

Hardware/ Software Co-Reliability of Configurable Digital Systems

M. Choi
Dept of ECE
University of Missouri-Rolla
Rolla, MO 65409-0040
choim@umr.edu

N. Park
Dept of CS
Oklahoma State University
Stillwater, OK 74078-1053
npark@a.cs.okstate.edu

Y. Kim and F. Lombardi
Dept of ECE
Northeastern University
Boston, MA 02115
{ybk,lombardi}@ece.neu.edu

Abstract

This paper investigates on the co-effect of hardware and software on the reliability as measured by quality level (or defect level) of configurable multichip module (CMCM) systems. Hardware architecture of CMCM can be configured to accommodate target application design. An application, as provided in a form of software, is partitioned and mapped on the provided configurable hardware. Granularity of an application can be used as a criteria of partitioning and mapping, and can determine the utilization pattern of hardware resources. The utilization pattern of CMCM determines the configuration strategy of available hardware resources based on the application's granularity. Different utilizations of an application design on CMCM may result in various impacts on escape tolerance, (i.e. the probability to avoid inclusion of hardware resources in the configuration that escaped from testing). A quality level model of CMCM is proposed to capture and trace the co-effect of hardware and software, referred to as co-reliability, with respect to escape-tolerance. Various configuration strategies are proposed and evaluated against various criterion granularity and utilization distributions based on the proposed models and evaluation techniques. Extensive analytical and parametric simulation results are shown.

1 Introduction

Field programming system (FPS) techniques have been emerging since the mid-1980's as a quick-turnaround alternative to mask-programmed gate arrays of up to a few-thousand gates [10, 17]. Field programmable gate arrays (FPGAs) and field programmable logic devices (FPLDs) technologies have been enabling these techniques [10, 17].

An emerging yet already standard way to reduce risk and time-to-market in system-level development, while retaining a high level of logic integration, lies in the use of FPGAs [17]. The use of FPGAs provides benefits during all

the stages of system development [17]. During prototyping, the devices provide and emulate functional hardware at-speed without the cost and delay by using wire-wrapped standard device technology for a prototyping or emulation of masked gate arrays [17]. FPSs integrating large numbers of FPGAs are typically bulky, expensive and slow because of the inherently low integration density and space utilization of FPGAs [14].

Programmability was introduced to Multichip Module (MCM) [18] technology in [15] and MCM technology was first applied and implemented on FPGAs in [14]. MCM packaging technology can reduce the cost and improve the utilization of FPSs [3, 6, 7, 10, 11, 12, 13, 14]. Also, it can dramatically increase the capability of field programmable logic devices and field programmable systems in such a way that it can cost-effectively deliver 4-8 times the capacity of the largest FPLDs and provide even larger reductions in the area of PCB-based FPSs [3, 6, 10, 12, 13, 14]. The design and special advantages of configurable MCM (CMCM) have been introduced and demonstrated in [3, 6, 10, 11, 12, 14, 16].

Due to the nature of MCMs, i.e. uneven known-good-yield, uneven fault coverage and imperfect diagnosis to mention a few [8], achievement of an acceptable assembly yield and requirement of product quality of CMCM are critical issues and to be assured with respect to the unique features due to the configurability.

There have been a few works introduced on assuring the quality level (QL) of MCMs under various features of MCMs such as uneven fault-coverage and imperfect diagnosis [8], repair process [5] and uneven known-good-yield [1]. [3] also investigated quality enhancement of reconfigurable MCM systems by redundancy utilization.

Hardware architecture is configured to accommodate a target application or multi-applications at a time or discrete times. An application provided in a form of software is partitioned and mapped on the hardware in an ad-hoc manner. *Granularity* of an application can be used as a basic criteria of partitioning and mapping process. The granularity-based

partitioning and mapping process determines the utilization pattern of hardware resources [2]. The utilization pattern of CMCM can be realized through configuration of available hardware resources in the presence of application's granularity. Different utilizations of same application design on CMCM may result in different degree of *escape tolerance* [1, 3, 5, 8], (i.e. the escape tolerance defined by the probability to avoid inclusion of hardware resources escaped from testing during configuration process and the escaped chips are tested and diagnosed as fault-free but they are actually faulty). A quality level model of CMCM is proposed to capture and trace as quality level (or defect level) is (can) be used as a measure of the reliability of (C)MCM systems design and implementation. The co-effect of hardware and software in terms of quality level is referred to as *co-reliability* with respect to escape-tolerance. A new approach should be developed to assure the overall quality of CMCM because different dies programmable on the CMCM have wide variations in quality level, which cannot be effectively evaluated by using previous approaches [1, 3, 8, 5] that did not consider the configurability of CMCMs either during assembly or implementation time.

