

Re: Debouncing....at About 1Mhz

Source: <http://sci.tech-archive.net/Archive/sci.electronics.design/2007-11/msg00078.html>

- *From:* John Fields <jfields@xxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 01 Nov 2007 15:14:36 -0500
-

On Thu, 01 Nov 2007 12:44:18 -0700, Tom Bruhns <k7itm@xxxxxxx> wrote:

On Nov 1, 11:17 am, John Fields <jfie...@xxxxxxxxxxxxxxxxxxxxxxxx> wrote:

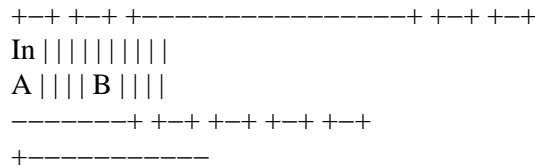
On Thu, 01 Nov 2007 08:17:25 -0700, John Larkin

<jjlar...@xx> wrote:

On Wed, 31 Oct 2007 21:53:15 -0800, D from BC <myrealaddr...@xxxxxxx> wrote:

This has got to be a classic signal clean up problem....

I need a circuit that triggers on edge A, then ignores about 0.1uS of jitter then triggers on edge B and then ignores a following 0.1uS of jitter.



|<0.1uS>| |<0.1uS >|

Re: Debouncing....at About 1Mhz

|< 0.5uS >|

Out +-----+
||
A' B'
-----+ +-----

Edge A to A' is ~ less than 10nS

Edge B to B' is ~ less than 10nS

All values are approximates.

"In" and "Out" are repeating waveforms.

I think I can do it with:

1 flip flop

1 >0.1us delay circuit

Sprinkled with gates..

Or maybe I need 2 flip flops..one for edge A
and one for edge B..

I'm not even sure yet which type of FF to
get.

If anybody has done this problem before and
doesn't mind sharing..let
me know a topology...

In the meantime, I'll be doodling until I get a
solution...

Re: Debouncing....at About 1Mhz

D from BC

Looks like you can do it with a dflop, a quad xor, and an RC.

Run the input through a delay-line edge detector (three gates of delay, then xor) and clock the dflop. Then rc lowpass the input and apply it to D. Q is the output.

???

Edge A to A' is ~ less than 10nS
Edge B to B' is ~ less than 10nS

JF

Right, so the xor plus the dflop clock to output needs to have less delay than that... should be easy with fast parts. The xor is just to get the same polarity clock pulse from each (leading) clock edge. The d input is RC delayed, so you capture the "old" level; thus you take Q-not as the output. RC must be long enough to get past the multiple transitions at each edge. In fact, you could do it, I think, with an SR f/f (cross-coupled NANDs) driven from a similar xor nanded with RC-filtered clock...

Something like this?: ;)

news:c56ki3h050okp6mkolm24oq7n67mvmt6rn@xxxxxxx

JF

.