AI based Media Functional Safety and Reliability verification in Safety-Critical Autonomous Systems

Suresh Vasu *Palanivel Guruvareddiar*

suresh.vasu@intel.com *palanivel.guruvareddiar@intel.com*

***Abstract*— Thanks to the advancements in processor compute, connectivity, and memory technology, the demand for video processing silicon is on the rise over the past few years. In addition to the traditional video entertainment applications, the explosive growth in AI technology enabled several machine vision applications as well. Today's Systems on Chips (SoCs) catering to these various multimedia applications are a complex chain of hardware with significant memory requirements. The importance of the functional Safety and reliability verification in the design is very critical and it has the possibility for catastrophic issues for the end users if it's not verified properly. The impact of soft errors in the silicon especially during the memory transactions to the end user experience and to the inference accuracy of the machine vision algorithm is largely an unexplored area. Our paper tries to bridge this gap by outlining a novel methodology and a unique verification framework that provides the ability to inject soft errors and study the impact of those errors on both video/image quality as well as on the inference accuracy. Using this verification framework, we studied the impact of soft errors in memory for the JPEG encoder hardware and expanded the study to include video encode hardware, where the encoded video streams will be used for both human consumption as well as for machine vision. For human consumption, the framework computes both objective video quality using standard metrics as well as subjective quality metrics using VMAF (Video Multi-Method Assessment Fusion). For machine vision, the framework executes AI workloads including object detection, tracking and classification and compute various metrics such as mean average precision, multi-object tracker accuracy as well as classification accuracy. The proposed verification methodology uses the power of SystemC and OpenVINO which provides a novelty in functional safety and reliability Verification.**

Keywords— Video, Encoder, Decoder, Soft Errors, Inference, Fault Injection

I. INTRODUCTION

Depending on the end application of an IC, the hardware design needs to conform to various levels of robustness that ensure functional safety. Standards like ASIL defined as per the ISO 26262 specification is one example of calibrating functional safety of a system [6]. While several parameters may affect system level robustness, this paper will discuss one aspect of reliability that affects Silicon hardware design – namely soft errors. Soft errors are transient errors that occur due to high energy particles that hit the logic or memory causing single or multiple bit/event upsets (SEU/MBU). These are different from permanent faults or hard errors which are recoverable.

Not all soft errors however result in functional errors. Even if the hardware has no detection or correction mechanism available, only a portion of all the soft errors will result in SDC (Silent Data Corruption) and the rest will not impact the output of the system [5]. It is this behavior that we study in the video hardware pipeline by measuring SDC versus benign errors in the Video output and its effect on downstream applications.

Quantifying the FIT (Failure In Time) rate of a SoC (System On Chip) is an important data point in determining the reliability of the system. The FIT rate is dependent on several factors like the microarchitecture, circuit behavior, design, workload, and time spent in use condition. The parameter determined by the microarchitecture, circuit components and workload in the system is commonly referred to as AVF (Architectural Vulnerability Factor) [1]. Logic structures in the design such as memory arrays, flops, latches, buffers, and registers may have different AVF derating factors. The sum of error rates of different components with appropriate AVF scaling, determines the error tolerance of an application.

The hardware under discussion here, is multimedia hardware which includes video encoders and decoders. This hardware typically has specialized logic with dense memory to be used as frame buffers at various stage din the pipeline. Memory arrays by construct have been shown to be more susceptible to bit flips resulting from soft errors Hence the study focuses for new on memory corruption within video hardware. The outputs from these hardware blocks can be further processed by consumers such as CPU cores / AI (Artificial Intelligence) engine / Media applications.

Object detection and classification using neural networks is one such use of inference Images/Videos output. This is the application chosen to study the downstream impact of soft error related failures in Video hardware [2].

The simple, popular, and accurate technique of RTL simulation is used inject faults into the memory arrays within media hardware as described in the next section. Video quality in terms of precision and recall values in popular object tracking neural networks is compared between error injected video outputs from encoder hardware and the ground truth value. The robust nature of the described verification framework allows us to experiment with a variety of multimedia hardware, inference workloads and NN models. The results from this study can be used to recommend appropriate area optimized error detecting mechanisms based on memory profiling and error susceptibility. This allows system designers to achieve the designated SoC FIT (Failure In Time) rate while still not wasting Silicon area on full blown error detecting or error correcting hardware.