Given results of testing (i.e. the numbers of good, bad and escaped chips [3, 8]), the quality level of CMCM is determined by whether escaped chips are *activated* (i.e. participating in the configuration of the device under implementation) or not. Hence, the *activation* or *no-activation* (i.e. not participating in the configuration of the device under implementation) of escaped chips for configuring the device under implementation should be taken into account to trace the impact of escaped chips in evaluating the quality level. Configuration is conducted so CMCM accommodate different functionalities; unlike the ordinary static MCM repair technologies investigated in [9], CMCM can be reconfigured to tolerate faults as well as escapes by activating redundant chips. Hence, each device is implemented by utilizing different number of chips, i.e. has different utilizations of chips on CMCM, and the utilization pattern is determined by the patterns of application designs. Therefore, the extent of *activating* escaped chips depends on the utilization of chips and the reconfiguration strategy.

There have been a few works on evaluating the quality level but the effect of the configurability on the quality level has not been adequately and efficiently addressed, especially no consideration has been given on the effect of utilization of the configurable chips and reconfiguration strategies. Hence, a new evaluation technique to take into account the utilization and the reconfiguration strategies should be developed from the standpoints of both hardware architecture to be configured affected by the characteristics of application.

The objective of this paper is to develop a new model to evaluate and assure the quality level of CMCM with respect

to CMCM's unique features such as configurability, escape tolerance, and hardware/software co-effect, thereby realizing hardware/software co-reliability-effective configuration strategies.

This paper is organized as follows. In the next section, previous works on the CMCM are reviewed, and basic principles and assumptions of the proposed approach is introduced. Section 2 describes basic concepts of the hardware/software co-quality of CMCM. Basic configuration strategies are proposed and studied in section 3. Then, the proposed approaches are analyzed and evaluated via parametric simulations in section 4. Final discussions and conclusions are presented in section 5.

2 Configuration for Hardware/Software Co-Quality

The assumptions of the proposed approach are as follows: 1) The CMCM is an array composed of N identical known-good-dies, whose known-good-yields and fault-coverages are even (identical). 2) Testing is performed during the assembly phase [8]. 3) Failure independence is assumed for the chips (known-good-dies) on CMCM. 4) Selection process of the chips on CMCM for configuration is random. 5) Configuration process is perfect.

CMCM is programmed and configured to implement devices of different functions which are implemented by utilizing different number of chips on the CMCM depending on the utilization pattern of the application under design. The utilization pattern must have a distribution (which is referred to as *FUD*, *Field-Utilization-Distribution*) with respect to the average and variance of the number of chips utilized. The *FUD* can be formalized and characterized by adjusting a normal distribution into a range of discrete domain.

The proposed *FUD* characterization is as follows.

1. *Coarse-Grained-Utilization (CGU)* : has high average with low variance of the number of chips, i.e. the majority of the devices under design and implementation utilize large portion of the CMCM.
2. *Medium-Grained-Utilization (MGU)* : has medium average with low variance of the number of chips, i.e. the majority of the devices under design and implementation utilize medium portion of the CMCM.
3. *Fine-Grained-Utilization (FGU)* : has low average with low variance of the number of chips, i.e. the majority of the devices under design and implementation utilize small portion of the CMCM.

Excluding escaped chips during chip selection for configuring (or reconfiguring) a device is referred to as *escape-tolerance* in this paper. To reduce a quorum size (i.e. the

number of chips needed to implement a device) by Reconfiguring CMCM to implement a device with reduced quorum size (i.e. a smaller number of chips needed to implement a device) or by redundant chips, the probability of excluding escaped chips from participating in configuring the device may be decreased. Hence, configurability may enhance quality level through *escape-tolerance*.

CMCM can be configured for the purpose of function design and implementation (i.e. when $g + e \geq Q$, where g , e are the number of good chips and escaped chips, respectively and Q is the number of quorum of quorum of the device under implementation); and can also be reconfigured for the purpose of fault-tolerance (i.e. when $g + e < Q$) and escape-tolerance in the presence of faulty/escaped resources detected (note that escaped resources are implicitly traced than explicit detection).

The proposed reconfiguration strategies for fault-tolerance purpose can be characterized as follows.

