NASCOM Keyboard

This document presents waveforms and descriptions that aim to show (some of) the inner workings of the NASCOM keyboard. It should be read in conjunction with a NASCOM 2 keyboard schematic (The NASCOM 1 keyboard schematic is *nearly* identical but has different component references). Refer to https://groups.io/g/Nascom-Computers/wiki/The-NASCOM-Keyboard for a general description and for faultfinding notes.

The keyboard is scanned under software control. Unless otherwise stated, the waveforms/timings were taken using NAS-SYS 3 running on a NASCOM 2 running at 4MHz with no wait-states. Code examination showed that the keyboard scanning code in NAS-SYS 1 is identical.

The NASCOM controls the keyboard with 2 output signals: RESET and CLOCK, also called “keyboard counter reset” and “keyboard counter clock”. These should not be confused with the RESET CPU signal which is generated on the keyboard. The NASCOM reads the keyboard status with 7 input signals, /S0.. /S6.

Illustration 1 is an overview of the periodic scan. Each scan consists of a high-going pulse on RESET (yellow) followed by 8 high-going pulses on CLOCK (pale blue). When no key is pressed, the pulses are evenly spaced. When one or more keys are pressed, the gap between pulses increases because NAS-SYS is performing additional processing to decode the key(s). In this example, NAS-SYS is in its idle loop waiting for key-presses; the period shown between scans is approximately 740us.

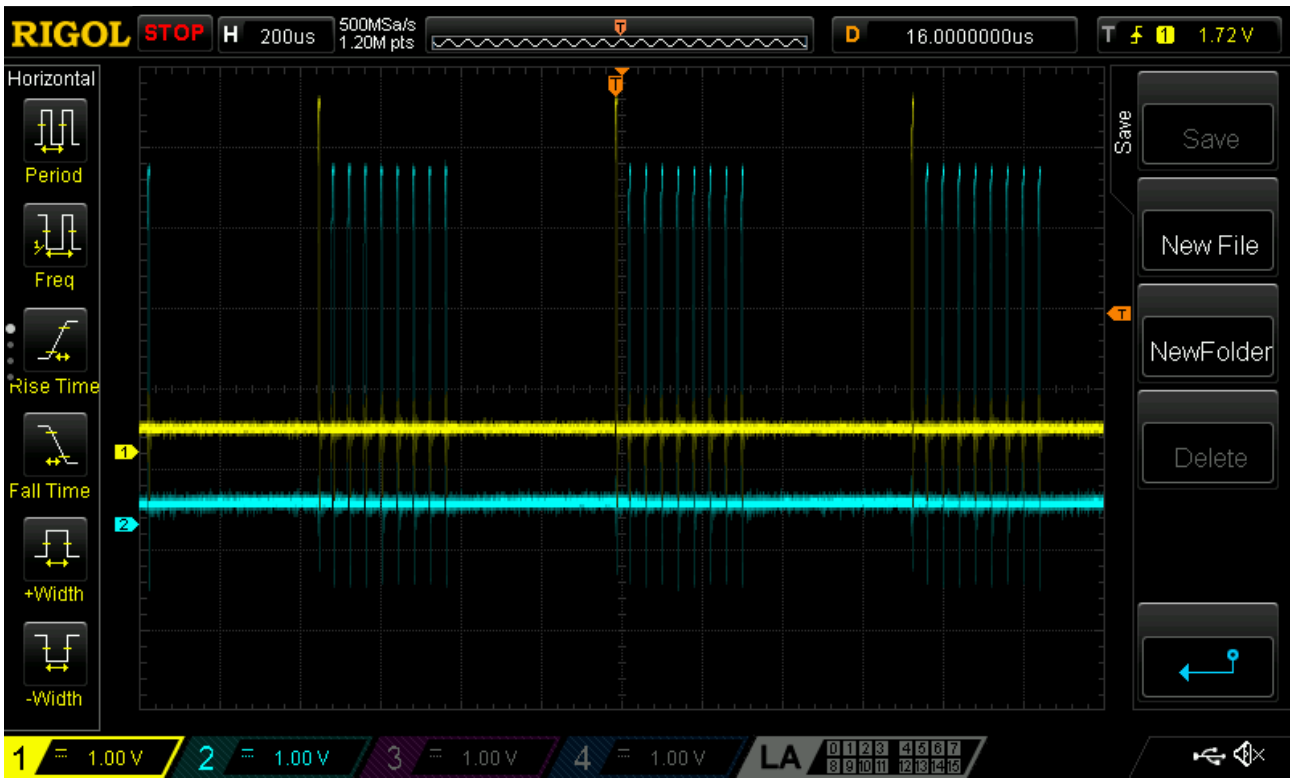

Illustration 1: Overview of periodic keyboard scan

Illustration 2 zooms in on a RESET (yellow, partially hidden by the cursor timings box) and two CLOCK pulses (blue). The pulse width of the RESET and CLOCK pulses are equal, and these widths do not vary (they are both imposed by the same subroutine, fflip). Each of these signals is loaded with 100pF capacitor to 0V, explaining the slugged rising edge.

