

Markov Modeling of Redundant System on Chip (SoC) Systems

Alexandra Loumidis^a, Todd Paulos, Ph.D.^b, Andrew Ho^c, and Douglas Sheldon, Ph.D.^d

^aHarvey Mudd College, Claremont, CA, USA, aloumidis@g.hmc.edu

^bJet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA,
todd.paulos@jpl.nasa.gov

^cJet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA,
andrew.h.ho@jpl.nasa.gov

^dJet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA,
douglas.j.sheldon@jpl.nasa.gov

Abstract: The increasing cost and decreasing availability of space-rated custom System on Chip (SoC) components has led to interest in using commercial components from terrestrial industries in space environments. Along with this interest, there exists a need to understand how the reliability of the chips, including common cause upsets, can impact the probability of mission success and risk.

This work models the failure and recovery of a system consisting of two Qualcomm Snapdragon processors with five upset types each. Four Markov models were created, modeling both recoverable and non-recoverable systems. Models 1 through 3 assume the system is recoverable while Model 4 accounts for a non-recoverable system. Model 1 assumes the rate of recovering both upset processors is the same as the rate of recovering one upset processor. Model 2 assumes that the processors can recover one at a time at two different recovery rates. Model 3 assumes that the boot-up time of the second processor is greater than the recovery time for a single processor.

MATLAB scripts were produced to plot availability of each model over time. The three system recoverable models achieved availability of greater than 0.970 after 10^6 seconds while the system non-recoverable model achieved availability of 0.344 after the same time period.

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Keywords: Markov, redundant, Snapdragon.

1. INTRODUCTION

Custom System on Chip (SoC) components, such as the RAD750, have been used on flight projects at the Jet Propulsion Laboratory due to their radiation-hardened properties and known reliability. However, these custom components are becoming both more expensive and harder to find, leading some teams to consider the use of commercially available, off-the-shelf components.

The Qualcomm Snapdragon 801 SoC is a commercially available processor that was successfully implemented in the Ingenuity Helicopter on Mars as part of the Guidance, Navigation and Control (GN&C) system. The processor provides visual inertial odometry, telemetry generation and data management functions.

Despite the ease of availability, commercial semiconductors like the Snapdragon 801 are known to experience a variety of transient events induced by the space radiation environment, raising concerns regarding their susceptibility to environmentally induced upsets and permanent failures. Methodologies

that are able to properly and accurately describe the impact these events can have on system operation are required as part of the overall risk management process.

This work seeks to explore the impact physical architecture has on possible mitigation of transient radiation induced upsets on a collection of Qualcomm Snapdragon SoCs and the additional support devices required for its operation. This involves modeling the upset behavior of two Qualcomm Snapdragon processors. Each processor has five different sub-components, each of which are susceptible to upsets that can cause the processor to be unavailable. The sub-components are Double Data Rate memory (DDR), Large File Storage memory (LFS), Ferroelectric Random Access Memory (FRAM), Power Management Integrated Circuit (PMIC), and System on Chip (SOC). As the two processors will be used together, the system is modeled using three states: a state where both processors are available, a state where one of the two processors is available and the other is unavailable, and a state where both processors are unavailable.

The chosen model type in this work is a Markov model. The main assumption of a Markov model is that future states depend only on the current state, not on the events that occurred before. Thus, the model is often called “memoryless” and assumes the Markov property. Markov models can be classified according to whether they are fully observable or partially observable, autonomous or controlled. A fully observable and autonomous Markov model is called a Markov chain. A Markov chain was chosen largely because of the model’s simplicity, and due to its applicability to this type of modeling.

2. METHODS

The modeled system consists of two Qualcomm Snapdragon processors, each of which is susceptible to five different types of upsets. Four models were created to represent different configurations of the system. Two of the four models were fully created in the commercially available BlockSim tool, while all models were implemented in MATLAB scripts. BlockSim models were created by individually laying out and labeling states and the transitions between them. MATLAB models, however, were created more easily, allowed for quicker changes to models, batch processing, and yielded the same results as the BlockSim models.