1. *Down-Sizing*: reconfiguration from a configuration consisting of g good chips, b bad chips, e escaped chips of a device with quorum q to a configuration of a device with quorum 1 through $q-1$ maintaining the same number of activated chips (i.e. the chips activated to participate in the device under implementation). Some or all of the escaped chips before reconfiguration can be inherited down to the CMCM after reconfiguration. However, the effect of the inherited escaped chips may be reduced.
2. *Up-Sizing*: reconfiguration from a configuration consisting of g good chips, b bad chips, e escaped chips for a device with quorum q to a configuration for a device with quorum $q+1$ through N (the number of all chips on the CMCM) by activating more chips. All of the escaped chips are inherited down to the new configuration and the additionally activated chips may increase the number of escaped chips.
3. *n-Activation*: reconfiguration from a configuration consisting of g good chips, b bad chips, e escaped chips for a device with quorum q to a configuration for a device with the same quorum q and more activated chips (i.e. as spares). The activation can be categorized as follows.
 - *min-activation* : activates $q - g - e$ chips, i.e. the minimum number of chips required to tolerate faulty chips.
 - *max-activation* : activates $N - g - b - e$ chips, i.e. the maximum number of chips available to activate to tolerate faulty chips.

The qualitative comparisons between the proposed strategies for fault-tolerance purpose are shown in Table (1).

Similar analysis can be applied to non-configurable MCM but no active configuration or reconfiguration processes apply so the n-Activation does not take place in non-configurable MCM.

3 Modeling the Configuration Processes

The process of testing and (re)configuring the CMCM as proposed in the previous section can be modeled and evaluated by using two models, i.e. the *Base model* (shown in Figure (2)) and *Field model* (shown in Figure (4)). From the *Base model*, each *base state* (where $G + B + E = N$) can be derived as have been shown in [8]. Each *base state* is mapped onto multiple *field states* in the *Field model* as is shown in Figure (4).

Figure (4) can be expressed as follows.

- Each *base state* (i.e. state (G, B, E)) is the initial state of the model and whose state probability is also the initial probability.
- (G, B, E) is mapped into states (q, g, b, e) s, where $q \leq G + B + E$ and $g + b + e \leq q$, and $g \leq G$, $b \leq B$, $e \leq E$.
- The state transition probability from a *base state* to each *field state* is calculated by $\frac{G \cdot C_q \cdot B \cdot C_b \cdot E \cdot C_e}{G + B + E \cdot C_q} \cdot Util(q)$, where $Util(q)$ is the probability to utilize the system with quorum q and is derived from *FUD*.
- The overall *Field model* is initialized after mapping all the *base states* into its corresponding *field states*.

Once the *Field model* is constructed, the (re)configuration process can be evaluated as follows.

1. In Figure (1), the state transitions of *Down-Sizing* are shown. The state transition probabilities from each state (q, g, b, e) (where q is the quorum of a device under implementation) to (q', g, b, e) is $\frac{Util(q')}{\sum_{i=1}^{q-1} Util(i)}$, where q' is the quorum of another device in implementation after reconfiguration and $1 \leq q' \leq q - 1$.
2. In Figure (5), state transitions of *Up-Sizing* are shown. The state transition probabilities from each state (q, g, b, e) to (q', g', b', e') is $\left(\frac{G-g \cdot C_{q'-g} \cdot B-b \cdot C_{b'-b} \cdot E-e \cdot C_{e'-e}}{G+B+E-g-b-e \cdot C_{q'-q}} \right) \left(\frac{Util(q')}{\sum_{i=q+1}^N Util(i)} \right)$, where $q' = g' + b' + e'$ and $g' \geq g$, $b' \geq b$, $e' \geq e$, and $g' \leq G$, $b' \leq B$, $e' \leq E$ because G, B, E are the number of chips tested as good, bad and escaped, respectively.
3. In Figure (3), (6), state transitions of *n-Activation* with *max-activation* and *min-activation* are shown. The

Table 1. Qualitative comparison between reconfiguration strategies under different FUDs

Comparison	CGU			MGU			FGU		
	Recon. Time	FT	Yield	Recon. Time	FT	Yield	Recon. Time	FT	Yield
<i>Down-Sizing</i>	4	1	1	1	2	2	1	4	4
<i>Up-Sizing</i>	2	3	3	1	2	2	3	2	2
<i>n-Act./min-act.</i>	1	4	4	1	3	3	4	1	1
<i>n-Act./max-act.</i>	3	2	2	2	1	1	2	3	3

state transition probabilities from each state (q, g, b, e) to (q', g', b', e') is $\frac{G-gC_{q'-q}B-bC_{b'-b}E-eC_{e'-e}}{G+B+E-g-b-eC_{q'-q}}$, where $q' = q + N - g - b - e$ for the *max-activation* and $q' = q + N - g - e$ for the *min-activation*; and $g \leq g' \leq G$, $b \leq b' \leq B$, $e \leq e' \leq E$ and $g' + b' + e' = q'$.