II. VERIFICATION CHALLENGES

A. *Verification Challenges*

Figure 1 shows the percentage of total IC/ASIC project time spent in verification [3]. From the case study and analysis done by the team from Mentor Graphics, the complexity for doing the verification has increased significantly for the past few years and the design size has also grown many folds. Due to many IPs getting integrated in the SOC, verification engineers are facing lots of challenges in the day-to-day debugging.

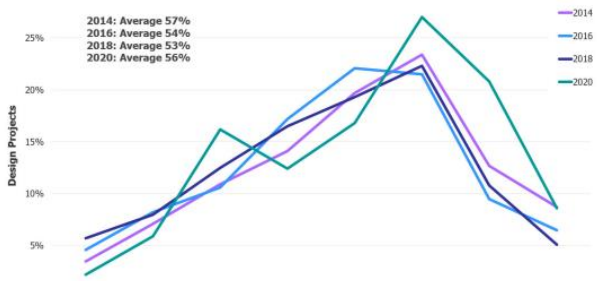

Figure 1. Percentage of IC/ASIC Project Time Spent in Verification [1]

Figure 2 shows where verification engineers spend their time (on average). [3]. The recent study shows that the verification engineer is spending more time in debugging during the chip development compared to all other verification tasks like test planning, testbench development, creating test cases, running simulation and support works. From the verification and validation engineer's experience, one of the aspects is for more debug time is due to not able to create the test content for Post Silicon Validation and no infrastructure is available to make the test content flexible.

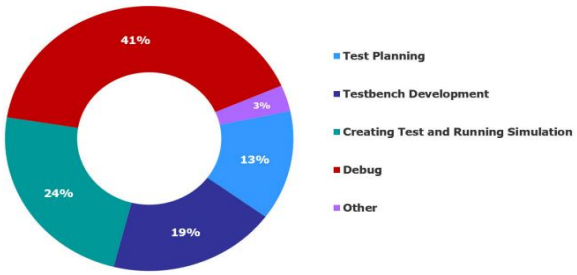

Figure 2. Where IC/ASIC Verification Engineers Spend Their Time [1]

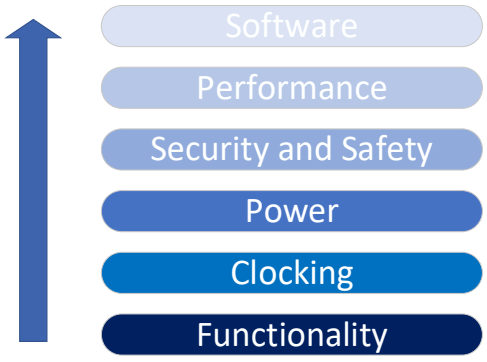

Figure 3. Verification Complexity Increase Trend

Figure 3 shows that the verification complexity increases from functionality to software. Security and Safety is one of the most critical verifications which makes or breaks the product in the competitive space.

III. SYSTEMC AND OPENVINO

SystemC [4] is a C++ class library and enables design and simulation of hardware descriptions in a software environment. SystemC allows designers to design and verify at all levels of abstractions in a common language. SystemC hardware model within the software environment. The model can, at any time, be verified for correctness in the software environment which often reduces simulation times. The higher level of abstraction introduced also promotes reuse. Currently SystemC offers an effective verification platform for high-level functional models because it introduces timing in a software environment.

Figure 4 shows the traditional way of using the SystemC model for the RTL verification. The testbench components like scoreboards, checkers interact with the SystemC model using the TLM (Transaction Level Modeling) ports to pass the configurations and pull the encoded data output. Once the data is collected, it gets compared with the RTL output to verify the data integrity. Based on the verification methodology, the SystemC model can be used as pixel-by-pixel comparison or frame by frame comparison.

