger a monostable multivibrator (each) which gates their clock inputs, thus effectively shoving the picture right and down – the outputs of these would have to be gated as well. Another possible option is to replace the 7493 with 74193 – which are presettable – and have them presettable to a “negative” number, which means the gating as is would work, but to do that I’d probably have to pre-divide the HSYNC clock by two (replace IC5 with a JK Flip Flop ?) rather than add the set/reset stage on the end.

Still nobody’s perfect.

What’s missing – well, at the end there is the mixing of the sync signals to produce composite video, and obviously there is no sync generation – these days the best way of doing this seems to be a microcontroller….. which would probably be more powerful than this computer.

Next up ; finish the emulator and replica versions, and write a test program for them which tests all the various bits and bobs.

The software emulator does actually emulate the bits on the ‘basic’ board – the buttons and LEDs and so on – and I have actually written a basic monitor for that system, but from now on I’m going to ignore those buttons, I think. In a real system the LEDs would go nuts in classic Blinkenlights style when you are writing to the video, because they are mapped onto the same space.