



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΗ
ΜΙΚΡΟΗΛΕΚΤΡΟΝΙΚΗ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Σχεδίαση και FPGA υλοποίηση συστήματος σε ψηφίδα για
στοχαστική προσομοίωση πολύπλοκων βιομοριακών
δικτύων με χρήση του NRM-SSA.**

Ευάγγελος Χ. Κουτσουραδής

Επιβλέπων: Ηλίας Σ. Μανωλάκος, Αναπληρωτής Καθηγητής

**ΑΘΗΝΑ
ΙΟΥΛΙΟΣ 2015**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σχεδίαση και FPGA υλοποίηση συστήματος σε ψηφίδα για στοχαστική προσομοίωση πολύπλοκων βιομοριακών δικτύων με χρήση του NRM-SSA.

Ευάγγελος Χ. Κουτσουραδής
A.M.: MM 231

ΕΠΙΒΛΕΠΩΝ: **Ηλίας Σ. Μανωλάκος**, Αναπληρωτής Καθηγητής

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ: **Ηλίας Σ. Μανωλάκος**, Αναπληρωτής Καθηγητής
Αγγελική Αραπογιάννη, Καθηγήτρια

Ιούλιος 2015

ΠΕΡΙΛΗΨΗ

Στα πλαίσια της διπλωματικής αυτής εργασίας ασχολούμαστε με τη σχεδίαση και την υλοποίηση ενός συστήματος σε ψηφίδα (System on Chip, SoC) που έχει τη δυνατότητα να προσομοιώνει αποδοτικά πολύπλοκα δίκτυα βιομοριακών αλληλεπιδράσεων. Συγκεκριμένα, εστιάζουμε στην ανάδειξη των πλεονεκτημάτων του υλικού επαναπρογραμματιζόμενης λογικής (Field Programmable Gate Array, FPGA) με στόχο την κατασκευή ενός ευέλικτου και πλήρως παραμετροποιημένου συστήματος για στοχαστική προσομοίωση, βασιζόμενοι στον αλγόριθμο Next Reaction Method (NRM) των Gibson και Bruck [1].

Αρχικά, εισάγουμε την έννοια της Συστημική Βιολογίας (Systems Biology) και επισημαίνουμε τη σπουδαιότητα των αλγορίθμων στοχαστικής προσομοίωσης, επικεντρώνοντας την προσοχή μας στον NRM-SSA. Εν συνεχεία, παρουσιάζουμε τη γενική ροή εργασιών του συστήματος στοχαστικής προσομοίωσης που αναπτύξαμε, συμπεριλαμβανομένου του λογισμικού και του υλικού που το απαρτίζουν. Περιγράφουμε όλα τα βήματα που απαιτούνται για: τη διαμόρφωση και τη σύνθεση του κατάλληλου SoC με βάση τις απαιτήσεις του χρήστη, την εκτέλεση της προσομοίωσης και την ανάλυση των αποτελεσμάτων. Έπεται η περιγραφή της αρχιτεκτονικής του NRM SoC, δίνοντας έμφαση στην κύρια υπολογιστική μονάδα του πυρήνα NRM, η οποία είναι η καρδιά του SoC και χάρη στην αποδοτική αρχιτεκτονική της μεγιστοποιείται η επίδοση του συστήματός μας. Στη συνέχεια, παρουσιάζουμε τη θεωρητική ανάλυση των επιδόσεων του NRM SoC και τα στατιστικά υλοποίησης του συστήματός μας με βάση τις γενικές παραμέτρους. Η απόδοση του NRM SoC διατηρείται σε πολύ υψηλά επίπεδα χάρη στην αποτελεσματική αρχιτεκτονική του γράφου εξαρτήσεων (Dependencies Graph, DG) και της κεντρική υπολογιστικής μονάδας του συστήματος, ικανή να εκτελέσει παράλληλα και αποδοτικά όλους απαιτούμενους υπολογισμούς του αλγορίθμου. Ενσωματωμένα Υπολογιστικά Συστήματα Λέξεις Κλειδιά: Βιολογία Συστημάτων Στοχαστική Προσομοίωση Next Reaction Method FPGA Σύστημα σε Ψηφίδα

ABSTRACT

This graduate thesis is concerned with the design and implementation of a flexible System on Chip (SoC) architecture for the efficient parallel stochastic simulation of biomolecular networks. Specifically, we focus on showing the benefits of reconfigurable logic (Field Programmable Gate Array, FPGA) in building a high performance application-specific SoC implementing the popular Next Reaction Method (NRM) Stochastic Simulation Algorithm (SSA) of Gibson and Bruck [1].

First we provide a short introduction to Systems Biology and the theoretical background that can help the reader understand the importance of stochastic simulation algorithms. We discuss the most well known algorithms, and in particular the NRM-SSA. Then, we present our general high performance stochastic simulation workflow, including its software and hardware components. We describe first all steps required in order to configure and synthesize the best SoC instance for a given biomodel simulation (user supplied SBML file), run the stochastic simulation in the FPGA, and analyze the obtained results. Then we introduce the architecture of our NRM-SSA SoC, emphasizing its main computational unit which is the heart of the cores of the SoC. It is this unit's optimal design that leads actually to the overall system's high performance.

Next we present the theoretical analysis of our SoC's expected performance and how FPGA resources utilization is affected by the SoC's architectural parameters. The achieved performance remains very high because we have designed an appropriate memory structure to store and access efficiently the Dependencies Graph (DG) of a biochemical reactions network, and a Processing Element (PE) unit capable to execute all NRM-SSA computations with minimum latency using three aggressively pipelined parallel datapaths.

In order to measure our system's performance, we have simulated several models, biologically relevant or benchmarks, which we can scale in terms of complexity (number of model reactions). The results not only validate our SoC's functionality, but also demonstrate the efficiency of the system when simulating both small but also very large scale biomolecular networks. The SoC's performance is not affected by the complexity of the model (number of reactions) it simulates but depends only on the average outdegree of its Dependencies Graph. As a result, the throughput of the SoC can exceed 3GCycle/sec even for very large biomodels, with 4K reaction channels, that can fit on a medium size FPGA.

In conclusion, the NRM-SSA SoC we have designed in this graduate thesis exploits the efficiency of reconfigurable hardware to provide a powerful tool for Systems Biology studies, that can accelerate significantly the stochastic simulation of large scale biomolecular networks. We hope it will also provide a design example that will inspire more and more hardware designers and students to get involved with Bio circuits and systems (bioCAS) design, a new field promising to contribute to the cause of Systems Biology for biological systems identification and personalized medicine.

SUBJECT AREA: Embedded Systems

KEYWORDS: Systems Biology, Stochastic Simulation, Next Reaction Method, FPGA, System on Chip

*Η παρούσα διπλωματική εργασία
είναι αφιερωμένη στη μητέρα μου,
που όλα αυτά τα χρόνια έκανε
τα πάντα για να με στηρίξει.*

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα διπλωματική εργασία εκπονήθηκε στα πλαίσια του Διατμηματικού Μεταπτυχιακού Προγράμματος Σπουδών “Μικροηλεκτρονικής”, του τμήματος Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Αθηνών. Είναι αποτέλεσμα μιας συλλογικής προσπάθειας και δεν θα είχε ολοκληρωθεί χωρίς τη συμβολή σημαντικών ανθρώπων, τους οποίους και θέλω να ευχαριστήσω.

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα κ. Ηλία Μανωλάκο, για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον αντικείμενο, για την αμέριστη συμπαράστασή του καθ’ όλη τη διάρκεια εκπόνησης και για κάθε λεπτό της ώρας που αφιέρωσε ώστε να μου αφυπνίσει όλες μου τις ερευνητικές ικανότητες που σήμερα κατέχω.

Από τις ευχαριστίες μου δεν θα μπορούσα να παραλείψω τους δύο συναδέλφους και φίλους μου που μοιραστήκαμε μαζί ώρες αγωνίας, ανησυχίας, χαράς και δημιουργίας. Τον Γεώργιο Προβελέγκιο και τον Ηλία Κουσκουμβεκάκη για την υπομονή, την επιμονή και πάνω από όλα τη θέλησή τους να μου μεταδώσουν άπλετη γνώση. Η συμβολή τους ήταν καθοριστική και δεν θα μπορούσα ποτέ να ολοκληρώσω μόνος μου ένα τέτοιου είδους εγχείρημα.

Επίσης, θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στον Δημήτριο Μανατάκη, που παρόλο το μικρό χρονικό διάστημα που γνωριζόμαστε, υπήρξε εξάαιρετος συνεργάτης και κατάφερε με τον δικό του τρόπο να με στηρίξει και να μου μεταδώσει ερευνητική εμπειρία προς την αναζήτηση της γνώσης.

Τέλος, ένα πολύ μεγάλο ευχαριστώ στην οικογένειά μου που τόσα χρόνια με στηρίζουν και δεν σταμάτησαν ποτέ να πιστεύουν σε εμένα, και ένα εξίσου μεγάλο ευχαριστώ στους φίλους μου που μια συζήτηση μαζί τους πάντα κάνει τον κόσμο να φαίνεται ομορφότερος.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ	12
1. ΕΙΣΑΓΩΓΗ	13
1.1 Η Εξέλιξη της Βιολογίας	13
1.2 Βιολογικά Δίκτυα – Μέθοδοι Προσομοίωσης	13
1.3 Διαθέσιμη Βιβλιογραφία - Κίνητρα	14
1.4 Στόχος της Διπλωματικής Εργασίας	15
1.5 Δομή της Διπλωματικής Εργασίας	15
2. ΜΕΘΟΔΟΙ ΣΤΟΧΑΣΤΙΚΗΣ ΠΡΟΣΟΜΟΙΩΣΗΣ	17
2.1 Μοντελοποίηση Βιολογικών Δικτύων	17
2.2 Ο Αλγόριθμος NRM των Gibson και Bruck	18
2.3 Η Διαφοροποίηση του NRM-SSA έναντι του FRM-SSA	19
2.4 Ο Γράφος Εξαρτήσεων - Dependencies Graph	19
3. ΠΡΟΔΙΑΓΡΑΦΕΣ ΣΥΣΤΗΜΑΤΟΣ	21
3.1 Προδιαγραφές της παραμετρικής αρχιτεκτονικής NRM SoC	21
3.2 Λειτουργία Συστήματος και Ροή Εργασιών	22
4. ΑΡΧΙΤΕΚΤΟΝΙΚΗ NRM ΣΥΣΤΗΜΑΤΟΣ ΣΕ ΨΗΦΙΔΑ	24
4.1 Δίκτυο σε Ψηφίδα - Network on Chip	24
4.2 Αρχιτεκτονική Πυρήνα NRM-SSA	24
4.3 Λειτουργία Πυρήνα και Ροή Δεδομένων	25
4.4 Διαφοροποίηση Πυρήνων FRM - NRM	27
5. ΑΡΧΙΤΕΚΤΟΝΙΚΗ NRM ΕΠΕΞΕΡΓΑΣΤΙΚΗΣ ΜΟΝΑΔΑΣ	28
5.1 NPM Processing Unit - NPU	28
5.2 NRM Processing Control Unit – NPCU	30
5.3 Processing Element - NPE	32
5.4 Dependency Graph (DG_MS)	33
5.5 Ροή Δεδομένων - NPU Datapath	36

6. Η ΚΕΝΤΡΙΚΗ ΜΟΝΑΔΑ ΕΠΕΞΕΡΓΑΣΙΑΣ - NRM PROCESSING ELEMENT	39
6.1 Βασική Λειτουργία και Ροή Δεδομένων	39
6.2 Πίνακας Χρόνων Ενεργοποίησης Αντιδράσεων	40
6.3 Δέντρο Συγκριτών – Comparator Tree (CoT)	41
6.4 Μονάδα Επανυπολογισμού – Recalculation Unit (RU)	43
7. ΕΠΙΔΟΣΕΙΣ - ΣΤΑΤΙΣΤΙΚΑ	47
7.1 Θεωρητική Ανάλυση Επιδόσεων	47
7.2 Στατιστικά Υλοποίησης	49
8. ΕΠΙΚΥΡΩΣΗ ΣΧΕΔΙΑΣΗΣ	55
8.1 Επικύρωση της Ορθότητας του NRM SoC	55
8.1.1 Lotka Volterra System Model	55
8.1.2 ASYN (Alpha-Synuclein) Model	57
8.2 Επικύρωση της Επίδοσης του NRM SoC	60
8.2.1 Linear Chain System (LCS)	60
8.2.2 Linear Chain System Extended (LCSE)	62
9. ΣΥΜΠΕΡΑΣΜΑΤΑ	66
ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ	67
ΣΥΝΤΜΗΣΕΙΣ, ΑΡΚΤΙΚΟΛΕΞΑ ΚΑΙ ΑΚΡΩΝΥΜΙΑ	68
ΑΝΑΦΟΡΕΣ	69

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1:	Ενδεικτικό παράδειγμα γράφου εξαρτήσεων αντιδράσεων [2]. . . .	19
Εικόνα 2:	Το ιστόγραμμα των επηρεαζόμενων αντιδράσεων του ASYN βιομοντέλου [3]. Ο άξονας y απεικονίζει τον αριθμό των αντιδράσεων του μοντέλου που έχουν το ίδιο πλήθος επηρεαζόμενων αντιδράσεων (D).	20
Εικόνα 3:	Ο χρήστης δίνοντας ως είσοδο το SBML [5] αρχείο του βιομοντέλου και το αρχείο παραμέτρων, θα λάβει ως έξοδο τα αποτελέσματα της επιθυμητής προσομοίωσης.	22
Εικόνα 4:	Η γενική αρχιτεκτονική δομή του NRM SoC. Το πλήθος των πυρήνων είναι παραμετρος του συστήματος.	24
Εικόνα 5:	Η αρχιτεκτονική ενός πυρήνας NRM-SSA.	26
Εικόνα 6:	Το RTL σχηματικό της μονάδας NPU.	28
Εικόνα 7:	Η αρχιτεκτονική της μονάδα NRM Processing Unit (NPU). . . .	29
Εικόνα 8:	Το RTL σχηματικό της μονάδας NPCU.	30
Εικόνα 9:	Τα σήματα εισόδου και εξόδου της μονάδας NPE.	32
Εικόνα 10:	Το RTL σχηματικό τη μονάδας DM_MS.	33
Εικόνα 11:	Ενδεικτικό παράδειγμα λειτουργίας του γράφου εξαρτήσεων. . . .	35
Εικόνα 12:	Η αρχιτεκτονική της μονάδα NRM Process Element (NPE). . . .	40
Εικόνα 13:	Ενδεικτικό παράδειγμα του πίνακα T και του δέντρου συγκριτών (CoT) ύψους $H = 3$	41
Εικόνα 14:	Η αρχιτεκτονική της Recalculation Unit.	45
Εικόνα 15:	Η ανάλυση της καθυστέρηση λόγω διοχέτευσης του πυρήνα NRM. . . .	47
Εικόνα 16:	Θεωρητικά αναμενόμενη διεκπεραιωτική ικανότητα (Reaction Throughput) του NRM SoC συναρτήσει του πλήθους των αντιδράσεων (m) και του μέσου όρου επηρεαζόμενων αντιδράσεων (D_{aver}). Συχνότητα λειτουργίας 200MHz.	49
Εικόνα 17:	Το διάγραμμα με τα στατιστικά υλοποίησης του NRM Core για $m = 512/1024$ και διαφορετικά D_{aver}	51
Εικόνα 18:	Το διάγραμμα με τα στατιστικά υλοποίησης του NRM Core για $m = 2048/4096$ και διαφορετικά D_{aver}	51
Εικόνα 19:	Το διάγραμμα με τα στατιστικά υλοποίησης του NRM SoC για $m = 512/1024$ και διαφορετικά D_{aver}	53

Εικόνα 20:	Το διάγραμμα με τα στατιστικά υλοποίησης του NRM SoC για $m = 2048/4096$ και διαφορετικά D_{aver}	53
Εικόνα 21:	Σύγκριση των αποτελεσμάτων του στοιχείου “Θήραμα” (Prey), NRM SoC vs Copasi.	56
Εικόνα 22:	Σύγκριση των αποτελεσμάτων του στοιχείου “Θηρευτή” (Predetar), NRM SoC vs Copasi.	56
Εικόνα 23:	Σύγκριση των αποτελεσμάτων του στοιχείου “Total Cytosolic WT ASYN Monomer”, NRM SoC vs Copasi.	58
Εικόνα 24:	Σύγκριση των αποτελεσμάτων του στοιχείου “Free Lamp2a”, NRM SoC vs Copasi.	58
Εικόνα 25:	Σύγκριση των αποτελεσμάτων του στοιχείου “Total Cytosolic WT ASYN Dimer”, NRM SoC vs Copasi.	58
Εικόνα 26:	Σύγκριση των αποτελεσμάτων του στοιχείου “Total Cytosolic WT ASYN Oligomers”, NRM SoC vs Copasi.	59
Εικόνα 27:	Σύγκριση της αναμενόμενη διεκπεραιωτική ικανότητα με την πραγματική απόδοση του συστήματος, εκτελώντας το LCS για $m = 512, 1024, 2048, 4096$ και $D_{aver} = 4$	61
Εικόνα 28:	Ο γράφος εξαρτήσεων (Dependencies Graph) του μοντέλου LCSE για τις πρώτες 8 αντιδράσεις με $D_{aver} = 5$	63
Εικόνα 29:	Σύγκριση της αναμενόμενη διεκπεραιωτική ικανότητα με την πραγματική απόδοση του συστήματος, εκτελώντας το LCSE για $m = 2048$ και $D_{aver} = \{17, 33, 65, 94\}$	65

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Όλα τα σήματα εισόδου και εξόδου του NRM Core, καθώς και οι γενικές παράμετροί του.	25
Πίνακας 2: Τα σήματα εισόδου και εξόδου της μονάδας NPU	28
Πίνακας 3: Τα σήματα εισόδου και εξόδου της μονάδας NPCU.	31
Πίνακας 4: Τα σήματα εισόδου και εξόδου της μονάδας NPE.	32
Πίνακας 5: Τα σήματα εισόδου και εξόδου της μονάδας DG_MS.	33
Πίνακας 6: Το πλήθος των απαιτούμενων κύκλων ρολογιού (L_{CoT}) συναρτήσει του πλήθους των αντιδράσεων m και του ύψους του δέντρου H	42
Πίνακας 7: Τα σήματα εισόδου και εξόδου της μονάδας Recalculation Unit (RU).	44
Πίνακας 8: Οι απαιτούμενοι κύκλοι ρολογιού των διακλαδώσεων της RU.	46
Πίνακας 9: Χαρακτηριστικά του FPGA XC7K325T.	49
Πίνακας 10: Τα στατιστικά υλοποίησης του NRM SoC ($C = 1$) για $m = n = 512$ και διαφορετικά D_{aver}	50
Πίνακας 11: Τα στατιστικά υλοποίησης του NRM SoC ($C = 1$) για $m = n = 1024$ και διαφορετικά D_{aver}	50
Πίνακας 12: Τα στατιστικά υλοποίησης του NRM SoC ($C = 1$) για $m = n = 2048$ και διαφορετικά D_{aver}	50
Πίνακας 13: Τα στατιστικά υλοποίησης του NRM SoC ($C = 1$) για $m = n = 4096$ και διαφορετικά D_{aver}	50
Πίνακας 14: Τα στατιστικά υλοποίησης του NRM SoC ($C = 8$) για $m = n = 512$ και διαφορετικά D_{aver}	52
Πίνακας 15: Τα στατιστικά υλοποίησης του NRM SoC ($C = 8$) για $m = n = 1024$ και διαφορετικά D_{aver}	52
Πίνακας 16: Τα στατιστικά υλοποίησης του NRM SoC ($C = 4$) για $m = n = 2048$ και διαφορετικά D_{aver}	52
Πίνακας 17: Τα στατιστικά υλοποίησης του NRM SoC ($C = 2$) για $m = n = 4096$ και διαφορετικά D_{aver}	53
Πίνακας 18: Οι γενικές παράμετροι του NRM SoC όπως αυτές ορίστηκαν για τα πειράματα του LVS.	56
Πίνακας 19: Οι γενικές παράμετροι του NRM SoC όπως αυτές ορίστηκαν για τα πειράματα του ASYN.	57
Πίνακας 20: Performance Evaluation - ASYN [3] Biomodel	59
Πίνακας 21: Οι γενικές παράμετροι του NRM SoC όπως αυτές ορίστηκαν για τα πειράματα του LCS.	61
Πίνακας 22: Performance Evaluation - LCS models $D_{aver} = 4$	61
Πίνακας 23: Οι γενικές παράμετροι του NRM SoC όπως αυτές ορίστηκαν για τα πειράματα του LCSE.	64
Πίνακας 24: Performance Evaluation - LCSE models $D_{aver} = \{17, 33, 65, 94\}$	64

ΠΡΟΛΟΓΟΣ

Η παρούσα διπλωματική εργασία εκπονήθηκε στην Αθήνα το 2015 στα πλαίσια του Διατμηματικού Μεταπτυχιακού Προγράμματος Σπουδών “Μικροηλεκτρονικής”, του τμήματος Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Αθηνών. Το περιεχόμενο της εργασίας είναι μέρος ερευνητικού έργου που πραγματοποιείται κάτω από την επίβλεψη του αναπληρωτή καθηγητή κ Ηλία Μανωλάκου. Τα αποτελέσματα τα οποία προέκυψαν από τη συγκεκριμένη έρευνα δημοσιεύτηκαν[6] και ελπίζουμε να αποτελέσουν τη βάση για την αναζήτηση καλύτερων μεθόδων στοχαστική προσομοίωσης βιομοριακών δικτύων αλληλεπίδρασης με τη χρήση υλικού επαναπρογραμματιζόμενης λογικής.

1. ΕΙΣΑΓΩΓΗ

1.1 Η Εξέλιξη της Βιολογίας

Μια από τις σημαντικότερες και ραγδαία εξελισσόμενες επιστήμες της εποχής μας, είναι η βιολογία. Η βιολογία εξετάζει τους μηχανισμούς της ζωής και πως αυτοί αλληλεπιδρούν. Η βαθύτερη κατανόηση των μηχανισμών αυτών θα οδηγήσει στη αποτελεσματικότερη αντιμετώπιση των ασθενειών που μαστίζουν το πλανήτη αλλά και τη βέλτιστη αξιοποίηση των φυσικών του πόρων [7]. Καθώς η βιοτεχνολογία αναπτύσσεται, ο άνθρωπος έχει τη δυνατότητα να μελετήσει τα βιολογικά συστήματα σε μεγαλύτερη κλίμακα αλλά και με μεγαλύτερη διακριτική ικανότητα. Η βιολογία έτσι επιμερίστηκε σε κλάδους, όπως η μικροβιολογία, η γενετική, και συνδέθηκε άρρηκτα και με συναφείς επιστήμες, όπως η βιοχημεία, η βιοφυσική, αλλά και η βιοστατιστική, κ.α. [8]. Κάθε ένας από τους κλάδους αυτούς επικεντρώνεται σε μια πτυχή της επιστήμης, αλλά όλοι μαζί συνθέτουν τη κοινή ευρεία βάση που αποκαλούμε “σύγχρονη βιολογία”.

Η σύγχρονη βιολογία στηρίζεται σε μεγάλο βαθμό στη Πληροφορική. Είναι γεγονός ότι οι σύγχρονες βιολογικές μελέτες μεγάλης κλίμακας παράγουν τεράστιο όγκο δεδομένων (Big Data) που απαιτούν ισχυρές υπολογιστικές μονάδες για την επεξεργασία τους και για την εξαγωγή χρήσιμων συμπερασμάτων [9]. Δύο νέοι συναφείς κλάδοι που αναπτύχθηκαν και θα ανθίσουν την τρέχουσα δεκαετία είναι η Υπολογιστική Βιολογία [10] και η Βιοπληροφορική [11]. Η Υπολογιστική Βιολογία επικεντρώνεται στην ανάπτυξη θεωρητικών μεθόδων και μαθηματικών μοντέλων για τη μελέτη της οργάνωσης, λειτουργίας και αλληλεπίδρασης βιολογικών συστημάτων σε διαφορετικά επίπεδα αφαίρεσης. Ενώ η Βιοπληροφορική ασχολείται κυρίως με την έρευνα και την ανάπτυξη κατάλληλων εργαλείων και προσεγγίσεων για την αποδοτικότερη μελέτη και συνδυασμένη ανάλυση και ερμηνεία των παραγόμενων δεδομένων των βιολογικών πειραμάτων. Για τους σκοπούς αυτούς, η Υπολογιστική Βιολογία και η Βιοπληροφορική αξιοποιούν μεθόδους κλάδων όπως η ανάλυση αλγορίθμων, οι βάσεις δεδομένων, η εξόρυξη γνώσης [12], η υπολογιστική θεωρία, η θεωρία ελέγχου [13], η θεωρία συστημάτων, η μικροηλεκτρονική [14], η μοντελοποίηση και προσομοίωση συστημάτων, η τεχνητή νοημοσύνη [15], η αναγνώριση προτύπων [16], η επεξεργασία σημάτων, η ανάπτυξη συστημάτων υλικού κ.α.

1.2 Βιολογικά Δίκτυα – Μέθοδοι Προσομοίωσης

Η σύγχρονη βιολογία απαιτεί την ανάπτυξη υπολογιστικών μοντέλων που να δίνουν τη δυνατότητα μελέτης in-silico (με χρήση υπολογιστή) της δομής και λειτουργίας τόσο των κυττάρων όσο και των πολυκυττάρων οργανισμών [17], με σκοπό την εκτέλεση πειραμάτων μέσω υπολογιστή. Ενισχύεται, λοιπόν, συνεχώς η ανάγκη για ανάπτυξη κατάλληλων μεθόδων και εργαλείων που θα μπορούν να προσομοιώνουν τη δυναμική συμπεριφορά των δικτύων αλληλεπιδράσεων βιομορίων σε διάφορα επίπεδα αφαίρεσης, κάτι που αποτελεί το κατεξοχήν αντικείμενο μελέτης του πλέον πρόσφατου κλάδου της Συστημικής Βιολογίας (Systems Biology) [18] [19].

Μέχρι σήμερα γνωρίζουμε δύο βασικές μεθόδους προσομοίωσης της δυναμικής συμπεριφοράς δικτύων βιοχημικών αντιδράσεων. Η πρώτη βασίζεται στη χρήση Συνήθων Διαφορικών Εξισώσεων (ODEs), ενώ η δεύτερη στηρίζεται σε αλγόριθμους στοχαστικής προσομοίωσης (Stochastic Simulation). Η χρήση Συνήθων Διαφορικών Εξισώσεων (ODEs) στηρίζεται σε δύο απλουστευτικές παραδοχές που δε συναντάμε στη φύση [20]. Η πρώτη είναι ότι το βιολογικό σύστημα που μελετάμε έχει ντετερμινιστική συμπεριφορά. Οι ντετερμινιστικές μέθοδοι παράγουν πάντα την ίδια έξοδο, αγνοώντας στοχαστικά φαινόμενα του περιβάλλοντος όπως ο “βιολογικός θόρυβος”. Η δεύτερη παραδοχή αφορά την ποσοτικοποίηση των μοριακών ειδών που αλληλεπιδρούν. Ενώ γνωρίζουμε ότι στη φύση το

πλήθος ενός μοριακού είδους είναι πάντα ένας φυσικός αριθμός και συνεπώς κινείται σε ένα φάσμα διακριτών τιμών, στις προσομοιώσεις με διαφορικές εξισώσεις η συνθήκη αυτή δεν εξασφαλίζεται πάντα. Λαμβάνοντας υπόψιν τις δύο αυτές απλουστευτικές παραδοχές, συμπεραίνουμε ότι τα αποτελέσματα των ντετερμινιστικών προσομοιώσεων με τη χρήση ODEs ενδέχεται να διαφέρουν από τη πραγματική συμπεριφορά ενός βιολογικού μοντέλου, ιδιαίτερα όταν αυτό καλείται να περιγράψει έντονα στοχαστικά φαινόμενα, όπως π.χ. η μεταγραφή (transcription) γονιδίων σε ένα ρυθμιστικό δίκτυο (gene regulatory network).

Σε αντίθεση με τις ντετερμινιστικές μεθόδους, οι αλγόριθμοι στοχαστικής προσομοίωσης (stochastic simulation algorithms, SSA) δεν χρειάζεται να στηριχθούν σε απλουστευτικές υποθέσεις και άρα πληρούν καλύτερα τις προϋποθέσεις του φυσικού κόσμου [20]. Οι αλγόριθμοι αυτοί εμπεριέχουν τουλάχιστον ένα στοιχείο στοχαστικότητας (π.χ. χρησιμοποιούν γεννήτριες "τυχαίων" αριθμών) και χρησιμοποιούν διακριτές τιμές για τους μοριακούς πληθυσμούς που μπορεί να είναι και πολύ μικρές (low species counts) όπως στο κύτταρο. Αν και η στοχαστική προσομοίωση είναι πιο κοντά στην πραγματική συμπεριφορά ενός βιολογικού συστήματος, για να θεωρηθούν τα αποτελέσματα της ικανοποιητικά, απαιτείται πολύ μεγάλος αριθμός στοχαστικών επαναλήψεων της προσομοίωσης ενός μοντέλου, με βάση μεθόδους Monte Carlo [21]. Ως εκ τούτου, η διαδικασία της στοχαστικής προσομοίωσης είναι πολύ χρονοβόρα και μπορεί να παράξει τεράστιο όγκο αποτελεσμάτων (εκατοντάδες με χιλιάδες καμπύλες μοριακών πληθυσμών). Τέλος, ένα ακόμα βασικό γνώρισμα των μεθόδων στοχαστικής προσομοίωσης είναι οι πεπερασμένες καταστάσεις στις οποίες το σύστημα μπορεί να μεταβεί. Η μετάβαση από τη μία κατάσταση στην άλλη πραγματοποιείται μέσω στοχαστικής διαδικασίας που μπορεί να μοντελοποιηθεί με Μαρκοβιανές αλυσίδες [22].

Στη παρούσα ερευνητική εργασία εξετάζουμε έναν δημοφιλή αλγόριθμο στοχαστικής προσομοίωσης που βασίζεται στην Μαρκοβιανή προσέγγιση. Ο αλγόριθμος στοχαστικής προσομοίωσης Next Reaction Method (NRM), των Gibson και Bruck [1], αποτελεί μια πρόσφατη βελτιωμένη μορφή του αλγορίθμου First Reaction Method (FRM) του Gillespie [20]. Ενώ ο αλγόριθμος FRM μπορεί να παραλληλοποιηθεί αποδοτικά, έχει υπολογιστική πολυπλοκότητα $O(m)$, όπου m ο αριθμός των αντιδράσεων του βιολογικού δικτύου, αντίθετως, ο αλγόριθμος NRM έχει μικρότερη υπολογιστική πολυπλοκότητα, $O(\log(m))$, αλλά είναι πολύ πιο δύσκολο να παραλληλοποιηθεί, γιατί στηρίζεται στη διατήρηση μιας κεντρικοποιημένης δομής δεδομένων (shared data structure) προκειμένου να μειώσει τους απαιτούμενους υπολογισμούς.

1.3 Διαθέσιμη Βιβλιογραφία - Κίνητρα

Μελετώντας τη σύγχρονη βιβλιογραφία και τις ήδη υπάρχουσες αρχιτεκτονικές για την υλοποίηση των αλγορίθμων FRM – NRM, διακρίνουμε την ανάγκη για υλοποίηση ενός αποδοτικού και πλήρως παραμετροποιημένου Συστήματος σε Ψηφίδα (System on Chip, SoC), χρησιμοποιώντας υλικό επαναπρογραμματιζόμενης λογικής (Field Programmable Gate Array, FPGA), το οποίο θα έχει τη δυνατότητα να προσομοιώνει παράλληλα και αποδοτικά πολύπλοκα δίκτυα βιομοριακών αλληλεπιδράσεων.

Με βάση τη διαθέσιμη βιβλιογραφία [23] [24] [25] [26] [27] [28] [29] [30] παρατηρούμε ότι οι ήδη υπάρχουσες υλοποιήσεις αντιμετωπίζουν διαφόρων ειδών προβλήματα που τους περιορίζουν τόσο στην απόδοση των συστημάτων όσο και στο μέγεθος των βιομοριακών μοντέλων που μπορούν να υποστηρίξουν.

Πιο συγκεκριμένα, για τον αλγόριθμο NRM εξετάσαμε τις δύο διαθέσιμες υλοποιήσεις στις αναφορές [24][26] και [25]. Στην πρώτη υλοποίηση [24][26] παρατηρούμε μια αξιόλογη προσπάθεια να σχεδιαστεί ένα σύστημα στο οποίο τα δεδομένα του μοντέλου αποθηκεύ-

ονται σε μεμονωμένες δομές δεδομένων (Thread Private Units, TPU) και διαμοιράζονται ένα πλήθος από κοινές μονάδες υπολογισμού (Thread Shared Units, TSU). Η βασική ιδέα της αρχιτεκτονικής αυτής είναι η παράλληλη εκτέλεση στοχαστικών προσομοιώσεων για βιομοντέλα με λιγότερο από $m = 1024$ αντιδράσεις, στοχεύοντας έτσι στην αύξηση της απόδοσης του συστήματος. Όμως, καθώς το πλήθος των υπολογιστικών μονάδων αυξάνεται, αφενός μεν αυξάνεται το πλήθος των παράλληλων στοχαστικών προσομοιώσεων που μπορούν να εκτελεστούν, αφετέρου δε, δημιουργούνται προβλήματα δρομολόγησης (routing problems) και κατά συνέπεια μειώνεται η συχνότητα λειτουργίας και η απόδοση του συστήματος αντίστοιχα.