3.1. MATLAB

MATLAB models were created in a script initially made to reproduce and verify the model seen in McMurtrey [1]. The modified code follows a format of creating transition matrices corresponding to each model and then using matrix multiplication to calculate and plot the state over a time interval. If P is a vector of probabilities, $P[0]$ is the initial state, and T is a square transition matrix where the dimension is equal to the number of states in the system, then after n steps the probability vector can be calculated as,

$$P[n] = T^n P[0] \quad (1)$$

Transition matrices corresponding to small models were constructed by inspection to serve as starting points for larger models. Transition matrices for large models were then made in MATLAB by applying patterns seen in the small models. Creating a new model in MATLAB did not require writing a new script, it only required creating the new model’s transition matrix.

Two MATLAB scripts were developed to visualize the availability of the created models. The first script plots the availability of select states based on model number and chosen parameters, also returning the availability of the states at the last step of simulation. A second script allows users to simulate several sets of parameters for one model at once. It takes in an array of parameters and returns a data table where each row is a run of the simulation and columns show the parameters used and state availability at the last step of simulation.

Two additional MATLAB scripts were written to examine the parameter sensitivity of a state's availability. One script calculates the availability of a state over time and repeats this calculation with one parameter varied. Each calculation is then overlaid on a single availability plot. A second script calculates the availability of a state in the last step of simulation only. This calculation is repeated with two parameters varied independently. The script then produces a contour plot with one varied parameter on the x-axis, the other varied parameter on the y-axis, and availability plotted as a series of curves, each labeled with an availability value.

3.2. Models

3.2.1 Notation

For systems with two processors, states in BlockSim and MATLAB were labeled according to a three-character notation, where the first character indicates the number of available processors, the second character indicates the first processor's upset type and the third character indicates the second processor's upset type. For systems with one processor, states were labelled according to a two-character notation, where the first character indicates the number of available states and the second character indicates the processor's upset type. The upset types are listed in Table 1a. An example of labeled states is in Table 1b.

Table 1a: Letter corresponding to upset component

Upset Component	Letter
No Upset	O
Memory DDR	D
Memory LFS	U
Memory FRAM	F
PMIC	P
SOC	S

Table 1b: Examples of state notations and their meanings

State	Number of Available States	First Upset Type Letter	Second Upset Type Letter
2OO	2	O	O
1DO	1	D	O
0DU	0	D	U
1O	1	O	not applicable
0P	0	P	not applicable

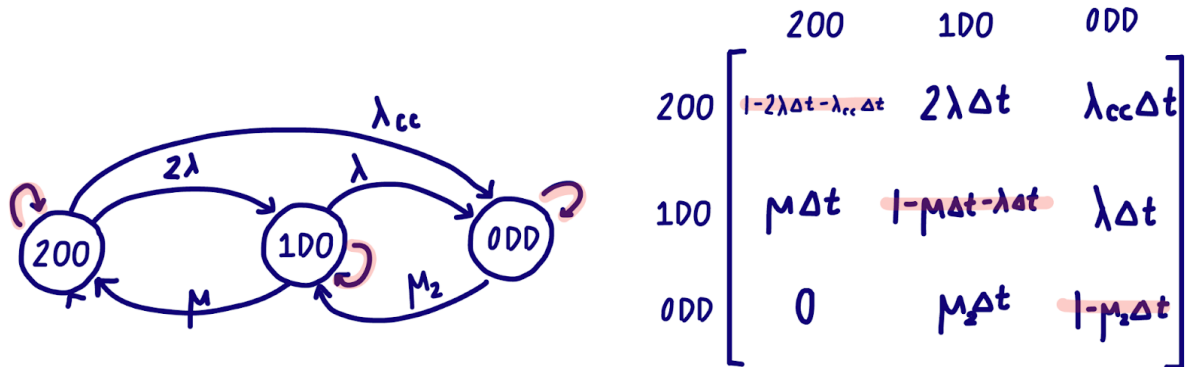
3.2.2 Model 1: Recoverable System, Single equals Dual Recovery Rate