The probability for each *base state* to be escape-free can be derived by summing up the steady state probabilities of escape-free states (which is referred to as *EFP, Escape-Free-Probability*) in the *Field model*. Therefore, the overall quality (*QL*) level can be derived by summing up all the *base states* multiplied by its probability to be escape-free as shown in the following.

$$QL = \sum_{0 \leq G \leq N} \sum_{0 \leq B \leq N} \sum_{0 \leq E \leq N} (P(G, B, E) \cdot EFP(G, B, E)) \quad (1)$$

where $EFP(G, B, E)$ can be expressed by

$$EFP(G, B, E) = \sum_{g \geq q \text{ or } b > N - q} P(q, g, b, e). \quad (2)$$

4 Parametric Analysis

In this section, the compound effect of the *FUD* and (re)configuration strategies on the *quality level* of field-programmable MCM will be studied through numerical experiments and performances of different (re)configuration strategies under various *FUDs* will be analyzed.

The number of chips on the CMCM is 10 in this paper. The overall known-good-yield (*Y*) of the CMCM is increased by 0.2 from 0.1 to 0.9 and the fault-coverage (*C*) of each chip is increased by 0.2 from 0.1 to 0.9. The average of the *FUD* is increased by 1 from 1 to 9 chips and the variance is increased by 1 from 1 to 9.

In Figure (7)-(8), the defect-level (*DL*) of *Down-Sizing* strategy at different *Y* and *C* under different average and variance of the *FUD* are shown. The followings are observed from these results. 1) The *DL* increases as *C* increases at low average (≤ 3) and low variance (≤ 3) of the *FUD*. 2) Given *C*, the *DL* increases as the variance of the *FUD* increases at the average < 9 and there exists a bound

on the maximum value of the *DL* when the average is 9. 2) The *DL* decreases as the average of the *FUD* increases at low *Y*, high *C* and low variance of the *FUD*.

In Figure (9)-(10), the defect-level (*DL*) of *Up-Sizing* strategy at different *Y* and *C* under different average and variance of the *FUD* are shown. The followings are observed from these results. 1) Given *C*, the *DL* increases as the average and the variance of the *FUD* increases except at high average, low variance of the *FUD* and low *Y*. 2) The *DL* converges at high *Y* (0.9) and high average and low variance of the *FUD*.

In Figure (11)-(12), the defect-level (*DL*) of *max-Activation* strategy at different *Y* and *C* under different average and variance of the *FUD* are shown. Same observations with the *Up-Sizing* strategy are shown except that there is no convergence at high *Y* (0.9), high average, and low variance of the *FUD*.

In Figure (13)-(14), the defect-level (*DL*) of *min-Activation* strategy at different *Y* and *C* under different average and variance of the *FUD* are shown. Same observations with the *max-Activation* strategy are shown except at low *Y* and $C \geq 0.5$.

By comparing results from the *DL* of *Down-Sizing*, *Up-Sizing* and *max-Activation* strategies, at low *Y* (0.1), $C (\geq 0.5)$ and low average (≤ 5), low variance (≤ 3) of the *FUD*, the *DL* of *Up-Sizing* shows the lowest and at high average of the *FUD*, the *DL* of *Down-Sizing* shows the lowest. Also, by comparing results from the *DL* of *max-Activation* and *min-Activation* strategies at low *Y* (0.1), $C (\geq 0.5)$ and medium average (3-7), low variance (1) of the *FUD*, the *DL* of *max-Activation* is smaller than that of *min-Activation* strategy.

Therefore, it can be concluded that, by using *Down-Sizing* strategy, *C*, the average and the variance of the *FUD* should be kept low to achieve a low *DL*. At low *Y*, high *C* and at low variance of the *FUD*, the average of the *FUD* should be kept high to achieve a low *DL*. By using *Up-Sizing* strategy, at high *Y*, average and low variance of the *FUD*, *C* may be kept low without any loss in the *QL*. By using *max-Activation* and *min-Activation* strategies, *C* has to be kept high and the average and the variance of the *FUD* should be kept low to achieve a low *DL*.