Η δεύτερη υλοποίηση [25] που μας απασχόλησε αποτελεί μια εξίσου ικανοποιητική υλοποίηση του αλγορίθμου NRM. Η συγκεκριμένη υλοποίηση βασίζεται στην σχεδίαση ενός συστήματος με πολλές παράλληλες υπολογιστικές μηχανές (Update Engines), κάθε μια υπεύθυνη για μία αντίδραση του μοντέλου που προσομοιώνεται. Η συγκεκριμένη υλοποίηση για μοντέλα με λιγότερες από $m = 64$ αντιδράσεις επιτυγχάνει απόδοση 1 Gsteps/sec, αλλά καθώς το πλήθος των αντιδράσεων αυξάνεται (με όριο της $m = 254$ αντιδράσεις) η απόδοση του συστήματος μειώνεται δραματικά και αυξάνονται ραγδαία οι απαιτήσεις σε υλικό (resources utilization).

1.4 Στόχος της Διπλωματικής Εργασίας

Έχοντας μελετήσει τη σχετική βιβλιογραφία, διακρίνουμε την ανάγκη για συστήματα στοχαστικής προσομοίωσης που θα λύσουν τα ήδη υπάρχοντα προβλήματα και θα αξιοποιούν αποδοτικότερα το διαθέσιμο υλικό. Η πρώτη προσπάθεια της ερευνητικής μας ομάδας [31] αποτέλεσε το βασικό πυλώνα για την περαιτέρω ενασχόλησή μας με τις εφαρμογές των αλγορίθμων στοχαστικής προσομοίωσης σε υλικό επαναπρογραμματιζόμενης λογικής. Βέβαια, σε μια πρόσφατη δημοσίευσή μας [32] καταφέραμε να υλοποιήσουμε ένα υψηλής απόδοσης λογισμικό στοχαστικής προσομοίωσης που βασίζεται στον αλγόριθμο FRM, χρησιμοποιώντας ένα σύστημα με πολλαπλούς πυρήνες (many cores).

Στα πλαίσια του ερευνητικού έργου, στο οποίο κατατάσσεται η παρούσα διπλωματική εργασία, στόχος μας είναι να επανασχεδιάσουμε το σύστημά μας ώστε να αυξήσουμε την απόδοση και την διεκπεραιωτική ικανότητά του, υλοποιώντας συστήματα σε ψηφίδα (System on Chip, SoC) τόσο για τον αλγόριθμο FRM όσο και για τον αλγόριθμο NRM. Η υλοποίηση του αλγορίθμου FRM έχει προηγηθεί και εκπονηθεί [33] επιτυχώς και αποτελεί τη βάση για την παρούσα διπλωματική εργασία.

Στην παρούσα διπλωματική εργασία στόχος μας είναι να αναπτύξουμε ένα αποδοτικό υπολογιστικό σύστημα που θα υλοποιεί τον αλγόριθμο Next Reaction Method (NRM), των Gibson και Bruck [1], και θα ανταπεξέρχεται πλήρως στις απαιτήσεις της στοχαστικής προσομοίωσης μοντέλων μεγάλης κλίμακας. Επιπλέον, θα δίνει στον χρήστη τη δυνατότητα να καθορίζει όλες τις επιθυμητές παραμέτρους, τόσο σε επίπεδο προσομοίωσης (χρόνος προσομοίωσης, χρόνος δειγματοληψίας αποτελεσμάτων, πλήθος επαναλήψεων κ.α.) όσο και σε επίπεδο υπολογιστικού συστήματος (πλήθος πυρήνων κτλ.). Για να επιτευχθεί ο στόχος μας σκοπεύουμε να αξιοποιήσουμε στο έπακρο τις δυνατότητες που μας προσφέρει το υλικό, εφαρμόζοντας τεχνικές πλήρους σωλήνωσης (pipeline), παράλληλης επεξεργασίας (parallel processing) και απόκρυψης της καθυστέρησης (latency hiding) σε επίπεδο υλικού.

1.5 Δομή της Διπλωματικής Εργασίας

Στις ενότητες που ακολουθούν δίνεται έμφαση στη τεχνική περιγραφή του συστήματος στοχαστικής προσομοίωσης που σχεδιάσαμε και αναπτύξαμε, βασιζόμενο στον αλγό-

ριθμο NRM, των Gibson και Bruck [1].

Συγκεκριμένα η διάρθρωση της διπλωματικής εργασίας είναι η εξής:

Ενότητα 2: Παρουσιάζεται η μοντελοποίηση βιολογικών δικτύων, αναλύονται τα βήματα του αλγορίθμου NRM-SSA και επισημαίνονται οι βασικές διαφορές του από τον FRM-SSA.

Ενότητα 3: Προδιαγράφεται η αρχιτεκτονική του συστήματος σε ψηφίδα και παρουσιάζεται η ροή εργασιών που απαιτούνται για την εκτέλεση μιας προσομοίωσης.

Ενότητα 4: Αναλύεται η αρχιτεκτονική και η λειτουργικότητα του NRM-SSA SoC, επισημαίνοντας τη διαφοροποίηση των συστημάτων FRM – NRM σε επίπεδο πυρήνα.

Ενότητα 5: Αναλύεται συνοπτικά η αρχιτεκτονική και η λειτουργικότητα των μονάδων από το υψηλότερο επίπεδο έως το επίπεδο της υπολογιστικής μονάδας του πυρήνα.

Ενότητα 6: Αναλύεται διεξοδικά η αρχιτεκτονική και η λειτουργικότητα των μονάδων που απαρτίζουν την κεντρική μονάδα επεξεργασίας του πυρήνα.

Ενότητα 7: Παρουσιάζονται τα στατιστικά αποτελέσματα της σύνθεσης του NRM-SSA SoC και πως αυτά εξαρτώνται από τις γενικές παραμέτρους του συστήματος. Η ενότητα ολοκληρώνεται με τη θεωρητική ανάλυση των επιδόσεων του συστήματος.

Ενότητα 8: Παρουσιάζονται οι επιδόσεις και τα αποτελέσματα πειραμάτων προσομοίωσης βιομοντέλων. Σχολιάζονται τα αποτελέσματα των προσομοιώσεων τόσο σε επίπεδο ορθότητας όσο και σε επίπεδο επιδόσεων σε σύγκριση με γνωστές πλατφόρμες προσομοίωσης.

Ενότητα 9: Στην τελευταία ενότητα εκθέτουμε τα συμπεράσματα που προέκυψαν από την παρούσα ερευνητική εργασία, παραθέτουμε τους προβληματισμούς μας και τις πιθανές μελλοντικές μας ενέργειες.

2. ΜΕΘΟΔΟΙ ΣΤΟΧΑΣΤΙΚΗΣ ΠΡΟΣΟΜΟΙΩΣΗΣ

2.1 Μοντελοποίηση Βιολογικών Δικτύων

Προκειμένου να προσομοιώσουμε ένα βιολογικό σύστημα, πρέπει πρώτα να περιγράψουμε τη συμπεριφορά του σε μορφή δικτύου βιοχημικών αντιδράσεων (βιομοντέλο). Ένα βιομοντέλο απαρτίζεται από n μοριακά είδη $\{S_1, \dots, S_n\}$ με αρχικούς πληθυσμούς $\{X_1, \dots, X_n\}$ που αλληλοεπιδρούν μέσω m αντιδράσεων $\{R_1, \dots, R_m\}$. Προκειμένου να απλοποιηθεί η ανάλυση θεωρούμε ότι όλα τα είδη είναι ομοιόμορφα κατανεμημένα μέσα σε έναν όγκο Ω στο εσωτερικό ενός κελιού μοναδιαίου μεγέθους. Η υπόθεση αυτή απλοποιεί τους υπολογισμούς, αγνοώντας τις χωρικές επιπτώσεις που υπάρχουν στον πραγματικό κόσμο. Ορίζοντας ως $X_i(t)$ τη συγκέντρωση του μοριακού είδους S_i τη χρονική στιγμή t , τότε η κατάσταση του συστήματος την χρονική στιγμή t περιγράφεται ως $\mathbf{X}(t) = (X_1(t), X_2(t), \dots, X_n(t))$ με αρχικές συνθήκες $\mathbf{X}_0(t) = \mathbf{x}_0$ κατά τη χρονική στιγμή $t = t_0$.

Η στοχαστική προσομοίωση ενός βιομοντέλου παρακολουθεί το παραπάνω διάνυσμα καταστάσεων των συγκεντρώσεων του δικτύου σε επιλεγμένα χρονικά διαστήματα, χωρίς τη ρητή επίλυση διαφορικών εξισώσεων από την οποία εξαρτάται η δυναμική του συστήματος. Με βάση την περιγραφή του μοντέλου, όταν μια αντίδραση R_μ ενεργοποιηθεί τότε η τρέχουσα κατάσταση \mathbf{x} του συστήματος ενημερώνεται και η νέα κατάσταση είναι η $\mathbf{X}(t + \tau) = \mathbf{x} + \mathbf{v}_\mu$. Θεωρούμε ότι $\mathbf{v}_\mu = (v_{1\mu}, \dots, v_{n\mu})$, όπου ο συντελεστής v_μ αντιπροσωπεύει την αλλαγή της στοιχειομετρίας του είδους S_i λόγω της ενεργοποίησης της αντίδρασης R_μ . Επίσης, κάθε αντίδραση R_μ σχετίζεται με ένα σταθερό ρυθμό πιθανότητας c_μ , ο οποίος είναι ανάλογος του ρυθμού της αντίδρασης και αντιστρόφως ανάλογος του όγκου Ω , όπως παρουσιάζεται στις εξισώσεις 7(α) και 7(β) της αναφοράς [20].

Η πιθανότητα ένας τυχαία επιλεγμένος συνδυασμός αντιδρώντων να αλληλεπιδράσει και να “πυροδοτήσει” μια αντίδραση R_μ στο επόμενο απειροελάχιστο χρονικό διάστημα $[t, t + dt]$ δίνεται από τον τύπο $c_\mu * dt$. Η τάση $a_\mu(\mathbf{x})$ (propensity) μιας αντίδρασης R_μ στην κατάσταση \mathbf{X} υπολογίζεται πολλαπλασιάζοντας τη σταθερά c_μ με το πλήθος των πιθανών συνδυασμών των μοριακών αντιδρώντων ειδών, όπως φαίνεται στις εξισώσεις (21) έως (26) της αναφοράς [20]. Έτσι, για τη δεύτερης τάξης αντιδράσεις με αντιδρώντα S_1 και S_2 , ισχύει ότι:

$$a_\mu = c_\mu * X_1 * X_2 \quad (1)$$

Η παραπάνω διαδικασία ακολουθεί την Μαρκοβιανή θεωρία, όπου η επόμενη κατάσταση εξαρτάται μόνο από την τρέχουσα. Προσομοιώνοντας κατά αυτόν τον τρόπο το μοντέλο, αποδίδονται οι διαδρομές των καταστάσεων $\mathbf{X}(t)$. Ο αρχικός αλγόριθμος στοχαστική προσομοίωσης (SSA) του Gillespie, που ονομάζεται Άμεση Μέθοδος (Direct Method, DM) [34], βασίζεται στην παραπάνω διατύπωση και επιταχύνει τη διαδικασία με την εισαγωγή μιας νέας συνάρτησης $p(\tau, \mu | \mathbf{x}, t)$, η οποία εκφράζει την πιθανότητα η επόμενη αντίδραση R_μ να συμβεί μέσα στο επόμενο απειροελάχιστο χρονικό διάστημα $[t, t + \tau]$, δεδομένου ότι η τρέχουσα κατάσταση του συστήματος είναι $\mathbf{X}(t) = \mathbf{x}$. Η διαδικασία αυτή πλεονεκτεί, διότι μπορεί να μεταβαίνει από τη μια χρονική στιγμή στην επόμενη χωρίς να χρειάζεται να υπολογιστούν οι ενδιάμεσες καταστάσεις, όπου καμία αντίδραση δεν συμβαίνει. Κατά αυτό τον τρόπο, η ενημέρωση των αντιδρώντων και των προϊόντων συμβαίνει σε διακριτές ποσότητες και η μέθοδος είναι εφαρμόσιμη ακόμα κι αν διατηρούνται σε χαμηλές τιμές οι συγκεντρώσεις των μοριακών ειδών, άλλο ένα αξιοσημείωτο πλεονέκτημα των στοχαστικών σε σχέση με τους ντετερμινιστικούς αλγορίθμους προσομοίωσης.

Αν και ο DM-SSA είναι αποδοτικός για μοντέλα μικρών βιοχημικών δικτύων, είναι δύσκολο να παραλληλοποιηθεί και απαιτεί πολύ περισσότερο χρόνο για μεσαία και μεγάλα μοντέλα. Λαμβάνοντας υπόψιν αυτούς τους περιορισμούς, ο Gillespie εισήγαγε ως εναλλακτική λύση έναν ισοδύναμο αλγόριθμο στοχαστικής προσομοίωσης, τον First Reaction

Method (FRM). Με βάση τον αλγόριθμο αυτόν, υπολογίζεται για κάθε αντίδραση R_μ ένας πιθανός (δυναμικός) χρόνος ενεργοποίησης τ_j . Το κανάλι αντίδρασης R_μ μικρότερο δυναμικό χρόνο τ_j προσδιορίζεται ως “νικητήρια αντίδραση” (winner reaction), η οποία θα ενεργοποιηθεί στο τέλος του “κύκλου αντιδράσεων” (Reaction Cycle, RC). Δεδομένου ότι ο υπολογισμός του χρόνου τ_j για κάθε αντίδραση είναι ανεξάρτητος από τους υπόλοιπους, ο αλγόριθμος επιδέχεται μεγάλο βαθμό παραλληλοποίησης.

Μια ενδιαφέρουσα παρατήρηση που προκύπτει μελετώντας τους αλγόριθμους του Gillespie αφορά την αλληλεξάρτηση των αντιδράσεων που χαρακτηρίζουν ένα βιομοντέλο. Παρατηρούμε ότι κατά την ενεργοποίηση μιας αντίδρασης επηρεάζονται οι συγκεντρώσεις ορισμένων μοριακών ειδών και συνεπώς απαιτείται ο επαναυπολογισμός μόνο των επηρεαζόμενων τάσεων $\alpha_j(\mathbf{x})$ και αντίστοιχα των χρόνων ενεργοποίησης των επηρεαζόμενων αντιδράσεων. Βασιζόμενοι σε αυτή τη κομβική παρατήρηση οι Gibson και Bruck [1] επινόησαν τον αλγόριθμο στοχαστική προσομοίωσης Next Reaction Method (NRM-SSA), με μικρότερη πολυπλοκότητα εκτέλεσης $O(\log(m))$ σε σύγκριση με τον αλγόριθμο FRM-SSA που έχει πολυπλοκότητα $O(m)$.

2.2 Ο Αλγόριθμος NRM των Gibson και Bruck

Ο αλγόριθμος στοχαστικής προσομοίωσης Next Reaction Method (NRM) τον οποίο καλούμαστε να υλοποιήσουμε είναι μια βελτιωμένη έκδοση του αλγορίθμου First Reaction Method (FRM).

Στην συνέχεια ακολουθούν τα βήματα [2] του αλγορίθμου NRM, των Gibson και Bruck:

1. Αρχικοποίηση του συστήματος:

- α. $t = t_0$ (καθορισμός των αρχικών πληθυσμών κάθε μοριακού είδους),
- β. $\mathbf{x} = \mathbf{x}_0$ αρχικοποίηση του χρόνου προσομοίωσης,
- γ. υπολογισμός του γράφου εξαρτήσεων,
- δ. αρχικοποίηση των τάσεων $\alpha_j(\mathbf{x})$ όλων των αντιδράσεων,
- ε. υπολογίζεται για κάθε αντίδραση η χρονική στιγμή ενεργοποίησης της αντίδρασης:

$$\tau_j = t + \frac{1}{\alpha_j(\mathbf{x})} * \ln\left(\frac{1}{r_j}\right) \quad (2)$$

όπου r_j είναι τυχαίοι αριθμοί στο διάστημα $[0, 1]$.

ζ. Αποθήκευση των τάσεων και των χρόνων τ_j .

2. Εύρεση της αντίδρασης R_μ με το μικρότερο χρόνο τ_μ ενεργοποίησης χρησιμοποιώντας μηχανισμούς όπως η ουρά προτεραιότητας, δέντρα συγκριτών κ.α.
3. Εκτέλεση της αντίδρασης R_μ ανανεώνοντας τη κατάσταση του συστήματος $\mathbf{x} = \mathbf{x} + \mathbf{v}_\mu$ (μεταβολή των πληθυσμών των μοριακών ειδών βάση του διανύσματος στοιχειομετρίας \mathbf{v}_μ της αντίδρασης R_μ) καθώς και το χρόνο προσομοίωσης $t = \tau$ όπου $\tau = \tau_\mu$.
4. Ανανέωση των χρόνων ενεργοποίησης των αντιδράσεων:

- α. Ανανέωση της χρονικής στιγμής ενεργοποίησης των επηρεαζόμενων αντιδράσεων με βάση τον μαθηματικό τύπο:

$$\tau_\mu = \left(\frac{1}{\alpha_\mu(\mathbf{x})}\right) * \ln\left(\frac{1}{r_\mu}\right) + t \quad (3)$$

- β. Υπολογισμός της χρονικής στιγμής ενεργοποίησης της αντίδρασης με τον μικρότερο χρόνο ενεργοποίησης με βάση τον μαθηματικό τύπο:

$$\tau_{\alpha} = \left(\frac{\alpha_{\alpha,old}}{\alpha_{\alpha}} \right) * (\tau_{\alpha,old} - t) + t \quad (4)$$

όπου r_j είναι ψευδοτυχαίος αριθμός στο διάστημα $[0, 1]$

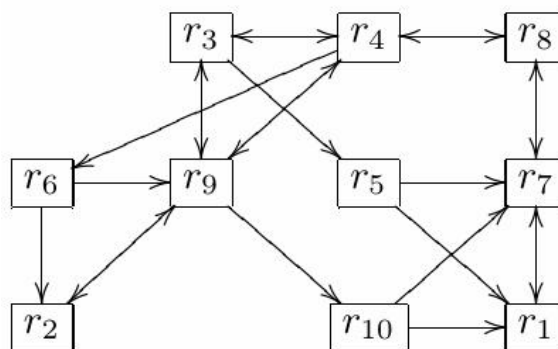
- Εάν το σύστημα έχει φτάσει τον επιθυμητό χρόνο προσομοίωσης τότε ο αλγόριθμος τερματίζει, αλλιώς μεταβαίνει ξανά στο βήμα 2.

2.3 Η Διαφοροποίηση του NRM-SSA έναντι του FRM-SSA

Προκειμένου να υλοποιηθεί ο NRM-SSA απαιτούνται τρεις βασικές αλλαγές [2] στον FRM-SSA. Η πρώτη αλλαγή αφορά την έννοια του χρόνου. Ο αλγόριθμος FRM υπολογίζει την σχετική απόσταση του χρόνου ενεργοποίησης της επόμενης αντίδρασης από την τρέχουσα χρονική στιγμή, σε αντίθεση με τον αλγόριθμο NRM που υπολογίζει την καθολική τιμή του χρόνου ενεργοποίησης όλων των αντιδράσεων. Η δεύτερη διαφοροποίηση σχετίζεται με την έννοια της εξάρτησης των αντιδράσεων. Με βάση τον αλγόριθμο των Gibson και Bruck, απαιτείται η κατασκευή ενός γράφου εξαρτήσεων, όπου για κάθε πιθανή νικητήρια αντίδραση να καθορίζονται οι αντιδράσεις για τις οποίες απαιτείται ο επαναυπολογισμός των τάσεων $\alpha_j(\mathbf{x})$ (propensity) και αντίστοιχα των χρόνων ενεργοποίησης τ_j . Τέλος, η τρίτη αλλαγή αφορά τις δομές δεδομένων που χρησιμοποιεί ο αλγόριθμος NRM έναντι του αλγορίθμου FRM. Δεδομένου ότι οι τάσεις $\alpha_j(\mathbf{x})$ και οι χρόνοι τ_j όλων των αντιδράσεων ενδέχεται να επαναχρησιμοποιηθούν στον επόμενο κύκλο αντιδράσεων, θα πρέπει να αποθηκευτούν σε κατάλληλες δομές δεδομένων ώστε να είναι πάντα διαθέσιμα.

2.4 Ο Γράφος Εξαρτήσεων - Dependencies Graph

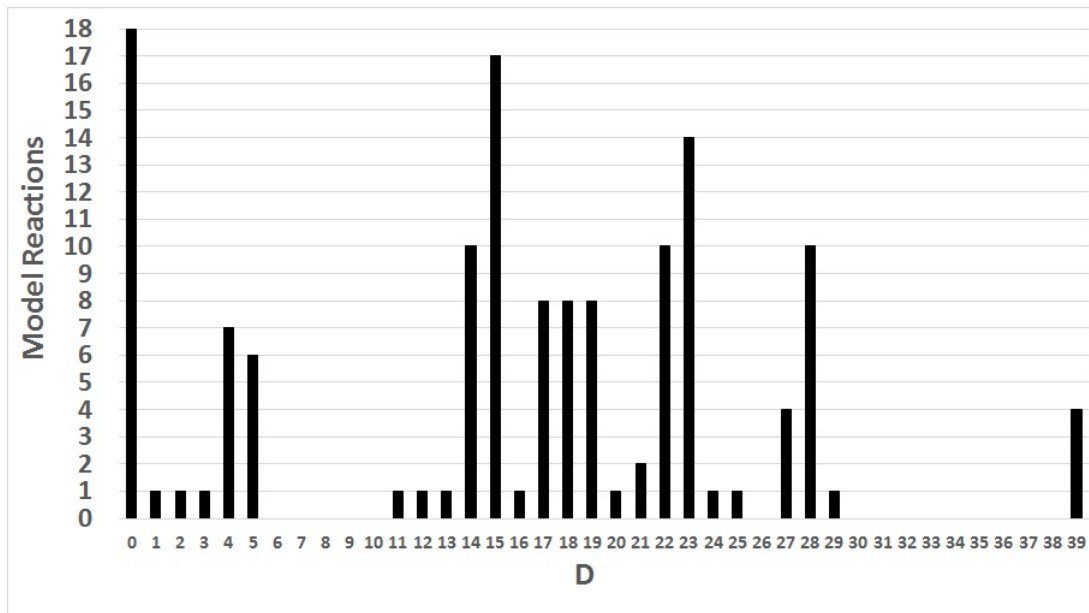
Η βασική διαφορά ανάμεσα στους αλγορίθμους στοχαστικής προσομοίωσης FRM και NRM είναι ότι ο δεύτερος κάνει χρήση γράφου εξαρτήσεων (Dependency Graph - DG) που απεικονίζει την εξάρτηση μεταξύ των αντιδράσεων. Όπως παρατήρησαν οι Gibson και Bruck, όταν μια αντίδραση συμβεί δεν επηρεάζονται οι συγκεντρώσεις όλων των μοριακών ειδών και συνεπώς δεν απαιτείται ο επαναυπολογισμός των τάσεων $\alpha_j(\mathbf{x})$ και των χρόνων τ_j όλων των αντιδράσεων.



Εικόνα 1: Ενδεικτικό παράδειγμα γράφου εξαρτήσεων αντιδράσεων [2].

Για τον λόγο αυτόν κατασκευάζεται ένας γράφος εξαρτήσεων όπου η ακμή από την αντίδραση με τον μικρότερο χρόνο προς άλλες αντιδράσεις υποδηλώνει ότι τουλάχιστον ένα αντιδρόν, ή ένα προϊόν, της “νικητήριας αντίδρασης” συμμετέχει ως αντιδρόν στις αντιδράσεις αυτές, τις λεγόμενες “επηρεαζόμενες αντιδράσεις”. Επομένως, σε κάθε κύκλο προσομοίωσης χρησιμοποιούμε τον γράφο εξαρτήσεων ώστε να εξάγουμε τις επηρεαζόμενες αντιδράσεις και να επαναπροσδιορίσουμε τον χρόνο ενεργοποίησης μόνο αυτών των επηρεαζόμενων αντιδράσεων. Στην Εικόνα 1 παρουσιάζεται ένα ενδεικτικό παράδειγμα γράφου εξαρτήσεων για ένα τυχαίο σύνολο αντιδράσεων.

Αναλύοντας ένα πραγματικό μοντέλο και σχεδιάζοντας το γράφο εξαρτήσεων μεταξύ των αντιδράσεων, θα διαπιστώσουμε ότι ο γράφος δεν είναι πάντα συμμετρικός καθώς δεν επηρεάζουν όλες οι αντιδράσεις το ίδιο πλήθος αντιδράσεων (D). Στην Εικόνα 2, δίδεται ως παράδειγμα το ιστόγραμμα του πλήθους των εξαρτήσεων ενός πραγματικού βιομοντέλου, του ASYN [3] [4]. Το ASYN μοντέλο χαρακτηρίζεται από $m = 136$ αντιδράσεις και $n = 90$ μοριακά είδη και προσομοιώνει τις επιβλαβείς επιδράσεις του πολυμερισμού της πρωτεΐνης άλφα-συνουκλείνης (ASYN) στην ομοιόσταση του κυττάρου, που πιστεύεται ότι συμβάλει στην εμφάνιση της νόσου του Πάρκινσον. Όπως παρουσιάζεται στο ιστόγραμμα της Εικόνας 2, από το σύνολο των αντιδράσεων: 18 αντιδράσεις επηρεάζουν $D = 0$ αντιδράσεις και 4 αντιδράσεις επηρεάζουν $D = 39$ αντιδράσεις, ενώ τελικά ο μέσος όρος επηρεαζόμενων αντιδράσεων είναι $D_{aver} = 16$.



Εικόνα 2: Το ιστόγραμμα των επηρεαζόμενων αντιδράσεων του ASYN βιομοντέλου [3]. Ο άξονας y απεικονίζει τον αριθμό των αντιδράσεων του μοντέλου που έχουν το ίδιο πλήθος επηρεαζόμενων αντιδράσεων (D).

Παρότι ο NRM-SSA, σε σύγκριση με τον FRM-SSA, κερδίζει σε απόδοση, είναι πολύ δύσκολο να παραλληλοποιηθεί σε επίπεδο υλοποίησης. Υπάρχουν περιπτώσεις όπου μια αντίδραση μπορεί να επηρεάσει όλες τις υπόλοιπες αντιδράσεις ενός βιομοντέλου. Σε αυτή την περίπτωση ο αλγόριθμος NRM χάνει το πλεονέκτημά του αφού πλέον απαιτούνται $O(m)$ βήματα εκτέλεσης για την ολοκλήρωση ενός κύκλου αντιδράσεων [2]. Ως εκ τούτου, ο FRM-SSA προτιμάται, στις περισσότερες παράλληλες υλοποιήσεις στοχαστικής προσομοίωσης, ενώ ο NRM-SSA προτιμάται για μοντέλα με πάρα πολλές αντιδράσεις και μικρό σχετικά μέσο όρο επηρεαζόμενων αντιδράσεων (D_{aver}).

Η διαθεσιμότητα εναλλακτικών υλοποιήσεων και για τους δύο αλγορίθμους διευκολύνει τη βέλτιστη επιλογή που εξαρτάται από το πλήθος των αντιδράσεων του μοντέλου (m) και από τα χαρακτηριστικά του ιστογράμματος εξαρτήσεων του μοντέλου (D_{aver}). Εν τέλει, στην ιδανική περίπτωση, ένας προσομοιωτής θα πρέπει να επιλέξει τον κατάλληλο αλγόριθμο προσομοίωσης με βάση την πολυπλοκότητα του μοντέλου και το ιστόγραμμα εξαρτήσεων, ώστε να μεγιστοποιείται η αποδοτικότητα του συστήματος στοχαστικής προσομοίωσης με ευέλικτο τρόπο.

3. ΠΡΟΔΙΑΓΡΑΦΕΣ ΣΥΣΤΗΜΑΤΟΣ

Στην ενότητα αυτή θα παρουσιαστούν οι προδιαγραφές και ο τρόπος λειτουργίας του παραμετρικού συστήματος NRM-SSA SoC που σχεδιάστηκε και αναπτύχθηκε με βάση το είδη υπάρχον υλικό για το σύστημα FRM [33]. Ο τρόπος λειτουργίας είναι παρόμοιος με αυτόν για το παραμετρικό FRM-SSA SoC, αλλά δεδομένου ότι είναι διαφορετικός ο αλγόριθμος που υλοποιείται, παρουσιάζονται σημαντικές διαφορές στις βασικές παραμέτρους τις οποίες και θα περιγράψουμε στη συνέχεια.

Το σύστημα απαρτίζεται από έναν γενικού σκοπού ηλεκτρονικό υπολογιστή (Host-PC) που αφενός φιλοξενεί τα δεδομένα του μοντέλου που θα προσομοιωθεί, και αφετέρου συνδέεται με μια αναπτυξιακή κάρτα που ενσωματώνει ένα ολοκληρωμένο κύκλωμα επαναπρογραμματιζόμενης λογικής (Field Programmable Gate Array, FPGA). Στην περίπτωση μας η αναπτυξιακή κάρτα είναι η KC705 της εταιρίας Xilinx [35], που ενσωματώνει ένα μεσαίου μεγέθους FPGA, το XC7K325-2FFG900C [36], με 203,800 LUTs, 445 BRAMs (36Kbit) και 840 DSPs. Για την ανάπτυξη του συστήματος χρησιμοποιήθηκαν εργαλεία της ίδιας εταιρίας και ο σχεδιασμός πραγματοποιήθηκε χρησιμοποιώντας τη γλώσσα περιγραφής υλικού VHDL. Τέλος, για την επικοινωνία του Host-PC με το NRM-SSA SoC χρησιμοποιείται το πρωτόκολλο επικοινωνίας PCIe.

3.1 Προδιαγραφές της παραμετρικής αρχιτεκτονικής NRM SoC

Ο πυρήνας NRM-SSA χαρακτηρίζεται από γενικές παραμέτρους που μπορούν ευέλικτα να καθορίζουν τη δομή του, τη συμπεριφορά του και το μέγεθος των βιομοντέλων που μπορεί να εκτελέσει με βάση τις παραμέτρους προσομοίωσης του χρήστη. Η δυνατότητα της παραμετροποίησης του πυρήνα κατά το στάδιο της σύνθεσης τον κατατάσσει στους “μαλακούς πυρήνες” (Soft IP Core) ειδικού σκοπού. Το συνολικό σύστημα NRM-SSA SoC είναι μια πολυπύρηνη αρχιτεκτονική, 32-bit, πλήρους διοχέτευσης (fully pipelined) με τη δυνατότητα να εμπεριέχονται ένας ή περισσότεροι διασυνδεδεμένοι πυρήνες στο ανώτερο επίπεδο.

Όπως προαναφέρθηκε το σύστημα είναι παραμετρικό σύμφωνα με τα χαρακτηριστικά ενός βιολογικού δικτύου. Οι παράμετροι δίνονται ως generic μεταβλητές στο υψηλότερο αρχείο (top-level) της υλοποίησης του SoC και κληρονομούνται σε κάθε επίπεδο του σχεδίου. Στη συνέχεια ακολουθούν τα χαρακτηριστικά που παραμετροποιούν την αρχιτεκτονική NRM SoC:

- Αριθμός αντιδράσεων – m . Η παράμετρος αυτή καθορίζει το βάθος της μνήμης όπου αποθηκεύονται οι αντιδράσεις του δικτύου και αντίστοιχα το μέγιστο επιτρεπόμενο πλήθος των αντιδράσεων ενός μοντέλου. Οι αντιδράσεις μπορεί να είναι μέχρι και 3ου βαθμού με 5 προϊόντα.
- Αριθμός μοριακών ειδών – n . Η παράμετρος αυτή καθορίζει το βάθος των μνημών όπου αποθηκεύονται τα μοριακά είδη και αντίστοιχα το μέγιστο επιτρεπόμενο πλήθος διαφορετικών μοριακών ειδών ενός μοντέλου.
- Μέσος όρος επηρεαζόμενων αντιδράσεων – D_{aver} . Η παράμετρος αυτή καθορίζει το πλάτος της μνήμης όπου αποθηκεύεται ο γράφος εξαρτήσεων και αντιστοιχεί στο μέγιστο επιτρεπόμενο μέσο όρο επηρεαζόμενων αντιδράσεων ενός μοντέλου.
- Αριθμός NRM πυρήνων – C . Η παράμετρος C (COREs) καθορίζει τον αριθμό των παράλληλων NRM πυρήνων που θα διαθέτει το σύστημα μετά τη σύνθεση, όπου κάθε πυρήνας μπορεί να εκτελεί ανεξάρτητες προσομοιώσεις ενός βιομοντέλου.