The basic operation of the keyboard is thus: RESET resets state and selects drive line 0. CLOCK selects each subsequenct drive line in turn, wrapping round from 7 back to 0. When a drive line is selected, the SR (the cross-coupled NAND gates on the right-hand side of the schematic) are cleared and then the drive line is pulsed. Any sense lines for which a key is pressed generates a low-going pulse on one of the /A.. /G signals, which sets one of the SR latches. NAS-SYS reads the state of the latches.

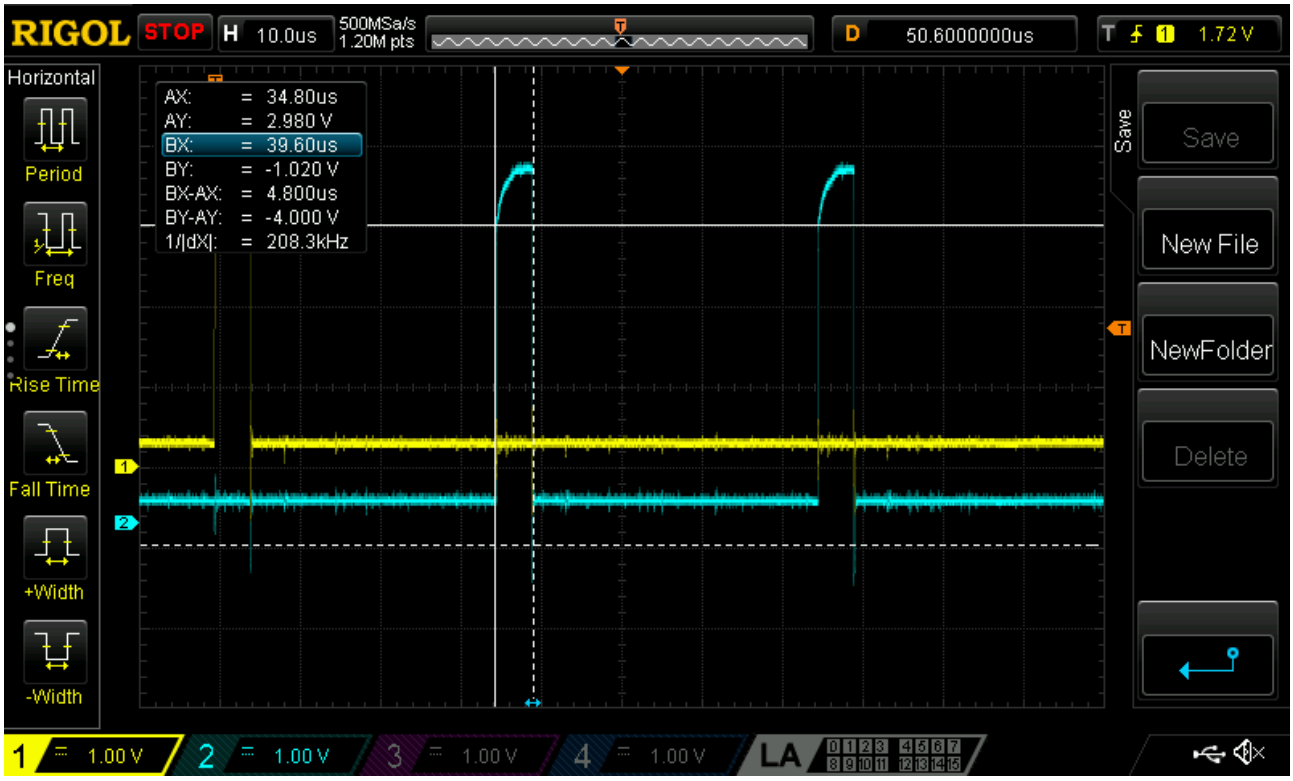

Illustration 2: CLOCK pulse

Both the RESET pulse and the CLOCK pulse trigger a chain of pulses, via the 74123 monstable, as shown in Illustration 3. RESET (yellow) and /CLOCK (IC4/3, pale blue) both have the effect of triggering monostable 1, generating the output 1Q (pink). This in turn triggers monostable 2, generating the output /2Q (dark blue).

The pink trace is inverted to generate /CLEAR, which resets the SR latches. The dark blue trace acts as a “strobe” into the 74145 decoder. The transformer effect in a depressed LICON switch causes the high-low transition of the dark blue trace to generate (via the CA3086 transistor array) an active-low SET pulse on one of the SR latches (signals /A.. /G).

