

## Re: How inaccurate is a 555 or 7555 REALLY?

---

*Source:* <http://sci.tech--archive.net/Archive/sci.electronics.design/2006-12/msg01872.html>

---

- *From:* [bill.sloman@xxxxxxxx](mailto:bill.sloman@xxxxxxxx)
  - *Date:* 7 Dec 2006 17:30:41 -0800
- 

John Fields wrote:

On 6 Dec 2006 16:18:51 -0800, [bill.sloman@xxxxxxxx](mailto:bill.sloman@xxxxxxxx) wrote:

John Fields wrote:

On 5 Dec 2006 21:21:13 -0800, [bill.sloman@xxxxxxxx](mailto:bill.sloman@xxxxxxxx) wrote:

Phil Allison wrote:

<[bill.sloman@xxxxxxxx](mailto:bill.sloman@xxxxxxxx)>

John  
Fields:

<snip>

Yes. I've seen them. All very nostalgic. Your recent exercise with a 4024 (the last post in this thread) reminds me of stuff I was doing in 1974 with a 4040, though I didn't decode with three diodes and a resistor, even back then.

-----  
Didn't know how? It's just an RDL AND...\$

Of course I knew how. But the resistors I would have had to use were 0.6" long and 0.2" wide on the board, so it would have wasted space, not to mention confusing the final test technicians and the service

Re: How inaccurate is a 555 or 7555 REALLY?

engineers.

A more modern programmable logic part – like the Xilinx Coolrunner series – with in-system programming, 1.8V supply and some really compact packages – looks as if it would be even nicer, but I've not yet been able to contrive an excuse to play with one of these parts.

---

You seem to be gravitating toward the "use a PIC" mentality when, for the OP's purpose, suggesting that he use anything he'd have to learn to program would be ridiculous. But that's never stopped you before, huh?

---

A PIC may be programmable, but it isn't programmable logic, and can't do a lot of things that are easy in programmable logic.

Programming programmable logic can be handled through a variety of interfaces, depending on the manufacturers programming software (usually available free for everything except bleeding-edge new parts, which the Xilinx CoolRunner isn't – Xilinx bought it from Philips).

Most of the ones I've run across have included a graphical option, which makes the process similar to hooking up logic gates and bistables.

I prefer to write out a series of logic equations.

I didn't find any of these approaches difficult to learn – nothing remotely as difficult as mastering a computer language, where it took me a week to learn Fortran 4, and nearly a year before I could produce a well formatted page of output (including printed graphs). The MACRO-8 assembly language for the PDP-8 was much easier – mainly because the teletype output made more sense.

There is a certain learning curve in learning how to use Boolean logic, and you need extra time learn how to hook up CMOS parts to realise that logic – most people would be better off putting the extra time into learning how to program programmable logic.

The choice of which programmable part should probably be referred to comp.arch.fpga, but one thing is for sure – you will be able get all the logic into a single modern PLD.

—

Bill Sloman, Nijmegen (but in Sydney at the moment)

.