Model 1 assumes that the system is recoverable and that the rate of recovering two processors is equal to the rate of recovering one processor. The model has a recovery rate μ and upset rate λ , and common cause factor cc . The common cause factor is a number between 0 and 1. Figure 1 shows a simplified version of Model 1, which describes three states, 2OO, 1DO, and 0DD, shown as circles. State 2OO represents a state with zero upsets, state 1DO with one upset of type D, and state 0DD with two upsets, both of type D. In the figure, transitions between states are represented by arrows. For arrows not highlighted in red, transition rates are labeled above the arrow and transition probabilities are listed in the transition matrix. Arrows highlighted in red indicate transitions from a state back to itself. For these transitions, transition rates are not shown on the diagram and transition probabilities are listed in the transition matrix, highlighted in red as well. Transition probabilities can be calculated as the transition rate multiplied by a time interval Δt . Note that the transition between 2OO and 0DD is labeled as λ_{cc} where $\lambda_{cc} = \lambda * cc$. Also note that the transition probability from 0DD to 2OO is equal to the transition probability from 1DO to 2OO, namely, $\mu\Delta t$.

3.2.3 Model 2: Recoverable System, Different Single Recovery Rates

Model 2 assumes that the system is recoverable and that items recover from upset one at a time at rates μ and μ_2 . The model has an upset rate λ and common cause factor cc . Figure 3. shows a simplified version of Model 2, which describes three states, 200, 1DO, and 0DD. Note that the transition probability from 0DD to 200 is 0, the transition probability from 0DD to 1DO is $\mu_2\Delta t$, and the transition probability from 1DO to 200 is $\mu\Delta t$.

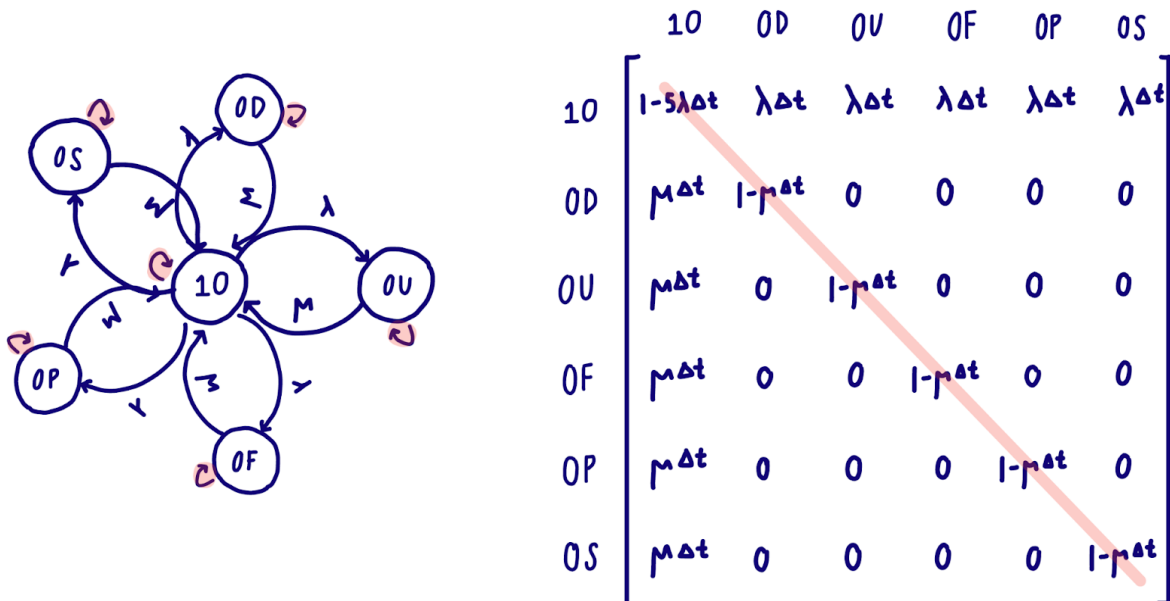
Figure 3: Simplified Model 2 and Corresponding Transition Matrix.