5 Discussion and Conclusions

This paper proposed a new approach to evaluate the co-effect of hardware and software on the reliability as measured by quality level (or defect level) of configurable multichip module (CMCM) systems. Hardware architecture of CMCM can be configured to accommodate target application design. An application, as provided in a form of software, is partitioned and mapped on the provided configurable hardware. *Granularity* of an application can be used as a criteria of partitioning and mapping, and can determine the utilization pattern of hardware resources. The utilization pattern of CMCM determines the configuration strategy of available hardware resources based on the application's granularity. Different utilizations of an application design on CMCM may result in various impacts on *escape tolerance*, (i.e. the probability to avoid inclusion of hardware resources in the configuration that escaped from testing). A quality level model of CMCM is proposed to capture and trace the co-effect of hardware and software, referred to as *co-reliability*, with respect to escape-tolerance. Various configuration strategies are proposed and evaluated against various criterion granularity and utilization distributions based on the proposed models and evaluation techniques. A new Markov-chain model has been developed to evaluate the quality level of CMCM based on the previous model [8] as an *Base model*, in which the steady-state probability of each *base states* to tolerate escapes has been evaluated. From extensive parametric analysis based on the proposed quality model, it is concluded that coarse-grained-utilization strategy is suitable for using *Up-Sizing* strategy for the CMCM with high *Y* and low variance of the *FUD*; and fine-grained-utilization is suitable for using *Down-Sizing* strategy for the CMCM with low *C* and variance of the *FUD*, and for using *max-Activation* or *min-Activation* strategies for the CMCM with high *C* and low variance of the *FUD*. With the proposed approach, co-reliability-effective CMCM design and implementation can be achieved by exploiting effective reconfiguration strategies in the presence of various distribution of utilization of application designs.

References

- [1] N. Park, F. Lombardi, V. Piuri, "Testing and evaluating the quality-level of stratified multichip module instrumentation", *Instrumentation and Measurement, IEEE Transactions on*, vol. 50 issue. 6 pp. 1615 - 1624, Dec. 2001
- [2] C. Bolchini, L. Pomante, F. Salice, and D. Sciuto, "Online fault detection in a hardware/software co-design environment: system partitioning", *System Synthesis, 2001. Proceedings. The 14th International Symposium on*, pp. 51-56, Oct. 2001
- [3] M. Choi, N. Park, F. Lombardi, and V. Piuri, "Quality enhancement of reconfigurable multichip module systems by redundant utilization", *Instrumentation and Measurement Technology Conference, 2001. IMTC 2001. Proceedings of the 18th IEEE*, pp. 204-209, May 2001
- [4] P. Christie, D. Stroobandt, "The interpretation and application of Rent's rule", *Very Large Scale Integration Systems, IEEE Transactions on*, vol. 8 issue. 6 pp. 639 - 648, Dec. 2000
