Designing for Hostile Environments

Author: Monte Dalrymple

Presented by: Kristen L. Reband

Overview

I. Single-event upset (SEU)

II. Triple-modular redundancy (TMR)

III. Hamming error-check bits

IV. Hamming Error-Check vs. TMR

V. Unused States

VI. Verifying the design

The Problem

- The basic problem goes by many names, one of them is single-event upset (SEU).
- SEU is the idea that the effect of an error starts on a single circuit node.
- Without this simplifying assumption, harding a design against errors is almost impossible
- Even though an error may propagate throughout the design via the normal circuit paths, the root of the error can be traced to a single node.

First Level of Defence

The simplest step in the defense against an SEU involves making the flip-flops impervious to upsets.

A common way of accomplishing this involves triple-modular redundancy (TMR)

Basically each individual latch in a flip-flop is replaced by three latches.

EPENSIVE!!!

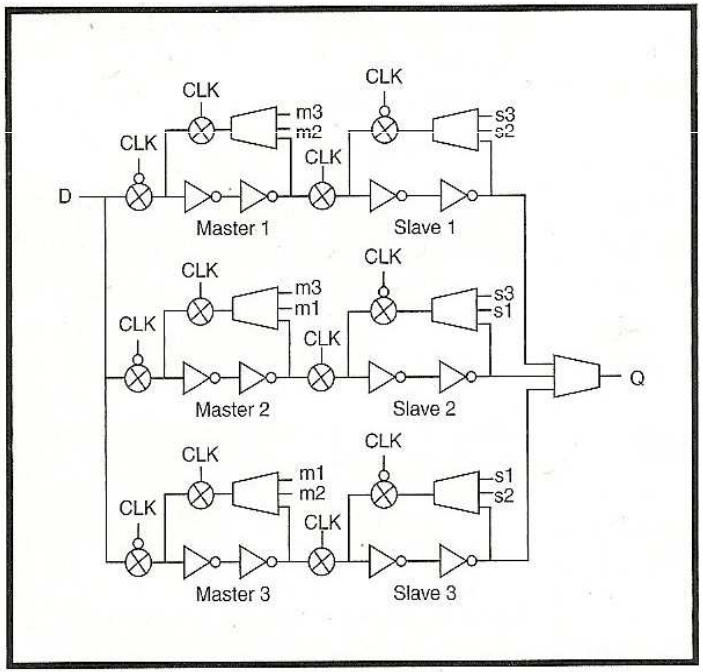

Figure 2—Hardening the flip-flop with triple-modular redundancy makes each flip-flop much more complex.

The Thorny Problem

- Occurs when an SEU happens when there is a clock edge.
- If a flip-flop samples an incorrect value, none of the TMR circuitry inside the flip-flop is going to correct the error.
- The more hostile the environment the more likely it is.
- One technique that can be used to handle errors that might make it through the TMR protection is to use a single-error correction of flip-flops.
- One of the most robust ways to do this is to add Hamming error-check bits to every collection of flip-flops.

Hamming Error-Check

Hamming error coding technique is a modified parity check approach in which m different groups of data bit positions are formed and each such group has a parity number (0 or 1) assigned to that group.

The data bit groups are chosen so that:

- Ø each group has approximately the same number of bits therein
- Ø for any two groups, a first group has at least two bit positions that differ from all the bit positions of a second group.

Each Hamming error check bit then becomes a parity error check for the bit values in one of these groups.

A knowledge of the group or groups where a parity error occurs allows the position of the single bit error to be determined directly.

Hamming Error-Check vs. TMR

- TMR is equivalent to a Hamming error-check code for 1 bit of data.
- Why they are not usually used together.
- Hamming error-check is one of the most robust single-error-correcting (SEC) codes.
  - Able to handle SEU that happen when there is an clock edge.
- TMR protects each group of flip-flop individually.
- So TMR it is more robust overall.

Unused States

- Most designers never paid much attention to the unused states in state machines.
- Rarely bother to make sure the unused states will not persist forever.
- During the design process, it is very handy to make the default states always to unknown in a simulation because it is an easy way to quickly fine design errors.
- But once the design is mostly debugged, it doesn't hurt to put in a defined exit path for every state into the design.

Will It Work?

- Verify that the design works properly in the absence of an upset. This requires a test suite that exercises:
  - Every instruction
  - Every flag combination
  - Enough data combinations to verify all of the logic paths.
- Once completed author used Verilog to made it easy to inject upsets into design by forcing the state of a flip-flop to toggle.
- Can add more complex conditions to the creation of the trigging signal
  - Wait for a certain amount of time
  - Certain instruction

Resources

R. W. Hamming, “Error Detecting and Error Correction Codes,” *Bell System Technical Journal,* Vol 29, 1950.

National Aeronautics and Space Administration, “Juno mission overview,” 2007 http://newfrontiers.nasa.gov/missions_juno.html

Answers Corp., “Radiation hardening,” 2007 www.answers.com/topic/radiationhardening

Patent Storm, “Validation of RAM-resident software programs,” 1994 http://www.patentstorm.us/patents/5357527-description.html