You can be amazing! Here's a case study. Devise a clean fix, to a widespread difficult problem in your industry. Become the primary go-to source for the solutions. In this example, tech aspects at the lowest level, can influence system behavior at the top level. It's "vertical."

Here we review a legacy problem that presented difficulties for system designers. We explore how we solved this topic cleanly and permanently, discovered and published methods to test for it, and shared the test programs with the customers, thereby establishing our employer as the primary source for understanding of this technology.

The problem of Address Float arose on asynchronous address inputs to a memory device. This problem existed in the early 1980's when system designers often allowed address lines to be unterminated for part of the cycle. As a result, these inputs would meander uncontrolled in non-valid (intermediate) logic states, resulting in multiple selection of internal decoders, and hence destruction of stored data.

Today system design techniques are improved, buses are better controlled, and this topic is no longer critical for buffer design. However, study of this interesting phenomenon is instructive, because it illustrates several important points:

(-1-) Make your circuits immune to out-of-spec conditions.
(-2-) Exceed specified performance to increase margin and guardband.
(-3-) Accept the customer's environment, until their practices improve.
(-4-) Test to quantify the circuit's behavior.
(-5-) Share the test techniques with customers so they can benchmark your product against others.

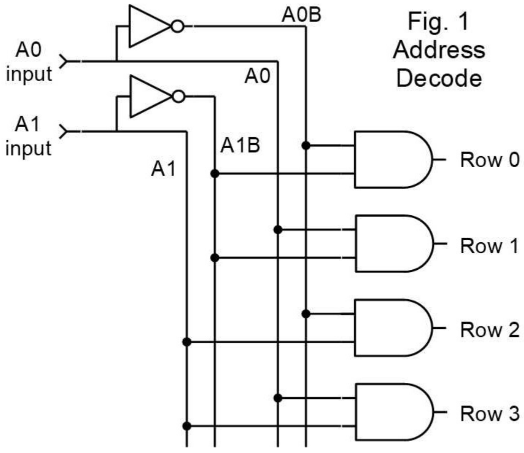

Fig. 1 Address Decode

The key insight is that, in order to tolerate a floating input, the buffer must contain sufficient DC margin. Since *time is not a parameter in a dc transfer curve*, additional temporal margin (time delay) *will not have any effect on this issue*. Many early static memory design projects failed, due to not recognizing this difference.

Let's briefly review the normal techniques for avoiding multiselection in ordinary conditions. Figure 1 shows a typical row decoder for a static memory. Only one Row Line may be enabled at any given time.

Address A0 selects between Row0 and Row1. If A0 and A0B are both asserted at the same time, two row lines will be selected, resulting in data corruption in the memory cells.

First let's recall how this decoder behaves in the time domain. Figure 2 shows the desired behavior of the Row Lines in response to a change in the address.

To avoid multi-selection, the input buffers and decoders are designed so that Row 0 is deselected before Row 1 is selected. This time delay is shown at the bottom of Figure 2. As long as the address inputs are at valid logic levels and have clean edges, the circuit will perform as desired.

So far, so good. These are the ordinary requirements when signals are behaving normally. But now, let's see what happens when addresses are floating.