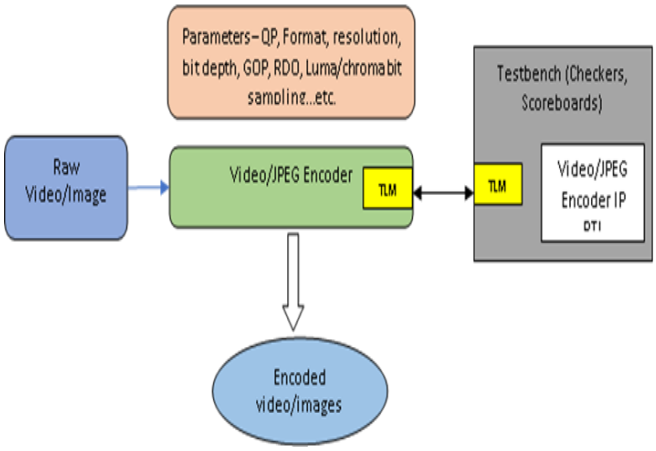

Figure 4. Traditional way of using SystemC Model for Verification

IV. SIMULATION AND ANALYSIS SETUP

A. *Design Under Test, Test Bench and Fault injection*

Statistical Fault injection is a well-known technique used for studying effects of errors in the system [9]. The design under test here is synthesizable RTL model of a Video / JPEG Encoder block. This module takes the raw video/images in the YUV format as input and encoders them as AVC/HEVC format as in case of Video, JPEG format in case of images as the output. In a real-world use case scenario, these Video/Image outputs are stored for later use by other applications. Memories in the form of SRAM arrays or register files within the RTL module is typical of any specialized hardware. These memories are also modeled in the DUT (Design Under test) as behavioral models, thus allowing bit flips to be injected at desired locations within the memory array.

The simulation and test environment has been built to randomize multiple parameters such as memory location, start time of fault deposit, fault persistence time and bit flip nature (0 or 1). This paper focuses on transient memory faults in the data path of the encoder hardware rather than control logic. This is because control path failures are more catastrophic and unrecoverable in nature thus eliminating any need for quality analysis. Datapath errors on the other hand may or may not cause a functional failure (SDC) and quality analysis is necessary to determine impact of these errors [11][12].