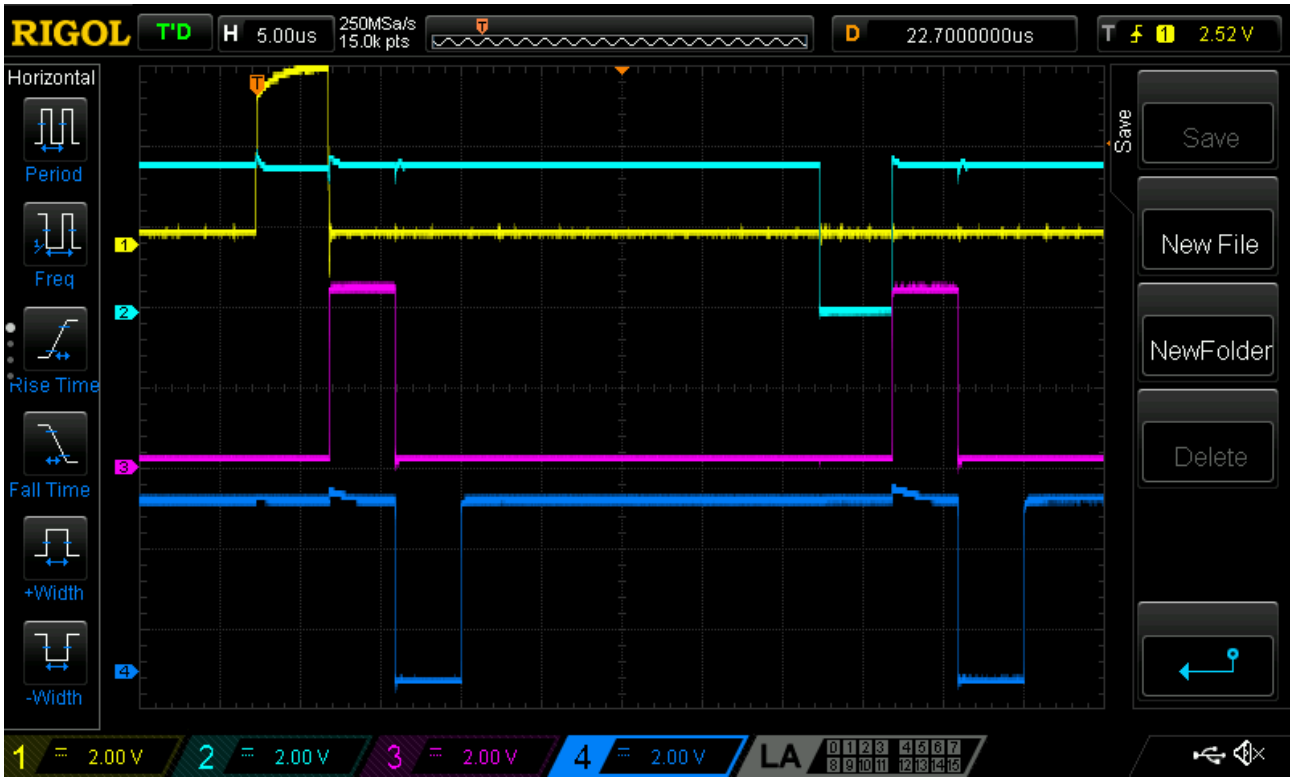

Illustration 3: Pulses from the monostable, triggered by RESET and CLOCK

Zooming out again (Illustration 4) shows RESET (yellow) followed by 8 /CLOCKs (pale blue). As before, dark blue is the output /2Q which acts as a “strobe” into the 74145 decoder.

Because both RESET and CLOCK trigger the monostable, it looks as though 9 drive lines are being sequenced, which is confusing because the schematic shows that there are only 8. This can be explained by looking at the pink trace. This is the /A signal, the set pulse for the SR latch associated with sense line 0, and it shows the effect of pressing the ‘-’ key, which is on drive 0, sense 0. There are 2 low-going pulses on /A. The RESET selects drive 0, the 8 CLOCK pulses select drive 1..8. However, the decoder works modulo 8 and so “drive 8” aliases to drive 0. Examination of the NAS-SYS code shows a read of the SR latches after the RESET and after each CLOCK. Therefore, NAS-SYS selects the drive lines in the sequence: 0, 1, 2, 3, 4, 5, 6, 7, 0. It only uses the result of the first scan of drive0 to determine the state of the SHIFT key.