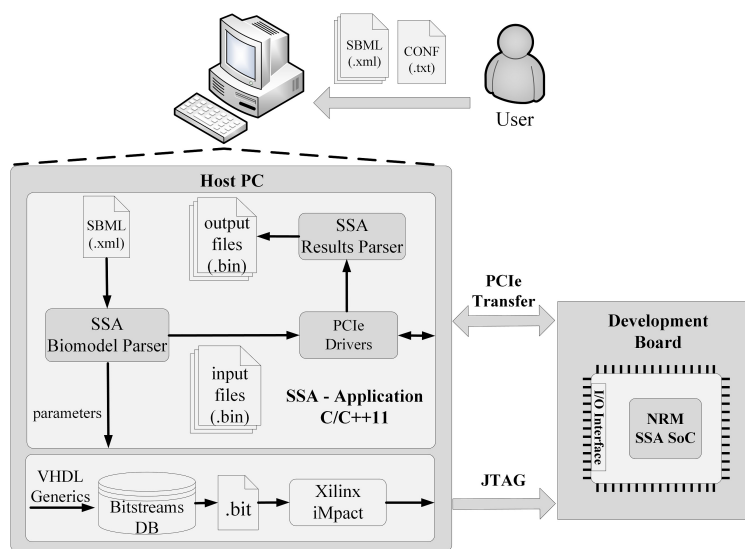
Κάθε μία από τις παραπάνω παραμέτρους καθορίζει τον τρόπο λειτουργίας του συστήματος, τόσο σε επίπεδο προσομοίωσης όσο και σε επίπεδο αρχιτεκτονικής. Αφενός οι

παράμετροι m , n και D_{aver} καθορίζουν τις απαιτήσεις ενός πυρήνα σε υπολογιστικούς πόρους και επιφάνεια με βάση τα χαρακτηριστικά μεγέθη ενός μοντέλου, και αφετέρου η παράμετρος C καθορίζει πόσοι πυρήνες θα εμπεριέχονται στο σύστημα μετά τη σύνθεση. Το πλήθος των πυρήνων εξαρτάται από το τελικό μέγεθος του πυρήνα με βάση τις υπολοίπες παραμέτρους αλλά και από τους διαθέσιμους πόρους του επιλεγμένου FPGA. Το σύστημα έχει σχεδιαστεί ώστε να μπορεί να επεκταθεί χωρίς περιορισμούς ανάλογα με το μέγεθος του FPGA που διαθέτει ο χρήστης.

3.2 Λειτουργία Συστήματος και Ροή Εργασιών

Το Host-PC είναι υπεύθυνο για την ομαλή διεξαγωγή της προσομοίωσης και, συνοπτικά, η εκτέλεση ενός βιομοντέλου ακολουθεί την εξής διαδικασία: αρχικά, με τη χρήση κατάλληλα υλοποιημένου λογισμικού μεταφέρονται στο NRM-SSA SoC τα δεδομένα του μοντέλου και οι επιθυμητές παράμετροι που ορίζει ο χρήστης. Δίνεται από το Host-PC το σήμα εκκίνησης της προσομοίωσης. Καθ' όλη τη διάρκεια της προσομοίωσης, το NRM-SSA SoC παράγει πληροφορίες οι οποίες συλλέγονται και αποθηκεύονται στο Host-PC. Με βάση τις πληροφορίες αυτές, παράγονται ένα ή περισσότερα αρχεία με τα αποτελέσματα και τα στατιστικά της προσομοίωσης. Ο χρήστης έχει τη δυνατότητα να επισκοπήσει τα αποτελέσματα, να αποθηκεύσει την επιθυμητή πληροφορία και εν τέλει να εκτελέσει μια νέα προσομοίωση με νέες παραμέτρους, δεδομένα ή αρχικές τιμές.

Η παραπάνω διαδικασία χωρίζεται σε 3 φάσεις και απεικονίζεται συνοπτικά στην Εικόνα 3.



Εικόνα 3: Ο χρήστης δίνοντας ως είσοδο το SBML [5] αρχείο του βιομοντέλου και το αρχείο παραμέτρων, θα λάβει ως έξοδο τα αποτελέσματα της επιθυμητής προσομοίωσης.

Στη πρώτη φάση ο χρήστης διαμορφώνει τη μορφή της αρχιτεκτονικής μέσω των γενικών παραμέτρων στο υψηλότερο (top-level) αρχείο σχεδίασης VHDL. Αφού καθορίσει τις γενικές παραμέτρους της αρχιτεκτονικής μέσω του λογισμικού ISE [37] παράγει το δυαδικό αρχείο bitstream [38] - το οποίο περιέχει τη περιγραφή της αρχιτεκτονικής σε επίπεδο κυκλώματος. Έπειτα, αφού παραχθεί το bitstream ο χρήστης μέσω του λογισμικού iMPact [39] προγραμματίζει το FPGA μεταφέροντας το bitstream μέσω του πρωτοκόλλου επικοινωνίας JTAG [40]. Εδώ θα πρέπει να τονιστεί ότι σκοπός της παρούσας έρευνας είναι τελικά να δημιουργηθεί μια βιβλιοθήκη από bitstreams προκειμένου το βήμα της παραγωγής του bitstream (πρώτη φάση) να αποφεύγεται καθώς αποτελεί χρονοβόρα διαδικασία. Στόχος είναι να απαιτείται να δημιουργηθεί ένα νέο bitstream μόνο σε εξαιρετικές περιπτώσεις όπου οι ανάγκες κάποιου χρήστη εξειδικεύονται τόσο που επιβάλουν το βήμα αυτό. Στις

περισσότερες περιπτώσεις οι παράμετροι του χρήστη χρησιμοποιούνται για την επιλογή του πλέον κατάλληλου bitstream (SoC instance) από αυτά που έχουν ήδη δημιουργηθεί εκ των προτέρων αξιοποιώντας τη παραμετρική φύση των SoC περιγραφών.

Με την ολοκλήρωση της φάσης του προγραμματισμού του FPGA ο χρήστης μπορεί να προχωρήσει στη δεύτερη φάση που περιλαμβάνει τον προγραμματισμό του NRM SoC συστήματος με το επιθυμητό βιομοντέλο και την εκτέλεση του in-silico πειράματος. Σε αυτή τη φάση ο χρήστης αποθηκεύει στο Host-PC το βιομοντέλο που επιθυμεί να προσομοιώσει σε standard SBML κωδικοποίηση [5]. Το SBML αρχείο περιέχει το σύνολο αντιδράσεων που περιγράφουν το βιολογικό μοντέλο καθώς και τα μοριακά είδη και τους αρχικούς πληθυσμούς τους. Τα στοιχεία αυτά πρέπει να μετατραπούν σε κατάλληλες δομές δεδομένων που αναγνωρίζει το NRM SoC σε δυαδική μορφή.

Τη μετατροπή αυτή την αναλαμβάνει ένα εργαλείο λογισμικού που αναπτύχθηκε στα πλαίσια του έργου και ονομάζεται SBML Biomodel Parser. Εκτός από τις αντιδράσεις του βιομοντέλου και τα μοριακά είδη θα πρέπει να ρυθμιστούν και οι παράμετροι που σχετίζονται με την εκτέλεση της προσομοίωσης του βιομοντέλου. Αυτές οι παράμετροι ρυθμίζονται από τον χρήστη μέσω του Αρχείου Ρυθμίσεων (configuration file) και είναι οι εξής:

- T_{sim} : Χρόνος ολοκλήρωσης της προσομοίωσης σε δευτερόλεπτα
- T_{sam} : Περίοδος δειγματοληψίας των αποτελεσμάτων σε δευτερόλεπτα.
- **Στοχαστικές Επαναλήψεις Προσομοίωσης (Repetitions - R)**: αριθμός των στοχαστικών επαναλήψεων που θα εκτελεστεί το πείραμα.
- **Μέγιστος Αριθμός Εσωτερικών Βημάτων (Max Internal Steps)**: μέγιστος αριθμό κύκλων αντιδράσεων (reaction cycles) που επιτρέπεται να μεσολαβήσουν κατά τη διάρκεια μιας περιόδου δειγματοληψίας προτού τερματιστεί μια επανάληψη του πειράματος.

Αφού συμπληρωθεί και το Αρχείο Ρυθμίσεων, τροφοδοτείται μαζί με το SBML αρχείο στο εργαλείο SBML Biomodel Parser προκειμένου να παραχθούν τα κατάλληλα δυαδικά αρχεία. Όταν οριστικοποιηθούν όλα τα απαραίτητα αρχεία δίδεται το σήμα εκκίνησης της προσομοίωσης. Καθ' όλη τη διάρκεια της προσομοίωσης το NRM-SSA SoC παράγει σε κάθε περίοδο δειγματοληψίας αποτελέσματα (πληθυσμοί μοριακών ειδών) τα οποία αποστέλλονται στο Host-PC και αποθηκεύονται σε ένα αρχείο αποτελεσμάτων (results.bin).

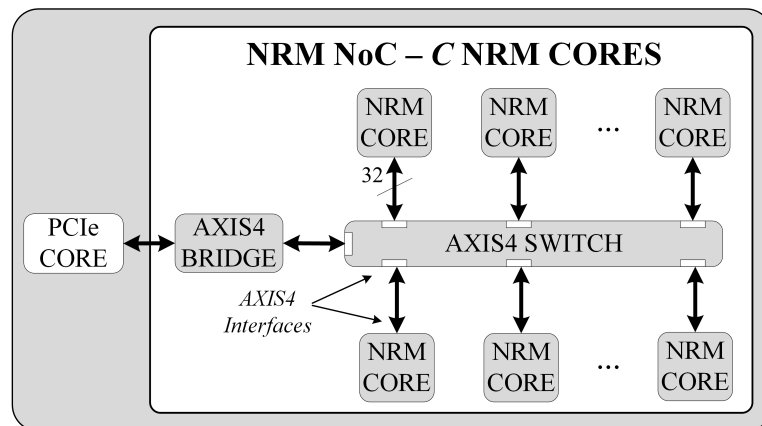
Με την ολοκλήρωση της προσομοίωσης ξεκινάει η τρίτη φάση του συστήματος η οποία περιλαμβάνει την ανάλυση των αποτελεσμάτων. Τα αποτελέσματα που παράγονται από τους NRM πυρήνες μεταφέρονται μέσω του PCIe πρωτοκόλλου στο Host PC όπου και αποθηκεύονται σε ένα δυαδικό αρχείο (results.bin). Με το τέλος της προσομοίωσης το αρχείο results.bin τροφοδοτείται σε ένα άλλο εργαλείο λογισμικού που αναπτύχθηκε στα πλαίσια του έργου και είναι υπεύθυνο για την ανάλυση των δεδομένων. Το εργαλείο, SSA Results Parser, αρχικά ξεχωρίζει τα δεδομένα που παρήγαγε ο κάθε NRM πυρήνας καθώς είναι δυνατόν ο κάθε πυρήνας να εκτελεί ένα τελείως διαφορετικό πείραμα. Έπειτα για κάθε πείραμα που διεξήχθη, το εργαλείο SSA Results Parser υπολογίζει μέσες τιμές, ελάχιστα και μέγιστα για όλους τους μοριακούς πληθυσμούς, εφόσον έχουν εκτελεστεί περισσότερες από μια στοχαστικές επαναλήψεις. Τα στατιστικά αυτά μαζί με τα αποτελέσματα κάθε επανάληψης αποθηκεύονται σε CSV αρχεία [41] προκειμένου να δίνεται η δυνατότητα στον χρήστη να τα αναλύσει περαιτέρω με άλλα λογισμικά. Επιπλέον το εργαλείο δημιουργεί CSV αρχεία με στατιστικά που σχετίζονται με τη προσομοίωση όπως: ο συνολικός αριθμός κύκλων προσομοίωσης που εκτελέστηκαν, ο συνολικός αριθμός περιόδων δειγματοληψίας που καταγράφηκαν καθώς και ο χρόνος που χρειάστηκε για να ολοκληρωθεί η κάθε επανάληψη.

4. ΑΡΧΙΤΕΚΤΟΝΙΚΗ NRM ΣΥΣΤΗΜΑΤΟΣ ΣΕ ΨΗΦΙΔΑ

4.1 Δίκτυο σε Ψηφίδα - Network on Chip

Όπως έχει αναφερθεί και στα προηγούμενα κεφάλαια, στόχος μας είναι η αρχιτεκτονική του NRM SoC να είναι ευέλικτη και όσο το δυνατόν περισσότερο παραμετρική. Οι γενικές παράμετροι (generic parameters) που σχετίζονται με την αρχιτεκτονική του NRM SoC αφορούν τόσο τα υποστηριζόμενα μοντέλα βιολογικών δικτύων όσο και τη δομή του συστήματος που ενσωματώνει έναν ή περισσότερους πυρήνες.

Το NRM SoC είναι σχεδιασμένο να περιέχει περισσότερους από έναν πυρήνες και για το λόγο αυτό ορίζεται η γενική παράμετρος C (COREs) ώστε να καθορίζεται, πριν τη σύνθεση, το πλήθος των πυρήνων NRM που θα εμπεριέχονται στο NRM SoC. Η τιμή της παραμέτρου εξαρτάται από τις υπόλοιπες παραμέτρους του συστήματος αλλά και από τους διαθέσιμους πόρους του FPGA (LUTs, FFs, BRAMs, DSPs).



Εικόνα 4: Η γενική αρχιτεκτονική δομή του NRM SoC. Το πλήθος των πυρήνων είναι παραμετρος του συστήματος.

Όπως φαίνεται στην Εικόνα 4, το NRM-SSA SoC είναι δυνατόν να περιέχει περισσότερους από έναν πυρήνες. Κάθε πυρήνας είναι ανεξάρτητος από τους υπόλοιπους και όλοι μαζί συνθέτουν ένα δίκτυο τοπολογίας αστέρα. Για να γίνει επιτυχής η επικοινωνία των πυρήνων με το Host-PC, κάθε πυρήνας συνδέεται στη θύρα ενός διανομέα (Switch) και χρησιμοποιούν ένα πρωτόκολλο επικοινωνίας που ακολουθεί πλήρως τις προδιαγραφές του AXI4 Stream Interface – AXIS4 του AMBA Bus [42] [43].

4.2 Αρχιτεκτονική Πυρήνα NRM-SSA

Ένας πυρήνας NRM-SSA είναι σε θέση να προσομοιώσει ένα οποιοδήποτε μοντέλο που έχει m αντιδράσεις (m στο διάστημα [2, 4096]), n μοριακά είδη (n στο διάστημα [2, 4096]) και μέσο όρο επηρεαζόμενων αντιδράσεων D_{aver} (D_{aver} στο διάστημα [2, 128]), αναλόγως τις γενικές παραμέτρους (generic parameters) που έχουν οριστεί κατά της σύνθεση του πυρήνα. Στην συνέχεια, αναλύεται ο τρόπος λειτουργίας του πυρήνα NRM-SSA καθώς και οι διαφορές του από τον αντίστοιχο πυρήνα FRM-SSA που είχαμε στη διάθεσή μας [33].

Συνοπτικά η λειτουργία των βασικών μονάδων του πυρήνα NRM-SSA είναι η εξής:

- **Input Interface System - IIS:** υπεύθυνη για τα εισερχόμενα δεδομένα του πυρήνα από το δίκτυο.
- **Output Interface System - OIS:** υπεύθυνη για τα εξερχόμενα δεδομένα του πυρήνα προς το δίκτυο.

- **Simulation Control Unit – SCU:** υπεύθυνη για τον έλεγχο και την εξέλιξη της προσομοίωσης που εκτελείται στο πυρήνα.
- **ST Initialization Memory System – STI MS:** υπεύθυνη για τη αποθήκευση των αρχικών τιμών των πληθυσμών των μοριακών ειδών με σκοπό την αρχικοποίηση του συστήματος στην αρχή κάθε επανάληψης της προσομοίωσης.
- **Species Update Unit – SPU:** υπεύθυνη για την ανανέωση των μοριακών πληθυσμών με βάση τη νικητήρια αντίδραση που έχει προκύψει στο τέλος κάθε κύκλου αντιδράσεων.
- **NRM Processing Unit – NPU:** είναι η καρδιά του NRM πυρήνα και εκτελεί τις βασικές λειτουργίες του NRM αλγορίθμου (μαθηματικές πράξεις και συγκρίσεις).

Ακολουθεί ο πίνακας όπου εμπεριέχονται όλα τα σήματα εισόδου και εξόδου της μονάδας NRM Core, καθώς και οι γενικές παράμετροι που καθορίζονται πριν τη σύνθεση του πυρήνα.

Πίνακας 1: Όλα τα σήματα εισόδου και εξόδου του NRM Core, καθώς και οι γενικές παράμετροί του.

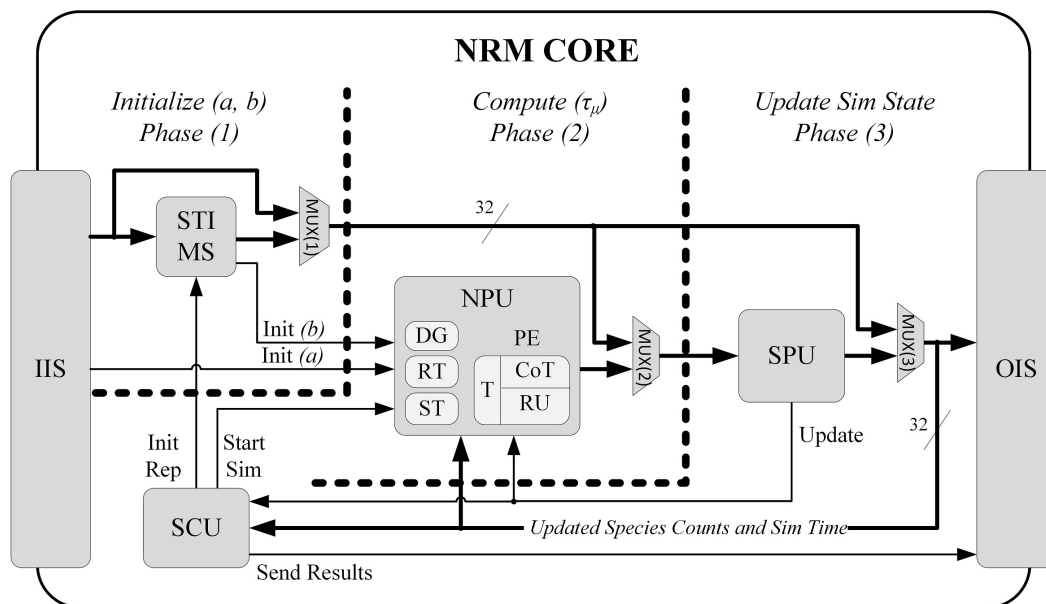
Όνομα Σήματος	Περιγραφή
Γενικές Παράμετροι (Generic Parameters)	
RT_SIZE	Το μέγιστο πλήθος αντιδράσεων (m) που μπορεί να υποστηρίξει ο πυρήνας.
ST_SIZE	Το μέγιστο πλήθος μοριακών ειδών (n) που μπορεί να υποστηρίξει ο πυρήνας.
DG_SIZE	Ο μέγιστος μέσος όρος των επηρεαζόμενων αντιδράσεων που μπορεί να υποστηρίξει ο πυρήνας (max Daver).
H	Το ύψος του δέντρου συγκριτών στην υπολογιστική μονάδα του πυρήνα.
Είσοδος	
s_axis_tid	Όλα τα απαραίτητα σήματα εισόδου με βάση το AXI4 Stream Interface – AXIS4 που εδραιώνουν την επικοινωνία του πυρήνα με τον διανομέα (Switch).
s_axis_tvalid	
s_axis_tready	
s_axis_tdata	
Έξοδος	
m_axis_status	Όλα τα απαραίτητα σήματα εξόδου με βάση το AXI4 Stream Interface – AXIS4 που εδραιώνουν την επικοινωνία του πυρήνα με τον διανομέα (Switch).
m_axis_tready	
m_axis_tvalid	
m_axis_tdata	

4.3 Λειτουργία Πυρήνα και Ροή Δεδομένων

Η Εικόνα 5 απεικονίζει την δομή ενός πυρήνα NRM-SSA. Στη συνέχεια παρουσιάζεται ο τρόπος λειτουργίας του και αναλύονται οι σχεδιαστικές επιλογές ως προς την διαφοροποίηση του NRM αλγορίθμου.

Φάση 1 - Αρχικοποίηση:

Η φάση αρχικοποίησης αποτελείται από δύο στάδια. Το πρώτο στάδιο (*Initialize 1.a*) εκτελείται μια φορά κατά την εκκίνηση της προσομοίωσης και περιλαμβάνει την αρχικοποίηση της δομής STI καθώς και των εσωτερικών δομών της μονάδας NPU (RT, DG, RNG).



Εικόνα 5: Η αρχιτεκτονική ενός πυρήνα NRM-SSA.

Η SCU είναι υπεύθυνη να προωθήσει τα δεδομένα από την είσοδο του πυρήνα στην είσοδο της εκάστοτε δομής ενεργοποιώντας το σήμα *Init(a)*. Το δεύτερο στάδιο (*Initialize 1.b*) εκτελείται στην αρχή κάθε επανάληψης της προσομοίωσης και περιλαμβάνει την αρχικοποίηση της δομής ST μέσα στη μονάδα NPU. Η SCU ενεργοποιεί το σήμα *Init_Rep*, ώστε να προωθηθούν τα δεδομένα από τη δομή STI στη δομή ST, ενεργοποιώντας το σήμα *Init(b)*.

Φάση 2 – Εύρεση της Επόμενης Νικητήριας Αντίδρασης:

Αφού ολοκληρωθεί η αρχικοποίηση, η SCU ενεργοποιεί το σήμα *Start_Sim* και ένας κύκλος αντιδράσεων ξεκινάει. Στη φάση αυτή, η μονάδα NPU εκτελεί όλες τις μαθηματικές και λογικές πράξεις που απαιτούνται ώστε να βρεθεί η επόμενη “νικητήρια αντίδραση”. Η SCU θα ενεργοποιήσει την συγκεκριμένη φάση εάν και μόνο αν ο τρέχον χρόνος προσομοίωσης είναι μικρότερος από τον μέγιστο επιθυμητό που έχει ορίσει ο χρήστης ($t_{sim} < T_{sim}$).

Φάση 3 – Ανανέωση πληθυσμών:

Όταν ολοκληρωθεί η Φάση 2, έχει βρεθεί η επόμενη νικητήρια αντίδραση, η αντίδραση δηλαδή με τον μικρότερο δυνητικά χρόνο ενεργοποίησης. Κατά τη Φάση 3, η μονάδα SPU εκτελεί τις πράξεις ανανέωσης των πληθυσμών των στοιχείων που συμμετέχουν στη νικητήρια αντίδραση, ενώ η SCU ανανεώνει τον τρέχοντα χρόνο προσομοίωσης και πραγματοποιεί τους εξής ελέγχους:

- Αν στη συγκεκριμένη χρονική στιγμή πρέπει να πραγματοποιηθεί δειγματοληψία τότε ενεργοποιείται ο μηχανισμός που προωθεί τα δεδομένα από την SPU στο Host-PC με τη βοήθεια της μονάδας OIS. Ο πυρήνας μεταβαίνει στη Φάση 2.
- Αν έχει παρέλθει ο επιθυμητός χρόνος προσομοίωσης, η προσομοίωση σταματάει και ο πυρήνας μεταβαίνει στη Φάση 1, ώστε να γίνει η αρχικοποίηση της δομής ST (*Initialize 1.b*) και να ξεκινήσει η επόμενη επανάληψη της προσομοίωσης (*Init_Rep*).
- Αν ο πυρήνας έχει εκτελέσει όλο το πλήθος των επαναλήψεων θα πρέπει να τερματίσει ολοκληρωτικά την προσομοίωση και να ενημερώσει το Host-PC.
- Διαφορετικά, συνεχίζεται η προσομοίωση μεταβαίνοντας στην Φάση 2.

Αν και η παραπάνω ροή δεδομένων είναι σταθερή, τόσο στον πυρήνα FRM-SSA [33]

όσο και στον πυρήνα NRM-SSA, παρόλα αυτά, τα δύο συστήματα έχουν και σημαντικές διαφορές που απαιτούν τον επανασχεδιασμό συγκεκριμένων μονάδων.

4.4 Διαφοροποίηση Πυρήνων FRM - NRM

Σε αυτό το επίπεδο σχεδίασης οι διαφορές που παρουσιάζουν ο πυρήνας NRM της παρούσας διπλωματικής εργασίας με τον αντίστοιχο πυρήνα FRM [33] είναι:

- Προστέθηκε μια κατάσταση αρχικοποίησης του γράφου εξαρτήσεων (Dependency Graph) μέσα στη μονάδα ελέγχου της προσομοίωσης του πυρήνα (SCU).
- Διαφέρει ο τρόπος διαχείρισης του χρόνου προσομοίωσης.

Το σύστημα για να αρχίσει να εκτελεί την προσομοίωση του μοντέλου, θα πρέπει πρώτα να αρχικοποιηθούν όλες οι παράμετροι, όλες οι δομές δεδομένων και όλα τα σήματα ελέγχου των μονάδων του πυρήνα NRM. Για τις ανάγκες του αλγορίθμου, μέσα στην επεξεργαστική μονάδα του πυρήνα (NRM Processing Unit, NPU), έχει σχεδιαστεί μια επιπλέον δομή δεδομένων που αποθηκεύει τον γράφο εξαρτήσεων στη μορφή πίνακα. Για τον λόγο αυτόν, μέσα στη μονάδα ελέγχου της προσομοίωσης του πυρήνα (SCU) έχει προστεθεί μια νέα κατάσταση που φροντίζει να αρχικοποιηθεί κατάλληλα ο πίνακας αυτός. Περισσότερες λεπτομέρειες για το γράφο εξαρτήσεων δίνονται σε επόμενη ενότητα που αναλύει το επίπεδο της υπολογιστικής μονάδας του NRM Core, την NPU (NRM Processing Unit).

Η δεύτερη αλλαγή επηρεάζει τη διαχείριση του χρόνου προσομοίωσης. Όπως έχει αναφερθεί και προηγουμένως, ο FRM αλγόριθμος υπολογίζει τη σχετική απόσταση του χρόνου της επόμενης αντίδρασης από την τρέχουσα χρονική στιγμή, ενώ ο NRM αλγόριθμος υπολογίζει την καθολική τιμή του χρόνου που θα εκτελεστεί η επόμενη αντίδραση. Επομένως, κατά τον FRM αλγόριθμο το αποτέλεσμα ενός κύκλου αντίδρασης είναι ο χρόνος που θα προστεθεί στον χρόνο προσομοίωσης, ενώ κατά τον NRM αλγόριθμο το αποτέλεσμα είναι ο ίδιος ο χρόνος προσομοίωσης. Και στις δύο περιπτώσεις η αντίδραση με τον μικρότερο χρόνο είναι η νικητήρια, αλλά στην περίπτωση του NRM ο χρόνος κάθε αντίδρασης αποθηκεύεται σε μια δομή δεδομένων και επαναχρησιμοποιείται στον επόμενο κύκλο αντίδρασης.

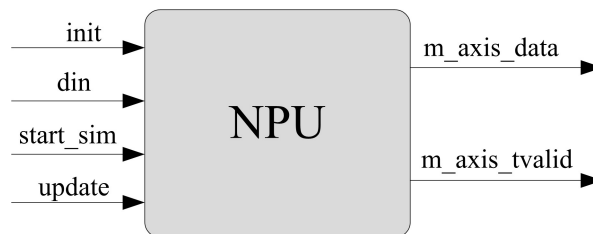
Συνεπώς, στο τέλος κάθε κύκλου αντιδράσεων η NPU προωθεί τα στοιχεία της νικητήριας αντίδρασης στην μονάδα SPU (Species Update Unit) ώστε να ενημερωθεί κατάλληλα ο πληθυσμός των μοριακών ειδών του μοντέλου σύμφωνα με την νικητήρια αντίδραση, και στη μονάδα SCU (Simulation Control Unit) η οποία είναι υπεύθυνη να αποθηκεύσει τον χρόνο της νικητήριας αντίδρασης, να ελέγξει αν πρέπει να γίνει δειγματοληψία σύμφωνα με την περίοδο δειγματοληψίας (T_{sam}), να τερματιστεί ή να συνεχιστεί η προσομοίωση αναλόγως αν η προσομοίωση έχει φτάσει στον μέγιστο επιθυμητό χρόνο προσομοίωσης (T_{sim}) που έχει ορίσει ο χρήστης ή όχι.

5. ΑΡΧΙΤΕΚΤΟΝΙΚΗ NRM ΕΠΕΞΕΡΓΑΣΤΙΚΗΣ ΜΟΝΑΔΑΣ

Μια από τις ιδιαιτερότητες του NRM Core ως προς τη σχεδίαση για υλοποίηση σε FPGA είναι η μη παραλληλοποιήσιμη δομή του αλγόριθμου και κατ' επέκταση του σχεδίου του NRM πυρήνα. Ο FRM αλγόριθμος απαιτεί τον υπολογισμό των χρόνων t_j όλων των αντιδράσεων, μια διαδικασία η οποία μπορεί να ανατεθεί σε περισσότερες από μια παράλληλες υπολογιστικές μονάδες (Processing Element – PE). Ο NRM αλγόριθμος, όμως, χειρίζεται πολύ μικρότερο αριθμό αντιδράσεων σε κάθε κύκλο αντίδρασης και συνεπώς δεν απαιτούνται παράλληλοι υπολογισμοί.

5.1 NPM Processing Unit - NPU

Η μονάδα NRM Processing Unit (NPU) είναι η υπολογιστική μονάδα του πυρήνα NRM και η αρχιτεκτονική της είναι διαφορετική από την αντίστοιχη FPU (FRM Processing Unit) του πυρήνα FRM [33]. Οι διαφορές επικεντρώνονται κυρίως στην εισαγωγή της δομής του γράφου εξαρτήσεων, στη δομή της υπολογιστικής μηχανής και στα διάφορα σήματα ελέγχου που προστίθενται για τον έλεγχο των εσωτερικών μονάδων της μονάδας NPU.

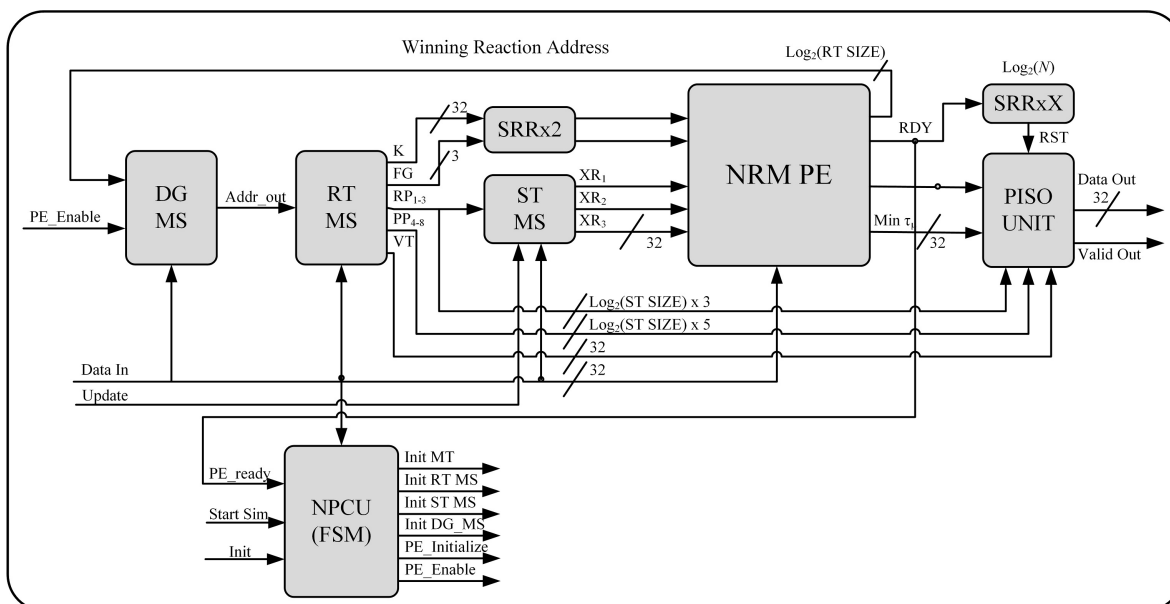


Εικόνα 6: Το RTL σχηματικό της μονάδας NPU.

Ο πίνακας που ακολουθεί περιέχει όλα τα σήματα εισόδου και εξόδου της μονάδας NPU.

Πίνακας 2: Τα σήματα εισόδου και εξόδου της μονάδας NPU

Όνομα Σήματος	Περιγραφή
Είσοδος	
init	Όταν είναι ενεργοποιημένο πραγματοποιείται διαδικασία αρχικοποίησης και τα δεδομένα από την είσοδο πρέπει να προωθηθούν στην μονάδα που ορίζει η μονάδας ελέγχου.
din	Το κανάλι εισόδου των δεδομένων από το υψηλότερο επίπεδο προς την NPU και αντίστοιχα προς τις εσωτερικές μονάδες της.
start_sim	Όταν είναι ενεργοποιημένο η προσομοίωση εκτελείται κανονικά.
update	Όταν ενεργοποιηθεί τότε στο κανάλι εισόδου έχουν έρθει οι ανανεωμένες τιμές των μοριακών ειδών. Οι νέες τιμές θα πρέπει να αποθηκευτούν στην εσωτερική δομή της NPU ώστε να είναι διαθέσιμες για τον επόμενο κύκλο αντίδρασης.
Έξοδος	
m_axis_tdata	Το κανάλι εξόδου δεδομένων από την NPU προς το υψηλότερο επίπεδο σχεδίασης.
m_axis_tvalid	Όταν ενεργοποιηθεί ενημερώνει το υψηλότερο επίπεδο σχεδίασης ότι τα δεδομένα στο κανάλι εξόδου της NPU έχουν χρήσιμες πληροφορίες.