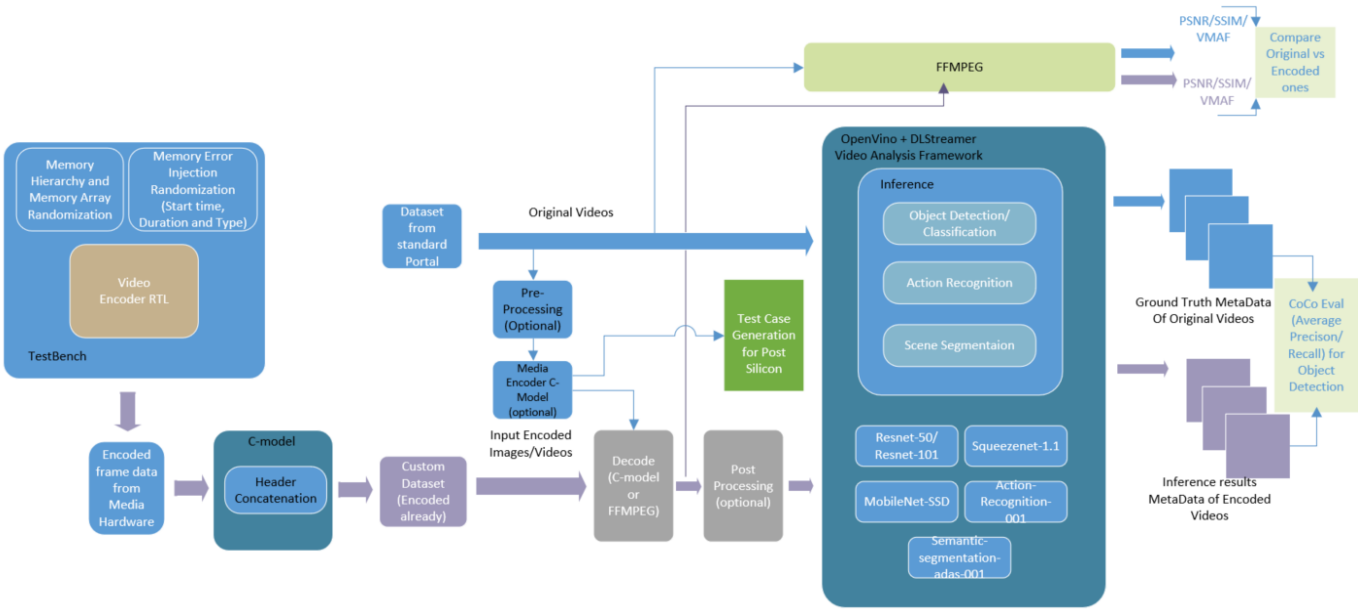

Figure 5. Simulation and analysis framework for media fault injection

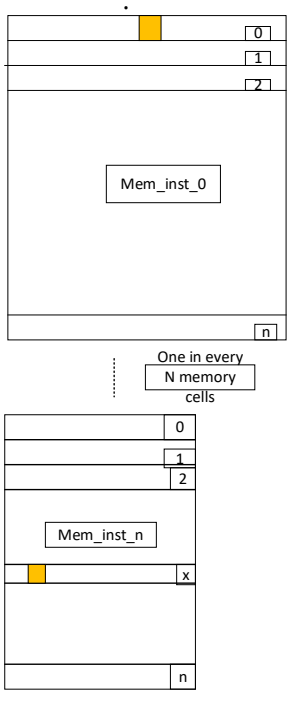

Fig 6. Memory cell impact by soft error

B. Media Analysis Framework and Verification Methodology

For the JPEG/Video analysis, the framework is using the OpenVino + DL Streamer [15], the classification analysis is done by NN models ie. Resnet-50 and the object detection accuracy analysis is done by the Coco Eval tool. OpenVino is an open-source toolkit for optimizing and deploying the AI inference. DL Streamer is streaming media analytics framework, based on Gstreamer multimedia framework, for creating complex media analytics pipelines. Using the DL Streamer, the required pipeline is created, the videos/images are processed using the same pipeline and metadata is collected.
For Video based object detection COCO Eval based methodology used for objection detection analysis [17]. The Coco format is a specific JSON structure dictating how labels and metadata are saved for an image dataset. In the process, the metadata is converted into the Coco format and the Coco Eval tool is used to the analyze the same; Coco eval tool provides the Average Mean Precision and Average Mean Recall values comparing the Ground Truth Metadata. F1 score is a weighted average of the precision and recall. Values range from 0 to 1, where 1 means highest accuracy. Coco eval tool uses the metadata details generated from original video and compares with the metadata generated from videos generated post fault injection in memories.

For JPEG images-based classification, Nerural Network models like Resnet-50 is used in which we will get the classification score when comparing with original image and images generated post fault injection in memories.

The sample DL streamer pipeline is show below:

Fig 7. DL streamer pipeline

The sample metadata is shown below:


```
[{
 "objects": [
  {
   "classification_layer_name:prob": {
    "confidence": 0.09943553805351257,
    "label": "assault rifle",
    "label_id": 413,
    "model": {
     "name": "MOBILENET_V2"
    }
   },
   "detection": {
    "bounding_box": {
     "x_max": 0.6390969395301087,
     "x_min": 0.61303097459388,
     "y_max": 0.7760099530452977,
     "y_min": 0.7403974098229809
    },
    "confidence": 0.9960777759552002,
    "label": "car",
    "label_id": 2
   },
   "h": 38,
   "roi_type": "car",
   "w": 50,
   "x": 1177,
   "y": 800
  },
```


Fig 8. Sample metadata extraction

F1 score is calculated by the below:

Precision = TP/ (TP + FP) whereas TP – True Positive, FP – False Positive

Recall – TP/ (TP + FN)

F1 Score = (2 x (Precision x Recall)) / (Precision + Recall)

For the study about the effect of the soft errors on the videos, Yolo-V3 Neural Network model is used for the Object Detection.

V. RESULTS

In this section we present the image and video data used in this study and the results from the simulation and object detection. Errors were introduced in randomly selected frames of a video/images, and regardless of functional failure, taken through the quality analysis framework. The results are summarized below in the form of both numerical scores and visually observable artifacts in the resultant output video/images.

A. JPEG

Four different input images were chosen from popular image data sets. Two key results are presented as graphs – The first graph talks about error % against fault persistence time – this proves that the lower the error persistence (with bit flip to 0 and bit flip to 1), lower is rate of failure. The next two graphs show a scatter plot of simulation results across memory instances and fault persistence time periods. This gives us an insight on which memory instances in the design are more tolerant to errors than others.

