

# The Embedded Muse 41

Editor: Jack Ganssle ([jack@ganssle.com](mailto:jack@ganssle.com))

December 24, 1999

## Metastability

Our digital world is marred by weird effects that's the stuff of nightmares. Simple timing errors can transform our "perfect" logic designs into nests of lurking disasters, disasters whose frequency is so rare we're inclined to call the failures "glitches" and dismiss them out of hand.

One of the more insidious problems lies in violating a flip flop's setup time. EVERY flop has a specified minimum setup time: the number of nanoseconds that data must be present at the device's input before the clock signal transitions. If the data appears inside of this window (say, if the min setup time is 2 nsec and data changes 1 nsec before clock changes) then the output of the flip flop is undefined. This condition is called "metastability", and is clearly a profound issue.

Fully synchronous systems don't suffer from metastability issues since, by definition, data changes at safe times. Instead, metastability comes into play when data is asynchronous to clock. Suppose a parallel encoder goes into a 12 bit latch that is clocked by the CPU when the processor is ready to read the data. The encoder is asynchronous to the CPU's clock, so there's no reasonable way to insure that the changing encoder data will meet the latch's setup time. Occasionally – not often – the latched data will be corrupt.

Now, let's examine the meaning of "corrupt". As a young engineer I learned that violating a flop's setup time would yield an unpredictable output: that is, try to clock in a one, and the output could be either a one or a zero. However, most papers on the subject suggest metastability yields very long propagation times. That is, instead of taking 5 nsec to clock the data through, the time can rise to many tens or hundreds of nanoseconds, the amount of time being a function of the device's construction and how badly the setup time is violated.

Well, a long propagation time can be a problem but sure seems less disastrous than having the device switch incorrectly. While giving an on-site seminar to a company in the southwest recently this question came up. Since then I've been rereading the literature trying to understand if metastability yields ONLY long prop times, or if it can indeed cause the flop to settle in a random state.

*Copyright 2000 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.*

My favorite book on high speed design (“High Speed Digital Design” by Howard Johnson and Martin Graham) talks only about an increase in prop delay. But scope pictures on pages 129 and 130, which show how the delay increases, also demonstrate the flop’s output switching correctly only about half the time. They’re trying to clock a one through, but the flop unpredictably seems to stabilize in either a one or a zero condition.

Another great reference on the subject is TI’s “Metastable Response in 5-V Logic Circuits” (<http://www-s.ti.com/sc/psheets/sdya006/sdya006.pdf>). This work, too, discusses the increase in prop delay without specifically referring to a flop possibly assuming an incorrect state. Timing diagrams 3 and 4, though, clearly suggest that such an error can and does happen.

Metastability problems were once at least somewhat visible, back when we used individual flip flops. Now, with hundreds or thousands of registers deeply buried inside of FPGAs, PLDs and ASICs, the ever lurking peril of metastability is much more difficult to diagnose.

So, here’s an end-of-year question for you gurus out there: we know that prop delay in the metastable region increases exponentially. Can the flop indeed also settle into an incorrect state? I’ll share the results in another issue of the Muse.

## **Thought for the Week**

From Mike O’Brien:

Twas the night before Y2K, and all through the nation  
We awaited The Bug, the Millennium sensation.  
The chips were replaced in computers with care,  
In hopes that ol’ Buggy wouldn’t stop there.

While some folks could think they were snug in their beds,  
Others had visions of dread in their heads.  
And Ma with her PC and me with my Mac,  
Just logged onto the Net, and kicked back with a snack.

When over the server, there arose such a clatter,  
I called Mister Gates to see what was the matter.  
But he was away, so I flew like a flash  
Off to my bank to withdraw all my cash.

And what with my eyes should I unhappily see?  
My good old Mac looked very sick to me.

*Copyright 2000 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.*

The hack of all hackers was looking so smug,  
I knew that it must be the Y2K Bug.

His image downloaded in no time at all,  
He whistled and shouted, Let all systems fall!  
Go Intel! Go Gateway! Now HP! Big Blue!  
Everything Compaq And Pentium too!  
All processors big, All processors small,  
Crash away! Crash away! Crash away all!

As I drew in my breath and was turning around,  
Out through the modem, He came with a bound.  
He was covered with fur, and slung on his back,  
Was a sackful of viruses, all set for attack.  
His eyes - how they twinkled! His dimples – how merry!

As midnight approached, things soon became scary.  
He had a broad little face and a round little belly,  
And his sack filled with viruses quivered like jelly.  
He was chubby and plump, perpetually grinning,  
And I laughed when I saw him (though my hard drive stopped spinning).

A wink of his eye, and a twist of his head,  
Soon gave me to know a new feeling of dread.  
He spoke not a word, but went straight to his work,  
He changed all the clocks, then turned with a jerk.

With a twitch of his nose, and a quick little wink,  
All things electronic soon went on the blink.  
He zoomed from my system, to the next folks on line,  
He caused such a disruption, could this be a sign?

Then I heard him exclaim, with a loud, hearty cry,  
Happy Y2K to all, kiss your PC good-bye!

## **About The Embedded Muse**

The Embedded Muse is an occasional newsletter sent via email by Jack Ganssle. Send complaints, comments, and contributions to him at [jack@ganssle.com](mailto:jack@ganssle.com).

To subscribe, send a message to [majordomo@ganssle.com](mailto:majordomo@ganssle.com), with the

*Copyright 2000 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.*

**The Ganssle Group, [www.ganssle.com](http://www.ganssle.com)**

words “subscribe embedded *your-email-address*” in the body. To unsubscribe, change the message to “unsubscribe embedded *your-email-address*”.

The Embedded Muse is supported by The Ganssle Group, whose mission is to help embedded folks get better products to market faster. We offer seminars at your site offering hard-hitting ideas - and action - you can take now to ***improve firmware quality and decrease development time***. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.

*Copyright 2000 by The Ganssle Group. All Rights Reserved. You may distribute this for non-commercial purposes. Contact us at [info@ganssle.com](mailto:info@ganssle.com) for more information.*

**The Ganssle Group, [www.ganssle.com](http://www.ganssle.com)**