Εικόνα 7: Η αρχιτεκτονική της μονάδα NRM Processing Unit (NPU).

Όπως και στο επίπεδο του πυρήνα, έτσι και στο επίπεδο της επεξεργαστικής μονάδας, επαναχρησιμοποιούνται οι περισσότερες βασικές μονάδες. Έτσι λοιπόν, στο επίπεδο σχεδίασης της NPU (NRM Processing Unit) εμπεριέχονται όλες οι μονάδες που εμπεριέχονται στην μονάδα FPU (FRM Processing Unit) [33], ελαφρώς τροποποιημένες. Τέλος, στην NPU έχει σχεδιαστεί και προστεθεί μια επιπλέον μονάδα διαχείρισης του γράφου εξαρτήσεων (Dependency Graph Memory System - DG MS).

Στην Εικόνα 7 παρουσιάζεται η αρχιτεκτονική της μονάδας NPU, ενώ η λειτουργία της και η ροή δεδομένων παρουσιάζεται εκτενώς στην Ενότητα 5.5.

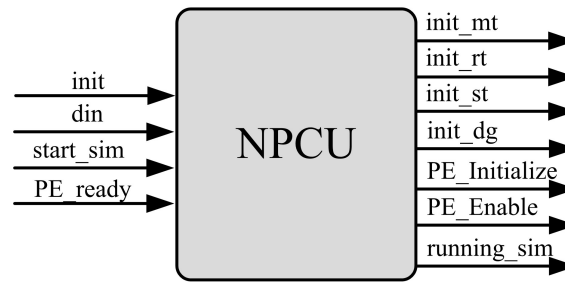
Στο επίπεδο της NPU, οι μονάδες που παραμένουν ίδιες και έχουν αναλυθεί εκτενώς είναι οι εξής:

- Reaction Table Memory System (RT_MS): Η μονάδα που διατηρεί σε μια δομή δεδομένων όλες τις πληροφορίες για τις αντιδράσεις του βιομοντέλου.
- ST Memory System (ST_MS): Η μονάδα που διατηρεί σε μια δομή δεδομένων τους πληθυσμούς των μοριακών ειδών του βιομοντέλου καθ' όλη τη διάρκεια μιας επανάληψης της προσομοίωσης.
- Parallel in Serial out (PISO): Μια βασική μονάδα που δέχεται παράλληλα δεδομένα στις εισόδους της τα οποία στη συνέχεια οδηγεί σειριακά στην έξοδο της, και συγκεκριμένα στην έξοδο της NPU.

Ενώ, οι μονάδες που προστίθενται ή τροποποιούνται είναι οι εξής:

- NRM Processing Control Unit (NPCU): Τροποποιείται ελαφρώς. Προστίθεται μια επιπλέον κατάσταση στην FSM, ώστε να εκτελείται κατάλληλα η αρχικοποίηση του γράφου εξαρτήσεων.
- NRM Processing Element (NPE): Πλήρης επανασχεδιασμός. Προστίθεται ένα δέντρο συγκριτών και τροποποιείται η ροή δεδομένων μέσα στην υπολογιστική μονάδα του πυρήνα.
- Dependency Graph Memory System (DG_MS): Η μονάδα που χειρίζεται τη δομή δεδομένων του γράφου εξαρτήσεων. Υπεύθυνη να εξαγάγει τις διευθύνσεις της νικητριάς και των επηρεαζόμενων αντιδράσεων.

5.2 NRM Processing Control Unit – NPCU



Εικόνα 8: Το RTL σχηματικό της μονάδας NPCU.

Η μονάδα ελέγχου της υπολογιστικής οντότητας NPU, η NRM Processing Control Unit (NPCU), στο μεγαλύτερο μέρος της παραμένει ίδια με την FPCU (FRM Processing Control Unit) [33] εκτός από τρία σημεία. Αρχικά, στις παραμέτρους του συστήματος προστίθενται μια ακόμα λέξη που δηλώνει το πλήθος των λέξεων που πρέπει να περιμένει το σύστημα ώστε να ολοκληρωθεί επιτυχώς η αρχικοποίηση του γράφου εξαρτήσεων. Εν συνεχεία, στην FSMD της μονάδας προστίθεται μια επιπλέον κατάσταση που σχετίζεται με την αρχικοποίηση του γράφου εξαρτήσεων. Στην κατάσταση αυτή η είσοδος του πυρήνα προωθείται στην είσοδο της μνήμης δεδομένων του γράφου και ενεργοποιείται το σήμα ελέγχου *init_dg* που σηματοδοτεί την ενεργοποίηση εγγραφής στη μνήμη. Το σύστημα θα παραμείνει στην κατάσταση αυτή μέχρις ότου διαβαστεί όλο το πλήθος των απαιτούμενων λέξεων που έχει οριστεί από τις παραμέτρους του συστήματος. Τέλος, ο αλγόριθμος NRM, ως πρώτο βήμα, απαιτεί την αρχικοποίηση όλων των δομών δεδομένων που σχετίζονται με τις τάσεις α_j και τους χρόνους τ_j όλων των αντιδράσεων. Για να πραγματοποιηθεί η αρχικοποίηση, η υπολογιστική μονάδα (NRM Processing Element, NPE) του πυρήνα καλείται να υπολογίσει για μια και μόνο φορά τις τάσεις α_j και τους χρόνους τ_j όλων των αντιδράσεων και να αποθηκεύσει τις τιμές αυτές στις κατάλληλες δομές δεδομένων. Μόλις ολοκληρωθεί η διαδικασία αρχικοποίησης, έχει προκύψει η πρώτη νικητήρια αντίδραση και το σύστημα είναι πλέον σε θέση να λειτουργήσει. Στο εξής, το NPE υπολογίζει τους χρόνους τ_j μόνο της νικητήριας αντίδρασης και των επηρεαζόμενων από αυτή αντιδράσεων. Για το λόγο αυτό, λοιπόν, η NPCU έχει σχεδιαστεί να παράγει δύο σήματα ελέγχου, το *PE_initialize*, όπου τα δεδομένα όλων των αντιδράσεων προωθούνται στο NPE προκειμένου να αρχικοποιηθούν οι δομές δεδομένων, και το *PE_enable*, όπου το NPE δέχεται μόνο τα δεδομένα της νικητήριας αντίδρασης και των επηρεαζόμενων αντιδράσεων. Και στις δύο περιπτώσεις, στο τέλος, προκύπτει μια νικητήρια αντίδραση, η αντίδραση δηλαδή με τον μικρότερο χρόνο ενεργοποίησης.

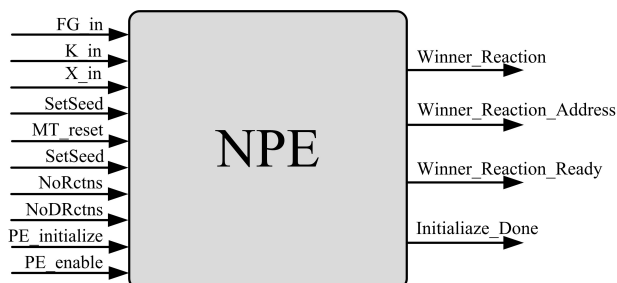
Ο πίνακας που ακολουθεί εμπεριέχει όλα τα σήματα εισόδου και εξόδου της μονάδας NPCU.

Πίνακας 3: Τα σήματα εισόδου και εξόδου της μονάδας NPCU.

Όνομα Σήματος	Περιγραφή
Είσοδος	
init	Όταν είναι ενεργοποιημένο πραγματοποιείται διαδικασία αρχικοποίησης και τα δεδομένα από την είσοδο πρέπει να προωθηθούν στην μονάδα που ορίζει η μονάδας ελέγχου.
din	Το κανάλι εισόδου των δεδομένων από το υψηλότερο επίπεδο προς την NPU και αντίστοιχα προς τις εσωτερικές μονάδες της.
start_sim	Όταν είναι ενεργοποιημένο η προσομοίωση εκτελείται κανονικά.
PE_ready	Ενημερώνει την NPCU ότι ένας κύκλος αντίδρασης έχει ολοκληρωθεί και μια νέα νικητήρια αντίδραση έχει προκύψει. Επίσης, το σήμα αυτό ενεργοποιεί τον μηχανισμό όπου συλλέγονται τα δεδομένα της νικητήριας αντίδρασης και μέσω της μονάδας PISO προωθούνται στην έξοδο της NPU.
Έξοδος	
init_mt	Σηματοδοτεί την αρχικοποίηση της δομής δεδομένων που σχετίζεται με τη γεννήτρια τυχαίων αριθμών, Mersenne Twister. Η δομή αυτή θα αρχικοποιηθεί μια και μόνο φορά, πριν την εκτέλεση της προσομοίωσης του μοντέλου.
init_rt	Σηματοδοτεί την αρχικοποίηση της δομής δεδομένων που σχετίζεται με τις πληροφορίες των αντιδράσεων, RT_MS. Η δομή αυτή θα αρχικοποιηθεί μια και μόνο φορά, πριν την εκτέλεση της προσομοίωσης του μοντέλου.
init_st	Σηματοδοτεί την αρχικοποίηση της δομής δεδομένων που σχετίζεται με τους πληθυσμούς των στοιχείων του μοντέλου. Η δομή αυτή αρχικοποιείται στην αρχή κάθε επανάληψης της προσομοίωσης.
init_dg	Σηματοδοτεί την αρχικοποίηση της δομής δεδομένων που σχετίζεται με τον γράφο εξαρτήσεων, DG_MS. Η δομή αυτή θα αρχικοποιηθεί μια και μόνο φορά, πριν την εκτέλεση της προσομοίωσης του μοντέλου.
PE_Initialize	Ενημερώνει τη μονάδα NPE ότι στα κανάλια εισόδου έχουν προωθηθεί τα δεδομένα (K, X, FG, κ.α.) όλων των αντιδράσεων. Ακολουθεί η αρχικοποίηση των εσωτερικών δομών δεδομένων του NPE.
PE_Enable	Ενημερώνει τη μονάδα NPE ότι στα κανάλια εισόδου έχουν προωθηθεί τα δεδομένα (K, X, FG, κ.α.) της νικητήριας αντίδρασης και των επηρεαζόμενων αντιδράσεων. Ακολουθούν όλες οι απαραίτητες πράξεις και συγκρίσεις ώστε να προκύψει η επόμενη νικητήρια αντίδραση.
running_sim	Ενεργοποιείται στην αρχή κάθε κύκλου αντιδράσεων ώστε να διαβαστούν όλα τα δεδομένα (K, X, FG, κ.α.) από τις κατάλληλες δομές δεδομένων και να προωθηθούν στα αντίστοιχα κανάλια εισόδου της μονάδας NPE.

5.3 Processing Element - NPE

Η μονάδα NPE δέχεται στην είσοδό της όλες τις απαραίτητες πληροφορίες που απαιτούνται προκειμένου να υπολογιστεί η χρονική στιγμή που θα εκτελεστεί μια αντίδραση. Για να πραγματοποιηθούν οι κατάλληλες πράξεις υπολογισμού οι πληροφορίες της νικητήριας αντίδρασης και των επηρεαζόμενων αντιδράσεων εξάγονται από τη μνήμη του συστήματος και προωθούνται στην είσοδο της μονάδας NPE. Η αρχιτεκτονική της μονάδας NPE θα αναλυθεί εκτενώς στο Κεφάλαιο 6.



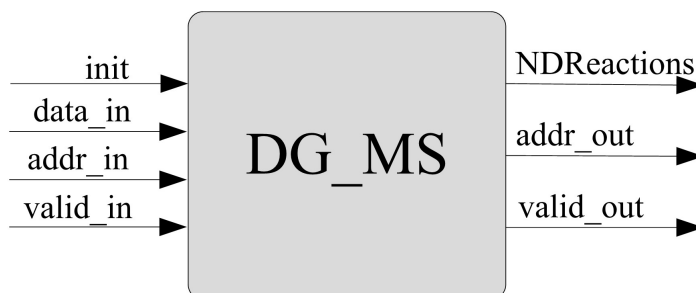
Εικόνα 9: Τα σήματα εισόδου και εξόδου της μονάδας NPE.

Πίνακας 4: Τα σήματα εισόδου και εξόδου της μονάδας NPE.

Όνομα Σήματος	Περιγραφή
Είσοδος	
FG_in	Παράμετρος που επηρεάζει τον υπολογισμό της τάσης α_j της εκάστοτε αντίδρασης.
K_in	Παράμετρος που επηρεάζει τον υπολογισμό της τάσης α_j της εκάστοτε αντίδρασης.
X_in	Ο πληθικός αριθμός των αντιδρώντων της εκάστοτε αντίδρασης.
MT_reset	Σήμα ελέγχου επαναφοράς της μονάδας παραγωγής τυχαίων αριθμών.
SetSeed	Σήμα ελέγχου εγγραφής στη δομή δεδομένων της μονάδας παραγωγής τυχαίων αριθμών.
Seed	Το κανάλι εισόδου δεδομένων της μονάδας παραγωγής τυχαίων αριθμών.
NoRctns	Το πλήθος των αντιδράσεων του μοντέλου που προσομοιώνεται.
NoDRctns	Το πλήθος των επηρεαζόμενων αντιδράσεων της νικητήριας αντίδρασης στον τρέχοντα κύκλο αντιδράσεων.
PE_initialize	Όταν ενεργοποιηθεί πραγματοποιείται η αρχικοποίηση των εσωτερικών δομών δεδομένων τη μονάδας NPE.
PE_enable	Το σήμα ενεργοποίησης της μονάδας NPE.
Εξοδος	
Winner_Reaction	Ο χρόνος της νικητήριας αντίδρασης.
Winner_Reaction_Address	Η διεύθυνση της νικητήριας αντίδρασης.
Winner_Reaction_Ready	Σηματοδοτεί την ολοκλήρωση της διαδικασίας εύρεσης νικητήριας αντίδρασης.
Initialize_Done	Σηματοδοτεί την ολοκλήρωση της αρχικοποίησης των δομών δεδομένων μέσα στη μονάδα NPE.

5.4 Dependency Graph (DG_MS)

Ο γράφος εξαρτήσεων, Dependency Graph, είναι αποθηκευμένος σε μια δομή δεδομένων μέσα στη μονάδα Dependency Graph Memory System (DG_MS). Η μονάδα DG_MS, εκτός από τον γράφο, περιλαμβάνει μια απλή FSM που είναι υπεύθυνη να δέχεται τη διεύθυνση της νικητήριας αντίδρασης και να κατευθύνει στην έξοδο τις διευθύνσεις των επηρεαζόμενων αντιδράσεων, τη μία μετά την άλλη. Οι διευθύνσεις αυτές προωθούνται στην μονάδα RT_MS όπου εξάγονται οι πληροφορίες της αντίδρασης και οι διευθύνσεις των στοιχείων που συμμετέχουν στην εκάστοτε αντίδραση. Αφού διαβαστούν και οι πληθυσμοί των μοριακών ειδών, με τη βοήθεια της μονάδας ST_MS, όλες οι πληροφορίες καταλήγουν στην είσοδο της μονάδας NPE.



Εικόνα 10: Το RTL σχηματικό τη μονάδας DG_MS.

Στη συνέχεια ακολουθεί ο πίνακας με τα σήματα εισόδου και εξόδου της μονάδας DG_MS.

Πίνακας 5: Τα σήματα εισόδου και εξόδου της μονάδας DG_MS.

Όνομα Σήματος	Περιγραφή
Είσοδος	
init	Όταν ενεργοποιηθεί πραγματοποιείται η αρχικοποίηση της μνήμης δεδομένων του γράφου. Τα δεδομένα από την είσοδο θα πρέπει να αποθηκευτούν στη μνήμη της μονάδας.
data_in	Το κανάλι εισόδου δεδομένων. Αφορούν μόνο τα δεδομένα αρχικοποίησης της μνήμης του γράφου.
addr_in	Η διεύθυνση της νικητήριας αντίδρασης.
valid_in	Όταν είναι ενεργοποιημένο η είσοδο addr_in έχει τη διεύθυνση της νικητήριας αντίδρασης και θα πρέπει η μονάδα DG_MS να εξάγει τις διευθύνσεις την νικητήριας αντίδρασης.
Έξοδος	
NDRactions	Στο κανάλι αυτό προωθούνται αρχικά η διεύθυνση της νικητήριας αντίδρασης και στη συνέχεια, η μία μετά την άλλη, οι διευθύνσεις των επηρεαζόμενων αντιδράσεων.
addr_out	Όταν το σήμα valid_out είναι ενεργοποιημένο τότε στην έξοδο της μονάδας έχουν εμφανιστεί διευθύνσεις που πρέπει να προωθηθούν τόσο στη μνήμη RT_MS όσο και στην μονάδα NPE.
valid_out	Μαζί με τις διευθύνσεις των αντιδράσεων η μονάδα DG_MS φροντίζει να ενημερώσει την μονάδα NPE για το πλήθος των επηρεαζόμενων αντιδράσεων της εκάστοτε νικητήριας αντίδρασης.

Η μονάδα DG_MS αποτελείται από μια δομή δεδομένων που διατηρεί τα δεδομένα του γράφου σε μορφή πίνακα, μια μονάδα PISO (Parallel-In to Serial-Out) που συμμετέχει στην αρχικοποίηση του πίνακα και μια FSMD που με βάση την είσοδο είναι υπεύθυνη να προωθήσει στην έξοδο τις διευθύνσεις των κατάλληλων αντιδράσεων.

Η δομή δεδομένων του γράφου εξαρτήσεων έχει τη μορφή ενός πίνακα και ακολουθείται μια συγκεκριμένη στρατηγική ώστε να μεγιστοποιηθεί η αποτελεσματικότητα και να εκμεταλλευτούμε στο μέγιστο το μέγεθος της μνήμης που επιλέγουμε να έχουμε στη διάθεσή μας. Ο πίνακας του γράφου έχει βάθος (αριθμός γραμμών) ίσο με το πλήθος των αντιδράσεων (m) του βιομοντέλου και επομένως η διεύθυνση της εκάστοτε αντίδρασης αντιστοιχεί στην ίδια διεύθυνση του πίνακα εξαρτήσεων. Δεδομένου ότι δεν έχουν όλες οι αντιδράσεις ίσο πλήθος επηρεαζόμενων αντιδράσεων, πρέπει αφενός να αποθηκευτεί το πλήθος των επηρεαζόμενων αντιδράσεων (D) της κάθε αντίδρασης και αφετέρου να αποθηκευτούν όλες οι διευθύνσεις (*pointers*) των επηρεαζόμενων αντιδράσεων της εκάστοτε αντίδρασης. Οι διευθύνσεις αποθηκεύονται η μία μετά την άλλη χρησιμοποιώντας συνεχόμενες θέσεις μέσα στον πίνακα, ώστε να ελαχιστοποιηθεί η απαιτούμενη μνήμη. Επειδή, όμως, οι διευθύνσεις των επηρεαζόμενων αντιδράσεων δεν ξεκινούν από μια συγκεκριμένη θέση, χρησιμοποιούνται δύο στήλες του πίνακα ως ευρετήριο. Στο ευρετήριο αποθηκεύεται η γραμμή (*row*) και η στήλη (*column*) από όπου θα ξεκινήσει η FSMD να εξάγει τις απαιτούμενες διευθύνσεις.

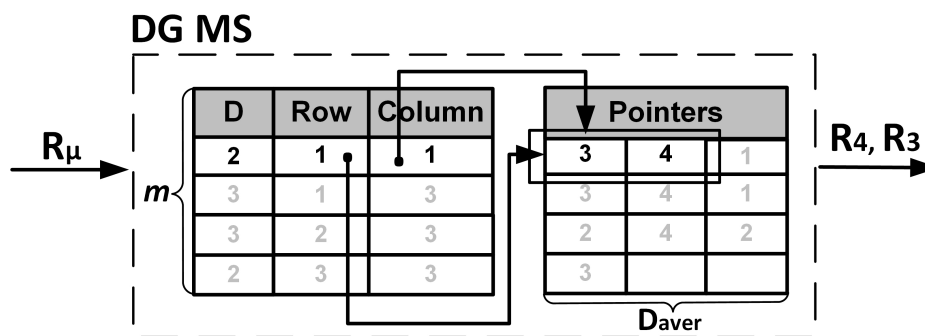
Έτσι, λοιπόν, ο πίνακας του γράφου εξαρτήσεων αποτελείται από δύο τμήματα. Το πρώτο τμήμα αποτελείται από τρεις στήλες. Στην πρώτη στήλη αποθηκεύεται το πλήθος των επηρεαζόμενων αντιδράσεων της εκάστοτε αντίδρασης. Στη δεύτερη και στη τρίτη στήλη αποθηκεύεται το ευρετήριο για το τμήμα του γράφου. Το δεύτερο τμήμα του πίνακα χρησιμοποιείται εξ ολοκλήρου για την αποθήκευση διευθύνσεων επηρεαζόμενων αντιδράσεων. Επομένως, το πλάτος του πίνακα ορίζεται ως $(D_{aver} + 3) * \log(m)$, όπου 3 στήλες χρησιμοποιούνται για το ευρετήριο και D_{aver} στήλες για τις διευθύνσεις των επηρεαζόμενων αντιδράσεων. Η παράμετρος D_{aver} ορίζεται ως γενική παράμετρος (*generic*) του πυρήνα και θα πρέπει να καλύπτει το μέσο όρο των επηρεαζόμενων αντιδράσεων του μοντέλου. Κατά αυτό τον τρόπο ορίζεται η βέλτιστη τιμή ώστε να ελαχιστοποιείται η απαιτούμενη μνήμη πριν τη σύνθεση του πυρήνα.

Στην Εικόνα 11 παρουσιάζεται ένα ενδεικτικό παράδειγμα πίνακα ενός γράφου εξαρτήσεων για ένα μοντέλο με 4 αντιδράσεις, όπου οι εξαρτήσεις ανά αντίδραση είναι ως εξής:

- Η αντίδραση R_1 επηρεάζει τις αντιδράσεις R_3 και R_4 .
- Η αντίδραση R_2 επηρεάζει τις αντιδράσεις R_1 , R_3 , και R_4 .
- Η αντίδραση R_3 επηρεάζει τις αντιδράσεις R_1 , R_2 , και R_4 .
- Η αντίδραση R_4 επηρεάζει τις αντιδράσεις R_2 και R_3 .

Στο παραπάνω παράδειγμα, ο μέσος όρος των επηρεαζόμενων αντιδράσεων είναι 2,5 αντιδράσεις. Για τον λόγο αυτό ο πίνακας έχει πλάτος 3 θέσεων ώστε να καλύψει τον μέσο όρο. Επίσης, είναι φανερό ότι για τις περισσότερες αντιδράσεις οι διευθύνσεις των επηρεαζόμενων αντιδράσεων επεκτείνονται σε περισσότερες από μια γραμμές του πίνακα. Προκειμένου η μονάδα DG_MS να εξάγει τις διευθύνσεις, τη μία μετά την άλλη, φροντίζει πάντα να έχουν διαβαστεί δύο συνεχόμενες γραμμές του πίνακα. Αν για παράδειγμα, μια αντίδραση χρειάζεται τρεις γραμμές ώστε να αποθηκευτούν οι επηρεαζόμενες αντιδράσεις της, η μονάδα DG_MS διαβάζει τις δύο πρώτες γραμμές και μόλις καλύψει όλες τις θέσεις της πρώτης γραμμής τότε διαβάζει την τρίτη γραμμή. Με αυτόν τον τρόπο η FSMD έχει στη διάθεσή της πάντα δύο συνεχόμενες γραμμές του πίνακα και μπορεί να εξάγει

συνεχόμενες διευθύνσεις στην έξοδο της μονάδας DG_MS.



Εικόνα 11: Ενδεικτικό παράδειγμα λειτουργίας του γράφου εξαρτήσεων.

Έτσι λοιπόν, εφόσον έχει πραγματοποιηθεί η αρχικοποίηση του συστήματος, η ροή των δεδομένων μέσα στη μονάδα που διαχειρίζεται τον γράφο εξαρτήσεων ακολουθεί τα εξής βήματα:

1. Στην είσοδο `addr_in` έχει προωθηθεί η διεύθυνση της επιθυμητής νικητήριας αντίδρασης.
2. Στην αρχή ενός κύκλου αντιδράσεων ενεργοποιείται το σήμα `valid_in` που σηματοδοτεί την έναρξη της εξαγωγής διευθύνσεων των επηρεαζόμενων αντιδράσεων της νικητήριας αντίδρασης από τον γράφο εξαρτήσεων.
3. Η FSMD της μονάδας DG_MS διαβάζει την γραμμή του πίνακα που αντιστοιχεί στη νικητήρια αντίδραση. Προωθεί στην έξοδο της μονάδας το πλήθος των επηρεαζόμενων αντιδράσεων και με βάση τη τιμή "row" του ευρετηρίου διαβάζει τις γραμμές του πίνακα με διεύθυνση "row" και "row+1", ώστε να έχει στη διάθεσή της δύο συνεχόμενες γραμμές του πίνακα.
4. Αναλόγως του πλήθους των επηρεαζόμενων αντιδράσεων η FSMD αρχίζει να προωθεί (`valid_out`) στην έξοδο της μονάδας DG_MS τις διευθύνσεις τους (`addr_out`), τη μία μετά την άλλη. Είναι απαραίτητο να εμφανιστεί στην έξοδο πρώτα η διεύθυνση της νικητήριας αντίδρασης και στη συνέχεια οι διευθύνσεις των επηρεαζόμενων αντιδράσεων.
5. Η μονάδα DG_MS θα σταματήσει να προωθεί διευθύνσεις επηρεαζόμενων αντιδράσεων στην έξοδο της μονάδας μόλις διαβαστούν από τον πίνακα τόσες διευθύνσεις όσο και το πλήθος των επηρεαζόμενων αντιδράσεων της αντίστοιχης νικητήριας αντίδρασης.
6. Η μονάδα DG_MS απενεργοποιείται έως ότου ξεκινήσει ένα νέος κύκλος αντιδράσεων.

Η μονάδα DG_MS είναι μια από τις σημαντικότερες μονάδες ενός πυρήνα NRM καθώς η έννοια της εξάρτησης των αντιδράσεων είναι εκείνη που κάνει τον αλγόριθμο NRM των Gibson και Bruck να διαφέρει αισθητά από τον αλγόριθμο FRM του Gillespie. Είναι εξίσου σημαντική, όμως, και ως προς τη σχεδίαση του πυρήνα καθώς καταλαμβάνει ένα μεγάλο ποσοστό των μνημών που έχουμε στη διάθεσή μας. Είναι σημαντικό, λοιπόν, να δίδεται ιδιαίτερη σημασία στο μέσο όρο των επηρεαζόμενων αντιδράσεων και στον ορισμό της γενικής παραμέτρου DG_SIZE πριν τη σύνθεση του πυρήνα.

5.5 Ροή Δεδομένων - NPU Datapath

Έχοντας αναλύσει τη λειτουργικότητα όλων των μονάδων, ολοκληρώνεται η περιγραφή της αρχιτεκτονικής της μονάδας NRM Processing Unit (NPU). Ακολουθεί η περιγραφή της ροής δεδομένων κατά την αρχικοποίηση και κατά την προσομοίωση ενός μοντέλου.

Πριν από την εκκίνηση της προσομοίωσης ενός βιομοντέλου, το σύστημα θα πρέπει να αρχικοποιηθεί. Η διαδικασία αυτή περιλαμβάνει την αρχικοποίηση των παραμέτρων του συστήματος, όλων των δομών δεδομένων και όλων των σημάτων ελέγχου που σχετίζονται με την προσομοίωση. Η μονάδα ελέγχου NPCU είναι υπεύθυνη ώστε να προωθήσει κατάλληλα την πληροφορία από την είσοδο της NPU στην είσοδο των επιμέρους μονάδων, ενεργοποιώντας το σήμα *init* που αντιστοιχεί στην εκάστοτε μονάδα. Όταν ολοκληρωθεί η αρχικοποίηση των βασικών δομών δεδομένων σε αυτό το επίπεδο σχεδίασης, η NPCU ενεργοποιεί την αρχικοποίηση των δομών μέσα στην υπολογιστική μονάδα NPE. Οι δομές μέσα στη μονάδα NPE είναι ο πίνακας **T**, που διατηρεί την χρονική στιγμή ενεργοποίησης κάθε αντίδρασης (καθολικός χρόνος), και ο πίνακας **A**, που διατηρεί τις τάσεις a_j για κάθε μια αντίδραση. Οι δύο αυτές δομές, όπως θα δούμε στην επόμενη ενότητα, είναι χρήσιμες για την ανανέωση του χρόνου t_j μιας επηρεαζόμενης αντίδρασης.

Αρχικοποίηση:

Αρχικά, το σύστημα είναι σε κατάσταση αναμονής, περιμένοντας να ενεργοποιηθεί το σήμα *init* στην είσοδο της μονάδας NPU ώστε να ξεκινήσει η αρχικοποίηση. Όταν ενεργοποιηθεί, η μονάδα NPCU μεταβαίνει στην κατάσταση *set_parameters* όπου ξεκινάει η αποθήκευση των παραμέτρων της προσομοίωσης. Οι απαραίτητοι παράμετροι προωθούνται από την είσοδο της NPU στην είσοδο της NPCU και είναι οι εξής:

- *Rt_words*: το πλήθος των λέξεων (32 bit) που πρέπει η NPCU να διαβάσει και να προωθήσει στη δομή δεδομένων που διατηρεί τις πληροφορίες των αντιδράσεων του βιομοντέλου.
- *Number of Species*: το πλήθος των στοιχείων του βιομοντέλου.
- *Number of Repetitions*: το πλήθος των επαναλήψεων της προσομοίωσης του βιομοντέλου.
- *Dg_words*: το πλήθος των λέξεων (32 bit) που πρέπει η NPCU να διαβάσει και να προωθήσει στη δομή δεδομένων που διατηρεί τις πληροφορίες του γράφου εξαρτήσεων των αντιδράσεων του μοντέλου.

Όταν όλες οι παράμετροι αποθηκευτούν τότε η NPCU μεταβαίνει στην κατάσταση *init_mt_state* και ξεκινάει η αρχικοποίηση της μονάδας παραγωγής τυχαίων αριθμών (Mersenne Twister Random Number Generator – MT RNG), η οποία χρησιμοποιεί τον αλγόριθμο Mersenne Twister [44]. Κατά το στάδιο αυτό η NPCU προωθεί την είσοδο της NPU στην είσοδο *Seed* της μονάδας NPE και παράγει το σήμα *init_mt* που συνδέεται με το σήμα *SetSeed* της μονάδας NPE. Κατ' αυτόν τον τρόπο ενεργοποιείται η εγγραφή δεδομένων στους πίνακες της μονάδας MT. Για την μονάδα MT απαιτούνται 624 λέξεις ώστε να γεμίσουν οι πίνακες και να ολοκληρωθεί η διαδικασία. Στη συνέχεια, η NPCU μεταβαίνει στην κατάσταση *init_rt_state* και ξεκινάει η αρχικοποίηση της δομής που εμπεριέχει όλες τις πληροφορίες των αντιδράσεων. Η παράμετρος *Rt_words* καθορίζει το πλήθος των λέξεων που περιμένει η NPCU να διαβάσει και να προωθήσει από την είσοδο της NPU στην είσοδο της μονάδας *RT_MS*, ενεργοποιώντας αντίστοιχα τον σήμα *init_rt* που στοχεύει στην εκκίνηση της εγγραφής των δεδομένων στον πίνακα RT. Αφού ολοκληρωθεί η αρχικοποίηση του πίνακα RT, το σύστημα προχωράει στην αρχικοποίηση του πίνακα STI στο επίπεδο του πυρήνα, ένας πίνακας που στόχο έχει να βοηθάει στην αρχικοποίηση του πίνακα ST της NPU στην αρχή κάθε επανάληψης. Η επόμενη κατάσταση στην

οποία θα μεταβεί η NPCU είναι η *init_dg_state*, όπου με βάση το πλήθος των λέξεων που καθορίζει η παράμετρος *Dg_words*, η NPCU θα στέλνει τα δεδομένα από την είσοδο της NPU στην είσοδο της μονάδας DG_MS και θα ενεργοποιήσει το σήμα *init_dg* που στοχεύει στην έναρξη της εγγραφής των δεδομένων στον πίνακα DG. Μετά την ολοκλήρωση της αρχικοποίησης της δομής του γράφου εξαρτήσεων, η NPCU μεταβαίνει σε μια κατάσταση αναμονής, μέχρις ότου το σύστημα να είναι έτοιμο να εκτελέσει την πρώτη επανάληψη της προσομοίωσης.

Αρχικοποίηση της μονάδας NPE:

Αφού ολοκληρωθεί η αρχικοποίηση των βασικών δομών, το σύστημα είναι έτοιμο να εκτελέσει την πρώτη επανάληψη της προσομοίωσης. Στην αρχή κάθε επανάληψης θα πρέπει να πραγματοποιηθούν δύο διαδικασίες αρχικοποίησης. Η πρώτη διαδικασία αφορά τον πίνακα ST όπου αποθηκεύονται οι πληθυσμοί των μοριακών ειδών του μοντέλου καθ' όλη τη διάρκεια της προσομοίωσης. Η μονάδα SCU είναι υπεύθυνη να διαβάσει τις αρχικές τιμές από την μονάδα STI_MS και να τις προωθήσει στην μονάδα NPU ώστε να αρχικοποιηθεί ο πίνακας ST. Η διαδικασία αυτή πραγματοποιείται στην αρχή κάθε επανάληψης της προσομοίωσης.