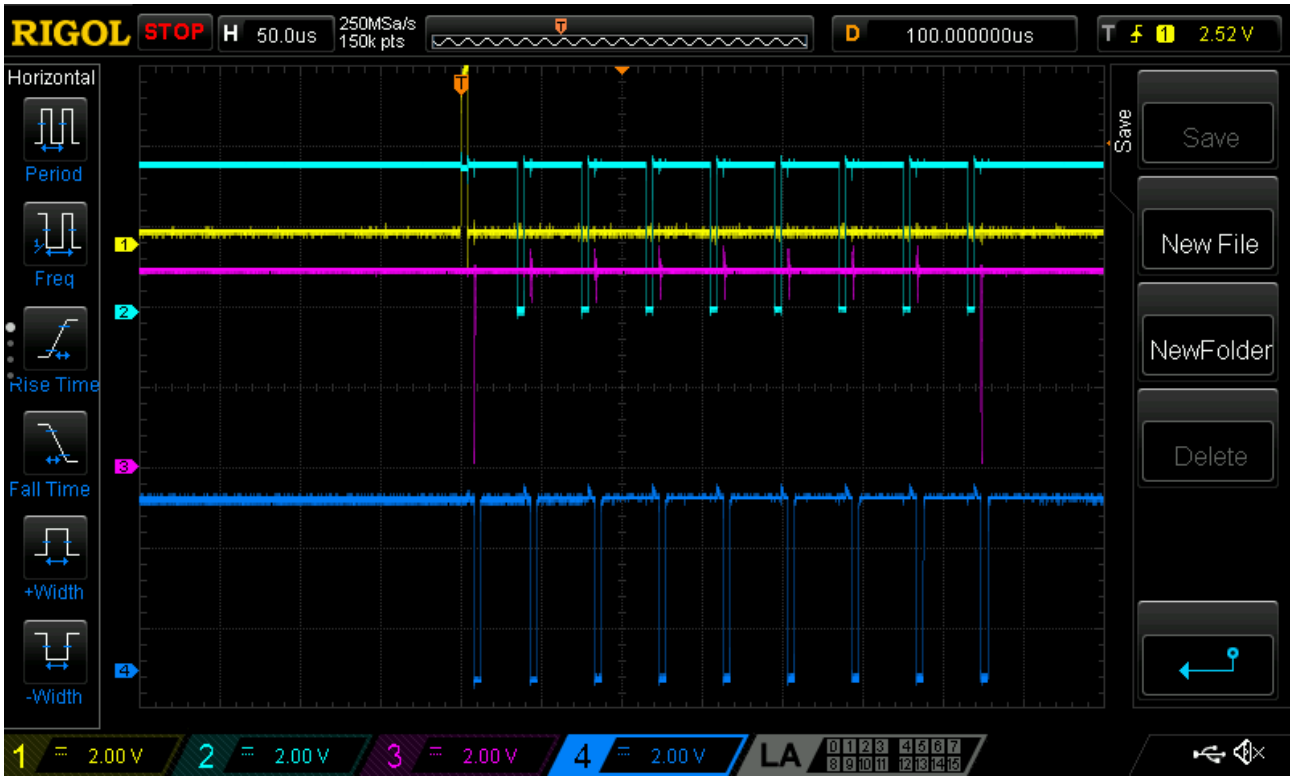

Illustration 4: Pressing the ‘-’ key pulses /A (pink) to set one of the SR latches

Illustration 4 showed the ‘-’ key pressed, and a pulse on /A when drive 0 is selected, Illustration 5 shows the same signals but with the ‘5’ key pressed: this time, the pulse on /A occurs when drive 1 is selected (drive 1 is selected by the first pulse on CLOCK).