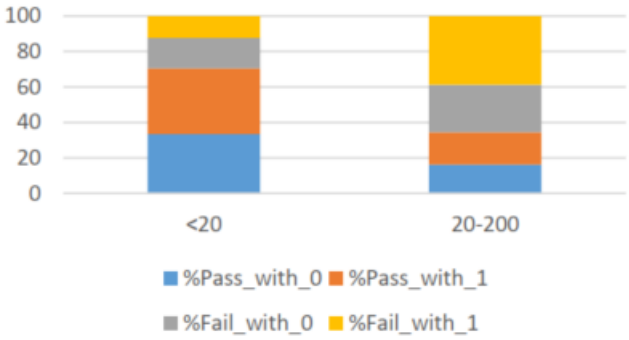

Fig 9. % sims vs fault persistence time (us)

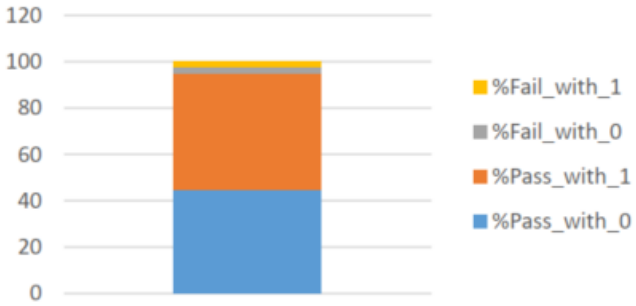

Fig 10. Transient soft error profile with less persistence

The graph shown in Figures 11 and 12 show that as fault persistence time increases, so does the SDC within the media hardware pipeline. Error rate also seems to increase when the memory bit cells flip to a logic high or ‘1’ instead of ‘0’.

Fig 11. Scatter Plot – pass and fail fault persistence for each memory instance.

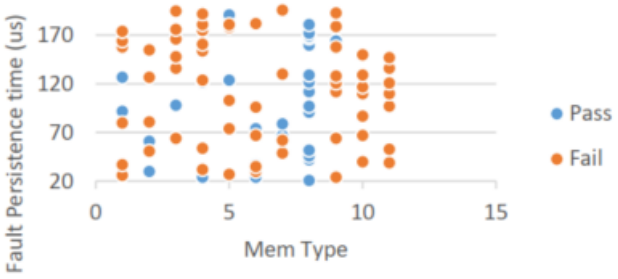

Fig 12. Pass and fail fault persistence time for each memory instance.

Three resultant images out of about 1300 simulations were wrongly classified by the analysis framework which uses the Resnet-50 NN model for image classification. In the figure 7 below, the inference framework has wrongly classified the image of a cat as a tiger due to soft error SDC simulated. The wrong inference occurs due to a high mismatch with the ground truth classification score. This is indicated by the outlier classification score as shown in Figure 13 which significantly higher the ground truth marked in blue.

Fig 13. Ground truth image Vs Wrongly Inferred image

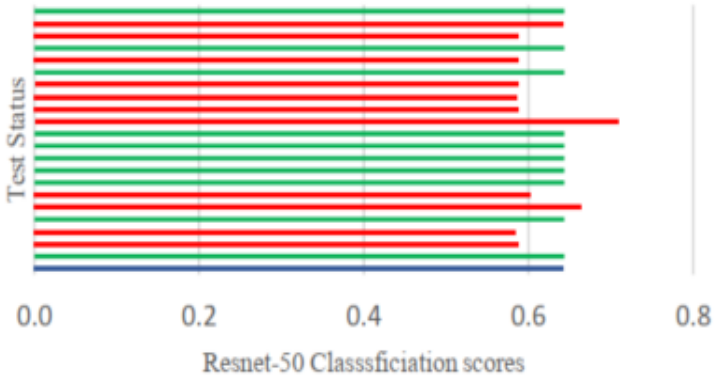

Fig 14. Classification scores Vs Test Status

B. HEVC/AVC

Fig 15. Crowd run ground truth Video.

Fig 16. Corrupted Video.

The average recall and the precision data by comparing the original crowdrun video ground truth is shown in shown below in the Fig 17.

```
[INFO] Object detection metrics of  car_toll_original.json
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.994
 Average Precision  (AP) @[ IoU=0.50       | area=   all | maxDets=100 ] = 0.994
 Average Precision  (AP) @[ IoU=0.75       | area=   all | maxDets=100 ] = 0.994
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 1.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 1.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.947
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.679
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.814
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.997
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 1.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 1.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.972
```

Fig 17. Snapshot of Precision and recall values from analysis framework.