Η δεύτερη διαδικασία αφορά την αρχικοποίηση των εσωτερικών δομών του NPE. Σύμφωνα με τον αλγόριθμο των Gibson και Bruke, για την ανανέωση του χρόνου t_j μια επηρεαζόμενη αντίδραση απαιτείται η προηγούμενη τιμή του χρόνου t_j και της τάσης a_j της αντίδρασης αυτής. Για τον λόγο αυτό, μέσα στη μονάδα NPE διατηρούνται οι χρόνοι t_j και οι τάσεις a_j όλων των αντιδράσεων, ώστε να χρησιμοποιηθούν αργότερα στις πράξεις ανανέωσης των χρόνων t_j των επηρεαζόμενων αντιδράσεων. Στην αρχή κάθε επανάληψης, οι δομές αυτές είναι κενές, επομένως απαιτείται τα δεδομένα όλων των αντιδράσεων να προωθηθούν στην μονάδα NPE ώστε να αρχικοποιηθούν. Η μονάδα ελέγχου NPCU στην αρχή κάθε επανάληψης ενεργοποιεί το σήμα *PE_initialize* που σηματοδοτεί την εκκίνηση της διαδικασίας αυτής. Η μονάδα NPE, όταν ολοκληρωθεί η αρχικοποίηση των δομών δεδομένων, έχει αποφασίσει για την πρώτη νικητήρια αντίδραση και ενημερώνει την μονάδα NPCU για την ολοκλήρωση της αρχικοποίησης. Στο εξής, το NPE υπολογίζει τους χρόνους t_j μόνο της νικητήριας αντίδρασης και των επηρεαζόμενων αντιδράσεων. Περισσότερες πληροφορίες για το NPE θα δοθούν στο Κεφάλαιο 6.

Κανονική λειτουργία:

Δεδομένου ότι έχει ολοκληρωθεί πλέον η αρχικοποίηση του συστήματος, ο πυρήνας NRM είναι σε θέση να εκτελέσει έναν ή περισσότερους κύκλους αντιδράσεων, μέχρις ότου ολοκληρώσει όλες τις επαναλήψεις για τον επιθυμητό χρόνο προσομοίωσης. Ένας κύκλος αντιδράσεων απαρτίζεται από μια συγκεκριμένη ροή δεδομένων μέσα στον πυρήνα και στις υπομονάδες του. Όπως αναφέρθηκε στην προηγούμενη παράγραφο, στο τέλος της αρχικοποίησης των εσωτερικών δομών της μονάδας NPE προκύπτει η πρώτη νικητήρια αντίδραση. Στην αρχή κάθε κύκλου αντιδράσεων η νικητήρια αντίδραση από τον προηγούμενο κύκλο αντιδράσεων είναι εκείνη που καθορίζει τα δεδομένα που θα προωθηθούν εκ νέου στην μονάδα NPE. Συγκεκριμένα, η μονάδα ελέγχου NPCU ενεργοποιεί την μονάδα DG_MS η οποία δέχεται τη διεύθυνση της νικητήριας αντίδρασης και εξάγει από τον γράφο εξαρτήσεων τις διευθύνσεις όλων των επηρεαζόμενων αντιδράσεων. Στη συνέχεια οι διευθύνσεις και τα δεδομένα της νικητήριας αντίδρασης και των επηρεαζόμενων αντιδράσεων προωθούνται στην μονάδα NPE. Η μονάδα NPE, με βάση τον αλγόριθμο, κάνει τις απαραίτητες ενέργειες προκειμένου να ανανεωθούν οι χρόνοι των παραπάνω αντιδράσεων και τελικά αποφασίζει ποια θα είναι η νέα νικητήρια αντίδραση από το σύνολο των αντιδράσεων του βιομοντέλου. Οι πληροφορίες για την νικητήρια αντίδραση προωθούνται στις μονάδες SPU και SCU, αφενός για να ανανεωθούν ο πληθυσμός των μοριακών

ειδών που συμμετέχουν στην αντίδραση και αφετέρου για να υπολογιστεί ο τρέχον χρόνος προσομοίωσης. Ένας νέος κύκλος αντιδράσεων ξεκινάει και η ίδια ροή δεδομένων επαναλαμβάνεται όσες φορές χρειαστεί μέχρι η προσομοίωση να φτάσει στον επιθυμητό χρόνο προσομοίωσης (T_{sim}) που έχει ορίσει ο χρήστης του συστήματος.

6. Η ΚΕΝΤΡΙΚΗ ΜΟΝΑΔΑ ΕΠΕΞΕΡΓΑΣΙΑΣ - NRM PROCESSING ELEMENT

Η μονάδα NRM Processing Element (NPE) αποτελεί την κεντρική μονάδα επεξεργασίας και είναι μια από τις βασικότερες μονάδες του πυρήνα NRM. Όπως έχει αναφερθεί στην Ενότητα 5.3, η μονάδα NPE δέχεται όλες τις πληροφορίες μιας αντίδρασης που απαιτούνται για να υπολογιστεί ο νέος χρόνος εκτέλεσης της αντίδρασης, είτε αυτή είναι η νικητήρια αντίδραση είτε είναι μια επηρεαζόμενη από τη νικητήρια αντίδραση.

6.1 Βασική Λειτουργία και Ροή Δεδομένων

Στη μονάδα NPE πραγματοποιούνται όλες οι αριθμητικές και λογικές πράξεις προκειμένου να προκύψει η νικητήρια αντίδραση κάθε κύκλου αντιδράσεων. Οι αριθμητικές πράξεις αφορούν την εφαρμογή των μαθηματικών τύπων του αλγορίθμου NRM, ενώ οι λογικές πράξεις είναι καθαρά πράξεις σύγκρισης και εύρεσης της αντίδρασης με τον μικρότερο αναμενόμενο χρόνο εκτέλεσης. Έτσι λοιπόν, η μονάδα NPE χωρίζεται σε δύο μονάδες, την Recalculation Unit (RU), που εκτελεί τις αριθμητικές πράξεις, και το Comparator Tree (CoT), που εκτελεί τις πράξεις σύγκρισης και εύρεσης της αντίδρασης με τον ελάχιστο χρόνο.

Τέλος, μέσα στη μονάδα NPE εμπεριέχονται δύο πολύ βασικές δομές δεδομένων:

- ο πίνακας **T** (Table T) στον οποίο αποθηκεύονται οι χρόνοι t_j όλων των αντιδράσεων.
- ο πίνακας **A** (Propensity Table) στον οποίο αποθηκεύονται οι τάσεις α_j όλων των αντιδράσεων.

Ο πίνακας **T** είναι αναπόσπαστο κομμάτι της μονάδας CoT, ενώ ο πίνακας **A** εμπεριέχεται στην μονάδα RU, έτσι ώστε να διευκολύνεται η ανάγνωση και εγγραφή από και προς τους πίνακες αυτούς.

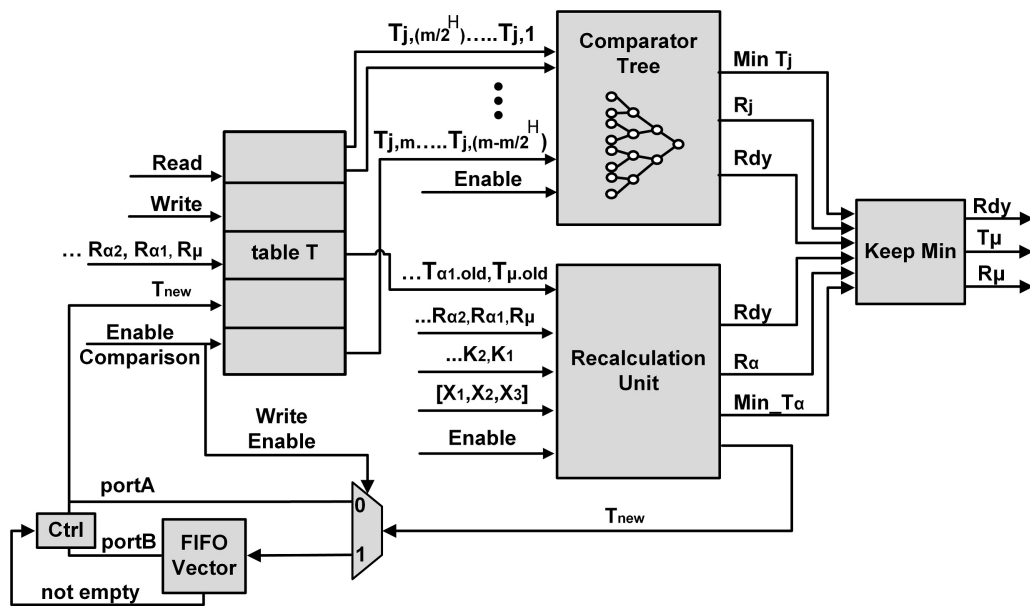
Σε κατάσταση αρχικοποίησης των δομών, η μονάδα RU ενεργοποιείται με το σήμα *PE_initialize*, δέχεται στην είσοδό της όλες τις πληροφορίες για όλες τις αντιδράσεις, υπολογίζει και αποθηκεύει τις τάσεις α_j όλων των αντιδράσεων, και τελικά υπολογίζει και προωθεί στον πίνακα **T** τους χρόνους t_j όλων των αντιδράσεων. Όσο η μονάδα RU παράγει χρόνους, τους συγκρίνει, έτσι ώστε στο τέλος της φάσης της αρχικοποίησης να έχει βρεθεί η πρώτη νικητήρια αντίδραση.

Κατά την εκτέλεση ενός κύκλου αντιδράσεων, ο πίνακας **T** τροφοδοτεί τη μονάδα RU με τους παλιούς χρόνους της νικητήριας αντίδρασης ($t_{\mu,old}$) και των επηρεαζόμενων αντιδράσεων ($t_{\alpha,old}$), ώστε οι χρόνοι αυτοί να ανανεωθούν και να αποθηκευτούν εκ νέου. Όσο η μονάδα RU κάνει τις απαιτούμενες πράξεις, η μονάδα CoT, χρησιμοποιώντας ένα δέντρο συγκριτών, συγκρίνει του χρόνους των υπολοίπων αντιδράσεων του πίνακα **T**. Η νέα νικητήρια αντίδραση προκύπτει από τη σύγκριση του αποτελέσματος της μονάδας CoT και από την μικρότερη τιμή ανανεωμένου χρόνου εκτέλεσης που έχει υπολογίσει η μονάδα RU.

Συνοπτικά τα βήματα που εκτελούνται για την εύρεση της αντίδρασης με τον μικρότερο χρόνο εκτέλεσης:

- Στην είσοδο της μονάδας NPE διοχετεύονται οι πληροφορίες για τη νικητήρια αντίδραση και τις επηρεαζόμενες από αυτήν αντιδράσεις.
- Με βάση τις διευθύνσεις των αντιδράσεων, οι χρόνοι $t_{\mu,old}$ και $t_{\alpha,old}$ εξάγονται από τον πίνακα **T**, προωθούνται στη μονάδα RU και τελικά στιγματίζονται ώστε να μην συμμετέχουν στις συγκρίσεις της μονάδας CoT.

- Στην είσοδο της μονάδας RU διοχετεύονται οι πληροφορίες για τη νικητήρια αντίδραση και τις επηρεαζόμενες από αυτήν αντιδράσεις. Καθώς τα δεδομένα εισέρχονται, η μονάδα RU αρχίζει να υπολογίζει τους καινούργιους χρόνους.
- Όταν προωθηθούν όλοι οι χρόνοι στην είσοδο της μονάδας RU, ενεργοποιείται η μονάδα CoT.
- Στο τέλος των μονάδων υπάρχει μια έξτρα λογική σύγκρισης που κρατάει την τρέχουσα ελάχιστη τιμή. Οι ελάχιστοι χρόνοι από τις δύο μονάδες συγκρίνονται και τελικά προκύπτει η νέα νικητήρια αντίδραση.
- Η διεύθυνση της νικητήριας αντίδρασης οδηγείται στην έξοδο και η μονάδα NPE μεταβαίνει σε κατάσταση αναμονής, έως ότου ξεκινήσει ο επόμενος κύκλος αντιδράσεων.



Εικόνα 12: Η αρχιτεκτονική της μονάδας NRM Process Element (NPE).

Τα παραπάνω βήματα υλοποιούνται από τη μονάδα NRM Processing Element (NPE) της οποίας η αρχιτεκτονική απεικονίζεται στην Εικόνα 12. Στη συνέχεια κεφαλαίου, γίνεται εκτενής περιγραφή των μονάδων RU και CoT, ενώ αναφέρονται περισσότερες λεπτομέρειες για τον τρόπο λειτουργίας της μονάδας NPE.

6.2 Πίνακας Χρόνων Ενεργοποίησης Αντιδράσεων

Στον πίνακα **T** αποθηκεύεται για κάθε αντίδραση η αναμενόμενη χρονική στιγμή ενεργοποίησής της. Σε κάθε κύκλο αντιδράσεων η αντίδραση με τον δυνητικά μικρότερο χρόνο είναι η “νικητήρια αντίδραση” (R_μ) και είναι αυτή που τελικά θα εκτελεστεί.

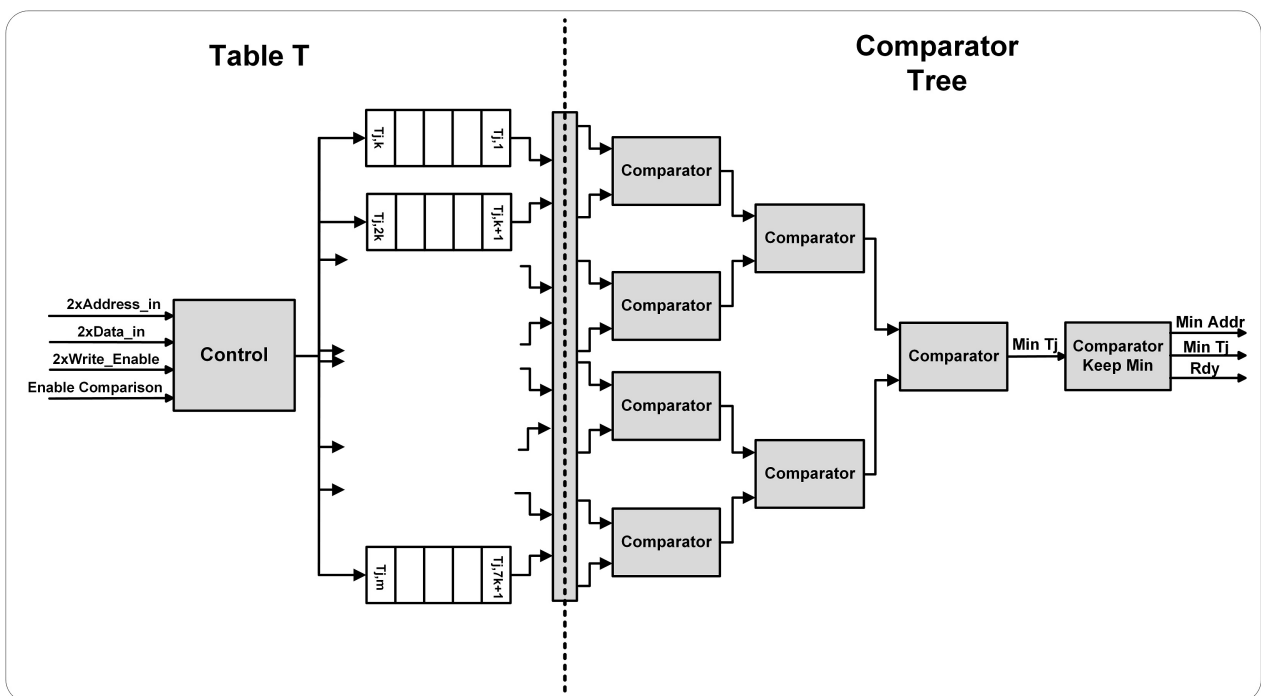
Ο πίνακας **T** και η μονάδα CoT συνθέτουν μια οντότητα που σκοπός της είναι η διαχείριση των χρόνων τ_j όλων των αντιδράσεων. Η οντότητα αυτή υποστηρίζει τρεις λειτουργίες:

- Ανάγνωση Δεδομένων: Η ανάγνωση των χρόνων τ_j των αντιδράσεων και η διοχέτευσή τους προς τη μονάδα RU.
- Εγγραφή Δεδομένων: Η εγγραφή των νέων χρόνων τ_j που προκύπτουν από υπολογισμούς της μονάδας RU. Ενεργοποιείται το σήμα εγγραφής δεδομένων *write_enable* και χρησιμοποιείται είτε η μία είτε και οι δύο διαθέσιμες πόρτες εγγραφής δεδομένων.
- Λειτουργία Σύγκρισης: Ενεργοποιείται η μονάδα CoT (δέντρο συγκριτών) ώστε να βρεθεί ο ελάχιστος χρόνος τ_j του πίνακα **T**. Ενεργοποιείται το σήμα

enable_comparison και τα δεδομένα του πίνακα διοχετεύονται σταδιακά στο δέντρο συγκριτών.

Δεδομένου ότι από μια μνήμη RAM μπορούμε να διαβάσουμε μέχρι δύο δεδομένα σε κάθε κύκλο ρολογιού, και προκειμένου να μεγιστοποιηθεί το επίπεδο της παραλληλίας, ο πίνακας **T** έχει καταμετρηθεί σε επιμέρους τμήματα. Ο λόγος είναι κυρίως για να πραγματοποιείται ταυτόχρονη διοχέτευση δεδομένων από τον πίνακα **T** στα φύλλα του δέντρου συγκριτών. Συνεπώς, αν 2^H είναι το πλήθος των φύλλων του δέντρου, αντίστοιχα σε τόσα κομμάτια χωρίζεται και ο πίνακας **T**. Κάθε τμήμα του πίνακα αποθηκεύει $m/2^H$ χρόνους αντιδράσεων και εν τέλει $m/2^{(H+1)}$ κύκλοι ρολογιού χρειάζονται για να διοχετευτούν όλα τα δεδομένα του πίνακα **T** στα φύλλα του δέντρου συγκριτών.

Το σχηματικό της Εικόνας 13 απεικονίζει την μορφή του πίνακα **T** στην περίπτωση που το δέντρο έχει ύψος $H = 3$. Ο πίνακας **T** και το δέντρο συγκριτών ενώνονται με ένα κατάλληλου μεγέθους κανάλι δεδομένων που στόχο έχει να προωθήσει παράλληλα τα δεδομένα από τον πίνακα **T** στο δέντρο συγκριτών.



Εικόνα 13: Ενδεικτικό παράδειγμα του πίνακα **T** και του δέντρου συγκριτών (CoT) ύψους $H = 3$.

Η δομή δεδομένων του πίνακα **T**, για λόγους ασφαλείας, μπορεί να διαβάσει και να γράψει μέχρι δύο δεδομένα σε κάθε κύκλο ρολογιού ($2xDATA_in$, $2xWrite_Enable$, $2xAddress_in$), ενώ κατά τη διάρκεια συγκρίσεων (*enable_comparison*) δεν επιτρέπονται η ανάγνωση και η εγγραφή δεδομένων από/προς τον πίνακα **T**. Με αυτόν τον τρόπο διασφαλίζεται η ορθή λειτουργία της μνήμης RAM με δύο πόρτες (Dual Port RAM – DPRAM) και αποφεύγεται η σύγκρουση διευθύνσεων.

6.3 Δέντρο Συγκριτών – Comparator Tree (CoT)

Η μονάδα CoT (Comparator Tree) είναι ένα δυαδικό δέντρο συγκριτών όπου κάθε φύλλο - κόμβος είναι ένας συγκριτής δύο εισόδων. Ρυθμίζοντας κατάλληλα το ύψος του δέντρου ρυθμίζουμε τον συνολικό αριθμό των κύκλων ρολογιού που απαιτούνται προκειμένου να προωθηθούν όλα τα δεδομένα από τον πίνακα **T** προς το δέντρο και να καθοριστεί η αντίδραση με το μικρότερο t_j του πίνακα **T**.

Όπως αναφέρθηκε προηγουμένως, ο πίνακας **T** και το δέντρο συγκριτών στην ουσία συν-

θέτουν μια οντότητα ώστε να απλοποιηθεί η σχεδίαση και να απλοποιηθεί η ροή των δεδομένων από τον πίνακα **T** προς τα φύλλα του δέντρου.

Τα μεγέθη που επηρεάζονται από την παράμετρο H είναι τα εξής:

- το πλήθος των φύλλων – συγκριτές (2^H)
- το πλήθος των τμημάτων που θα επιμεριστεί ο πίνακας **T** (2^H)
- το βάθος του κάθε τμήματος του πίνακα **T** ($m/2^H$)

Συνεπώς, το ύψος του δέντρου καθορίζει το πλήθος των κύκλων που απαιτούνται για να βρεθεί στον πίνακα **T** η αντίδραση με τον δυνητικά μικρότερο χρόνο ενεργοποίησης. Δεδομένου ότι m είναι το πλήθος των αντιδράσεων και H το ύψος του δέντρου, ορίζουμε ως L_{CoT} (CoT Latency) τη συνολική καθυστέρηση σε κύκλους ρολογιού (cc) που απαιτούνται για να ολοκληρωθεί η διαδικασία σύγκρισης και ισχύει ότι:

$$L_{CoT} = \frac{m}{2^{H+1}} + H + 1 \quad (5)$$

Οι όροι της εξίσωσης είναι οι εξής:

- το βάθος της μνήμης σε κάθε φύλλο ($\frac{m}{2^{H+1}}$)
- το ύψος του δέντρου (H)
- ένας κύκλος καθυστέρησης ανάγνωσης από την μνήμη (Πίνακα **T**) (1 cc)

Ο πίνακας που ακολουθεί παρουσιάζει τη συνολική καθυστέρηση (L_{CoT}) που απαιτείται για διαφορετικό πλήθος αντιδράσεων (m) και διαφορετικό ύψος δέντρου (H). Τα κελιά που το περιεχόμενό τους είναι ελαφρώς χρωματισμένα με γκριζό χρώμα αποτελούν ακραίες ή μη αποδεκτές τιμές για τη δική μας υλοποίηση. Βασική προϋπόθεση να ισχύει η ανισότητα $L_{CoT} < L_{RU}$.

Πίνακας 6: Το πλήθος των απαιτούμενων κύκλων ρολογιού (L_{CoT}) συναρτήσει του πλήθους των αντιδράσεων m και του ύψους του δέντρου H

H	1	2	3	4	5	6	7
m							
1	2	3	4	5	6	7	8
2	3	3	4	5	6	7	8
4	3	4	4	5	6	7	8
8	4	4	5	5	6	7	8
16	6	5	5	6	6	7	8
32	10	7	6	6	7	7	8
64	18	11	8	7	7	8	8
128	34	19	12	9	8	8	9
256	66	35	20	13	10	9	9
512	130	67	36	21	14	11	10
1024	258	131	68	37	22	15	12
2048	514	259	132	69	38	23	16
4096	1026	515	260	133	70	39	24

Τέλος, είναι σημαντικό να αναφερθεί ότι το ύψος του δέντρου (H) ορίζεται πριν τη σύνθεση του πυρήνα και επιλέγεται κατάλληλα ώστε η τιμή L_{CoT} να μη ξεπερνά την συνολική καθυστέρηση της μονάδας RU , που όπως θα δούμε στη συνέχεια, ορίζεται ως $L_{RU} = 42 + D$.

Κατά αυτόν τον τρόπο η μονάδα CoT λειτουργεί παράλληλα με τη μονάδα RU, επιτυγχάνοντας έτσι την απόκρυψη της επιπλέον καθυστέρησης που χρειάζεται ώστε να βρεθεί η αντίδραση του πίνακα T με τον μικρότερο χρόνο.

6.4 Μονάδα Επανυπολογισμού – Recalculation Unit (RU)

Η μονάδα Recalculation Unit (RU) είναι κατάλληλα σχεδιασμένη ώστε να εκτελεί τους μαθηματικούς τύπους (3) και (4) του αλγορίθμου Next Reaction Method (NRM). Στη συνέχεια, αναλύεται διεξοδικά η ροή των δεδομένων και όλες οι σχεδιαστικές επιλογές, ώστε η μονάδα RU να πληροί τις προϋποθέσεις του αλγορίθμου NRM.

Θεωρώντας ότι:

- t είναι ο χρόνος προσομοίωσης την τρέχουσα χρονική στιγμή,
- α_j η τάση (propensity) μιας αντίδρασης με βάση τα δεδομένα της αντίδρασης (K, FG) και τον πληθυσμό των αντιδρώντων (X_1, X_2, X_3)
- r_j ένας τυχαίος αριθμός στο διάστημα $[0, 1]$
- $\alpha_{j,old}$ η τάση (propensity) μιας αντίδρασης στον προηγούμενο κύκλο αντιδράσεων,
- $T_{j,old}$ ο χρόνος μιας αντίδρασης στον προηγούμενο κύκλο αντιδράσεων.

Ο μαθηματικός τύπος (3) χρησιμοποιείται για την ανανέωση του χρόνου της νικητήριας αντίδρασης και δεν διαφέρει από αυτόν που χρησιμοποιεί ο FRM αλγόριθμος.

$$\tau_\mu = \left(\frac{1}{\alpha_\mu(x)}\right) * \ln\left(\frac{1}{r_\mu}\right) + t \quad (6)$$

Ο μαθηματικός τύπος (4) χρησιμοποιείται για την ανανέωση του χρόνου μιας επηρεαζόμενης αντίδρασης.

$$\tau_\alpha = \left(\frac{\alpha_{\alpha,old}}{\alpha_\alpha}\right) * (T_{\alpha,old} - t) + t \quad (7)$$

Σε αυτό το σημείο χρειάζεται να αναφέρουμε τα εξής:

- Όλοι οι αριθμοί που συμμετέχουν στις πράξεις είναι αριθμοί κινητής υποδιαστολής απλής ακρίβειας (IEEE 754)[45]
- Για την παραγωγή ψευδο-τυχαίων αριθμών χρησιμοποιήθηκε μια μηχανή βασισμένη στον αλγόριθμο Mersenne Twister [44]. Για να παράγουμε τυχαίους αριθμούς στο πεδίο τιμών $[0, 1]$ θεωρούμε τον αριθμό που παράγεται ως σταθερής υποδιαστολής και τον μετατρέπουμε σε κινητής υποδιαστολής. Με την σωστή μετατροπή, ο τελικός τυχαίος αριθμός που προκύπτει είναι κινητής υποδιαστολής απλής ακρίβειας στο επιθυμητό διάστημα.
- Για τις απαραίτητες πράξεις αριθμών κινητής υποδιαστολής χρησιμοποιούνται διαθέσιμες μονάδες της εταιρίας Xilinx (Core Generator). Οι μονάδες αυτές υλοποιούν τον πολλαπλασιασμό, την αφαίρεση, την πρόσθεση, τον πολλαπλασιασμό, τη διαίρεση, τον φυσικό λογάριθμο και την μετατροπή αριθμών από σταθερή σε κινητή υποδιαστολή.
- Όσο αναφορά τον υπολογισμό της τάσης α_j (propensity) μιας αντίδρασης του βιομοντέλου, η μονάδα Propensity Unit δέχεται ως είσοδο τον πληθυσμό των αντιδρώντων και τις παραμέτρους FG και K μιας αντίδρασης και υπολογίζει την τάση της αντίδρασης (μέχρι 3^{75} τάξης).

- Για την υλοποίηση του αλγορίθμου, οι τάσεις όλων των αντιδράσεων αποθηκεύονται σε έναν τοπικό πίνακα ώστε να επαναχρησιμοποιηθούν σε μετέπειτα υπολογισμούς. Ο πίνακας αυτός (Propensity Table) έχει βάθος m , όσες είναι οι αντιδράσεις του μοντέλου και υλοποιείται με μνήμες RAM (Dual Port).

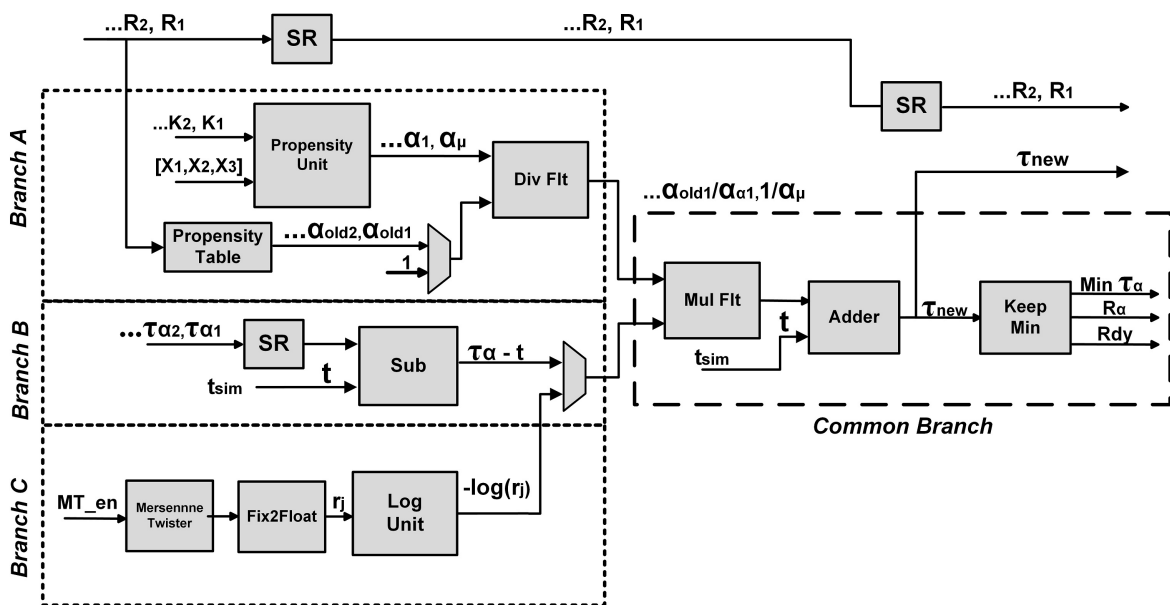
Πίνακας 7: Τα σήματα εισόδου και εξόδου της μονάδας Recalculation Unit (RU).

Όνομα Σήματος	Περιγραφή
Είσοδος	
Enable	Όταν είναι ενεργοποιημένο, η μονάδα RU δέχεται στην είσοδο αρχικά τα δεδομένα της νικητήριας αντίδρασης και στη συνέχεια τα δεδομένα των επηρεαζόμενων αντιδράσεων.
Initialize	Όταν είναι ενεργοποιημένο, η μονάδα RU δέχεται στην είσοδο τα δεδομένα όλων των αντιδράσεων του βιομοντέλου ώστε να πραγματοποιηθεί η αρχικοποίηση των δομών του NPE.
Seed, SetSeed, MT_reset	Σήματα ελέγχου της μονάδας παραγωγής τυχαίων αριθμών, Mersenne Twister. Ενεργοποιούν την εγγραφή (Seed, SetSeed) ή την επαναφορά της μονάδας σε αρχική κατάσταση (MT_reset).
FG, K	Παράμετροι που επηρεάζουν τον υπολογισμό της τάσης α_j της εκάστοτε αντίδρασης.
X1, X2, X3	Οι πληθυσμοί των αντιδρώντων της εκάστοτε αντίδρασης.
NReactions	Το πλήθος των αντιδράσεων του μοντέλου (m).
NDRactions	Το πλήθος των επηρεαζόμενων αντιδράσεων της νικητήριας αντίδρασης (D).
T_old, R_old_address	Ο χρόνος $T_{j,old}$ και η διεύθυνση (R_μ, R_{a1}, \dots) της εισερχόμενης αντίδρασης.
Tsim	Η τρέχουσα τιμή του χρόνου προσομοίωσης.
Έξοδος	
T_new, R_new_address	Ο ανανεωμένος χρόνος $T_{j,new}$ και η διεύθυνση (R_μ, R_{a1}, \dots) της εξερχόμενης αντίδρασης.
T_new_flag	Όταν ενεργοποιηθεί η μονάδα RU ενημερώνει το υψηλότερο επίπεδο της μονάδας NPE ότι στην έξοδο της μονάδας T_new και R_new_address πρέπει να προωθηθούν στον πίνακα T.
Min_Tj	Ο μικρότερος χρόνος $T_{j,new}$ που προέκυψε από τους υπολογισμούς της μονάδας RU.
Min_Tj_addr	Η διεύθυνση της αντίδρασης με τον μικρότερο χρόνο εκτέλεσης που δίνεται στην έξοδο.
Min_Tj_Rdy	Όταν το σήμα Min_Tj_Rdy ενεργοποιηθεί τότε στην έξοδο (Min_Tj, Min_Tj_addr) έχει δοθεί η αντίδραση με τον μικρότερο χρόνο εκτέλεσης που υπολογίστηκε από τη μονάδα RU.