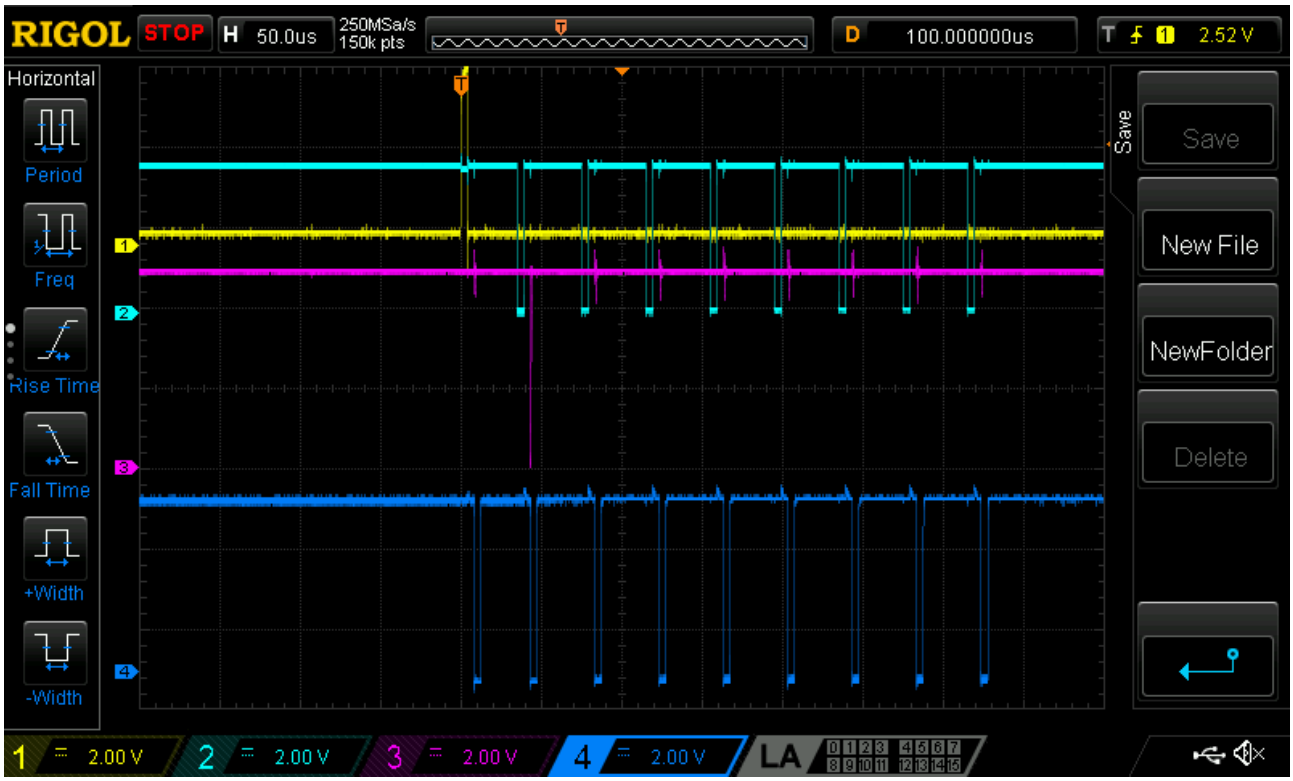

Illustration 5: Pressing the '5' key pulses /A (pink) to set one of the SR latches

Illustration 6 zooms in on the pulse on /A (pink) which sets one of the SR latches. As before, it also shows RESET (yellow), the active edge of the first /CLOCK (pale blue) and the monostable /2Q output (dark blue). The pulse on /A is about 270ns wide.

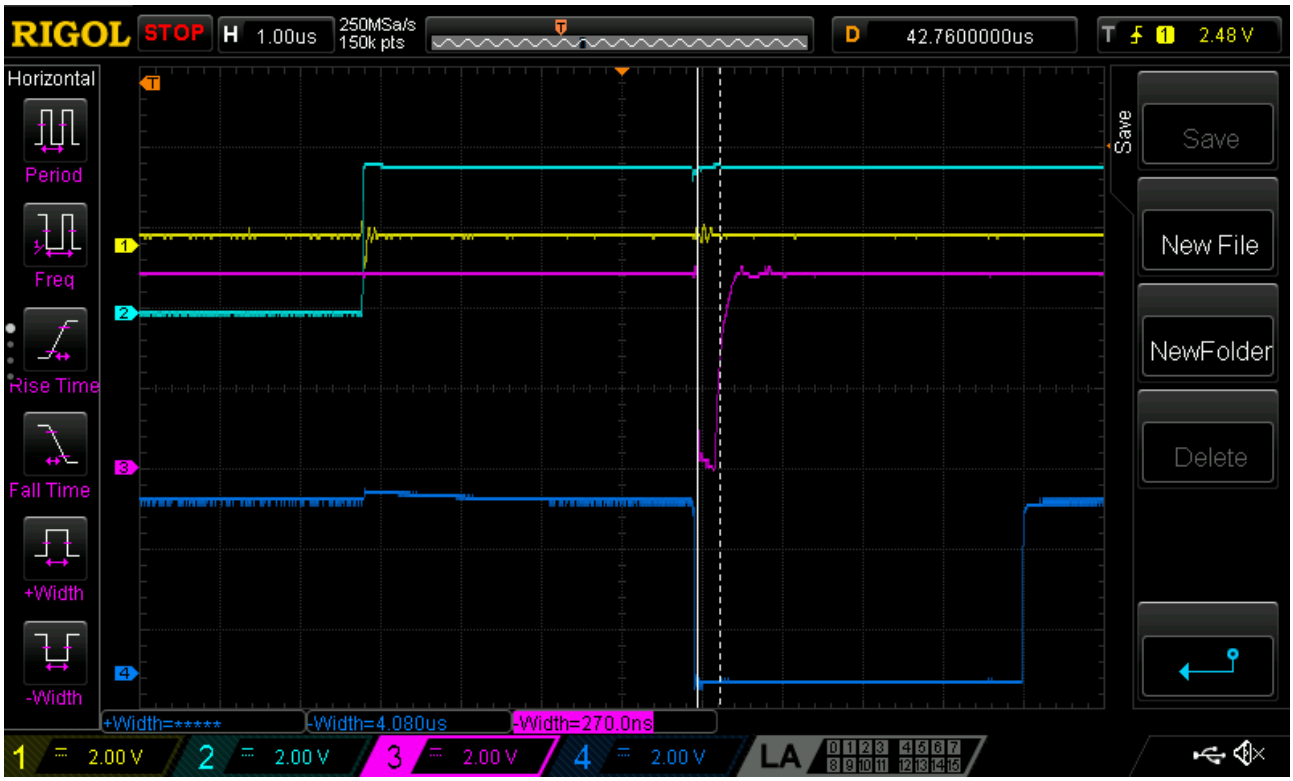

Illustration 6: Zoom in on /A (pink) which sets one of the SR latches

In Illustration 7 the pink trace is the output of the SR latch set by /A. The pink signal goes 0→1 as a result of the low-going edge of /A (see Illustration 6) and goes 1→0 as a result of the low-going edge of /CLEAR (which corresponds to the high-going edge of the pink signal in Illustration 3).

