

Re: EEPROM checksum error

Source: <http://sci.tech-archive.net/Archive/sci.electronics.basics/2005-03/0486.html>

From: Robert Monsen (*rcsurname_at_comcast.net*)

Date: 03/09/05

Date: Wed, 09 Mar 2005 12:44:58 -0800

Charles Edmondson wrote:

- >
- > *For you, this might be a design issue. As power ramps up/ramps down,*
- > *different components react differently. Some have internal caps that*
- > *make them hold state a little longer than others, or are just more*
- > *sensitive to power supply levels. Think about the programming sequence.*
- > *What could provide it in your circuit. What could PREVENT it in your*
- > *circuit!*
- >
- > *I have often found that start up conditions are not fully considered in*
- > *design. You just assume that the power comes up all at once, smoothly.*
- > *In reality, different voltage rails come up differently. Filter caps*
- > *take time to charge up to voltage. Good design takes that into account,*
- > *sometimes adding POR circuits to make sure that power is steady before*
- > *starting things up, and quick shut down sequences to turn everything off*
- > *before the power goes below limits. It's like preventing race*
- > *conditions and logic glitches. Sometimes, you just have to take a good*
- > *look at the failure modes...*
- >

Typically, parallel eeproms have active low write enable and chip select pins. If those pins are slow to come up, I guess it can cause problems.

However, one typical example is the atmel parallel eeproms. They require an 0xAA, followed by an 0x55 to be written to special addresses before they'll go into write enabled mode. The address pins need to be set to 1555 for the 0xAA, and 0xAAA for the 0x55. Getting this to happen because of a startup seems incredibly far-fetched.

There is also a set of hardware features that protect against inadvertent writes. The AT28BV64B has a power on delay of 10ms after Vcc comes up; thus, these glitches would have to be at least 10ms. If OE is low, writes are inhibited; thus, the glitch would have to be restricted to the WR and CE pins. If either CE or WE are high, writes are inhibited; thus, the glitch would have to be on both of these, but not on OE. Also, pulses of less than 15ns on either WE or CE won't initiate a write cycle. After this dance, one has to go through the software programming sequence to get it to really go into write mode.

Again, startup writes to these things sounds incredibly far-fetched. Perhaps after trillions of power cycles inducing random noise (1,000,000 monkeys?).

I'm guessing your example eeprom wasn't one of these, and was protected in some other way. Either that, or the chip firmware was flawed, and the sequence above could be circumvented in some situations.

The OP hasn't indicated what EEPROM he is using, or what kind of usage (whether it's getting programmed in the field or not). Thus, we are all speculating without any real information.

--

Regards,

Robert Monsen

"Your Highness, I have no need of this hypothesis."

- Pierre Laplace (1749-1827), to Napoleon,
on why his works on celestial mechanics make no mention of God.