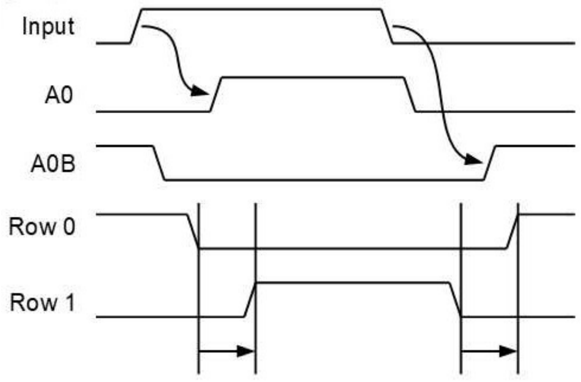

Fig. 2 Time Separation of Selected Rows

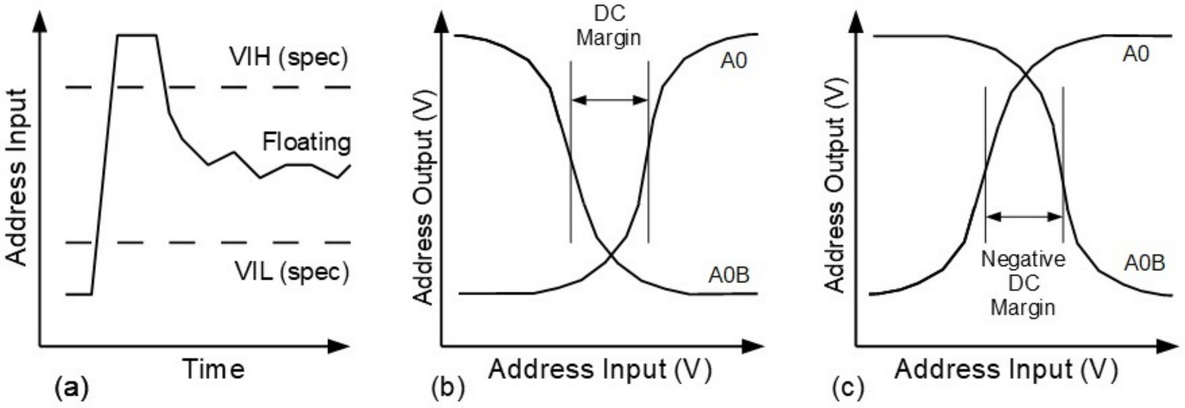

Fig. 3 (a) Address Float, (b) DC Margin, (c) Negative DC Margin

Now suppose (Fig. 3a) that an address input meanders in the intermediate range, between a zero and one level, for a period of time. Notice that is a DC condition. To avoid multiselection, time delay does not help. There must be some separation in the DC transfer curve (Fig. 3b) so that the *true* and *complement* lines (A0 and A0B) are both low, in this region.[1] If there is a range of input for which both are high (Fig. 3c), the buffer may be said to have *Negative DC Margin*, causing multiselection.

The key insight is that since *time is not a parameter in a DC transfer curve*, additional time delay will not improve the DC margin, but rather only slows down the RAM.

Testing

We needed to devise a test for address float. Standard memory testers in production environments do not allow any condition which mimics a floating address. However, a simple trick allowed us to test for this issue. Memory testers allow you to program the logic high and low levels while cycling. We performed a simple set of memory read cycles in which the output level (programmable) was set to a point in between "zero" and "one," and repeated this cycle for multiple levels. (Fig. 4.) After these cycles, the memory would be read (using normal input levels) and checked for any alteration in stored data.

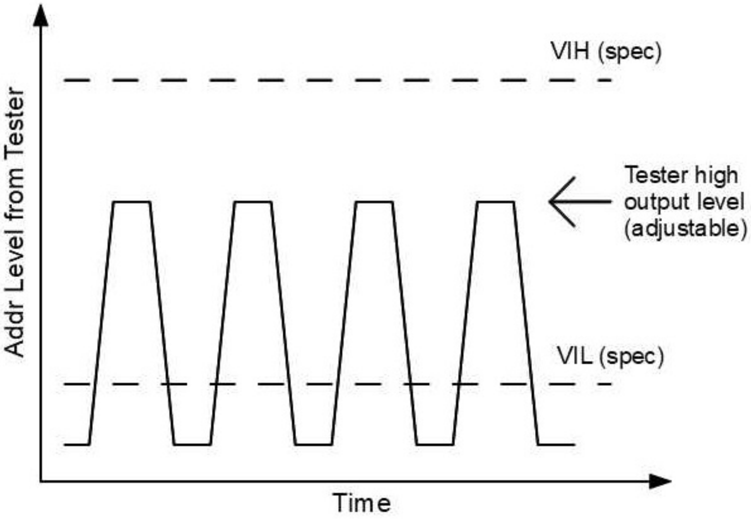

Fig. 4 AC test to detect Address Float sensitivity

It was found that this test correlated well with customer experience of RAMs in their systems.