Το σχηματικό της Εικόνας 14 παρουσιάζει τη ροή των δεδομένων στη μονάδα RU. Η ροή των δεδομένων χωρίζεται σε τρεις διακλάδωσεις. Η πρώτη διακλάδωση (Branch A) αφορά τον υπολογισμό της τάσης μιας αντίδρασης (α_j , Propensity Unit) και τη διαίρεση των αριθμών $\frac{1}{\alpha_j}$ και $\frac{\alpha_{j,old}}{\alpha_j}$. Η δεύτερη διακλάδωση (Branch B) αφορά μόνο τις επηρεαζόμενες

αντιδράσεις και υπολογίζει τη διαφορά $\tau_{j,old} - t$. Τέλος, η τρίτη διακλάδωση (*Branch C*) υπολογίζει τον λογάριθμο ενός τυχαίου αριθμού $\ln(\frac{1}{r_j})$ και ενεργοποιείται τόσο για την νικητήρια αντίδραση όσο και για κάθε επηρεαζόμενη αντίδραση που η προηγούμενη τιμή της τάσης $\alpha_{\alpha,old} = 0$. Συγκεκριμένα, αν $\alpha_{\alpha,old} \neq 0$ τότε η RU επανυπολογίζει τον χρόνο ενεργοποίησης μιας επηρεαζόμενης αντίδρασης με βάση τον μαθηματικό τύπο (4) ενεργοποιώντας κατάλληλα την πρώτη και τη δεύτερη διακλάδωση, διαφορετικά με βάση τον μαθηματικό τύπο (3) ενεργοποιώντας κατάλληλα την πρώτη και τη τρίτη διακλάδωση. Σε κάθε περίπτωση όλες οι έξοδοι των τριών διακλαδώσεων καταλήγουν μέσω πολυπλεξίας σε μια κοινή διακλάδωση (*Common Branch*) ώστε να ολοκληρωθούν όλες οι απαιτούμενες μαθηματικές πράξεις.

Στην έξοδο της μονάδας RU έχει τοποθετηθεί μια μονάδα σύγκρισης (*Keep Min*) έτσι ώστε όταν όλες οι πράξεις ολοκληρωθούν η μονάδα RU να έχει κρατήσει την αντίδραση με τον μικρότερο ανανεωμένο χρόνο. Το αποτέλεσμα της μονάδας RU θα συγκριθεί στη συνέχεια με το αποτέλεσμα του δέντρου συγκριτών, ενώ οι ανανεωμένοι χρόνοι $\tau_{j,new}$ θα προωθηθούν προς αποθήκευση στον πίνακα **T**.



Εικόνα 14: Η αρχιτεκτονική της Recalculation Unit.

Με βάση την αρχιτεκτονική της μονάδας RU (Εικόνα 14), οι περισσότερες μονάδες κινητής υποδιαστολής τοποθετούνται σειριακά, η μία μετά την άλλη, ώστε η έξοδος της μίας να είναι η είσοδος της επόμενης μονάδας. Γνωρίζοντας ότι κάθε μονάδα που εκτελεί αριθμητικές πράξεις με αριθμούς κινητής υποδιαστολής απαιτεί ένα συγκεκριμένο πλήθος κύκλων ρολογιού, καταλήγουμε στο εύλογο συμπέρασμα ότι η μονάδα RU απαιτεί ένα πολύ συγκεκριμένο πλήθος κύκλων ρολογιού για την ολοκλήρωση της διαδικασίας ανανέωσης του χρόνου μιας αντίδρασης.

Επίσης, το εργαλείο της Xilinx (Core Generator), που παράγει τις αριθμητικές μονάδες κινητής υποδιαστολής, μας δίνει τη δυνατότητα να αυξομειώσουμε τα επίπεδα σωλήνωσης (pipeline) και αντίστοιχα το απαιτούμενο πλήθος κύκλων ρολογιού (καθυστέρηση) της εκάστοτε μονάδας. Πρέπει να γίνει κατανοητό ότι ο βαθμός σωλήνωσης των μονάδων επιλέχθηκε με στόχο τον καλύτερο δυνατό συνδυασμό: τελικής συχνότητας λειτουργίας και απαιτούμενο πλήθος κύκλων ρολογιού. Οι δύο αυτές τιμές είναι αντιστρόφως ανάλογες, διότι όσο αυξάνονται τα επίπεδα σωλήνωσης στο εσωτερικό μιας μονάδας τόσο αυξάνεται η συχνότητα της μονάδας (ισχύει και το αντίστροφο). Η υπολογιστή ισχύς ενός συστήμα-

τος, όμως, δεν εξαρτάται μόνο από τη συχνότητα λειτουργίας του, αλλά, και από το πλήθος των κύκλων ρολογιού που απαιτούνται ώστε να ολοκληρωθούν οι πράξεις με τη μεγαλύτερη συχνότητα εκτέλεσης. Στη δικής μας περίπτωση, όλες πράξεις κινητής υποδιαστολής εκτελούνται μέσα στη μονάδα RU.

Στον Πίνακα 8 που ακολουθεί παρουσιάζεται το πλήθος των απαιτούμενων κύκλων ρολογιού για κάθε μία από της αριθμητικές μονάδες που συνθέτουν τη μονάδα RU. Ο Πίνακας 8 είναι χωρισμένος με τέτοιο τρόπο ώστε να απεικονίζεται η συνολική καθυστέρηση για κάθε μια από τις διακλαδώσεις της RU, δίνοντας έμφαση στον συγχρονισμό των διακλαδώσεων.

Πίνακας 8: Οι απαιτούμενοι κύκλοι ρολογιού των διακλαδώσεων της RU.

		<i>RU Latency</i>							
<i>BRANCH</i>	<i>Input</i>	<i>Branch Components Latency</i>				<i>Common Branch</i>		<i>Output</i>	<i>Total</i>
A	Enable Register 1	Propensity		Div	Mult	Adder	Register	42	
		14	17	4	5	1			
B	Enable Register 1	Sub		SR	Mult	Adder	Register		
		11	20	4	5	1			
C	Enable Register 1	Control Old	MT	Fix2Float	Log	Mult	Adder	Register	
		4	1	6	20	4	5	1	

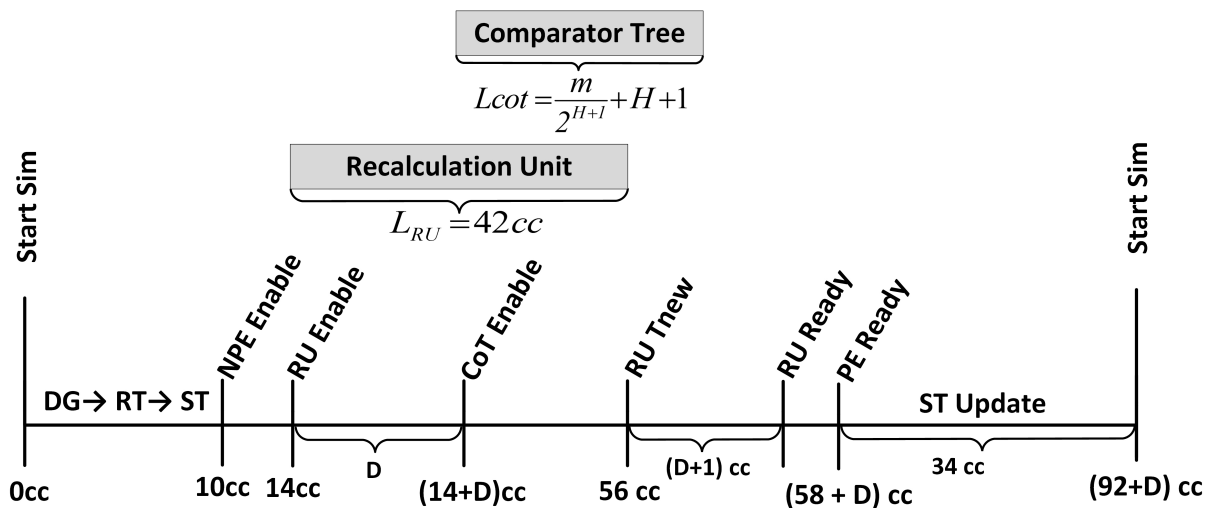
Συνεπώς, προκειμένου η RU να επαναυπολογίσει τον χρόνο ενεργοποίησης μιας αντίδρασης χρειάζεται 42 κύκλους ρολογιού. Αν D είναι το πλήθος των επηρεαζόμενων αντιδράσεων τότε ορίζουμε ως $L_{RU} = 42 + D$ (cc) τη καθυστέρηση ώστε να ολοκληρωθούν όλες οι απαιτούμενες πράξεις από τη μονάδα RU. Τέλος, όπως αναφέρθηκε στην Ενότητα 6.3, είναι σημαντικό να τονίσουμε ότι με βάση τη τιμή L_{RU} και το πλήθος των αντιδράσεων (m) ορίζουμε κατάλληλα το ύψος του δέντρου συγκριτών (C_{OT}) ώστε να ισχύει $L_{COT} < L_{RU}$, στοχεύοντας έτσι στην παράλληλη λειτουργία των μονάδων και στην ελαχιστοποίηση της συνολικής καθυστέρηση του NPE.

7. ΕΠΙΔΟΣΕΙΣ - ΣΤΑΤΙΣΤΙΚΑ

7.1 Θεωρητική Ανάλυση Επιδόσεων

Στις προηγούμενες ενότητες έγινε εκτενής παρουσίαση των τεχνικών περιγραφών όλων των κύριων μονάδων του NRM πυρήνα και του NRM SoC. Στην ενότητα αυτή παρουσιάζονται τα αποτελέσματα της θεωρητικής ανάλυσης της καθυστέρησης λόγω διοχέτευσης (latency) της αρχιτεκτονικής, καθώς και πώς αυτή επηρεάζει την συνολική απόδοση του συστήματος.

Όπως έχει οριστεί σε προηγούμενη ενότητα, κατά την προσομοίωση ενός μοντέλου το σύστημα μεταβαίνει από την τρέχουσα χρονική στιγμή στην επόμενη με βάση το χρόνο ενεργοποίησης της νικητήριας αντιδράσεων (*winner reaction*), της αντίδρασης δηλαδή με το μικρότερο χρόνο ενεργοποίησης. Κάθε μετάβαση καθορίζεται στο τέλος ενός κύκλου αντίδρασης (*reactions cycle, RC*) μετά από συγκεκριμένο πλήθος κύκλων ρολογιού που είναι ανάλογο του πλήθους των επηρεαζόμενων αντιδράσεων (D). Έχοντας μελετήσει διεξοδικά την αρχιτεκτονική της επεξεργαστικής μονάδας του NRM SoC είμαστε πλέον σε θέση να υπολογίσουμε την απαιτούμενη καθυστέρηση λόγω διοχέτευσης (latency) για να ολοκληρωθεί ένας κύκλος αντιδράσεων και να εκτιμήσουμε την απόδοση του συστήματος.



Εικόνα 15: Η ανάλυση της καθυστέρησης λόγω διοχέτευσης του πυρήνα NRM.

Όπως υποδεικνύει η Εικόνα 15, ένας κύκλος αντιδράσεων ξεκινάει με την ενεργοποίηση του σήματος *start_sim* και ολοκληρώνεται με την εκκίνηση του επόμενου κύκλου αντίδρασης. Η ροή των δεδομένων ξεκινάει από τον γράφο εξαρτήσεων (DG), από όπου εξάγονται οι διευθύνσεις των επηρεαζόμενων αντιδράσεων, στη συνέχεια καταλήγουν στον πίνακα των αντιδράσεων (RT) και στον πίνακα των μοριακών ειδών (ST) από όπου εξάγονται όλα τα δεδομένα που συμμετέχουν στον επαναυπολογισμό του χρόνου ενεργοποίησης της εκάστοτε επηρεαζόμενης αντίδρασης.

Στη συνέχεια ενεργοποιείται η κεντρική επεξεργαστική μονάδα του πυρήνα (NPE), η οποία εκτελεί όλες τις αριθμητικές και λογικές πράξεις που χρειάζονται ώστε να προκύψει η επόμενη νικητήρια αντίδραση. Όπως φαίνεται στο διάγραμμα της Εικόνας 15, αρχικά ενεργοποιείται η μονάδα RU προκειμένου να διαβαστούν και να στιγματιστούν οι D χρόνοι των αντιδράσεων από τον πίνακα **T**. Αφού αποδεσμευτεί ο πίνακας **T**, ενεργοποιείται το δέντρο συγκριτών CoT ώστε να βρεθεί η αντίδραση με το μικρότερο χρόνο t_j του πίνακα **T**. Το ύψος του δέντρου καθορίζεται αποκλειστικά από το πλήθος των αντιδράσεων (m) ώστε να ισχύει $L_{CoT} < L_{RU}$ (Ενότητα 6.3). Κατά αυτό τον τρόπο, το CoT ολοκληρώνει τις συγκρίσεις πριν η RU ολοκληρώσει τις αριθμητικές πράξεις, αποφεύγοντας έτσι την επι-

πλέον καθυστέρηση που απαιτείται για τον καθορισμό της αντίδρασης με τον μικρότερο χρόνο ενεργοποίησης του πίνακα **T**. Συνεπώς, ορίζουμε ως καθυστέρηση (latency) της μονάδας NPE ως $L_{NPE} = \max\{L_{RU}, L_{CoT}\} + D + 1 + 6 = L_{RU} + D + 7$ (cc), άρα $L_{NPE} = 49 + D$ (cc).

Όταν η μονάδα NPE υπολογίσει την επόμενη νικητήρια αντίδραση, τότε ο πυρήνας NRM μεταβαίνει στην κατάσταση ανανέωσης των πληθυσμών των μοριακών ειδών με βάση την στοιχειομετρία που ορίζει η νικητήρια αντίδραση. Για τον λόγο αυτό απαιτούνται επιπλέον 34 κύκλοι ρολογιού μέχρις ότου το σύστημα να είναι έτοιμο να ξεκινήσει τον επόμενο κύκλο αντίδρασης.

Εν κατακλείδι, ένας κύκλος αντίδρασης απαιτεί συνολικά L_{RC} κύκλους ρολογιού (cc) προκειμένου να καθοριστεί η επόμενη νικητήρια αντίδραση και το σύστημα να μεταβεί στην επόμενη κατάσταση. Στο σημείο αυτό πρέπει να τονίσουμε το γεγονός ότι ο τελικός απαιτούμενος χρόνος ενός κύκλου αντιδράσεων είναι ανεξάρτητος από το μέγεθος του μοντέλου (m) και καθορίζεται από το πλήθος των επηρεαζόμενων αντιδράσεων (D). Στην περίπτωση ενός πραγματικού μοντέλου, όπως το ASYN [3], το D είναι πιθανό να διαφέρει σε κάθε κύκλο αντίδρασης και για το λόγο αυτό ορίζουμε τον μέσο όρο επηρεαζόμενων αντιδράσεων D_{aver} .

Έχοντας αναλύσει την καθυστέρηση της κάθε μονάδας αλλά και τον συνολικό χρόνο καθυστέρησης ενός κύκλου αντιδράσεων, είμαστε πλέον σε θέση να εκτιμήσουμε την συνολική απόδοση του NRM Core. Δεδομένου ότι η μετάβαση του συστήματος από τη μια κατάσταση στην επόμενη εξαρτάται από την χρονική καθυστέρηση L_{RC} , χρησιμοποιούμε αυτό το μέγεθος για τη θεωρητική μελέτη της διεκπεραιωτικής ικανότητας (Reaction Throughput) του NRM Core:

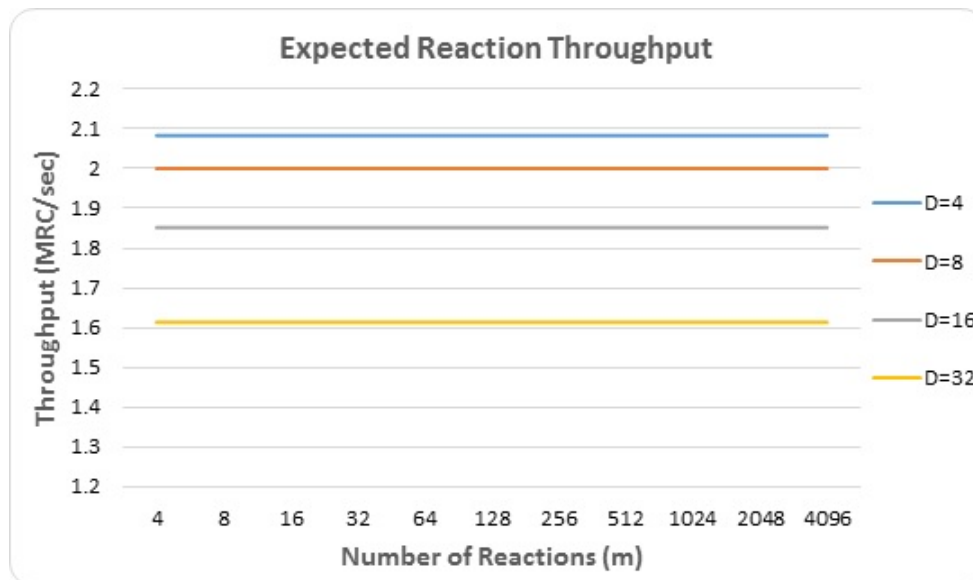
- **Reaction Throughput (Mega Reactions Cycles / sec):** Το μέγεθος αυτό καθορίζει το πόσα εκατομμύρια κύκλοι αντιδράσεων (RCs) του δοθέντος μοντέλου ολοκληρώνονται σε ένα δευτερόλεπτο από το NRM Core (σε **MRC/sec**).

Για την εκτίμηση του MRC/sec θα πρέπει πρώτα να έχουν καθοριστεί τόσο το μέγεθος L_{RC} όσο και η συχνότητα λειτουργίας του συστήματος. Ο μαθηματικός τύπος (8) δίνει το Reaction Throughput ενός πυρήνα του συστήματος σε MRC/sec, αποδεικνύονται ότι εξαρτάται άμεσα από το πλήθος των επηρεαζόμενων αντιδράσεων (D_{aver}) των RCs ενός βιομοντέλου και όχι από το συνολικό πλήθος των αντιδράσεων (m):

$$Reaction\ Throughput = \frac{1}{(92 + D_{aver})t_{cycle} 10^6} (MRC/sec) \quad (8)$$

Στη συνέχεια ακολουθεί το διαγράμματα για την θεωρητική επίτευξη της διεκπεραιωτικής ικανότητας (Reaction Throughput) ενός NRM Core για διάφορους συνδυασμούς αριθμών αντιδράσεων του βιομοντέλου (m) και μέσο όρο επηρεαζόμενων αντιδράσεων (D_{aver}).

Όπως παρατηρούμε από το διάγραμμα της Εικόνας (16), με συχνότητα λειτουργίας τα 200MHz ($t_{cycle} = 5ns$) η διεκπεραιωτική ικανότητα (Reaction Throughput) του NRM Core δεν επηρεάζεται από το πλήθος των αντιδράσεων (m) αλλά από το μέσο όρο των επηρεαζόμενων αντιδράσεων D_{aver} του εκάστοτε μοντέλου. Κατά αυτό τον τρόπο επιβεβαιώνεται η χρησιμότητα του NRM έναντι του FRM για βιομοντέλα με $D_{aver} \ll m$. Ωστόσο, είναι κατανοητό ότι όσο αυξάνεται το D_{aver} ομοίως αυξάνεται το μέγεθος L_{RC} και συνεπώς μειώνεται η διεκπεραιωτική ικανότητα του συστήματος.



Εικόνα 16: Θεωρητικά αναμενόμενη διεκπεραιωτική ικανότητα (Reaction Throughput) του NRM SoC συναρτήσει του πλήθους των αντιδράσεων (m) και του μέσου όρου επηρεαζόμενων αντιδράσεων (D_{aver}). Συχνότητα λειτουργίας 200MHz.

Τέλος, αν το NRM SoC περιλαμβάνει περισσότερους από έναν πυρήνες NRM τότε πολλαπλασιάζουμε τη διεκπεραιωτική ικανότητα με τον αριθμό των πυρήνων (C) ώστε να προκύψει η συνολική διεκπεραιωτική ικανότητα του NRM SoC.

$$Total\ Reaction\ Throughput = C * \frac{1}{(92 + D_{aver})t_{cycle}10^6} (MRC/sec) \quad (9)$$

7.2 Στατιστικά Υλοποίησης

Στην ενότητα αυτή παρουσιάζονται τα στατιστικά υλοποίησης του NRM SoC με έναν ή περισσότερους πυρήνες για διάφορες τιμές των VHDL Generic παραμέτρων. Για τη σύνθεση και την υλοποίηση του συστήματος χρησιμοποιήθηκε το λογισμικό ISE 14.7 [37] της εταιρίας Xilinx [35] και όλα τα στατιστικά που παρουσιάζονται σε αυτήν την ενότητα έχουν προκύψει από το στάδιο της υλοποίησης (Implementation). Η υλοποίηση του συστήματος έγινε με ένα μεσαίου μεγέθους FPGA, το XC7K325T-2FFG900 [46] της οικογένειας Kintex7 [36] της εταιρίας Xilinx. Ακολουθεί ένας πίνακας με τα κύρια χαρακτηριστικά του FPGA XC7K325T.

Πίνακας 9: Χαρακτηριστικά του FPGA XC7K325T.

XC7K325T					
Flip-Flops	LUTs	Slices	BRAMs (36Kbit)	DSPs	IOBs
407,600	203,800	50,950	445	840	500

Ακολουθούν οι πίνακες με τα στατιστικά υλοποίησης του NRM SoC με έναν πυρήνα ($C = 1$) για διαφορετικές διαμορφώσεις με βάση τις γενικές παραμέτρους: πλήθος αντιδράσεων (m), πλήθος μοριακών ειδών (n) και μέσο όρο επηρεαζόμενων αντιδράσεων (D_{aver}).

Πίνακας 10: Τα στατιστικά υλοποίησης του NRM SoC ($C = 1$) για $m = n = 512$ και διαφορετικά D_{aver} .

NRM SoC (C = 1) m = n = 512 (H = 3)					
D_{aver}	Flip-Flops	LUTs	BRAMs (36Kbit)	DSPs	Freq. (MHz)
4	16,062 (3%)	15,726 (7%)	36 (7.5%)	33 (3%)	240
8	16,151 (3%)	15,793 (7%)	37 (8%)	33 (3%)	230
16	16,644 (4%)	16,266 (7%)	39 (8.5%)	33 (3%)	230
32	17,278 (4%)	16,892 (8%)	43 (9%)	33 (3%)	220

Πίνακας 11: Τα στατιστικά υλοποίησης του NRM SoC ($C = 1$) για $m = n = 1024$ και διαφορετικά D_{aver} .

NRM SoC (C = 1) m = n = 1024 (H = 4)					
D_{aver}	Flip-Flops	LUTs	BRAMs (36Kbit)	DSPs	Freq. (MHz)
4	16,388 (4%)	16,208 (8%)	45 (9.5%)	33 (3%)	230
8	16,997 (4%)	16,445 (8%)	46 (10%)	33 (3%)	220
16	17,319 (4%)	16,816 (8%)	48 (10.5%)	33 (3%)	220
32	19,156 (4%)	18,338 (9%)	53 (11%)	33 (3%)	220

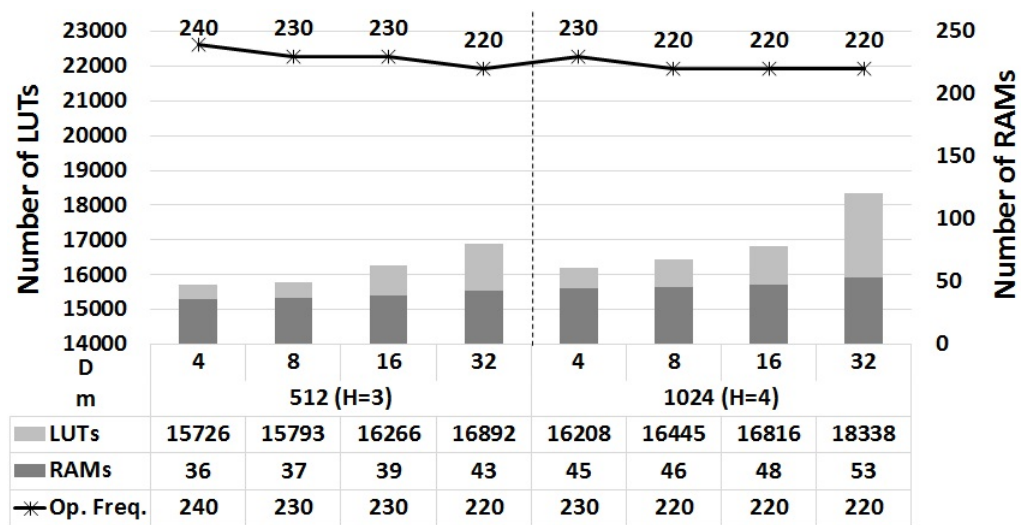
Πίνακας 12: Τα στατιστικά υλοποίησης του NRM SoC ($C = 1$) για $m = n = 2048$ και διαφορετικά D_{aver} .

NRM SoC (C = 1) m = n = 2048 (H = 5)					
D_{aver}	Flip-Flops	LUTs	BRAMs (36Kbit)	DSPs	Freq. (MHz)
4	17,701 (4%)	17,684 (9%)	74 (17%)	33 (3%)	210
8	17,805 (4%)	17,756 (9%)	77 (17%)	33 (3%)	200
16	18,317 (4%)	18,123 (9%)	82 (18.5%)	33 (3%)	200
32	19,633 (4%)	19,411 (10%)	92 (21%)	33 (3%)	200

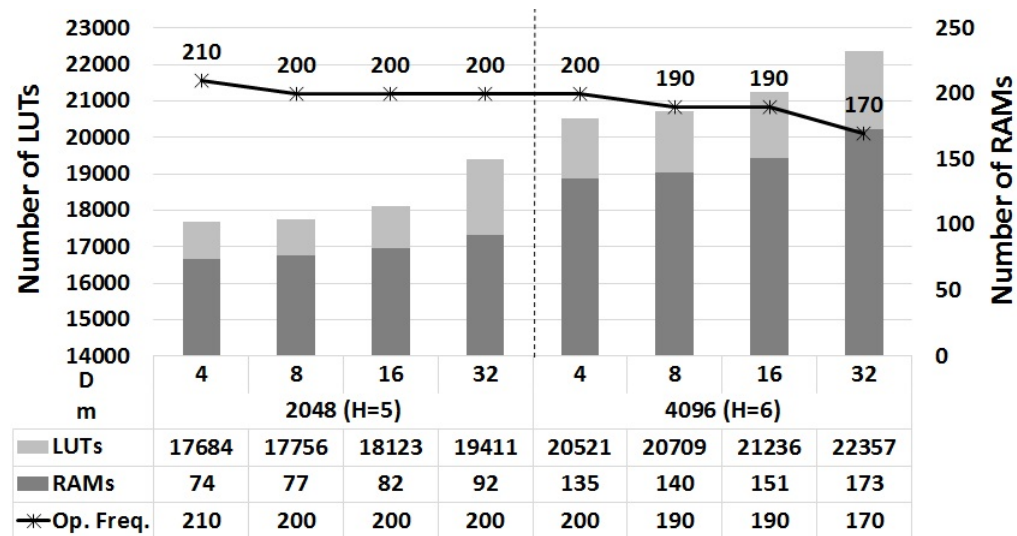
Πίνακας 13: Τα στατιστικά υλοποίησης του NRM SoC ($C = 1$) για $m = n = 4096$ και διαφορετικά D_{aver} .

NRM SoC (C = 1) m = n = 4096 (H = 6)					
D_{aver}	Flip-Flops	LUTs	BRAMs (36Kbit)	DSPs	Freq. (MHz)
4	20,471 (5%)	20521 (10%)	135 (30%)	33 (3%)	200
8	20,550 (5%)	20709 (10%)	140 (31%)	33 (3%)	190
16	21,192 (5%)	21236 (10%)	151 (34%)	33 (3%)	190
32	22,475 (5%)	22357 (11%)	173 (39%)	33 (3%)	170

Ακολουθούν διαγράμματα με τα στατιστικά υλοποίησης του NRM SoC με έναν πυρήνα ($C = 1$) όπως αυτά παρουσιάζονται στους προηγούμενους πίνακες. Από τα διαγράμματα παραλείπουμε το πλήθος των DSPs διότι παραμένει σταθερό (33 – 3%) καθώς και το πλήθος των Flip-Flops δεδομένου ότι ακολουθούν παρόμοια κλιμάκωση με τα LUTs.



Εικόνα 17: Το διάγραμμα με τα στατιστικά υλοποίησης του NRM Core για $m = 512/1024$ και διαφορετικά D_{aver} .



Εικόνα 18: Το διάγραμμα με τα στατιστικά υλοποίησης του NRM Core για $m = 2048/4096$ και διαφορετικά D_{aver} .

Μελετώντας τα παραπάνω διαγράμματα των διαφορετικών υλοποιήσεων διαπιστώνουμε την κλιμάκωση των στατιστικών καθώς αυξάνονται οι παράμετροι m , n και D_{aver} . Χαρακτηριστικό είναι επίσης το γεγονός ότι διατηρώντας σταθερό το πλήθος των αντιδράσεων (m) και αυξάνοντας το πλήθος των επηρεαζόμενων αντιδράσεων (D_{aver}) παρουσιάζεται μείωση της συχνότητας λειτουργίας του NRM – SoC. Η μείωση αυτή οφείλεται στην αύξηση του μεγέθους του γράφου εξαρτήσεων και στον τρόπο με τον οποίο το εργαλείο σύνθεσης συνδυάζει τις BRAMs ώστε να παράγει το επιθυμητό μέγεθος του γράφου. Όπως έχει αναφερθεί στην Ενότητα 5.4, ο γράφος εξαρτήσεων αποθηκεύεται με τη μορφή πίνακα του οποίου το πλάτος είναι $(D_{aver} + 3) * \log(m)$ και επηρεάζεται άμεσα από τη παράμετρο D_{aver} . Επομένως, όσο αυξάνεται η παράμετρος D_{aver} , περισσότερες BRAMs συνδυάζονται, προκαλώντας προβλήματα δρομολόγησης (routing problems) και επηρεάζοντας έτσι τη συχνότητα λειτουργίας του συστήματος.

Ακολουθούν οι πίνακες με τα στατιστικά υλοποίησης του NRM SoC με το μέγιστο δυνατό αριθμό πυρήνων για διαφορετικές διαμορφώσεις του συστήματος με βάση τις γενικές παραμέτρους: πλήθος αντιδράσεων (m), πλήθος μοριακών ειδών (n) και μέσο όρο επηρεαζόμενων αντιδράσεων (D_{aver}). Το πλήθος των πυρήνων (C) εξαρτάται από τις απαιτήσεις του συστήματος σε πόρους και κατά πόσο είναι δυνατόν να καλυφθούν στο διαθέσιμο FPGA.

Πίνακας 14: Τα στατιστικά υλοποίησης του NRM SoC ($C = 8$) για $m = n = 512$ και διαφορετικά D_{aver} .

NRM SoC ($C = 8$) $m = n = 512$ ($H = 3$)					
D_{aver}	Flip-Flops	LUTs	BRAMs (36Kbit)	DSPs	Freq. (MHz)
4	92,857 (22%)	99,221 (48%)	201 (45%)	33 (3%)	200
8	98,081 (24%)	102,906 (50%)	209 (47%)	33 (3%)	200
16	99,303 (24%)	104,213 (51%)	230 (51.5%)	33 (3%)	200
32	101,745 (24%)	106,625 (52%)	257 (58%)	33 (3%)	190

Πίνακας 15: Τα στατιστικά υλοποίησης του NRM SoC ($C = 8$) για $m = n = 1024$ και διαφορετικά D_{aver} .

NRM SoC ($C = 8$) $m = n = 1024$ ($H = 4$)					
D_{aver}	Flip-Flops	LUTs	BRAMs (36Kbit)	DSPs	Freq. (MHz)
4	109,569 (26%)	108,593 (53%)	269 (60.5%)	33 (3%)	180
8	110,297 (27%)	109,341 (53%)	281 (63%)	33 (3%)	160
16	111,687 (27%)	111,250 (54%)	297 (67%)	33 (3%)	170
32	114,257 (28%)	113,619 (55%)	333 (74.5%)	33 (3%)	170

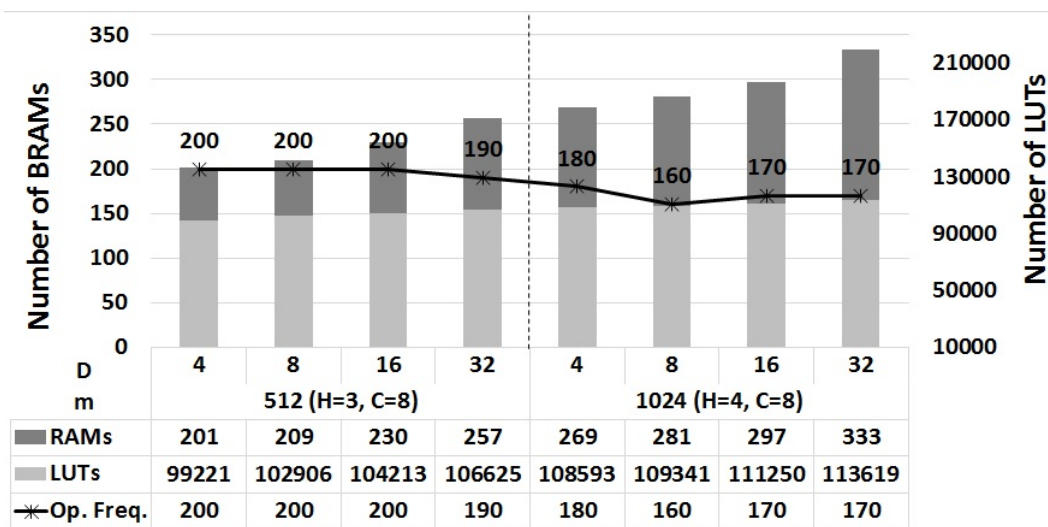
Πίνακας 16: Τα στατιστικά υλοποίησης του NRM SoC ($C = 4$) για $m = n = 2048$ και διαφορετικά D_{aver} .