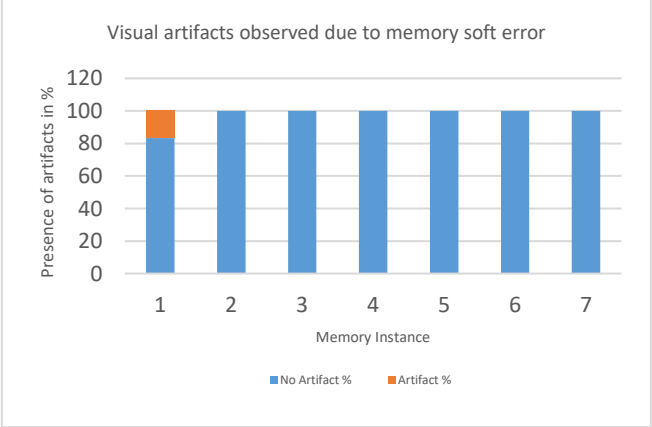

Fig 18. Graphical representation of error susceptibility for visual artifacts in video encoder memories

Results of F1-score are below, and it shows that approximately 5% accuracy reduction due to the soft error in one of the Media Workloads.

Recall	Precision	F1-score
0.972	0.947	0.95933715

SystemC + OpenVINO based Media Functional Safety and Reliability validation has cut down the specific content creation as compared to the time taken in the previous projects internally. Upfront it helped to use the customer workloads in to do the verification in the IP level environment. Also, the automation based on the Media Encoder System C-models + OpenVINO has helped to use the Neural network models to understand the features impacted due to the specific memories.

VI. CONCLUSION AND FUTURE WORK

Fault injection is one of the most accurate ways to model errors and understand the error tolerance or reliability of hardware. It is also a common practice to adopt other architectural models for hardware rather than RTL models. This way, simulation time for multiple bits in the design can be greatly reduced, while still using the analysis framework. A selective error detect / correct structure for memories can be proposed based on the results. Those memory instances or categories that are found to be most susceptible to errors can be chosen for error detection / correction. This decreases overall area overhead sue to these schemes.

SystemC + OpenVINO based NN models for video functional safety validation is an effective methodology which will help in day-to-day validation activities, reduce the manual efforts of test content creation, and save significant execution time due to the automation. This methodology will help during the design development stage to prepare all the test content, verify in IP level simulation, and use the power of an AI tool i.e OPENVINO.

REFERENCES

[1] SS. Mukherjee, J. Emer, and S. K. Reinhardt, "The Soft error problem: An architectural perspective," *11th International Symposium on High-Performance Computer Architecture*. IEEE, 2005

[2] Li, Guanpeng, et al. "Understanding error propagation in deep learning neural network (DNN) accelerators and applications." *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 2017.

[3] *https://uobdv.github.io/Design-Verification/WilsonResearchGroupFunctionalVerificationStudy/2020-WRGFV-Study/ic-asic-trend-report_2020-wilson-research-verification-study_hfoster.pdf*

[4] https://www.accellera.org/downloads/standards/systemc

[5] https://www.intel.com/content/www/us/en/developer/tools/openvino-toolkit/overview.html

[6] https://media.xiph.org/video/derf/

[7] Thomas, Anna, and Karthik Pattabiraman. "Error detector placement for soft computation." *2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2013.

[8] Breuer, Melvin A. "Multi-media applications and imprecise computation." *8th Euromicro Conference on Digital System Design (DSD'05)*. IEEE, 2005.

[9] Sharif, Uzair, Daniel Mueller-Gritschneder, and Ulf Schlichtmann. "Investigating the Inherent Soft Error Resilience of Embedded Applications by Full-System Simulation." *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2020.

[10] J. Athavale, A. Baldovin, R. Graefe, M. Paulitsch and R. Rosales, "AI and Reliability Trends in Safety-Critical Autonomous Systems on Ground and Air," *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2020

[11] Andreas Herkersdorf, Hananeh Aliee, Michael Engel, Michael Glaß, Christina Gimmler-Dumont, Jörg Henkel, "Resilience Articulation Point (RAP): Cross-layer dependability modeling for nanometer system-on-chip resilience", Microelectronics Reliability, 2014.

[12] V. B. Kleeberger *et al.*, "A Cross-Layer Technology-Based Study of How Memory Errors Impact System Resilience," in *IEEE Micro*, vol. 33, no. 4, pp. 46-55, July-Aug. 2013.

[13] https://towardsdatascience.com/on-object-detection-metrics-with-worked-example-216f173ed31e

[14] https://pro.arcgis.com/en/pro-app/latest/tool-reference/image-analyst/how-compute-accuracy-for-object-detection-works.htm

[15] https://www.intel.com/content/www/us/en/developer/videos/ready-steady-stream-openvino-toolkit-dl-streamer.html

[16] https://debuggercafe.com/evaluation-metrics-for-object-detection/