An added advantage is that we are thus able to test individual address buffers for the issue, by directing this waveform to a single address bit, and keeping all others at full spec levels (zero or one). This ability was crucial in performing the correlations to buffer DC margin, and thereby proving that the margin was in fact the cause of the float performance (shown in next section.)

It was also possible to vary the length of time that the input spent in the mid level, and characterize the duration required to produce a failure.

This process was then repeated in the inverse, with the high level at spec, and the low level in the intermediate range.

[1] This was pointed out in an early paper, "Two 13-ns CMOS SRAMs...", Flannagan et al., *IEEE Journal of Solid State Circuits*, October 1986, pp. 692-703, Figure 3.

Float Performance Correlated to DC Margin

The DC margin of an existing production part was characterized from simulation, and correlated to the performance on our address float test, and to the customer reports. Using our method to test individual addresses, the following three groups of address buffers were found:

Buffers	DC Margin Range (mV)	Float Performance
Group 1	103 to 107	Pass
Group 2	25 to 45	Marginal
Group 3	-50 to -70	Fail

A design revision was performed on this production part, increasing the input DC margin on the Group 2 and Group 3 buffers to over 100 mV. The result was that the chip completely passed the customer's requirement in their systems.

We also provided the testing method (illustrated in Fig. 4) to our customers. When you have a technical advantage over the competition, it's beneficial to ensure that your customers are able to test to the issue effectively.

Contribution of Hysteresis and Latches

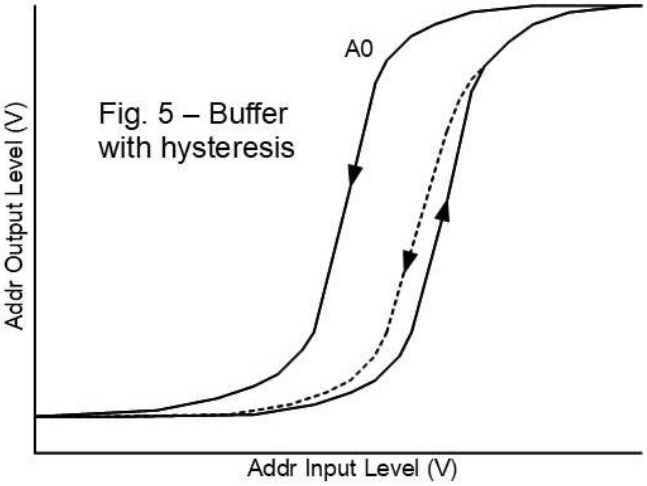
Fig. 5 – Buffer with hysteresis

A small amount of hysteresis is helpful for added noise reduction. But does it help in address float?

Actually, absent a low DC crossover, the hysteresis loop by itself *does not solve address float*. Here's why.

A simple weak feedback path provides the familiar double-threshold curve. The insight is that when the input enters the non-valid (transition) region (rising arrow in Fig. 5), without fully switching the latch portion, then the return path (dotted line) will be very near, or possibly on-line, with the initial excursion. The hysteresis loop is helpful in noisy conditions; but for address float, it must also be accompanied by a true DC margin for the circuit.

Consider an address buffer for which the input is changing very slowly, as shown in Fig. 6. The slow ramp is a violation of the input specs for the product. But our purpose is to provide immunity to non-valid conditions. A simple inverting buffer may respond to this ramp with oscillation, as shown. Typically this would be caused by feedback through the local power supply or other paths.

So, let's consider our old friend, the semaphore latch. It does provide more noise tolerance. While helpful, it must be accompanied by a low DC crossover, for a clean solution to address float. Let's see how that will look in a design.