- [5] N. Park, F.J. Meyer, F. Lombardi, "Quality-effective repair of multichip module systems", *Defect and Fault Tolerance in VLSI Systems, 2000. Proceedings., IEEE International Symposium on*, Oct. 1996
- [6] R. Amerson, R. Carter, W. Culbertson, P. Kuekes, G. Snider, and L. Albertson, "An FPGA for multi-chip reconfigurable logic" *Custom Integrated Circuits Conference, 1996., Proceedings of the IEEE 1996*, pp. 137-143, May 1996
- [7] J. Darnauer and W. Dai, "Tradeoffs in chip and substrate complexity and cost for field programmable multichip modules" *Innovative Systems in Silicon, 1996. Proceedings., Eighth Annual IEEE International Conference on*, pp. 217-227, Oct. 1996
- [8] N. Park, S. Kim and F. Lombardi, "Modeling quality reduction of multichip module systems due to uneven fault-coverage and imperfect diagnosis" *Defect and Fault Tolerance in VLSI Systems, 1996. Proceedings., 1996 IEEE International Symposium on*, pp. 168-176, Nov. 1996
- [9] T.A. Wassick, L. Economikos, "Open repair technologies for MCM-D", *Components, Packaging, and Manufacturing Technology, Part B: Advanced Packaging, IEEE Transactions on*, Vol. 18, No. 1, pp. 154-162, Feb. 1995
- [10] J. Darnauer, T. Isshiki, P. Garay, J. Ramierz, V. Maheshwari, and W. W. Dai, "A silicon-on-silicon programmable multichip module (FPMCM) - Integrating FPGA and MCM technologies", *Components, Packaging, and Manufacturing Technology, Part B: Advanced Packaging, IEEE Transactions on*, Vol. 18, No. 4, pp. 601-608, Nov. 1995
- [11] S. Goldberg and S.J. Upadhyaya, "Utilizing spares in multichip modules for the dual function of fault coverage and fault diagnosis", *Defect and Fault Tolerance in VLSI Systems, 1995. Proceedings., 1995 IEEE International Workshop on*, pp. 234-242, Nov. 1995
- [12] R. Terrill, "Novel 2-sided multi-chip-module used to create a 50,000 gate programmable logic device", *Proceedings of the IEEE Multi-Chip Module Conference, 1995*, pp. 56-60, Feb. 1995
- [13] V. Maheshwari, J. Darnauer, J. Ramirez, and W. Dai, "Design of FPGAs with area I/O for field programmable MCM", *Proceedings of the 1995 ACM 3rd International Symposium on Field-Programmable Gate Arrays*, pp. 17-23, Feb. 1995
- [14] J. Darnauer, P. Garay, T. Isshiki, J. Ramierz, and W. W. Dai, "A field programmable multi-chip module (FPMCM)", *FPGAs for Custom Computing Machines, 1994. Proceedings. IEEE Workshop on*, pp.1-10, Apr. 1994
- [15] S. Burman and N.A. Sherwani, "Programmable multichip modules", *IEEE Micro*, Vol. 13, No. 2, pp. 28-35, Apr. 1993
- [16] I. Dobbelaere, A. E. Gamal, D. How, and B. Kleveland, "Peripheral circuit design for field programmable MCM systems", *Multi-Chip Module Conference, Proceedings 1992 IEEE*, pp.119-122, Mar. 1992
- [17] D. McCarty, "System development using Actel field programmable gate arrays", *Compon Spring '90. Intellectual Leverage. Digest of Papers. Thirty-Fifth IEEE Computer Society International Conference*, pp.470-476, Feb. 1990
- [18] R. R. Johnson, "Multichip module : next-generation packages", *IEEE Spectrum*, Vol. 27 Issue. 3, pp. 34-48, Mar. 1990.