3.2.4 Model 3: Recovery Time < Boot-up Time

Model 3 accounts for the case where the time it takes to boot-up the second board is greater than or equal to the recovery time. Since it would be faster to continue using one processor in this case, a model is presented as seen in Figure 4. The model has a recovery rate μ , and an upset rate λ . It contains a 1O state in the center, indicating one non-upset state. The model also includes five surrounding states, 0D, 0U, 0F, 0P, and 0S, indicating states with one upset of the types listed.

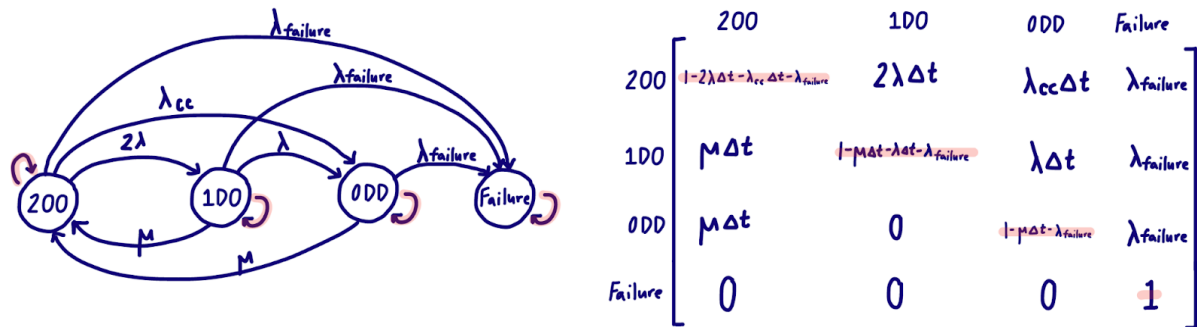
Figure 4: Model 3 and Corresponding Transition Matrix



3.2.5 Model 4: Non-recoverable System Upsets

Model 4, shown in Figure 5, was created to account for non-recoverable upsets. This model is based on Model 1 and introduces a new state, “Failure” that can be transitioned into from any other state with a transition probability $\lambda_{failure}\Delta t$. The transition rate, $\lambda_{failure}$, is assumed to be low and in simulations it has been somewhat arbitrarily chosen as 0.1/day, though this value can and should be substituted with an experimentally determined value in the future.

Figure 5: Model 4 and corresponding transition matrix.



4. RESULTS

The models were simulated in MATLAB with parameters that reflect the processor’s behavior: $\mu = 0.033/s$ and $\mu_2 = 0.033/s$, $\lambda = 1.6E-4/s$ (memory upset rate) and $\lambda = 1.6E-5/s$ (upset rate for other components), $\lambda_{failure} = 1.6E-6/s$ (non-recoverable failure rate), $cc = 0.10$ (common cause). Note that only a select group of states is plotted in part for readability and in part to show how the number of upsets is related to the availability of a state.

The availability of the 200, 100, and 000 states were plotted for Model 1, Model 2, and the Model 4 in figures 6a, 6b, and 6c. The availability values after 10^6 seconds for the 200 state were determined to be about 0.975, 0.970, and 0.344 for those respective models. The availability of the 200, 100, and 000 states were plotted for the Model 3, Figure 7. The availability value after 10^6 seconds for the 200 state was determined to be about 0.989.

