

# Parity errors seen on ASR9k

## Contents

[Introduction](#)

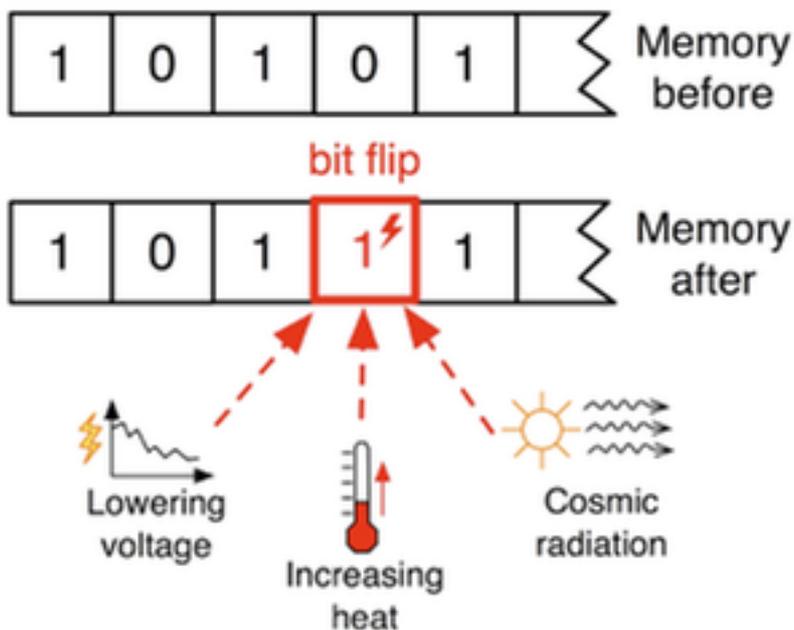
[Problem](#)

[Solution](#)

[NP Soft Error Handling Improvements](#)

## Introduction

A parity error is a bit flip in memory. In electronics and computing, electrical or magnetic interference from internal or external sources can cause a single bit or memory to spontaneously flip to the opposite state. This event makes the original data bits invalid and is known as a parity error.



Typically these errors fall into two different types of errors, soft and hard.

**Soft parity errors**, these events are transient and random. They usually will only be seen once in a particular bank of memory.

**Hard parity errors**, are caused by a physical malfunction of the memory hardware or by the circuitry used to read and write memory cells. These are usually seen repeatedly and require replacement.

Most parity errors are caused by electrostatic or magnetic-related environmental conditions. The majority of single-event errors in memory chips are caused by: **background radiation** (such as neutrons from cosmic rays, nuclear facilities), **electromagnetic interference (EMI)**, and **electrostatic discharge**. These events may randomly change the electrical state of one or more memory cells or may interfere with the circuitry used to read and write memory cells.

## Problem

Parity errors are a fact of life when it comes to high density memory as is used in the ASR9k linecards. So how we handle them is really all we can have control over. Some ASR9k (xmen/typhoon) linecards, under rare conditions, may encounter layer 1 cache errors. These show up as a kernel panic in data cache or instruction cache (**DCPERR** or **ICPERR**). Another observed error is in the various memory banks used by the NPs (network processors) on the linecards. These usually are seen starting with the following types of error logs:

%PLATFORM-NP-0-NON\_RECOVERABLE\_SOFT\_ERROR

%PLATFORM-NP-3-ECC

%PLATFORM-PFM-0-CARD\_RESET\_REQ

The problem here is the DCPERR/ICPERR result in a full linecard reload. The same was true for the vast majority of the various NP memory banks as well. This is obviously not ideal as most linecards have multiple NPs. Why affect all NPs on the linecard if only 1 NP has an issue.

## Solution

For the DCPERR and ICPERR errors seen on the typhoon LC CPU caches, we have a solution to avoid the need to panic and reload the linecard. This is done with [CSCux30405](#) . Currently integrated in version 5.3.3 and above.

For the NP memories this gets a lot more complicated. There has been a great effort to scrub the various memories to see which we can safely ignore or come with a less impacting way to recover. The majority of which have been integrated into 5.3.3 and above and there have been umbrella SMUs built on the majority of the popular releases.

*Note: This has also caused a collateral of [CSCvc69282](#) where we may see a kernel crash due to continued interrupts.*

## NP Soft Error Handling Improvements

Over the last half of 2015 and early 2016, numerous improvements were made to NP soft error handling for both Typhoon and Tomahawk. Handling for many different memories was converted from a method which required a linecard reload to something more graceful such as repairing the error in memory or performing an NP fast reset. Handling for errors which do not have a functional impact but which can not be cleared ("sticky") was also improved so that the errors would not longer continue to recur. In addition, several bugs were fixed, especially for errors which occur in NP instruction memory or the internal TCAM. Approximately 80-90% of previously non-recoverable errors are now recoverable and do not require a linecard reload.

All of these improvements and fixes are integrated in the 5.3.3 releases and above. The fixes are also available in umbrella SMUs for all major maintenance releases:

434 - [CSCux16975](#)

512 - [CSCux44633](#)

513 - [CSCux16975](#)

531 - [CSCux34531](#)