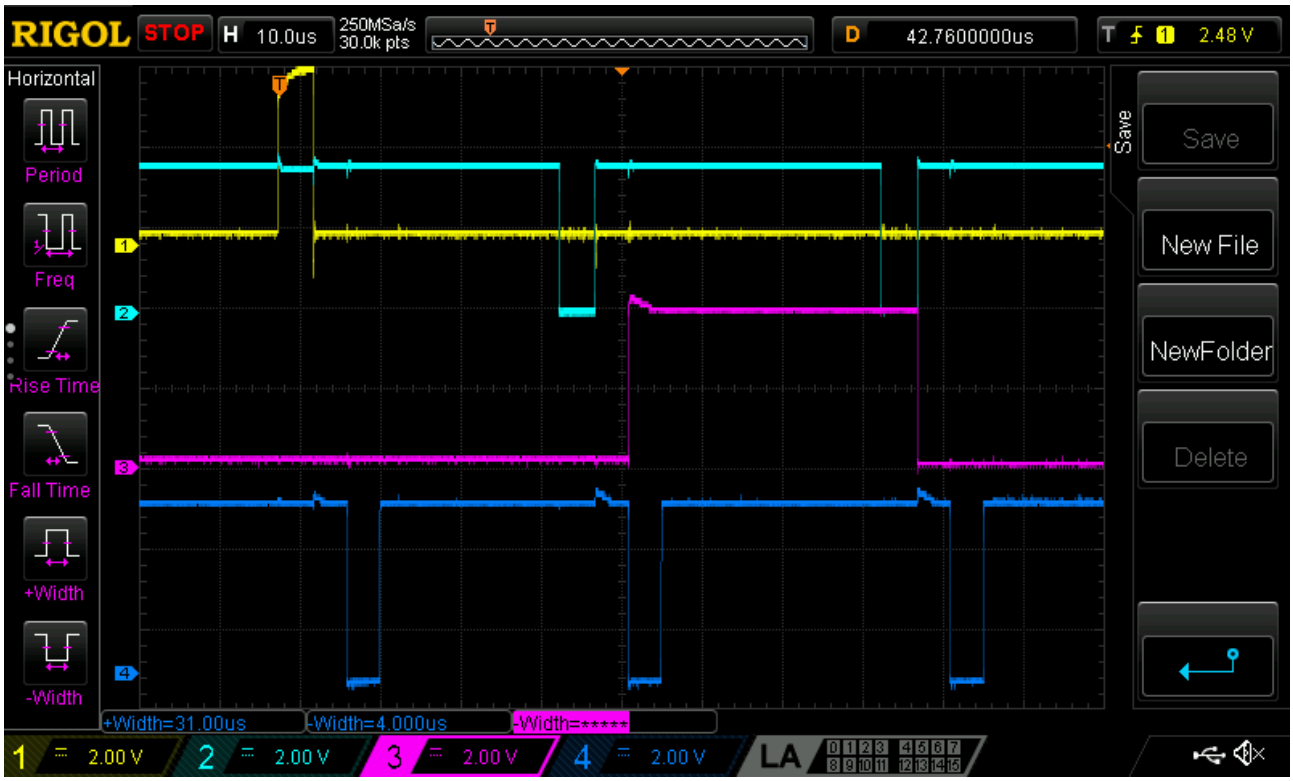

Illustration 7: Output (pink) of SR latch set by /A

Finally, we can put it off no longer; we have to look at the analogue stuff going on around the keys and the CA3086 transistor arrays. First thing to say is that it is super-confusing to look at the outputs of the SN74145 decoder; I expected only one output at a time to be asserted, low. Actually it seemed that all of them are asserted for each clock. A second look at the schematic explains it: the SN74145 has open-collector outputs and all of the outputs are connected together at the “top” of the columns (after each drive line has passed through the primary coil of the LICON switches in the column) This common signal is connected through 1k resistor to +5V.

When one open collector output is selected from the SN74145, it draws a current through the corresponding selected drive output and the 1k resistor - but no current through the other outputs. As a result of the current draw, the ‘common’ voltage drops. This is the perceived behaviour when you look at the SN74145 outputs with an oscilloscope or logic analyser. The key thing is the rate of change of current in the selected drive line and not the voltage.

Because the SN74145 outputs are open-collector, the low-going transitions on the drive lines are fast (active) while the high-going transitions are slow, RC curves. It is the fast low-going edge that is responsible for inducing a pulse in the secondary that is large enough to switch one of the transistors in the CA3086 array to generate a set pulse to one of the SR latches.

Illustration 8 and Illustration 9 show the behaviour of the sense 0 line when a key is pressed. As before, the pale blue trace is the active edge of the first /CLOCK. The dark blue trace is the LICON end of R27, and the pink trace is the CA3086 end. The change in voltage level when the key is pressed is enough to switch the transisor.

Each transistor (that's in use) in the CA3086 array has 2k2 in its collector and 2k2 plus 22 ohms in its emitter, so for DC conditions the collector can't fall below 2.5V (plus a bit) which is logic 1 for TTL. However, for transient conditions, when a key is pressed, the 2k2 in the emitter is bypassed by 10nF (C3 on NASCOM 1 kbd, C5 on NASCOM 2 kbd). The AC gain is thus much higher (100) and the collector is able to fall briefly to a logic 0, setting one of the SR latches. This one capacitor is common to all the keys.