NRM SoC ($C = 4$) $m = n = 2048$ ($H = 5$)					
D_{aver}	Flip-Flops	LUTs	BRAMs (36Kbit)	DSPs	Freq. (MHz)
4	58,838 (14%)	65,999 (32%)	261 (58.5%)	33 (3%)	190
8	64,948 (15%)	59,386 (29%)	271 (61%)	33 (3%)	200
16	59,890 (14%)	66,835 (32%)	291 (65%)	33 (3%)	180
32	61,238 (15%)	68,095 (33%)	329 (74%)	33 (3%)	180

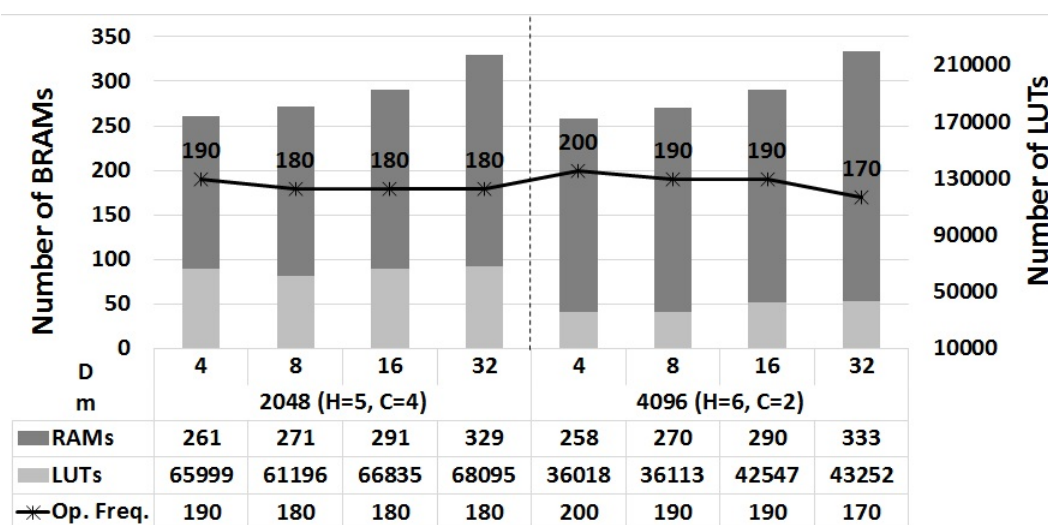
Πίνακας 17: Τα στατιστικά υλοποίησης του NRM SoC ($C = 2$) για $m = n = 4096$ και διαφορετικά D_{aver} .

NRM SoC ($C = 2$) $m = n = 4096$ ($H = 6$)					
D_{aver}	Flip-Flops	LUTs	BRAMs (36Kbit)	DSPs	Freq. (MHz)
4	36,573 (8%)	36,018 (17%)	258 (58%)	33 (3%)	200
8	36,727 (9%)	36,113 (17%)	270 (61%)	33 (3%)	190
16	40,439 (9%)	42,547 (20%)	290 (65%)	33 (3%)	190
32	41,519 (10%)	43,252 (21%)	333 (75%)	33 (3%)	170

Ακολουθούν διαγράμματα με τα στατιστικά υλοποίησης του NRM SoC για διαφορετικές τιμές των παραμέτρων m , n , D_{aver} και στο μέγιστο αριθμό πυρήνων (C) ανά υλοποίηση.



Εικόνα 19: Το διάγραμμα με τα στατιστικά υλοποίησης του NRM SoC για $m = 512/1024$ και διαφορετικά D_{aver} .



Εικόνα 20: Το διάγραμμα με τα στατιστικά υλοποίησης του NRM SoC για $m = 2048/4096$ και διαφορετικά D_{aver} .

Στα διαγράμματα των Εικόνων 19 και 20 απεικονίζονται οι απαιτήσεις του NRM SoC για διαφορετικές υλοποιήσεις με βάση τις επιλεγμένες γενικές παραμέτρους. Εύλογα συμπε-

ραίνουμε ότι όσο μικρότερες τιμές επιλέγουμε για τις παραμέτρους m και n τόσο περισσότερους πυρήνες μπορούμε να ενσωματώσουμε στο σύστημα. Ενώ, επίσης, είναι λογικό να παρουσιάζεται αυξομείωση της συχνότητας λειτουργίας καθώς αυξάνονται οι πυρήνες ή αυξάνεται η παράμετρος D_{aver} . Όπως αναφέρθηκε και προηγουμένως, η πυκνότητα του τελικού σχεδίου προκαλεί προβλήματα δρομολόγησης είτε λόγω των απαιτήσεων σε επιφάνεια του FPGA (Slices, LUTs) είτε λόγω των απαιτήσεων σε μνήμη (BRAMs).

8. ΕΠΙΚΥΡΩΣΗ ΣΧΕΔΙΑΣΗΣ

Στις προηγούμενες ενότητες παρουσιάστηκε η αρχιτεκτονική NRM SoC και τα στατιστικά υλοποίησης για διαφορετικούς συνδυασμούς των παραμέτρων. Στην παρούσα ενότητα, θα παρουσιαστεί η μεθοδολογία που ακολουθήθηκε για τον έλεγχο και την επικύρωση της λειτουργίας της αρχιτεκτονικής, τόσο στο επίπεδο ορθότητας των αποτελεσμάτων του NRM SoC, όσο και στο επίπεδο επικύρωσης της επίδοσης του συστήματος όπως αυτή έχει αναλυθεί στην Ενότητα 7.1.

Για την επικύρωση του σχεδίου χρησιμοποιήσαμε το COPASI [47], έναν δημοφιλές προσομοιωτή υλοποιημένο σε C++. Ο προσομοιωτής COPASI εκτέλεσε όλες τις απαραίτητες προσομοιώσεις σε επεξεργαστή τελευταίας γενιάς (Intel i7) με συχνότητα λειτουργίας 2.4 GHz ενώ είχε στη διάθεση του 32GB μνήμης RAM.

8.1 Επικύρωση της Ορθότητας του NRM SoC

Κατά τη σχεδίαση ενός πολύπλοκου υπολογιστικού συστήματος, όπως το NRM SoC, ο έλεγχος της λειτουργικότητας (design verification) και ο έλεγχος της ορθότητας (validation) του συστήματος παίζουν ιδιαίτερα σημαντικό ρόλο. Η διαδικασία ελέγχου της λειτουργικότητας πραγματοποιείται τόσο κατά τη σχεδίαση των επιμέρους μονάδων του συστήματος όσο και στο τελικό ολοκληρωμένο σύστημα. Στην περίπτωση μας χρησιμοποιήσαμε το λογισμικό προσομοίωσης iSim [48] της εταιρίας Xilinx, εκτελώντας αρχικά προσομοιώσεις λογικής συμπεριφοράς (behavioral simulation [49]) του κυκλώματος και στη συνέχεια προσομοιώσεις χρονισμού σε επίπεδο πύλης (timing simulation [50]). Κατά αυτό τον τρόπο, διευκολύνεται η σχεδίαση και ο συγχρονισμός των πολύπλοκων συστημάτων, και αποφεύγονται σημαντικές αστοχίες στο τελικό ενοποιημένο σχέδιο.

Αφού ολοκληρώθηκε ο έλεγχος της λειτουργικότητας του NRM SoC εκτελώντας μια πληθώρα από κατάλληλα προγράμματα επαλήθευσης (VHDL testbenches), καλούμαστε να εκτελέσουμε ένα πραγματικό πείραμα προσομοίωσης που να επικυρώνει την ορθότητα του συστήματος. Η επαλήθευση των αποτελεσμάτων πραγματοποιείται συγκρίνοντας τα αποτελέσματα του NRM SoC με τα αποτελέσματα του λογισμικού COPASI [47], εκτελώντας στοχαστική προσομοίωση αρχικά με το βιομοντέλο LVS (Lotka Volterra System)[51] και στη συνέχεια με το βιομοντέλο ASYN [3].

8.1.1 Lotka Volterra System Model

Το βιομοντέλο LVS (Lotka Volterra System) έχει τα εξής χαρακτηριστικά:

- $m = 3$
- $n = 2$
- $D_{aver} = 2$

Ακολουθούν οι παράμετροι της προσομοίωσης του βιομοντέλου LVS:

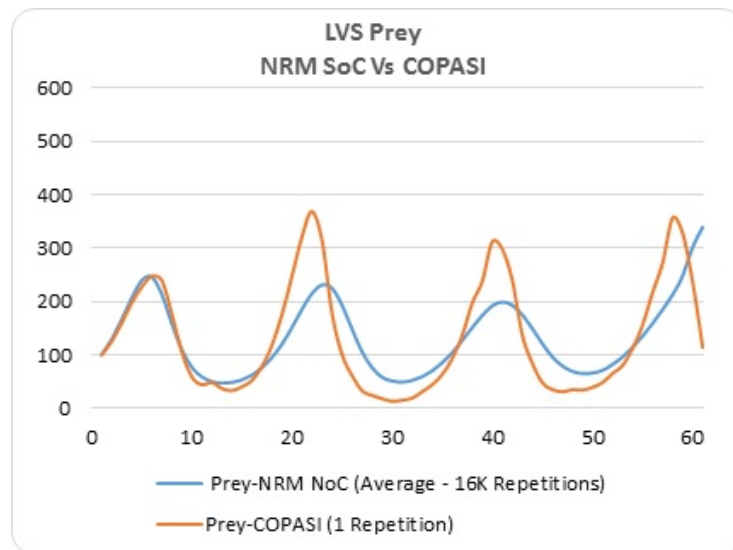
- T_{sim} : 60 s (διάρκεια προσομοίωσης)
- T_{sam} : 1 s (περίοδος δειγματοληψίας αποτελεσμάτων)
- R: 16K (στοχαστικές επαναλήψεις)
- Max Internal Steps: 1000K

Και τέλος, στον Πίνακα 21 παρατίθενται οι γενικές παράμετροι του NRM SoC όπως αυτές ορίστηκαν πριν τη σύνθεση του συστήματος και οι οποίες καλύπτουν στο έπακρον τις ανάγκες της προσομοίωσης:

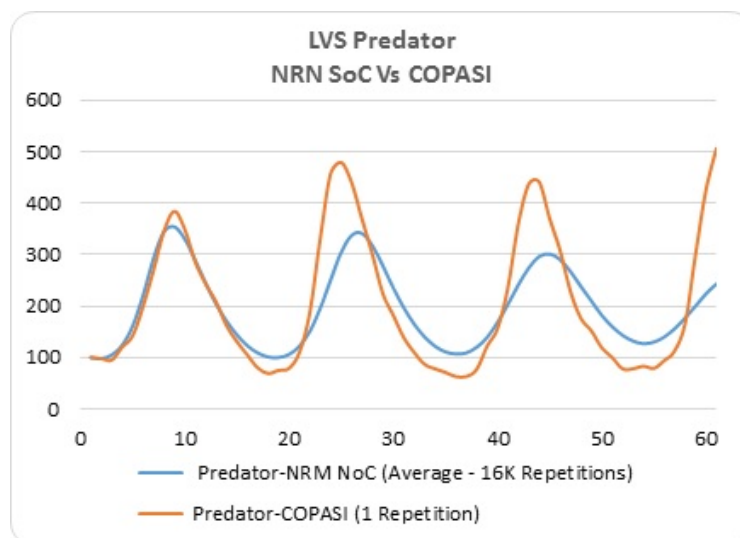
Πίνακας 18: Οι γενικές παράμετροι του NRM SoC όπως αυτές ορίστηκαν για τα πειράματα του LVS.

NRM SoC Configuration					
C	RT_SIZE	ST_SIZE	DG_SIZE	H	Frequency
16	128	128	4	1	210

Για την μεγιστοποίηση της απόδοσης του συστήματος, χρησιμοποιούμε $C = 16$ NRM πυρήνες, όπου κάθε πυρήνας είναι ανεξάρτητος, χρησιμοποιεί τη δική του αλληλουχία ψευδο-τυχαίων αριθμών και στη συγκεκριμένη περίπτωση εκτελεί 1000 επαναλήψεις της στοχαστικής προσομοίωσης. Τέλος, εκτελούμε το ίδιο πείραμα στο COPASI χρησιμοποιώντας τις ίδιες αρχικές συνθήκες και παραμέτρους. Ακολουθούν τα διαγράμματα των αποτελεσμάτων του NRM SoC και του COPASI.



Εικόνα 21: Σύγκριση των αποτελεσμάτων του στοιχείου “Θήραμα” (Prey), NRM SoC vs Copasi.



Εικόνα 22: Σύγκριση των αποτελεσμάτων του στοιχείου “Θηρευτή” (Predetar), NRM SoC vs Copasi.

Στις Εικόνες 21 και 22 συγκρίνουμε τα αποτελέσματα του NRM SoC με τα αποτελέσματα του COPASI, για τη συμπεριφορά του “θηράματος” (prey) και τη συμπεριφορά του “θηρευτή” (predator) αντίστοιχα. Και στις δύο περιπτώσεις η συμπεριφορά των μοριακών

ειδών είναι σχεδόν πανομοιότυπη, αποδεικνύοντας την ορθότητα της στοχαστικής προσομοίωσης που εκτελεί το NRM SoC.

Τέλος, συγκρίνουμε τον πραγματικό χρόνο εκτέλεσης της προσομοίωσης του NRM SoC και του COPASI, θέλοντας έτσι να δείξουμε την αύξηση της απόδοσης του συστήματός μας έναντι της σειριακής εκτέλεσης του πειράματος από τον προσομοιωτή COPASI. Στο NRM SoC η εκτέλεση $R = 16K$ στοχαστικών επαναλήψεων διήρκεσε $T_{real} = 6.01$ δευτερόλεπτα (sec) πραγματικού χρόνου, ενώ ο προσομοιωτής COPASI χρειάστηκε $T_{real} = 116$ δευτερόλεπτα (sec) για το ίδιο πείραμα. Στη δοκιμή με το βιομοντέλο LVS το σύστημα NRM SoC με 16 πυρήνες στα 210 MHz αποδεικνύεται 19.3 φορές ταχύτερο (speedup factor) σε σύγκριση με τον προσομοιωτή COPASI.

8.1.2 ASYN (Alpha-Synuclein) Model

Το βιομοντέλο ASYN (Alpha-Synuclein) [3][4] έχει τα εξής χαρακτηριστικά:

- $m = 136$
- $n = 90$
- $D_{aver} = 16$

Ακολουθούν οι παράμετροι της προσομοίωσης του βιομοντέλου LVS:

- T_{sim} : 7 ημέρες (διάρκεια προσομοίωσης)
- T_{sam} : 1 ώρα (περίοδος δειγματοληψίας αποτελεσμάτων)
- R: 16K (στοχαστικές επαναλήψεις)
- Max Internal Steps: 1000K

Και τέλος, στον Πίνακα 19 παρατίθενται οι γενικές παράμετροι του NRM SoC όπως αυτές ορίστηκαν πριν τη σύνθεση του συστήματος και οι οποίες καλύπτουν στο έπακρον τις ανάγκες της προσομοίωσης:

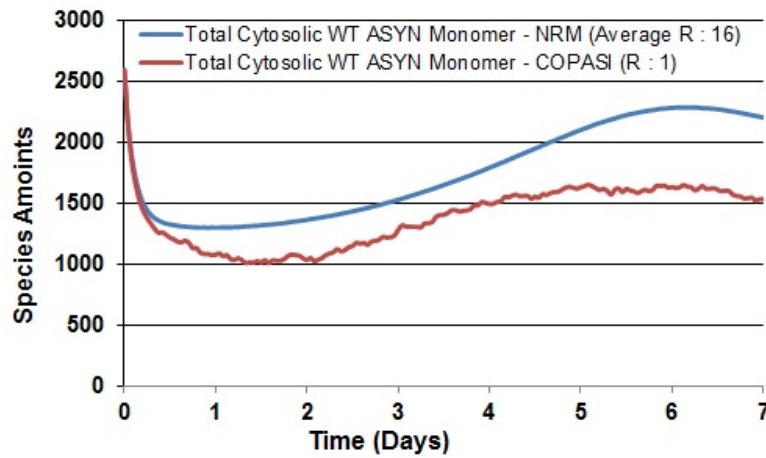
Πίνακας 19: Οι γενικές παράμετροι του NRM SoC όπως αυτές ορίστηκαν για τα πειράματα του ASYN.

NRM SoC Configuration					
C	RT_SIZE	ST_SIZE	DG_SIZE	H	Frequency
16	256	256	9	2	200

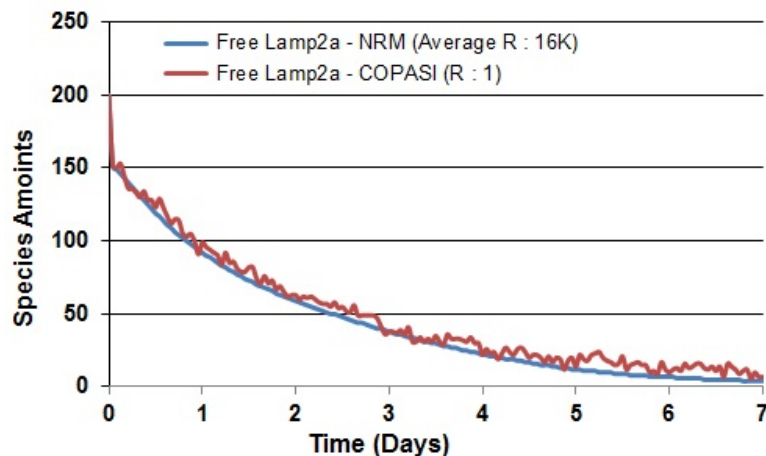
Σε αυτό το σημείο αξίζει να αναφερθεί ότι η επιλεγμένη γενική παράμετρος DG_SIZE του πυρήνα έχει επιλεγθεί με τέτοιο τρόπο ώστε η δομή δεδομένων του γράφου εξαρτήσεων να έχει αρκετές θέσεις ώστε να μπορεί καλύψει το πλήθος των διευθύνσεων που απαιτούνται από το μοντέλο που προσομοιώνεται. Στη συγκεκριμένη περίπτωση, το ASYN έχει $m = 136$ αντιδράσεις και $D_{aver} = 16$. Επομένως, απαιτούνται $136 * 16 = 2176$ διευθύνσεις για τον γράφο εξαρτήσεων του μοντέλου, ένα πλήθος που υπερκαλύπτεται από τις $256 * 9 = 2304$ διευθύνσεις που μπορεί να καλύψει η συγκεκριμένη σύνθεση του συστήματος που χρησιμοποιήσαμε.

Όπως και προηγουμένως, για την μεγιστοποίηση της απόδοσης του συστήματος, χρησιμοποιούμε $C = 16$ NRM πυρήνες, όπου κάθε πυρήνας είναι ανεξάρτητος, χρησιμοποιεί τη δική του αλληλουχία ψευδο-τυχαίων αριθμών και στη συγκεκριμένη περίπτωση εκτελεί 1000 επαναλήψεις της στοχαστικής προσομοίωσης. Τέλος, εκτελούμε το ίδιο πείραμα στο COPASI χρησιμοποιώντας τις ίδιες αρχικές συνθήκες και παραμέτρους. Ακολουθούν τα διαγράμματα των αποτελεσμάτων του NRM SoC και του COPASI.

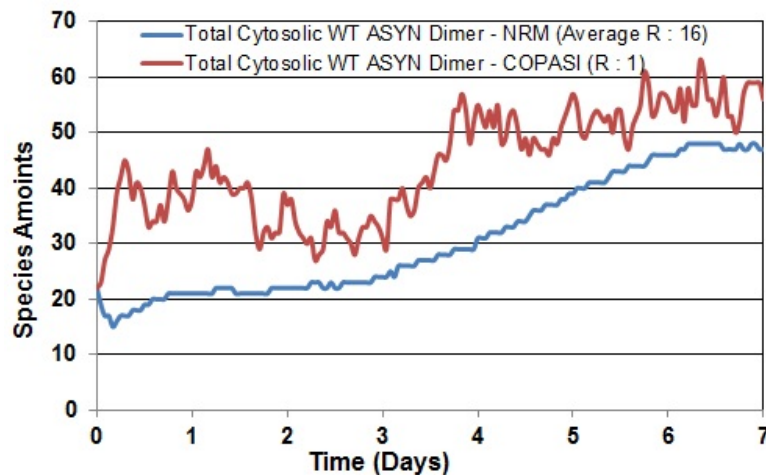
Στις Εικόνες 23 24 25 26 απεικονίζονται τα αποτελέσματα της προσομοίωσης του βιομοντέλου ASYN σε σύγκριση με τα αντίστοιχα αποτελέσματα του προσομοιωτή COPASI. Η επιλογή των μοριακών πληθυσμών που παρουσιάζονται έγινε σύμφωνα με βάση τη δημοσίευση του μοντέλου ASYN [4].



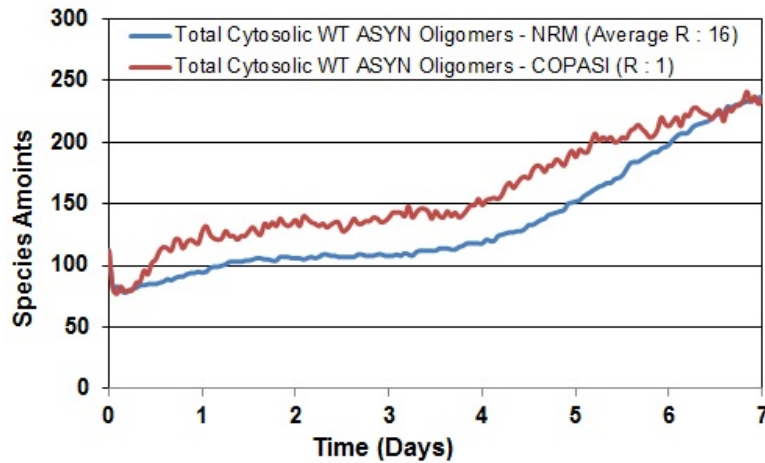
Εικόνα 23: Σύγκριση των αποτελεσμάτων του στοιχείου “Total Cytosolic WT ASYN Monomer”, NRM SoC vs Copasi.



Εικόνα 24: Σύγκριση των αποτελεσμάτων του στοιχείου “Free Lamp2a”, NRM SoC vs Copasi.



Εικόνα 25: Σύγκριση των αποτελεσμάτων του στοιχείου “Total Cytosolic WT ASYN Dimer”, NRM SoC vs Copasi.



Εικόνα 26: Σύγκριση των αποτελεσμάτων του στοιχείου “Total Cytosolic WT ASYN Oligomers”, NRM SoC vs Copasi.

Στο σημείο αυτό, απαιτείται να αναφέρουμε ότι τα αποτελέσματα του λογισμικού COPASI που παρουσιάζονται στις παραπάνω εικόνες, δεν προέρχονται από την εκτέλεση 16000 επαναλήψεων (όπως συμβαίνει για τα αποτελέσματα του NRM SoC) και συνεπώς είναι φυσιολογικό να παρουσιάζονται ορισμένες ανωμαλίες. Παρόλα αυτά, η σύγκριση των αποτελεσμάτων αποδεικνύει ξανά την ορθότητα του NRM SoC.

Θέλοντας να συγκρίνουμε την επίδοση του συστήματός μας με βάση το μοντέλο ASYN, συγκρίνουμε τον πραγματικό χρόνο εκτέλεσής του στο COPASI και στο NRM SoC αντίστοιχα. Συγκεκριμένα, στο NRM SoC η εκτέλεση $R = 16K$ στοχαστικών επαναλήψεων διήρκεσε $T_{real} = 77.6$ δευτερόλεπτα (sec) πραγματικού χρόνου, ενώ ο προσομοιωτής COPASI χρειάστηκε $T_{real} = 2,768$ δευτερόλεπτα (sec) για το ίδιο πείραμα. Στη δοκιμή με το βιομοντέλο ASYN το σύστημα NRM SoC με 16 πυρήνες στα 200 MHz αποδεικνύεται 35.67 φορές ταχύτερο (speedup factor) σε σύγκριση με τον προσομοιωτή COPASI.

Πίνακας 20: Performance Evaluation - ASYN [3] Biomodel

	COPASI ^a	NRM-SSA SoC
Repetitions	16000	16000 ^b
T_{real} (sec)	2768	77.6
Speed-Up	1	35.67
Reactions Throughput (MRC/s)	N/A	30.47
Throughput ^c (MCycles/s)	N/A	3290.76

^a Intel Core i7 4790K CPU at 4.4GHz with 32 GB of RAM under GNU/Linux OS.

^b 16 NRM-SSA Cores (200MHz)

^c Throughput is obtained by multiplying Reactions Throughput (MRC/s) by the average number of clock cycles per reaction cycle $L_{RC} = 108cc$

Τέλος, θέλοντας να μετρήσουμε την απόδοση του συστήματος (Throughput, MCycles/sec), χρησιμοποιούμε τα αποτελέσματα από τη στοχαστική προσομοίωση του ASYN. Ο λόγος που επιλέγουμε το ASYN είναι γιατί πρόκειται για ένα πραγματικό μοντέλο, αρκετά πο-

λύτπλοκο, έχει μέσο όρο αντιδράσεων $D_{aver} = 16$ και, τέλος, η στοχαστική προσομοίωση που εκτελέσαμε ανταποκρίνεται σε πραγματικές παραμέτρους ($T_{sim} = 7$ ημέρες, $T_{sam} = 1$ ώρα).

Διαπιστώνουμε λοιπόν ότι εκτελώντας στοχαστική προσομοίωση σε ένα σύστημα με 16 πυρήνες NRM, με συχνότητα λειτουργία 200MHz, η απόδοση του συστήματός μας ξεπερνά τα 3GCycles/s. Η υψηλότερη απόδοση (Throughput) που έχει αναφερθεί στη σχετική βιβλιογραφία μέχρι σήμερα.

8.2 Επικύρωση της Επίδοσης του NRM SoC

Στο τελευταίο μέρος της επικύρωσης του NRM SoC καλούμαστε να εκτελέσουμε στοχαστική προσομοίωση δοκιμαστικών μοντέλων (benchmarks) κλιμακούμενης πολυπλοκότητας. Όπως έχει αναφερθεί στην Ενότητα 7.1, το σύστημά μας δεν επηρεάζεται από την κλιμάκωση του πλήθους των αντιδράσεων (m) ενός βιομοντέλου, αλλά από τη κλιμάκωση του μέσου όρου των επηρεαζόμενων αντιδράσεων (D_{aver}). Με βάση τον μαθηματικό τύπο (16) μπορούμε να υπολογίσουμε την αναμενόμενη διεκπεραιωτική ικανότητα (Reaction Throughput) του NRM SoC, γνωρίζοντας την περίοδο λειτουργίας (t_{cycle}) του συστήματος και το D_{aver} του μοντέλου που προσομοιώνουμε.

Για τις ανάγκες της επικύρωσης της επίδοσης του NRM SoC δημιουργήσαμε δύο φόρμουλες δοκιμαστικών μοντέλων, που κλιμακώνονται τόσο στο πλήθος των αντιδράσεων (m), όσο και ως προς το πλήθος των επηρεαζόμενων αντιδράσεων (D_{aver}). Στις δύο ενότητες που ακολουθούν επιβεβαιώνουμε την επίδοση του συστήματος για κλιμακούμενο m και D_{aver} .

8.2.1 Linear Chain System (LCS)

Η πρώτη δοκιμή που στοχεύει στην επικύρωση της επίδοσης του συστήματος, χρησιμοποιεί ένα Γραμμικό Σύστημα Αλυσίδας (Linear Chain System, LCS) που έχει τη δυνατότητα να κλιμακώνεται ως προς το πλήθος των αντιδράσεων, ενώ το πλήθος των επηρεαζόμενων αντιδράσεων είναι $D_{aver} = 4$ για όλες τις αντιδράσεις. Η φόρμουλα παραγωγής του μοντέλου LCS είναι η εξής:

$$S_{i \bmod m} + S_{(i+1) \bmod m} \xrightarrow{k_i} S_{(i+2) \bmod m} + S_{(i+3) \bmod m} \quad (10)$$

Όλες οι αντιδράσεις του μοντέλου είναι 2^{ou} βαθμού, όπου $i = 1, 2, \dots, m$, με τη παράμετρο m να κυμαίνεται από 512 έως 4096 αντιδράσεις. Κάθε μια από τις αντιδράσεις όταν ενεργοποιηθεί επηρεάζει $D = 4$ αντιδράσεις, επομένως, για το συγκεκριμένο μοντέλο ισχύει $D_{aver} = 4$ ανεξαρτήτως m .

Στα πειράματα που εκτελέσαμε, επιλέξαμε το μοντέλο LCS να έχει τα εξής χαρακτηριστικά:

• m :	512	1024	2048	4096
• n :	512	1024	2048	4096
• D_{aver} :	4	4	4	4

Ακολουθούν οι παράμετροι της προσομοίωσης του βιομοντέλου LCS:

- T_{sim} : 100 δευτερόλεπτα (διάρκεια προσομοίωσης)
- T_{sam} : 1 δευτερόλεπτο (περίοδος δειγματοληψίας αποτελεσμάτων)
- R: 1 (στοχαστικές επαναλήψεις)
- Max Internal Steps: 1000K

Και τέλος, στον Πίνακα 21 οι γενικές παράμετροι των διαμορφώσεων του NRM SoC όπως

αυτές ορίστηκαν πριν τη σύνθεση του συστήματος και οι οποίες καλύπτουν στο έπακρον τις ανάγκες της προσομοίωσης:

Πίνακας 21: Οι γενικές παράμετροι του NRM SoC όπως αυτές ορίστηκαν για τα πειράματα του LCS.

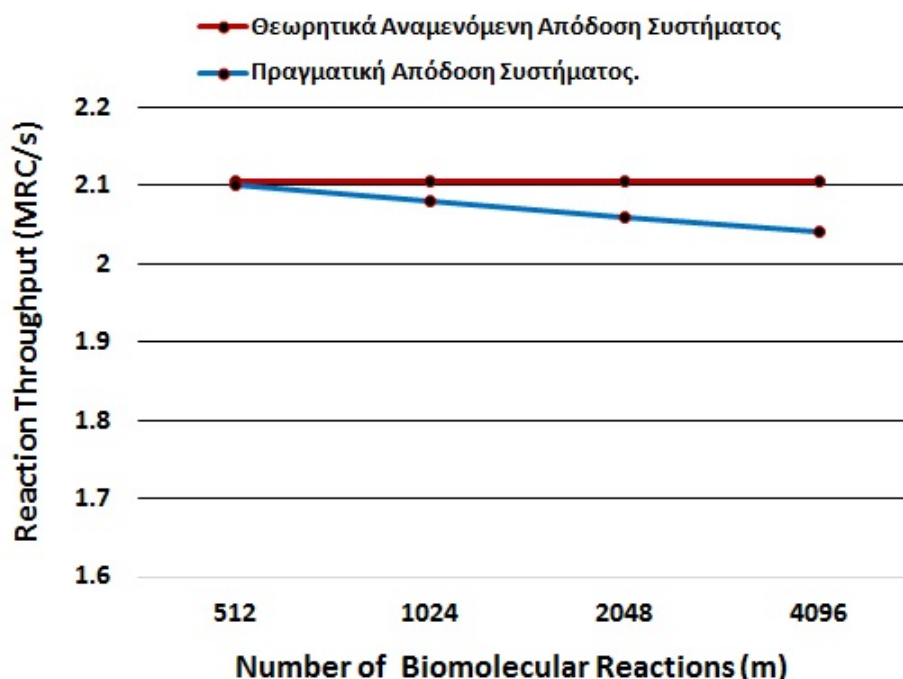
NRM SoC Configuration					
C	RT_SIZE	ST_SIZE	DG_SIZE	H	Frequency
1	512	512	4	3	200
1	1024	1024	4	4	200
1	2048	2048	4	5	200
1	4096	4096	4	6	200

Ακολουθεί ο πίνακας με τα αποτελέσματα της προσομοίωσης του μοντέλου για τις τέσσερις διαφορετικές διαμορφώσεις του συστήματος.

Πίνακας 22: Performance Evaluation - LCS models $D_{aver} = 4$

NRM SoC - xc7k325t-2ffg			
m (Number of Reactions)	T_{real} (Seconds)	Reactions Cycles (Millions)	Reactions Throughput (MRC/s)
512	48.55	102.04	2.10
1024	98.30	204.48	2.08
2048	198.59	409.37	2.06
4096	301.08	614.39	2.04

Στο διάγραμμα της Εικόνας 27, συγκρίνεται η θεωρητικά αναμενόμενη διεκπεραιωτική ικανότητα (Reaction Throughput) του συστήματος με την πραγματική απόδοση του συστήματος.