Additional plots were made to plot availability while varying parameters. Figure 8 plots the availability of the 200 state for Model 1 as μ varies from 0.02/s to 0.04/s in increments of 0.0025/s. As the recovery rate μ increases, the availability increases as well, though the relationship between μ and availability is nonlinear. Changes in an already high μ (such as the change from 0.0375/s to 0.04/s) cause a smaller increase in availability than the same change to a lower μ (such as the change from 0.02/s to 0.0225/s). Figure 9 plots the availability of the 200 state for Model 1 after 1000 seconds as μ varies from 0.02/s to 0.04/s and cc varies from 0 to 20%. The plot shows that a higher μ and a higher cc leads to greater availability.

Figure 6a: Model 1 Availability Plot

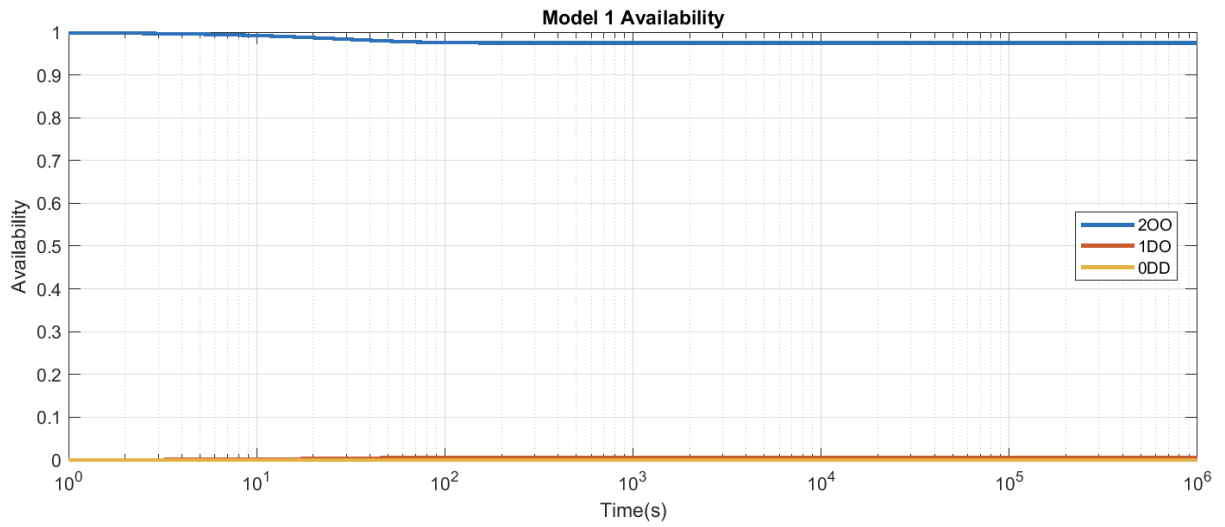


Figure 6b: Model 2 Availability Plot

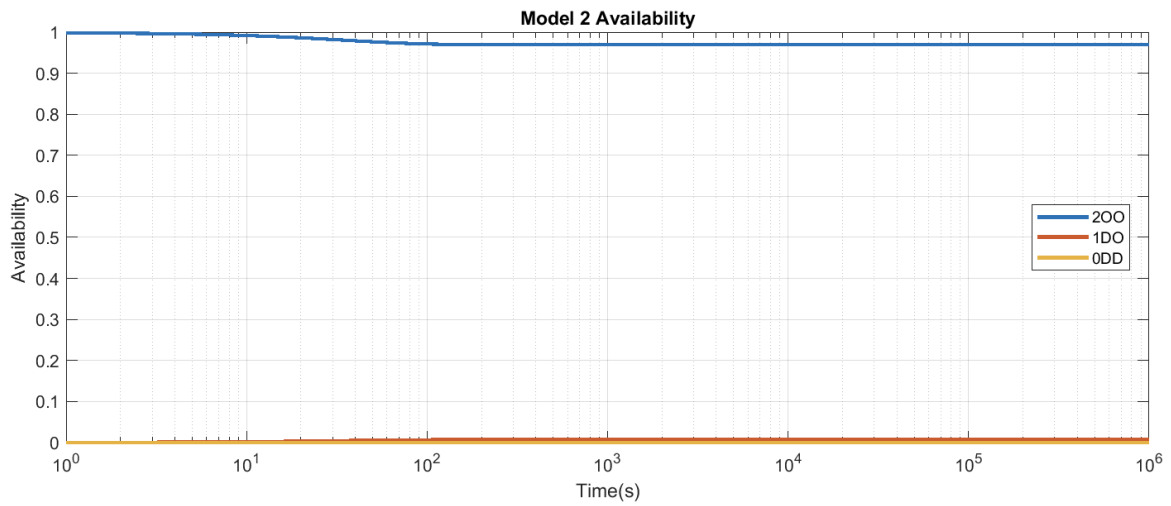


Figure 6c: Model 4 Availability Plot

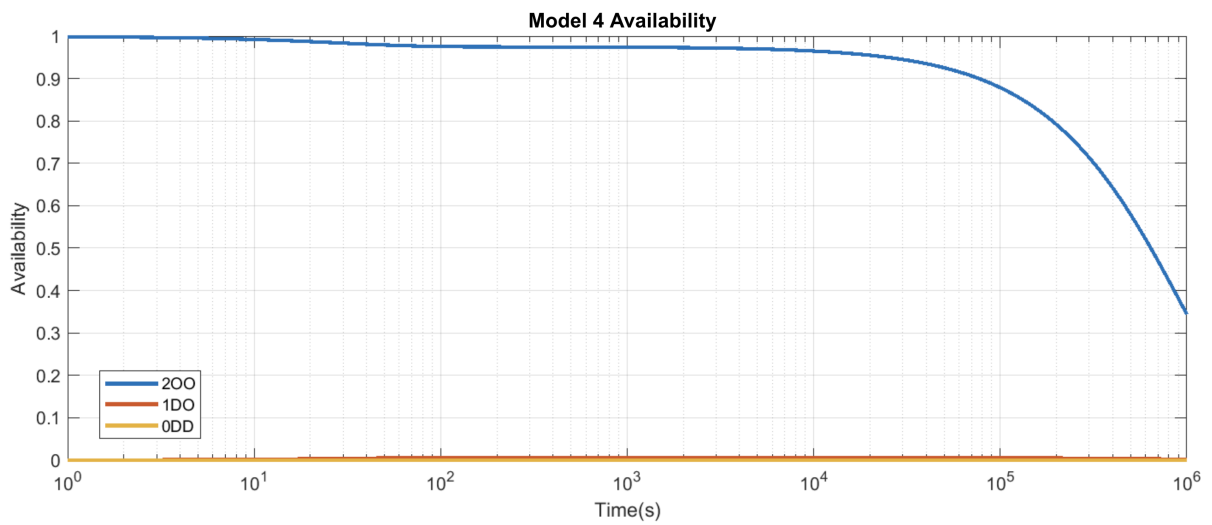


Figure 7: Model 3 Availability Plot

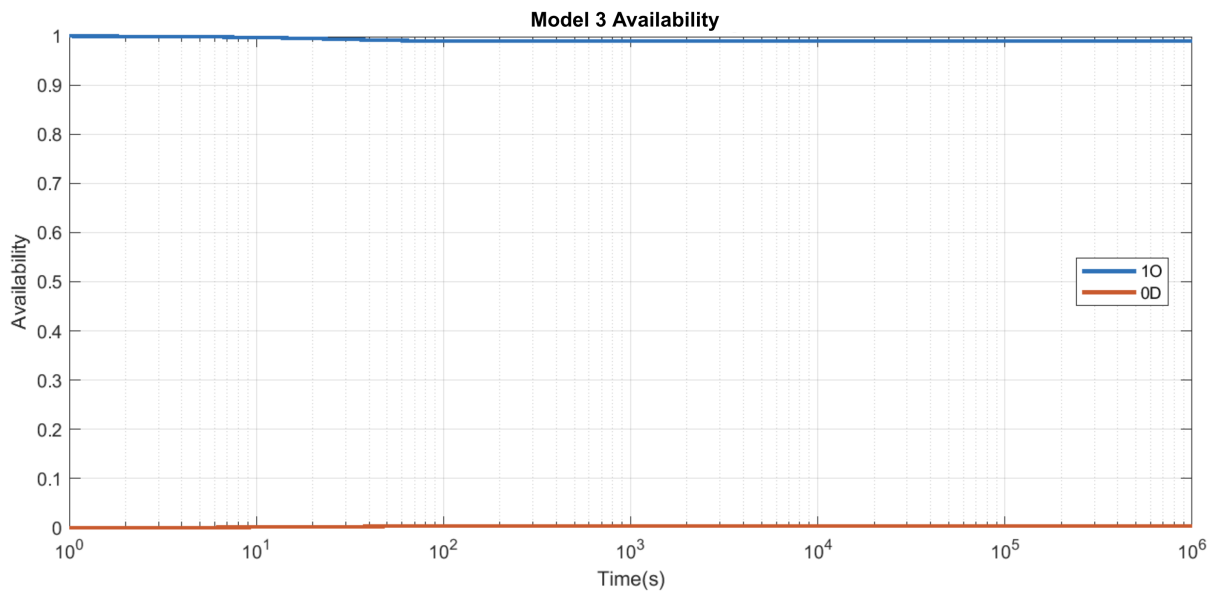


Figure 8: Model 1 Availability of 200 State with Varying μ .