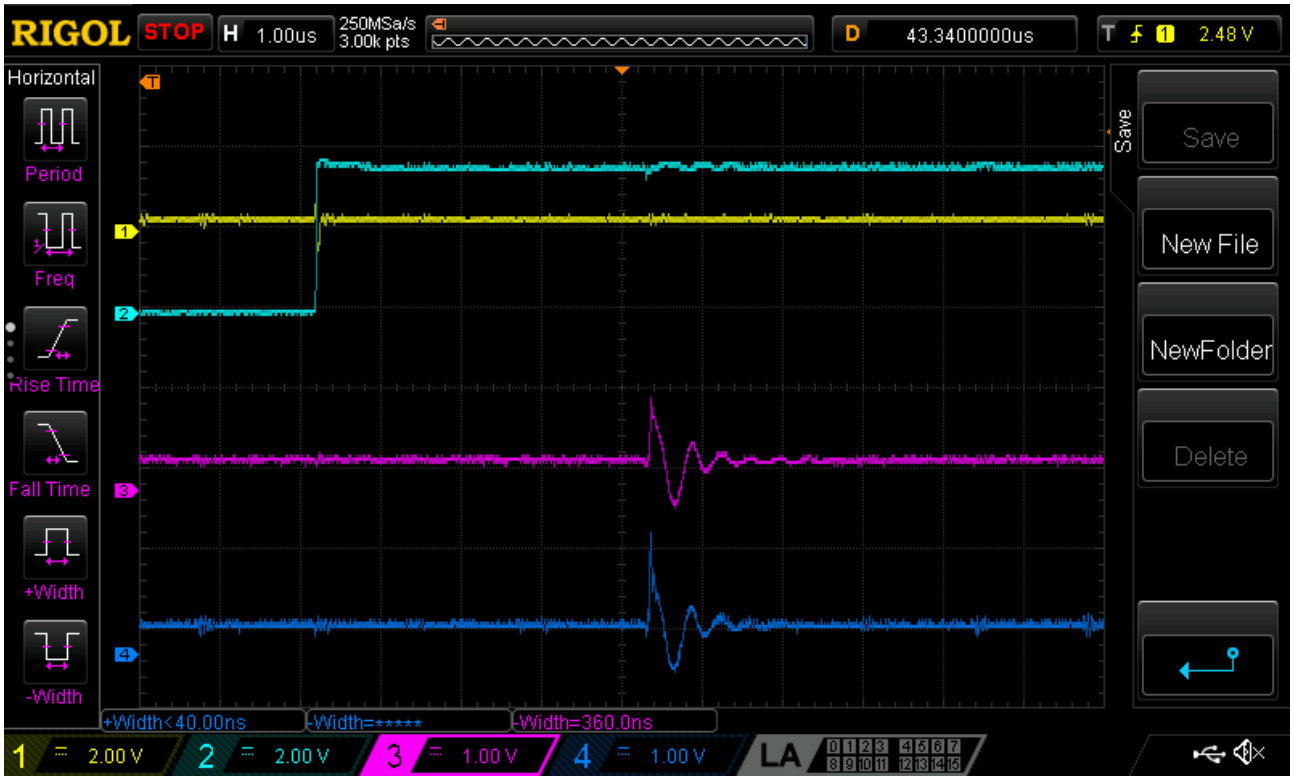

Illustration 8: Sense line 0 (pink, dark blue) with no key pressed

Illustration 9: Sense line 0 (pink, dark blue) when '5' key pressed

Finally, Illustration 10 is like Illustration 9 but now the pale blue trace is showing the pulse on /A caused by the changed voltage levels in the sense line.

Illustration 10: Sense line 0 (pink, dark blue) when '5' key pressed and resultant pulse on /A

Keyboard repair

I have seen 2 faulty keyboards. In both cases, the fault was that a particular set of keys did not work; checking these keys on the schematic showed that they had a common “sense” line and the fault was isolated to a transistor in one of the CA3086 arrays. NOS replacement parts (typically the National Semiconductor LM3086) are available (ebay). An alternative is to wire 4 BC549 (or similar NPN) transistors on a 14-pin DIL header.

When probing the keyboard note that the 7493 counter has power/gnd on pins 5/10 and not the conventional 7/14. This caught me out when I put my scope ground on pin 7 (which is NC).

Shift/Enter Pause/Break in BASIC

In NASCOM BASIC, the Shift/Enter keystroke combination is used to pause execution (typically screen output) of a running program; a second Shift/Enter stops the program.

The BASIC ROM routine to check these keys is named SFTENT. It does not use a monitor call but performs port 0 read/write drectly. The Shift and Enter keys are both on row 0 and so the code only checks this row rather than performing a full keyboard scan.