Εικόνα 27: Σύγκριση της αναμενόμενη διεκπεραιωτική ικανότητα με την πραγματική απόδοση του συστήματος, εκτελώντας το LCS για $m = 512, 1024, 2048, 4096$ και $D_{aver} = 4$

Με βάση τη θεωρητική ανάλυση, δεδομένου ότι το πλήθος των επηρεαζόμενων αντιδράσεων παραμένει σταθερό $D = 4$ τότε η απόδοση του συστήματος παραμένει σταθερή. Στο διάγραμμα της εικόνας 27 παρατηρούμε ότι η πραγματική απόδοση του συστήματος είναι πολύ κοντά στην αναμενόμενη και μειώνεται ελάχιστα καθώς αυξάνεται το πλήθος των αντιδράσεων. Η κλιμάκωση αυτή δικαιολογείται αν αναλογιστεί κανείς ότι κατά το στάδιο της αρχικοποίησης, καθώς αυξάνεται το m , απαιτείται περισσότερος χρόνος ώστε το σύστημα να αρχικοποιήσει όλες τις δομές του συστήματος.

8.2.2 Linear Chain System Extended (LCSE)

Δεδομένου ότι το μοντέλο LCS κλιμακώνεται ως προς το πλήθος των αντιδράσεων (m) και όχι ως προς το πλήθος των επηρεαζόμενων αντιδράσεων (D_{aver}), δημιουργήσαμε ένα καινούργιο μοντέλο για το οποίο ο χρήστης μπορεί να ορίζει το πλήθος των επηρεαζόμενων αντιδράσεων. Για να το πετύχουμε αυτό δημιουργούμε σύνολα αντιδράσεων στα οποία συμμετέχει ένα κοινό μοριακό είδος, που ενδεικτικά ονομάζουμε “υπόστρωμα” (Y). Ακολουθεί η γενική φόρμουλα για το μοντέλο LCSE 2^{ns} τάξης:

$$R_i: S_i + Y_L \xrightarrow{k_1} S_{i+1} \quad [i \bmod g \neq 0] \quad (11)$$

$$R_i: S_i + Y_L \xrightarrow{k_1} S_{i+1-g} \quad [i \bmod g = 0] \quad (12)$$

$$R_i': S_{i+1} \xrightarrow{k_2} S_i + Y_L \quad [i \bmod g \neq 0] \quad (13)$$

$$R_i': S_{i+1-g} \xrightarrow{k_2} S_i + Y_L \quad [i \bmod g = 0] \quad (14)$$

$$L = \left\lfloor \frac{i-1}{g} \right\rfloor + \left(\frac{m}{2} + 1 \right) \quad (15)$$

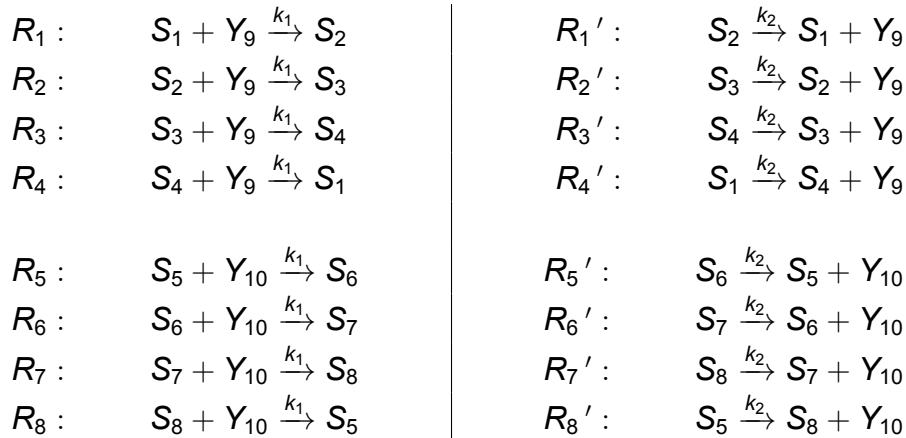
όπου $i = 1, 2, \dots, m/2$, με συνολικά $m/2$ αντιστρέψιμες αντιδράσεις, τις οποίες αναλύουμε σε m αντιδράσεις χρησιμοποιώντας τις σχέσεις (11) - (14). Ορίζουμε k_1 τον συντελεστή των αντιδράσεων 2^{ns} τάξης και k_2 των συντελεστή των αντιδράσεων 1^{ns} τάξης, και για τους οποίους πρέπει να ισχύει $k_1 \ll k_2$. Η παράμετρος g ορίζει το πλήθος των αντιδράσεων στις οποίες συμμετέχει το ίδιο “υπόστρωμα” (Y), και προκειμένου το μοντέλο να έχει ομαλή συμπεριφορά, θα πρέπει το πλήθος των αντιδράσεων m να είναι ακέραιο πολλαπλάσιο της παραμέτρου g . Επομένως, αν c είναι το πλήθος των ομάδων των αντιστρέψιμων αντιδράσεων με το ίδιο “υπόστρωμα” (Y) και συνεπώς το πλήθος των μοριακών ειδών Y , τότε ισχύει ότι $m/2 = c * g \Rightarrow c = m/(2 * g)$. Η παραπάνω γενική φόρμουλα επεκτείνεται με παρόμοιο τρόπο για τη δημιουργία του μοντέλου LCSE 3^{ns} τάξης.

Με βάση την παραπάνω ανάλυση, ορίζουμε τις σχέσεις (16) και (17), για το συνολικό πλήθος των μοριακών ειδών (n) του μοντέλου και το μέσο όρο των επηρεαζόμενων αντιδράσεων (D_{aver}) αντίστοιχα:

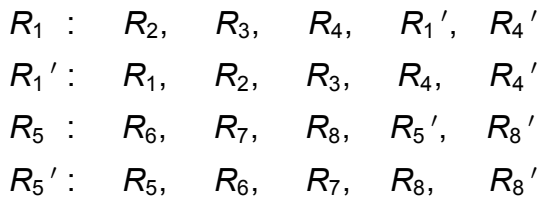
$$n = \frac{m}{2} + c \quad \Rightarrow \quad n = \frac{m * (g + 1)}{2 * g} \quad (16)$$

$$D_{aver} = g + 1 \quad (17)$$

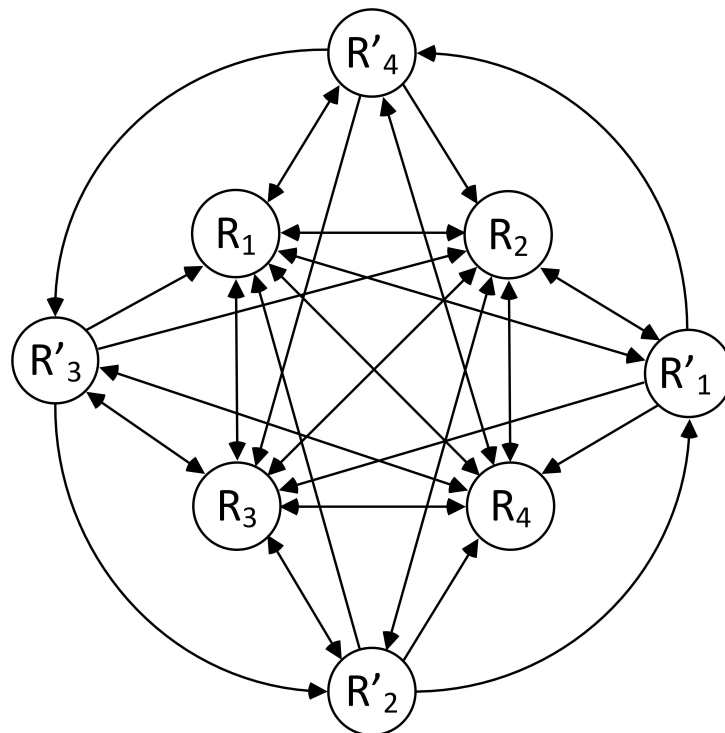
Ακολουθεί ένα ενδεικτικό παράδειγμα του μοντέλου LCSE για $m = 16$ αντιδράσεις και $g = 4$. Με βάση τις σχέσεις (16) και (17) προκύπτουν συνολικά $n = 16 \cdot (4+1) / 2 \cdot 4 = 80 / 8 = 10$ μοριακά είδη και μέσος όρος επηρεαζόμενων αντιδράσεων $D_{aver} = 4 + 1 = 5$ αντιδράσεις.



Στην συνέχεια, παρουσιάζουμε ενδεικτικά τις επηρεαζόμενες αντιδράσεις των αντιδράσεων R_1, R_1', R_5 και R_5' :



Η Εικόνα 28 απεικονίζει ένα κομμάτι του γράφου εξαρτήσεων του μοντέλου LCSE και συγκεκριμένα περιγράφει τις εξαρτήσεις των πρώτων 8 αντιδράσεων του παραπάνω παραδείγματος ($D_{aver} = 5$).



Εικόνα 28: Ο γράφος εξαρτήσεων (Dependencies Graph) του μοντέλου LCSE για τις πρώτες 8 αντιδράσεις με $D_{aver} = 5$

Προκειμένου να μετρήσουμε την απόδοση του συστήματός μας και πως τελικά επηρεάζεται από το μέσο όρο επηρεαζόμενων αντιδράσεων, εκτελέσαμε τέσσερις διαφορετικές διαμορφώσεις του μοντέλου LCSE με τα εξής χαρακτηριστικά:

• m :	2048	2048	2048	2046*
• g :	16	32	64	93
• c :	64	32	16	11
• n :	1088	1056	1040	1034
• D_{aver} :	17	33	65	94

*Παρατήρηση : Για το συγκεκριμένο πείραμα, θέλοντας να διατηρήσουμε το πλήθος των βιομοριακών αντιδράσεων λιγότερες από 2048, επιλέξαμε να ισχύει $g = 93$ και συνεπώς οι παράμετροι του μοντέλου αναλύονται ως εξής:

- Αντιδράσεις : $m = 2 * 11 * 93 = 2046$
- Μοριακά είδη : $n = 1023 + 11 = 1034$
- Μέσος όρος επηρεαζόμενων αντιδράσεων : $D_{aver} = 93 + 1 = 94$

Ακολουθούν οι παράμετροι της προσομοίωσης του βιομοντέλου LCSE για όλες τις διαμορφώσεις:

- T_{sim} : 1 δευτερόλεπτα (διάρκεια προσομοίωσης)
- T_{sam} : 0.01 δευτερόλεπτο (περίοδος δειγματοληψίας αποτελεσμάτων)
- R : 1 (στοχαστικές επαναλήψεις)
- Max Internal Steps: 1000K

Στον Πίνακα 23 αναφέρονται όλες οι γενικές παράμετροι των διαμορφώσεων του NRM SoC όπως αυτές ορίστηκαν πριν τη σύνθεση του συστήματος και οι οποίες καλύπτουν στο έπακρον τις ανάγκες της προσομοίωσης.

Πίνακας 23: Οι γενικές παράμετροι του NRM SoC όπως αυτές ορίστηκαν για τα πειράματα του LCSE.

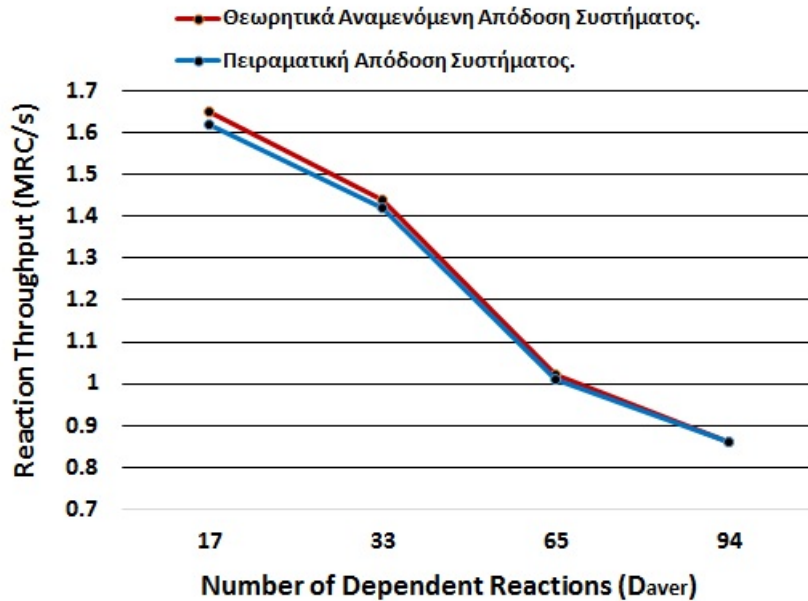
NRM SoC Configuration					
C	RT_SIZE	ST_SIZE	DG_SIZE	H	Frequency
1	2048	2048	17	4	180
1	2048	2048	33	4	180
1	2048	2048	65	4	160
1	2048	2048	129	4	160

Ακολουθεί ο πίνακας με τα αποτελέσματα της προσομοίωσης του μοντέλου για τις τέσσερις διαφορετικές διαμορφώσεις του μοντέλου LCSE.

Πίνακας 24: Performance Evaluation - LCSE models $D_{aver} = \{17, 33, 65, 94\}$

NRM SoC - LCSE $m = 2048$			
D_{aver}	T_{real} (Seconds)	Reactions Cycles (Millions)	Reactions Throughput (MRC/s)
17	6.31	10.244	1.62
33	7.21	10.241	1.42
65	10.11	10.237	1.01
94	11.91	10.227	0.86

Όπως και στο προηγούμενο πείραμα, στην Εικόνα 29 συγκρίνουμε για το μοντέλο LCSE την θεωρητικά αναμενόμενη διεκπεραιωτική ικανότητα (Reaction Throughput) του συστήματος με την πραγματική απόδοση του συστήματος. Στην συγκεκριμένη περίπτωση επιλέξαμε να διατηρήσουμε σταθερό το πλήθος των αντιδράσεων $m = 2048$ και να αυξήσουμε σταδιακά το πλήθος των επηρεαζόμενων αντιδράσεων $D_{aver} = \{17, 33, 65, 94\}$.



Εικόνα 29: Σύγκριση της αναμενόμενη διεκπεραιωτική ικανότητα με την πραγματική απόδοση του συστήματος, εκτελώντας το LCSE για $m = 2048$ και $D_{aver} = \{17, 33, 65, 94\}$

Στο διάγραμμα της Εικόνας 29 παρατηρούμε ότι η πραγματική απόδοση του συστήματος είναι ελάχιστα μικρότερη από την αναμενόμενη αλλά μειώνεται αισθητά καθώς αυξάνεται το πλήθος των επηρεαζόμενων αντιδράσεων. Η κλιμάκωση της απόδοσης δικαιολογείται δεδομένου ότι έχουμε επιλέξει να εξαρτάται από την παράμετρο D_{aver} , για την οποία ισχύει $D_{aver} \ll m$. Με αυτόν τον τρόπο το σύστημά μας είναι πολύ πιο αποδοτικό για μεγάλα μοντέλα με $m > 1024$ αντιδράσεις, έναντι των ήδη υπαρχόντων υλοποιήσεων. Ενώ, τέλος, αποδεικνύεται και πάλι η ορθότητα της θεωρητικής ανάλυσης των επιδόσεων που πραγματοποιείται στο Κεφάλαιο 7.1.

9. ΣΥΜΠΕΡΑΣΜΑΤΑ

Στα πλαίσια της παρούσας διπλωματικής εργασίας, σχεδιάσαμε και αναπτύξαμε επιτυχώς ένα Σύστημα σε Ψηφίδα (SoC) που να προσομοιώνει αποδοτικά πολύπλοκα δίκτυα βιομοριακών αλληλεπιδράσεων που περιγράφουν τη συμπεριφορά κυτταρικών υποσυστημάτων, υλοποιώντας τον δημοφιλή αλγόριθμο στοχαστικής προσομοίωσης Next Reaction Method (NRM) των Gibson και Bruck. Σε όλη τη διάρκεια της σχεδίασης, της υλοποίησης και της επικύρωσης του συστήματος ήρθαμε αντιμέτωποι με πληθώρα από προκλήσεις, οι οποίες όμως δεν αποτέλεσαν εμπόδιο στην επιτυχή ολοκλήρωση του συστήματος. Είμαστε σε θέση να δηλώσουμε ότι το σύστημα αυτό αποτελεί διεθνή καινοτομία[6] και ξερνά σε επιδόσεις και δυνατότητες κλιμάκωσης τις υπάρχουσες λύσεις.

Κατά το στάδιο της σχεδίασης, δεδομένου ότι ο αλγόριθμος NRM είναι εγγενώς σειριακός, προβληματιστήκαμε ιδιαίτερα στο να βρούμε τον πλέον κατάλληλο τρόπο να επωφεληθούμε από τις δυνατότητες που παρέχει μια υλοποίηση υλικού. Για τον λόγο αυτόν, επικεντρωθήκαμε ιδιαίτερα στη πλήρη διοχέτευση (pipelining) για τη μείωση της καθυστέρησης (latency) του διαύλου επεξεργασίας (datapath) που απαιτείται για την εύρεση της επόμενης νικητήριας αντίδρασης, χρησιμοποιώντας μονάδες παράλληλης σύγκρισης δεδομένων (CoT), και υλοποιώντας τις μαθηματικές πράξεις με υψηλής απόδοσης μονάδες (RU). Δόθηκε ιδιαίτερη σημασία στη παράλληλη λειτουργία αλλά και στο συγχρονισμό των μονάδων CoT και RU με τρόπο που να επιτρέπει, μέσω της επιλογής του ύψους H του δένδρου, η σταθερή καθυστέρηση της RU να “κρύβει” (latency hiding) τη καθυστέρηση του CoT, ώστε να ελαχιστοποιούνται οι απαιτούμενοι κύκλοι ρολογιού ενός κύκλου αντιδράσεων (RC latency) και να αποφεύγεται η χρήση επιπλέον μονάδων ελέγχου. Εν τέλει, το σύστημά μας είναι σε θέση να εκτελεί μεγάλης κλίμακας βιομοντέλα χωρίς να επηρεάζεται η απόδοσή του από το συνολικό πλήθος των αντιδράσεων (m) αλλά να εξαρτάται αποκλειστικά και μόνο από το μέσο όρο των επηρεαζόμενων αντιδράσεων (D_{aver}) του βιομοντέλου, κάτι που αποτελεί καινοτόμο προσέγγιση στο πρόβλημα.

Επιπλέον, έχοντας υπόψη την πολυπλοκότητα των δύο αλγορίθμων (FRM, NRM) αλλά και την τελική απόδοση των συστημάτων που αναπτύξαμε για κάθε έναν από αυτούς, μας ενδιαφέρει να μελετήσουμε τη βέλτιστη επιλογή αλγορίθμου με βάση την πολυπλοκότητα του βιομοντέλου: πλήθος αντιδράσεων (m) και μέσος όρος επηρεαζόμενων αντιδράσεων (D_{aver}). Μελλοντικός μας στόχος είναι να μπορούμε να εκτελούμε στοχαστική προσομοίωση επιλέγοντας τον πλέον κατάλληλο αλγόριθμο για δοθέν βιομοντέλο με στόχο την μεγιστοποίηση της απόδοσης. Αυτό θα γίνεται αυτόματα από το σύστημα μας, χωρίς ο τελικός χρήστης να χρειάζεται να αναλύσει τα χαρακτηριστικά της προσομοίωσης που επιθυμεί να εκτελέσει, προκειμένου να αποφασίσει την βέλτιστη λύση.

Τέλος, μελετώντας τα στατιστικά υλοποίησης (Ενότητα 7.2) καταλήγουμε στο συμπέρασμα ότι με ένα μεγαλύτερο FPGA το σύστημά μας θα είναι σε θέση να καλύψει μοντέλα με πλήθος αντιδράσεων μεγαλύτερο ακόμα και από 4096 ($m > 4096$). Με βάση τη σημερινή τεχνολογία το XC7K325T θεωρείται ένα μετρίου μεγέθους FPGA, γεγονός που περιορίζει την περαιτέρω κλιμάκωση του NRM SoC. Συνεπώς, κατά το στάδιο της σύνθεσης του NRM SoC μας απασχόλησε ιδιαίτερα το μέγιστο επιτρεπόμενο πλήθος αντιδράσεων και κατά πόσο επηρεάζεται η συχνότητα λειτουργίας του συστήματος όσο αυξάνονται οι απαιτούμενοι πόροι. Πρόκειται για ένα ενδιαφέρον ζήτημα που χρήζει περαιτέρω διερεύνησης, χρησιμοποιώντας μεγαλύτερου μεγέθους και καλύτερης τεχνολογίας FPGAs. Παρόλο που η συχνότητα μειώνεται ελαφρώς όσο το σχέδιο μεγαλώνει, η μείωσή της δεν είναι δραστική λόγω της επιθετικής διοχέτευσης (pipelining) που χρησιμοποιήθηκε στη ανάπτυξη του πλήρως παραμετρικού σχεδίου.

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Βιολογία Συστημάτων	Systems Biology
Γράφος Εξαρτήσεων	Dependencies Graph
Κεντρικοποιημένης Δομής Δεδομένων	Shared Data Structure
Κύκλος Αντιδράσεων	Reaction Cycle
Μεταγραφή	Transcription
Νικητήρια Αντίδραση	Winner Reaction
Ρυθμιστικό Δίκτυο	Gene Regulatory Network
Στοχαστική Προσομοίωση	Stochastic Simulation
Συνήθεις Διαφορικές Εξισώσεις	Ordinary Differential Equations
Σύστημα σε Ψηφίδα	System on Chip
Τεράστιος Όγκος Δεδομένων	Big Data

ΣΥΝΤΜΗΣΕΙΣ, ΑΡΚΤΙΚΟΛΕΞΑ ΚΑΙ ΑΚΡΩΝΥΜΙΑ

RC	Reaction Cycle
CoT	Comparator Tree
CSV	Comma Seperated Values
DG	Dependencies Graph
DG MS	Dependencies Graph Memory System
DM	Direct Method
FPGA	Field Programmable Gate Array
FRM	First Reaction Method
IIS	Input Inteface System
MT	Mersenne Twister
NoC	Network on Chip
NPCU	NRM Processing Control Unit
NPE	NRM Processing Element
NPU	NRM Processing Element
NRM	Next Reaction Method
ODEs	Ordinary Differential Equations
OIS	Output Interface System
PISO	Parallel in Serial Out
RT MS	Reaction Table Memory System
RU	Recalculation Unit
SCU	Simulation Control Unit
SoC	System on Chip
SPU	Species Update Unit
SSA	Stochastic Simulation Algorithm
ST MS	Species Table Memory System
T_{sam}	Sampling Period
T_{sim}	Simulation Time

ΑΝΑΦΟΡΕΣ

- [1] M. A. Gibson and J. Bruck, “Efficient Exact Stochastic Simulation of Chemical Systems with Many Species and Many Channels,” *The Journal of Physical Chemistry A*, vol. 104, no. 9, pp. 1876–1889, 2000.
- [2] Myers, Chris J., *Engineering genetic circuits*, ser. Chapman & Hall/CRC mathematical and computational biology series. Boca Raton: CRC Press, 2010.
- [3] E. Ouzounoglou *et al.*, “In silico modeling of the effects of alpha-synuclein oligomerization on dopaminergic neuronal homeostasis,” *BMC Systems Biology*, vol. 8, no. 1, p. 54, 2014. [Online]. Available: <http://www.ebi.ac.uk/biomodels-main/BIOMD0000000559>
- [4] E. Ouzounoglou *et al.*, “Modeling of alpha-synuclein effects on neuronal homeostasis,” Model of the Month, *EMBL-EBI BioModels Database*, Available: <https://www.ebi.ac.uk/biomodels-main/static-pages.do?page=ModelMonth/2015-03>, [Mar, 2015]
- [5] M. Hucka *et al.*, “The Systems Biology Markup Language (SBML): a medium for representation and exchange of biochemical network models,” *Bioinformatics*, vol. 19, no. 4, pp. 524–531, 2003.
- [6] E. Koutsouradis *et al.*, “Scalable FPGA accelerator of the NRM algorithm for efficient stochastic simulation of large-scale biochemical reaction networks,” in *Digital System Design (DSD), 18th Euromicro Conference on*, August 2015.
- [7] Wikipedia, “Biology — Wikipedia, The Free Encyclopedia,” [Online; accessed 02-July-2015]. [Online]. Available: www.en.wikipedia.org/wiki/Biology
- [8] Wikipedia, “Template: Branches of biology — Wikipedia, the free encyclopedia,” [Online; accessed 02-July-2015]. [Online]. Available: www.e.wikipedia.org/wiki/Template:Branches_of_biology
- [9] S. Fields, “The interplay of biology and technology,” *Proceedings of the National Academy of Sciences*, vol. 98, no. 18, pp. 10 051–10 054, 2001.
- [10] Wikipedia, “Computational biology — Wikipedia, the free encyclopedia,” [Online; accessed 02-July-2015]. [Online]. Available: https://en.wikipedia.org/wiki/Computational_biology
- [11] Wikipedia, “Bioinformatics — Wikipedia, the free encyclopedia,” [Online; accessed 02-July-2015]. [Online]. Available: <https://en.wikipedia.org/wiki/Bioinformatics>
- [12] Wikipedia, “Data mining — Wikipedia, the free encyclopedia,” [Online; accessed 02-July-2015]. [Online]. Available: https://en.wikipedia.org/wiki/Data_mining
- [13] Wikipedia, “Control theory — Wikipedia, the free encyclopedia,” [Online; accessed 02-July-2015]. [Online]. Available: https://en.wikipedia.org/wiki/Control_theory
- [14] Wikipedia, “Network analysis (electrical circuits) — Wikipedia, the free encyclopedia,” [Online; accessed 02-July-2015]. [Online]. Available: [https://en.wikipedia.org/wiki/Network_analysis_\(electrical_circuits\)](https://en.wikipedia.org/wiki/Network_analysis_(electrical_circuits))
- [15] Wikipedia, “Artificial intelligence — Wikipedia, the free encyclopedia,” [Online; accessed 02-July-2015]. [Online]. Available: https://en.wikipedia.org/wiki/Artificial_intelligence
- [16] Wikipedia, “Pattern recognition — Wikipedia, the free encyclopedia,” [Online; accessed 02-July-2015]. [Online]. Available: https://en.wikipedia.org/wiki/Pattern_recognition
- [17] A. Finkelstein *et al.*, “Computational challenges of systems biology,” *Computer*, vol. 37, no. 5, pp. 26–33, May 2004.
- [18] A. P. Heath and L. E. Kavradi, “Computational challenges in systems biology,” *Computer Science Review*, vol. 3, no. 1, pp. 1 – 17, 2009.
- [19] M. Tomita, “Whole-cell simulation: a grand challenge of the 21st century,” *Trends in*

- Biotechnology*, vol. 19, no. 6, pp. 205–210, June 2001.
- [20] D. T. Gillespie, “Stochastic Simulation of Chemical Kinetics,” *Annual Review of Physical Chemistry*, vol. 58, no. 1, pp. 35–55, 2007.
- [21] Wilkinson, Darren J, *Stochastic modelling for systems biology*. CRC press, 2011.
- [22] Wikipedia, “Markov chain — Wikipedia, the free encyclopedia,” [Online; accessed 02-July-2015]. [Online]. Available: https://en.wikipedia.org/wiki/Markov_chain
- [23] M. Yoshimi *et al.*, “An FPGA Implementation of a Data-Driven Stochastic Biochemical Simulator with the Next Reaction Method,” in *IEEE Field Programmable Logic and Applications*, Aug 2007, pp. 254–259.
- [24] M. Yoshimi *et al.*, “Practical implementation of a network-based stochastic biochemical simulation system on an FPGA,” in *IEEE Field Programmable Logic and Applications (FPL)*, Sept 2008, pp. 663–666.
- [25] D. Thomas and H. Amano, “A fully pipelined FPGA architecture for stochastic simulation of chemical systems,” in *IEEE Field Programmable Logic and Applications (FPL)*, Sept 2013, pp. 1–7.
- [26] M. Yoshimi, “A Study on Monte-Carlo Biochemical Simulation for Field-Programmable Gate Arrays,” Doctoral Thesis, School of Science for Open and Environmental Systems Graduate School of Science and Technology Keio University, Mar 2009.
- [27] H. Yamada *et al.*, “A Modular Approach to Heterogeneous Biochemical Model Simulation on an FPGA,” in *Reconfigurable Computing and FPGAs, 2009. ReConFig '09. International Conference on*, Dec 2009, pp. 125–130.
- [28] M. Yoshimi *et al.*, “An FPGA Implementation of High Throughput Stochastic Simulator for Large-Scale Biochemical Systems,” in *Field Programmable Logic and Applications, 2006. FPL '06. International Conference on*, Aug 2006, pp. 1–6.
- [29] N. Azizi *et al.*, “Reconfigurable molecular dynamics simulator,” in *Field-Programmable Custom Computing Machines, 2004. FCCM 2004. 12th Annual IEEE Symposium on*, April 2004, pp. 197–206.
- [30] J. F. Keane, “A compiled accelerator for biological cell signaling simulations,” in *Proceedings of the International Symposium on Field-Programmable Gate Arrays FPGA*, 2004, pp. 233–241.
- [31] O. Hazapis and E. Manolakos, “Scalable FRM-SSA SoC Design for the Simulation of Networks with Thousands of Biochemical Reactions in Real Time,” in *IEEE Field Programmable Logic and Applications (FPL)*, Sept 2011, pp. 459–463.
- [32] E. Kouskoumvekakis, D. Soundris, and E. Manolakos, “Many-core CPUs can deliver scalable performance to stochastic simulation of large-scale biochemical reaction networks,” in *High Performance Computing Simulation (HPCS), 2015 International Conference on*, July 2015.
- [33] P. Georgios, “Σχεδίαση και FPGA Υλοποίηση IP πυρήνων και SoCs για παράλληλη Στοχαστική Προσομοίωση Βιολογικών Δικτύων,” Master Thesis, University of Athens, Informatic and Telecommunications, July 2015.
- [34] Gillespie, Daniel T., “Exact stochastic simulation of coupled chemical reactions,” *The Journal of Physical Chemistry*, vol. 81, no. 25, pp. 2340–2361, 1977.
- [35] Xilinx.com, “All Programmable Technologies from Xilinx Inc.” 2015, [Accessed: 5-July-2015]. [Online]. Available: <http://www.xilinx.com/>
- [36] Xilinx.com, “Kintex7 Product Table,” [Accessed: 5-July-2015]. [Online]. Available: http://www.xilinx.com/publications/prod_mktg/Kintex7-Product-Table.pdf
- [37] Xilinx.com, “ISE Design Suite,” [Accessed: 5-July-2015]. [Online]. Available: <http://www.xilinx.com/products/design-tools/ise-design-suite.html>
- [38] Xilinx.com, “Running the Generate Programming File Process for FPGAs,”

- [Accessed: 5-July-2015]. [Online]. Available: http://www.xilinx.com/support/documentation/sw_manufactures/xilinx11/ise_p_generate_fpga_programming_file.htm
- [39] Xilinx.com, “iMPact Overview,” [Accessed: 5-July-2015]. [Online]. Available: http://www.xilinx.com/itp/xilinx10/isehelp/pim_c_overview.htm
- [40] B. Vermeulen, T. Waayers, and S. Bakker, “IEEE 1149.1-compliant access architecture for multiple core debug on digital system chips,” in *Test Conference, 2002. Proceedings. International*, 2002, pp. 55–63.
- [41] Wikipedia, “Comma-separated values — Wikipedia, the free encyclopedia,” [Online; accessed 02-July-2015]. [Online]. Available: https://en.wikipedia.org/wiki/Comma-separated_values
- [42] Arm.com, “AMBA Specifications - ARM,” [Accessed: 5-July-2015]. [Online]. Available: <http://www.arm.com/products/system-ip/amba-specifications.php>
- [43] T. Henriksson and P. van der Wolf, “TTL Hardware Interface: A High-Level Interface for Streaming Multiprocessor Architectures,” in *Embedded Systems for Real Time Multimedia, Proceedings of the 2006 IEEE/ACM/IFIP Workshop on*, Oct 2006, pp. 107–112.
- [44] M. Matsumoto and T. Nishimura, “Mersenne Twister: A 623-dimensionally Equidistributed Uniform Pseudo-random Number Generator,” *ACM Trans. Model. Comput. Simul.*, vol. 8, no. 1, pp. 3–30, Jan. 1998.
- [45] “IEEE Standard for Floating-Point Arithmetic,” *IEEE Std 754-2008*, pp. 1–70, Aug 2008.
- [46] Xilinx.com, “7 Series FPGAs Overview,” [Accessed: 5-July-2015]. [Online]. Available: http://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf
- [47] S. Hoops *et al.*, “COPASI - a COmplex PATHway Simulator,” *Bioinformatics*, vol. 22, no. 24, pp. 3067–3074, 2006.
- [48] Xilinx.com, “ISE Simulator (ISim),” [Accessed: 5-July-2015]. [Online]. Available: <http://www.xilinx.com/tools/isim.htm>
- [49] Xilinx.com, “Behavioral Simulation,” [Accessed: 5-July-2015]. [Online]. Available: http://www.xilinx.com/support/documentation/sw_manufactures/xilinx11/platform_studio/ps_c_sim_behavioral_simulation.htm
- [50] Xilinx.com, “Timing Simulation,” [Accessed: 5-July-2015]. [Online]. Available: http://www.xilinx.com/support/documentation/sw_manufactures/xilinx11/ise_c_simulation_timing.htm
- [51] M. V. Schneider, *In Silico Systems Biology*. Totowa, NJ: Humana Press, 2013.