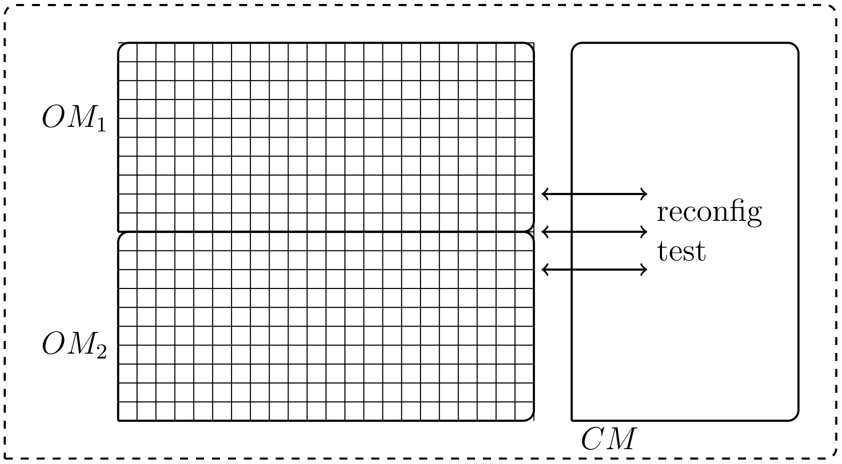
Fault Tolerant Circuits Using Evolutionary Repair Mechanisms

The issue of digital circuit degradation is certainly an important one; as devices become more complex and expensive, the cost of circuit failure is very high. A single "stuck-on" fault can render an entire complex system inoperable. To date, existing methods focus on *detecting* rather than *repairing* these faults. This document proposes a method and framework of fault identification and repair in consumer-oriented markets using a hardware evolution approach. This system results in highly robust and reliable digital circuitry. The targeted platform for initial proof of concept (PoC) demonstrations is Field Programmable Gate Array (FPGA) devices.

Previous research investigating evolutionary self healing circuitry has fallen short on several fronts. One approach demonstrated that an evolutionary approach was feasible, but the methods used are impractical in field use (Salvador 2011). Their method was limited to the processing element (PE) level rather than the Configurable Logic Block (CLB) level and the self repair was limited to offline use. That is, during reconfiguration, the device was inoperable. Other work displayed promise, but was only demonstrated though simulation, and no furthering of their work was made (Lohn 2003).

With this in mind, the proposed method of circuit repair addresses many of these points, and even promises other advantages. First, the proposed method performs *online reconfiguration*, that is, the device is completely operable during partial reconfiguration. To the user, no interruption or degradation of service is detectable. Our method also utilizes intrinsic hardware evolution, where physical tests are run on the hardware block being reconfigured. This has the added benefit of significant speed and accuracy gains over software simulation (extrinsic evolution). The proposed method operates well for Single Event Upset (SEU) errors through partial FPGA scrubbing (Heiner 2009), as well as permanent/cumulative faults.



The basic design is presented in Figure 1. The system comprises of three components located on-chip: the two Operation Modules OM_1, OM_2, and the Configuration Module CM. The OMs implement the actual function of the chip, for example, an image filter, with only one of two modules operating at any given time. The CM continually runs test vectors through the running module, ensuring correct operation. When an incorrect result is returned, the CM re-routes input/output access to the second OM, minimizing external exposure to faulty operation. The CM then attempts to diagnose the fault to a single location. An evolutionary algorithm run on the CM then designs an alternate design at the CLB level, avoiding use of the damaged area. Using the existing design as a starting point, the evolutionary algorithm tests possible configurations on the damaged operation module, until a potential solution is found. At this point, the workaround configuration is applied to the faulty OM, and is re-activated. The second OM is turned back off to save resources. In practice, both the OMs and the CM can be placed on the same FPGA board, with the CM implemented as a soft processor (e.g. Xilinx MicroBlaze). Alternately, the CM can be implemented as a separate hard processor.