When the monitor scans the keyboard it asserts *either* RESET or CLOCK. When SFTENT scans the keyboard it allows RESET and CLOCK to overlap, as shown in Illustration 11. The illustration shows RESET (yellow), inverted CLOCK (pale blue) and port 0 /IORD (dark blue).

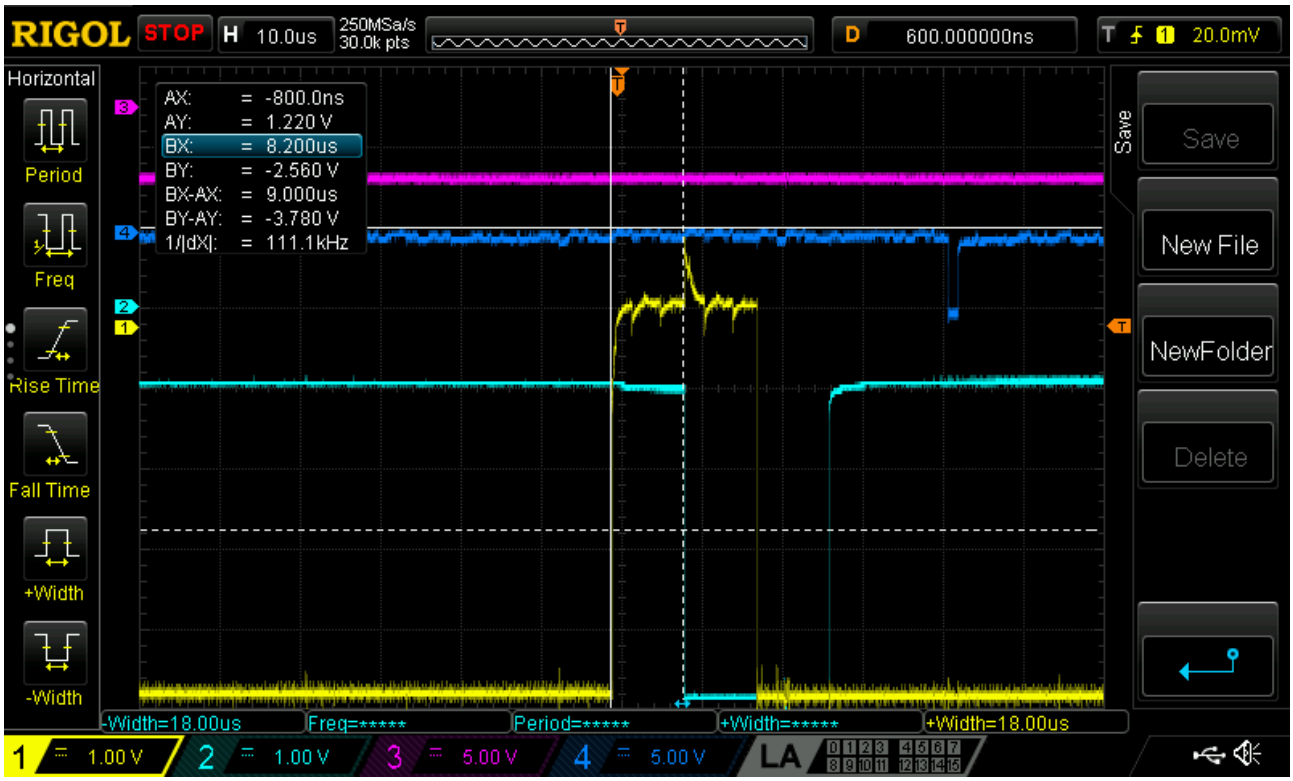

Illustration 11: BASIC Shift/Enter scan

This behaviour uncovered a bug (reported by Ian Bradbury) in my pi-pico-based NASCOM keyboard replacement, and I then realisd that an equivalent bug exists in the PS/2 keyboard decode logic on the NASCOM 4. This is the correct way to interprete the signals:

- RESET is an active-high level-sensitive signal which overrides CLOCK: the row count remains 0 while RESET is 1, regardless of the state if CLOCK
- CLOCK is edge-sensitive signal: when RESET is 0, a rising edge on CLOCK increments the row count, modulo 8.

Referring back to Illustration 11, RESET remains high when the rising edge occurs on CLOCK (falling edge on the inverted CLOCK shown) and so the row remains at 0. In the buggy implementation, both RESET and CLOCK were treated as rising-edge sensitive: the result was that the row count was advanced to 1 so that the Shift/Enter combination was not detected.

Errors, omissions and comments to the author, please: foofoobedoo@gmail.com

Date	**Author**	**Notes**
09-Feb-2020	Neal Crook	First release.
01-Mar-2020	Neal Crook	Fix typos, revise analogue description after feedback from Dave Roberts; additional technical detail from Dave and John.

28-Mar-2020	Neal Crook	Clarify 8 vs 9 scan lines after inspecting NAS-SYS source. Add reference to WIKI page.
23-Jun-2025	Neal Crook	General tidy-up. Add material about BASIC “break” and some debug notes and emulation notes