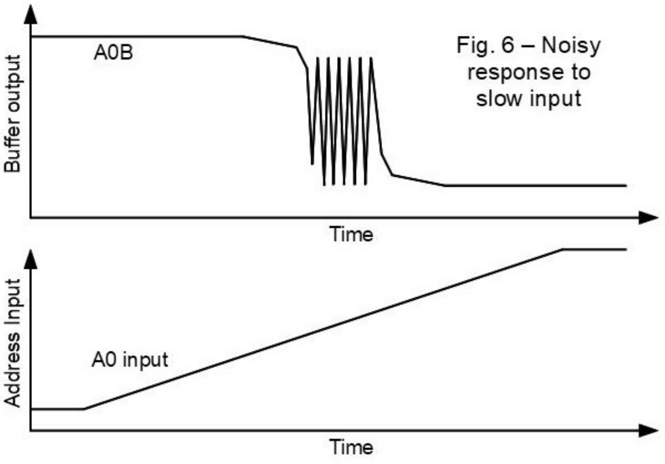
Fig. 6 – Noisy response to slow input

Key Insight: Parallel Paths with Individually Adjustable Switching Levels

For new designs, we devised an address buffer with two separate parallel paths, one for A(true) and the other for A(complement), so that the input switching level could be adjusted differently for each one, and providing a much higher DC margin than with a single circuit.[2] In the circuit of Figure 7, there are two immediate input stages in parallel; D1 and D2. The input switching level of D1 is higher than that of D2, providing controllable DC margin for the buffer.

The design of a buffer with a combination of AC margin, DC margin, hysteresis, and the semaphore latch, provides a circuit with truly broad applicability.

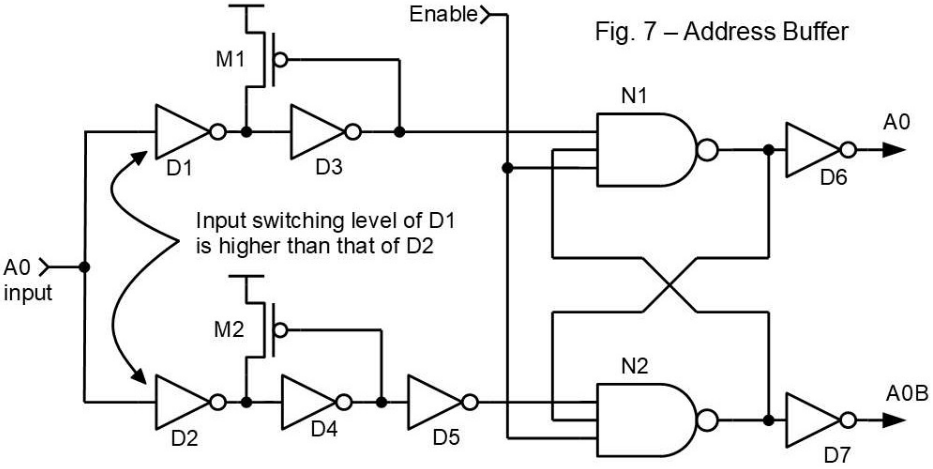

Fig. 7 – Address Buffer

Input switching level of D1 is higher than that of D2

Let's summarize these design characteristics. The address buffer combining all of these features (Fig. 7) provides two separate input stages, input switching level of D1 higher than that of D2, creating a deadband in which A0 and A0B are both low. M1 and M2 provide weak feedback for hysteresis that is separately adjustable for each path. N1 and N2 represent the semaphore latch. "Enable" is a signal which when low will cause both A0 and A0B buses to be low, disabling all decoders. The Enable signal could be placed on a common N-channel pulldown shared between N1 and N2.

This concludes our exploration of the address input buffer with high immunity to out-of-spec address bus characteristics, including address float.

Other signal issues, such as line reflections, incident-wave switching, and point-to-point design, are very interesting; we will explore them in another essay.

In summary, good design choices are matched to your customer's needs. Characterize the environment in which a circuit is intended to work. Communicate with the technical staff at the destination and share information on how to test the product. Here again, communication among technical teams is just as important as signaling between digital components.

Thanks to John Carlsen for advice and encouragement during this write-up. I hope that everyone has found this essay useful. Your comments and suggestions are always welcome! Regards, - Steve

Published by **Flannagan I.C. Technical Services & Consulting, Inc.** WWW.FLANIC.COM

The information is provided "as is." We make no representation or warranty of any kind, as to the content or its applicability for any purpose, and assume no liability or responsibility for any errors, omissions or interruptions in the information included in this article, and specifically disclaim any duty to update such content.

[2] Flannagan & Voss, *Memory input buffer with hysteresis and DC margin*, U.S. Patent 4,807,198 – Feb. 21, 1989