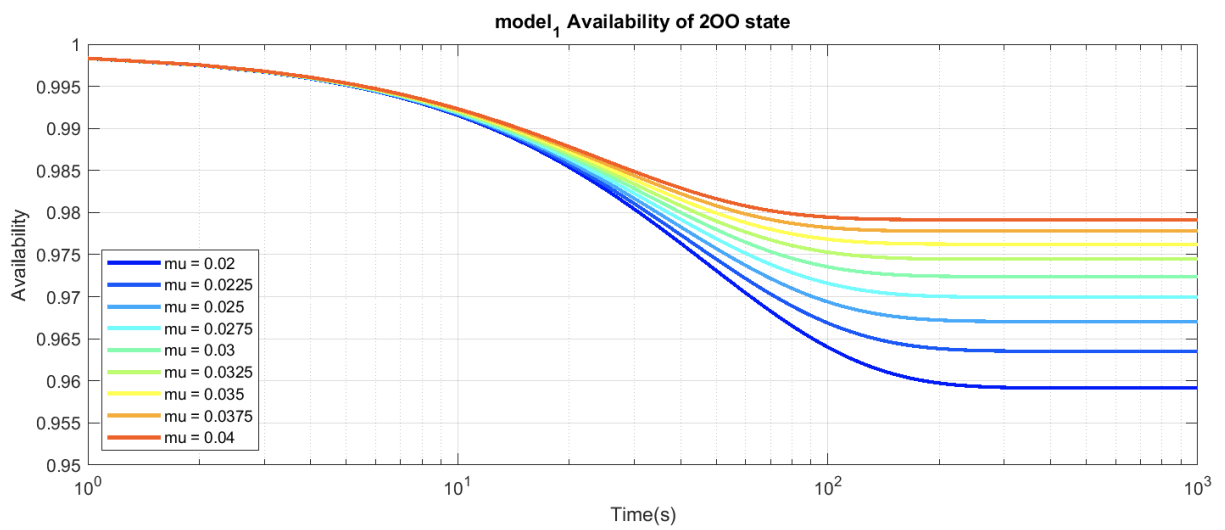
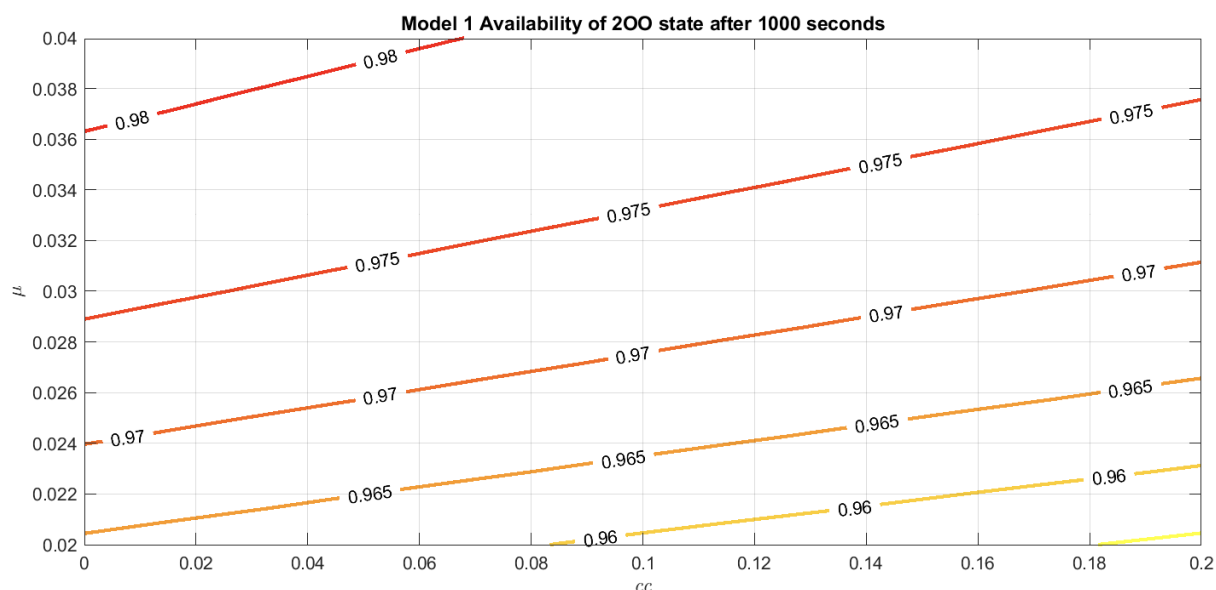


Figure 9: Model 1 Availability of 200 State with Varying μ and cc .



5. CONCLUSION

This work produced four models for a system with two Qualcomm Snapdragon processors with five types of upsets each. The work explored recoverable and non-recoverable systems and produced MATLAB scripts to allow for future simulations and sensitivity analysis.

Additionally, preliminary work has been done to model the system for larger recovery periods, such as 12- and 24-hour periods which would be a typical recovery period for a Single Event Upset (SEA); future work should continue this effort into faster recovery times. An additional model was created to count the number of upsets in a simulation and was tested with Model 3. Future work would include testing the validity of the upset-counting-model and applying it to Model 1, Model 2, and Model 4. Furthermore, work could also be done to model different recovery rates for each type of upset.

This effort demonstrates that Markov models may be used in the future to convey information about the reliability of the processor and whether a system with two processors is reliable enough to be used in space environments. Initial results show that a high level of availability can be achieved with short reset periods, although short reset periods are not guaranteed unless built into the design. Short reset periods are justifiable and can be practically manifested as restarting application processes on various computing cores inside the Snapdragon SoC for example. The robustness of the models supports the further investigation of ultra-high performance commercial semiconductor devices like the Qualcomm Snapdragon and their ability to provide game changing technology demonstrations like Ingenuity.

Acknowledgements

This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by the JPL Student Internship Program (SIP).

References

- [1] D. McMurtrey et al., "Estimating TMR reliability on FPGAs using Markov models," 2008. [Online]. Available: <http://scholarsarchive.byu.edu/facpub/149>