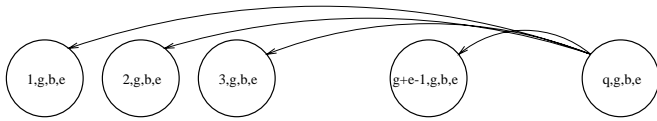


Figure 1. State transitions in *Down-Sizing*

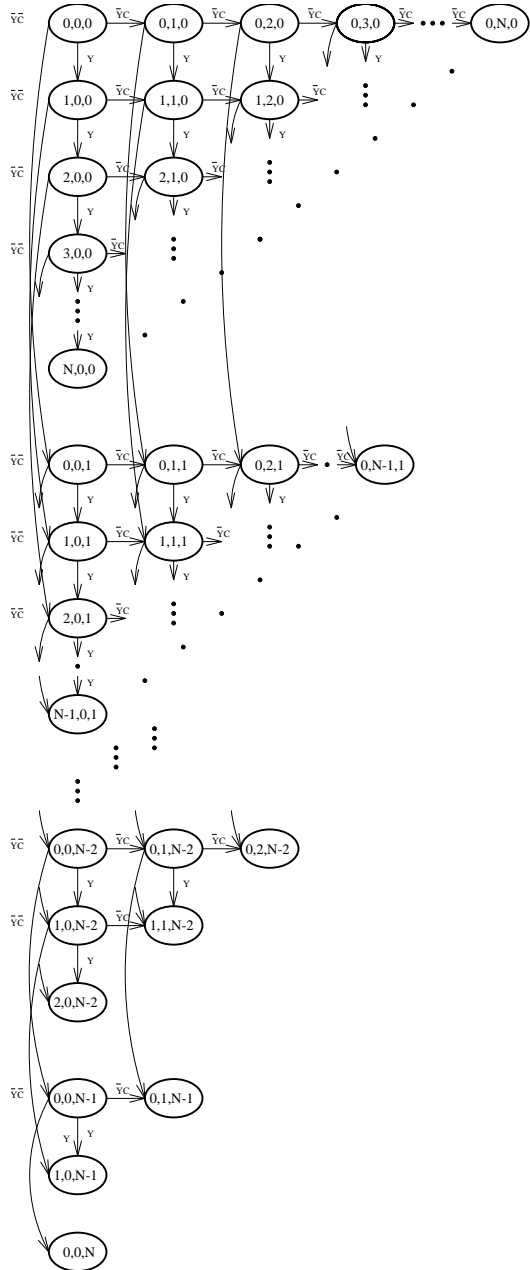


Figure 2. State transition diagram in the *Base* model

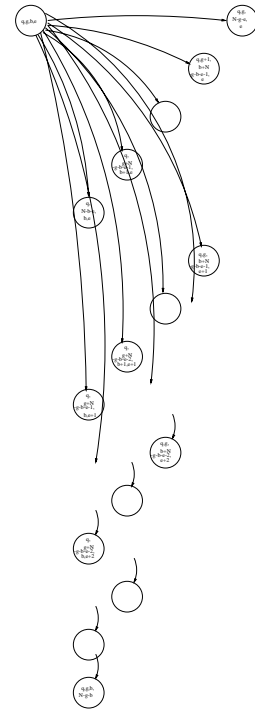


Figure 3. State transitions in *n-Activation (Max)*

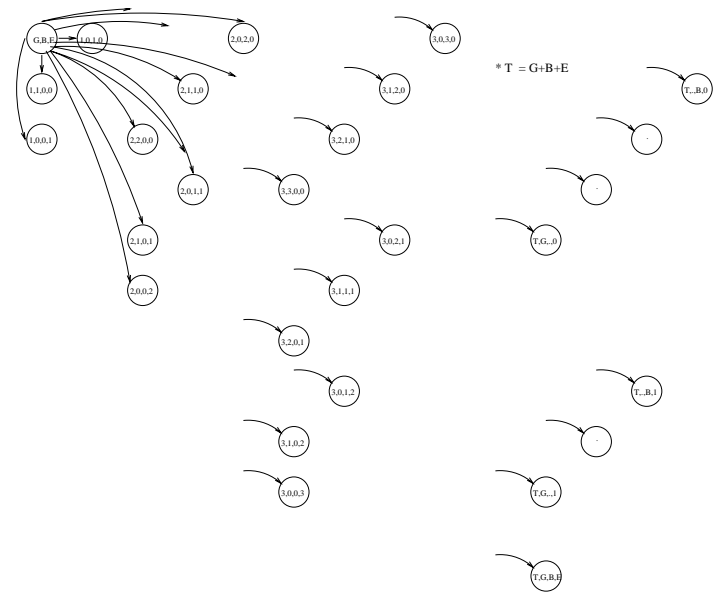


Figure 4. Initial state transition diagram from each *base state* to *field states*

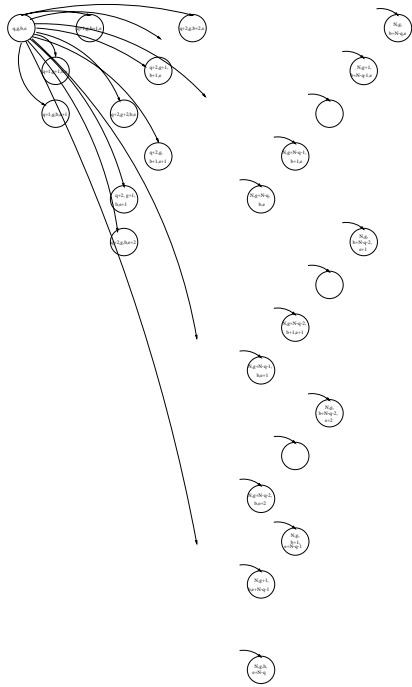


Figure 5. State transitions in *Up-Sizing*

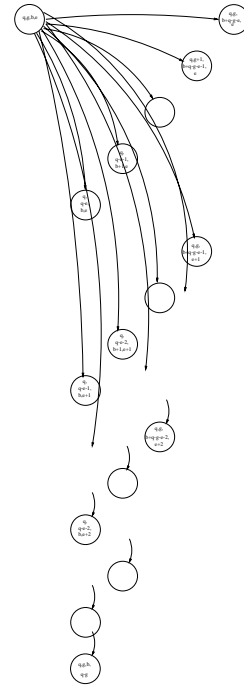


Figure 6. State transitions in *n-Activation (Min)*

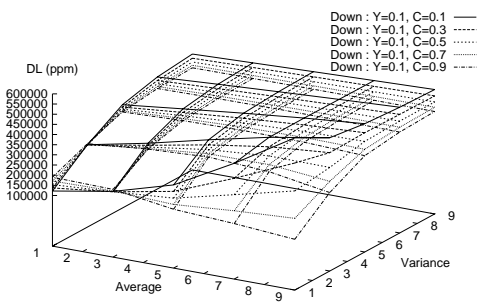


Figure 7. DL (in *ppm*) by using *Down-Sizing* at $Y=0.1$

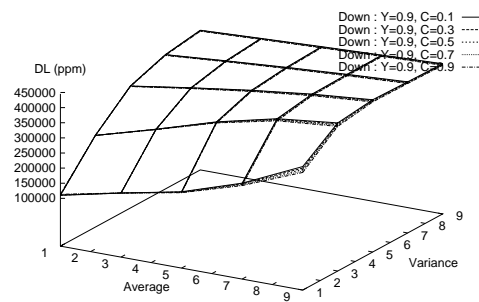


Figure 8. DL (in *ppm*) by using *Down-Sizing* at $Y=0.9$

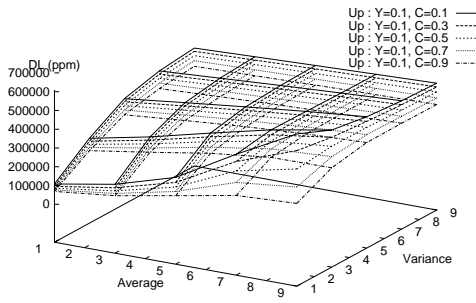


Figure 9. DL (in ppm) by using *Up-Sizing* at Y=0.1

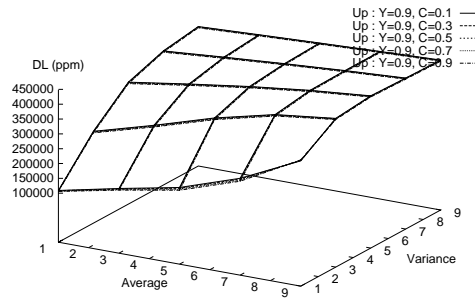


Figure 10. DL (in ppm) by using *Up-Sizing* at Y=0.9

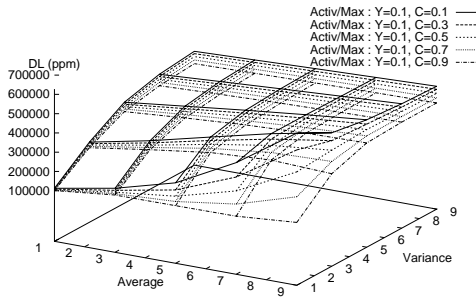


Figure 11. DL (in ppm) by using *max-activation* at Y=0.1

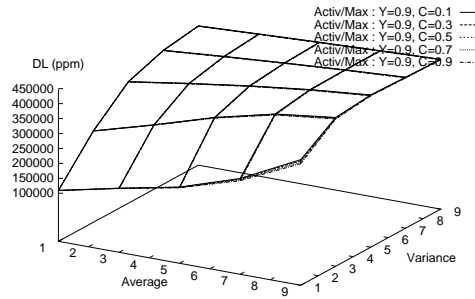


Figure 12. DL (in ppm) by using *max-activation* at Y=0.9

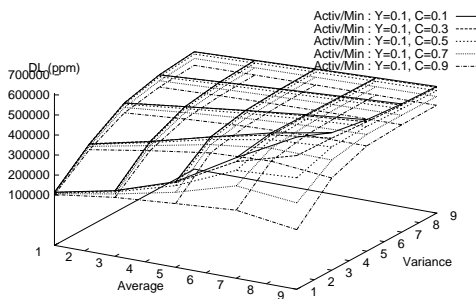


Figure 13. DL (in ppm) by using *min-activation* at Y=0.1

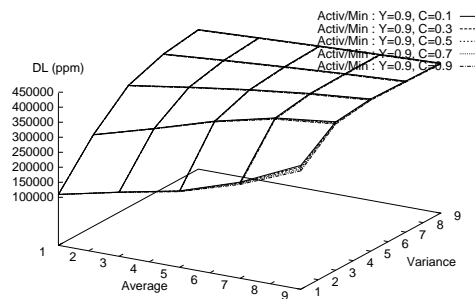


Figure 14. DL (in ppm) by using *min-activation* at Y=0.9